mbed & LPC1114 入門

1. 目次

2. はじめに

この文書では、mbed 開発環境を使って NXP 社のマイコン、LPC1114 を使うための方法を説明します。想定している読者は、これからマイコンを触ってみたいと考えている人です。内容は mbed の紹介から始めて簡単な LED の点滅の説明まで行います。

クラウド開発環境である mbed や、それを使った LPC1114 向けのアプリケーションの開発方法などについては、いくつもの記事がネット上に存在します。しかしながら、多くは残念ながら断片的なものであり、初めて mbed と LPC1114 に触れる人のための一貫した文書にはなっていません。そこでこの文書では詳細の全てをカバーしない代わりに、第一歩から始めて、一応のアプリケーションの完成までを通して解説します。PC とプログラミング、および電子回路に関する基本的な知識さえあれば、ほとんどの人がこの文書を読むだけで LED 点滅アプリケーションを自分で書いて LPC1114 に書き込み、実行できるようになるはずです。

この文書では開発環境として Windows PC を使用しています。使用するツールやプログラムは Windows 8.1 で動作を確認していますが、Windows 7 や Windows 8 でも同様に動くはずです。また、使用するツールやプログラムは同様に MacOS や Linux でも動作しますが、多少使い方が変わってきます。

2014 年 4 月 10 日

酔漢

3. クラウド開発環境 mbed

この章ではソフトウェア開発環境 mbed について説明します。

3.1. mbed とは

mbed は、クラウドベースのソフトウェア開発環境で、ARM CORTEX-M シリーズ CPU を使用したマイコンのソフトウェアを開発することが出来ます。

クラウドベースの開発環境ですので、作成したソフトウェアは手元の PC には残りません。ソフトウェアやそのソフトウェアを実行させるターゲットの情報は全てクラウド、つまりは mbed のサーバーに格納されています。

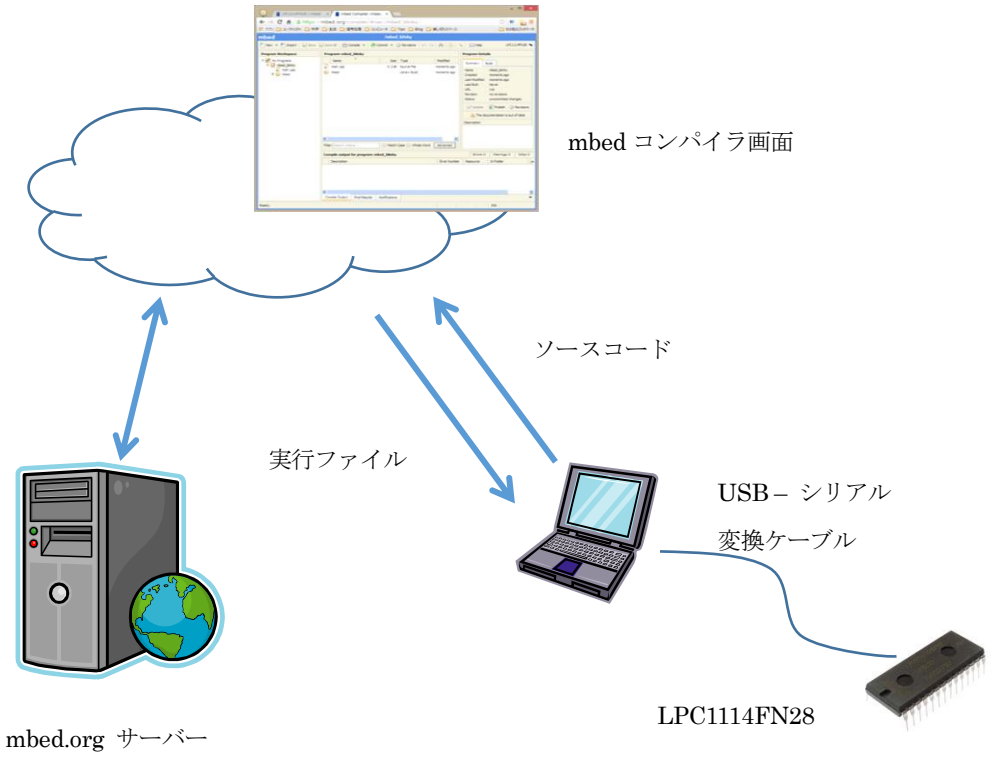

図 3-1 クラウドベースの開発環境

ソフトウェアが手元に残らないと聞くと、少し不安を感じるかも知れません。しかし、手元に置かない事にはたくさんメリットがあります。まず、バックアップの心配をしなくて済みます。サーバーの管理は mbed の管理組織が行っていますので、バックアップも定期的に取られています（多分）。次に、自分で開発環境のインストールをする必要もアップデートをする必要もありません。常に最新の開発環境を使うことが出来ます。また、ネットに接続できて WEB ブラウザを起動できるならどこでも開発が可能です。

開発環境としての mbed は、当初は NXP 社のマイコンを使った mbed ボード専用でした。しかし、その後対応ボードをゆっくりと増やし、今ではボードだけでは無く単体のマイコンにも対応しています。また、NXP 社以外のボードにも対応しています。

ブラウザを使って動作する点を除くと、mbed は普通の統合開発環境(IDE)によく似ています。つまり、エディタを使ってプログラムを開発し、その場でコンパイルすると実行ファイルを生成します。通常の IDE の場合、コンパイルすると実行ファイルがファイルシステムの上に作られますが、mbed はクラウド・アプリケーションであるため、コンパイルすると実行ファイルがダウンロード可能になります。

3.2. アカウントを作る

mbed はクラウド・アプリケーションですので、使用するにはアカウントを作らなければなりません。アカウント作成および維持は無料です。

アカウントを作成するには、まず https://mbed.org/ へブラウザで移動してください。そして login or signup をクリックします。

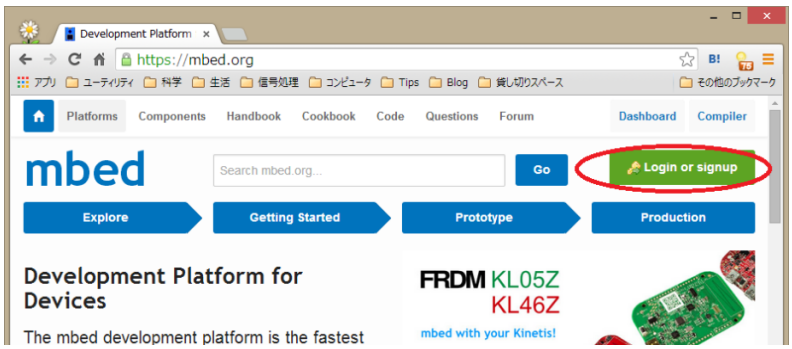

図 3-2 アカウントを作成する

アカウント作成のためには、メールアドレスの他、希望するアカウント名、パスワードなどの入力が求められます。アカウントの作成に成功すると再び https://mbed.org/ に戻りますが、今度はログオン状態ですので右上に自分が作成したアカウント名が表示されています。

3.3. プラットフォームを追加する

アカウントを設定したら、今度はプラットフォームを追加します。mbed でプラットフォームとは、ターゲット・ボードやターゲット・マイコンを指します。この文書ではプラットフォームとして LPC1114 の DIP 版である LPC1114FN28 を使います。LPC1114 については後で詳しく説明しますが、まずは mbed にこの IC を登録して開発が出来るように準備します。

プラットフォームを指定するには、画面上に並んでいるボタンから”Platform”をクリックします。

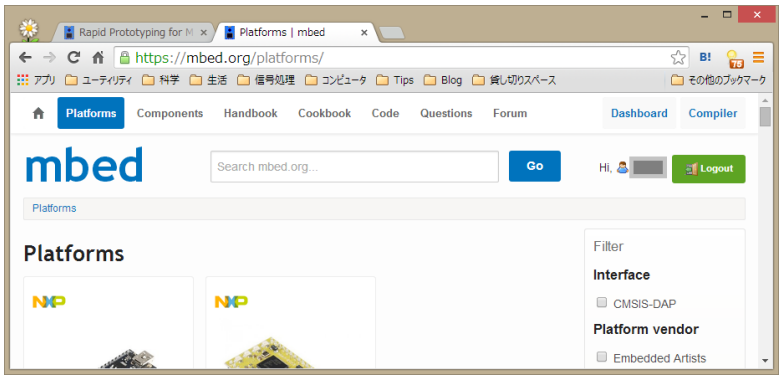

図 3-3 プラットフォームを選ぶ

すると、プラットフォーム選択ページに移りますので、スクロールしていきます。中ほどに LPC1114FN28 が写真入りで紹介されていますので、見つかったらクリックしてください。

図 3-4 LPC1114FN28 を選ぶ

すると、LPC1114FN28 の解説ページに移ります。このページには LPC1114FN28 が DIP マイコンであること、NXP 社製であることなどが紹介されています。

このマイコンで間違いなければ、画面右にある”Add to your mbed”をクリックしてください。これで自分のアカウントで LPC1114FN28 が利用できるようになりました。

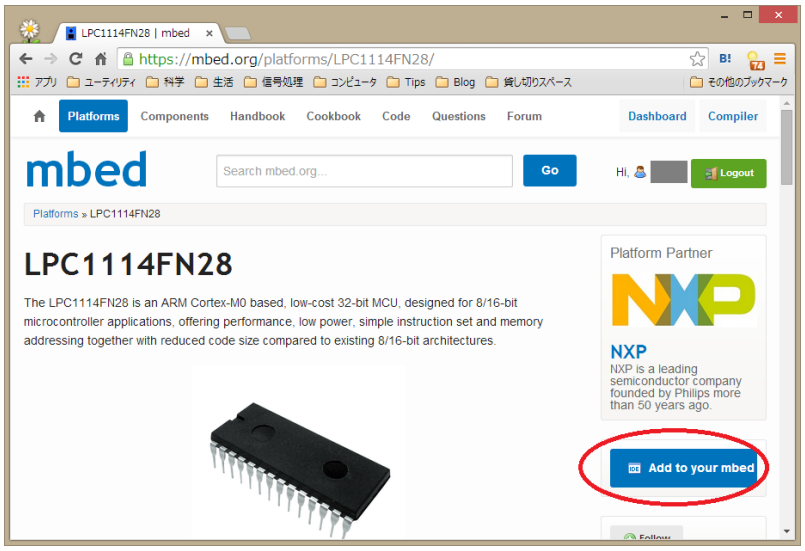

図 3-5　LPC1114FN28 を追加する

3.4.　テスト・プロジェクトを作る

LPC1114 をターゲットとして使う準備ができましたので、次にテスト・プロジェクトを作ってみます。

プロジェクトとは mbed での仕事の管理単位です。ひとつのプログラムがひとつのプロジェクトで管理されると考えておけばいいでしょう。

プロジェクトを作るには、mbed.org のホーム画面上で、右上の”compiler”をクリックします。すると、ウィ

ンドウが一つ開いて mbed の開発環境画面が表示されます。

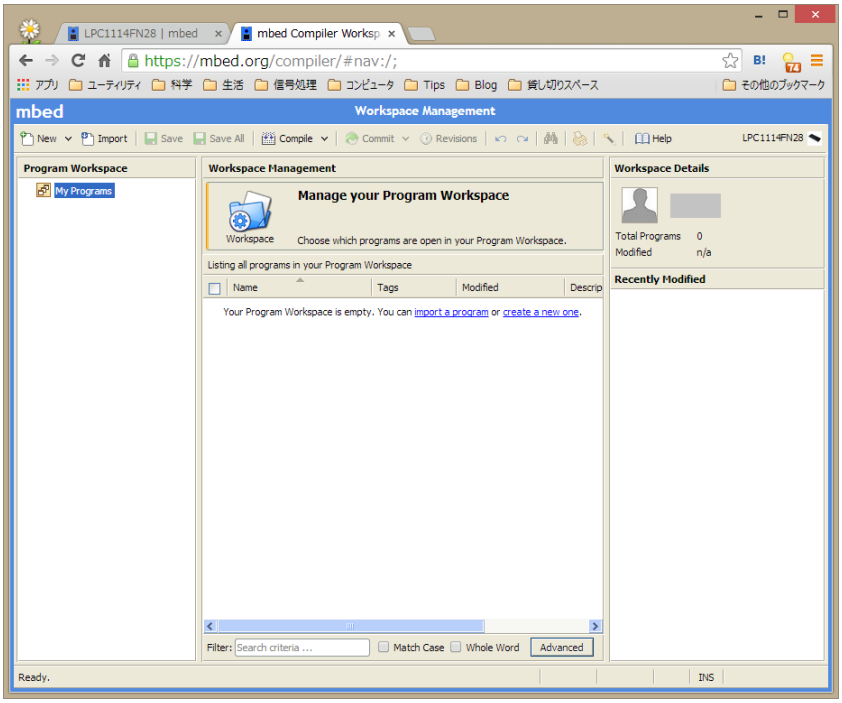

図 3-6 統合開発環境

この画面は mbed の統合作業環境画面です。この画面上でプロジェクトの管理、プログラム編集、ビルドといった一般的な作業が行われるほか、mbed から使えるヘルプの検索、表示、また、ほかの人が作って公開しているプログラムのインポートなどが行えます。

画面を注意して見ましょう。右上に LPC1114FN28 とあります。これは現在使えるターゲット・プラットホームが LPC1114FN28 であることを示しています。

次にプロジェクトを作ってみます。プロジェクトを作るには、画面上左上の”New”アイコンをクリックします。すると”Create New Program” ダイアログが現れます。

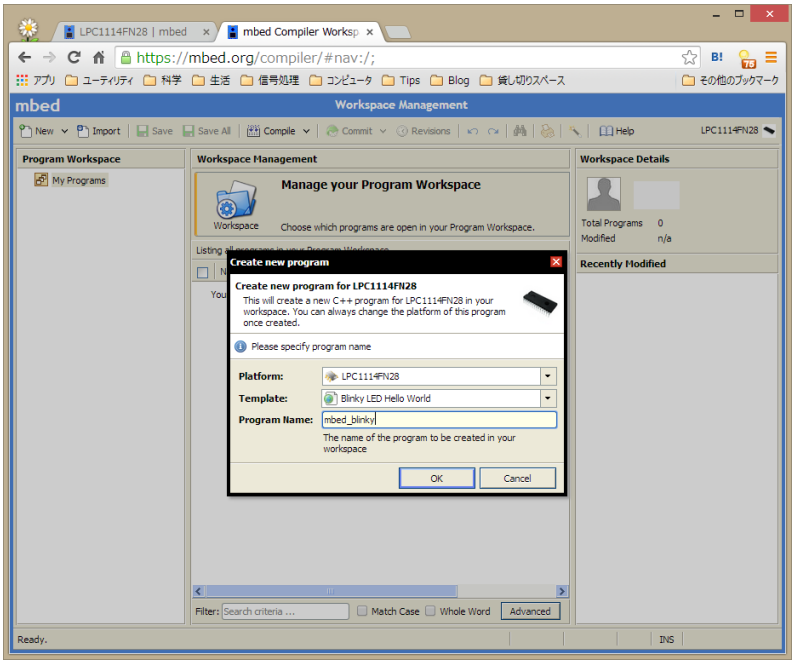

図 3-7 プロジェクトを作る

このダイアログではこれから作るプロジェクトがどんなものかを指定できます。Platform にはすでに設定した LPC1114FN28 が最初から指定されていますので、これをそのまま使いましょう。Template は、新しく作るプロジェクトの下敷きとなる雛形プロジェクトです。これは Blinky LED を選んでおきます。ご想像通り、これは LED 点滅プログラムです。プロジェクト名は適当に選んでおいてください。

OK ボタンを押すと、自動的にプロジェクトが生成されます。

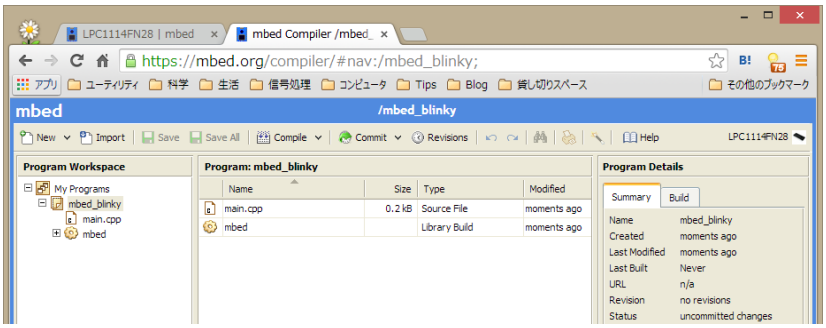

図 3-8 プロジェクトが生成された様子

これがプロジェクトの初期状態です。左の Project Worspace にプロジェクトが現れます。

初期状態ではプロジェクトには main.cpp というファイルがひとつあるだけです。Project Workspace で main.cpp をクリックすると画面右側にエディタが現れてファイルの編集が可能になります。mbed はクラウド環境ですので、ここで編集される main.cpp は手元の PC ではなく、クラウドの向こうのサーバーにあります。

Project Workspace にある mbed という名前の歯車アイコンはヘルプファイルを束ねており、mbed 環境で使える API はこちらから漁ることができます。ヘルプはペリフェラル機能毎に分類されており、便利です。

一通り眺めたら、Project Workspace のプロジェクト名をクリックして選択状態にし、画面上の”Compile”アイコンをクリックしてみましょう。サーバー上でコンパイルが行われ、成功すると実行ファイル(bin ファイル)のダウンロードが始まります。

ダウンロードが完了したら、とりあえずは機能チェック完了です。

4. LPC1114 とは

この章では NXP 社の LPC1114 について説明します。

4.1. 32bit マイコン LPC1114

LPC1114 は、NXP 社の 32bit マイコンであり、同社の幅広いファミリ展開のうち 32bit 品の低位ファミリである LPC1100 シリーズの一つです。LPC1100 シリーズの下位にはローエンドの LPC800 ファミリがあり、LPC1100 は下から 2 番目に当たります。

LPC1114 は 50MHz で動作する 32bit CPU の周辺にペリフェラル、4kB の SRAM、32kB の FLASH ROM、RC オシレーター、JTAG デバッグ回路を集積しており、バイパス・コンデンサを接続して 3.3V を給電するだけでコンピュータ・システムとして動作します（図 4-1）。この図は、LPC1114 のデータシートから引用しました。

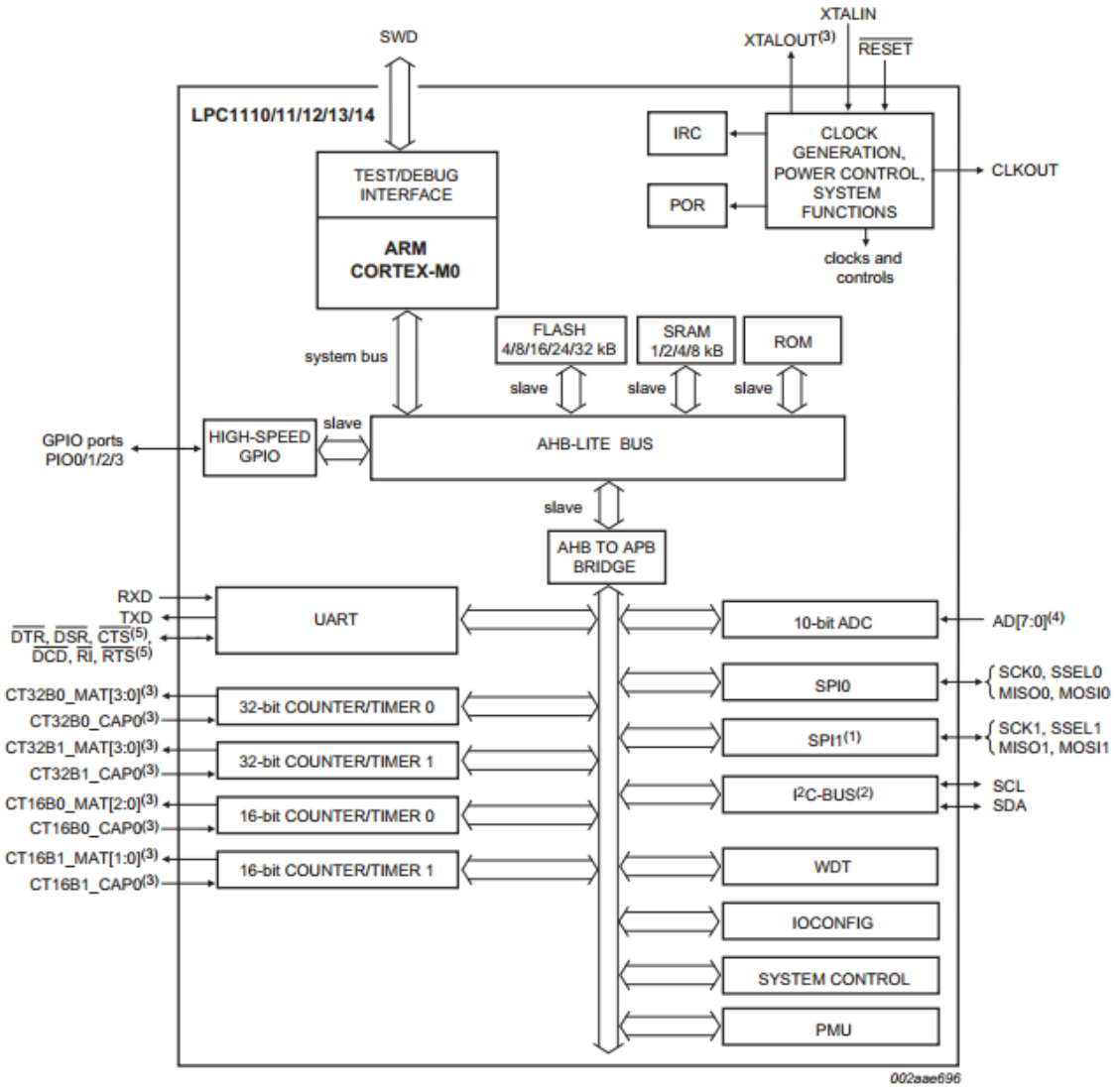

図 4-1 LPC1114

LPC1114 は小規模である分非常に低価格です。28pin DIP 品である LPC1114FN28 は、秋葉原の店頭でも 1 個 100 円程度で販売されています。DIP 品はブレッドボードに挿して試作ができるほか、片面単層基板への実装も用意ですので、低価格製品の制御などに便利に使うことができます。

4.2. CORTEX-M0 コア

LPC1114 に内蔵されている CPU は、ARM 社の 32bitCPU である CORTEX-M0 コアです。

CORTEX-M0 コアは、組み込み用プロセッサの主流である CORTEX-M3 コアよりさらに下位の位置づけで、レジスタのプログラミング・モデルを CORTEX-M3 と同じとしながら、命令を大幅に削って基本的なモノだけにしています。そのため、コアの面積が非常に小さく、低価格、低消費電力製品に広く使われています。

CORTEX-M0 コアは低価格製品ではありますが、32 ビット演算、32 ビットアドレッシング、割り込み、例外、といった基本的な機能はすべて用意しています。そのため、C 言語や C++言語を使った効率の高い開発が可能です。

利用できるコンパイラは ARM 社から発売されているほか、Free Software Foundation の GCC がポーティングされています。GCC は多くのサードパーティで各社のマイコンやボード向けにスタートアップ・ルーチン(CRT)やライブラリが用意されており、手軽に使用することができます。

また、本文書で触れている mbed ももちろん対応しています。

5. USB-シリアル変換アダプタ

この章では、LPC1114 へのプログラムの書き込みに使う USB-シリアル変換アダプタについて説明します。

5.1. なぜ USB-シリアル変換アダプタが必要か

USB-シリアル変換アダプタは LPC1114 へのプログラム書き込みに使用します。

プログラム書き込みには二つの方法があります。一つはデバッガを使う方法です。これは SWD と呼ばれるデバッグ端子からの書き込みで、専用のハードウェアを使います。SWD を使う方法は簡単で、これが使えるならそれが一番です。

もう一つの方法がシリアル・ポートからの書き込みです。この書き込みは LPC シリーズで古くから採用されている方法です。リセット時にブート・モード・ピンを L にしておくことで、マイコンを ROM 書き込みモードに投入し、シリアルからのデータを書き込んでいくという物です。

USB-シリアル変換アダプタは低価格の物は 1500 円程度からあるので、気軽に使えることが利点です。なお、上に述べたようにシリアル変換アダプタからの書き込み時には、リセット信号とブート・モード・ピンを制御するためにスイッチが二つ必要になります。

なお、ここで使うアダプタは TTL 信号入出力です。RS-232C 信号ではありませんので注意してください。

5.2. アダプタの例

ここではいくつかのアダプタを例としてあげます。

5.2.1 UB232R

FTDI 社製の UART モジュールです。基板に IC 等と並べて実装するのにむいています。ピンは DIP8 ピンに見えますが、実は幅が 400mil ですのでソケットに入りません。注意が必要です。

http://www.ftdichip.com/Products/Modules/DevelopmentModules.htm

図 5-1 UB232R

5.2.2 MM- FT 232

サンハヤト製の UART モジュールです。

はじめから電子工作を意識した設計になっており、基板上のピンヘッダにさして使う設計になっています。逆接に注意が必要ですが、制御ピンが出ており、リセット、モード・スイッチを押さなくても FLASH ROM ラ

イタからリセットできる点が便利です。

https://www.sunhayato.co.jp/products/details.php?u=1555&id=02017

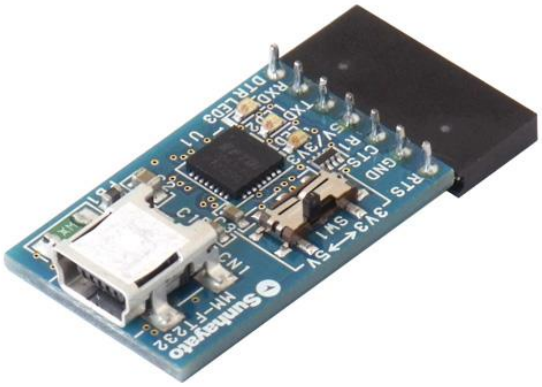

図 5-2 MM-FT232

5.3. アダプタの接続

シリアル・アダプタを使用する際には、接続に注意が必要です。

接続はクロス接続です。つまり、アダプタの TX と LPC1114 の RX、アダプタの RX と LPC1114 の TX を接続します。

書き込みには TX/RX ピンの他、リセット・スイッチとモード・スイッチを用意する必要があります。配線は 6 章の回路図を参照してください。この二つのスイッチの替わりにシリアル・アダプタの制御信号を接続することも可能です。その場合はスイッチを取り去るか、アダプタ側に直列抵抗を入れてアダプタの出力信号が直接 GND に流れないようにしてください。

なお、USB シリアル・アダプタを使うには、PC 側にデバイスドライバをインストールする必要があります。デバイスドライバはシリアル・アダプタのメーカー・サイトからダウンロードできます。

5.4. COM ポートのポート番号

この後の章で、USB-シリアル変換基板越しにプログラムを LPC1114 に書き込む方法を説明します。しかし、そのためには前もって COM ポートのポート番号を調べておく必要があります。

古典的な PC では、COM ポートは順に COM1、COM2、COM3…と割り当てられていました。しかしながら、USB-シリアル変換アダプタを使うと、必ずしもポート番号は連続になりません。また、非常に大きなポート番号が割り当てられる可能性があります。この割り当ては PC によって変りますので、PC を帰る度に調べておく必要があります。

COM ポートのポート番号は、デバイスマネージャによって調べることが出来ます。デバイスマネージャの開き方は、Windows のバージョンによって異なるため、自分の PC での開き方を調べてください。

COM ポートはポートデバイスの下にカテゴリ分けされています（図 5-3）。この例では、COM3 が、USB-シリアル変換アダプタの COM ポート番号です。

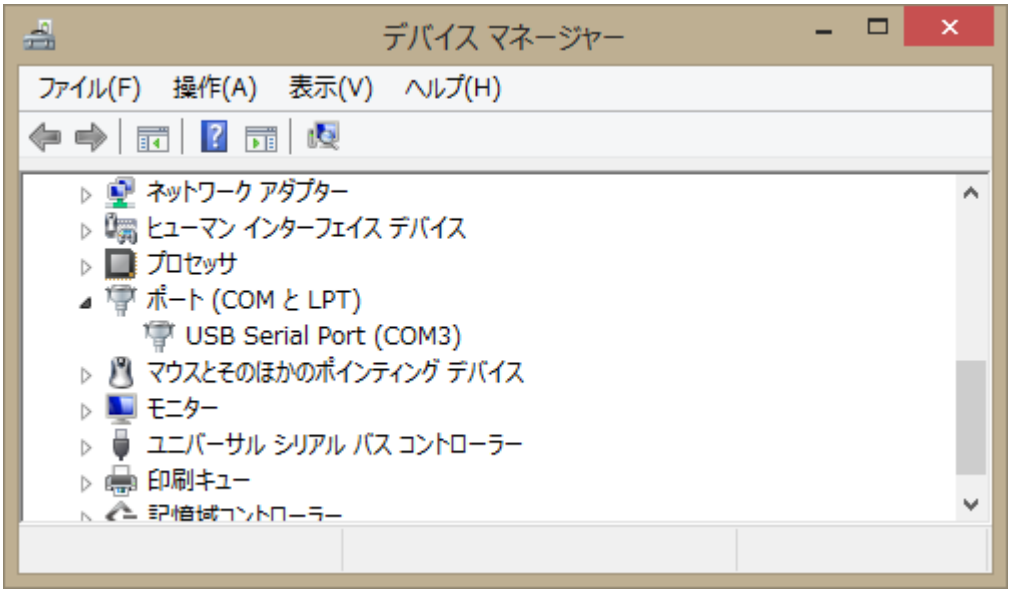

図 5-3 COM ポートの調べ方

6. テスト回路

テストに使用した回路を例として挙げます。

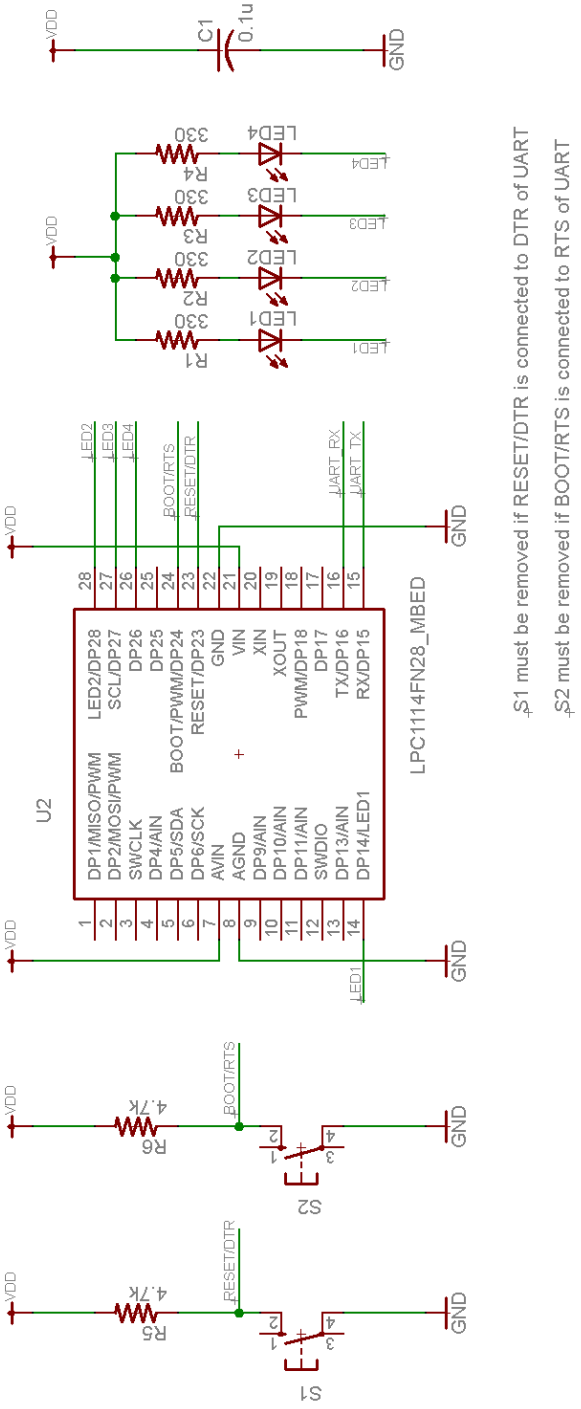

図 6-1 試験回路

S1、S2 は、リセット、ブートの制御をシリアルから行う場合には取り去ってください。あるいは適当な抵抗

を間に入れてもいいでしょう。

7. mbed で LPC1114 のプログラムを作る

この章では mbed コンパイラが自動生成したプログラムをざっと見て、mbed プログラミングについて簡単に解説します。

7.1. テスト・プロジェクトをもう一度

embed 環境で新しいプロジェクトを作る際にはテンプレートを指定することができます。その際、mbed_blinky を選ぶと、LED 点滅プロジェクトが作られることはすでに説明しました。そのメイン・プログラムは以下のようになっています。

main.cpp

```
#include "mbed.h"

DigitalOut myled(LED1);

int main() {
    while(1) {
        myled = 1;
        wait(0.2);
        myled = 0;
        wait(0.2);
    }
}
```

プログラム 7-1mbed_blinky

ユーザーがプログラムを組むときにはこのプログラムをもとに書き替えればいいでしょう。このプログラムは非常に簡単ですが、mbed の特徴をとらえています。

とりあえずはっきりしているのは最初に “mbed.h” を読みこまなければならないということで、これは mbed のライブラリ関数やピン定義を含んだ非常に便利なヘッダファイルです。

7.2. ピンの指定

次にソフトウェアからピンを指定する方法を説明します。mbed のソフトウェアからピンを指定する方法は一風変わってはいますが、いったん理解すると合理的で使いやすい方法です。

説明のために、以下に mbed コンパイラから見た LPC1114FN28 のピンの名前を示します。この図は、mbed.org のプラットフォーム紹介ページより引用したものです[1]。

ここでピンの名前とは、青い枠で囲った dp##という文字列です（##は数）。たとえば dp1、dp6 などがそれです。よくよく比べてみると分かりますが、dp という文字列の後に来る数は、実は LPC1114 の物理的なピン番号になっています。これが mbed 環境の特徴的なところです。マイコンは内部に入出力ペリフェラルを持っており、そのレジスタを制御することで、マイコンごとに定められたピンを制御できます。この方法は、ソフトウェア・技術者から見ると実際のピンではなく、ペリフェラルのビットを操作していることになります。

[1] http://mbed.org/platforms/LPC1114FN28/

ペリフェラルのビット操作はマイコンのマニュアルが指示している通りのことですので間違いではありません。しかし、実際の物理ピンとは異なる境界でレジスタ境界が現れるなどから、プログラムの書き換えなどは不便なことがあります。

mbed のやり方だと、回路図を見てピン番号をソフトで指定すればピンの指定が終わりますので大変便利です。

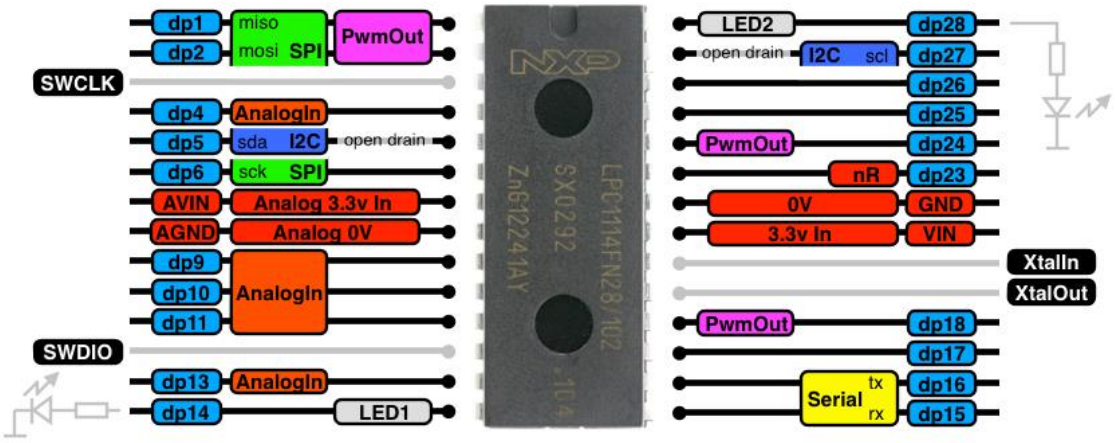

図 7-1 mbed LPC1114 のピン

さて、このピン指定が実際のプログラムでどう扱われているか見てみましょう。先のプログラムを見ると、ピン番号が指定されるのは 3 行目だけです。

```
DigitalOut myled(LED1);
```

これは C++言語の変数宣言です。意味は「DigitalOut 型の変数 myled を定義する。コンストラクタの引数には LED1 を与える」ということです。DigitalOut はクラス型であり、内部に変数のほかメソッドを持つことができます。その DigitalOut 型は初期化時に引数としてピン番号を割り当てます。つまり、ディジタル出力を制御する変数をピン番号に作ることになります。

プログラムではピン番号が LED1 となっていますが、これはマクロであり、図のとおり、LED1 は dp14 ピンを表します。これを変更して LED2 を点滅させたい場合、コンストラクタの引数は LED1 あるいは dp28 を与えてください。

ピンに信号を出力するのは簡単で、上のようにピンを割り当てた DigitalOut 型の変数に対して、1 また 0 を代入してください。1 を代入すれば H が、0 を代入すれば L が出力されます。

7.3. タイマーの制御

mbed 開発環境はタイマーを使った制御にも対応しています。

今回のサンプルプログラムでは、wait()関数がその機能を持っています。この関数は浮動小数点型の引数を受け取り、それを秒と解釈して指定秒数だけ待ちます。以下の例を見てみましょう。

```
wait(3.1);
```

この場合、3.1 秒の間プログラムは停止し、その後自動的に実行を継続します。

mbed_blinky の場合は H を出力して 0.2 秒、L を出力して 0.2 秒待っています。つまり、0.4 秒周期で動作しています。

7.4. コンパイル

完成したプログラムは、mbed 開発環境の compile ボタンを押すことで、実行形式に変換できます。実行形式は拡張子.bin を持つバイナリ・ファイルで、コンパイルが終了するとサーバーから自動的にダウンロードされます

このファイルは PC にダウンロードされるだけです。ターゲットである LPC1114FN28 に書き込むには、別のツールが必要です。それについては次の章で説明します。

8. ROM ライタ

mbed コンパイラの生成したプログラムを実行するには、そのプログラムを LPC1114FN28 の内蔵 Flash ROM に焼かなければなりません。この章では、そのためのツールである ROM ライタと、ROM ライタ用にファイル形式を変換する HEX コンバータについて説明します。

8.1. mbed コンパイラの生成物

前の章で説明したとおり、mbed コンパイラは実行ファイルを生成します。この実行ファイルは PC 上にダウンロードされますので、何らかの方法で LPC1114FN28 に書き込まなければなりません。

その書き込みにはいくつかのツールが使えます。しかしながら、以下の条件を全部満たしているフリーのツールとなると、実は見当たらないのです。

- 商用にも自由に使える
- バイナリ・ファイルを直接読み込んで LPC1114FN28 に書き込むことが出来る。
- Windows 用の実行形式で配布されている。

そこで、以下ではバイナリ・ファイルをいったん Intel HEX フォーマットに変換してから LPC1114 に書き込む方法を簡単に説明します。

8.2. BIN2HEX とは

BIN2HEX は、ht-lab によるフリーな HEX コンバータで、Windows 向けに開発されています。

http://www.ht-lab.com/freeutils/bin2hex/bin2hex.html

BIN2HEX を使って、mbed コンパイラが生成したバイナリ・ファイルを、ROM ライタの入力として扱える Intel HEX フォーマットに変換することが出来ます。

変換方法は簡単です。コマンドラインから次のように実行します。

```
bin2hex mbed_blinky_LPC1114.bin mbed_blinky_LPC1114.hex
```

最初の引数は mbed コンパイラが生成したバイナリ・ファイルです。二番目の引数は Intel Hex ファイルを出力するファイル名です。

BIN2HEX はバイナリ・ファイルの最大サイズが 64kB ですが、LPC1114FN28 の ROM サイズは 32kB ですので問題ありません。

8.3. LPCSP とは

LPCSP は ChaN 氏による NXP マイコン向けの ROM ライタで、Windows のコマンドラインで動作します。

http://elm-chan.org/works/sp78k/report.html

LPCSP を使って、BIN2HEX ファイルで変換した mbed コンパイラの生成ファイルを LPC1114FN28 に書き込む事が出来ます。

書き込みは大変簡単で、次の順序で操作します。

1. 5 章に説明したとおりに LPC1114 基板と PC を USB-シリアル変換基板を通して接続する。
2. 「COM ポートのポート番号」に説明した方法で、COM ポートの番号を調べておく。

3. LPC1114FN28 のリセット入力とモード入力を同時に L にする。
4. リセット入力を H に戻す
5. モード入力を H に戻す
6. コマンドラインで LPCSP を実行する
7. LPC1114FN28 を再度リセットして、書き込んだプログラムを実行する

コマンドラインで実行する LPCSP には以下のように引数を与えます。

```
Lpcsp mbed_blinky_lpc1114.hex -P3
```

-P 引数に続く数字の 3 は、USB-シリアル変換基板が COM3 に割り当てられている場合の設定です。この部分は、各自の PC のマッピングを調べてから指定してください。

なお、USB-シリアル変換基板に必要な制御線が出ていれば、リセット入力やモード入力をスイッチで操作しなくてもコマンドライン引数からコントロールすることが出来ます。これについては説明しませんので、興味のある方は調べてみるといいでしょう。

8.4. 便利なツール

LPC1114FN28 への書き込みには他にも便利なツールがあります。ここでは二つだけ紹介しておきます。

8.4.1 lpc21isp

lpc21isp は、定番と言ってもよい書き込みツールです。

http://sourceforge.net/projects/lpc21isp/

Windows, Linux, MacOS で動作するよう開発されており、商用開発への利用も可能です。また、バイナリ・ファイル、HEX ファイルいずれも受け付けることが出来るため、BIN2HEX が不要です。

唯一残念なのは、実行形式のファイルが配布されていないことです。そのため、自分でビルドして使うことになります。これはオープン・ソースとしてはごく当たり前のことですが、ソフトウェア開発に携わっていない人には敷居が高いかも知れません。

このツールも、制御線を使ったリセット入力、モード入力のコントロールができます。

8.4.2 書き込み補助ツール

LPCSP や lpc21isp をコマンドラインではなく GUI から利用する補助ツールもあります。

http://mbed.org/users/okini3939/notebook/flash-program/

筆者は使ったことがありませんが、試してみるのもいいかもしれません。

9. 開発のサイクル

mbed を使った LPC1114FN28 のプログラム開発は、

1. mbded IDE によるプログラム入力
2. mbed コンパイラによる実行ファイルの生成
3. HEX ファイルへの変換
4. ROM ライタによる LPC1114FN28 への書き込み

のような流れになります。実行中にエラーがみつかったら、原因を調べて 1 から繰り返します。

10. 終わりに

10.1. その他の mbed プラットフォーム

mbed は元々ボードと一対一対応のプロジェクトでした。つまり、mbed 開発環境は mbed ボード専用でした。

いまでは NXP に留まらず、Fressscale、ST、Nordic Semiconductors の SoC を使った多種のボードに対応しているほか、この文書で説明したとおり、ボードでは無く IC にまで対応を広げています。

対応プラットフォームについては、mbed のサイトを参照してください。

http://mbed.org/platforms/

10.2. コミュニティ

mbed はクラウドサービスであることを利用して、サイト内にフォーラムを作っています。ここではユーザー同士の質疑が行われています。また、mbed アカウントに付随して自分自身の WEB ページを作ることが出来るため、多くの人が技術的なヒントを公開しています。

さらに踏み込んで、mbed で利用可能な多くのソフトウェアやミドルウェアが公表されています。

日本でもユーザーによる数多くの情報をネット上で見る事が出来ます。

10.3. 参考リンク

- LPCZone：NXP LPC マイコン情報　http://www.nxp-lpc.com/
- [lang:ja] mbed LPC1114 での遊び方
  http://mbed.org/users/ytsuboi/notebook/getting-started-with-mbed-lpc1114-ja/
- Mbed LPC1114 を試す
  http://morecatlab.akiba.coocan.jp/lab/index.php/2013/10/mbed-lpc1114/comment-page-1/

10.4. 履歴

- 初版 2014/Apr/20
- 第二版 2014/Apr/24