

mbed 講習会

2013 年度生用

梶本研究室

ver.1 2012.3.25

ver.2 2012.4.11

ver.3 2013.1.4

Change Log

Ver2 全般的な回路図の微修正、指導方法について追記

Ver3 4章で割り込みの追加、8.3、8.4の課題を変更

1 はじめに

3年生の宿題（DxLib+mbed）で、ブレッドボードを用いた mbed の基本的な導入については済んでいるものとして、ここでは mbed をより本格的に活用していきます。

ここで作成したモジュールは研究の中で標準的なインターフェースとして使用しますので、きれいに作りましょう。

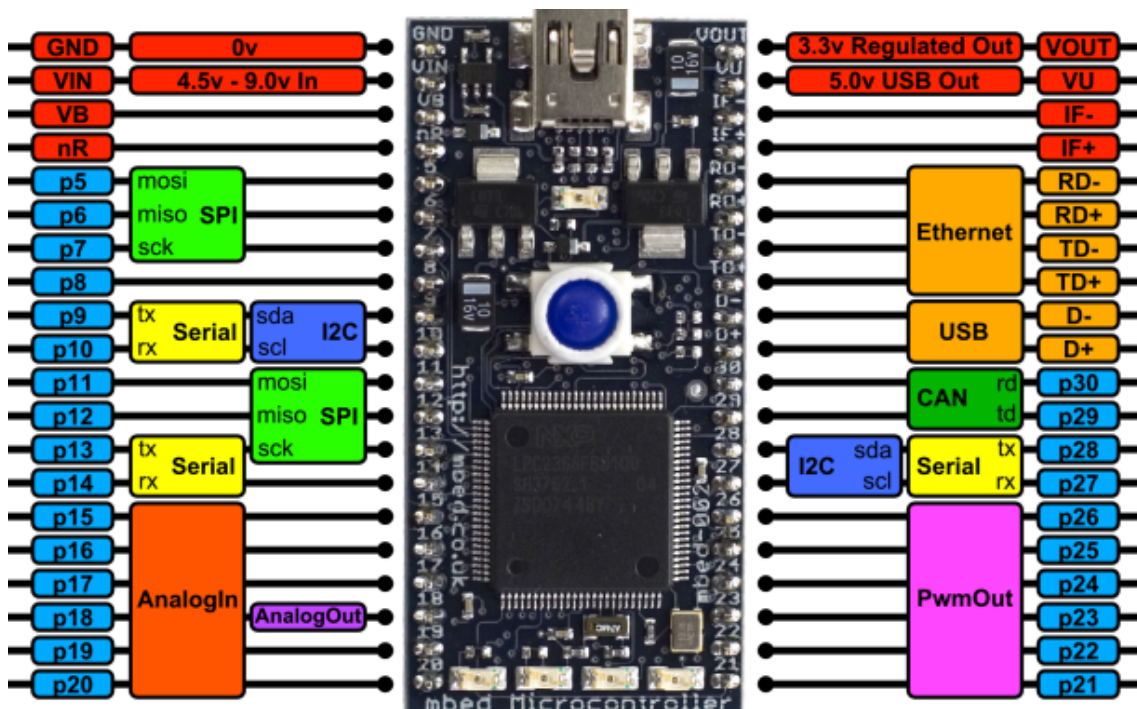


図 1 mbed ピン配置(<http://mbed.org/handbook/mbed-NXP-LPC1768>)

1.1 手元において欲しい本

電子工作に関する参考書は沢山ありますが、標準的な研究室新入生（ハンダ付け経験なし）には「電子工作の素」を勧めます。



1.2 課題の進め方

mbed はできることがあまりに多いので、まず研究室標準のインタフェースとして使えるところまで皆で行います。その後、個々のテーマに関しては選択課題として振り分けて行います。発表会はこの選択課題について行います。

全員が行うのは以下のとおりです。これは最初の一週間で行います。

- 2 章「回路の準備」：ブレッドボードで行ったことを自作の基板で行います。
- 3 章「外付けスイッチ」：外付けスイッチによる入力を行います。割込みの練習を兼ねます
- 4 章「LED の駆動」：LED を点灯させます。
- 5 章「RC サーボモータの駆動」：RC（ラジコン）サーボモータを動かします。
- 6 章「アナログ出力」：内蔵の DA 変換器を使います。
- 7 章「SPI 通信」：SPI 通信によるポートの拡張例として DA 変換ポートの増設を行います。

一日目 2～3 章、二日目 4～5 章、三日目 6 章、四日目 7 章、五日目予備日というペースです。

また選択課題は以下のとおりです。これは後半の一週間で行います。

- 8.1 「7 セグメント LED」：7 セグメント LED を動かします。シフトレジスタを使う例題にもなっています。
- 8.2 「振動呈示型データグローブ」：曲げセンサと振動モータを動かします。

- エラー! 参照元が見つかりません。「通信機能を持つセンサモジュールの使用」: I2C 通信機能を持つセンサを使います。ここでは加速度センサを使って衝撃を計測、振動出力することによって叩いた対象の材質感を表現する事を試みます。
- 8.4「超音波センサの使用」: 割込みプログラムの実例として超音波センサを使います。2つ使うことで擬似的にテルミンを作ります。
- 8.5「リニアアクチュエータの駆動」: ポテンショメータの値を読んでリニアアクチュエータを動かします。Hブリッジ回路を用います。
- 8.6「DC モータの駆動」: エンコーダカウンタの値を読んで DC モータを動かします。Hブリッジ回路を用います。
- 8.7「外付け AD」: 12bit、8ch の AD 変換器を二つ使い、またフィルム状力センサを用いて力分布を測定します。

1.3 レポート

最後にレポートを書いてください。添削します。

- Word を標準とします。Word 講習を兼ねていますので、図表番号への参照などの参照機能およびアウトライン定義は必ず行なって下さい。
- 実験中に取得した図（特にオシロスコープの画面キャプチャ）を貼って下さい。
- 回路図など、テキストにある内容を繰り返す必要は無いので、結果と考察のみ述べたコンパクトなレポートとして下さい。
- コツはレポートを書きながら実験を行うことです。これは研究の場面でも同じです。データはリアルタイムに文書化していきましょう。
- 提出されたレポートの添削と回覧は研究室全体で行います。

1.4 指導方法（指導者向け）

2012 年にやってみた感じでは次の指導が必要です。ほうっておいて野生力をつけるという方針もありえますが、この講習に限って前半は集中的に指導し、後半も適切な苦勞をさせてください。

- 初日冒頭、2～5 章の説明。
 - ハンダ付けの方法
 - スイッチ回路の動作の原理
 - LED 回路の抵抗計算の原理（LED の特性）、テストで LED の極性を知る方法
 - RC サーボの駆動の前提となる PWM の説明
 - オシロスコープによる測定（オートセット、CH1 をトリガとする方法）

- 3日目冒頭、6～7章の説明
 - オペアンプ入門
 - SPI 入門、シフトレジスタの説明から、LTC1660 のマニュアルを見ながらタイミングチャートと送信データ構造の理解
 - オシロスコープによる測定（外部トリガを使う方法）
- 2週間目冒頭
 - テーマ割り振り（これは1週間目最後でも可）、部品配布
 - ファイル経由のプログラム間通信の説明
 - 基板間の接続をする課題が多いので、幾つかのコネクタの使い方の説明。圧着端子とフラットケーブル用端子は少なくとも教える。熱収縮チューブの使い方も

この講習は研究室的技能の中心となりますので、集中的に指導しましょう。特に2週目は、1週目と異なり講習資料は不完全ですので、マニュアルの隅々まで徹底理解しないと出来ません。多くの人はそうした体験は初めてで、ほぼ 100%できないはずです。それを「**マニュアルをしっかりと読めばできる**」という**体験**に持っていくのは指導者の力です。苦勞させつつ無駄な苦勞はさせないようにコントロールしてください。

問題解決（デバッグ）の効率的な手順も指導してください。例えば望ましい値が計測値として出ない場合、最初はほぼ 100%「プログラムのどこがわるいのだろうか」でフリーズします。これをまずテストで確かに電圧が出ていることを確認して「ハードの問題かソフトの問題かを切り分ける」、次にターミナルソフトで表示して「通信の問題かそれ以外かを切り分ける」、という二分岐による**高速な可能性探索**を体得することが必須です。実際初めてだと、**テストやオシロスコープがデバグに使うものという概念がありません**（レポート用のデータを取るものという認識）。「**テスト／オシロで測ってみた？**」は最初に聞くようにしてください。

また2週間目の内容は一種のプロジェクトなので、**プロジェクトマネジメントの観点**からの指導をぜひお願いします。はじめはほぼ 100%プロジェクトマネジメントの感覚がなく、「回路を全部作り」→「プログラムを全部書き」→「全然動かなくてハマる」のが通例です。例えば 8ch 動かす回路だったら、1ch 分だけ回路を作り、その部分のプログラムを書いてみて、という感覚は必ず身につけさせてください（なおプロジェクトマネジメントのできていない学生は、同時に「自分が何をやっているか」理解していない確率が高いです。心細いのでとりあえず回路づくりの作業に打ち込んでいるのかもしれませんが。早期発見してください）

プログラムだけをとっても、このプロジェクトマネジメントが行われているかに注意してください。if 文、for 文、while 文を完璧に理解していても、ある課題に対してどのような

プログラム構造を作るべきかという大きな視点は持っていないのが通例です。これはフローチャートを書けというのではなく、ホワイトボードで擬似的なプログラムを書くという感じで議論しながら、プログラムの構造を作っていく感覚を身につけさせてください。ホワイトボードでの議論に慣れさせるためにも意図的にたくさん使ってください。

最後に、「わかっていない『はい』」は徹底的に糾弾してください。言うまでもなく研究の基本ですが、研究のディスカッションの場で糾弾しても、うるさい先輩と思われがちです。しかし講習会場で徹底追求すれば、それは課題を解くことに直結するので利益が見えやすいです。つまり「わかっていないことをごまかさないこと」が実際に有益であることを体験できる貴重な機会になります。

以上、本講習は単なる技術紹介を超えて

- マニュアル、ソースを完全に理解する必要があるという感覚
- デバグ（二分岐）の概念、テスタやオシロがデバグ用ツールであるという常識
- プロジェクトマネジメントの概念、ソースコードは構造設計が大切という体験
- わかっていないことをごまかさない習慣

を体得することを目的としています。

2 回路の準備

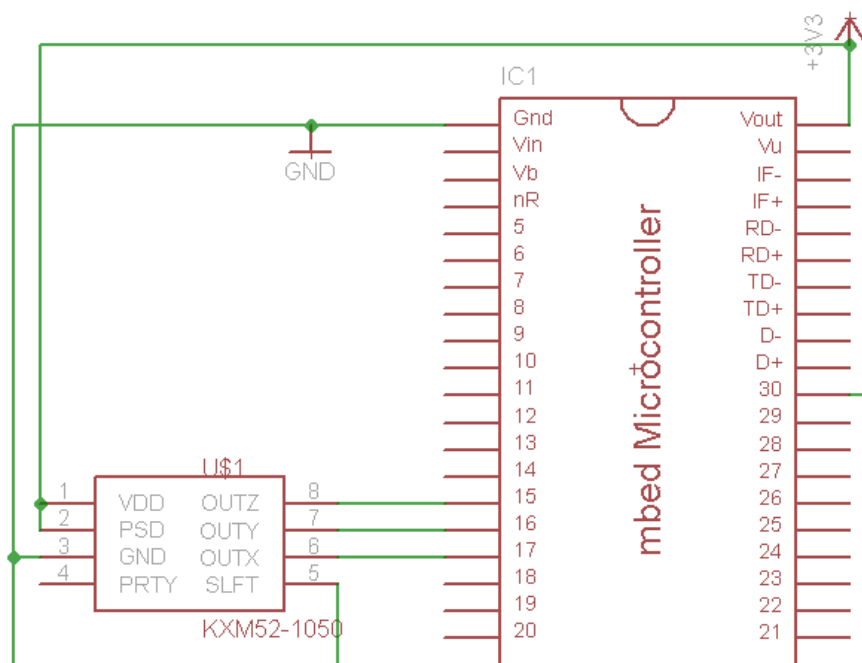
春休みにブレッドボードを用いて作った加速度センサ回路を作成する。ただし回路は配線しやすいように微修正してある。

配線は錫メッキ線、あるいはジュンフロン線（非常に細い）を用いる。無用に太い線は使わない（ただしモータを駆動する部分や電源部分には太い線を用いる）。加速度センサは後で外すので 8 ピンの IC ソケットを用いる。mbed 用には 20 ピンのシングルソケットを 2 つ用いる。

ハンダ付けの方法は下記リンクや「電子工作の素」などを参照する。

ハンダ付けの方法 <http://www.youtube.com/watch?v=S5f7jueQHr8>

良いハンダ付け形状など <http://homepage1.nifty.com/x6/elecmake/solder.htm>



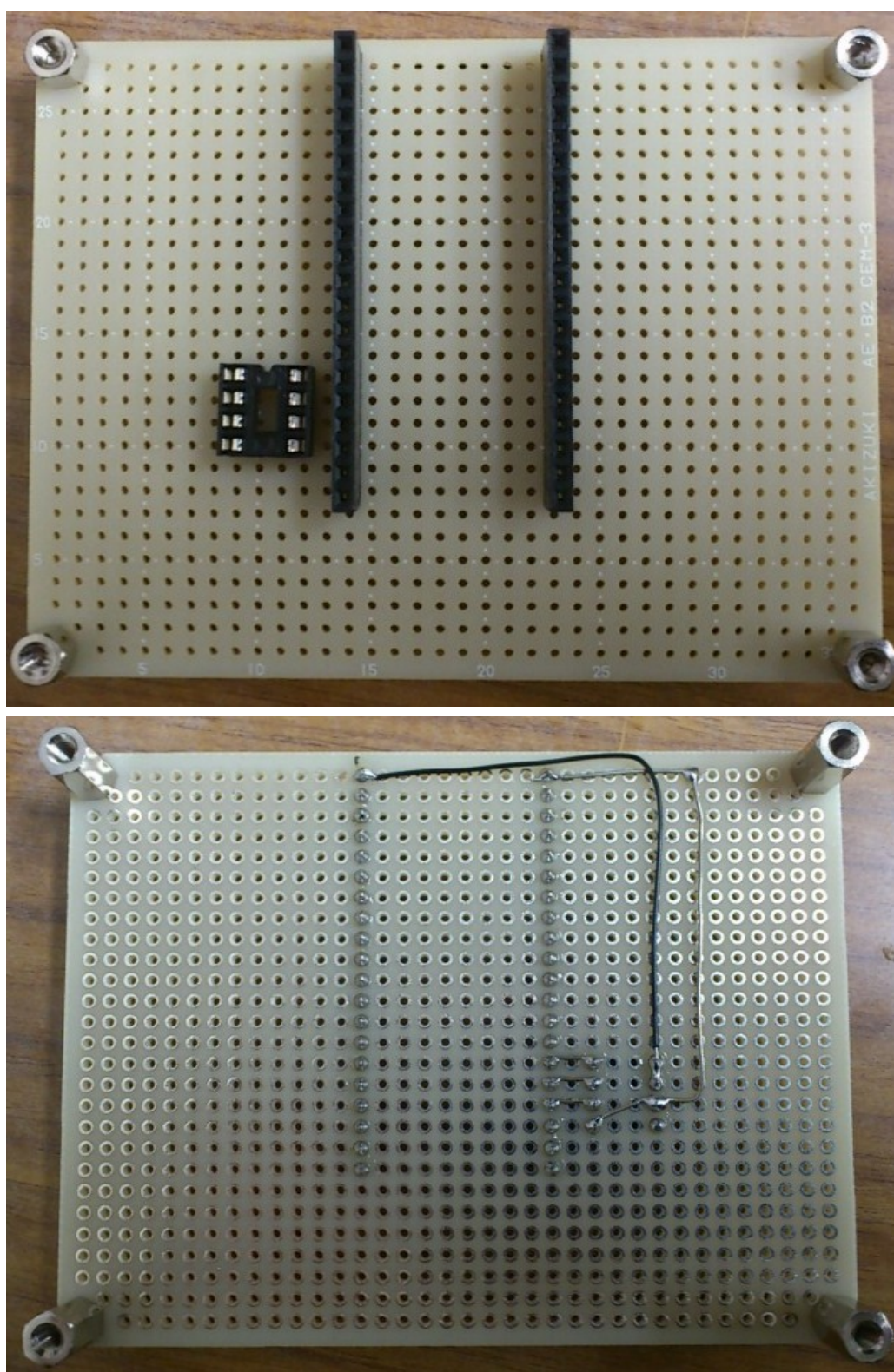


図 2 加速度センサ回路

3 外付けスイッチ

DigitalIn を用いて、外部からスイッチ入力を受け付けるようにする

<http://mbed.org/handbook/DigitalIn>

回路構成上の注意：タクトスイッチの向きに注意。90 度間違えることがある。ボタンを押したときにどこが導通するかはテストでチェックする。

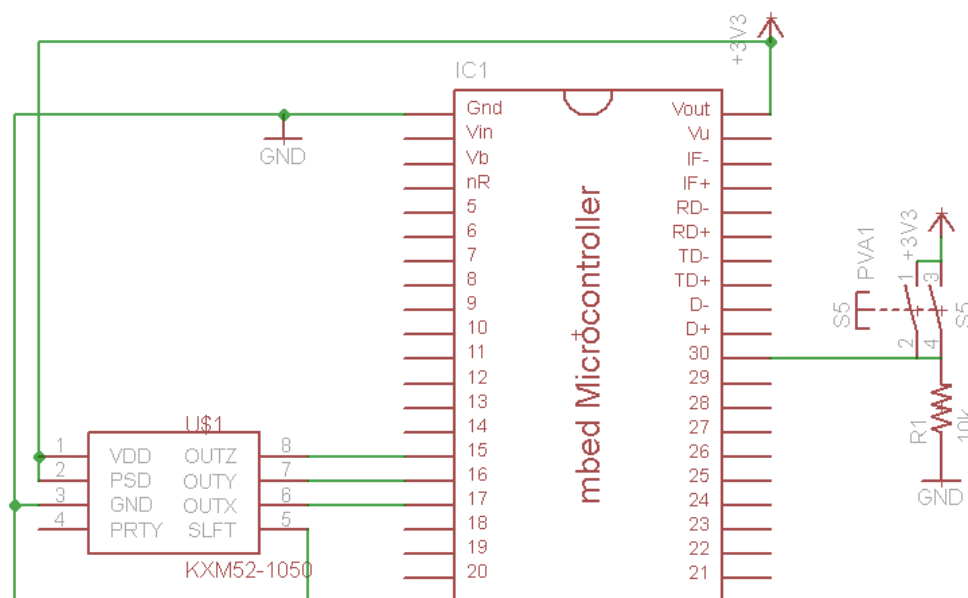
web のサンプルを使って mbed 上の LED を光らせてみる。ただし入力ピンは p5 ではなく (P5 は後で使う)、p30 とする。

[課題1] スwitchのステータスをシリアル通信で PC に伝えるようにする。例えばスイッチが押されている間は”a”を、押されていない間は”b”を送信し続けるなど。PC 側はシリアル通信ソフトを用いる (TeraTerm, RS232c など)

Rs232c <http://www.vector.co.jp/soft/win95/hardware/se369900.html?ds>

TeraTerm <http://sourceforge.jp/projects/ttssh2/>

[課題2] スwitch回路に抵抗が必要な理由を考察せよ



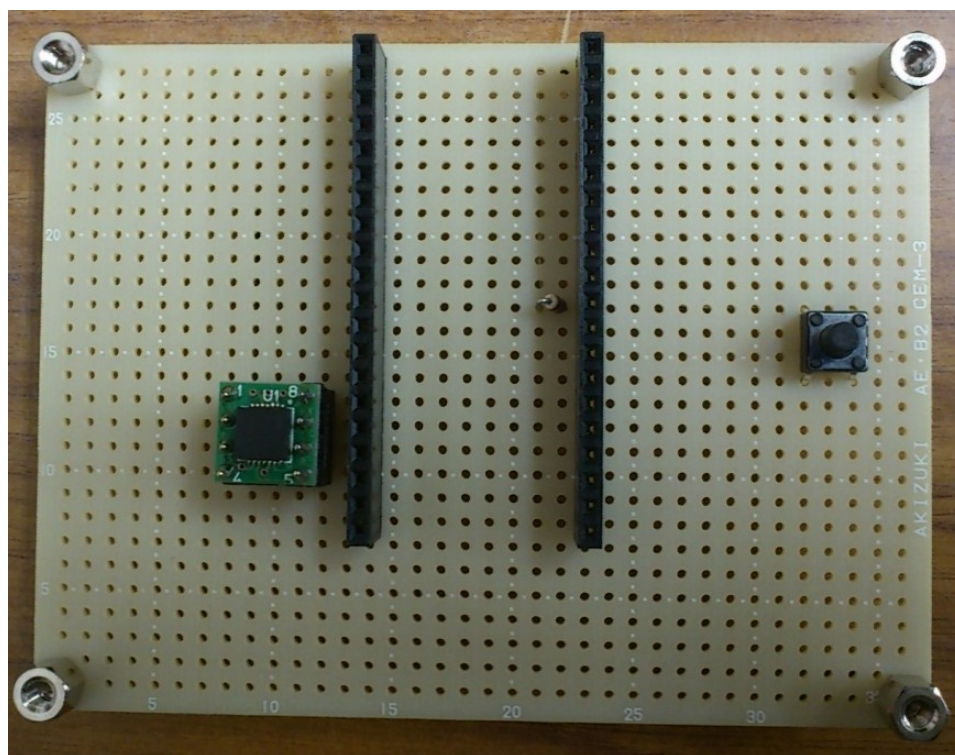


図 3 タクトスイッチ回路

4 mbed で LED 駆動

外付け LED を駆動する。ここでは p29 に接続する。

- LED の極性はテストで確認すること。
- LED に直列に接続する抵抗（制限抵抗）はここでは $1\text{k}\Omega$ としているが、電源電圧が変わる場合等もあるので計算できなければならない。抵抗値の計算方法は下記リンクなどに記載されているので自習する。大体 $1\text{mA}\sim 10\text{mA}$ 流れるようにする。

<http://www.rank-a.com/html/led.html>

<http://www.ops.dti.ne.jp/~ishijima/sei/letselec/letselec11.htm>

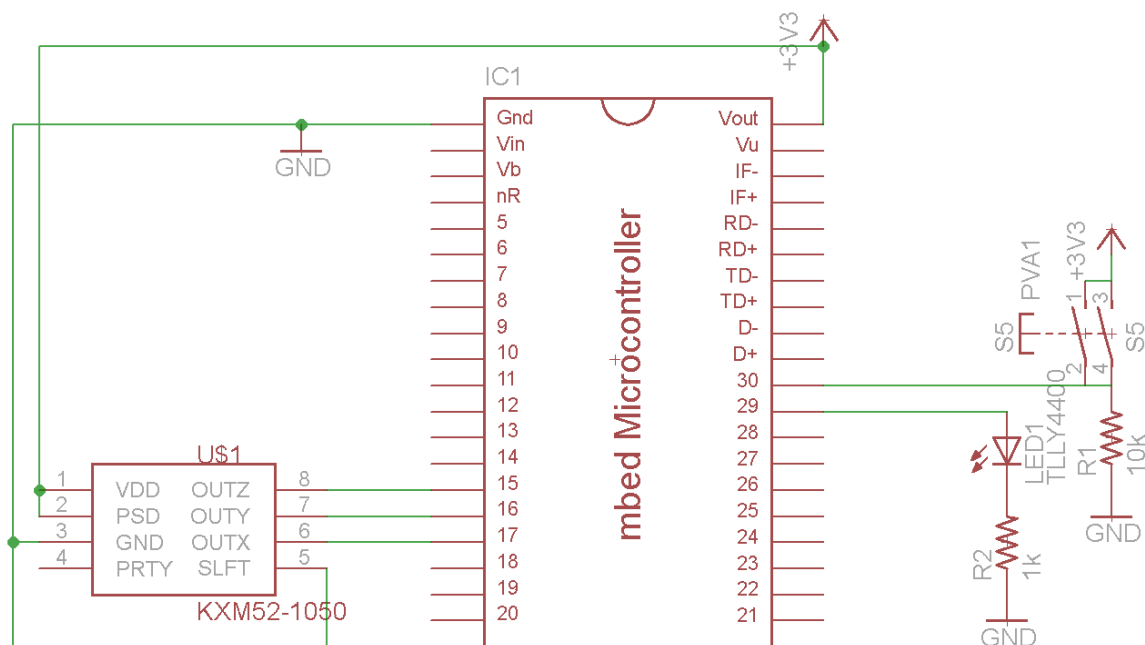
- 抵抗のカラーコードは読めるようになること。「電子工作の素」などを参照。

[課題3] タクトスイッチを押すと LED が光るようにする

[課題4] 割り込みプログラミングについて自習し、割り込みを用いたプログラムに改変する。

スイッチを押すたびに LED の状態が切り替わるようにする。

[課題5] LED を流れる電流を計測する（ \Rightarrow 抵抗間の電圧を計測する）。どれだけの電流が流れていることになるか計算する。その値は計算と比較して妥当か。



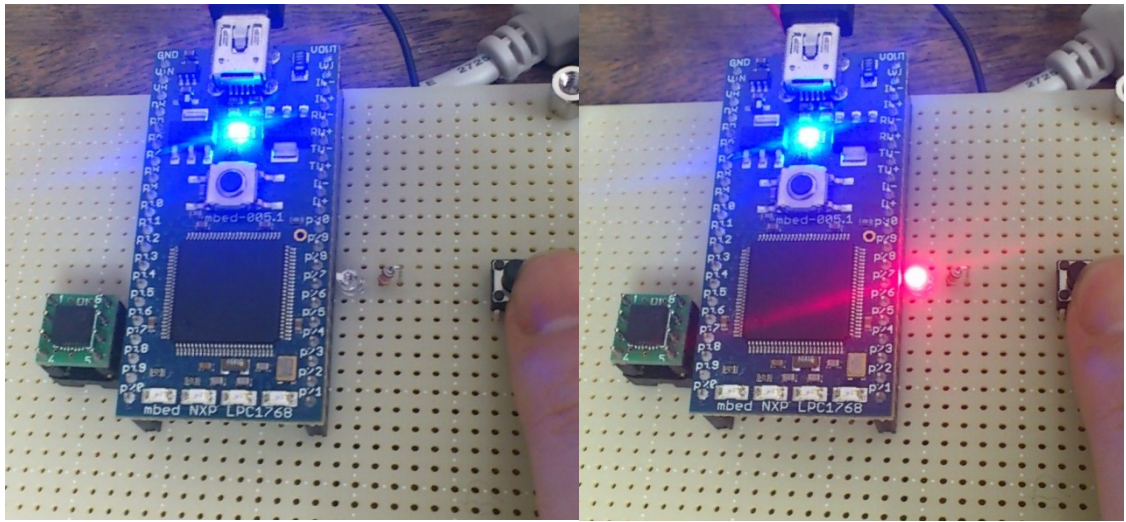


図 4 LED 回路 (タクトスイッチを押すと光る)

5 mbed で RC サーボモータ駆動 (PWM)

P21 から p26 まだが PWM 出力可能なピン。PWM 出力を利用して RC サーボモータを動かすことが出来る。ここでは RC サーボモータの信号線を P21 につなげる。

RC サーボモータの駆動には大電流が必要でこれまでの USB 給電では限界があるため外部電源 (5V) をつなげる。RC サーボの電源はここから取る。

(参考)

- RC サーボモータの制御方法について
http://berry.sakura.ne.jp/technics/servo_control_p1.html
- MiniStudio の小型 RC サーボ RB50 の配線
<http://www.ministudio.co.jp/Cgi-bin/Order-JP/DetailJp.asp?GoodsNum=202>
<http://www.ministudio.co.jp/Japanese/Goods-RB50-1.htm>

(サンプルソース)

- PWM 制御のサンプル
<http://mbed.org/handbook/PwmOut>
- RC サーボに特化したライブラリ。こちらを使っても良い。
<http://mbed.org/cookbook/Servo>

[課題6] PC からのシリアル通信でサーボモータの姿勢を制御する。例えば 0-9 の数値を送る。PC 側はシリアル通信ソフトで良い。オシロスコープで PWM 信号を観察する。パルス幅、パルス間隔は計算どおりとなっているか (オシロスコープで観察と書いてある場合はその画面をオシロの機能でキャプチャし、レポートに添付すること。以降も同様)

6 mbed でアナログ出力（+オペアンプ最初的一步）

mbed は 1ch, 10bit の DA および 6ch, 12bit の AD を持っている（ただし DA を使う場合は AD がひとつ減る）。すでに加速度センサを使う際にこの AD を 3ch 使用した。

18 番ピンから出力する。サンプルソースを用いて出力波形を確認する。

<http://mbed.org/handbook/AnalogOut>

オペアンプを使ってスピーカを駆動する。回路図（図 6）を参照のこと。次のことが行われている。

- ① ボリュームとして DA 信号を可変抵抗（10k Ω 程度）で分圧。
- ② オペアンプ（新日本無線製 NJM4580）の電圧フォロア（ボルテージフォロア）回路でインピーダンス変換。
- ③ コンデンサ（10 μ F 程度）によって直流成分をカット。
- ④ イヤフォンジャックからイヤフォンやスピーカに出力。

（注意）

- ・ 電解コンデンサの極性に注意。＋側をオペアンプの出力側とする（これはなぜか）
- ・ 可変抵抗の 3 つの端子の使い方はテストで確認する
- ・ イヤフォンジャックは基板に直付けできるタイプのものもあるがここではケーブルタイプを用いる。圧接端子の使い方の講習も兼ねる。

（参考）

オペアンプおよび電圧フォロア回路について：オペアンプについて全く知らない場合は全部読むこと。

http://picavr.uunyan.com/op_amp.html

[課題7] PC からのシリアル通信でスピーカの音を制御する。1-8 の数値を送ると「ドレミファソラシド」が鳴るようにする。PC 側はシリアル通信用ソフトで良い。オシロスコープで波形の変化を確認する。（正弦波を出力するには？）

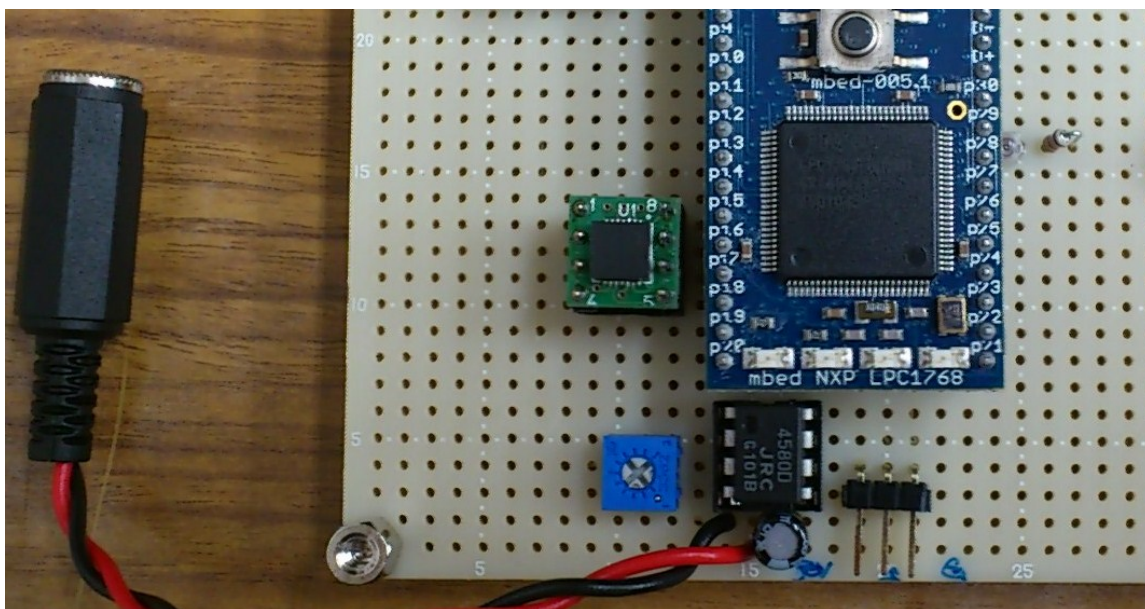
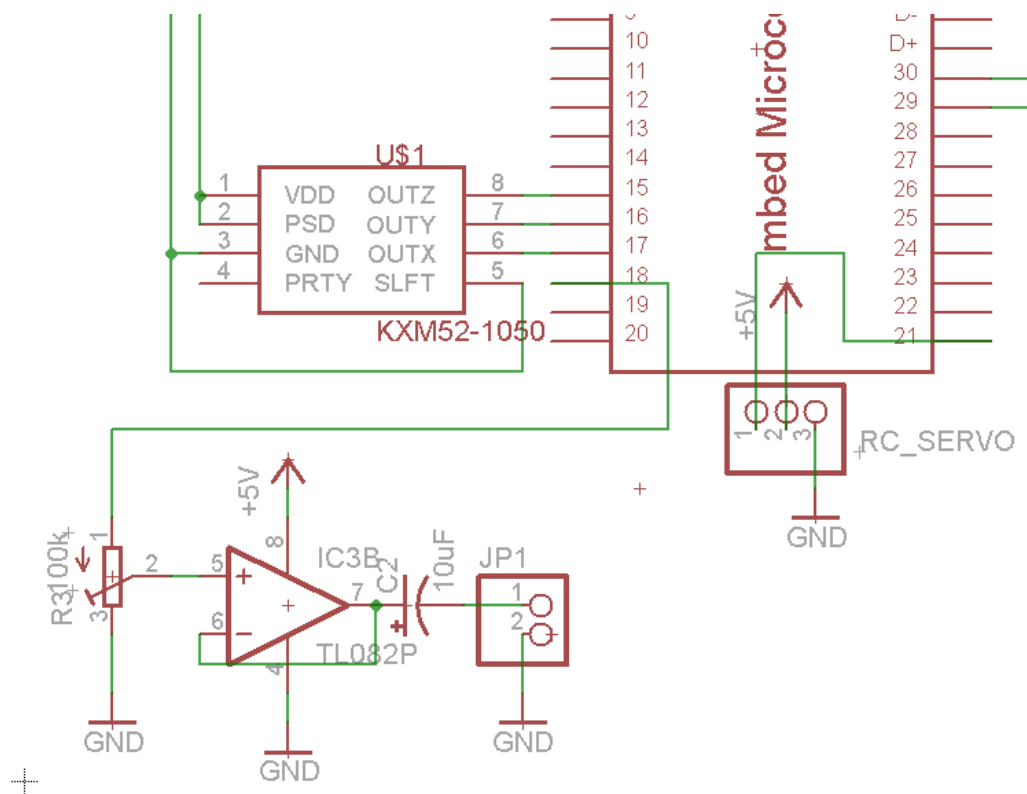


図 6 アナログ出力⇒オペアンプ回路（電圧フォロア） **TODO:圧接端子版に変更**

7 外付けの DA (SPI 通信によるポートの拡張)

mbed には SPI 通信モジュールが二つ用意されている。これを使うことでポートを拡張することが出来る。ここでは 8ch の D/A ポートを実現する。

SPI 通信、およびシフトレジスタについては自習すること。

外付けの DA 変換器としてオクタル 10bit D/A コンバータ LTC1660CN を用いる。特にこの課題はマニュアルを読む練習を兼ねている。

<http://akizukidenshi.com/catalog/g/gI-02794/>

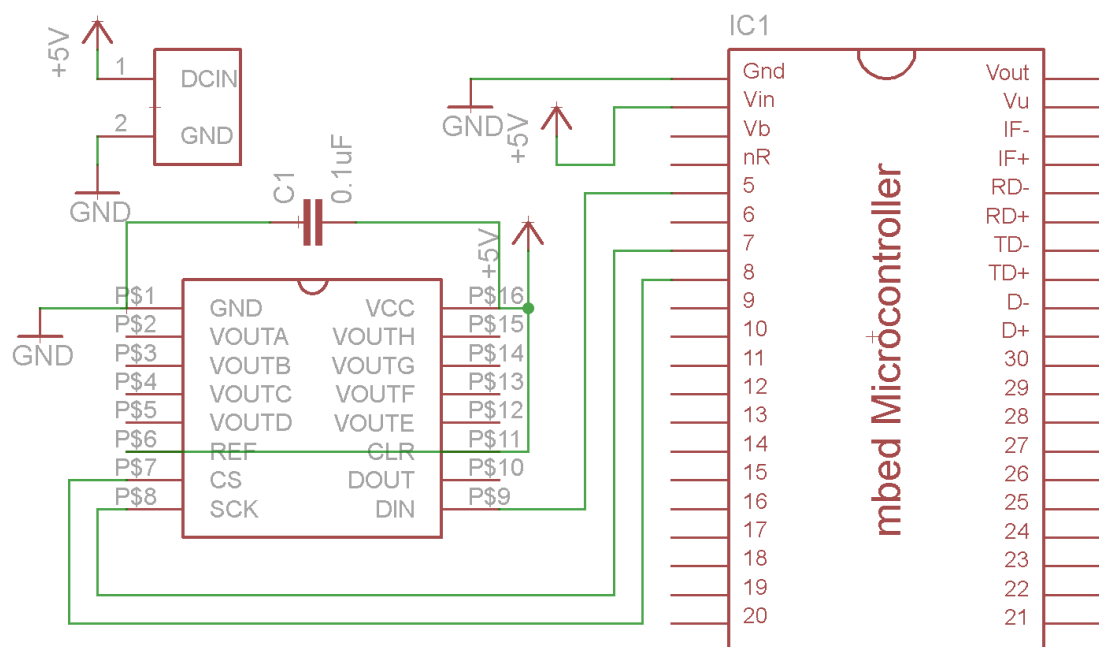
SPI 通信のサンプルソース

<http://mbed.org/handbook/SPI>

SPI 通信

http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

- DA 変換器の電源とグラウンドの間に電圧安定化のためのコンデンサを入れる。コンデンサの数値コードは読めるようになること。「電子工作の素」などを参照。



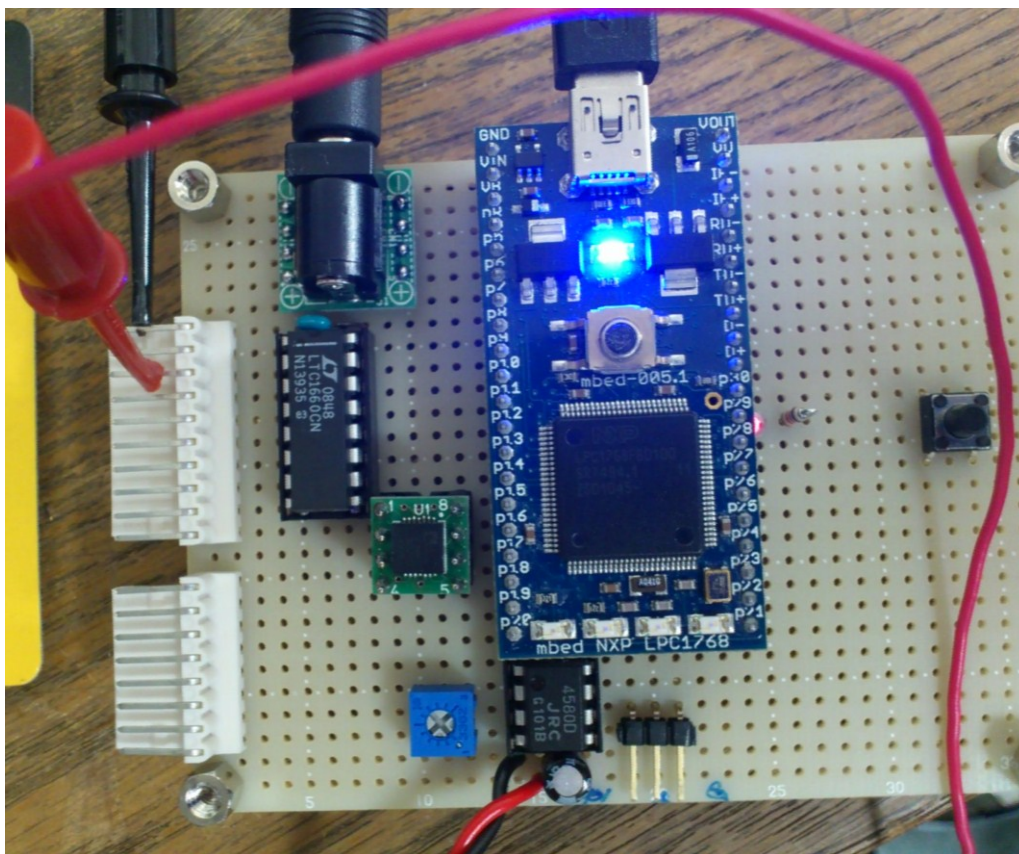


図 7 8ch DA ポート回路 (10pin の端子を使い、余った 2pin はグランドにしている。またこの写真では mbed の AD6ch を 8pin の端子で出力している。この場合は余った 2pin はグランドと 3.3V)

```

#include "mbed.h"

SPI spi(p5, NC, p7);      // mosi (出力 master output slave input) , miso (入力) , scl
                           (クロック)。今回は入力不要なので NC(not connected)とする
DigitalOut cs(p8);        //chip select ピン

Serial pc(USBTX, USBRX); // tx, rx

int main() {
    int t=0;
    short spiData,DA;
    char channel=0;
    // SPI のセットアップ。16bit、立ち上がりエッジ、1MHz clock
    spi.format(16,0);
    spi.frequency(1000000);

    while (1) {
        t++;
        //矩形波
        if (t%2==0) {
            DA=0x3FF;
        } else {
            DA=0;
        }
        cs = 0; //clock を有効化
        //データの整形。マニュアル参照のこと
        spiData = (((channel&0x07)+1)<<12) | ((DA&0x3FF)<<2);
        spi.write(spiData);
        cs = 1; //clock を無効化し、データをロード
        wait_ms(1);
    }
}

```

図 8 8ch D/A 変換器 LTC1660 を動かす最初のサンプルプログラム

[課題8] サンプルプログラムを動かし、オシロスコープを使って波形を観察する。つぎに

クロック信号とデータ信号を同時に観察する（CS ピンをトリガとする。**オシロスコープのトリガの使い方の練習も兼ねている**ので調べること）。

二つの信号のタイミングはどのような関係にあるか。それらは LTC1660 のデータシートのタイミングチャートと比較して妥当か。

16bit のデータ信号はプログラムとどのような関係にあるか。またそれらは LTC1660 のデータシートと比較して妥当か。

[課題9] 波形を適当な正弦波に変更、プログラム中のウェイトを無くしオシロスコープで観察する。データ更新一周期にどのくらいの時間がかかっているか。

[課題10] チャンネルを指定して電圧を出力できる関数を作成する。

```
void DAout(char channel, float voltage)
```

LTC1660 のデータシートによると、チャンネルを指定するにはどうすれば良いか。また電圧を 0.0 から 5.0 までの float で指定するにはどうすれば良いか。

その上で、8ch それぞれから指定した異なる周波数の正弦波を出力し、オシロスコープで波形を観察する。

以上により、**DA8ch(10bit)**、**AD6ch(12bit)**、**多数のデジタル入出力**を備えた標準的なインタフェースデバイスを手に入れたことになります。この後の選択課題ではさらに各種のインタフェース拡張を行います。

研究室の別の講習ではインタフェース社の PCI ボード (DA/AD/Enc) を扱います。mbed とこれらの PCI ボードは機能としては重なっていますが、研究開発の流れの中では別の役割を果たします。まず PCI ボードで素早いプロトタイピングと検証を繰り返し、最終的なデモでは mbed 化する、という流れになります（よって両方必要、と今は判断しています）。

8 選択課題

一人1課題を行う。すべての課題がある種のインタフェース拡張であり、最終的に PC との通信を行う。PC 側は **Dxlib** のプログラムを動かし、ユーザからの何らかの入力を反映して **mbed** 側も変化するようにする。

説明は完全では無い（回路図も未検証で間違っているかもしれない）ので、必要であれば部品のマニュアルを読む。

…その前に、「ファイル共有による通信」を学習しておきます。

すべての課題で、**DxLib** を用いていますが、実世界のセンシング情報と組み合わせるにはひとつ難しい点があります。**DxLib** のプログラムは通常、画面のリフレッシュレート (60Hz 等) でループを回しますが、**mbed** からのデータや **mbed** を介した制御は数百 Hz になることがあります。つまり、**必要なリアルタイム性のレベルが異なる**場合があるということです。**mbed** 側から送られるデータを遅くしても良いですが、大抵シリアルバッファにデータが「溜まり」、少し時間遅れを感じる、といったことが典型的に生じます（春休みの宿題や3年生実験で苦労したはず）。

このように異なる粒度の時間を同時に扱うには、一般にはひとつのプログラム中でスレッドを使えば出来ます。が、ここでは PC 側で二つのプログラムを動かし、共有ファイルを用いたプログラム間通信を行います（図 9）。研究室の環境では他の場合への応用（例えば PC 同士の通信や、他のソフトウェアとの連携）が楽なのでこの方法を使います。力がついたらスレッドによる方法や他の一般的なプログラム間通信も試してみてください。

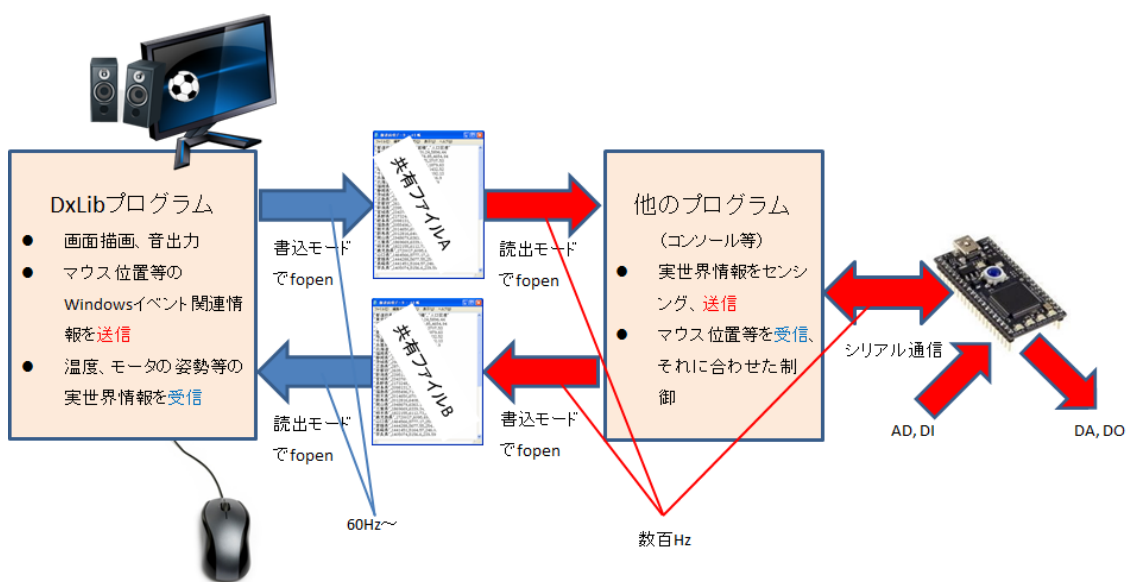


図 9 共有ファイルを用いた DxLib プログラムと他のプログラムの協調

[課題11] PC_COMMUNICATION のフォルダを開き、送信、受信の二つのプログラムを動かす。このサンプルは片方からもう片方に数値を送っている。動作を確かめた上、共有ファイルを増やし、反対側からもう片方にも別の数値を送るようにする（双方向の通信を実現する）。

fread、fwrite、fseek 関数を使っている。これらを使用している部分の動作を一行ずつ説明すること。なおこの方法で画像の送受信等も可能である。

[課題12] [課題 11] のソースを使い、二人一組で PC 間通信を行う（ファイル名を変更するだけ）。

[課題 11] のプログラムの片方を DxLib プログラムとし、もう片方をシリアル通信用のコンソールプログラムとすれば、研究室で扱うたいの課題に対応できます。

（参考）mbed のシリアル通信は USB を介しているため、規格上リフレッシュレートは 1kHz が上限です。リフレッシュレートが 1kHz でもオーディオ信号(40kHz 程度)が送れるのは、一回に大量のデータを送っているからで、データ転送量自体は最大約 1Mbps です。リアルタイム性という点ではあくまでも最大 1kHz であり、データの送受信だけで数 ms かかることに注意してください。

8.1 7 セグメント LED を動かす（初級～中級）

7 セグメント LED(シャープ製 GL9A040G)で数字を表示させる。これは数字を表示するための 8 つの LED の集合であり、アノードコモン、カソードコモンの 2 種類がある(図 10)。今回使用するのはアノードコモンであり、「吸い込み」電流によって点灯させることを前提としている。

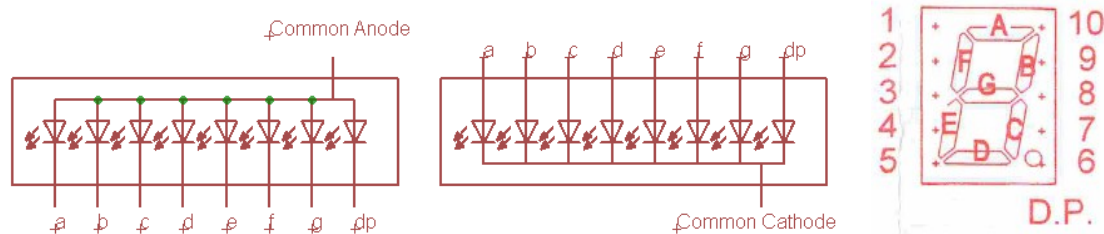


図 10 7セグメント LED の内部構造。(左) アノードコモン、(中) カソードコモン、(右) LED の配置

8 つの LED を光らせる必要があるために 8 本のピンが必要である。そこでポート拡張のた

めシフトレジスタ (今回は 74HC164) を用いる。SPI 通信を使ってデータを送る。74HC164 にはラッチが無い、データ移動の最中にも LED が点灯することに注意。ラッチを持つ IC もある (例えば 74HC595。またすでに扱った D/A 変換 IC も)。

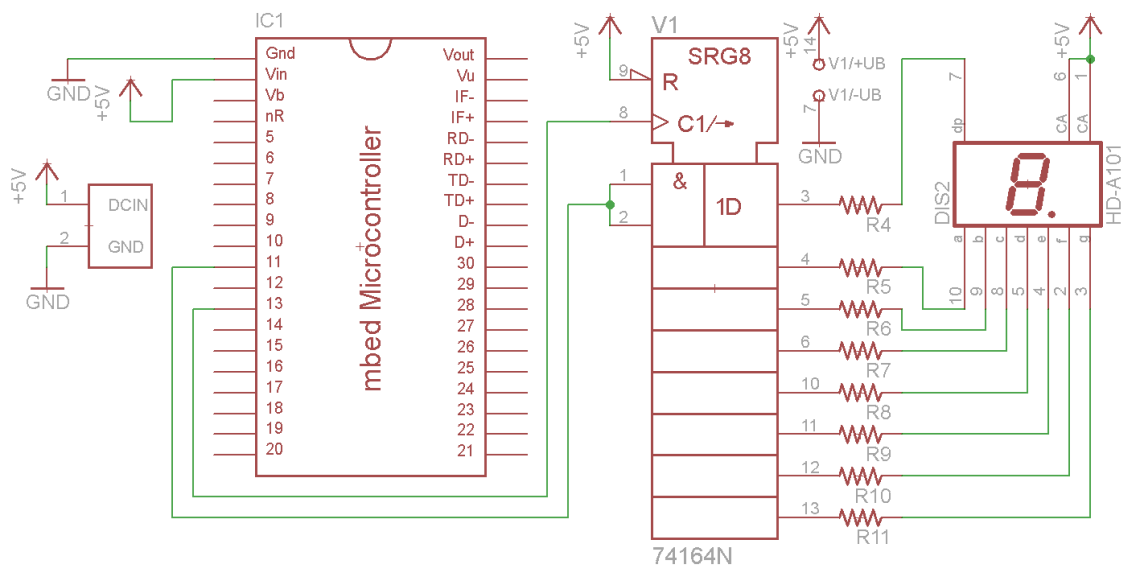


図 11 7セグメント LED 点灯回路

- 回路保護のため 74HC164 と 7セグメント LED は IC ソケットで半田付けすること。
- 74HC シリーズは入出力ピン一つあたり最大 4mA までしか電流を流すことができない。このため LED 用の抵抗 R2-9 はやや大きめのものを使う (計算すること)。

[課題13] PC から送られてきた 0~9 の数値を表示する。さらにシフトレジスタを二段連結して二つの 7セグメント LED を駆動し、PC から送られてきた文字を 16 進数で表示する。(シフトレジスタの連結は前段の最後の出力を次段の入力とすれば良い)。PC 上でも DxLib を用いて数値をグラフィカルに表示する。

8.2 振動提示型データグローブ (初級~中級)

軍手に 5つの曲げセンサ+5つの振動モータを取り付け、簡単なデータグローブとする。曲げセンサの値がある閾値を超えたら振動モータを駆動させることで、バーチャルな物体を触る状況を実現する。

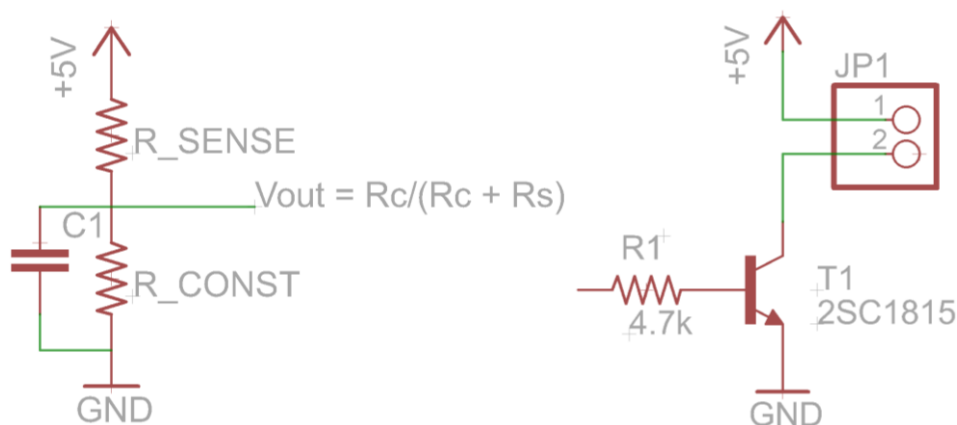


図 12 曲げセンサ回路（簡易版）と振動モータ回路

- 曲げセンサは曲がりが大きくなるほど抵抗値が下がる。この時図のように固定抵抗(R_c)との間で分圧回路を組むと、簡易的に曲げに応じた電圧出力が得られる。この出力をそのまま A/D 入力に入れば良い。固定抵抗の値は実験的に適したものをを用いるが、例えば $10k\Omega \sim 100k\Omega$ 程度である。またノイズを低減するために固定抵抗と並列に小さなコンデンサ（例えば $0.01\mu F \sim 0.1\mu F$ ）を入れる場合が多い。
- 振動モータはトランジスタによって駆動する。PWM 出力可能な P21～P26 から出力する。抵抗値は適当なので変える必要があるかもしれない。トランジスタ（ここでは NPN トランジスタ）について知らなければ自習のこと。

[課題14] 指の曲げに応じて振動の強さを変える。5 つの振動モータそれぞれが PWM 出力可能であるので、例えば曲げが大きくなるにつれて振動の強さが大きくなるという制御が可能である。以上は mbed で完結するが、PC 側 (DxLib) で何が出来るかも検討する。最低限、状態の表示は行う。

8.3 通信機能を持つセンサモジュールの使用（中級～上級）

これまでに扱ったセンサモジュールのほとんどはアナログ電圧出力をするものであるが、数多くのものが SPI 通信または I2C 通信機能を備えている。ここではその例としてボッシュ社の加速度センサ BMA180 を取り上げる。この加速度センサはこれまで使ってきたものよりもダイナミックレンジが広く、例えばバチを叩いた時の衝撃を計測することもできる。

http://www.switch-science.com/products/detail.php?product_id=386

このモジュールと mbed を接続する(BMA180 用ライブラリは非常にたくさん見つかる)。さらに mbed の DA 出力にオーディオ用パワーオペアンプ(LM675 等)を接続し、振動子 ForceReactor を駆動する。LM675 の回路はマニュアルを参照すること。

[課題15] ここでは HACHISick (あるいは Dayon) の模擬を行なう。加速度センサモジュールと振動子を棒に取り付け、ある閾値以上の衝撃が加わった時に振動子が駆動されるようにすることで、叩いた対象の材質感を表現する。ハウリングを防ぐにはどうしたらよいか。PC 画面にシンバル等の楽器を表示し、画面を実際に叩いた際に楽器も揺れ動くようにする。

8.4 超音波センサの使用（中級）

[課題 4] において、ピンの立ち上がりや立下りを検出する割込みプログラミングを行ったが、これを使う必要がある場面として超音波センサを取り上げる。ここでは浅草ギ研製超音波距離センサー(PING)))を用いる。このセンサは距離情報をピン電圧が上がっている時間によって表しているためピンの監視が必要であり、割込みプログラミングが適している。

<http://mbed.org/users/rosienej/code/Ping/>

[課題16] これを2つ使い、テルミンっぽいものを作る。音出力用アンプは LM675 などを用い、外部電源 12V で駆動すること。波形の切り替え（正弦波、矩形波、三角波など。テルミン自体についても調べる）は PC 側から行なう。他に PC ができることを考える。

8.5 リニアアクチュエータの駆動（中級～上級）

研究室標準のリニアアクチュエータである Firgelli 社製 PQ12-P を駆動する。

AD によるポテンショメータの読み取り、Hブリッジ回路による制御+PWMを行う。ポテンショ（可変抵抗）は分圧して A/D ポートで読み取る（図 13）。制御用 IC にはここでは TA8429HQ を使用する。

PQ12-P <http://www.firgelli.com/>

Hブリッジ回路について <http://www.picfun.com/motor03.html>

モータ駆動（Hブリッジ駆動）用ライブラリ <http://mbed.org/cookbook/Motor>

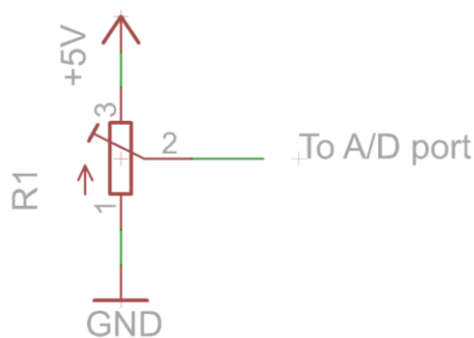


図 13 ポテンショメータ回路

駆動回路は TA8429 のマニュアルを参照すること (図 14)。ただし電源電圧は 24V ではなく 12V を使用する。1,2 番ピンに mbed からの入力を加えることで、正転、逆転、ブレーキ、ストップを実現する。

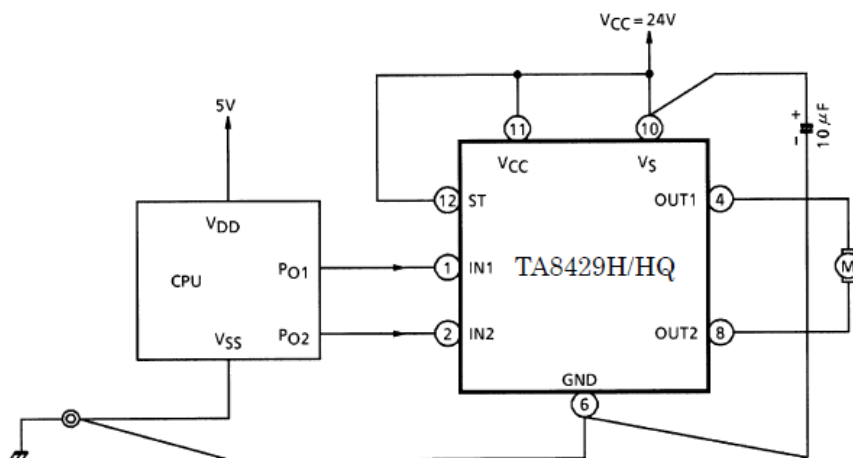


図 14 駆動回路(TA8429 マニュアルより)

[課題17] リニアモータの先端にフィルム状力センサを付け、加わる力も測定する。
指で押した時に、リニアモータの振る舞いによって「柔らかい」材質を再現する。またこの材質の変形の様子を何らかの形で DxLib で表示する。

8.6 DC モータの駆動 (中級～上級)

研究室標準の DC モータである MAXON 社製 DC モータを駆動する。これ自体はインタフェースボード講習会でも行うので、B4 はなるべくこの課題を選ばないこと。

Hブリッジ回路による制御+PWM

エンコーダライブラリを用いてパルスカウントを行う。

エンコーダ用ライブラリ QEI <http://mbed.org/cookbook/QEI>

モータ駆動については 8.5 節と同様。

以上により、**mbed** を研究室の研究で使ういろいろな場面に対応出来ました。この講習では省いた他の機能としては、インターネットへの接続、**USBAudio** 以外の **USB** デバイスなどがあります。例えばインターネットへの接続はデバイスの無線化に力を発揮します。ここまで出来たなら対応は容易でしょう。

9 （参考）mbed で PC との USB 通信

mbed オンボードの usb 端子ではなく、外付けの usb 端子を使って高速かつ大容量のデータのやり取り。

まず、プログラム書き込み用の USB 端子ではなく、外付け USB 端子を用意する。

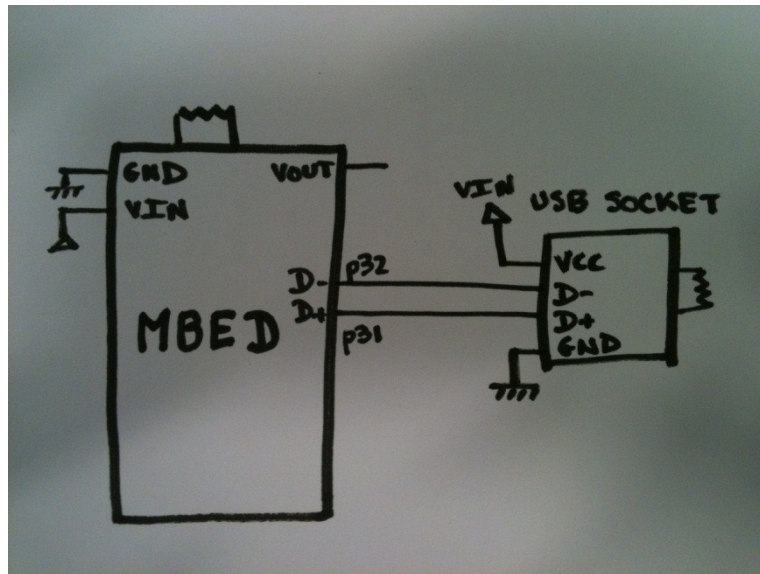


図 16 USB コネクタの外付け(<http://mbed.org/handbook/USBDevice> より)

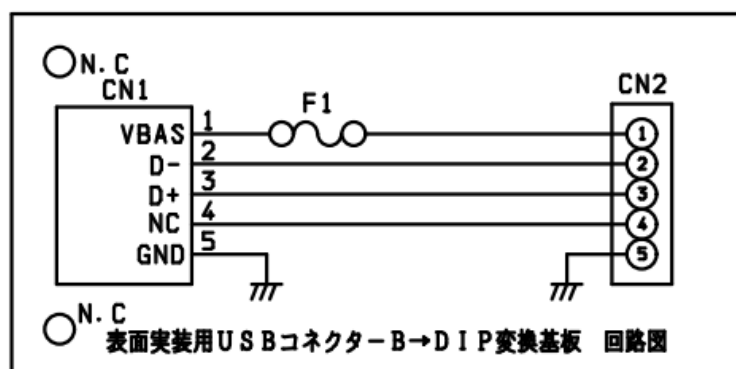


図 17 ブレッドボード用ミニBメスUSBコネクタDIP化キット（秋月）配線
(<http://akizukidenshi.com/catalog/g/gK-05258/> より)

9.1 USBAUDIO, USBKEYBOARD, USBMOUSE クラスを試してみる

多くの USB デバイスとして使うことが出来る。これによって USB-Serial のみではない、幅広い応用が可能となる。

例えば Keyboard のサンプルは下記

<http://mbed.org/handbook/USBKeyboard>

この中の、“Import This Program”をクリックしてプログラムを導入する。

元の USB ケーブルを外し、外付けの USB コネクタに接続すると、キーボードデバイスとして認識されるはずである。メモ帳などを開くと文字列が書き込まれることがわかる。

自分で作成したプログラム中に USB 機能をインポートするには、

<http://mbed.org/users/samux/libraries/USBDevice/m24owv>

の、“Import this library to a program”をクリックしてライブラリを導入する。

9.2 USBHID クラスを試してみる

もっともシンプルなサンプルをコンパイルしてみる

<http://mbed.org/users/samux/libraries/USBDevice/m24owv/docs/classUSBHID.html>

USBHID example.

```
#include "mbed.h"
#include "USBHID.h"

USBHID hid;
HID_REPORT recv;
BusOut leds(LED1,LED2,LED3,LED4);

int main(void) {
    while (1) {
        hid.read(&recv);
        leds = recv.data[0];
    }
}
```

図 18 USBHID のもっともシンプルなサンプル

(<http://mbed.org/users/samux/libraries/USBDevice/m24owv/docs/classUSBHID.html>)

USBHID クラスを使う

<http://mbed.org/handbook/USBHID>

<http://mbed.org/cookbook/USBHID-bindings->

<http://mbed.org/cookbook/USBHID-C-bindings>

ここではフリーの USBHID 用 API である hidapi を使う

<http://www.signal11.us/oss/hidapi/>

こちらのダウンロードページから、hidapi-0.6.0.zip をダウンロード、解凍する。(0.7.0 は Visual Studio で上手くコンパイルできない模様)

解凍後、windows⇒hidapi.sln からサンプルを開く

多チャンネルのデータ (例えば多チャンネル DA/AD 信号) のやり取りはこれが一番良いか。

現在のところ、データ通信に USB-HID クラスを用いるメリットは複数バイトの情報の送受信の際にデータの区切りがどこから始まるか等を気にしなくて良い点、程度である。USB-HID クラスには一回の送受信が最大 64byte という重要な縛りがあり、このため USB2.0 の速度を十分に生かしたデータ通信ができない (これは現在 mbed で実装されている他の USB-***クラスも同様である)

実際の所どうやら普通にシリアル通信+MODDMA、MODSERIAL による DMA 利用、によって、921.6kbps の通信が安定に行えるようであり、スピードとしても遜色ない (もしかするとより安定している)。ただし複数バイトを (データの切れ目を間違えること無く) 送受信し続けるための工夫が欲しい。MIDI 規格を流用するとか。独自規格だと例えば近藤科学の ICS コマンドというのがある (<http://kondo-robot.com/sys/archives/2628>)

//MODSERIAL from <http://mbed.org/cookbook/MODSERIAL>

//MODDMA <http://mbed.org/users/p3p/libraries/MODDMA/m3x0ky>

10 参考文献

文中で取り上げたもの以外で参考になる or 手元に置くと良いものを挙げます。大体研究室においてあります。一度めくってみて下さい。

[1] 「オペアンプ基礎回路再入門」

標準的な教科書。 オペアンプは一度ちゃんと勉強する必要があります。

[2] 「すぐ使える！オペアンプ回路図」

[3] トランジスタ技術 SPECIAL「OP アンプによる実用回路設計」

[4] トランジスタ技術 SPECIAL「OP アンプ IC 活用ノート」

[5] トランジスタ技術 SPECIAL「徹底図解 デジタル・オシロスコープ活用ノート」

CQ 出版の SPECIAL シリーズ。内容は標準的で、入門を卒業した人の次のステップ用で、長く座右に置けるタイプです。これ以外にも研究室に置いてあるトランジスタ技術 SPECIAL はどれも良い参考書。

[6] トランジスタ技術

研究室で毎月購読しています。だんだん読めるようになるでしょう。分かるところだけ拾い読みしましょう。

[7] PIC とセンサの電子工作

PIC を使った電子工作本ですが、センサ関係の解説が豊富でためになります。

[8] 日経エレクトロニクス、日経サイエンス

研究室で毎月購読しています。「社会人なら日経」ではないですが意外にみんな読んでいます。興味を持てるところだけ拾い読みしましょう。

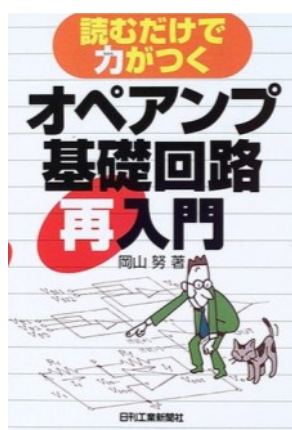


図 19 推薦図書