

[イントロダクション]

付属基板を取り出してUSBに挿し込むだけ！

USBマイコン基板で パソコンのI/O制御にチャレンジ！

● 付属基板をパソコンのUSBポートに挿すだけで始められる

本書には、78K0 USBマイコン基板が付属しています。この付属基板の特徴は、なんとと言っても「はんだ付け不要ですぐに使える」(図1)ことです。

袋から取り出した付属基板をパソコンのUSBポートに挿して、ハイパーターミナルなどの通信ソフトウェアを起動すれば、すぐにパソコンを使った装置制御の世界を実体験できます(図2)。

このマイコンには「トラ技BIOS」というソフトウェアが書き込まれています。「トラ技BIOS」は、USB経由のデータのやりとりをサポートしているので、ユーザが作ったプログラムをダウンロードして実行したり、フラッシュ・メモリへの書き込みなどを行ったりすることができます(図3)。すなわち、ダウンロー

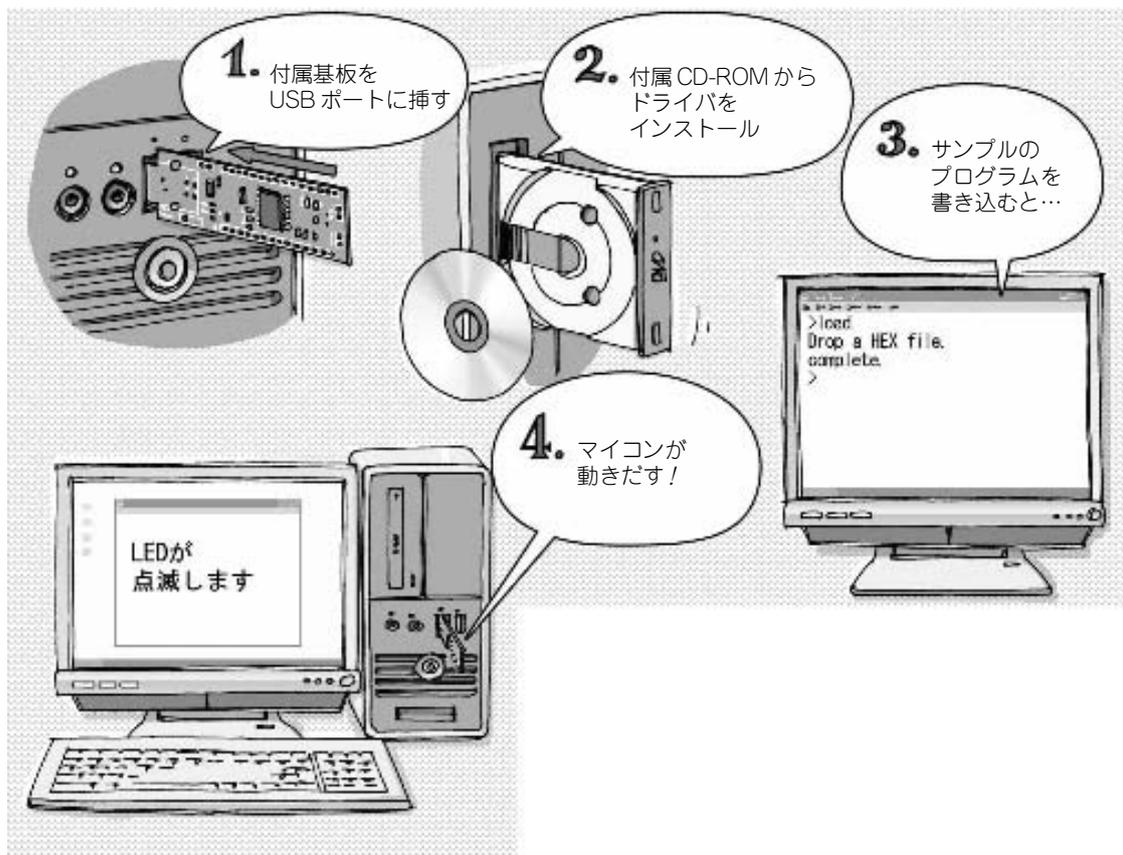


図1 本書にはUSBに挿せばいきなり動くマイコン基板が付いている

付属 CD-ROM のコンテンツ

● CD-ROM の内容と対応 OS

図 A の画面は、CD-ROM にある index.htm です。CD-ROM の内容がすべて参照できます。直接 CD-ROM 中のフォルダやファイルを参照したい場合は、「マイ コンピュータ」からドライブを右クリックし、開いたメニューでエクスプローラを選んでください。

付属 CD-ROM には、本書に関連する以下のファイルが含まれています。

- 作成したソース・ファイル
- マイコンに書き込み済みの USB 通信用ソフトウェア トラ技 BIOS
- USB デバイス・ドライバ
- アセンブラ RA78K0 W4.01 と統合開発環境 PM + V6.30
- C コンパイラ CC78K0 W4.00
- デバイス・ファイル DF780731
- マニュアルなど (μ PD78F0730 ユーザーズ・マニュアルほか)

本書の対応 OS は、Windows XP (64 ビット版には対応していない)、Windows 2000 です。

▶ 付属基板の USB デバイス・ドライバ

付属 CD-ROM の driver フォルダに収録されています。インストールの手順は、第 2 章を参照ください。CD-ROM をドライブに挿入すると、Windows の標準設定では図 A のような画面が自動で開きますが、ドライバのインストール作業には支障ありません。

▶ サンプル・プログラムの参照方法

CD-ROM ドライブの projects フォルダに収録されている自己解凍ファイルを実行すると、特に指定しない限り C:\¥CQ¥ TR0808 にフォルダを作ります。

フォルダは自動では開きません。「マイ コンピュータ」などから該当フォルダを開いて、hex ファイルやソース・ファイルなどを参照してください。

▶ 開発ツールのインストールに ID が必要

図 A の①のリンク先、または tools フォルダにある「インストール前にお読みください.txt」の中に記述されています。

[第1章]

USBに挿すだけで動き出す

付属USBマイコン基板の魅力と できること

本書付属の78K0 USBマイコン基板は、動かすだけなら、はんだ付けも追加部品も不要です。パッケージから取り出してパソコンのUSBポートに挿し込むだけです。

1-1 78K0 USBマイコン基板はここがすごい

付属基板は、NECエレクトロニクスのUSB内蔵ワンチップ・マイコン μ PD78F0730の周囲に、最低限の周辺回路を搭載したコンパクトなものです。

付属基板の端はUSBポートに合わせたエッジ形状に加工されているので、そのままパソコンに挿し込めます。拡張に必要な端子はすべて2.54mmピッチのコネクタに出ているので、写真1-1のように、万能基板に載せてDIP28ピンICのように使うこともできます。図1-1に示すように、連結ピンの太い方を付属基板に挿入します。

● USB通信機能を持つ

Microsoft Windowsがインストールされたパソコンにつなぎ、パソコン側にドライバをインストールするだけで、実験を開始できます。

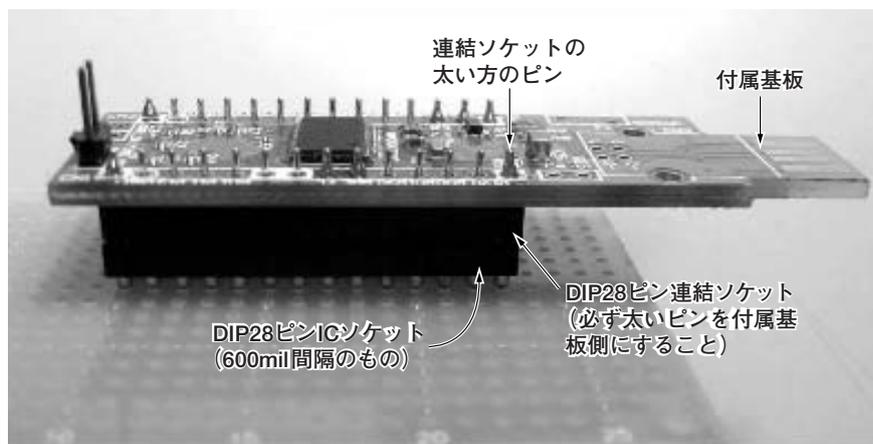


写真1-1 万能基板に実装したところ
DIP28ピンの連結ソケットとICソケットを使う

[第2章]

基板上のLEDを点滅させたりプログラムを書き込んだりする

付属78K0 USBマイコン基板を動かしてみよう

本章では、78K0 USBマイコン基板(付属基板)をUSBデバイスとしてパソコンにインストールし、マイコンにプログラムを書き込んで動作させる方法について説明します。

2-1 付属基板を動かすまで

付属基板をパッケージから取り出したら、さっそく動かしてみましょ。動かすまでのステップは、おおよ次のようになります。

- ① 付属基板をパッケージから取り出してパソコンのUSBポートに挿入
- ② 付属CD-ROMも取り出してパソコンに挿入
- ③ デバイス・ドライバをインストール
- ④ トラ技BIOSを動かしてみる
- ⑤ サンプル・プログラムをマイコンにダウンロードして動かしてみる

本章では、これらのステップについて詳しく解説していきます。

● Windowsパソコンを用意する

USBポートとCD-ROMドライブが付いていてマイクロソフトのWindows 2000またはWindows XPが

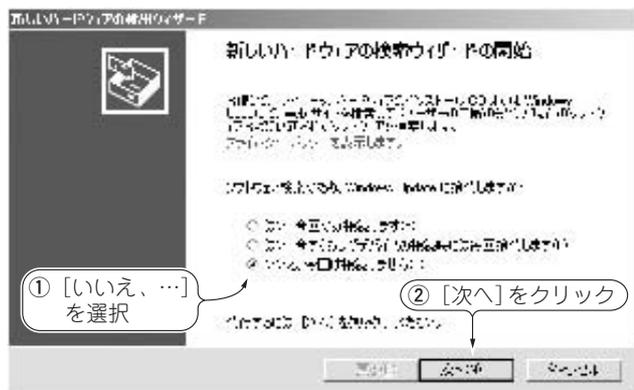


図2-1 デバイス・ドライバのインストール画面①
付属基板をUSBポートに挿し込むと表示される

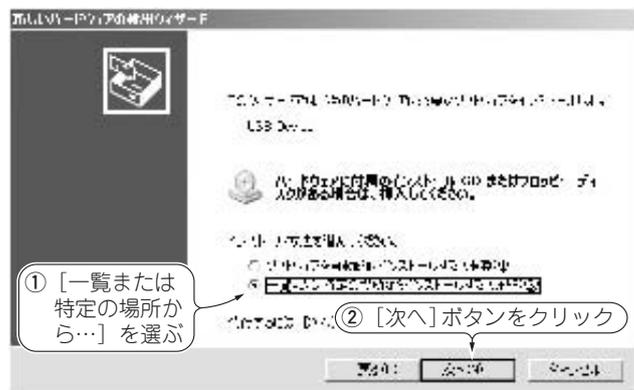


図2-2 デバイス・ドライバのインストール画面②
必要なソフトウェア(デバイス・ドライバ)は付属CD-ROMから読み込む必要があるので、[一覧または特定の場所...]を選ぶ

Tera Term を使った通信だけでは物足りない…

オリジナルの Windows ソフトを作る方法

オリジナルの組み込み機器では、専用のコントロール・ソフトウェアを使いたい場合もあります。このようなソフトウェアを開発する方法を、簡単に説明します。

● Windows API を使った COM ポートのプログラミング

自作プログラムで本書の付属基板を扱うには、仮想 COM ポートと通信するプログラムを書く必要があります。こういったデバイス・ドライバの操作は、Windows の場合、ファイル入出力のように扱われます。

そこで、簡単なプログラムをリスト 2A-1 に示します。これは、超簡易版のターミナル・ソフトウェアです。自作プログラムを作成する際のひな型としてご利用ください。

プログラムの大まかな流れは、以下の順番になっています。

- ① CreateFile 関数で COM ポートを開き、オブジェクト・ハンドルを得る
- ② DCB 構造体をセットする
- ③ WriteFile 関数、ReadFile 関数で通信をする
- ④ CloseHandle 関数でオブジェクト・ハンドルを閉じる

なお、使用したコンパイラは、

- Borland C++ Version 5.6
- Microsoft Visual Studio 2005 に付属のコンパイラ

です。

● Windows API を使って COM ポートを開く

COM ポートをオープンするには、CreateFile 関数を呼び出すときに、ファイル名の代わりにドライバ名を指定します。

ドライバ名とは「¥¥.¥」の後ろに COM ポート名を付けた特殊なファイル名です。COM ポートの 7 番であれば「¥¥.¥COM7」となります。

COM ポートのオープンが成功すると、CreateFile 関数はデバイス・ドライバへのオブジェクト・ハンドルを返します。

失敗した場合は、INVALID_HANDLE_VALUE で定義された値を返しますが、INVALID_HANDLE_VALUE はゼロではないので、エラー処理に注意してください。

[第3章]

CPUアーキテクチャ/ペリフェラルから外付けジャンパの使い方まで

付属基板と78K0マイコンの機能

3-1 搭載マイコンμPD78F0730のアーキテクチャ

μPD78F0730 (以下, 78F0730)は, 78K0コアを内蔵したワンチップ・マイコンで, 図3-1に示すアーキテクチャを持っています. そしてこのマイコンは, 表3-1に示す特徴を備えています.

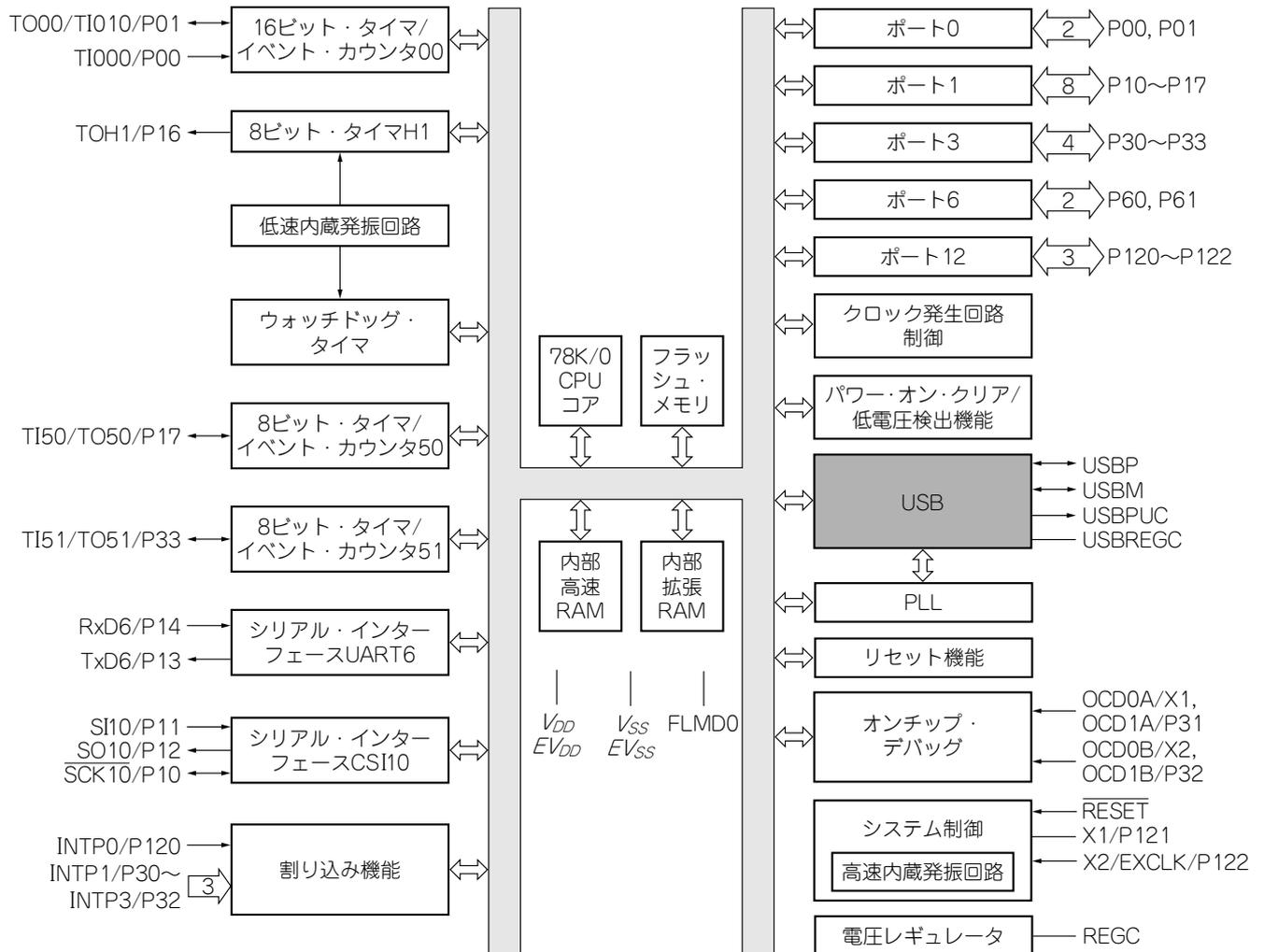


図3-1⁽¹⁾ 付属基板に搭載されているワンチップ・マイコンμPD78F0730の内部ブロック図
USBとそれに必要なPLL回路などを除けば一般的なマイコンの構成

USB 2.0ハイ・スピード・モードにも使える

高速通信システム向けに精度が向上したセラミック発振子

● 圧電性を利用したセラミック発振子「セラロック」

付属基板で使われている発振子「CSTCE16M0V13L**-R0」は，“セラロック”というセラミック発振子です。“セラロック”は，村田製作所の登録商標です。圧電性を持つセラミックスの機械的共振を利用した発振子です。

圧電性とは，機械的に歪ませると電圧が発生(圧電効果；Piezoelectric Effect)し，逆に電圧を加えると伸びたり縮んだりするという性質をいいます(図3A-1)。これらの効果は電気エネルギーと機械エネルギーを相互に変換するもので，特定の周波数で強く相互作用します。このことを利用して，特定の周波数で共振する発振子を作ることができます。

圧電性物質としては，古くからロッシェル塩や水晶など単結晶が知られています。セラロックは，大き

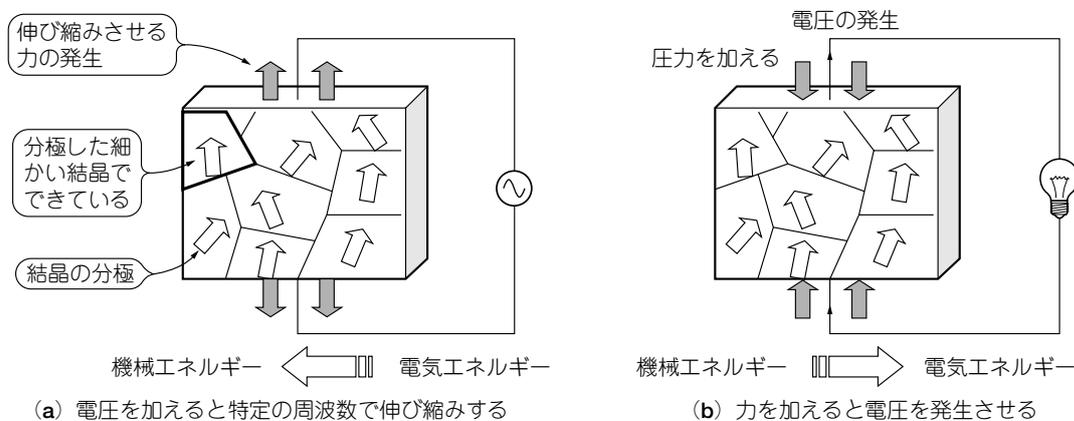


図3A-1 セラミック発振子の素材「セラミックス」の性質

表3A-1 各種発振子の特徴

| 名称 | シンボル | 価格 | 形状 | 調整 | 周波数初期精度 | 長期安定性 |
|----------|------|----|-----|----|---------|---------|
| LC | | 安価 | 大きい | 要 | ±2.0% | あまり良くない |
| CR | | 安価 | 小さい | 要 | ±2.0% | あまり良くない |
| 水晶振動子 | | 高価 | 大きい | 不要 | ±0.001% | 優れている |
| セラミック発振子 | | 安価 | 小さい | 不要 | ±0.5% | 優れている |

[第4章]

78K0 マイコンのプログラミングを始める前に…

ソフトウェア開発ツールのセットアップ

本章では、78K0 マイコンのプログラムを作るためのツールのセットアップ方法について解説します。

4-1 78K0 マイコンのプログラム開発の流れ

78K0 マイコンのプログラムはC言語やアセンブリ言語で記述します。C言語で記述したソース・プログラムは人が読んで理解できますが、78K0 マイコンはそれを理解することはできません。一方、78K0 マイコンが理解できるバイナリのコードはメモリに書き込まれた16進数の羅列であって、人間はそれを理解することができません。そこで、人間が理解できるC言語のソース・プログラムをマイコンが理解でき

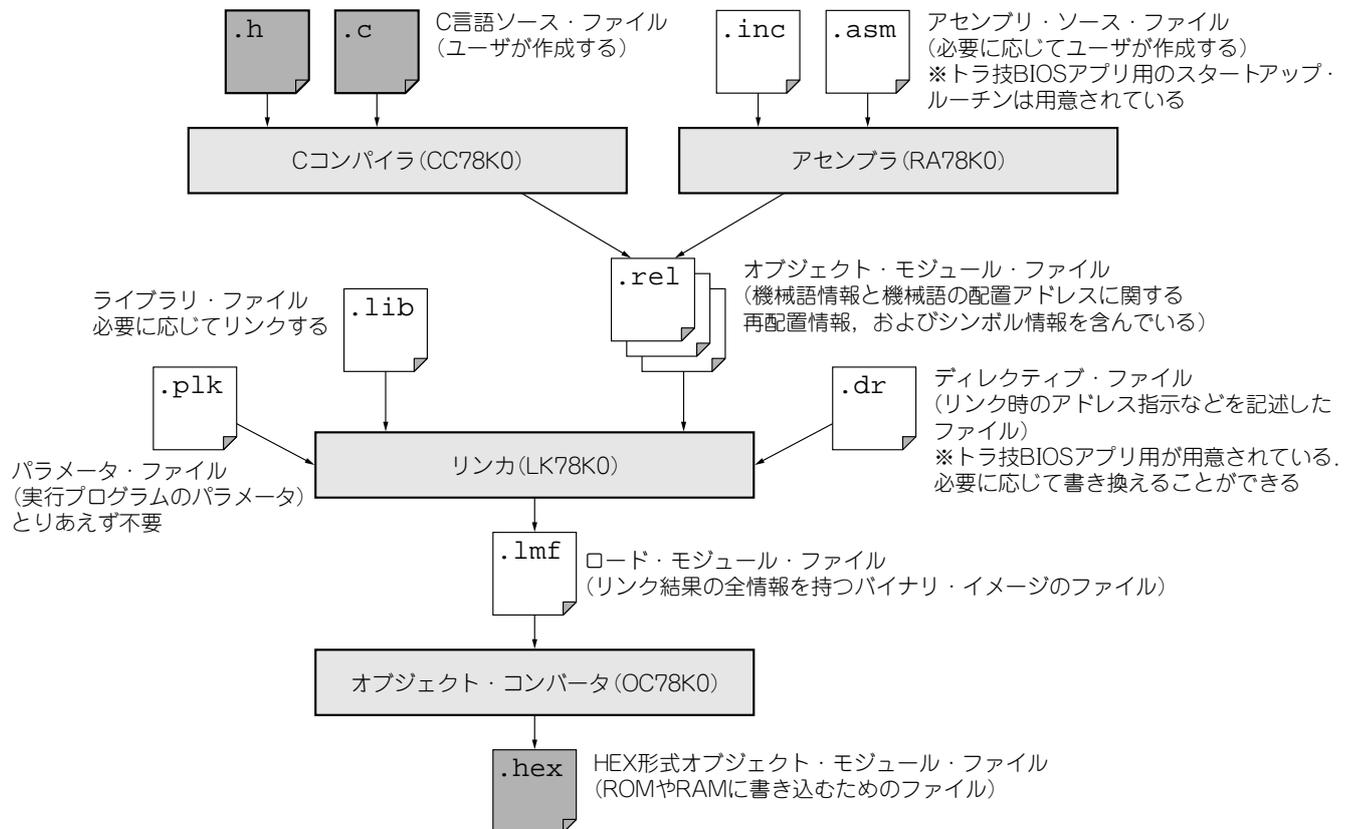


図4-1 78K0 マイコンの開発ツールとプログラム作成の流れ
ユーザが作るのは基本的にはソース・ファイルだけ

[第5章]

いよいよ付属基板を動かす

USB 通信プログラムを作成する

本章では、付属基板で動作するプログラムの作り方について以下の順に説明します。

- ① 統合開発環境PM+でプロジェクトを作る
- ② LEDを点滅させたりUSBの通信を行ったりスイッチの値を読み込んだりといった簡単なプログラムを作る
- ③ プログラムをフラッシュ・メモリに書き込んで、パソコンなしで(スタンド・アローンで)動作させる

5-1 開発ツールを用いたプログラムの作成

一般的に開発ツールでは、どんなマイコンを対象にしてどんなコンパイラを使うのか…といった設定をまず行う必要があります。この設定は、マイコンの開発ツールの場合にはプロジェクトなどと呼ばれます。

● ワークスペースの作成

78K0マイコン用の統合開発環境PM+ V6.30では、プロジェクトのことをワークスペースと呼びます。

Windowsのスタート・メニューから[すべてのプログラム]-[NEC Electronics Tools]-[PM+ V6.30]と選んでPM+を起動します。

▶ ワークスペースの名前と保存フォルダを設定

PM+のメイン・メニューから[ファイル]メニューにある[ワークスペースの新規作成]を実行します。

図5-1のダイアログが開くので、ワークスペース・ファイル名に、開発するプロジェクトの名前を入れ

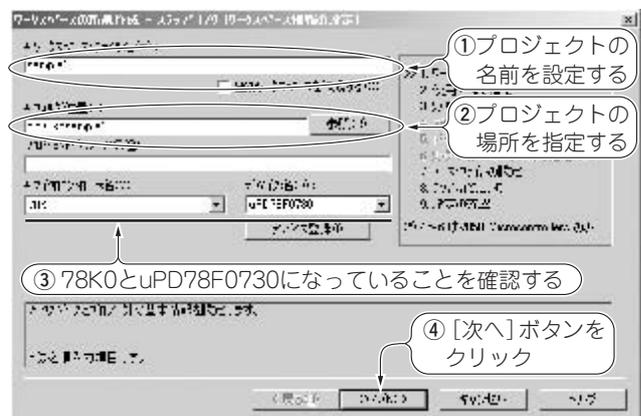


図5-1

プログラム作成の準備①

PM+の[ファイル]-[ワークスペースの新規作成]で開く、プロジェクト名とフォルダを指定する

Cコンパイラの標準ライブラリを使う方法

78K0マイコンの開発ツールは、本来、ユーザ・アプリケーションとスタートアップ・コードと一緒にリンクして、ROM用のHEXファイルを作成するものです。トラ技BIOS上で動かすような使い方を想定していません。そのため、本章の記事ではライブラリは使わないようにして、確実にプログラムをビルドする方法を紹介しました。

ところが本文で紹介した方法で、

```
for(i=0;i<50000;i++)
```

のようなコードを書くと、リンク時にエラーが出るかと思えます。なぜならば、`i<50000`という文はライブラリを使うからです。ライブラリが使えないと、大きな整数の演算や乗除算ができません。

ここでは、アプリケーションで78K0開発環境に用意された標準ライブラリを使う方法を紹介します。ただし、ライブラリについてはまだまだ未解明な部分があり、確実に動作が保証できる方法ではないので、ご注意ください。

● 設定方法

① コンパイラ・オプションの設定

図5-17 (p.84)のコンパイラ・オプションの設定ダイアログで、標準のライブラリを使用するのチェックをONにしてください。必要に応じて浮動小数点对応の`sprintf`もチェックしてください。

② リンク・ディレクティブ・ファイルの編集

次に、リンク・ディレクティブ・ファイル(RAMAPP.drまたはROMAPP.dr)を編集し、コメントアウトされている下の3行(ROMの場合はIXRAMではなくUROMになる)、

```
#MERGE @@LCODE := IXRAM
#MERGE @@R_INIS := IXRAM
#MERGE @@R_INIT := IXRAM
```

これらの行の頭にある#を外し、

```
MERGE @@LCODE := IXRAM
MERGE @@R_INIS := IXRAM
MERGE @@R_INIT := IXRAM
```

としてください。これでライブラリを使用するプログラムがビルドできます。

付属基板特有のトラブルへの対応

プログラムがハングアップしたときの チェック・ポイント

本書のシステムは、ブート・コードとアプリケーションを分離してビルドするのが特徴です。しかし、これは78K0開発ツールが本来想定していない使い方です。

普通の78K0マイコンの感覚でプログラムを作ると、いろいろな問題が生じる可能性があります。作ったプログラムが原因不明のハングアップをするときには、以下のような点を疑ってみてください。

● 知らないうちにライブラリを使っていないか？

浮動小数点数や `sprintf` 関数を使う場合はもちろんのこと、78K0のCコンパイラCC78K0は予期せぬところでライブラリを使おうとします。例えば簡単な時間待ちを入れたい場合、`for`文で、

```
unsigned short i;
for(i=0;i<50000;i++) {}
```

とすると、ライブラリがリンクされてしまいます。一方、

```
for(i=0;i<30000;i++) {}
```

と書けば、ライブラリはリンクされません。

CC78K0は32767以下の数の比較なら `cmpw` 命令と `bnc` 命令で対応しますが、32767を超える比較は `@@lscmp` というライブラリ関数を必要とするようです。同様に乗算、除算を行う場合も要注意です。

ライブラリを使いたい場合は、本章のAppendix Aを参照してリンク・ディレクティブ・ファイルを編集し、`@@R_INIT`、`@@R_INIS`、`@@LCODE`の各セクションが有効になるようにしてください。

● グローバル変数の初期値設定に失敗していないか？

トラ技BIOSでは、グローバル変数の初期値は不定です。

```
int hensu=5; // このhensuは初期化されない
```

のように記述しても初期化されません。

グローバル変数を初期化したい場合は、`main`関数の先頭で、

```
hensu = 5;
```

のように明示的に記述するか、`const`を付けてください。

```
const int hensu=5; // このhensuは初期化される
void main() {
...
}
```

同様に、未指定のグローバル変数の初期値はゼロにはなっていないので注意してください。

[第6章]

PC-マイコン間の通信遅れ要因を把握して応用製作に活かす

78K0 マイコンのI/Oの応答性能

第5章では、パソコンからUSBを経由してLEDを点灯させたり、スイッチの入力を読み取ったりする方法について紹介しました。

ところが、この方法を応用して複雑なシステムを制御しようとする、動作速度が遅くて使いものにならないのではないかと思います。

そこで本章では、USBと78K0 USBマイコン基板を通じて、外部の機器を操作するときの問題となる応答速度を向上させるためのヒントを紹介します。

6-1 付属基板のUSB通信速度はどのくらい？

応答パフォーマンスを向上させるには、現状の付属基板とUSBを介したI/Oコントロールの応答性能を知る必要があります。

まず、通信速度を測ることにします。USBでは一般に、IN(ターゲット→ホスト)とOUT(ホスト→ターゲット)の速度が異なるので、両方測ることが必要です。

■ 通信の速度

● IN(ターゲット→ホスト)側の通信速度は46Kバイト/s

IN側の速度の測り方はとても簡単です。1Mバイト程度のデータを転送したときの通信時間を測り、
通信速度 = データ・サイズ ÷ 通信時間
で求めます。

256バイトを4096回送信するマイコン側のプログラムは次のようになります。

```
for(i=0; i<4096; i++) {  
    trg_senddata(tmp, 256);  
}
```

このルーチン呼び出す前に、tmpというバッファにはあらかじめ適当な文字列をセットしておきます。実際に測定してみると23秒かかったので、通信速度は46Kバイト/sということになります。この値を10倍すればRS-232-Cに換算したビット・レートになります。

RS-232-Cに換算すれば455 kbpsなので、まずまずの速度でしょう。速度は使用しているパソコンの処理速度にも若干影響されるので、あくまでも目安と考えてください。

[第6章 Appendix]

短時間で処理を済ませたいときに有効なメモリ領域

データの出し入れが速い高速 RAM 領域

● 自由に使えるのは FE20～FEFF の 152 バイト

処理スピードを速めたいときは、マイコン内部の高速 RAM にデータ (変数など) を格納するとよいでしょう。内部高速 RAM は、0XFB00～0XFEFF の 1024 バイトの範囲に存在しています (図 6A-1)。トラ技 BIOS や USB ファームウェア、ブートローダ、スタック領域も内部高速 RAM を使用しています。それらの領域を壊さないように、ユーザ・プログラムは空いている領域を使うようにしてください。

▶ 領域 1 (FB00～FCB5)

USB ファームウェアとトラ技 BIOS は、高速 RAM 領域のうち 0XFB00～0XFCB5 の 438 バイトを使用しています。これはグローバル変数として使用している領域です。トラ技 BIOS がバージョンアップされるとこの先頭アドレスも変わる可能性があります。

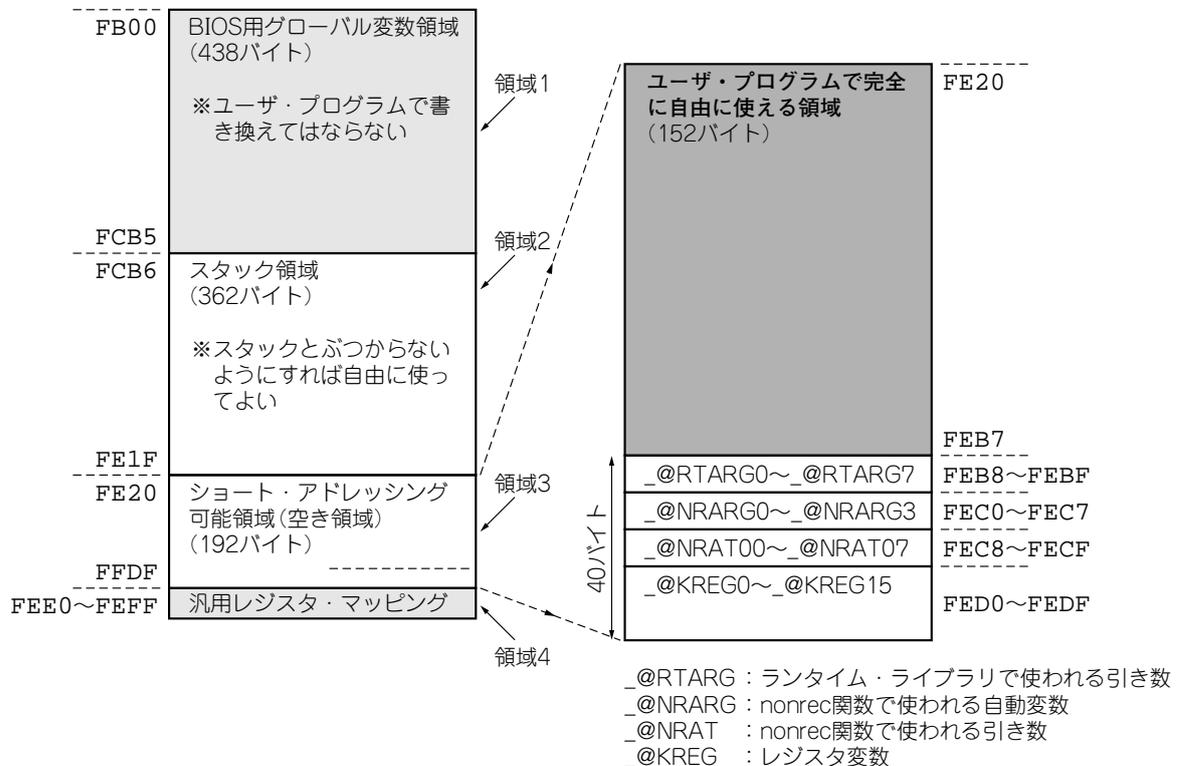


図 6A-1 付属基板上の 78K0 マイコン内部の高速 RAM の割り当て

[第7章]

マイコンに自分自身をプログラミングする機能を持たせる

ユーザ・プログラムでROMを書き換える方法

付属基板に搭載されているマイコン μ PD78F0730は16 Kバイトのフラッシュ・メモリ (ROM) を内蔵しています。付属基板では図7-1のメモリ・マップに示すように、前半の8 Kバイト (0000H~1FFFH) をブートローダとトラ技BIOSで使用し、後半の8 Kバイトをユーザ・プログラムやデータの格納用に開放しています。

78K0マイコンに内蔵されているRAMやフラッシュ・メモリ上のユーザ・プログラムで、フラッシュ・メモリの内容を書き換えることをセルフ・プログラミングといいます。本章では、セルフ・プログラミングを利用して、2000H番地以降の内蔵フラッシュ・メモリに、データを格納する方法を解説します。

7-1 ROMユーザ領域のプログラムでROMユーザ領域の内容を書き換える方法

セルフ・プログラミングを行うためには、NECエレクトロニクスが提供するライブラリを使用します。このライブラリはUSBブートローダ内のサービスAPIから呼び出されますが、そのサービスAPIは

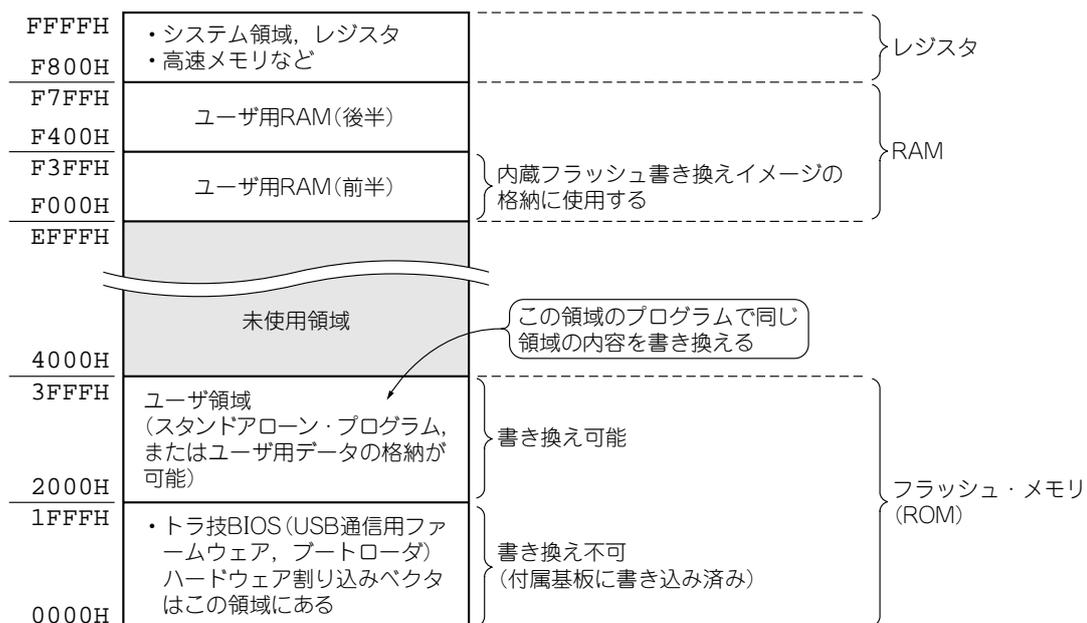


図7-1 付属基板に搭載された78K0マイコンのメモリの割り当て

[第8章]

処理に優先順位をつけてCPUを効率良く動かす

78K0 マイコンの割り込みプログラミング

8-1 割り込み機能を使うメリット

● 緊急度の高い処理を優先させることができる

割り込みは、マイコンを使ううえで必ずマスタしておきたい機能の一つです。マイコンのプログラムは命令順に実行されていきます。しかし割り込みを使うと、その流れを変えて緊急度の高い処理を優先的に行わせることができます。処理が済んだら元のルーチンに戻り、何事もなかったかのように元の処理を続行します。例えていうならば、読書しているときに電話がかかってきたときに、一時的に対応するイメージです(図8-1)。

● CPUの負荷が軽くなる

割り込みを使うとソフトウェアの負荷を軽減させることもできます。例えば、データがいつどんなタイミングで受信されるかは分からない通信を行うプログラムでは、割り込みを使わなければ、ループの中で常に受信フラグを監視していなければなりません。処理が重くて規定の時間内に受信フラグを見ることができないような状況では、受信データを取りこぼしてしまうことになるでしょう[図8-2(a)]。



図8-1 マイコンの割り込み機能のイメージ

[第9章]

任意の周期信号を生成したり、狙ったタイミングで割り込みをかけたりする

78K0 マイコンのタイマ機能の使い方

付属基板に搭載されたマイコン μ PD78F0730 の中には、第3章の図3-1に示すように、さまざまな内蔵ペリフェラルが入っています。ペリフェラルとは「周辺機器」のことで、CPUから独立して自律的に動作するハードウェアのことです。時間を計ったり、周波数を測るときに利用するタイマやカウンタは内蔵ペリフェラルの代表的な存在です。これらの単純で高速な応答が求められる処理は、専用のハードウェアに任せてしまった方がCPUの負担が減ります。正確な時間をカウントできるので、ほとんどの組み込みマイコンにハードウェアで内蔵されています。付属基板に搭載された μ PD78F0730 には、ウォッチドッグ・タイマを含めてタイマが5種類も入っていて(表9-1)、小規模マイコンながら充実しています。

本章では内蔵タイマの使い方を説明します。すべての機能を紹介すると膨大な量になるので、最も基本的な使い方である「インターバル・タイマ」と「PWM信号の発生」についてのみ説明します。より詳しい使い方や応用例は、 μ PD78F0730 ユーザーズ・マニュアル⁽¹⁾の第6章～第9章(付属CD-ROMに収録)を参照してください。

9-1 タイマを使ってできること

ワンチップ・マイコンでは、時間の計測と割り込みの発生、そしてPWM(Pulse Width Modulation)信

表9-1 78K0 マイコンが内蔵するタイマの機能と用途(ウォッチドッグ・タイマを除く4種類)

| 項目 | 16ビット・タイマ/ イベント・カウンタ00 | 8ビット・タイマ/ イベント・カウンタ50 | 8ビット・タイマ/ イベント・カウンタ51 | 8ビット・ タイマH1 | |
|-------------|---------------------------|--------------------------|--------------------------|----------------|----|
| | 00 | 50 | 51 | H1 | |
| 機能で分類 | 精度(ビット) | 16 | 8 | 8 | 8 |
| | コンペア・レジスタ | 2個 | 1個 | 1個 | 2個 |
| | 割り込み発生 | ○ | ○ | ○ | ○ |
| | 外部イベント・カウント | ○ | ○ | ○ | × |
| | 値の読み出し | ○ | × | × | × |
| 用途で分類 | キャプチャ機能 | ○ | × | × | × |
| | インターバル・タイマ | ○ | ○ | ○ | ○ |
| | 方形波出力 | ○ | ○ | ○ | ○ |
| | PWM出力 | ○ | ○ | ○ | ○ |
| | 周期可変PWM出力 | ○ | × | × | ○ |
| キャリア・ジェネレータ | × | × | × | ○ | |

↑
最も機能が充実している

[第10章]

マスター-スレーブの関係やUSBデバイスに書き込まれた情報

いまさら聞けないUSB規格のいろいろ

10-1 USBコネクタの種類

● 3種類のコネクタ形状

USB (Universal Serial Bus) のコネクタには、写真10-1や図10-1に示すように、

- シリーズA
- シリーズB
- シリーズ・ミニB

の3種類のプラグ/レセプタクルがあります。

● ホスト側とデバイス側で使い分けられている

シリーズAのプラグはホスト，またはハブの下位ポートで用いられます。シリーズBとシリーズ・ミニ



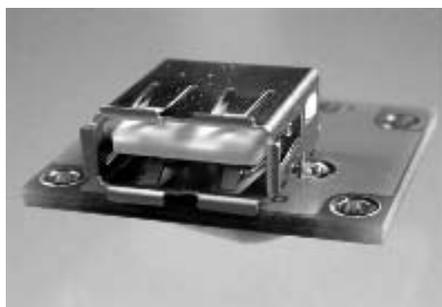
(a) シリーズAプラグ



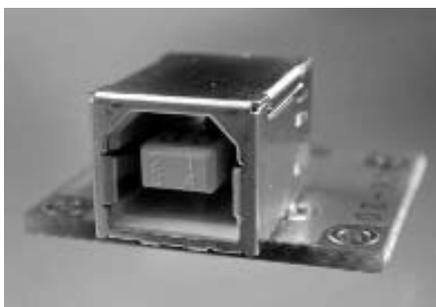
(b) シリーズBプラグ



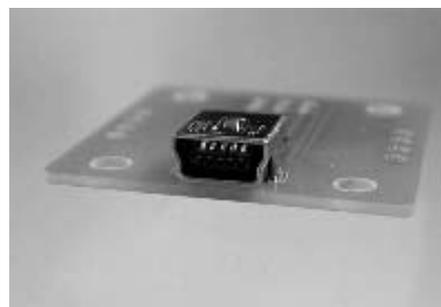
(c) シリーズ・ミニBプラグ



(d) シリーズAレセプタクル



(e) シリーズBレセプタクル



(f) シリーズ・ミニBレセプタクル

写真10-1 USBコネクタの外観

2007年にはマイクロB、マイクロABというシリーズも規格化された