

マイコン・ツールキット入門

2.5

mbed 入門

60 秒で Hello world!

■岡野 彰文 (NXP セミコンダクターズジャパン)

概要

現在販売されているマイコンは非常に多種多様ですが、これを使って「何か」を作るには非常に面倒な作業が必要でした。初めての人にはこれらが大きな障壁になり、なにかと「とっつきにくい」ものとなってしまっていました。

■これまでのマイコンでは..

▶セットアップが困難

「初めての人にとっての困難」には、例をいくつか挙げることはできますが、中でもまず最初にして最大のものは「開発環境の構築(一連のツール類のインストール)」でしょう。

▶インストール完了まで

マイコンを使い始める際には、まず最初にそのマイコンに適した開発環境を探し、それが動くパソコンと OS を準備、開発ソフトをインストール。さらにサンプルコードを探してきてコンパイル。実行ファイルができたならマイコンへ書き込み。これには専用ツールとそのドライバも必要です。さあここで初めて自分の「何か」をつくり始めることができます。

▶マイコンの仕様を確認、使い方を習得

いよいよ開発をスタートするのですが、またもう一難。マイコンは各チップによってその使い方が異なります。たとえばマイコンのあるピンでハイ/ローを出力させるだけでも、所定のレジスタを所定の方法で初期化して、適切な操作が必要です。これにはサンプルコードやチップのユーザマニュアルを細

かく確認していかなくてはなりません。

▶ライブラリやサンプルコード

さらにより高度なこと、たとえば USB を使ったり、インターネットに接続するような場合、このコードを自前で用意するのは非常に大変です。ほかから調達するにしてもなかなか困難であったりします。

▶解決法は?

近年の電子工作の火付け役のマイコン基板、Arduino はこのような問題を上手く解決した一例と言えるでしょう。

ここで紹介する「mbed」はさらに進んだ解決方法を提供します。

■なぜ「mbed」なのか?

▶60 秒で Hello world!

mbed は「たった 60 秒で最初のプログラムを作ることができる」ことをキャッチフレーズにしています。ここから説明することは次の URL のちょうど 1 分のビデオにまとめられています。http://mbed.org/blog/entry/138/

▶高速プロトタイピングツール

今回ここで紹介する「mbed」(エンベッドまたはエムベッドと読みます)は、英国 ARM 社が開発した「高速プロトタイピング」ツールです。

製品の設計を始める前に、アイデアを実際に動くものにしてみることにより、そのコンセプトを自分で検証してみたり他人にデモを見せることが可能になりよりよいものづくりが可能になります(図-1)。

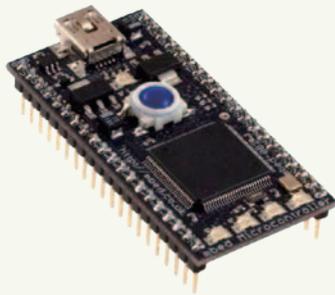


図-1 mbed 基板

もちろん mbed はプロトタイピングだけにとどまることなく、そのまま最終製品への組み込みや、教育のツールなどとして広範囲な応用が可能です。

■ 特長

mbed は、これまでのマイコン基板にはない、いくつかの特長を備えています。

▶32bit ARM プロセッサ搭載：

ARM Cortex-M3 コアを採用した NXP 社製 LPC 1768 を搭載。96MHz クロック動作で高速処理が可能です。またこのチップにはフラッシュ 512KB, SRAM 64K のメモリが内蔵されており、より余裕のあるプログラムを動作させることができます。

▶オンライン・コンパイラ：

USB のついたパソコンと最新の Web ブラウザさえあれば開発が可能です。面倒な開発ツールのインストールは必要ありません！

mbed は Web ブラウザからアクセスし、サーバ上で動作する「オンライン・コンパイラ」で開発を行います。

▶言語, ライブラリ：

プログラム言語は C++。mbed のハードウェアや, Ethernet や USB はクラス・ライブラリにより高度に抽象化されているため、その詳細に因われることなくアイデアの実装に集中できます。

もちろん, C++ を知らなくても、簡単なライブラリの使い方 (API) さえ理解すれば、あとは C 言語の知識でプログラム可能です。

mbed の Web サイトではコードやアイデアの共有のためのツールを完備しているので、豊富な参

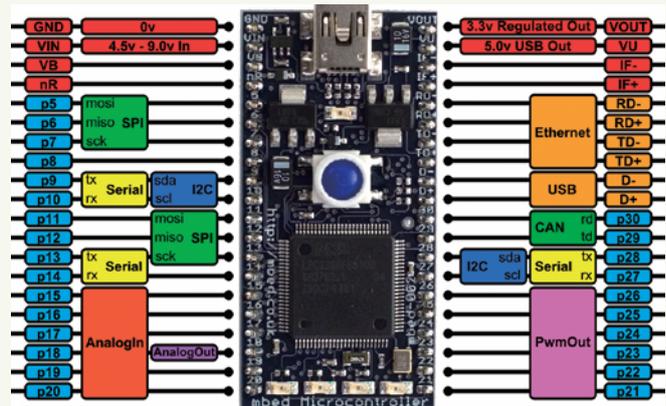


図-2 インタフェース

考資料やサンプルコードなどから新しいアイデアを得ることも可能です。

▶ドラッグ・アンド・ドロップで書き込み：

マイコンへのプログラムの書き込みに専用のツールは不要です。パソコンに接続すると USB ストレージデバイス (USB メモリ) として認識される mbed に、コンパイル後サーバからダウンロードしたファイルをコピーするだけです。

▶基板形状, インタフェース：

40 ピンの DIP 形状の基板になっており、ブレッドボードやユニバーサル基板での使用に便利です。

電源は USB ポートからの給電のほかに 4.5V ~ 9V の外部電源で動作、また周辺チップへ供給するための 3.3V 出力が用意されています。

各ピンはデジタル入出力はもちろん、アナログ入力 6 本、アナログ出力 1 本、シリアル 3 本、それぞれ 2 系統の I2C と SPI を使用することができます (図-2 参照)。

▶CAN, USB, Ethernet：

さらに CAN, USB (ホスト/デバイスは切替), Ethernet が用意されています。

USB, Ethernet は mbed に物理層が用意されているため、コネクタを取り付けるだけで動作させることができます。

▶豊富なライブラリとサンプルコード：

ライブラリが高度に抽象化されていることは先に述べましたが、これを使ったサンプルが豊富なことももう 1 つの特長です。

同サイト内で共有されているライブラリとそのサンプルコードは、1クリックで自分のコンパイル環境にそれを持っていくことができ、即座に動作を確認できます。

各ピンの制御等はクラス・ライブラリによるAPIが用意されインスタンス(C言語なら変数)への代入や読み出しで簡単に行うことができます。これらのAPIについての詳細はWebサイトに詳細と豊富なサンプルが提供されています。

高度なライブラリについては、たとえばテキスト表示液晶やOLEDを使うサンプルではprintfを用いた印字や簡単な描画ライブラリ、Ethernetを用いたサンプルではHTTPサーバの実行やメールサーバへのアクセスやTwitterへの投稿、SDカードやUSBメモリのサンプルではfopen等の標準関数でのアクセスがすでに用意されています。

必要なパーツ

本稿では以下のパーツを使います

- mbed (基板本体) [s1][s4][s5][s8][s10]
- ブレッドボード [s1][s2][s4][s5]
- 220 Ω程度の抵抗器 [s1][s2][s4][s5]
- LED [s1][s2][s4][s5]

これらのすべての部品は秋葉原や日本橋の有名な電子部品店や通販サイトで購入可能です。

準備とインストール

■ mbed をパソコンに接続

mbed に同梱されている USB ケーブルで mbed をパソコンにつなぎましょう (これは通常の USB ケーブルと同じものなので他のケーブルでも代用可能です)。

パソコンは Windows でも Mac でも Linux でも構いません。接続すると mbed はパソコンからストレージデバイス(USBメモリ)として認識されます。

この中には「MBED.HTM」というファイルが入っており、これをダブルクリックすると Web ブラウザが開いて mbed の Web サイトにつながります。

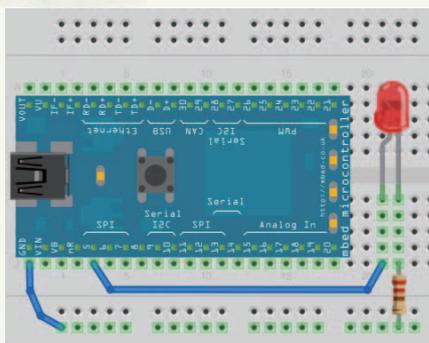


図-3 配線図

■ 開発環境は「Web ページを開く」だけ

mbed を使う場合、開発環境のインストールは必要ありません。開発に必要なすべてはサーバに用意されています。

ただし、このオンライン・コンパイラを使うには自分のアカウントが必要です。1台の mbed にはアカウントを作るためのキーが含まれています。最初にこのキーを使って自分のユーザ・アカウントを作りましょう (アカウント作成の詳細は次の URL を参照：http://mbed.org/users/nxpfan/notebook/lets_get_started_jp/)。

登録後、自分のソースコードを Web サイトに保存、コンパイルや公開などができるようになります。

プログラミングと動作確認

■ LED をつなぐ

ブレッドボードを使って mbed に LED を接続します。LED に流れる電流を制限するための抵抗を忘れずに直列につないでおきましょう (図-3 参照)。

■ ソースコードとコンパイル

この例では単純な LED の点滅を試してみます。

mbed サイトにログインした状態で右上に表示されている「Compiler」のリンクをクリックします (図-4 参照)。

そうすると mbed の IDE (統合開発環境) 画面が Web ブラウザ内に表示されます。

次に左上に目をやると「New」と書かれた、新しいプログラムを作るためのボタンが目に入ります。これをクリックするとダイアログが開き、プログラ

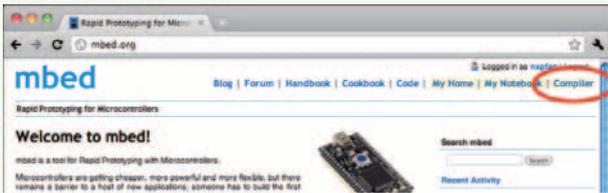


図-4 mbed サイト

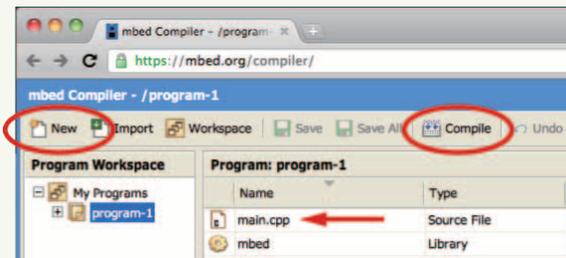


図-5 IDE (統合開発環境) 画面

ムにどのような名前を付けるか聞いてきます。残念ながら日本語の名前をつけることはできません。アルファベット、数字、アンダースコア ("_") の文字を使うことができます(図-5 参照)。

名前を付けるとプログラムの「ひな形」が用意されます。「main.cpp」と「mbed」の2つのファイルができています。「main.cpp」はプログラムのソースコード、「mbed」は mbed を動作させるために必要なライブラリです。

「main.cpp」をダブルクリックすると、そのファイルを開いて編集できます。このコードは mbed に搭載されている LED を点滅させるサンプルになっています。コードを見ると、(1) まず mbed ライブラリを使うために mbed.h ファイルをインクルードし、(2) DigitalOut のインスタンス (C 言語の変数のに相当)「myled」を作成。これを作る際の「LED1」の指定で mbed 基板上的 LED を指すように。(3) プログラムがスタートすると myled に 0.2 秒ごと、1 と 0 が交互に代入され、これが LED の点滅となります。

ここでは 5 番ピン出力を切り替えたいのでソースコード 3 行目の「LED1」の部分で「p5」に書き換えておきます(図-6 参照)。

これで準備ができました。

いよいよコンパイルします。

```
#include "mbed.h"
DigitalOut myled(p5); //ここを変更します

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

図-6 サンプルコード

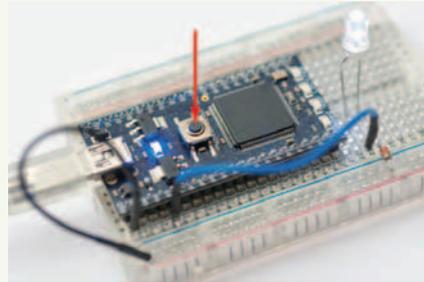


図-7 リセットボタンを押す

ページ上部中ほどの「Compile」ボタンを押すと、サーバ側でコンパイルが行われます。成功するとファイル(.bin という拡張子になっている)をダウンロードするように言うてくるので、これを mbed 内に保存します。

あとは mbed 基板真ん中についているリセットボタンを押せば、LED が点滅します(図-7)。

まとめ

ここで紹介したように mbed は非常に簡単に始めることができ、かつ充実したライブラリなどを用いて高機能なアプリケーションを作成することができます。このライブラリやサンプルコードは日々、ユーザによって追加されています。

また作成したプログラムは mbed だけでなく、単体のチップ(LPC1768)でも動作するため、自作の基板を用意してその上で実行させることもできます。mbed の世界をどうぞお楽しみください。

<http://www.nxp-lpc.com/>

(2011 年 5 月 9 日受付)

■ 岡野彰文 akifumi.okano@nxp.com

NXP セミコンダクターズでインタフェース関連製品のアプリケーションを担当。マイコンチームに「動かしてみよう」と手渡された mbed を使ってみて大きなショックを受け、それ以降 mbed を担当、製品の評価やデモに活用している。