

第1章

USB インターフェースによる汎用 I/O の実現

…使用する USB デバイスとファームウェアの概要

本キットに添付のボード/ソフトウェアは、簡単にUSBバスを自作機器用のI/Oインターフェースとして利用できるようにするため、USB汎用インターフェース・ボード（UCT-203）、専用ファームウェア（EzFirm/FX2）、デバイス・ドライバ（EZUSB.SYS）などをセットにした、USB汎用I/Oボード・キットです。

1-1 USBでI/O制御を行う

●USBの壁

パソコンを使ってちょっとした機器を制御しようとした場合、従来はパラレル・ポートやシリアル・ポートを使うことが一般的でした。ところが最近のノートパソコンなどではこれらのポートをもたず、周辺装置の接続はUSBのみというものが珍しくありません。

USBは、電源の供給がホスト側から行われることや、データ転送速度がこれらのレガシィなポートよりもはるかに速いこと、ハブを使うことで多数の機器を同時に接続できることなど利点の多いバスであり、レガシィ・ポートからUSBへの移行はパソコンや周辺機器メーカーとしては必然であるとも言えるでしょう。

ところが、USBは単純なI/Oポートではないため、パソコンに自作の周辺装置を接続したいというときにはUSB自体が大きな壁となってしまいます。図1に示すように、以前は単純にI/Oポートに直接アクセスするだけでよかったのが、USB経由になったとたんに面倒極まりないことになってきます。

USBターゲットを製作するためには、USBプロトコルの理解や、USBコントローラの入手とデバイスの理解、ボードの準備やファームウェアの作成が必要です。ファームウェアを作成するためにはコンパイラなどの開発ツールが必要です。サイプレス社が推奨しているKeil社の製品版コンパイラは約30万円もします。

ターゲット側だけでは動きませんので、Windowsドライバを作る必要もあります。Windows用のドライバ作成にはVisualC++のほかにDDKも必要ですが、これにはMicrosoftのMSDN（マイクロソフト・デベロッパーズ・ネットワーク）を購入し、会員登録をしなければなりません。

またUSB機器として認識させるためには機器に固有のベンダIDが必須ですが、これは勝手に使ってよいものではなく、USBの管理を行っているUSB.ORGのインプリメンターズ・フォーラムに会費を払って入会し、取得する必要があります。

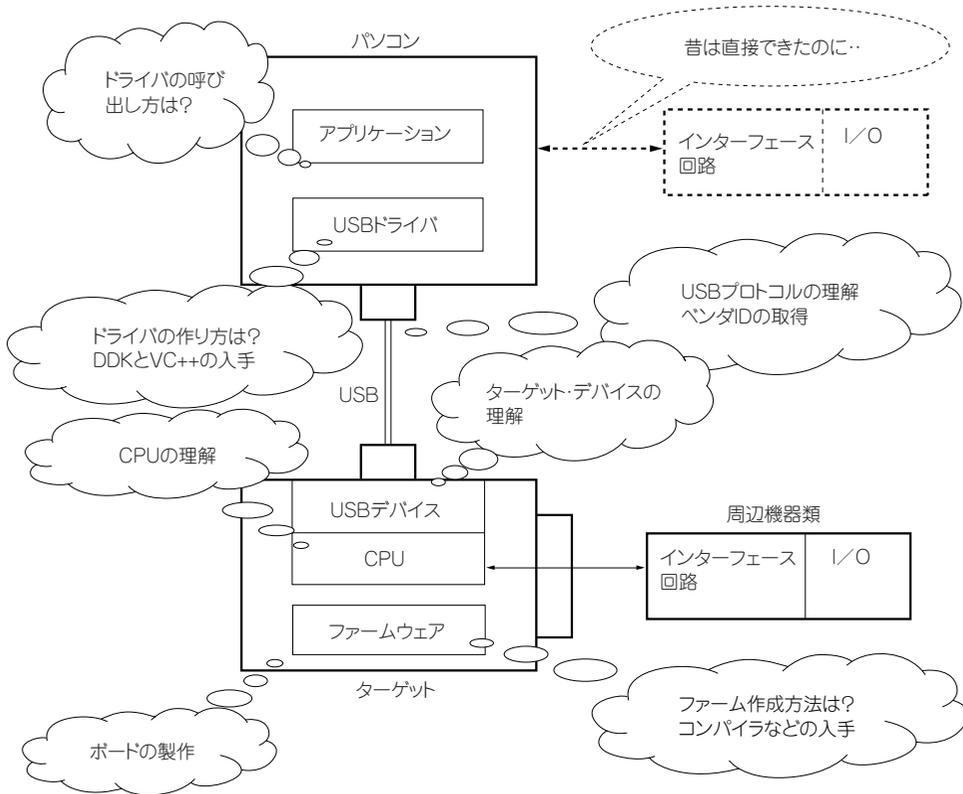


図1 USBでインターフェースするときの壁

USBターゲット機器を作ることが目的ならばまだしも、単にリレーやLED、スイッチなどを付けたいという程度のことのために、これだけの努力と出費を強いられるのは辛いものがあります。

●汎用I/Oを実現するUSBインターフェース・ボード

そこで、本キットではUSBはあくまでも自作機器などをコントロールするためのI/O（入出力）と考え、USBインターフェース対応の汎用I/Oボードを用意し、この先に自作機器を接続するという考え方にしました。つまり、パソコンの先にUSBから汎用的に利用できるI/Oポートが増設され、このポートをアクセスするためのライブラリなどが提供された状態になるわけです。

図2に示したように、本キットを利用することで、VisualBasicやVisualC++などから簡単にボード（UCT-203）の先に繋いだ機器を制御することができるようになります。また、簡単な動作テストなら、サイプレス社の開発ツールである「EZ-USBコントロール・パネル」（付属CD-ROM内の開発ツールをインストールするとき一緒にインストールされる）を利用するのもよいでしょう。

プログラミングのためのツールをもっていない場合には、マイクロソフト社のサイトで無償配布されている

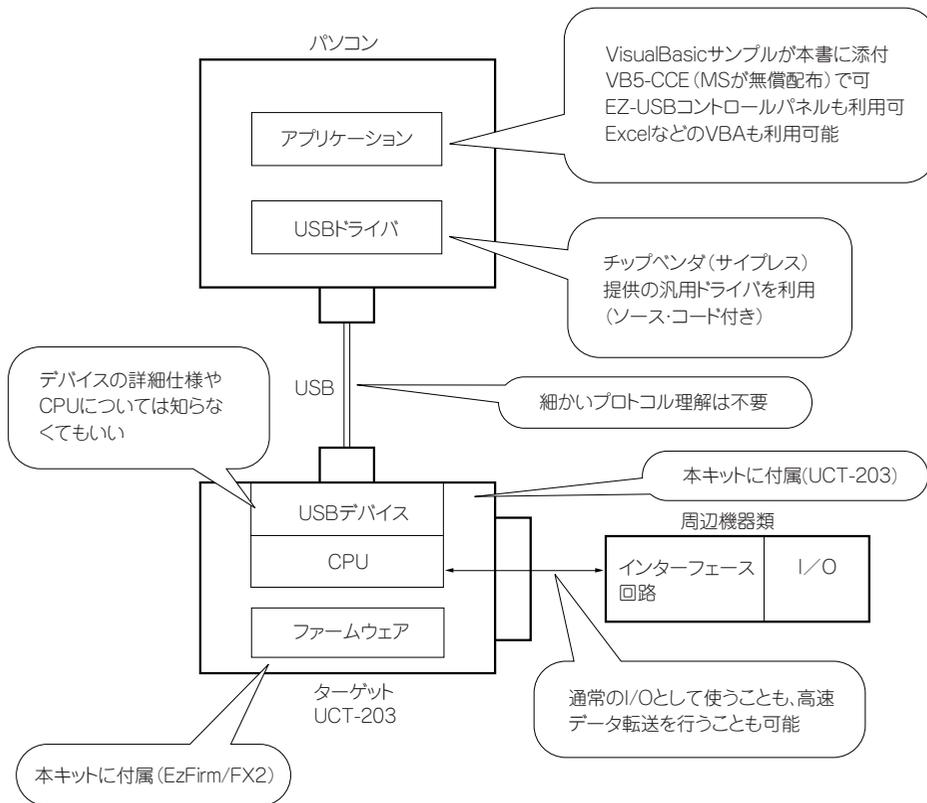


図2 本書添付のセットでUSBを使う

VisualBasic Control Creation Edition (VB5-CCE) が利用可能です。本書付属のサンプル・プログラムは、このCCE版で動作を確認しています。

また、本書のサンプル・プログラムとしてVB用に用意した標準モジュール (*.BASファイル) はExcelがマクロ言語として使用しているVBA (Visual Basic for Application) にインポートして利用することができます。Officeがプレインストールされているパソコンも一般的になっていますので、新たにプログラミング・ツールを用意せずに済むこのような方法も便利かもしれません。ただし、VBAはVBとの互換性はかなりの程度あるとはいえ、ドライブ・リスト・ボックスやタイマ・コントロールなどが無いなど、一部機能が削られています。

本書の応用編で紹介するサンプル・プログラムではタイマを使用していますので、これらはそのまま動かすことはできませんが、単純にボタンを押したらLEDが点滅するといった程度のアプリケーションであれば、VBAでも充分です。図3は、実際にExcelで作った簡単なON/OFFコントロールとスイッチ読み込みのマクロを実行した画面です (ファイルはExcelDIO.xlsで、ハードウェアは応用編で紹介している早押し判定器を利用した)。

ExcelのVBAからI/Oが制御できることを利用して、たとえば読み出したデータをExcelに取り込んで処理させるといった使いかたも面白いでしょう。

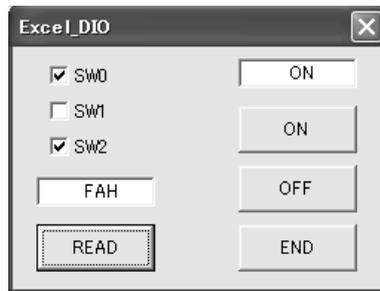


図3 ExcelのVBAで作ったデジタル出力の実行画面

このように、パラレル・ポートやシリアル・ポートなどを使っていたときと同様に、アプリケーション・レベルから簡単にI/O操作が行えるだけでなく、ケーブル経由での電源供給やPlug & Play、パラレル・ポートをしのご高速データ転送といった、USBならではの恩恵にもあずかれます。

また、本書では扱いませんでしたが、添付のUSBインターフェース・ボードは汎用I/O専用ではなく、単にEZ-USB/FX2が乗ったボードとして利用することができます。自分でファームウェアを作ってダウンロードすれば、オリジナルのファームウェアをもったUSB機器として動作させることも可能です。汎用I/Oとしての動作に飽き足らなくなったら、USBの学習を兼ねてファームウェアやドライバを独自に作成して動かしてみるのもよいかもしれません。

●キットの構成

本セットは以下のような構成になっています。②～⑤は付属CD-ROMに収録されています。

- ① USB2.0汎用インターフェース・ボード (UCT-203)
- ② 汎用I/Oファームウェア (EzFirm/FX2)
- ③ EZ-USBシリーズ用汎用ドライバ (EZUSB.SYS；サイプレス社提供)
- ④ EZ-USBコントロール・パネル (サイプレス社提供，EzFirm/FX2ダウンロード用に使用)
- ⑤ VisualBasicアプリケーション (応用編のサンプル・プログラム)

次にそれぞれの要素について簡単に説明しておきます。

① USB2.0汎用インターフェース・ボード (UCT-203)

使用しているUSBコントローラは、8ビットCPUコアやプログラム・メモリなどを内蔵したサイプレス社のワンチップUSB2.0コントローラ、EZ-USB/FX2の56ピン・バージョンです。デバイスの型番はCY7C68013-56です。FX2ファミリのなかではピン数の少ないタイプになりますが、三つの8ビットI/Oポート (PORTA, PORTB, PORTD) をもっており、スレーブFIFOモードやGPIFモードを使った高速転送にも対応しています。

② 汎用I/Oファームウェア (EzFirm/FX2)

UCT-203を汎用のI/Oボードとして動かすためのファームウェアです。EZ-USB/FX2内のSRAMにダウンロー

ドし、USBターゲットとしての動作を処理する8051コアのオブジェクトです。

単純なI/Oポートのリード/ライトだけでなく、高速データ転送モードにも対応しています。本ファームウェアはオブジェクト（ダウンロード・ファイル）のみでの提供となります。また、本キットに添付のEzFirm/FX2は付属のボード（UCT-203）専用です。他のボードなどへの流用は禁止します。

③ EZ-USB シリーズ用汎用ドライバ (EZUSB.SYS)

サイプレス社がソース・コードを含めて無償提供しているWindows98/2000/XPで利用可能なEZ-USBファミリー用の汎用ドライバです。

ファームウェアのEzFirm/FX2は、ベンダID、プロダクトIDをEZ-USB/FX2のデフォルト値と同一にしているため、起動後もこの汎用ドライバをそのまま利用することができます。ドライバがもっているAPIは、ドライバのソース・コード（インストールしたフォルダ内を参照）を見ればわかるとおり、IOCTLを使ったものになっています。汎用I/Oとして使う場合に利用するIOCTLは限られていますので、本書応用編のVisualBasicのサンプル・ソース（付属CD-ROMに収録）を見ていただくほうがわかりやすいかもしれません。

④ EZ-USB コントロール・パネル

EzFirm/FX2は、ボード（UCT-203）には書き込まれていませんので、サイプレス社が提供しているEZ-USBコントロール・パネルというデバッグ用のソフトウェア・ツールを使ってUSB経由でEZ-USB/FX2の内部メモリ（SRAM）にダウンロードします。内部メモリはSRAMベースですので、電源投入後、またはリセットをかけたあとに毎回ダウンロードしなおす必要があります。そのため、UCT-203ではシリアルEEPROMを搭載しており、そちらに書き込んでおくこともできるようになっています（p.10を参照）。

EZ-USBコントロール・パネルは非常に多くの機能をもっており、デバッグ支援ツールとしても役立つことでしょう。

⑤ VisualBasic アプリケーション

応用編で紹介するサンプル・プログラムは、VB5-CCE（Control Creation Edition）で動作確認を行っています。VB5-CCEは付属CD-ROMには収録されていませんが、以下のURLからダウンロード可能です。

インストーラ▼

<http://download.jp.microsoft.com/developer/vbasic/controls/vb5ccein.exe>

ヘルプ・ファイル▼

<http://download.microsoft.com/download/vbcc/Help/4.71.0121.0/W9XNT4/JA/CCEHELP.EXE>

なお、ドライバのAPIを呼び出すコードを毎回書くのは少々煩雑であることから、サンプル・プログラムではこの部分をラップするようなサブルーチンを用意し、標準モジュールとしてまとめてあります。

これを使うと、たとえばPIOモードでのポート・ライトは、

EZ_PIOWrite(ポート番号, データ)

読み出しは、

`EZ_PIORead` (ポート番号)

という具合に、単なるI/Oアクセスと同じ感覚で使うことができますようになります。

この標準モジュールはVBAにインポートすることもできますので、Excelなどからコントロールしたい場合にも便利でしょう。

1-2 添付ボードUCT-203の使いかた

本キットに添付されているUSB汎用インターフェース・ボード (UCT-203) の基本的な使いかたは次のようになります。本書の「プロlogue」で同様の解説を行いましたので、すでにソフトウェアのインストールやボードの接続確認が済んでいる方は、以下については読み飛ばしていただいてもかまいません。

① EZ-USB開発ツールのインストール

EZ-USB開発ツールのなかには、汎用USBドライバ、ファームウェアのダウンロードなどに使うEZ-USBコントロール・パネルなどが含まれていますので、これをインストールします。付属CD-ROMの中にある

`EZ-USB_devtools_version_261700.exe`

を実行してください。インストールが行われます。

② Windowsアプリケーションの開発と制御対象のハードウェア

単に付属CD-ROM内のサンプル・プログラムの実行ファイルを動かしたり、EZ-USBコントロール・パネルを使うだけならばこのステップは不要です。サンプル・プログラムをベースにオリジナルのソフトウェアを作るときにはVisualBasicが手軽で良いでしょう。サンプルのコードを元にVisualC++などで作成されてももちろんかまいません。ボードを制御するための特別なDLLなどは不要ですので、①のステップでインストールが完了していれば、他のソフトウェアのインストールは不要です。

VisualBasicの製品版をもっていない場合には、Microsoftのサイトで無償で配布されているVB5-CCE (Control Creation Edition) を使うこともできます。VB5-CCEは実行ファイル (* .EXEファイル) が作成できないなどの制約はありますが、製品版がすでに.NETに移行した今となってはかなり古いバージョンということになりますが、本書のサンプル・アプリケーション程度のもを作るうえでは十分な機能をもっています。また、単純にボタンが押されたらI/Oを行う程度であれば、ExcelのVBAなどを利用することもできます。

制御対象のハードウェアとしては、さまざまなものがあるでしょうから一概に言うことはできませんが、本書の応用編を参考にして簡単なものから試してみるのがよいと思います。本書のAppendix A (pp.75~83) に、UCT-203と一般的なハードウェアとのインターフェース回路例を示してあります。

③ 添付ボードの接続

本キット添付のUSB汎用インターフェース・ボード (UCT-203) をパソコンのUSBポートに接続します。接続先はUSB1.1/2.0のいずれでもかまいません。接続するとデバイス認識画面が出て、自動的にドライバが組み込まれます。

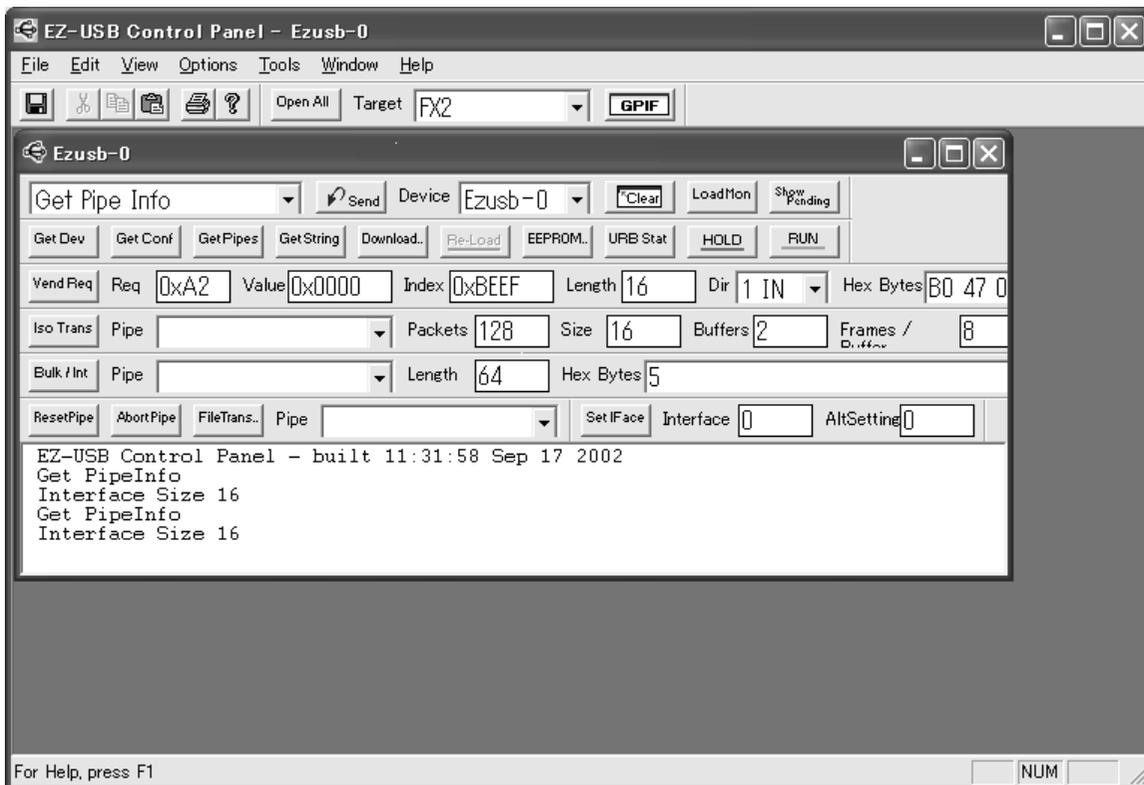


図4 ボードを接続した直後にEZ-USBコントロール・パネルを起動した状態

④ EZ-USBコントロール・パネルの起動

ドライバの自動組み込み画面が消えたらEZ-USBコントロール・パネルを起動します。Windowsのスタート・メニューから、

スタート→プログラム→Cypress→USB→EZ-USB Control Panel

として起動します。

起動すると次のように、「Ezusb-0」が認識されてチャイルド・ウィンドウが出ます。図4は「GetPipes」ボタンを押したときの状態です。データ用のパイプ（エンド・ポイント）はない状態になっています。

⑤ ファームウェア（EzFirm/FX2）のSRAMへのダウンロード

EZ-USBコントロール・パネルの一番上の右にある「GPIF」ボタンの隣のTargetリスト・ボックスで、「FX2」を選択します。ここがFX2になっていないと、ダウンロードできたように見えてもまったく動きませんので注意してください。

選択したら、チャイルド・ウィンドウの中の「Download」ボタンを押して、付属CD-ROMに収録されている

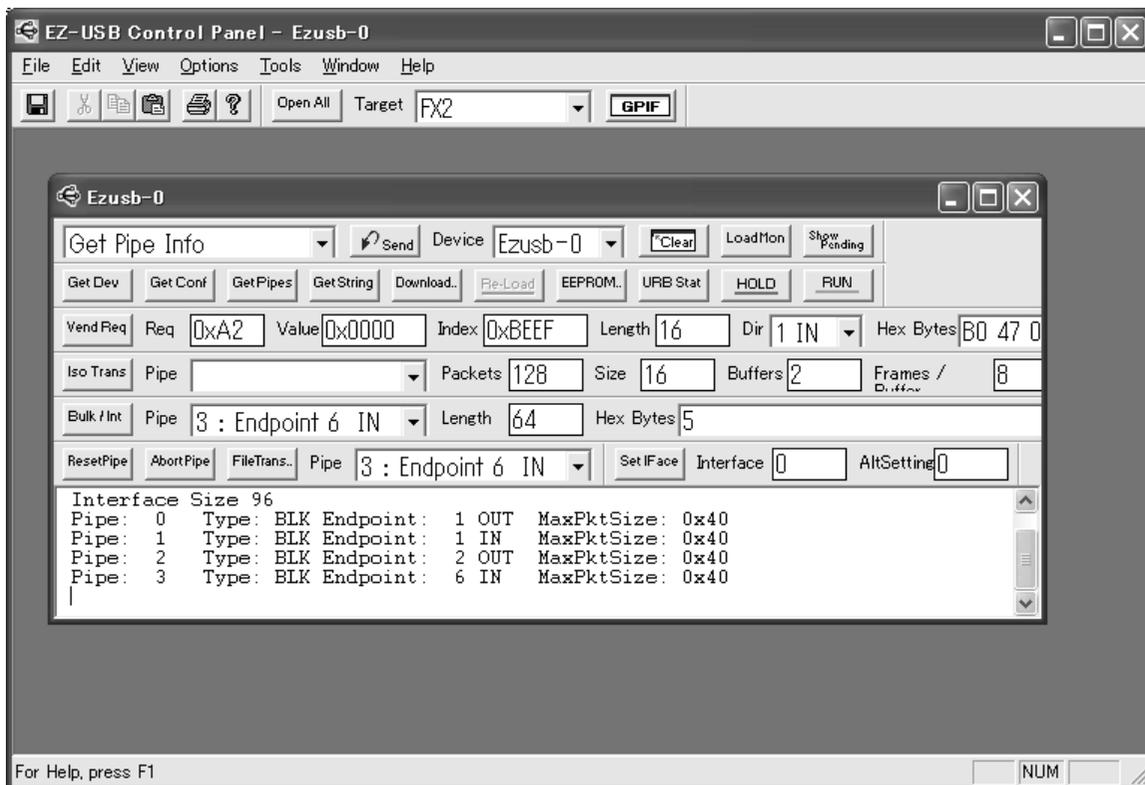


図5 EzFirm/FX2ダウンロード後に起動した状態

EZFIRMFx2.HEX ファイルを選択します。

ダウンロードが終わるとファームウェア EzFirm/FX2が起動し、いったんデバイスが切り離され、再接続されたような動作をします。ベンダID、プロダクトIDはデフォルトUSBデバイスと同じですので、画面上では同じデバイスが再接続されたようになります。

⑥ ダウンロード完了確認

ダウンロード完了後、EZ-USBコントロール・パネルを終了して再度起動するか、あるいはチャイルド・ウィンドウを閉じてからメニューで「File→Open All Devices」を選択すると、図5のようにデータ用のエンド・ポイントが4個あるデバイスとして認識されます。図はUSB1.1ポートに接続したときのものです。USB2.0ポートに接続した場合には、Pipe2とPipe3のMaxPktSizeが0x200（512バイト）になります。

⑦ アプリケーションの起動

これで、VisualBasicアプリケーションから、USBインターフェース・ボードの先に接続した機器のコントロー

ルができるようになります。異常動作などでボードをリセットしたり、USBケーブルの着脱を行った場合には、再度EzFirm/FX2のダウンロードからやりなおしてください。

なお、本書の「プロローグ」の「ファームウェアの書き込み」で示したように、UCT-203上のシリアルEEPROMにファームウェアを書き込んであれば、電源ONですぐに使用することができるので便利です。

1-3 USB2.0 コントローラ EZ-USB/FX2

本キット添付のファームウェア EzFirm/FX2を使うかぎり、USB コントローラ EZ-USB/FX2の詳細を理解する必要はありませんが、おおまかな仕様や機能については理解しておくほうが、ボードを利用するうえでも便利であると思いますので、ここでEZ-USB/FX2の概要について説明しておきます。

詳細についてはデバイスのドキュメントを読んでください（インストール先のフォルダ内を参照）。EZ-USB/FX2のマニュアルはレジスタの説明などがまとまっておらず、必要な情報を探すのがやや面倒ですので、AcrobatReaderの検索機能を使って関連するキーワードが使われているところを探しながら読むとよいと思います。

●EZ-USB/FX2の概要

添付ボードに搭載されたUSBコントローラは、サイプレス社のUSB2.0（480Mbps）対応のEZ-USB/FX2（以下、FX2と略す）です。FX2はカウンタ/タイマ機能などを内蔵した8051コアに8Kバイトのプログラム・メモリなどを内蔵したワンチップ・マイコンと、USB2.0コントローラを組み込んだ多機能USBターゲット・コントローラです。

FX2の内部ブロックは図6のようになっています。CPUコア、USBコントローラ、8Kバイトのプログラム/データRAMに加えて、高速データ転送をサポートするGPIF（General Programmable Interface）と呼ばれる伝送制御ユニットがあります。

GPIFはプログラマブルな転送制御コントローラで、これを利用することで数十Mバイト/秒を越えるような高速伝送を実現可能です。

●FX2はUSBをダウンロード・ケーブルとして使える

一般的なワンチップ・マイコンでは電源投入後、まずCPUが起動してROM内にあるプログラムを実行することが一般的ですが、FX2の場合には少しようすが異なります。FX2の場合、電源投入後にまずUSBコントローラが起動し、CPUはリセットされたままになっているのです。

これではまったく動かないではないかと思われるかもしれませんが、実はFX2のUSBコントローラは非常にインテリジェントにできており、CPUを介さずにホストからのコマンドを処理できるのです。

電源投入後、USBコントローラはまずI²Cバス上に接続されたシリアルEEPROMの先頭バイトを読み出します。もし、プログラムが入っているというIDコードが読み出されれば、以降の内容をFX2の内蔵SRAMに転送して、CPUのリセットを解除します。

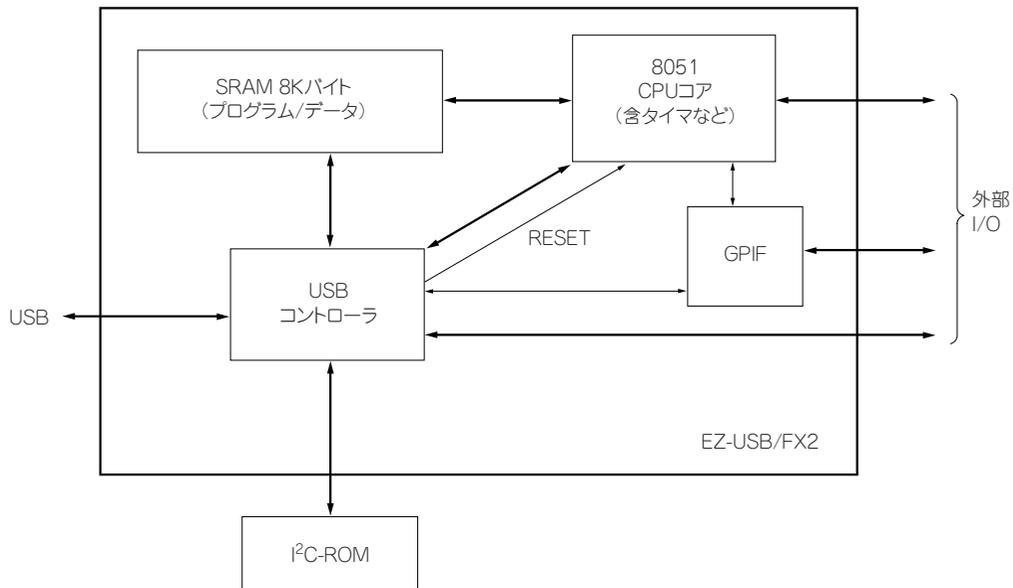


図6 USBコントローラEZ-USB/FX2の内部構成

シリアルEEPROMが接続されていない場合や、先頭バイトの情報が無効な場合には、USBコントローラはデフォルトのUSBターゲット機器として起動します。CPUがリセットされたままの状態ではUSBコントローラがホストとのやりとりを処理し、ホストからはUSBデバイスとして認識されます（ベンダID：04B4，プロダクトID：8613）。これを「デフォルトUSBデバイス」と呼んでいます。

もちろん、単にUSBデバイスとして認識されるだけでは面白くありません。FX2のUSBコントローラがユニークなのは、プログラムをFX2の内部SRAMに転送し、CPUのリセットを解除するベンダ・リクエスト・コマンドをもっているということです。

これを利用すると、まずデフォルトUSBデバイスとして起動してから、プログラムをUSB経由でFX2の内部SRAMに転送し、その後CPUのリセットを解除するという使いかたができるようになります。つまり、FX2はUSBをプログラムのダウンロード・ケーブルとして使うことができるUSBコントローラなのです。

本書でも、この機能を利用して、まずデフォルトUSBデバイスとして起動してから、ファームウェア EzFirm/FX2をダウンロードして動かすという方法をとっています（シリアルEEPROMに書き込む方法については「プロローグ」を参照）。

●起動の仕組み

FX2についてわかったところで、最初に戻って、ダウンロードなどによって何が起きているのかおさらいしておくことにしましょう。

ポイントは、

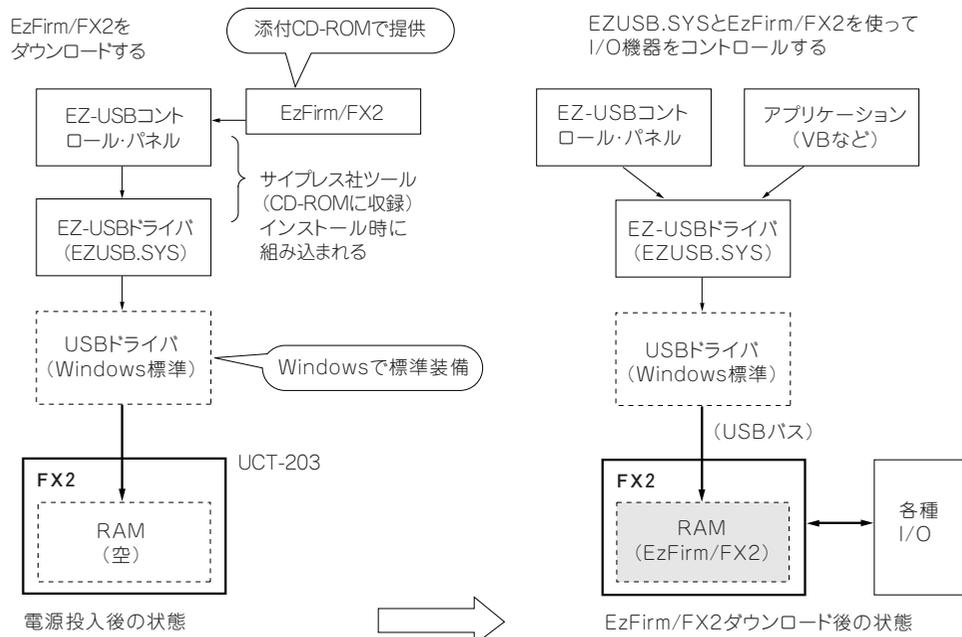


図7 ファームウェアのダウンロード前後の状態の変化

- ① 電源投入後はデフォルトUSBデバイスとなり、ファームウェアのダウンロードを待つ
- ② ファームウェアのダウンロード後はI/Oボードとして動作する
- ③ ベンダID、プロダクトIDはダウンロード前後で変わらない
- ④ サイプレス社のドライバは汎用ドライバである

の4点です。ボードのベンダIDやプロダクトIDは同じままで、ファームウェア・ダウンロードによって動作モードが切り替わるような状態をイメージされてもよいでしょう。

サイプレス社のドライバ (EZUSB.SYS) はファームウェア・ダウンロード専用のドライバではなく、任意のベンダ・リクエストを発行したり、エンド・ポイントのリード/ライトなどを行うことができるようになっている汎用ドライバですので、EZ-USBコントロール・パネルから利用するだけでなく、VisualBasicなどからUSBインターフェース・ボードをコントロールすることができるため、これを利用して各種制御を行うことにしたのです。

(1) EzFirm/FX2ダウンロード前の状態

図7は、電源投入直後 (USBケーブル接続直後) の状態と、ファームウェア (EzFirm/FX2) をダウンロードしたあとの状態を示した図です。

左側が最初にUSBケーブルを繋いだ直後の状態です。FX2がデフォルトUSBデバイスとしてベンダID = 04B4h, プロダクトID = 8613hのデバイスとして起動します。FX2内部のCPUコア (8051相当) はまだ動いて

いません。先ほど触れたとおり、USBコントローラ部分だけでUSBデバイスとして動作できるFX2の特徴です。

パソコン側では、Windows標準のUSBバス管理を行っているドライバの上にEZUSB.SYSが乗った状態になっています。EZUSB.SYSは汎用的なドライバで、任意のエンド・ポイントへのリード/ライトや、ベンダ・リクエストの発行などが自由に行えるように設計されています。EZ-USBコントロール・パネルもこのドライバを利用して、各種の操作を行うようになっています。

(2) EzFirm/FX2ダウンロード後の状態

FX2内蔵のUSBコントローラはファームウェア・ダウンロード用のベンダ・リクエストを受け付けることができますので、この機能を使ってEZ-USBコントロール・パネルから内蔵SRAMにプログラムをダウンロードします。ダウンロードが終了するとCPUのリセットが解除され、ファームウェアが起動します。これが図の右側の状態です。

EzFirm/FX2が起動すると、デバイスをいったんディスコネクト状態（USBから切り離された状態）にしてから、まったく同じデバイスID/プロダクトIDで接続状態に復帰します。ホストからはあたかもUSBコネクタが抜かれて、新たに接続しなおしたように見えるため、ホスト側は改めてデバイスの認識を実行しなおします。IDが同じままでエンド・ポイントの構成情報などが変化することになりますが、問題はありません。

ベンダIDやプロダクトIDが同じですので、ドライバはEZUSB.SYSがそのまま利用されます。EZUSB.SYSは汎用ドライバなので、EzFirm/FX2とのやりとりも自由に行えます。これを利用して、VisualBasicなどで作ったアプリケーションもEZUSB.SYS経由でUSBボードを制御するというわけです。

1-4 EzFirm/FX2によるエンド・ポイント構成

FX2はコントロール・エンド・ポイント（EP0）、インタラプトIN用およびOUT用エンド・ポイントをそれぞれ一つ（EPIINとEPIOUT）と、バルクIN/OUT用エンド・ポイントをもっています。

EzFirm/FX2を起動したときのエンド・ポイント構成は図8のようになります。インタラプトIN/OUT用のエンド・ポイントはFX2に内蔵されたシリアル・ポートを使ったデータ入出力を行うためのものですが、本キット添付ボード（UCT-203）に使った56ピン・パッケージ品にはシリアル・ポート端子がありませんので、この二つのエンド・ポイントは特に意味をもちません。

バルクIN/OUTエンド・ポイントは、次に説明する高速データ転送に対応したエンド・ポイントです。エンド・ポイントの構成やバッファ・サイズ、バンク数などはコンフィグレーション・レジスタの設定によって変えられるようになってはいますが、EzFirm/FX2ではバルクIN/OUTとも512バイト（USB2.0接続時、1.1では64バイト）×4バンク構成のものを一つずつもつ設定にしています。

4バンク構成になっているため、OUT方向ならば外部I/Oが低速であったりデータ伝送の準備ができていないなどの理由で転送が停止しても、最大2048バイト（512×4バンク）ぶんのデータは受け取ることができ、また、IN方向ならばホスト・パソコン側のデータ引き取りが多少遅くなっても、最大2048バイトぶんのデータをFX2内部に蓄えることができます。

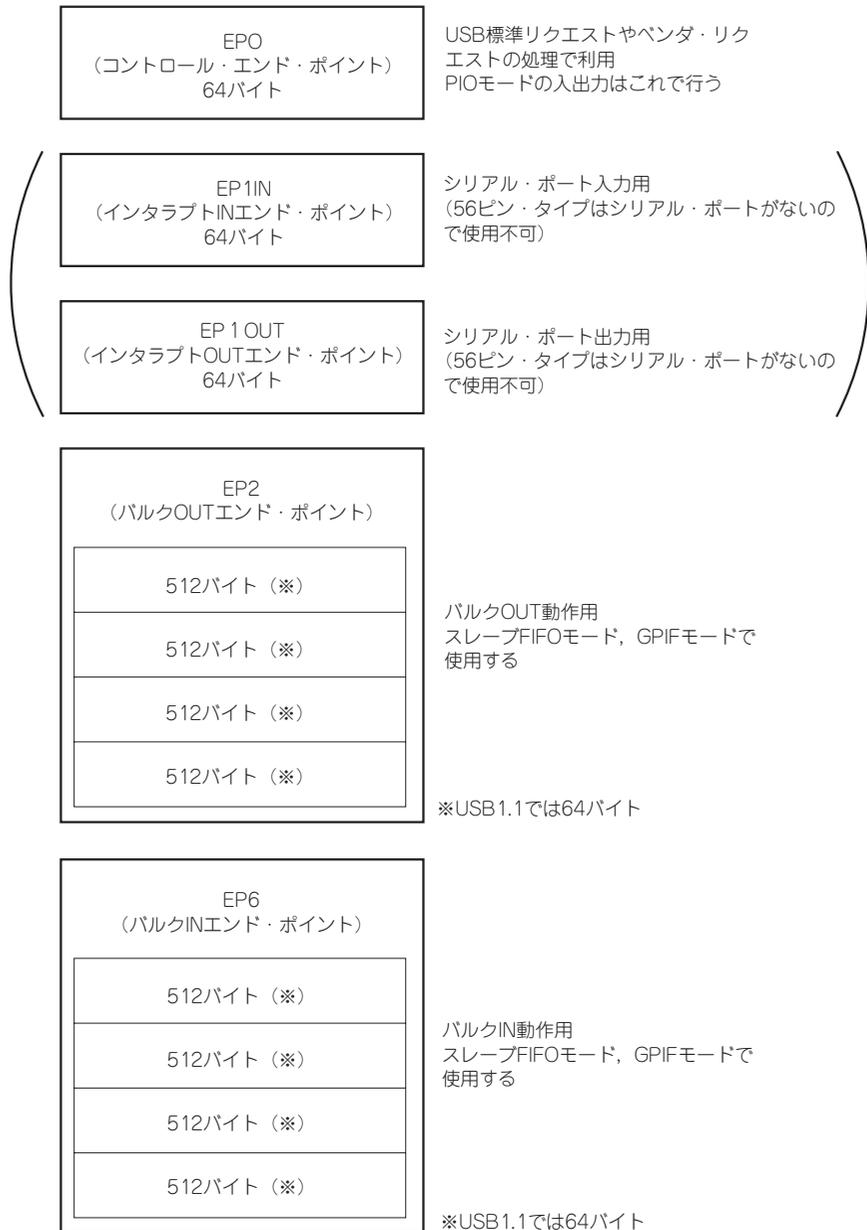


図8 ファームウェア (EzFirm/FX2) 動作時のエンド・ポイント構成

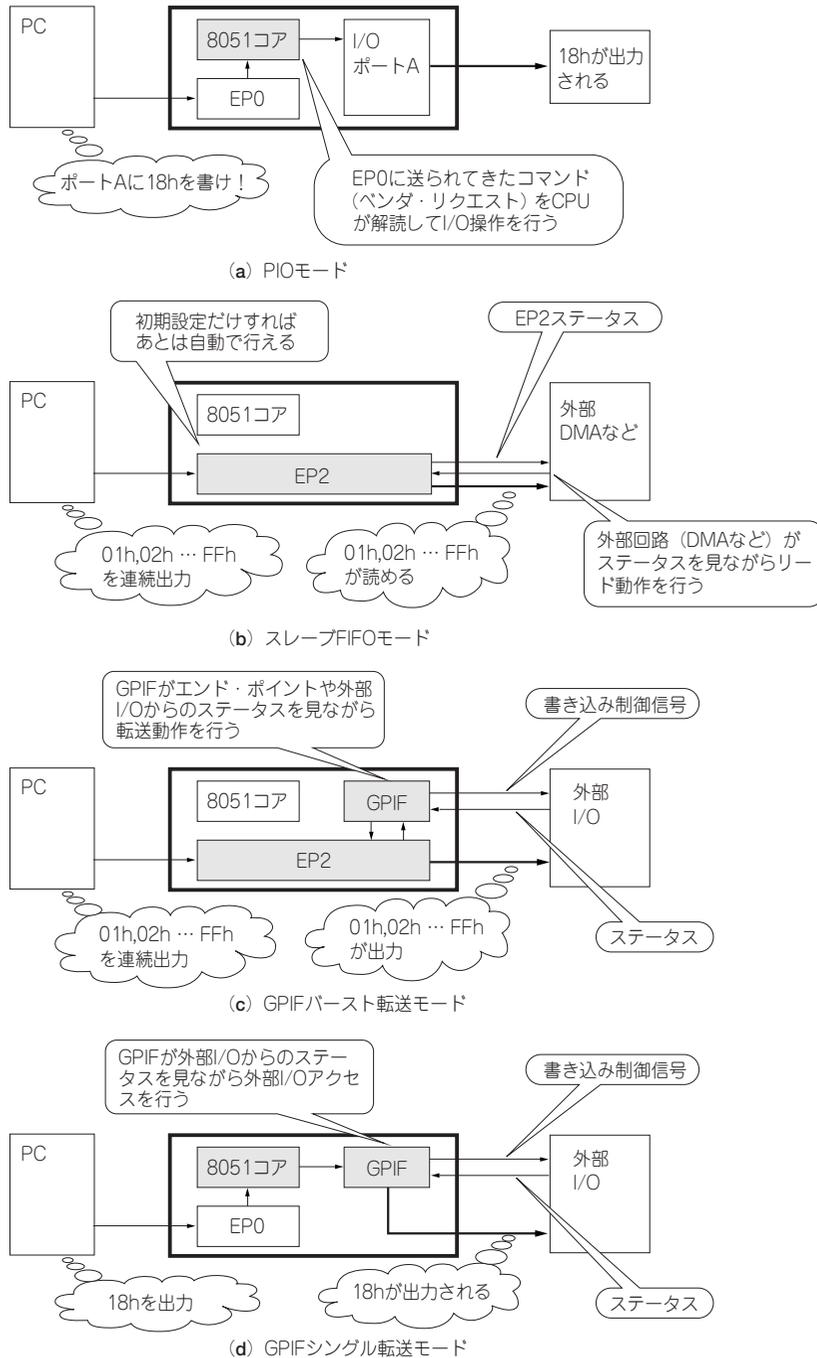


図9 4種類の動作モード

●データ転送モード

EzFirm/FX2を使ったデータ転送には次のようなものがあります (図9)。

- ① PIO モード
- ② スレーブFIFO モード
- ③ GPIF バースト転送モード
- ④ GPIF シングル転送モード

なお、GPIF バースト・モードと GPIF シングル・モードは共通する部分も多いため、単に GPIF モードとしてまとめて説明することもありますので注意してください。

① PIO モード

PIOモードは、ホスト（パソコン）が発行したベンダ・リクエスト・コマンドをFX2内部のCPUコア（8051）が解釈してPIOピンの状態の設定や読み出しを行うモードです [図9(a)]。8051コアが一つずつコマンドを解釈してPIOピンをアクセスするという方法になるためデータ転送速度はあまり稼げませんが、LEDのON/OFFやスイッチの状態を読み込むような用途には簡単で便利なモードです。

ただし、当然のことながら、PIOモードでは480MbpsというUSB2.0の高速な転送機能が生かされませんし、1バイト送るごとに外部とのピン制御を行うのは非常に面倒です。このため、FX2ではより簡単にUSB2.0の高速伝送に対応する仕組みとして、CPUを関与させずに自動的に外部とのデータ転送を行う動作モードをもっています。これが次に説明する「スレーブFIFOモード」と「GPIFバースト転送モード」です。

② スレーブFIFOモード

スレーブFIFOモードは、FX2がI/Oデバイスのようになり、外部につながれたCPUやDMAコントローラなどがFX2のエンド・ポイント・バッファのリード/ライトを行うモードです [図9(b)]。外部のデバイスからは、ちょうどFIFOメモリのようなイメージでフラグ（FullフラグやEmptyフラグ）をチェックしながら、リード/ライト信号を制御してデータ入出力することができます。

③ GPIFバースト転送モード

GPIFバースト転送モードは、FX2が外部I/O機器のマスタとなって能動的にデータ転送を行うモードです [図9(c)]。GPIFモードのときに、外部機器とのハンドシェイクなど、データ転送のための信号制御やタイミングのコントロールを行うのがGPIF（General Programmable InterFace）です。

④ GPIFシングル転送モード

GPIFシングル転送モードは、PIO転送モードのGPIF版と言えるようなモードです [図9(d)]。パソコンからのリード/ライト要求があると、CPUはPIOポートを直接アクセスするのではなく、GPIFにデータ入出力を行います。

GPIFを使った場合、I/O機器側からのステータスを判定しながら動作することができるため、少々複雑なバス

表1 USBインターフェース・ボード、ファームウェア、ドライバなどの仕様

種別	項目	仕様	備考
ボード	名称	UCT-203	ニッコー電子製
	USBコントローラ	CY7C68013 (EZ-USB/FX2)-56ピン	サイプレス製、USB2.0準拠(8051CPUコア内蔵)
	最大データ転送速度	480Mbps	バス上の最大速度
	エンド・ポイント数	コントロール・エンド・ポイント×1	
		バルクINエンド・ポイント×最大2	EzFirm/FX2では×1で4バンク構成で使用
	バルクOUTエンド・ポイント×最大2	EzFirm/FX2では×1で4バンク構成で使用	
電源	+5V単一	USBケーブルから供給(ボード内は3.3Vで動作)	
ファームウェア	名称	EzFirm/FX2	PastelMagic製
	I/O動作モード	PIOモード	単純なパラレルI/Oとして使用
		GPIFバースト転送モード	データ幅は16ビット固定
		GPIFシングル転送モード	
		スレーブFIFOモード	データ幅は16ビット固定
ベンダID/プロダクトID	04B4h/8613h	FX2のデフォルトと同一	
ドライバ	名称	EZUSB.SYS	サイプレス製(Windows98/2000/XPで使用可)
	発生可能な要求	ベンダ・リクエスト	IOCTLで行う
		バルク/インタラプト伝送	
		アイソクロナス伝送	
		USB標準リクエスト	
その他			
デバッグ用ツール	EZ-USBコントロール・パネル		サイプレス製(Windows98/2000/XPで使用可)

制御も簡単に行うことができます。つまり、プログラマブルなCPUバスのように使うことができます。

単一アクセスでよいならばGPIFシングル転送モードを使い、大量のデータを一気に送るときにはGPIFバースト転送モードを使うという具合に切り替えて使うのが便利です。たとえば、サイプレス社のリファレンス・デザインの一つである「USB-ATAPI変換アダプタ」の場合、コマンド・レジスタやステータス・レジスタなどのアクセスにはGPIFシングル転送モードを使い、セクタ単位のデータ・リード/ライトにはGPIFバースト転送モードを使っています。

スレーブFIFOモードやGPIFモードは一見難しいように思えるかもしれませんが、いったん設定を済ませてしまえば、ホスト側からは単にEP2(バルクOUTエンド・ポイント)へのライト、あるいはEP6(バルクINエンド・ポイント)のリードや1ワード・アクセスのベンダ・リクエストを発行するだけで済むため、アクセス自体はPIOモードよりも効率が良く、しかも簡単です。FX2の性能を生かすうえでも、ぜひ活用していただきたい機能です。

具体的な使いかたについては、FX2のドキュメントやサンプル・プログラムなどを参考にしてください。

●仕様

本セットに添付のUSB汎用インターフェース・ボード(UCT-203)やファームウェアなどの仕様の概略を、表1に示しておきます。