

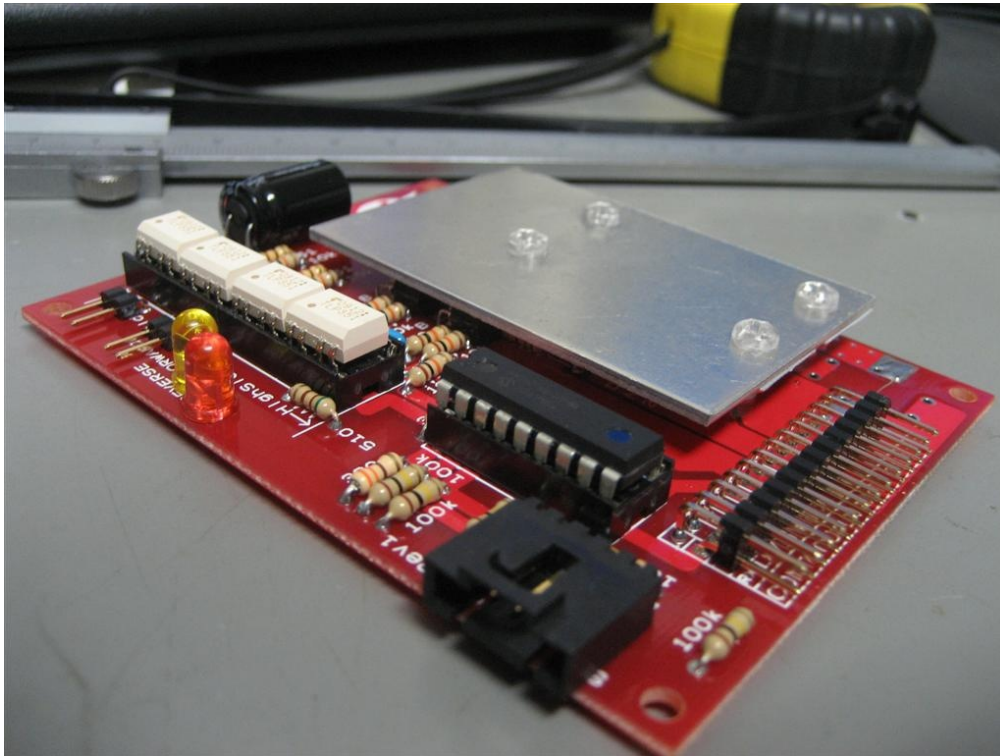
SPI モータドライバ Ver1

呉高専電子情報工学科 4年:中谷 寿洋

概要

SPI モータドライバ Ver1 はノーマルモータドライバ Ver2 の改良版にあたります。
サーボ処理等を必要とするモータに向いています。
実績:高専ロボコン 2009 年度, 2010 年度

外見



特徴

1. 16F88 の SPI 通信を使った制御方式

16F88 内蔵の MSSP モジュールを使って SPI 通信を行うことでメインマイコンから制御します。
またセンサー入力用のコネクタを設けており、フォトインタラプタ、ボリューム、
ロータリーエンコーダなどを接続することができます。さらに SPI 通信は双方向通信であるため、
メイン側マイコンからフィードバック処理を行うことも出来ます。

2. シンプルな構成

一般的な PchNch 混合 H ブリッジを 16F88 と TLP351 で駆動しています。

3. ヒートシンクによる放熱

アルミ製のヒートシンクを搭載することでファンなしで 4W 程度、ファンありで 10W 程度の
放熱を行う事ができます。

4. LED による動作確認

赤色と黄色の LED を搭載し、モータドライバが動作しているか目視で確認することができます。
また、ロジック側に LED が付いているためモータ側電源を接続しなくても動作の確認が行えます。

(赤字:大きな改良点, 青字:小規模 or 前作からの引き継ぎ)

仕様

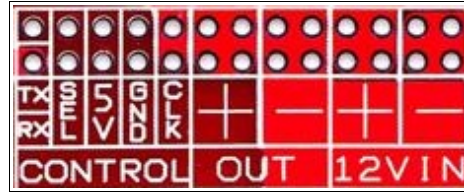
仕様	説明	備考
モータ側電圧	12-16.8V	LiPo4セルが限界,定格上20Vだがマージンで3Vほど確保
ゲート駆動電圧	モータ側電圧と一緒	
モータ側連続電流	12A	コネクタと接続ケーブルの定格で制限
モータ側最大電流	74A	ケーブルが死ぬので数秒間のみ
オン抵抗(MAX)	Nch:7.2mΩ, Pch:20mΩ	
ロジック側電圧	5V	
ロジック側電流	20mA程度	
PWM周波数	7.8kHz	

通信方式	SPI通信	通信フォーマットは後述
センサー入力	低速ロータリーエンコーダ ポテンシオメータ フォトインタラプタ	ソフトウェア処理なので高速品は処理落ちします。
制御機能	サーボ, 可動範囲制限	P-D制御サーボ, フォトインタラプタによる可動範囲制限

ピン配置-制御 & モータ

制御側 (CONTROL)

GND	制御側電源入力	0V
5V	制御側電源入力	5V
TX	SPI_DATA_OUT(SDO)	SPIデータ出力
RX	SPI_DATA_IN(SDI)	SPIデータ入力
SEL	SPI_SEL(SS)	SPIセレクト入力
CLK	SPI_CLOCK_IN(SCI)	SPIクロック入力



モータ & 電源側 (OUT, 12VIN)

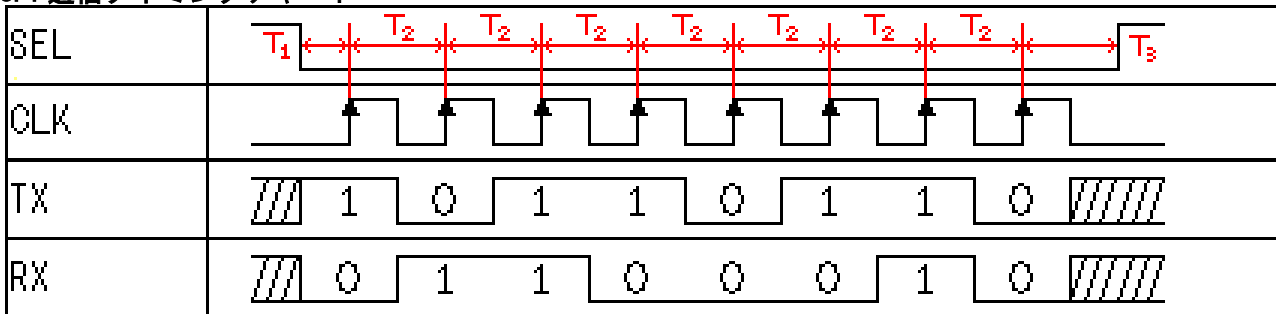
12VIN	モータ側電源入力	12-16.8V
OUT	出力	

ピン配置-センサー入力コネクタ

C	デジタル入力	TTL
B	デジタル入力	TTL
A	デジタル & アナログ入力	TLL, 8BitA/D
5V	センサー電源出力	5VOUT
GND	センサー電源出力	GND



SPI 通信タイミングチャート



通信について

SEL ピンがHからLにすると T₁後に TX から 7Bit 目のデータが出力されます。
 CLK の立ち上がりごとにビットが進んでいきます。CLK の周期は T₂以上にする必要があります。
 0Bit 目、終了後 T₃分だけ待機してから SEL を L から H に戻せば通信終了です。
 この例ではモータドライバ側からの出力:10110110、入力:01100010 をしています。
 T₁:2us, T₂:500ns, T₃:2us

SPI 通信コマンド & センサーとの接続

モータドライバに搭載されたプログラムによって動作が違います。
 また、対応するセンサーも違うので注意が必要です。

汎用プログラム

フォトインタラプタやスイッチ用に用いる。

マザーボード→モータドライバ

7bit	回転方向	1:正転, 0:逆転
6-0bit	PWM値	127:最大, 0:最低

例外:PWM 値が 0 の時 7bit 目が 1 ならブレーキ, 0 ならフリー

モータドライバ→マザーボード

7-3bit	空きビット	常に0
2bit	デジタル入力C	H:1, L:0
1bit	デジタル入力B	H:1, L:0
0bit	デジタル入力A	H:1, L:0

センサー→モータドライバ

A, B, C のデジタル入力に 5V, TTL レベルの信号を出力するセンサーを取り付けます。
 (なお、以降、デジタル入力には 5V, TTL レベルのセンサーを接続するものとします。)
 必要がなければセンサーをつなぐ必要はありません。
 また、A 入力だけにセンサーをつなぐといったことも許容されます。

ロータリーエンコーダ用プログラム

低速なロータリーエンコーダ用に用いる。

マザーボード→モータドライバ

汎用プログラムと同等なので省略

モータドライバ→マザーボード

7-0Bit	カウンタ値	0-255
--------	-------	-------

注意:

オーバーフローすると 254→255→0 または 1→0→255 となる。

SPI のアクセス周期よりカウンタが早ければ 2 以上一気に変化する。

センサー→モータドライバ

A, B のデジタル入力にロータリーエンコーダの A 相, B 相を接続します。

A 相, B 相を逆に接続した場合は逆方向にカウンタされるので状況に応じて利用してください。

サーボ用プログラム

ボリュームを使ってサーボを構成します。PD 処理を用いているため、P ゲイン, D ゲインを設定する必要がありますが、通信によって指定することはできません。そのため、状況にあわせてモータドライバ側のプログラムを書き換えて調整する必要があります。

マザーボード→モータドライバ

7-0Bit	目標A/D値	0-255
--------	--------	-------

例外: 0 か 255 を指定した場合はサーボ処理を中断してフリーになる。

モータドライバ→マザーボード

7-0Bit	現在A/D値	0-255
--------	--------	-------

センサー→モータドライバ

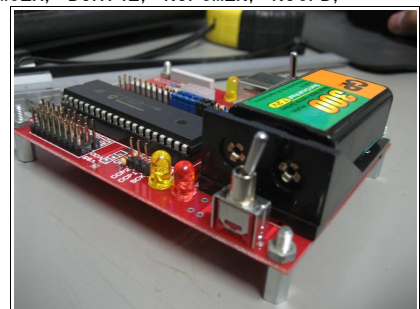
A アナログ入力にボリュームの中間端子を接続します。両端の端子にはそれぞれ 5V, GND を接続してください。なお、5V, GND の方向を間違えて接続すると振り切れてしまうので、**ボリュームをモータに取り付ける前に動作を確認**しておいてください。

プログラムサンプル(汎用 or ロータリーエンコーダプログラム用のサンプル)

マザーボード側のプログラムサンプルです。XBee 実験基板用。

```
//XBee 実験ボード用の設定, セラロック 20MHz 用 (PIC18F4520)
#include <18f4520.h>
#define HS, NOWDT, PUT, NOPROTECT, NOLVP, BROWNOUT, NOCPD, NOWRT, MCLR, BORV42, NOFCMEN, NOCPB,
NOPBADEN, CCP2C1//, CCP2B3
#define delay(clock = 20000000)//20MHz
#define fast_io(A)
#define fast_io(B)
#define fast_io(C)
#define fast_io(D)
#define fast_io(E)
#define DEF_TRISA 0b00000000
#define DEF_TRISB 0b00000000

#define DEF_TRISC 0b10010000//XBee 用に RC7 を入力モードに
#define DEF_TRISD 0b00000000
#define DEF_TRISE 0b00000000
//SPI 関連の定義
#define SPI_OUT PIN_C5
#define SPI_IN PIN_C4
#define SPI_CLK PIN_C3
#define SPI_SEL_1 PIN_A0
//プロトタイプ宣言
void spi_reset_select();
int s_driver1_set_value(int value);
void main() {
    //ポートの初期化
```



```

output_a(0);
output_b(0);
output_c(0);
output_d(0);
output_e(0);
//TRISの設定
set_tris_a(DEF_TRISA);
set_tris_b(DEF_TRISB);
set_tris_c(DEF_TRISC);
set_tris_d(DEF_TRISD);
set_tris_e(DEF_TRISE);
//SPI通信の設定
setup_spi(spi_master | spi_l_to_h | spi_clk_div_64);
spi_reset_select();
while(1){
    s_driver1_set_value(0b11111111);//正転
    delay_ms(1000);
    s_driver1_set_value(0b00000000);//フリー
    delay_ms(1000);
    s_driver1_set_value(0b01111111);//逆転
    delay_ms(1000);
    s_driver1_set_value(0b10000000);//ブレーキ
    delay_ms(1000);
}
}
void spi_reset_select(){
    output_high(SPI_SEL_1);
}
int s_driver1_set_value(int value){
    int temp;
    output_low(SPI_SEL_1);
    delay_us(2);
    temp = spi_read(value);
    delay_us(2);
    output_high(SPI_SEL_1);
    return temp;
}
}

```

プログラムサンプル (サーボ用のサンプル)

サーボ用プログラムのサンプルです。変更箇所のみ掲載。

モータドライバにボリュームを接続しておき、ボリュームをギヤードモータに取り付けておいて下さい。※モータのプラスマイナスに注意。分からない場合は手で試してフィードバックがかかるか試しておいてください。

```

while(1){
    s_driver1_set_value(50); //角度 50
    delay_ms(1000);
    s_driver1_set_value(200); //角度 200
    delay_ms(1000);
}

```

設計メモ

1. 放熱について

ファンが無い状態でも 540 モータ等の高負荷モータを駆動することは可能ですが、長時間又は高負荷をかけて駆動すると MOSFET がかなり発熱し破壊されることがあります。540 モータを駆動するときはヒートシンクの温度に注意してください。

マニュアル上では強制空冷しない場合 4W と書いてありますが、回路の置き方次第で大きく変化します。平積みする場合は特に注意が必要です。

2. 電源について

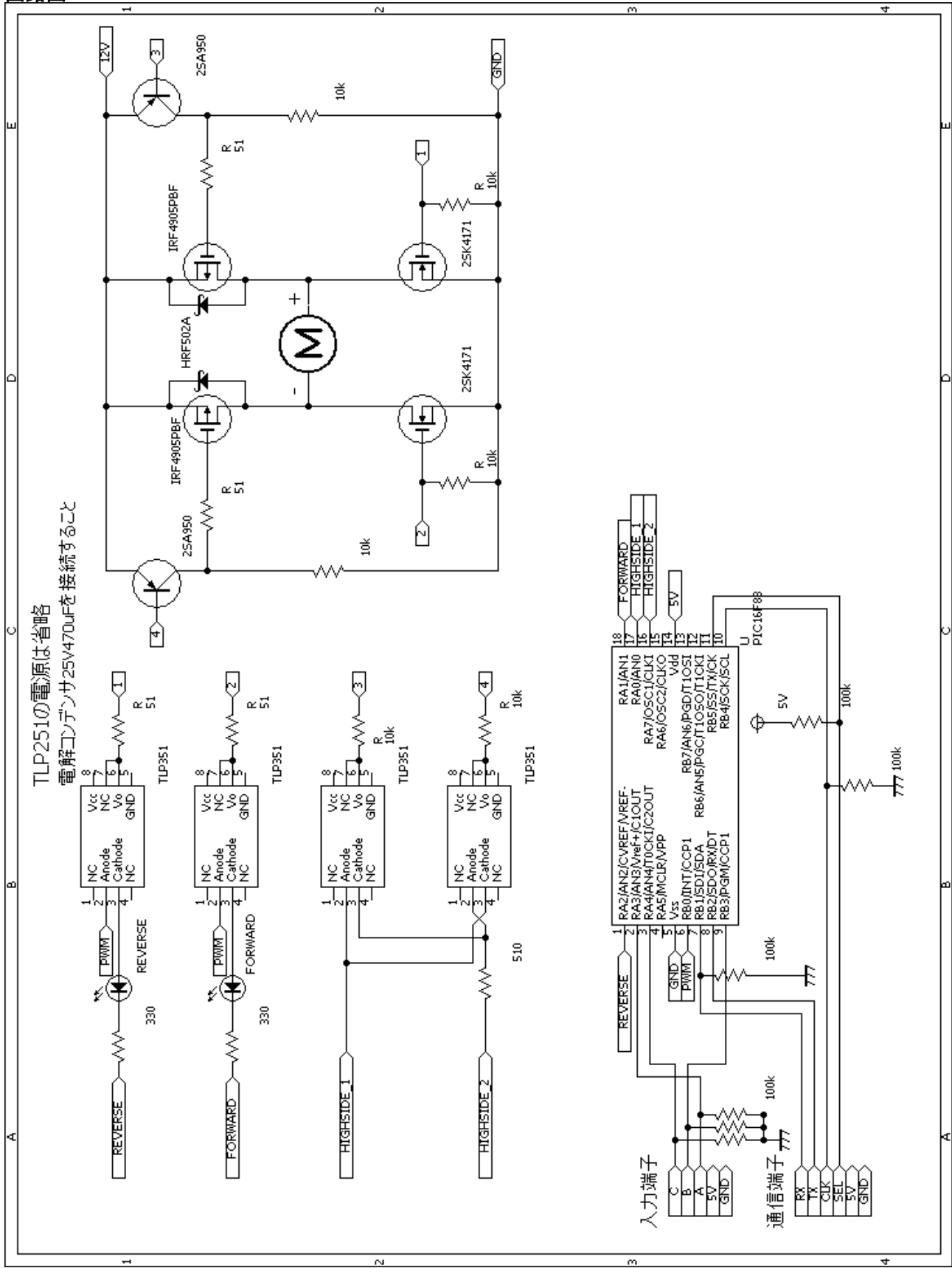
このモータドライバは NiCd12V バッテリーと LiFe13.2V バッテリーでテストしました。

LiPo バッテリーでのテストはまだ不十分でないので注意してください。

3. フォトカプラについて

回路図上では TLP351 となっていますが、部室に大量の TLP251 があったので置き換えられている物もあります。TLP351の方が待機電流が少ないので出来ればそちらをおすすめします。

回路図

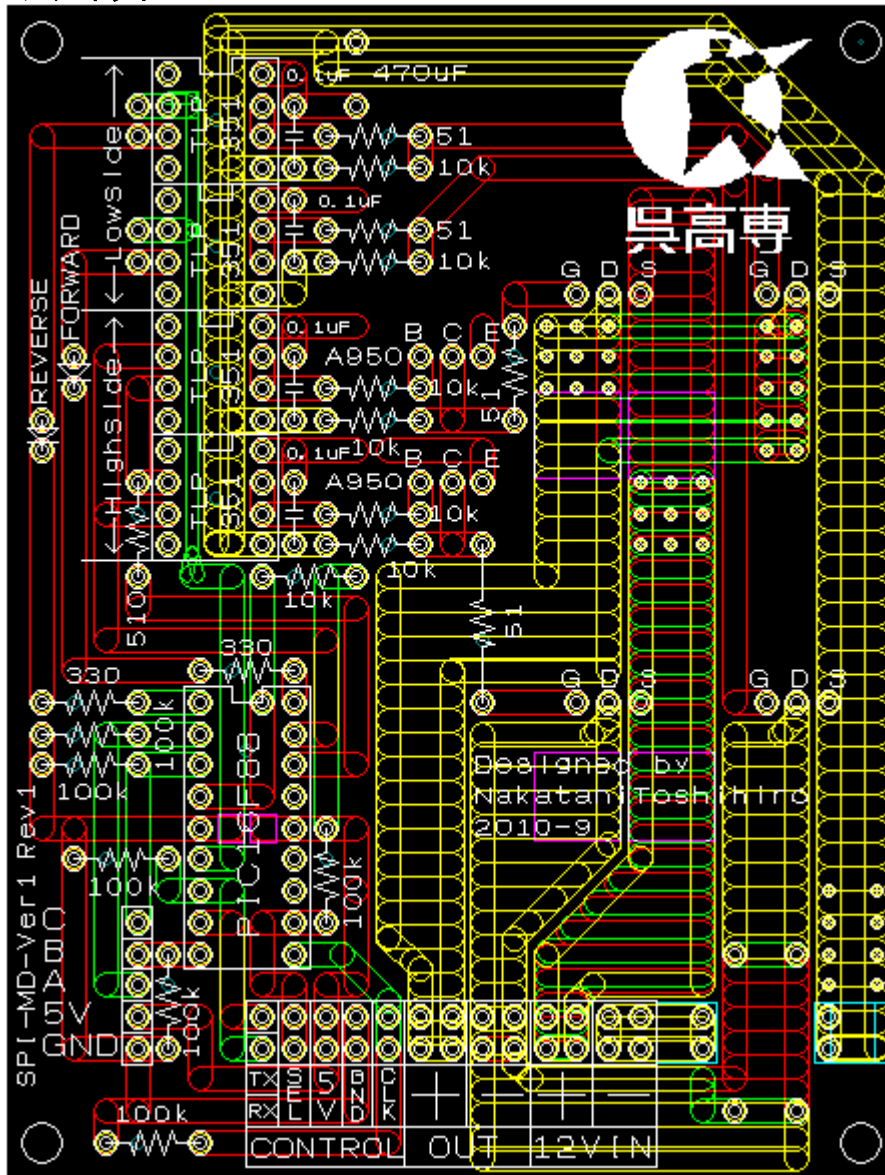


部品表

SPIモータドライバVer1 部品表

商品名	価格	数量	合計	
2SK4171	¥131	2	¥262	NchMOSFET
IRF4905PBF	¥166	2	¥332	PchMOSFET
TLP351	¥100	4	¥400	MOSFET用フォトカプラ
16ピンソケット	¥10	2	¥20	Icソケット
18ピンソケット	¥10	1	¥10	Icソケット
EEUFM1V471	¥54	1	¥54	高リップル対応35V470uF電解コンデンサ
2SA950	¥10	2	¥20	トランジスタ
HRF502A	¥50	2	¥100	20V5Aショットキーダイオード
PIC16F88	¥200	1	¥200	制御マイコン
LED	¥5	2	¥10	
Molex,70553-0004	¥77	1	¥77	センサー用コネクタ
抵抗各種	¥1	19	¥19	
基板代	¥307	1	¥307	
		合計	¥1,811	

基板図-スクリーンショット



基板-スキャン

