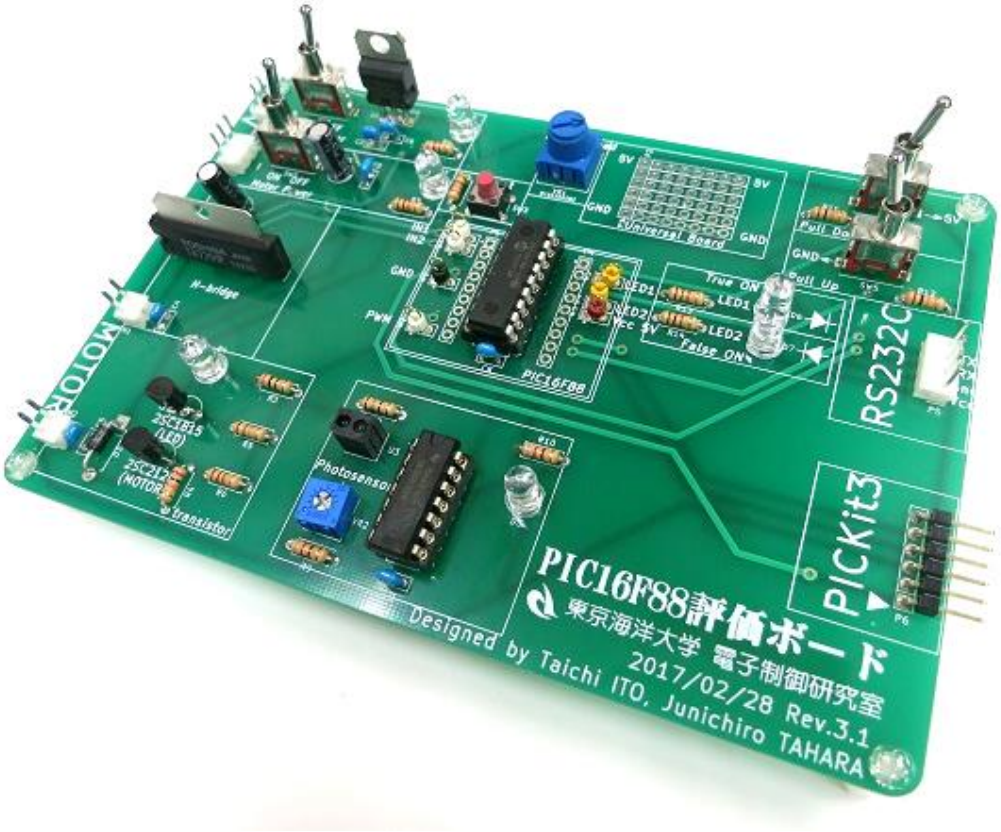
東京海洋大学 海洋工学部 海洋電子機械工学科

制御システム工学演習 -リファレンスボード編-



学籍番号 _____

名前 _____

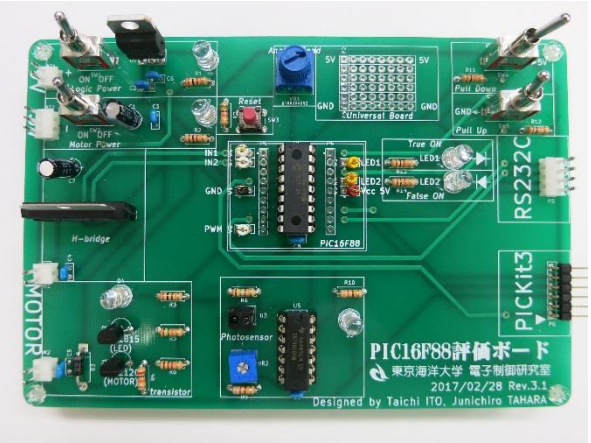
作成日 2017/10/28

最終更新日 2017/11/02

制御システム工学演習　-リファレンスボード編-

マイクロプロセッサ　PIC16F88 を中心とし，ライントレースロボットの基本となる．電源回路，センサー回路，モータドライブ回路等を加えたボードである．

本ボードは海洋電子機械工学科が制御システム工学演習の為に設計した．学習用リファレンスボードを用いる事により電子回路・マイコンのプログラミングを通じてライントレースロボット作成へスムーズに移行できるように設計され，資料も作成されている．



田原研究室の Web サイト上に，本ボードの使い方が記載されている．また，サンプルプログラムをダウンロードできる．下記 Web ページを参照のこと．

東京海洋大学 田原研究室 > 講義 > 制御システム工学演習

URL:　http://www2.kaiyodai.ac.jp/~jtahar0/posts/post.html

目次

1　電源回路の基礎

電気・電子回路において，電源は最も重要である．一定の電圧を出力し，マイクロプロセッサ等の電子機器に安定した電力を供給する．一般に電気・電子回路は GND を基準とし，これより高い電圧を+，低い時を-としている．よって回路には必ず GND が存在する．今回は電池-側を GND として回路を構成する．本リファレンスボードでは電源電圧を 5V とした．よって角形電池(9V)より電子回路の電源(5V)を作成しなくてはならない．

回路図 A2 または**図 1-1** を確認すると L7805 と書かれた IC がある．この IC を三端子レギュレータ**(図 1-2)**と呼ぶ．三端子レギュレータ(L7805)は入力電圧(9~7V)から安定した出力電圧である 5V を出力する．入力側の電源には高周波ノイズを除去するためのセラミックコンデンサー(0.33uF)を付ける．出力側には電流変化に対応するためコンデンサ(0.1uF)を付ける．なお，電解コンデンサーには極性がある事に注意すること．

また，入力側には電源スイッチがついている．今回は ON/OFF によく使われるトグルスイッチ**(図 1-3)**が使われており．真ん中の 2 番ピンを常時接続し，トグルにより 1,3 の接続先を変更している．

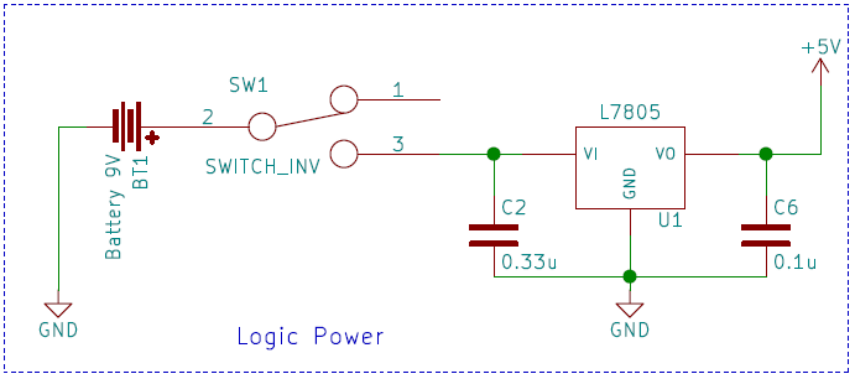


図 1-1　電源回路図

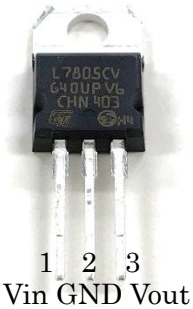


図 1-2　三端子レギュレータ

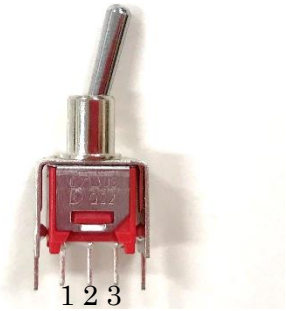


図 1-3　トグルスイッチ

次に，作成した電源に負荷を接続する．今回は負荷に LED**(図 1-4)**を用いる．LED には極性があり足が長い方がアノード(A)，短い方がカソード(K)と呼ばれる．アノード+側にカソード-を GND に接続する．このとき注意しなくてはいけない点は電源に対し負荷は並列に接続する必要がある．

また，LED を使うには守らなければいけない規定値がありデータシートに記載されている．
注意する点は最大順方向電圧Vf_{max}，最大順方向電流If_{max},最大電力P_{max},順方向電圧Vf,順方向電流Ifを用いる．このとき電流制限用の抵抗を付けないと LED に過電流が流れ熱で破壊される．
ここで LED(D1)(型番:OSW5DK5111A)は Vf=3.1V，If=20mA，電源 5V の時の電流制限抵抗 R1 は？

$5 < Vf + If * R$

$5 < 3.1 + 20 * 10^{-3} * R$

$R > 95[\Omega]$

となり，100Ω程度を選択すればよい事になる．この場合 LED はかなり明るく光るためこれより大きな 2kΩ程度を使っている．**(図 1-5)**

図 1-4 LED

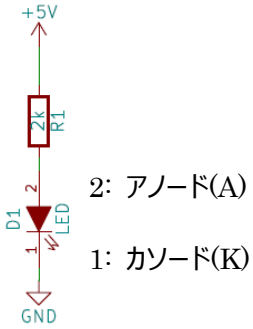


図 1-5 LED 点灯回路図例

問題

Q1　三端子レギュレータとは何か調査せよ．

Q2　5V から 3.3V を出力する回路を設計せよ．（ヒント：3.3V 出力のレギュレータを使う）

Q3　-9V 入力から-5V を出力する回路を設計せよ．　（ヒント：負電源用-5V 出力のレギュレータを使う）

Q4　リファレンスボードの回路図の電源部分を赤点線で囲みなさい．またこのときの部品リストを作成しなさい．

2 マイクロプロセッサの基礎とテスト回路

本実験でつかうマイクロプロセッサは PIC16F88**(図 2-1)**を用いる．Microchip 社の産業用 8bit マイコンである．各社が多様なマイクロプロセッサを出荷しているが主な物を上げておく．（AVR, ARM, MSP430）

PC 用の CPU と異なり，計算能力よりも外部入出力・省電力・回路規模の小型化に中心をおいて設計されている．

ここでは，PIC16F88 の基本的な回路とピンの配置**(図 2-2)**を説明する．先ずは最低限必要回路な回路を構成する．今回は 1)~5)の回路を接続しマイコン用のテスト回路の回路図と実態配線図の作成を行う．

図 2-1 PIC16F88

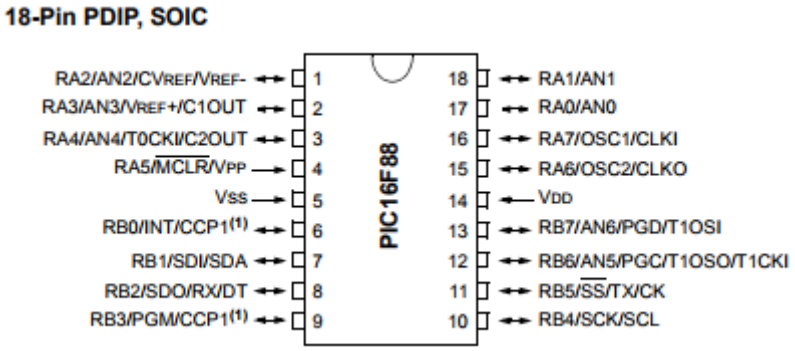


図 2-2 PIC16F88 ピンダイアグラム（データシートより引用）

1)電源回路(Vss，Vdd) 回路図 A1-A2

１．電源回路の基礎 で説明した 5V の電源回路を接続する．産業用マイコンは電源の動作範囲が 3~5.5V など比較的緩やかである．しかし基準電圧も決められているのでこれに従うこと．PIC16F88 の場合は 5V である．PIC16F88 の電源 Vdd(14 番ピン)に 5V，Vss(5 番ピン)に GND を接続する．またノイズに対して堅牢にするため，PIC の Vdd, Vss 間には 0.1uF のセラミックコンデンサ(回路図 B4, 部品番号 C8)を接続する．

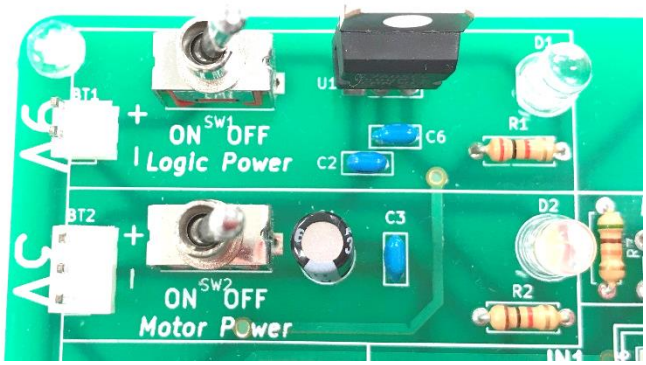


図 2-3 電源回路部分

2)リセット回路(MCLR) 回路図 B3

リセット回路は必ず必要である．MCLR(4 番ピン)は常時 5V にプルアップ(3 章で説明)する必要がある．今回はプッシュスイッチを押したときに 0V(GND に接続)，OFF の時に 5V となる様に接続されている．

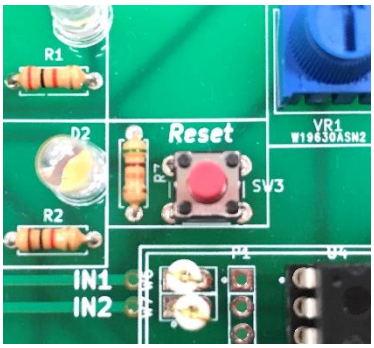


図 2-4 リセット回路部分

3)ISP プログラムライター接続用端子 回路図 C6

PC からマイコンへプログラムを書き込む為の端子である．PGC(クロック)(12 番ピン)と PGD(データ)(13 番ピン)の 2 つの PIN を接続する．さらにこれに Vdd と GND，MCLR の 5 ピンを外部に引き出して，PIC のプログラマ PICkit3 に接続する**(図 2-6)**．今回は 6 ピンのピンヘッダーを使っている**(図 2-5)**．

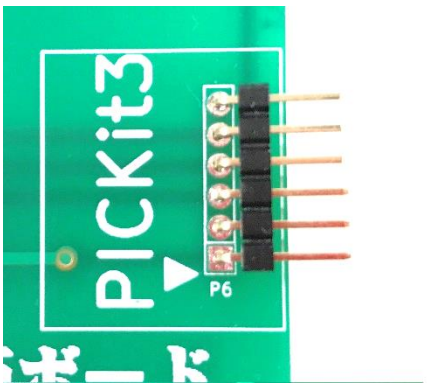


図 2-5 接続用端子部分

図 2-6 PICkit3 接続時

4)確認用 LED 回路図 A5-B5

今回は動作確認用の LED を 17 番ピンに電流制限抵抗を付け GND に接続する．必ず電流制限抵抗を付けること．このとき接続された 17 番ピンに 5V の電圧が生じたとき(Hi)，LED は ON となる．逆に 17 番ピンが 0V の時（Low)となる．

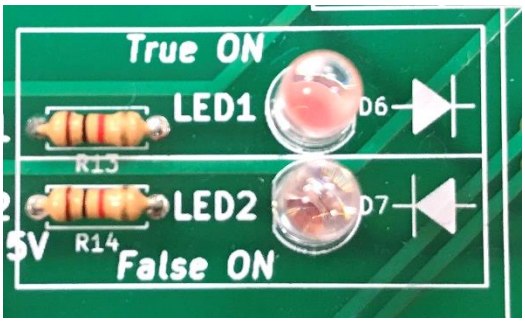


図 2-7 確認用 LED 部分

5)動作確認用 SW 回路図 A5

今回は動作確認用のトグルスイッチをプルダウンで接続する(3 章で説明)で接続する．プルダウン抵抗がないと誤入力が発生するため必ず接続する．

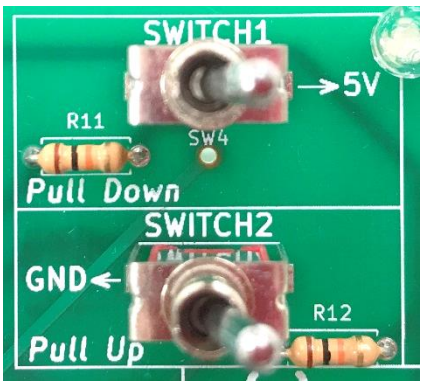


図 2-8 確認用スイッチ部分

問題

Q5　リファレンスボードの回路図の 5V 電源から PIC の Vss へ向かっている電源ラインを赤色で引け．また GND は黒で引くこと．

Q6　リセット回路を破線で囲み，PIC の MCLR と抵抗・スイッチを確認せよ

Q7　PIC に接続した LED 回路を破線で囲み，接続先ピン番号と電流制限抵抗を確認せよ．

Q8　PIC に接続した SW 回路を破線で囲み，接続先ピン番号とプルダウン抵抗を確認せよ．

Q9 テスト回路だけの回路図を作成せよ

Q10　テスト回路の部品表を作成せよ.

回路図から実態配線図の作成

先ほど，Q10 で作成した回路図から，実際配線図を作成してみる.

テストボードは 1)~5)を全て接続した物となる.　実際の回路を作成するにはいろいろな手法があるが今回はグラフ用紙を用いて書いていく手法を取る.

3 マイコンと外部回路の接続

マイコン回路と LED やスイッチをつなぐためにはプルアップとプルダウンと呼ばれる手法を使う．これはマイコンの PIN の出力・入力の電流の方向を決めてやればよい．
ただし，マイコンの PIN は 10mA 程度しか流す事が出来ないため，必ず電流制限抵抗を付ける事．さもないとマイコンが壊れてしまう．また，小電流のためマイコンのピンに直接モータを繋ぎ動かすことはできない．

1)HI レベルドライブ

図 3-1 に示す様に，PIN に LED やトランジスタを接続する手法である．LED のアノードを PIN 側に，カソードを GND 側に接続するのがポイントである．このとき PIN を HI にすれば電流が GND 側に流れる．つまり，LED のアノードからカソードに電流が流れ LED は点灯する．（正論理）．テストボードの回路 True ON はこれになる．

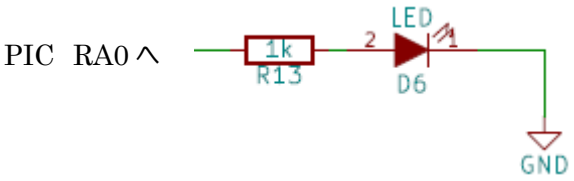


図 3-1 HI レベルドライブ回路図

2)LOW レベルドライブ

今度は逆に PIN を LOW にしたときに LED が ON になる回路である．PIN を LOW にしたとき電源から PIN 側に電流が流れる（負論理）．テストボードの回路 False ON はこれになる．LED のアノードを 5V 電源側に，カソードを PIN 側に接続する

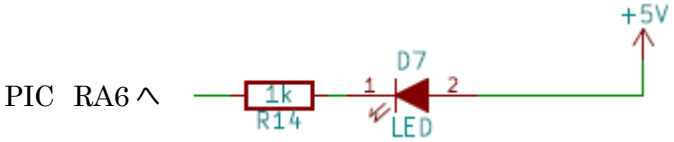


図 3-2 LOW レベルドライブ回路図

3)プルアップとプルダウン

プルダウンとプルアップは SW 回路に良く用いられる．抵抗のつなぎ方で論理が異なるので注意すること．

プルダウン回路は図 3-3 の様に SW が押された時はプルダウン抵抗に 5V から電流が流れ，出力は 5V となり，マイコンには 1 が入力される．SW が押されないときはプルダウン抵抗を介して GND に接続されているため出力は 0V となりマイコンには 0 が入力される．

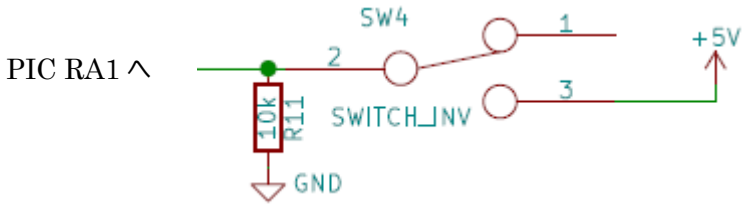


図 3-3 プルダウンスイッチ回路図

プルアップ回路は図 3-4 の様に SW が押されていないときはプルアップ抵抗に 5V から電流が流れ出力は 5V となりマイコンには 1 が入力される． SW が押された時は出力は GND と直結されるため 0V となりマイコンには 0 が入力される．

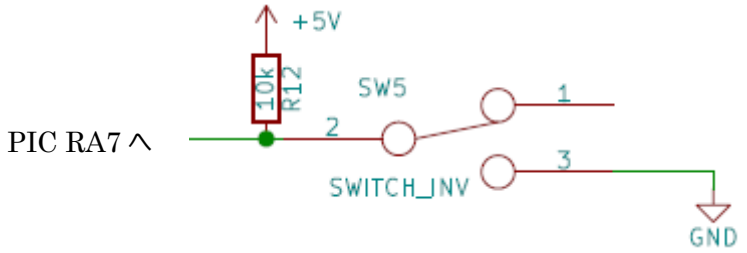


図 3-4 プルアップスイッチ回路図

問題

Q11 リファレンスボードの回路図より，HI レベルドライブと LOW レベルドライブ回路を確認せよ

Q12 リファレンスボードの回路図より，プルアップ，プルダウン回路を確認せよ

4 LED 点滅プログラムを作る

サンプルプログラム: 16F88_Ref3_Blink.X	
電池	9V のみ
RS232C	不要
モータ	不要
動作説明	LED1 と LED2 が 500ms おきに交互に点灯する

まず，手始めに LED を点灯するプログラムを作成して見よう.
PIC 用プログラムの作成には 2 つの方法がある.
1. PIC の統合開発環境 MPLAB X IDE(無償)をインストールして使う方法
2. クラウドベースの PIC 統合開発環境 MPLAB Xpress(無償)をインターネットブラウザで開き、ソフトをインストールせずに使う方法
どちらの方法も，田原研究室の Web サイト上に使い方が大変わかりやすく記載されている． 下記 URL を参照のこと.
東京海洋大学 田原研究室 > 講義 > 制御システム工学演習
URL: http://www2.kaiyodai.ac.jp/~jtahar0/posts/post.html

PIC16F88 評価ボードのピン配置は図 4-1 のようになっている． 今回は，15,16,17,18 番ピンに接続された LED と SWITCH を動作させて見る.
LED1 は HI レベルドライブで 17 番ピンに，LED2 は LOW レベルドライブで 15 番ピンに接続されている.
SW との接続はプルアップ，プルダウンを用いる． プルダウンで接続されたのが SWITCH1，プルアップで接続されたのが SWITCH2 である.

設定	#define	接続先		PIC16F88			接続先	#define	設定
出力	M_IN1	H-Bridge: IN1	1	RA2	RA1	18	SWITCH1	SWITCH1	入力
出力	M_IN2	H-Bridge: IN2	2	RA3	RA0	17	True ON LED	LED1	出力
入力	VOLUME	VR1	3	RA4	RA7	16	SWITCH2	SWITCH2	入力
		Reset	4	RA5 (MCLR)	RA6	15	False ON LED	LED2	出力
		GND	5	Vss	Vdd	14	5V		
		Not Connect	6	RB0	RB7 (PGD)	13	PGD		
		Not Connect	7	RB1	RB6 (PGC)	12	PGC		
入力		RX	8	RB2 (RX)	RB5 (TX)	11	TX		出力
出力	TR	Transistor	9	RB3	RB4	10	Photosensor	SENSOR	入力

図 4-1 PIC16F88 評価ボードにおけるピン配置

図から分かるように PIC マイコンには各ピンに機能が割り当てられており始めに使う機能によって設定を変

えなくてはならない.
List1 の 12~26 行目の CONFIG1 と CONFIG2 は PIC16F88 を使うための設定となっており変更してはならない.
28 行目はメーカーが提供している初期化のヘッダーファイルをインクルードしている．３０行目が重要で今回は 8MHz の周波数をマイコンの動作基準クロックとしている．この設定により時間の設定がキチンと行えるため，500ms の待ち時間(55 行目)を実行可能となる.
32~40 行はリファレンスボードのピンの設定である．LED1 は RA0 ピンに接続されている事をしめしている．無くても良いが設定しておくとプログラムが分かり安くなる.
さて，42~60 行がメインのプログラムとなる.
44 行目は内蔵クロックの速度の設定である．今回はこの数値とする.
46~47 はポート A,B を初期化する.
48~49 行はポート A，B の PIN の設定である．TRIS レジスタはピンの機能を設定である.
ピンのビットを出力:0，入力:1 として設定する.
TRISA の場合は RA1,RA7 をデジタル入力，RA4 をアナログ入力としているので２進数で 0b10010010 と設定する(表 4-1)．１６進数になれていれば 0x92 と設定してもよい.
例えば全てを入力にしたいときは 0xFF で，全てを出力にしたいときは 0x00 となる.
同様に TRISB も設定する(表 4-2).

表 4-1 TRISA の設定

ピン名	RA7	RA6	RA5	RA4	RA3	RA2	RA1	RA0
接続先	SW	LED	Reset	VR	H_Brd.	H_Brd.	SW	LED
入出力	入力	出力	-	入力	出力	出力	入力	出力
TRISA	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**0**

表 4-2 TRISB の設定

ピン名	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
接続先	PGD	PGC	TX	Sensor	Tr.	RX	N.C.	N.C.
入出力	-	-	出力	入力	出力	入力	-	-
TRISB	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**

50 行目は AD コンバータ(9 章)の設定レジスタであるが今回は AD コンバータを使用しないので 0 を設定している.
52~59 行目には繰り返すプログラムを記載する while(1) { 繰り返し処理 } や
for(;;) { 繰り返し処理 }が入る.
まず，HI レベルドライブ(正論理)の LED1 を考える．53 行目の LED1=1; で RA1 が ON となり，接続された LED が点灯する.
次に，__delay_ms(500);で 500ms 点灯時間させる．次に LED1=0;で OFF となる．同様に

500ms 待つ．これにより 1Hz の点灯パターンとなる．
これとは逆の low レベルドライブ(負論理)の LED2 は LED1 とは逆の消灯=>点灯=>消灯のパターンで動作する．

作成したプログラムを，PIC に書き込んでみよう
田原研究室の Web サイト上に，書き込み方法が大変わかりやすく記載されている．下記 URL を参照のこと．
MPLAB X IDE の使い方
http://www2.kaiyodai.ac.jp/~jtahar0/posts/activity19.html

問題

Q13 プログラムを改造して，LED1,LED2 を同時に点灯，消灯するプログラムを作成せよ

Q14 点灯時間を 2Hz とするプログラムを作成せよ．

5 RS232C によるデバッグ

サンプルプログラム: 16F88_Ref3_RS232C.X	
電池	9V のみ
RS232C	必要
モータ	不要
動作説明	電源を入れると PC に対して「Hello World!」と送信．その後は PC から受信した文字を送信(Echo)する．

次に，RS232C によるデバッグについて説明する．RS232C は PC と産業用機器(計測器・シーケンサ)の標準接続機器となっている．RX，TX，GND の３つのケーブルを使う事で通信を行う．通信条件は 9600bps(bit/sec)，8bit，ノンパリティとする．

ハードウェアの概略図を**図 5-1** に示す．注意しなくてはならないのは，PC 側の RS232C のレベルは ±12V である．マイコンボードは 5V なので直接つなぐことは出来ないのでレベル変換 IC**(図 5-3)**（MAX232, ADM3202 が有名）を使う．今回は Dsub9pin**(図 5-2)**の中に組み込んである．USB-シリアル変換には

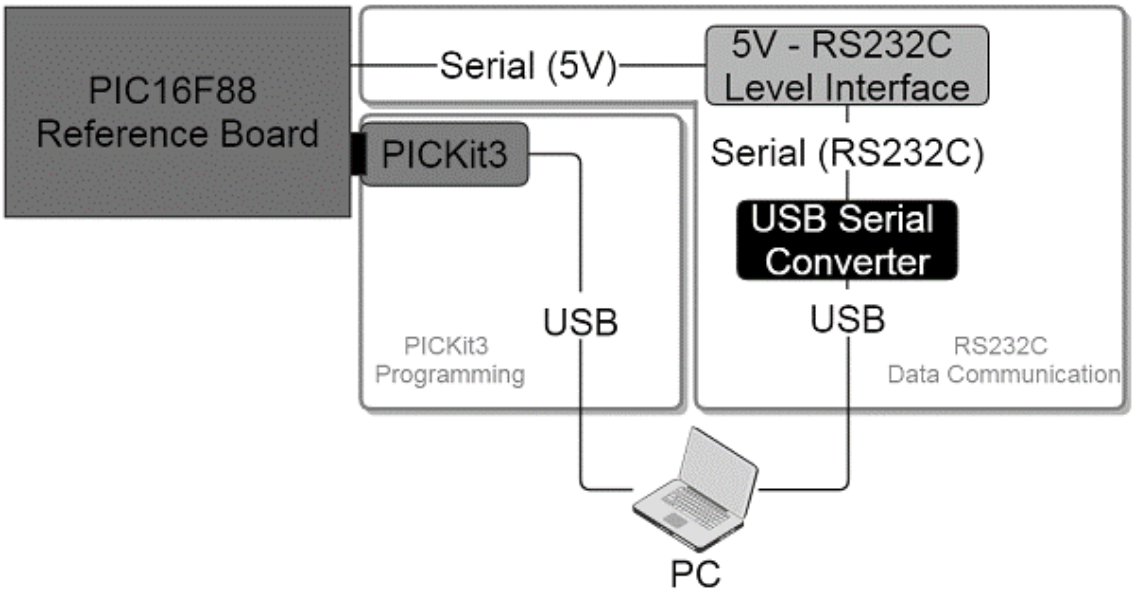


図 5-1 ハードウェア概略図

図 5-2 Dsub9pin コネクタ付きケーブル

図 5-3 RS232C レベル変換基板

また，PIC16F88 の 7 番ピンが RX，11 番ピンが TX と決まっているので配線するときには注意が必要である.

さてプログラムで使う場合は 30 行目の uart.h を include して使う．これは PIC16F88 で RS232C を使うための設定がここに書いてある．なお，uart.h と uart.c は必ず，main.c と同じフォルダーにおくこと.
また，mian.c の 52 行目の TRISB は RB2 が入力となるので 3bit 目が 1 となる事に注意する.
55 行目はシリアルの初期化を行い 9600bps に設定する．59~62 行目で rs232c に文字が送られる.
printf は通常の c 言語と同様である．65 行目で一文字を受信し 66 行目でそのまま印字している.
なお，今回使用している XC8 コンパイラの stdio ライブラリは，浮動小数型の printf 関数印字(例:printf("%f",float);)には対応していない．また，scanf 関数にも対応していない．文字を受信したいときはサンプルプログラムのように getch 関数を使って 1 文字ずつ読み込む.
プログラムを PIC16F88 に書き込み，PC で TeraTerm を起動する．キーボードより文字を入力するとその文字が PC の TeraTerm の画面に出力される.

TeraTerm の使い方

PIC16F88 評価ボードとシリアル通信(RS232C)で文字列を受信・送信するには、Windows の場合、ターミナルエミュレータの TeraTerm(無償)を使う。TeraTerm を使うときの手順を説明する.

1) PIC16F88 評価ボードと PC を接続する

図 5-4 RS232C レベル変換モジュール接続時

基板の「RS232C」と印字された部分のコネクタに、RS232C レベル変換モジュールを取り付ける。モジュールの反対側(USB 端子)は PC に接続する。

2) PIC16F88 評価ボードの接続されている PC の COM ポート番号を確認する

スタート > コントロールパネル > ハードウェアとサウンド > デバイスマネージャー を開く (図 5-5).

ポート をクリックし，USB Serial Port が COM の何番に割り当てられているか確認する(図 5-6).

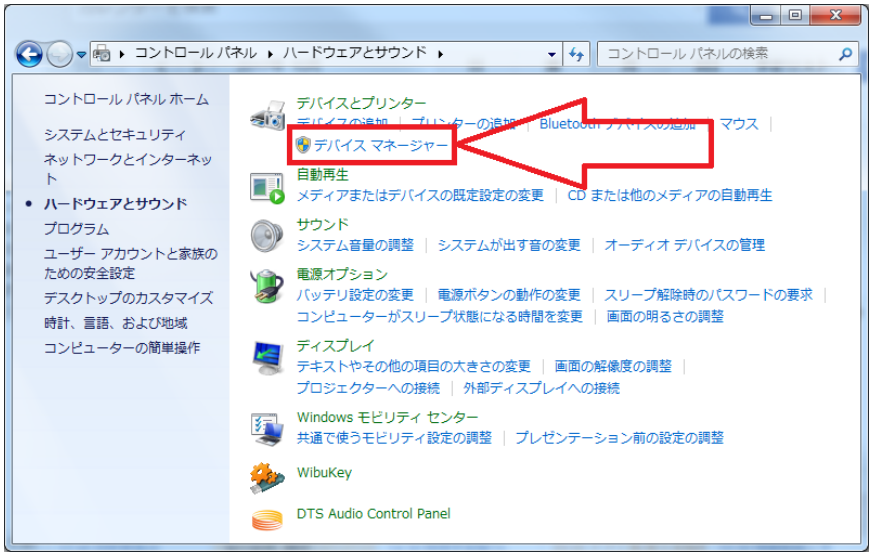


図 5-5 コントロールパネル画面

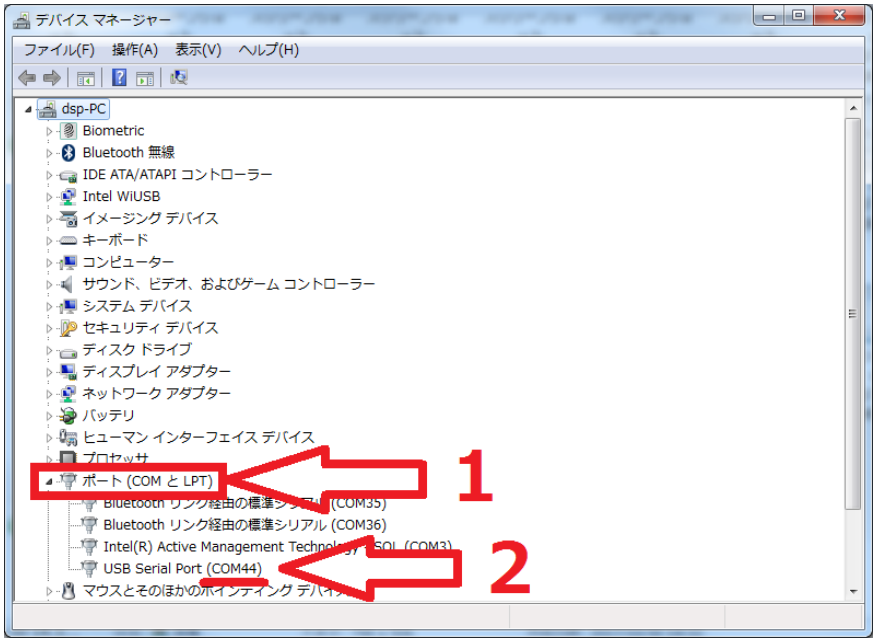


図 5-6 デバイスマネージャー画面

3) TeraTerm を起動する

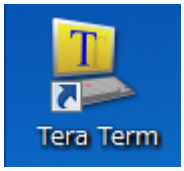


図 5-7 TeraTerm アイコン

4) シリアルポートに接続する

先ほど確認した PIC16F88 評価ボードの接続されている COM ポート番号を選択して OK をクリックする.

図 5-8 TeraTerm 新しい接続画面

5) 端末の設定

設定 > 端末 の順にクリックし，端末の設定画面を開く.

図 5-9 のように改行コード設定を「受信: LF」「送信: CR+LF」に変更し，「ローカルエコー」にチェックを入れる．最後に OK をクリックする.

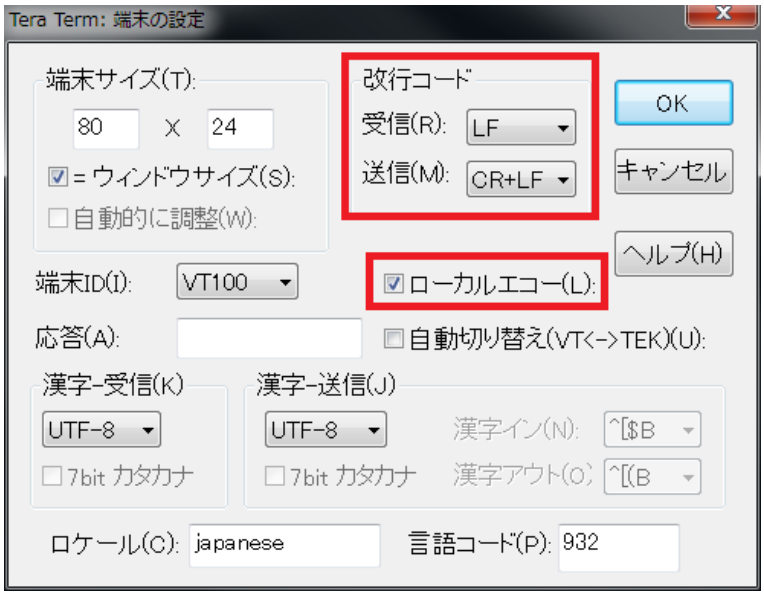


図 5-9 TeraTerm 端末の設定画面

6) シリアルポート設定

設定 > シリアルポート の順にクリックし，シリアルポート設定画面を開く.

図 5-10 のように，「ボー・レート: 9600」「データ: 8bit」「パリティ: none」「ストップ: 1bit」「フロー制御: none」に設定されているか確認する．最後に OK をクリックする．

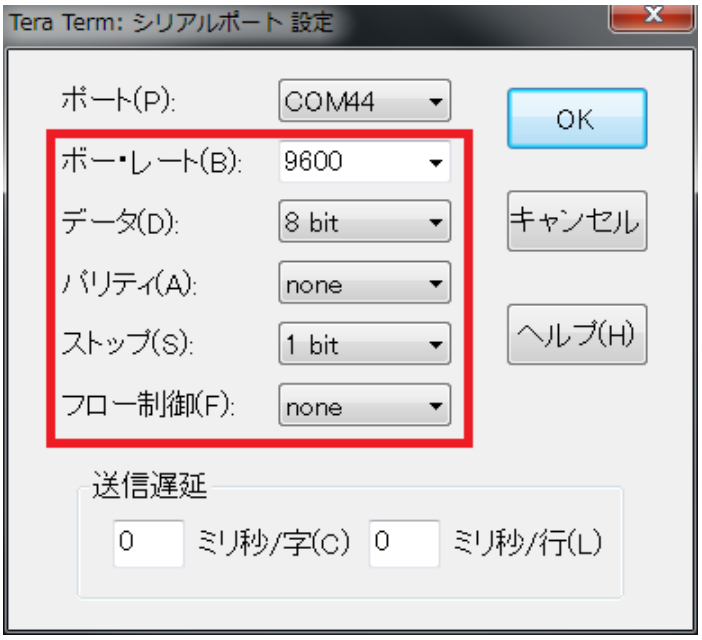


図 5-10 TeraTerm シリアルポート設定

以上で接続は完了．TeraTerm の画面に入力した文字は PIC16F88 評価ボードに送られ，PIC16F88 評価ボードから送られた文字は TeraTerm の画面に表示される．

問題

Q15 送信する文字列を，Hello World から自分の名前に変えてみよ．

Q16 getch 関数を改造して，文字列を読み込む関数を作れ．なお改行は¥n で行え．
難易度が高い問題．ヒント: サンプルプログラム 16F88_Ref3_Special.X の中に同様な処理を行なっている部分があるので見ておくこと．

6 SW の入出力

サンプルプログラム: 16F88_Ref3_Switch.X	
電池	9V のみ
RS232C	必要
モータ	不要
動作説明	Pull Down と Pull Up のトグルスイッチからの入力がそれぞれ HIGH か LOW か、PC に送信する.

スイッチを入力する場合は TRISA,TRISB を設定する必要がある．出力したい PIN のビットを 1 とすればそのピンは入力モードとなる．（4 章 表 4-1,表 4-2 参照）
本ボードでは，RA1,7 はスイッチなので入力モードとなる．また 3 章の Hi, Low ドライブの論理，プルアッププルダウンの論理を表にまとめる.

表 6-1 HI レベルドライブ(True ON LED)

RA0	ピン出力	LED
0	0V	ON
1	5V	OFF

表 6-2 LOW レベルドライブ(False ON LED)

RA6	ピン出力	LED
0	0V	OFF
1	5V	ON

表 6-3 プルアップ(Pull Up SWITCH)

SWITCH	ピン入力	RA7
2-1 接続	5V	1
2-3 接続	0V	0

表 6-4 プルダウン(Pull Down SWITCH)

SWITCH	ピン入力	RA1
2-1 接続	0V	0
2-3 接続	5V	1

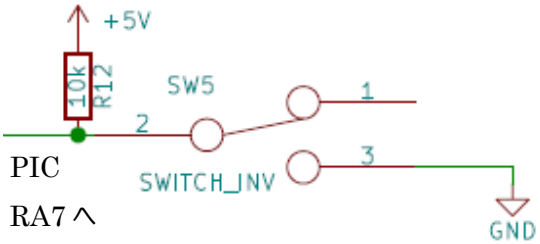


図 6-1 プルアップスイッチ回路図

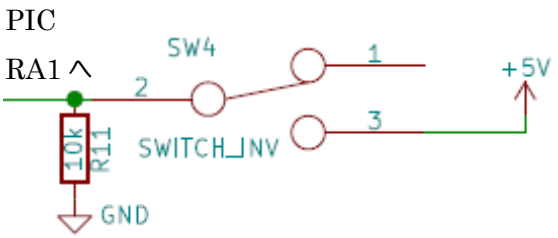


図 6-2 プルダウンスイッチ回路図

7 センサーの入力

サンプルプログラム: 16F88_Ref3_Sensor.X	
電池	9V のみ
RS232C	必要
モータ	不要
動作説明	赤外線が反射してセンサーが受光したか、していないか、PC に送信する.

センサーの入力回路は RB4 に接続されている. 今回のセンサー(RPR220)は赤外線反射センサーであり, 赤外線が反射されると 4 (C:コレクタ)から 3 (E:エミッタ)に電流が流れる. つまり 5V が生じる. そしてインバータ 74HC04 は NOT 演算子であるので, 入力 5V(Hi)の論理が否定された物(Low)が RB4 へ出力される. この関係を表 7-1 にまとめる. なお, インバータ 74HC04 は赤外線センサーの出力を安定させるために使っている. VR2 を調節することでセンサーの感度が変えられる.

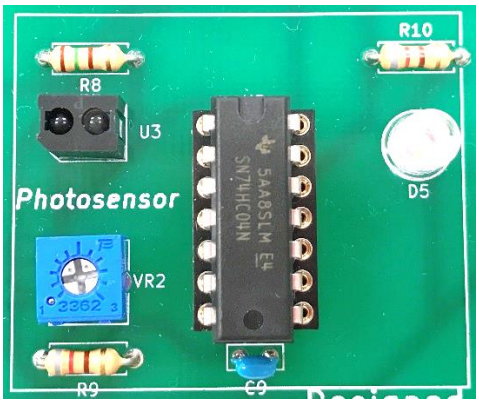


図 7-1 センサー部分

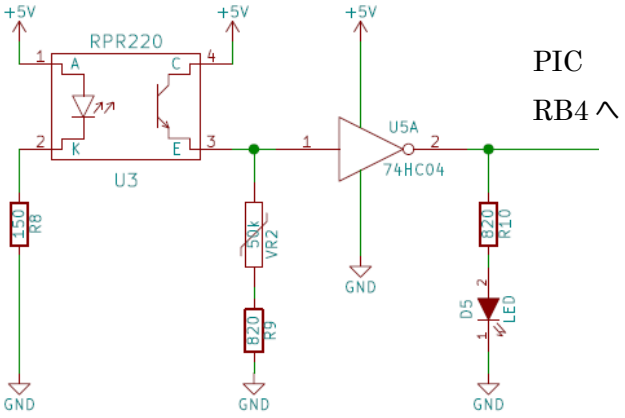


図 7-2 センサー部分回路図

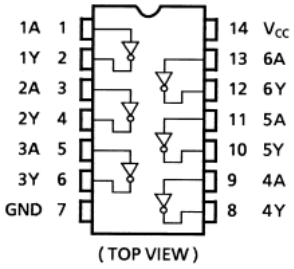


図 7-3 インバータ 74HC04 ピン配置図 (データシートより引用)

表 7-1 センサー部分のロジック

赤外線	センサー出力	74HC04 入力	74HC04 出力	LED D5	PIC RB4
反射	5V	5V(Hi)	0V(Low)	OFF	0
無反射	0V	0V(Low)	5V(Hi)	ON	1

問題

Q17 センサーの状態によって LED を ON/OFF するプログラムを作れ

8 モーターのドライブ

サンプルプログラム: 16F88_Ref3_Motor.X	
電池	9V，3V 両方必要
RS232C	必要
モータ	必要．H-bridge 部のコネクタ(M1)または Transistor 部のコネクタ(M2)に接続すること.
動作説明	SWITCH1 でモータを ON OFF する．H-bridge 部(M1)にモータを繋いでいる場合，モータ停止時に SWITCH2 でモータの正転逆転設定を切り替える.

マイクロプロセッサでモーターを動作させるためには，LED(20mA)と違い大電流(100~500mA)を流す必要がある．そのためトランジスタや FET もしくはモータードライバーIC を用いて制御回路を構成する．始めに，Tr を使った回路について説明する．次にモータードライバーIC である H ブリッジを使った制御手法について説明する.

1) トランジスタによるモータードライブ

今回は，NPN トランジスタである 22SC2120-Y を用いる．本トランジスタは大きな電流を流す事が可能(800mA)なためモータをドライブする事が可能である．小信号の増幅にもちいられる 2SC1815 は 150mW である.

回路図**(図 8-2)**をみて見よう．トランジスタを電子スイッチとして用いる．抵抗 R4 でプルダウンし，抵抗 R6 が電流制限抵抗である．トランジスタのベースに接続された PIC の RB3 を ON にするとコレクタ-エミッタ間に電流が流れてモータが回転する．なおこのときダイオード D3 について注意してほしい．これはフライホイールダイオードと呼ばれる．一見意味が無いように思えるがモータが逆転やブレーキがかかった時にトランジスタに高電圧の信号が逆流する事がある．このときトランジスタや PIC が故障してしまうため，これを防止するためである．なおこのとき通常とは逆の接続であるカソードを電源側，アノードをトランジスタ側に接続する.

また，セラミックコンデンサ C5 はモータのノイズ除去に使われる.

なお，フライホイールダイオードはリレーを動作させる時にも使われる.

また，この接続で速度を変える時はどうすればよいであろうか？RB3 を高速に ON/OFF すれば良い．つまり PWM 制御を行えば良いことになる．PWM 制御について 11 章で説明する．また，実際の回路では FFT**(図 8-5)**を使いもっと大きなモータをドライブさせている.

今回は SW4 を ON にするとモータが回転するプログラムを作成してみる.

ここで，注意することはほとんど前半に説明したが TRISA,TRISB の設定には注意すること.

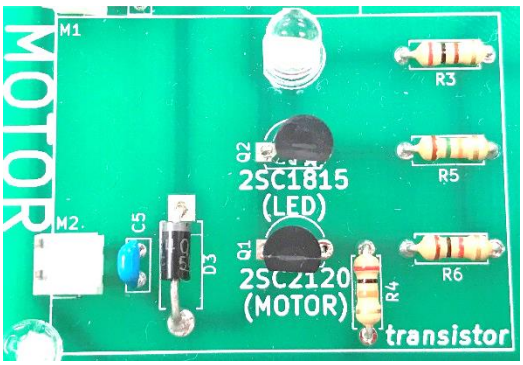
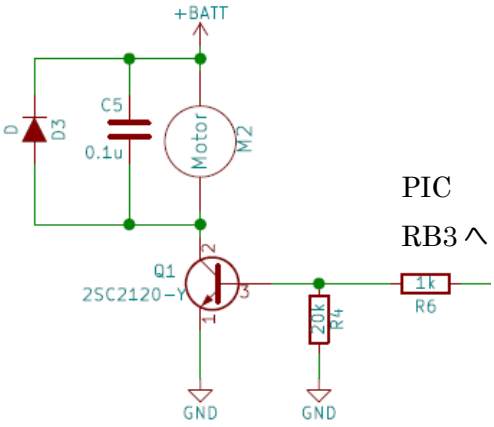


図 8-1 トランジスタ部分



PIC
RB3 へ

図 8-2 トランジスタ部分回路図

E:エミッタ
C:コレクタ
B:ベース

印字面を上にして
ECB(えくぼ)と覚えよう

図 8-3 NPN 型トランジスタ 2SC2120

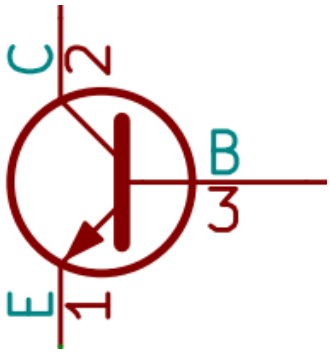


図 8-4 NPN 型トランジスタ回路図記号

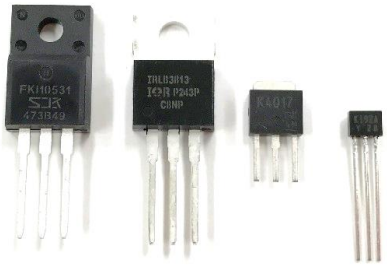


図 8-5 FET

2) H ブリッジによるモータードライブ

前回の回路では，一方方向にしか回転させる事が出来なかったが，車やロボットアームでは正転・逆転を行いたい．このような場合は H ブリッジ回路**(図 8-6)**を用いる．また表の論理を使う事でモーターの回転方向を制御する事が出来る．

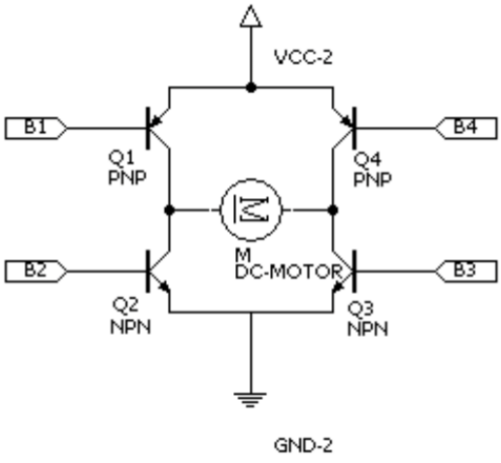


図 8-6 H ブリッジ回路

表 8-1 H ブリッジ回路 ロジック表

B1	B2	B3	B4	モード
Low	Low	High	High	正転
High	High	Low	Low	逆転
High	Low	Low	High	停止

実際には，H ブリッジ IC(TA7291P)を使いる **(図 8-9)**．このときモータと OUT1，OUT2 端子を接続する．また IN1 端子を RA3 に IN2 端子を RA2 に接続する．

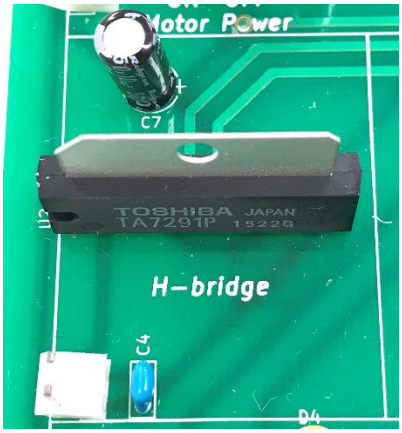


図 8-7 H ブリッジ部分

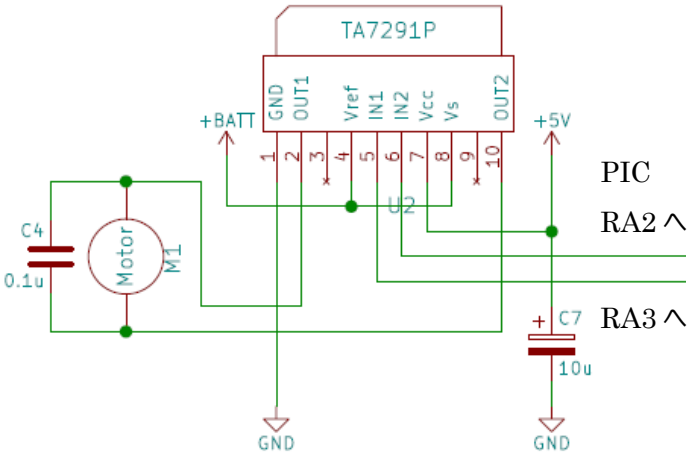


図 8-8 H ブリッジ部分回路図

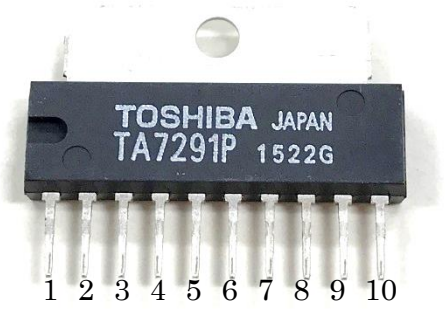


図 8-9 TA7291P

このときの論理を表 8-2 に示す.

これに基づき，回転数の制御をするソフトウエアを作ってみる.

サンプルプログラムは，SW4 が ON ときモータドライブ許可 OFF の時停止．SW5 が ON の時正転 SW5 が OFF の時逆転とするプログラムである.

モータが正転逆転している事が確認出来る.

表 8-2 モータドライバ IC TA7291P ファンクションロジック表

IN1(5 番ピン)	IN2(6 番ピン)	OUT1(2 番ピン)	OUT2(10 番ピン)	モード
Low	Low	∞	∞	ストップ
High	Low	High	Low	正転
Low	High	Low	High	逆転
High	High	Low	Low	ブレーキ

問題

Q18 センサーが ON になったときにモータが回転するプログラムを作れ

9 AD コンバータ

サンプルプログラム: 16F88_Ref3_ADC.X	
電池	9V のみ
RS232C	必要
モータ	不要
動作説明	可変抵抗(VR1)からの電圧値を AD 変換した値を 250ms おきに PC に送信する.

AD コンバータはセンサーの電圧を計測することに使われる. 今回は 5V の電圧を 10bit で計測する. 10bit は 0~1023 までの重みを持つので 5.0/1024=4.88mV 刻みで計測する事が出来る. PIC16F88 は AN0~4 までの 5ch も AD コンバータを持っている. 本リファレンスボードは AN4 にボリュームを接続している. 今回はこのボリュームの出力電圧を計測する.

サンプルプログラムを見てみよう. AD コンバータの設定をしているのは 57 行目である. AN4 を有効にするので右から 5 番目のビットを 1 にして ANSEL = 0b00010000;とする. 例えば AN0 を有効にする場合では右から 1 番目のビットを 1 にして 0b00000001; とする.
次にレジスタの ADCON0 の設定は表を見てほしい. これにより, AD コンバータの準備の第一段階ができた. 最後に, ADCON1 で有るがこれは基準の電圧を電源電圧(5V)とする意味である. 今回は詳細を省く. はじめは ADCON1 と ADCON2 の設定はおまじないと思っておいても良い.

ADCON0

7	6	5	4	3	2	1	0
AD 変換クロック指定		アナログ CH 選択			GO_nDONE AD 開始・停止	-	ADON
00:CLK/2,		000:CH0			0:停止	常に 0	0:AD OFF
01:CLK/8		~			1:開始		1:AD ON
10:CLK/32		101:CH5					
11:内部発振							
1	0	0	0	0	1	0	0

44,45 行が AD コンバータの設定関数である.
setADCChannel 関数は指示された AD コンバータのチャンネルを設定している.
AN4 で channel=4 の時,

ADCN0 &= 0b11000111; は ADCN0 =b10000001 & 0b11000111 であるから&演算を行うと．ANCN0=0b10000001 となる．（アナログ CH 選択をリセット．全て 0 にする）
次に ADCON0 |=(channel << 3);であるから，ADCON0=0b10000001 | (0b100 << 3) =0b10000001 | 0b00100000 = **0b10100001**となる．つまり表より読み解くと，AD 変換は CLK/32 周期で行い，AD4 を利用し，AD 変換は停止して，AD 機能は ON となっている．

readADCValue 関数は AD 変換値を 10bit で返す．まず，AD 変換を監視している GO_nDONE レジスタに 1 をセットすると AD 変換が開始される．GO_nDONE が終了すると 0 となる．このとき，10bit の AD 変換値の 10~9bit 分は ADRESH に記憶される．また 8~1bit 分は ADRESL に記憶される．これを 10bit の数値に戻すために，ADRESH を左に 8bit シフトして計算している．

計算例

	bit	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
上位	ADRESH	-								0	0	0	0	0	0	1	0
下位	ADRESL	-								1	1	1	0	1	1	1	1
	AD 変換値	0	0	0	0	0	0	1	0	1	1	1	0	1	1	1	1
HEX	0x	02								EF							
DEC		751															

実際に使った例が，65~67 である．setADCChannel(4)で 4ch を設定し，ans = readADCValue();で 10bit の AD 変換値を読んでいる．なお 66 行の__delay_us(50);はアクィジション時間であり AD 変換にかかる時間である．十分に時間を取ることが必要である．

問題

Q19 AD コンバータの出力と，ボリュームの接続されている PIC の RA4 ピンと GND 間の電圧の関係を確かめよ．

RA4 と GND 間電圧[V]	AD 値 DEC (10 進数表記)	AD 値 HEX (16 進数表記)
0		
1		
2		
3		
4		
5		

10 タイマー割り込み

サンプルプログラム: 16F88_Ref3_Timer0.X	
電池	9V のみ
RS232C	必要
モータ	不要
動作説明	Timer0 割り込みを使って LED1 と LED2 を 1 秒おきに交互に点灯させる．RS232C に hello world と送信する.

タイマー割り込みとは，一定周期で決まった動作を確実に起こさせる時に行う．例えば 1 秒ごとに LED を点滅させると言った仕事に向いている.
今までの LED の ON/OFF を周期的に行うためには__delay_ms();等を使い CPU を無駄に動作させてきた．しかし，タイマー割り込みを使うと複数の仕事を擬似的に同時に行う事が出来る．マイコンは一定時間ごとに複数のタスクを切り替えて使う事が出来る．マイコンを使う際に重要となる機能の一つである.
重要な所はサンプルプログラム 39~42 行目の変数定義である．重要なのは 42 行目の 1 秒間に何回カウンターを回すかと言う点である．今回はプリスケーラと言うマイコンのクロック(8Mhz)を 256 分割した物を単位として使う．このときの最小割り込み時間 Ts は式 1 より 0.033 秒となる.

$$(1[s]/\text{クロック}[Hz]) * 4 * \text{プリスケーラ設定値} * (255 - \left(TMR0\ \text{初期値}\right) - 1) = \text{割り込み間隔}[\text{s}] \quad \text{（式 1）}$$

$$(1/8000000) * 4 * 256 * \left(255 - (0 - 1)\right) \approx 0.033[\text{s}]$$

これを 1/Ts とした回数が 1 秒間に必要なカウンター値となる．今回は 1 秒ごとに処理したいので 1/Ts=30.52 なので CNT_DATA を 30 としている.
また，53~55 行目がタイマーのプロトタイプ宣言であり initTMR0 はタイマーのモードを決める．setTMRO が内部タイマーに値をセットする．55 行目が特殊な書き方であるが isrTMR0()が実際にタイマー割り込みで実行したい関数である．つまり，0.033 秒おきにタイマー割り込みが発動し，void interrupt isrTMR0()関数部分が呼び出される．この関数の中に，周期的に動作させたい事を書く.
今回は LED を反転させている．101 行目がビット演算子(排他的論理和)を使い反転を行っている.
98 行目で cnt 値を減らしていき，99 行目で 0 になったら LED 反転等を行う処理を記述している.
メイン関数の内部では，72~76 行目の順番を必ず守って実行する事.
タイマー割り込みは初心者には難しいがロボットの高度な制御には必要な部分である.

問題

Q20 タイマー割り込みを使い，2Hz で LED を点滅させよ.

11 PWM 制御

PWM (Pulse Width Modulation（パルス幅変調）)制御はモータの速度制御の有名な手法である. 図 11-1 に示されるようにある一定周期 T の時にT_{on}となる時間の比率でモータに与える電力を制御する手法である.

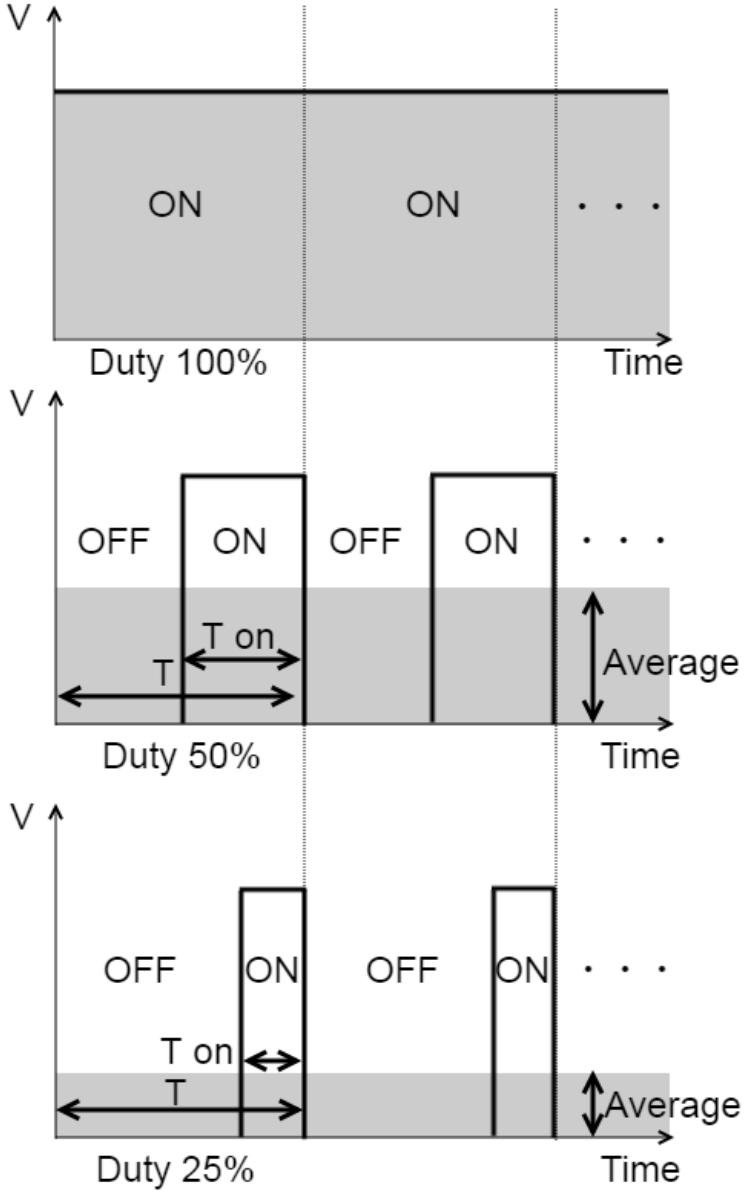


図 11-1 PWM 原理図

このとき 2 つの手法がある．それはタイマー割り込みを使う方法と CCP(Compare/Capture/PWM)と言われる比較器を使った手法がある.

モータ制御に特化したマイクロプロセッサー等では CCP モジュールが多数準備されている．(PIC16F88 は 1 個)

ここでは，始めにタイマー割り込みを使った例を説明する．次に CCP モジュールについて説明する.

1) Timer0 割り込みを使った PWM 制御

サンプルプログラム: 16F88_Ref3_PWM_Timer0.X	
電池	9V，3V 両方必要
RS232C	必要
モータ	必要．H-bridge 部のコネクタ(M1)または Transistor 部のコネクタ(M2)に接続すること.
動作説明	Timer0 割り込みを利用して PWM 出力する．AD コンバータの読み取り値を PWM の Duty 比とする．モータを H-bridge 部(M1)に接続した場合、モータ停止時に SWITCH1 で正転逆転設定を切り替えることができる.

タイマー割り込みについては前章で説明した．前章ではタイマー割り込みを使って LED を点滅させたが，ここではモータの PWM 制御に利用する．基本的な使い方は LED 点滅と同じである．PWM 制御の場合は割り込みの周期 T を短くする．そして，On になる時間T_{on}の長さを調節することにより Duty 比を変え，モータの回転数を決める.

サンプルプログラムを見てみよう.

43~46 行目でタイマー割り込み関係の変数定義をしている．10 章の式 1 より，今回の割り込み時間は

$(1/8000000) * 4 * 2 * \big(255 - (0 - 1)\big) = 2.56 * 10^{-4}[\mathrm{s}] = 0.256[\mathrm{ms}]$

となる.

また，グローバル変数 cnt を定義している．これは，初期値を 16 とし，1 回割り込みが発生するごとに -1 されていき，0 になると 16 に戻る変数である．126~152 行目の void interrupt isrTMR0()関数を見てみよう．10bit の分解能(0~1023 の値をとる)を持つ AD コンバータ値を 64 で割ることにより 4bit の分解能(0~15 の値をとる)に変換し，このときの値を PWM の Duty 比に設定している．つまり，モータの回転速度は 0~15 までの 16 段階に設定されるということである.

例えば，AD コンバータ値が 700 だった場合

700/64=10 (整数型の計算のため，小数点以下は切り捨てられる)

この値よりも cnt 値が小さいときに，モータが On になる．すなわちその間がT_{on}にあたる.

割込回数	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
cnt	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
出力	off	off	off	off	off	off	off	on	on	on	on	on	on	on	on	on

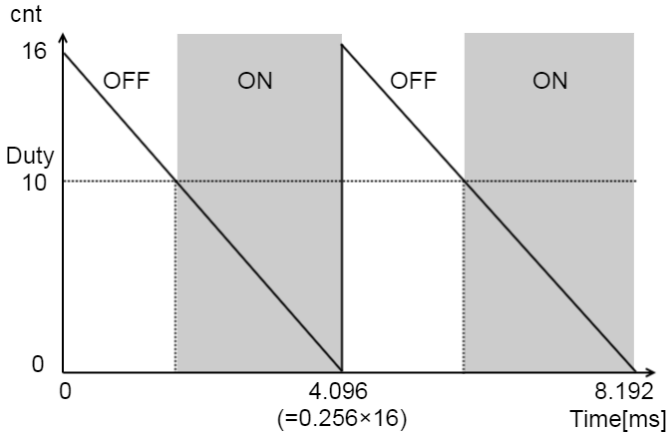


図 11-2 Timer0 割り込み利用 PWM 原理図

AD コンバータ値が大きければ Duty 比は大きく，モータが On になる時間T_{on}は長い．よって回転は速くなる．逆に AD コンバータ値が小さければ Duty 比は小さく，モータが On になる時間は短い．よって回転は遅くなる．

サンプルプログラムでは，モータ停止時に SWITCH2 を切り替えることにより H-Bridge 回路に接続されたモータの回転方向設定を変えることができる．モータを Transistor 回路に接続しても回転するが，回転方向は一定である．

なお，ライントレースロボットでは，タイマー割り込みによる PWM 制御を用いる．

2) CCP モジュールを使った PWM 制御

サンプルプログラム: 16F88_Ref3_PWM_CCP.X	
電池	9V，3V 両方必要
RS232C	必要
モータ	必要． Transistor 部のコネクタ(M2)に接続すること．
動作説明	CCP を利用した PWM の動作確認プログラム．AD コンバータの読み取り値を PWM の Duty 比とする．モータは transistor 部のコネクタ(M2)に接続する．

CCP モジュールは PWM を簡単に実装できるようにしたマイクロプロセッサーのハードウエアの機能である．

CCP モジュールは PIC16F88 では 1 つしかついていないため今回実験で作成するロボットには使えない．

PIC24F シリーズには 6 本もの PWM を制御できるものもあり，これらはブラシレスモータ(3 相モータ)の制御に使われている．ドローンのモータ制御にも多用されている．

まず，CCP モジュールは CPU クロック(8MHz)とプリスケーラー(256)，内部倍率(4)により，周期 Ts は Ts=256*4*1/8M=0.128mS=7.8KHz となる．

さらに 10bit(1024)で分割されるので**「周期 7.8KHz で 10bit の分解能を持つ PWM」**を作ることが出来る．1bit の時間の重みはどれくらいかというと 0.125uS であり非常に細かな速度制御が可能であると

いえる．また，設定された duty の値(0~1023)をオーバーするまでは RB3 が ON となる．それ以降は OFF となる．例えば 50%のパワーを出したいときは duty=512 とすればよい．

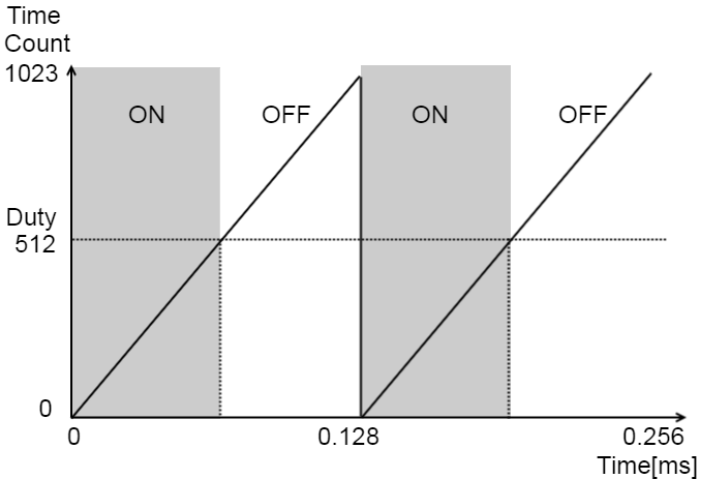


図 11-3 CCP 原理図

39~41 行目が今回の設定する**「周期 7.8KHz で 10bit の分解能を持つ PWM」**の初期設定値である．

48~50 が CCP を設定する関数であり，中身は 82~97 行に記載されている．実際に使うのは setPWM1Duty 関数であり，７４行に記載されている．今回はボードのボリュームを回転させることによりその速度を変化できるようにしている．

RB3 に接続されたトランジスタに PWM 信号をいれモータを回転させている．

Timer0

CPU 動作クロック	8	MHz
CPU 動作周期	0.125	uS
プリスケーラ	256	倍
プリスケーラ内部倍率	4	倍
設定最小時間	32.768	mS
周期時間	1	S
カウンター値(1S)	30.51757813	-

CCP

CPU 動作クロック	8	MHz
CPU 動作周期	0.125	uS
プリスケーラ	256	倍
プリスケーラ内部倍率	4	倍
周期時間	0.128	mS
周期(Hz)	7.8125	kHz
最大分解能 10bit	1024	-
設定最小時間(1bit の重み)	0.125	uS

12 おまけ ブザーからメロディーを鳴らす

サンプルプログラム: 16F88_Ref3_Buzzer.X	
電池	9V，3V 両方必要
RS232C	必要
モータ	不要
ブザー	必要．H-bridge 部のコネクタ(M1)に接続すること．
動作説明	Timer0 割り込みを利用した PWM を応用して，ブザーからメロディーを流すプログラム．メロディーは 3 曲書き込まれており，どの曲を流すかは SWITCH1 と SWITCH2 を切り替えることで選択することができる．

これまでとは少し趣向を変えて，PIC16F88 評価ボードから音楽を流してみよう．モータ制御で使った PWM を応用すれば，音階に合った周波数を発生されることができ，モータの代わりにブザーを取り付けることでメロディーを鳴らすことも簡単である．

1) ブザーを鳴らすサンプルプログラムを実行してみよう！

1. サンプルプログラムを MPLAB で開く

東京海洋大学 田原研究室 > 講義 > 制御システム工学演習 > PIC16F88 評価ボード ページ最下部よりサンプルプログラム「16F88_Ref3_Buzzer.X」をダウンロードできる．

URL: http://www2.kaiyodai.ac.jp/~jtahar0/posts/activity72.html

2. PIC16F88 評価ボードの H-bridge 出力端子(図 13-1 の位置)にブザーを接続する．

電池は単 3 電池・9V 電池両方必要である．

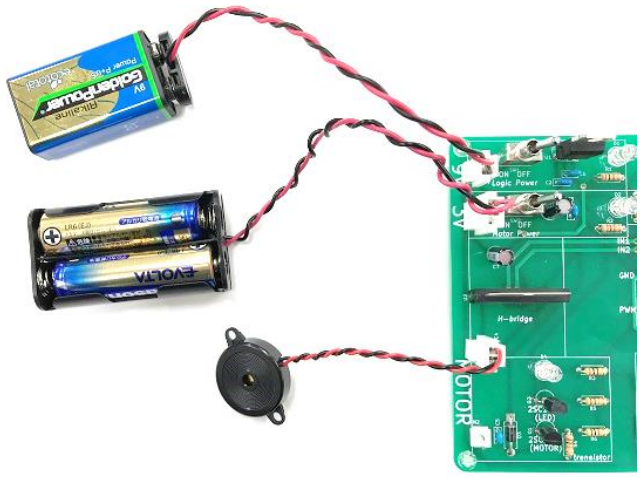


図 13-1 ブザーと電池の取り付け方

3. PIC16F88 評価ボードの Logic Power と Motor Power スイッチを両方 ON にして，サンプルプログラムを PIC16F88 評価ボードに書き込む

4. 書き込みが完了すると，ブザーからメロディーが流れる.
メロディーは 3 曲書き込まれており，どの曲を流すかは SWITCH1 と SWITCH2 を切り替えることで選択することができる．何の曲が流れるかは，流してからのお楽しみである．お試しあれ.

2) サンプルプログラムを改造して自分の好きな曲を鳴らしてみよう！

プログラムの書き方は非常に簡単． main 関数の while 文(無限ループ)の中に，楽譜に従って tone 関数を並べていけばよい． tone 関数の記述方法については次に説明する.

3) tone 関数について

インターネットで「tone 関数」と検索すると，Arduino 向けの関数が出てくるが，ここで説明する tone 関数は PIC 向けに作ったオリジナル関数であり，Arduino 向け tone 関数とは使い方が異なるため注意.

tone('音階名',オクターブ,音の長さ);

・音階名
音階はアルファベット表記で入力する．シャープの場合は小文字にする.

入力記号	対応する音階
C	ド
c	ド#
D	レ
d	レ#
E	ミ
F	ファ
f	ファ#
G	ソ
g	ソ#
A	ラ
a	ラ#
B	シ
(スペースまたはアンダーバー)	休符

・オクターブ

3,4,5 のいずれかの数字を入力する．数字が大きいほど高い音になる．

・音の長さ

入力数値	対応する音符
1	全音符
2	2 分音符
4	4 分音符
8	8 分音符♪
16	16 分音符

よって，例えばドの 8 分音符を鳴らす場合は tone('C',4,8); と記述する．

また，4 分休符を入れる場合は tone(' ',0,4); と記述する．

謝辞

H28 年度~H29 年度にかけてボードの作成と資料の作成を電子制御実験室にて行った．本テキストをまとめるにあたり，修士学生 伊藤大智君の協力無くては出来なかった．田原の手書きの回路図やプログラムからリファレンスボードの CAD データを作成し，テスト基板の作成を行い，海外への発注を行ってくれた．また，資料作成時には汚い手書き原稿をまとめてくれた事には大いに感謝したい．さらに，25 枚ものボードを作成するに当たり，卒研学生の桐谷公基君が開発したリフロー炉を使う事により一日で製作できた．また，部品・PC の準備には中村圭君，齋藤幹太君，宮野枝梨花さんの協力を得た．当研究室のスタッフ全員を駆り出して，急ピッチで進められたがなんとかまとめる事ができた．

今後共，本資料・ボードを良くしていくにはバグの報告や改善点を報告してもらうことが必要である．是非とも使用者の皆様にお願いしたい．

2017/11/02 田原 淳一郎

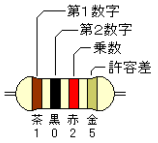
制御システム工学演習　–電子部品の基礎-

抵抗

この抵抗のカラーコードは茶・黒・橙・金．　1・0・3・±5%.

よって **10**×10^**3**=10000[Ω]=10[kΩ]　許容差**±5%**

［第1図］カラーコードの読み方



上記の抵抗は
10×10²Ω
＝10×100Ω
＝1000Ω
＝1kΩ
許容差 ±5%

色	第1数字	第2数字	乗　数	許容差
黒	0	0	10^0(×1Ω)	
茶	1	1	10^1(×10Ω)	±1%
赤	2	2	10^2(×100Ω)	±2%
橙	3	3	10^3(×1kΩ)	
黄	4	4	10^4(×10kΩ)	
緑	5	5	10^5(×100kΩ)	±0.5%
青	6	6	10^6(×1MΩ)	
紫	7	7		
灰	8	8		
白	9	9		
金			10^{-1}(×0.1Ω)	±5%
銀			10^{-2}(×0.01Ω)	±10%
無				±20%

コンデンサ

電解コンデンサ

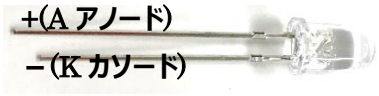
極性がある．足の長いほうが+，白い線のあるほうが-

積層セラミックコンデンサ

極性はない．104 と印字されているコンデンサの容量は

10×10^**4**[pF] = 100000[pF]=0.1[μF]

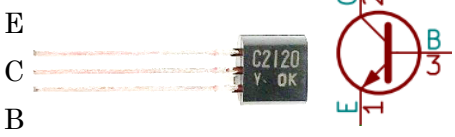
LED



LED に取り付ける抵抗の求め方
(電源電圧 Vcc[V]-順方向電圧 Vf[V])÷順方向電流 If[A]=抵抗値 R[Ω]

極性がある．足の長いほうが+（アノード）,真上から見て円の一部が平らになっているほうが-（カソード）

トランジスタ（NPN 型）



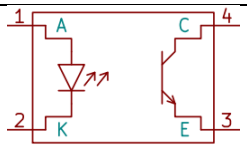
平らな面（印字面）を上に向けた状態

E: エミッタ
C: コレクタ
B: ベース

モータドライバ（TA7291P）

反射型フォトセンサ（RPR220）

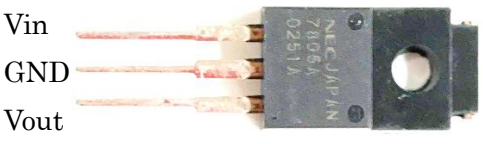
赤外線の反射を検知する

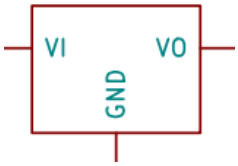


真上から見たときのピン配置

K: カソード　E: エミッタ
A: アノード　C: コレクタ

レギュレータ





印字面を上に向けた状態

ダイオード

極性がある．白い線のあるほうがー（カソード）

A（アノード）から K（カソード）へのみ電流が流れる

2017 年 11 月 2 日　作成者: 伊藤大智

制御システム工学演習　–PIC16F88 ピン配置表-

設定	#define	接続先		PIC16F88			接続先	#define	設定
			1	RA2	RA1	18			
			2	RA3	RA0	17			
			3	RA4	RA7	16			
			4	RA5 (MCLR)	RA6	15			
			5	Vss	Vdd	14			
			6	RB0	RB7 (PGD)	13			
			7	RB1	RB6 (PGC)	12			
			8	RB2 (RX)	RB5 (TX)	11			
			9	RB3	RB4	10			

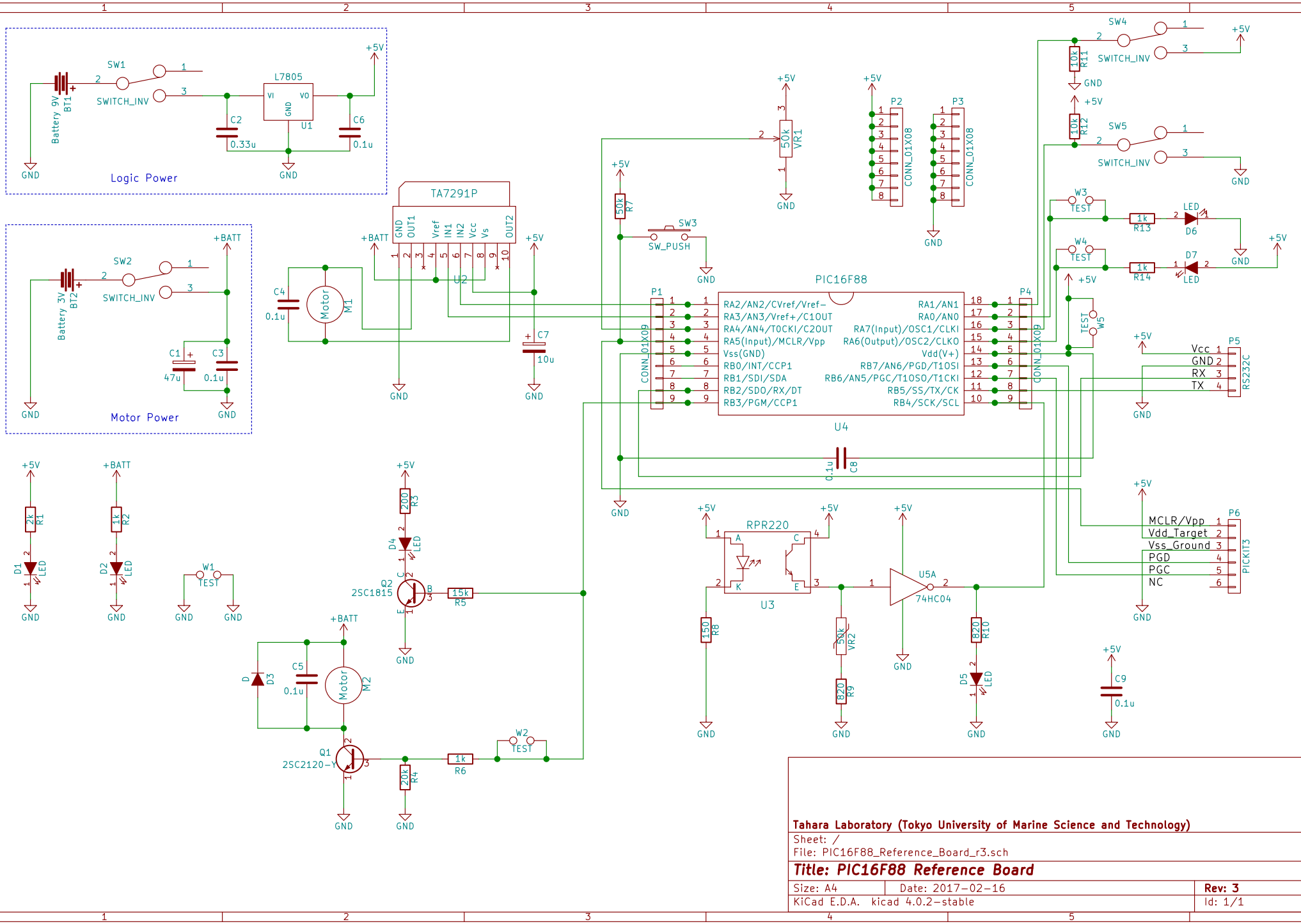
注) RA5, RA7 は入力専用ピン,　RA6 は出力専用ピン

記入例 (PIC16F88 評価ボード)

設定	#define	接続先		PIC16F88			接続先	#define	設定
出力	M_IN1	H-Bridge: IN1	1	RA2	RA1	18	SWITCH1	SWITCH1	入力
出力	M_IN2	H-Bridge: IN2	2	RA3	RA0	17	True ON LED	LED1	出力
入力	VOLUME	VR1	3	RA4	RA7	16	SWITCH2	SWITCH2	入力
		Reset	4	RA5 (MCLR)	RA6	15	False ON LED	LED2	出力
		GND	5	Vss	Vdd	14	5V		
		Not Connect	6	RB0	RB7 (PGD)	13	PGD		
		Not Connect	7	RB1	RB6 (PGC)	12	PGC		
入力		RX	8	RB2 (RX)	RB5 (TX)	11	TX		出力
出力	TR	Transistor	9	RB3	RB4	10	Photosensor	SENSOR	入力

学籍番号＿＿＿＿＿＿＿＿＿＿

名前＿＿＿＿＿＿＿＿＿＿


Logic Power
Motor Power
PIC16F88
TA7291P
L7805
RPR220
74HC04
SW_PUSH
SWITCH_INV
CONN_01X08
RS232C
PICKIT3
Tahara Laboratory (Tokyo University of Marine Science and Technology)
Sheet: /
File: PIC16F88_Reference_Board_r3.sch
Title: PIC16F88 Reference Board
Size: A4
Date: 2017-02-16
Rev: 3
KiCad E.D.A. kicad 4.0.2-stable
Id: 1/1