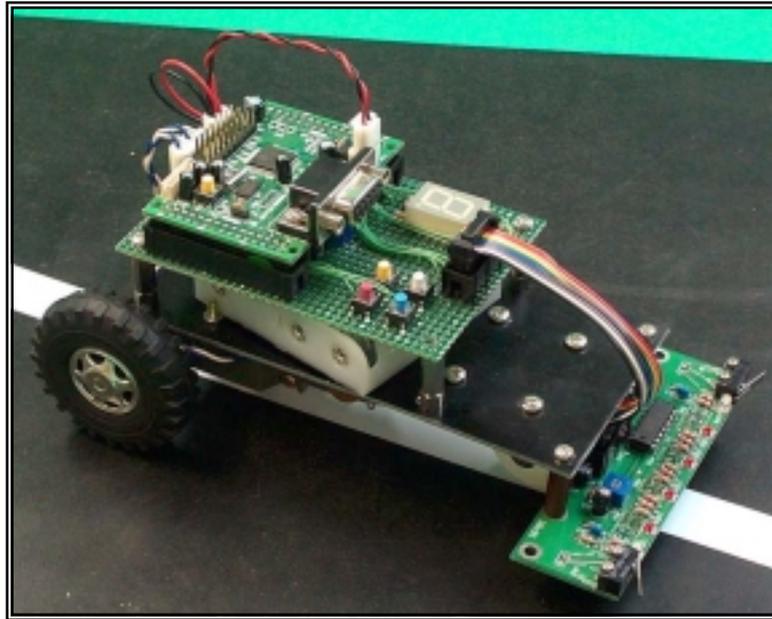


速度可変ライン走行カー組立てキット

# “F1 ミニ”

## 組立マニュアル

Version\_060904



|       |                    |       |
|-------|--------------------|-------|
| 第 1 章 | モータドライブ基板の組立て      | P. 1  |
| 第 2 章 | センサ基板              | P. 6  |
| 第 3 章 | センサ基板の自作           | P. 8  |
| 第 4 章 | ライン走行カーの組立て        | P. 10 |
| 第 5 章 | プログラムリスト“f1mini.c” | P. 17 |
| 付 録   | モータ制御の考え方, 等       | P. 27 |

### !!!! 注意 !!!!

- ◆ ‘F1 ミニ’ は組立てキットです。作る前に必ずマニュアルをしっかりと読んで、組立てのポイントを理解して下さい。特に、低学年の方が組立てるときは保護者や指導の先生に一読してもらい、組立て作業に間違いがないようにしましょう。
- ◆ 工具の使用には十分注意してください。特に、電気ドリルやニッパなどの刃物による怪我、ハンダゴテによる火傷など、事故に注意してください。ハンダゴテで火傷したときは直ぐに氷で冷やさなくてはなりません。
- ◆ 小さなお子様のそばでの組立ては避けた方がよいでしょう。小さな部品の飲み込みや散逸、ハンダゴテの接触などの危険性があります。
- ◆ 写真撮影には試作品を使用したため、外観が実際の製品とは異なる場合があります。

# 第1章

## モータドライブ基板の組立て

1-1. 工具の準備

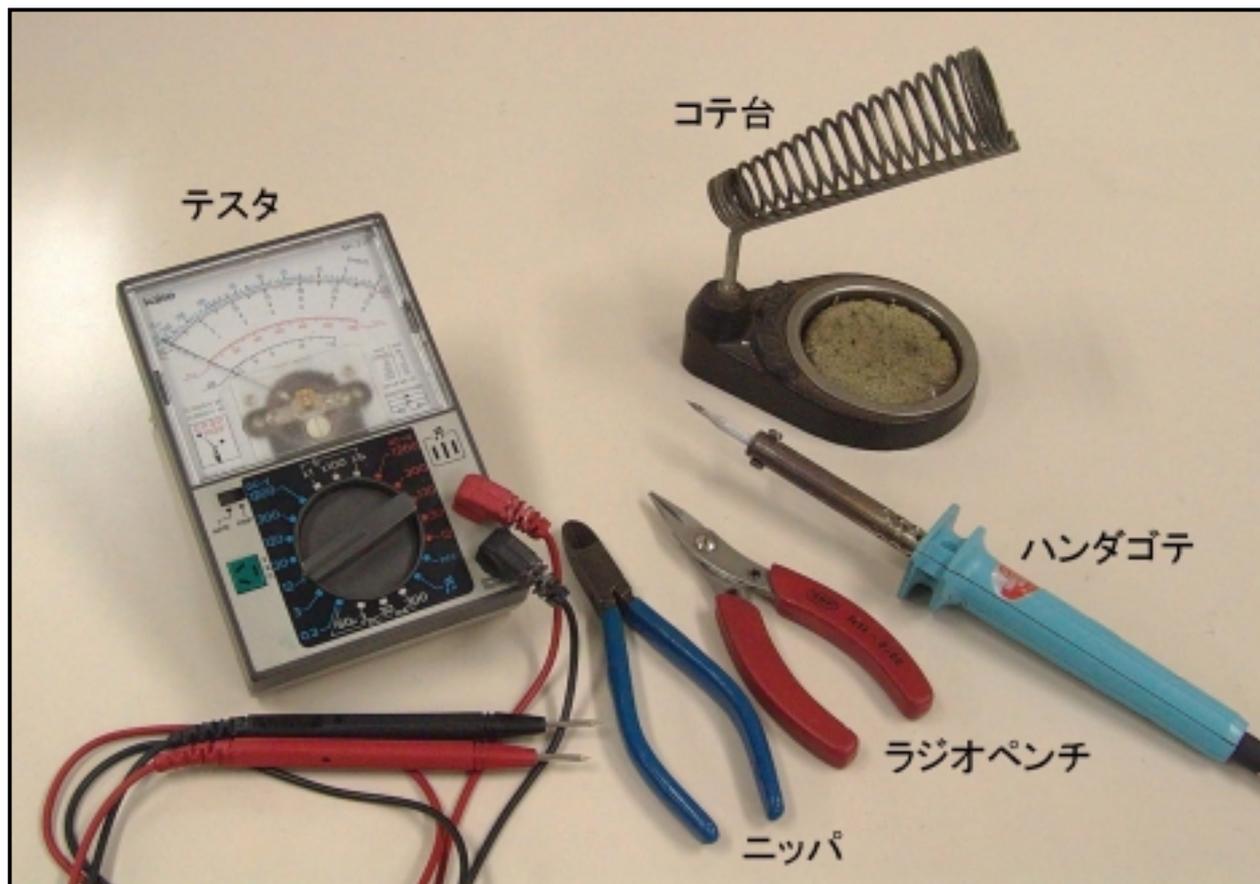
1-4. 組立て

1-2. 部品の確認

1-3. 回路図と実装図

### 1-1. 工具の準備

ハンダゴテは 30~40W 程度のものを用意して下さい。キットにハンダは入っていません。太さ 1mm 程度のハンダを用意して下さい。余分なリード線を切るためのニツパ、リード線を折り曲げるためのラジオペンチ、細かい配線に使用するピンセット、コテ台などを用意します。また、電圧や抵抗値を計るためにテスタが必要です。



## 1-2. 部品の確認

まず最初に、部品表と比較して部品が全てそろっているか確認しましょう。部品によっては相当品使用の場合もあります。(部品が足りないときは巻末記載の連絡先までお問い合わせください。)

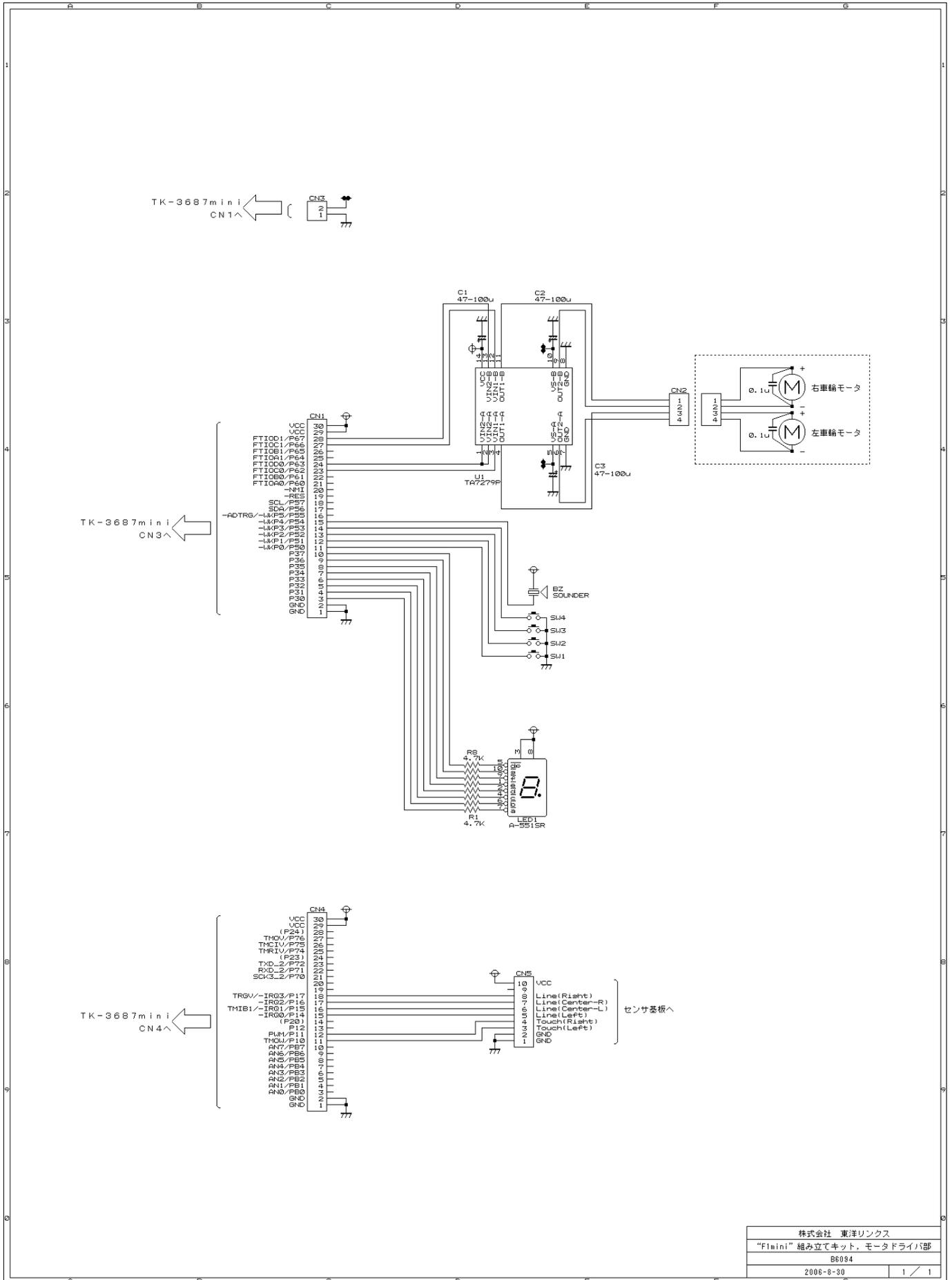
### “F1 mini”組み立てキット

| 部品番号                        | 型名, 規格        | メーカー                | 数量            | 付属数量 | 備考                 |
|-----------------------------|---------------|---------------------|---------------|------|--------------------|
| <b>1 ■モータドライバ基板</b>         |               |                     |               |      |                    |
| 2                           | U1&ICソケット     | TA7279P             | 東芝            | 1    | 1 ICソケットは縦に割って使用   |
| 3                           | LED1          | A-551SR             | PARA LIGHT    | 1    | 1 コモンアノード 7セグLED   |
| 4                           | サウンダ          | QMX-05<br>EPM121A0A | スター精密<br>メガセラ | 1    | 1 *1               |
| 5                           | SW1~4         | SKHHAK/AM/DC        | ALPS          | 4    | 4 *1               |
| 6                           | R1~8          | 4.7KΩ               |               | 8    | 8                  |
| 7                           | C1~3          | 47~100μF/16V        |               | 3    | 3                  |
| 8                           | CN1,4         | HIF3FC-30PA-2.54DSA | HRS           | 2    | 2 *1               |
| 9                           | CN5           | HIF3FC-10PA-2.54DSA | HRS           | 1    | 1 *1               |
| 10                          | CN2           | B4P-SHF-1AA         | JST           | 1    | 1                  |
| 11                          | CN3           | B2P-SHF-1AA         | JST           | 1    | 1                  |
| 12                          | ラッピングケーブル     | 1m                  |               | 1    | 1                  |
| 13                          | メッキ線          |                     |               | 0    | 0 ハンダ面結線用 *2       |
| 14                          | ユニバーサル基板      | B6093(95×72mm)      | 東洋リンクス        | 1    | 1                  |
| 15                          |               |                     |               |      |                    |
| <b>16 ■CPUボード(組立キットの場合)</b> |               |                     |               |      |                    |
| 17                          | CPUボード        | TK-3687mini         | 東洋リンクス        | 1    | 1 フラットパッケージ実装済み    |
| 18                          | REG1          | TA48M05F(S)         | 東芝            | 1    | 1                  |
| 19                          | X2            | 32.768KHz           |               | 1    | 1 サブクロック           |
| 20                          | D3            | 1SS133-T72          | ROHM          | 1    | 1 *1               |
| 21                          | LED1          | HLMP-6300#A04       | HP            | 1    | 1 *1               |
| 22                          | C3,19         | 47~100μF/16V        |               | 2    | 2                  |
| 23                          | C4,6,17,18,20 | 10μF/16V            |               | 5    | 5                  |
| 24                          | SW1           | SKHHAK/AM/DC        | ALPS          | 1    | 1 *1               |
| 25                          | CN1           | B2P-SHF-1AA         | JST           | 1    | 1 電源用              |
| 26                          | CN3,4         | HIF3FB-30DA-2.54DSA | HRS           | 2    | 2 基板間コネクタ, ハンダ面に実装 |
| 27                          | CN5           | D-Sub9pin           |               | 1    | 1                  |
| 28                          | JP1           | 2pin                |               | 1    | 1 ピンとソケットのセット      |
| 29                          |               |                     |               |      |                    |
| <b>30 ■機構部品</b>             |               |                     |               |      |                    |
| 31                          | ノイズ吸収用コンデンサ   | 0.1~0.47μF          |               | 2    | 2                  |
| 32                          | モータ接続用ケーブル    | 4芯×16cm             |               | 1    | 1                  |
| 33                          | ネジ            | 3-8mm               |               | 6    | 6                  |
| 34                          | ネジ            | 3-25mm              |               | 6    | 6                  |
| 35                          | ナット           | 3φ                  |               | 12   | 12                 |
| 36                          | スペーサ          | H=20mm              |               | 6    | 6                  |
| 37                          |               |                     |               |      |                    |
| <b>38 ■その他</b>              |               |                     |               |      |                    |
| 39                          | タイヤ           | 50φ                 |               | 2    | 2                  |
| 40                          | アルミ板          | 72×150×1.2mm        |               | 1    | 1                  |
| 41                          | キャスター         |                     |               | 1    | 1                  |
| 42                          | センサ基板         | B6062               | 東洋リンクス        | 1    | 1 完成品              |
| 43                          | センサ基板接続ケーブル   | 10芯×12cm~13cm       |               | 1    | 1 両端コネクタ付き         |
| 44                          | ギアボックス        | ツインモータギアボックス        | 田宮模型          | 1    | 1                  |
| 45                          | 二股電源ケーブル      |                     |               | 1    | 1                  |
| 46                          | 電池ボックス        |                     |               | 1    | 1 単3×4本用           |
| 47                          |               |                     |               |      |                    |

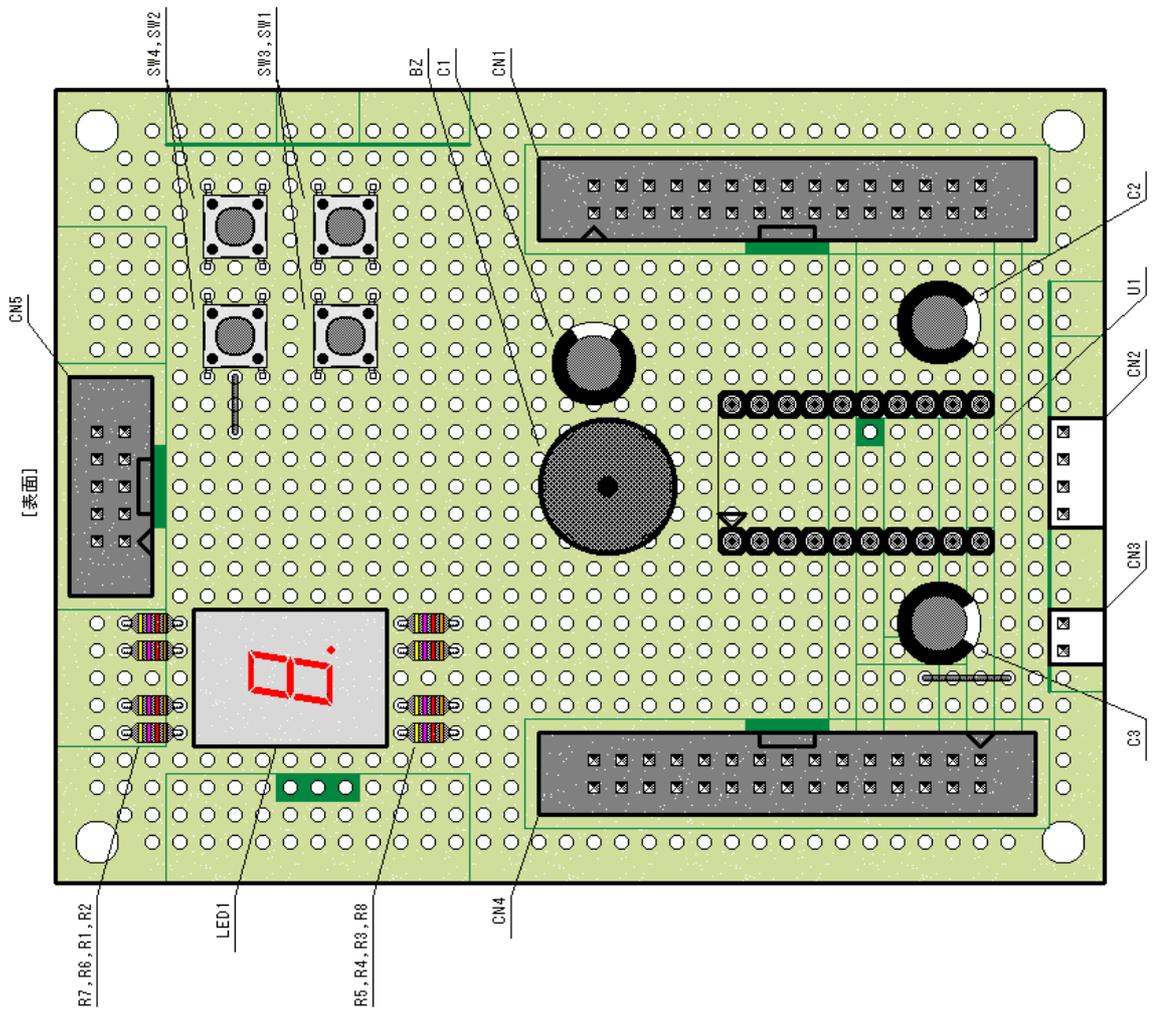
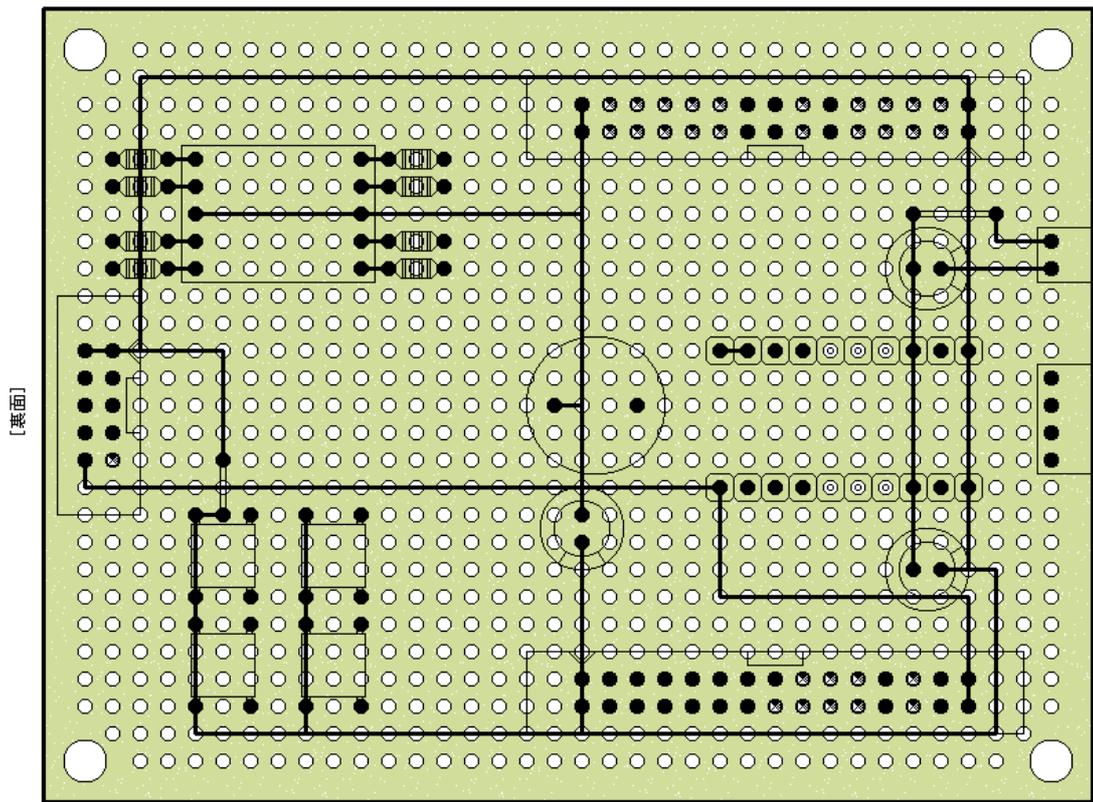
(\*1)相当品を使用することがあります。

(\*2)ラッピングケーブルの被覆をはがし2本をよじって使用します。また、抵抗やコンデンサの足も流用できます。

# 1-3. 回路図と実装図



|                          |       |
|--------------------------|-------|
| 株式会社 東洋リンクス              |       |
| "F1mini"組み立てキット、モータドライバ部 |       |
| B6034                    |       |
| 2006-8-30                | 1 / 1 |



## 1-4. 組立て

### ■ ドライバ基板

最初に、実装図の表面(部品面)を見てユニバーサル基板に部品を載せます。次に、実装図の裏面(半田面)を見てメッキ線での配線を済ませてください。最後に、回路図を見ながらラッピングケーブルで残りの配線を行ないます。

配線が終了したら、回路図どおり配線されているかもう一度確認して下さい。確認方法は、テスタで部品面の端子間の抵抗を測り、導通があるか、すなわち  $0\Omega$  か否かで判断します。また、半田付けがきちんと行なわれているか見ておきましょう。動かない原因の大部分は配線ミスと半田付け不良です。

### ■ CPU ボード

付属 CD の中に「TK-3687mini 組み立て手順書」があります。このマニュアルを見ながら TK-3687mini を組み立ててください。ただし、F1 ミニ付属の TK-3687mini の部品にはピンヘッダーは入っていません。

組み立てが終わったら、TK-3687mini にプログラムを書き込みます。付属 CD の中の 'f1mini.mot' を FDT を使って TK-3687mini にダウンロードしてください。FDT の基本的な使い方は「TK-3687mini 組み立て手順書」に記されています。これを参考にして下さい。

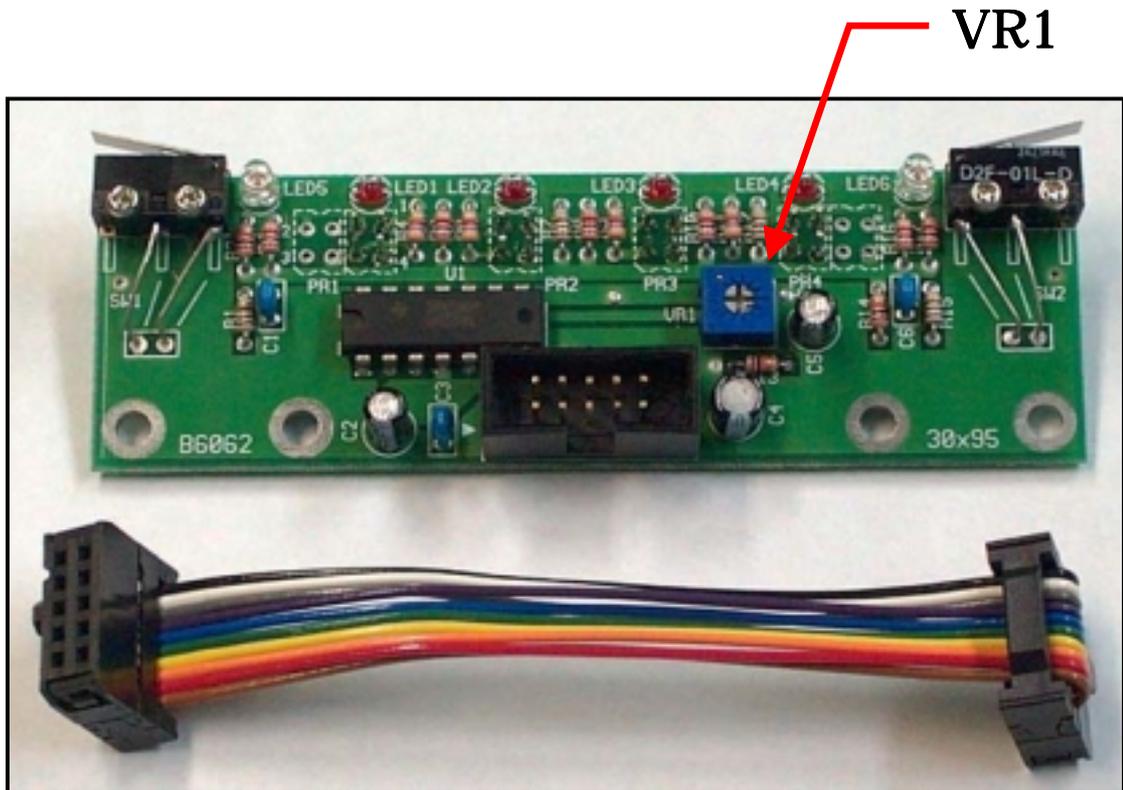
## 第2章

### センサ基板

- 2-1. 構成部品
- 2-2. 感度調整
- 2-3. 回路図

センサ基板は完成品です(ただし、センサー括調整)。そのままお使いください。

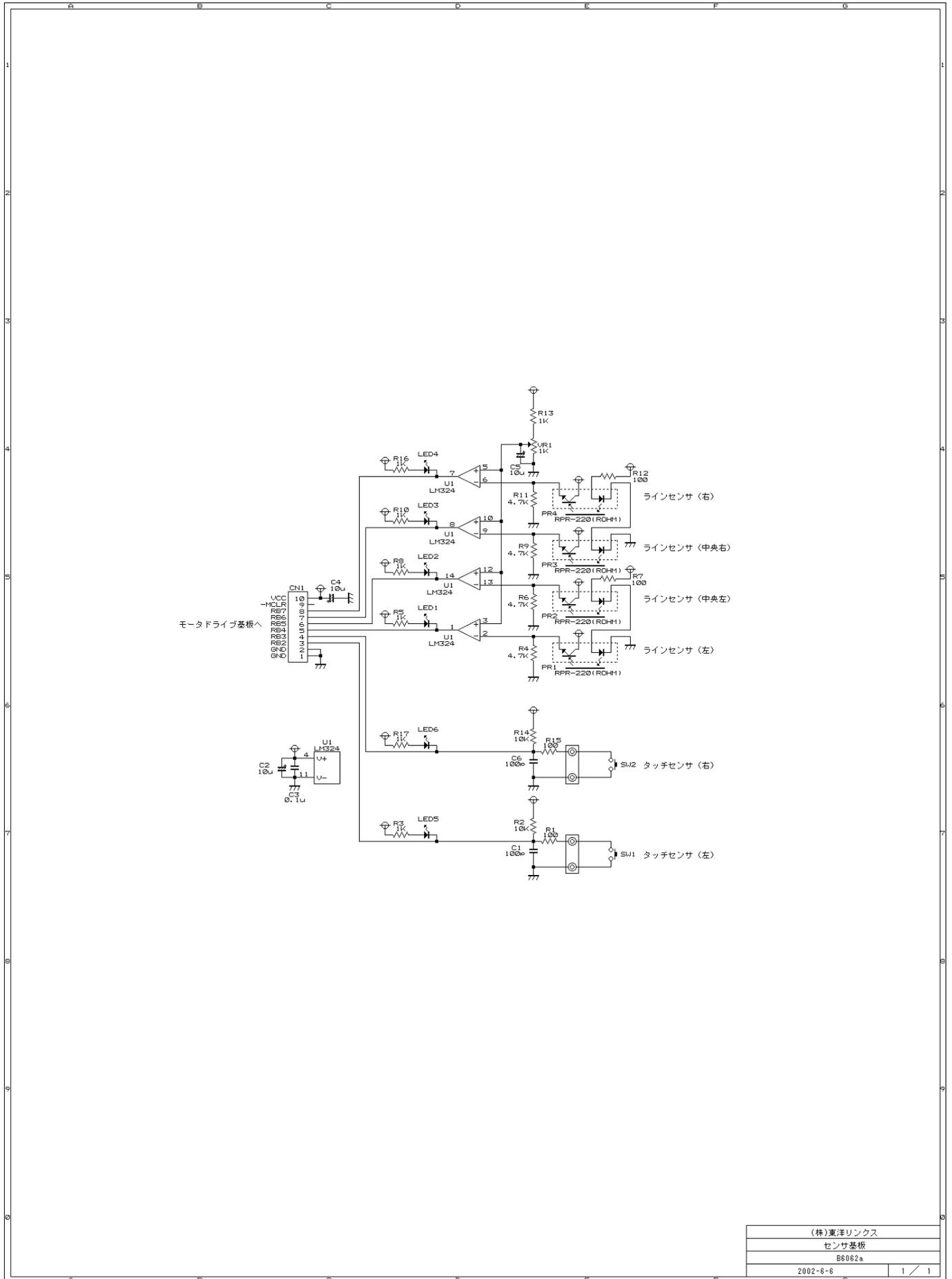
#### 2-1. 構成部品



#### 2-2. 感度調整

上の写真, 中央やや右の VR1 はセンサの感度調整を行うボリュームです。あらかじめ調整していますが, ラインのテープの種類によっては再調整が必要になるかもしれません。調整の仕方については 4 章で詳しく説明しますので位置だけ確認しておいて下さい。

## 2-3. 回路図



★注意: センサー一括調整で自作する場合は、センサの感度をあわせてください。★

# 第3章

## センサ基板の自作

3-1. 部品表

3-2. 回路図

### 3-1. 部品表

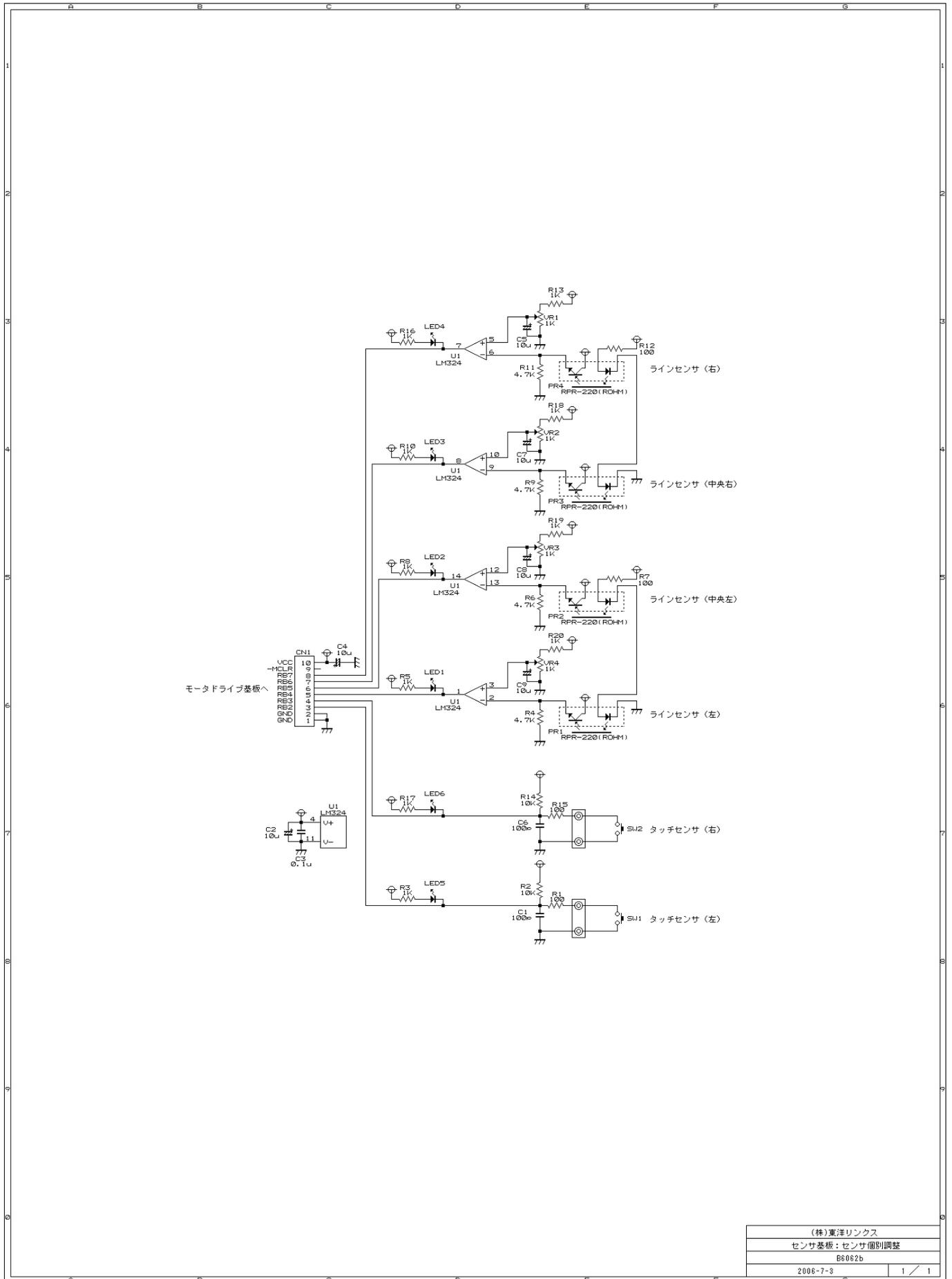
#### センサ基板(センサ個別調整)

|    | 部品番号                            | 型名, 規格              | メーカー  | 数量  | 付属数量 | 備考         |
|----|---------------------------------|---------------------|-------|-----|------|------------|
| 1  | <b>■センサ基板</b>                   |                     |       |     |      |            |
| 2  | U1                              | LM324N/AN           |       | 1   | 1    |            |
| 3  |                                 |                     |       |     |      |            |
| 4  | PR1~4                           | RPR-220             | ROHM  | 4   | 4    |            |
| 5  | LED1~4                          | BR2222S             | スタンレー | 4   | 4    | *1         |
| 6  | LED5, 6                         | TLRA123             | 東芝    | 2   | 2    | *1         |
| 7  |                                 |                     |       |     |      |            |
| 8  | R1,7,12,15                      | 100Ω                |       | 4   | 4    |            |
| 9  | R2,14                           | 10KΩ                |       | 2   | 2    |            |
| 10 | R3,5,8,10,13,16<br>R17,18,19,20 | 1KΩ                 |       | 10  | 10   |            |
| 11 | R4,6,9,11                       | 4.7KΩ               |       | 4   | 4    |            |
| 12 | VR1,2,3,4                       | RJ-6P1KΩ(102)       | COPAL | 4   | 4    | *1         |
| 13 |                                 |                     |       |     |      |            |
| 14 | C3                              | 0.1μF(セラミック)        |       | 1   | 1    |            |
| 15 | C1,6                            | 100pF(セラミック)        |       | 2   | 2    |            |
| 16 | C2,5,7,8,9                      | 10μF/16V            |       | 5   | 5    |            |
| 17 | C4                              | 47~100μF/16V        |       | 1   | 1    |            |
| 18 |                                 |                     |       |     |      |            |
| 19 | SW1,2                           | D2F-02L-D/D2F-L-D   | OMRON | 2   | 2    |            |
| 20 | ビス/ナット                          | 2φ, 10mm(SW1,2,固定用) |       | 2/2 | 2/2  |            |
| 21 |                                 |                     |       |     |      |            |
| 22 | CN1                             | HIF3FC-10PA-2.54DSA | HRS   | 1   | 1    | *1         |
| 23 |                                 |                     |       |     |      |            |
| 24 | ラッピングケーブル                       | 50cm                |       | 1   | 1    |            |
| 25 | メッキ線                            |                     |       | 0   | 0    | ハンダ面結線用 *2 |
| 26 |                                 |                     |       |     |      |            |
| 27 | ユニバーサル基板                        |                     |       | 1   | 1    |            |
| 28 |                                 |                     |       |     |      |            |

(\*1)相当品を使用することがあります。

(\*2)ラッピングケーブルの被覆をはがし2本をよじて使用します。また、抵抗やコンデンサの足も流用できます。

### 3-1. 回路図



|               |       |
|---------------|-------|
| (株)東洋リンクス     |       |
| センサ基板：センサ個別調整 |       |
| B6082b        |       |
| 2008-7-3      | 1 / 1 |

# 第4章

## ライン走行カーの組立て

4-1. 工具の準備

4-4. モータの結線と取り付け

4-7. センサ基板の取り付け

4-2. アルミ板の穴あけ

4-5. 前輪の取り付け

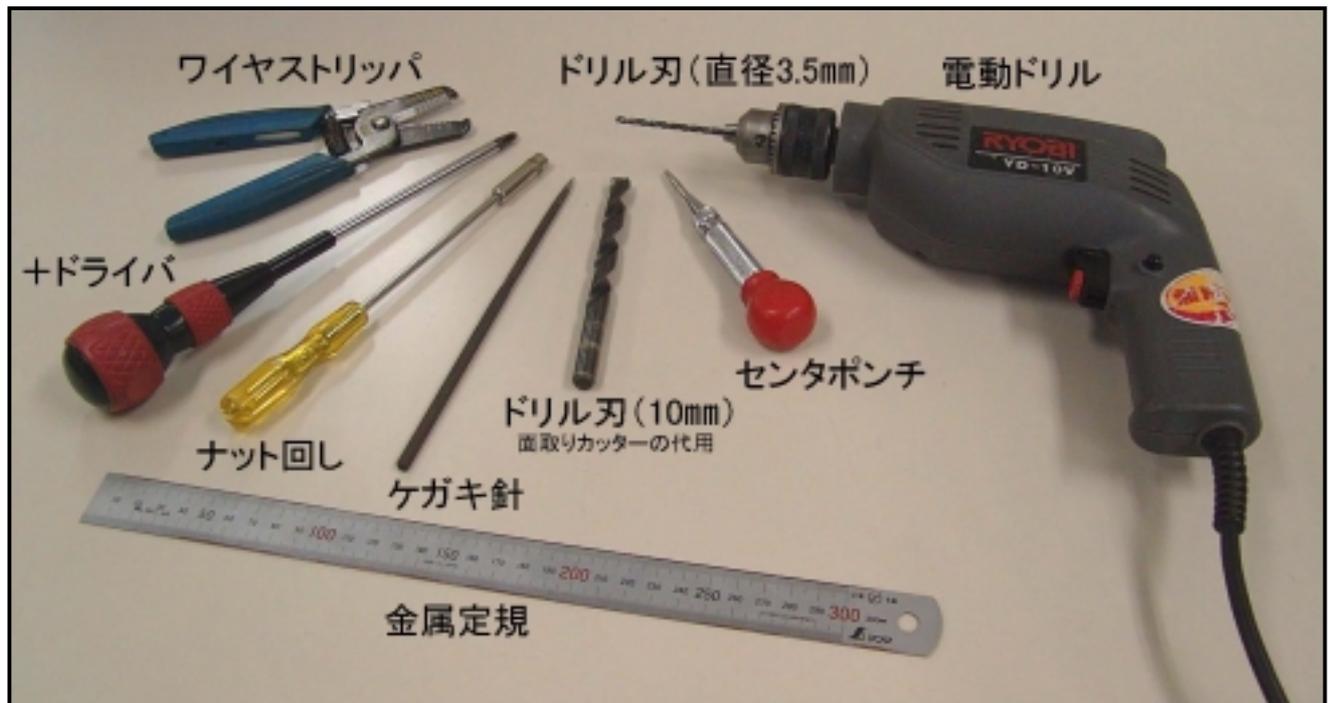
4-8. TK-3687miniと電池ボックスの取り付け

4-3. ツインモータギヤボックスの組立て

4-6. ドライバ基板の取り付け

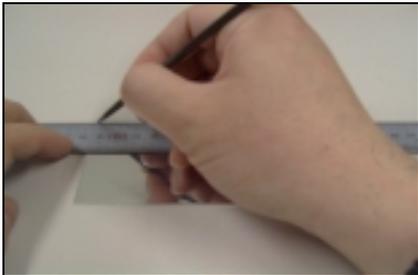
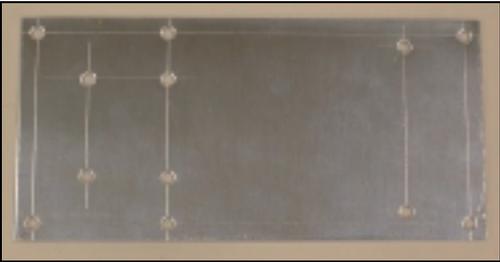
### 4-1. 工具の準備

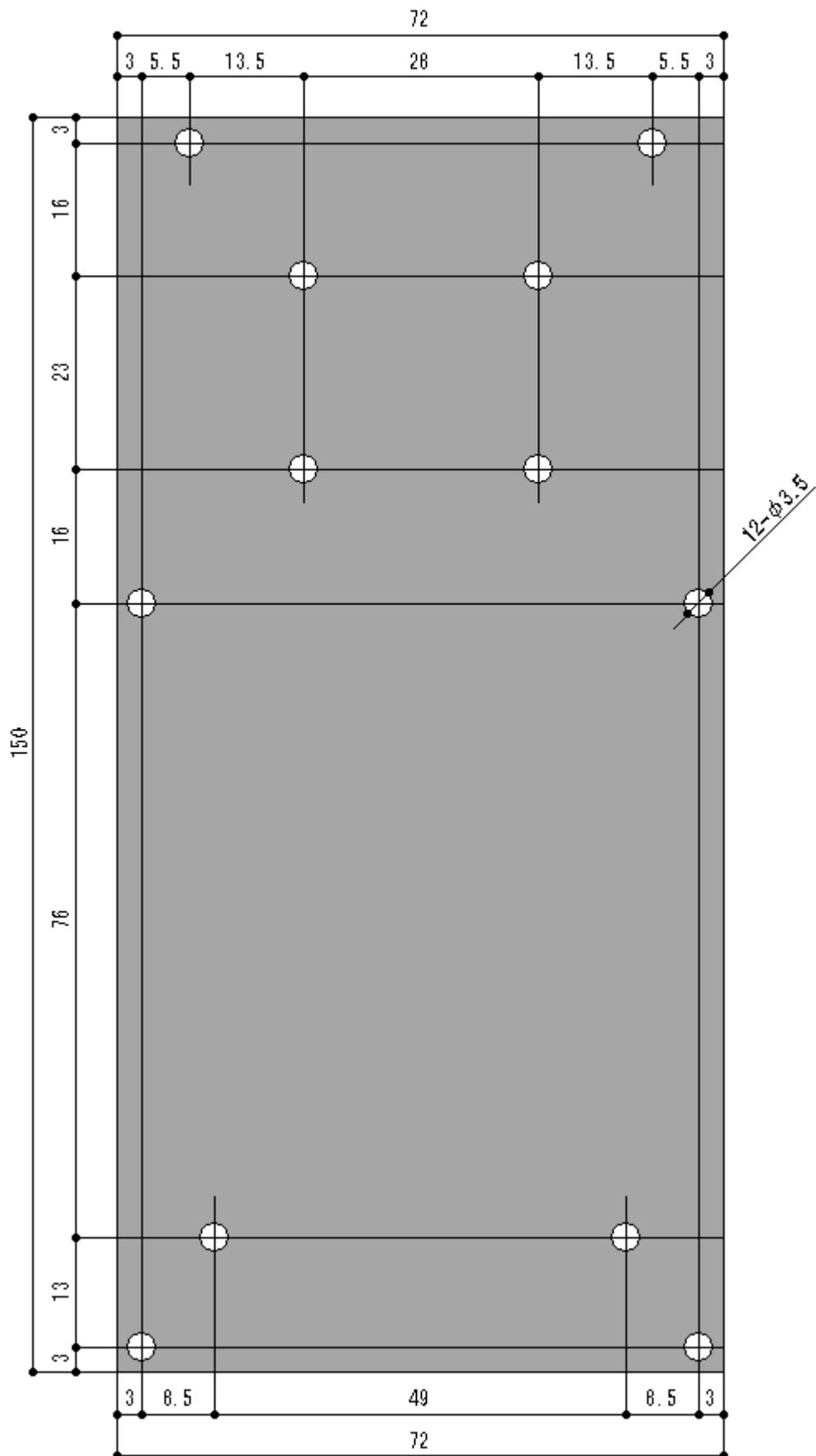
アルミ板の穴あけにケガキ針、金属定規、センタポンチ、電動ドリル、直径3.5mmの金属用のドリル刃、穴あけ後のバリ取りのために面取りカッター（少し大きめのドリルで代用可）を用意します。また、組立てには+ドライバとナット回しを使用します。ツインモータギヤボックスの組立てにはカッタ、またはニッパが必要です。モータへの結線のために30～40W程度のハンダゴテとワイヤストリッパを用意します。



## 4-2. アルミ板の穴あけ

12 ページの図面通りアルミ板に穴をあけます。

|   |  |   |
|---|--|---|
|    |    | <p>金属定規とケガキ針を使って穴あけ位置に印をつけます。</p>                 |
|    |    | <p>穴あけ位置が決まったらセンタポンチでくぼみをつけます。</p>                |
|   |  | <p>直径 3.5 mmのドリルを使って穴をあけます。</p>                   |
|  |  | <p>ドリルで穴をあけると、穴のまわりにバリが出るので、面取りカッター等でバリを取ります。</p> |

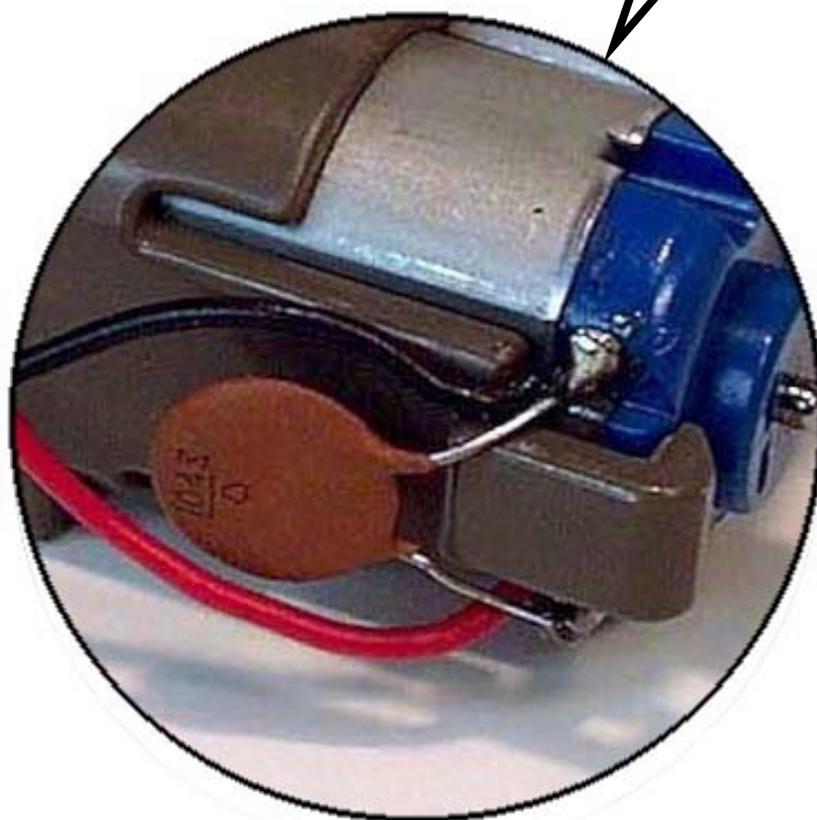
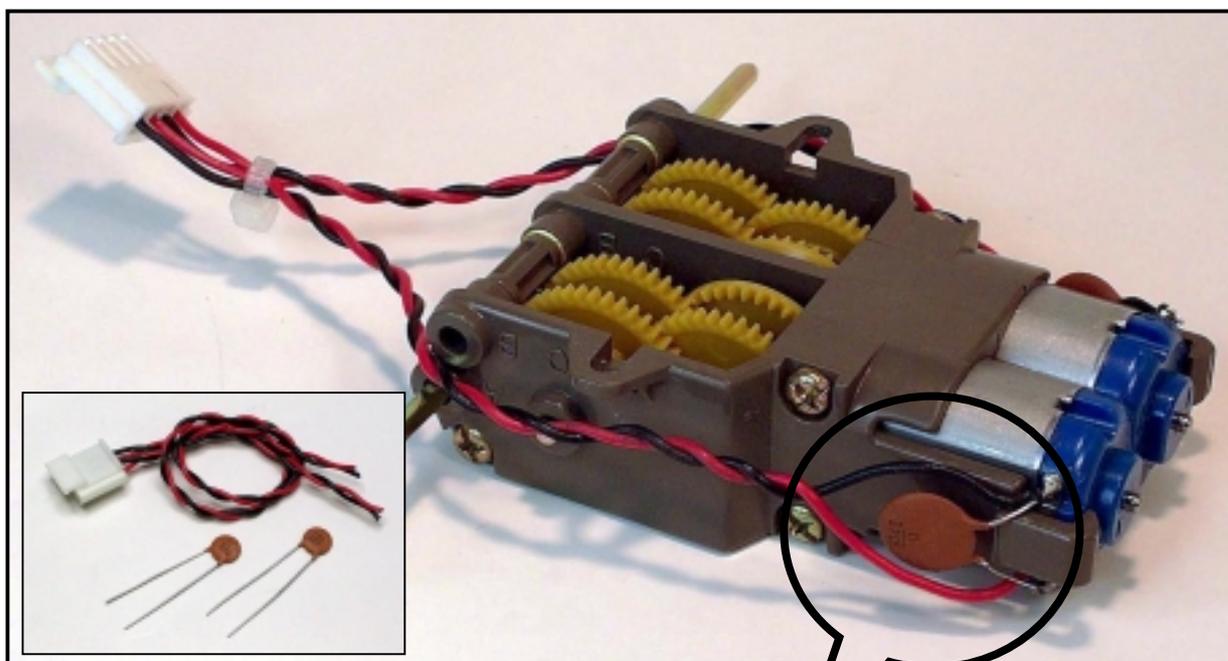


### 4-3. ツインモータギヤボックスの組立て

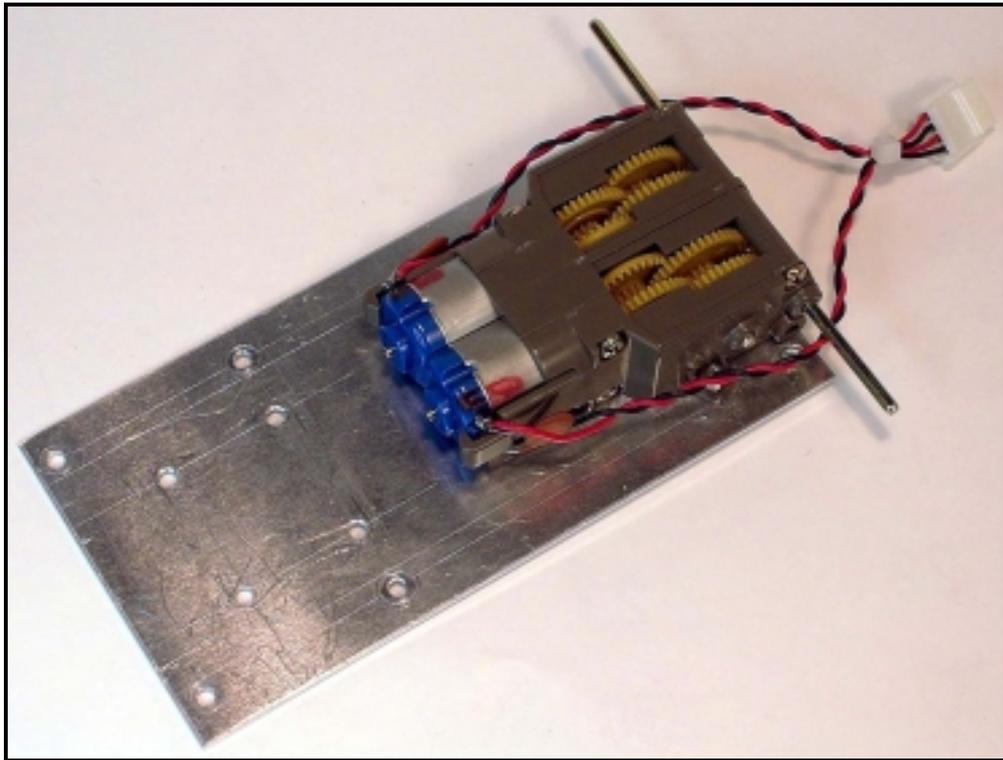
ツインモータギヤボックス内の組立て図を見て、標準仕様(Aタイプ)で組立ててください。

### 4-4. モータの結線と取り付け

4ピンコネクタがついているケーブルの被服をワイヤストリッパでむいてモータにハンダ付けします。コネクタの1, 2番が右車輪用のモータに, 3, 4番が左車輪用のモータにつながります。下の写真ではケーブルの色が赤黒になっていますが, 製品では白青になっているものもあります。その場合は, 白を赤, 青を黒として結線してください。ケーブルと一緒に付属のコンデンサ(0.1  $\mu$ F)もハンダ付けします。コンデンサはモータの発生ノイズを低減するのに必要ですから必ず取り付けてください。

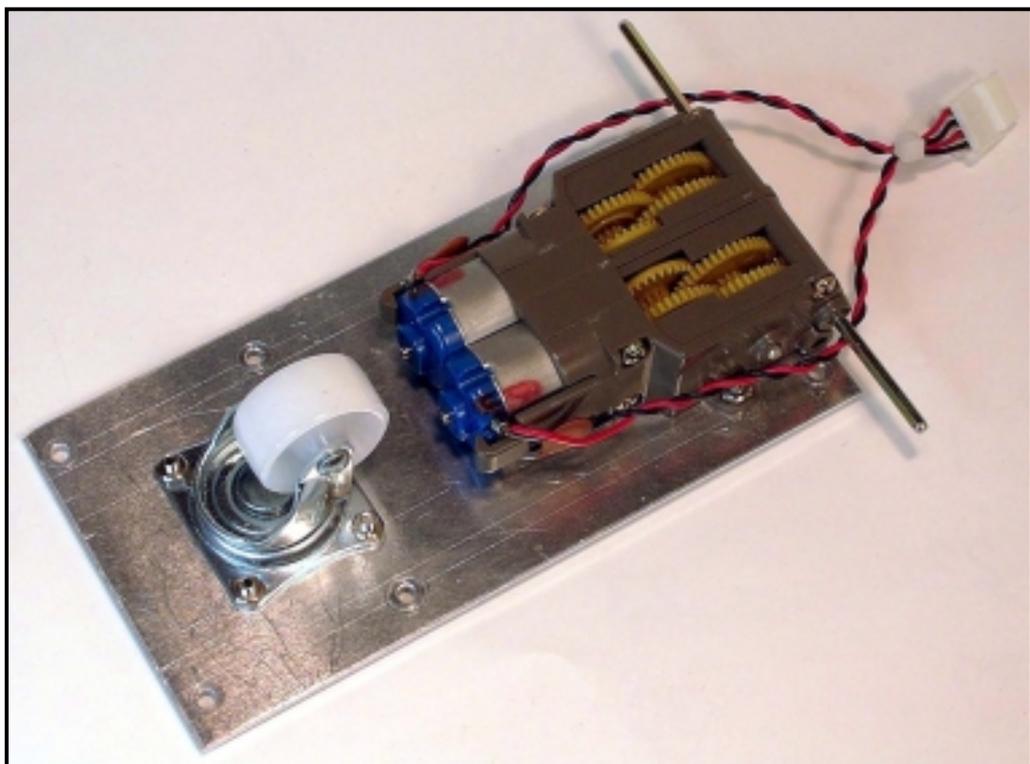


ハンダ付けが終わったら、アルミ板に8mmのネジをモータ側から差し込んで反対側からナットで取り付けます。



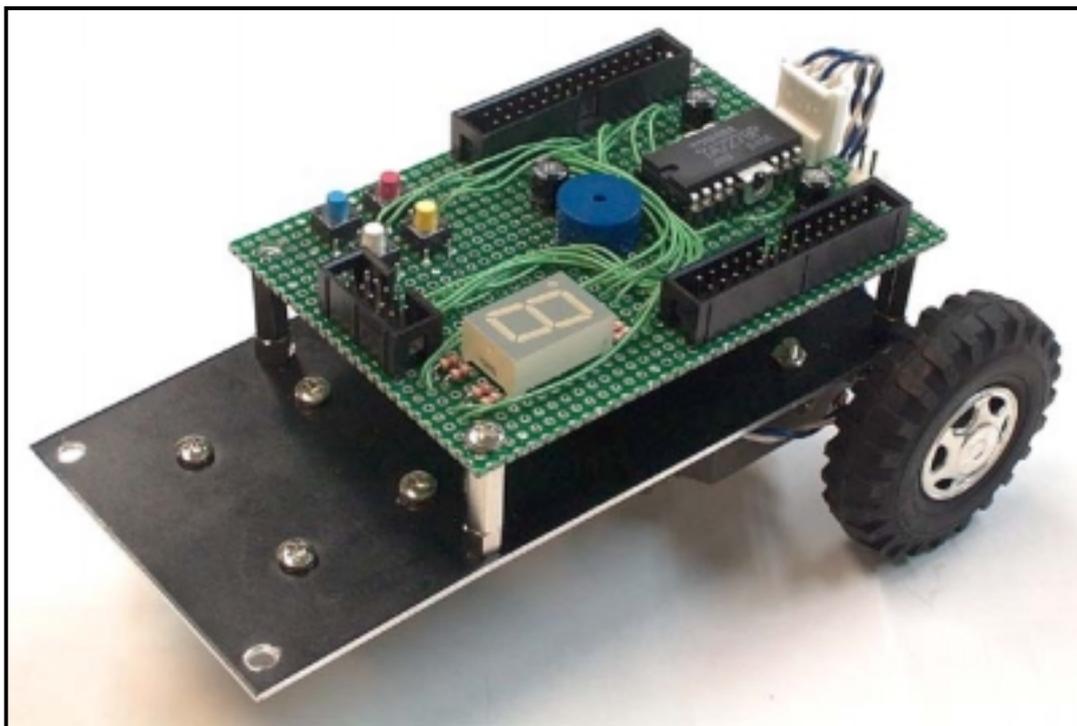
#### 4-5. 前輪の取り付け

キャスターをアルミ板に8mmのネジとナットで取り付けます。



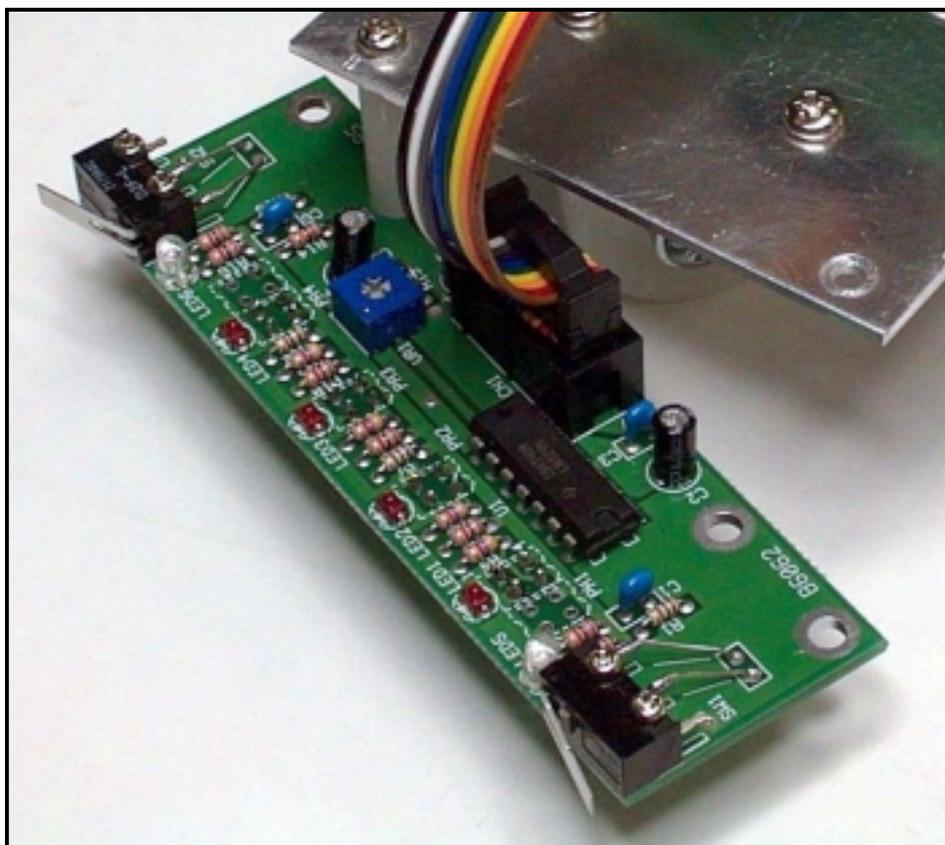
#### 4-6. ドライバ基板の取り付け

ドライバ基板は 20 mmのスペーサを介してアルミ板に 25mm のネジとナットで取り付けます。前後を間違えないようにして下さい。ドライバ基板を取り付けたあと、後輪をシャフトに挿します。



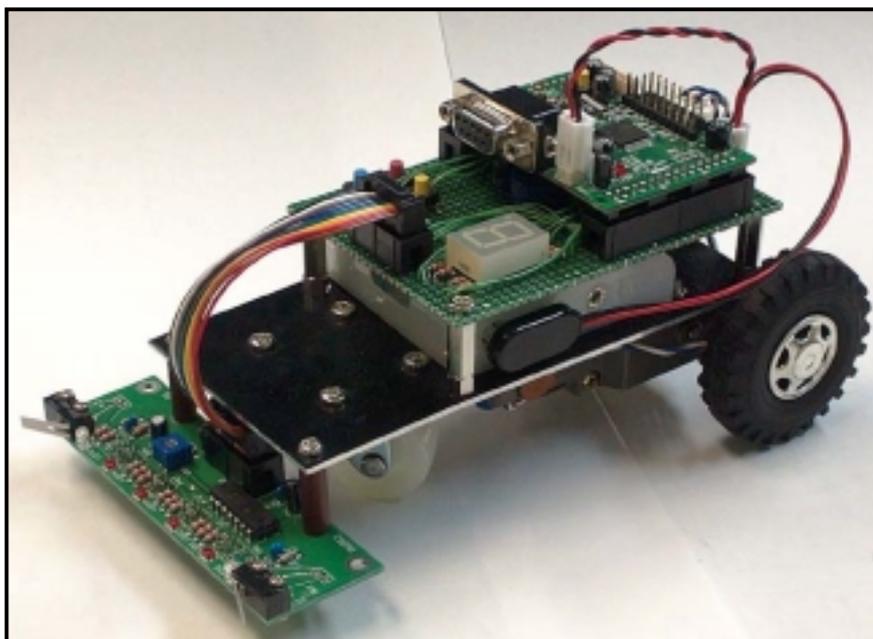
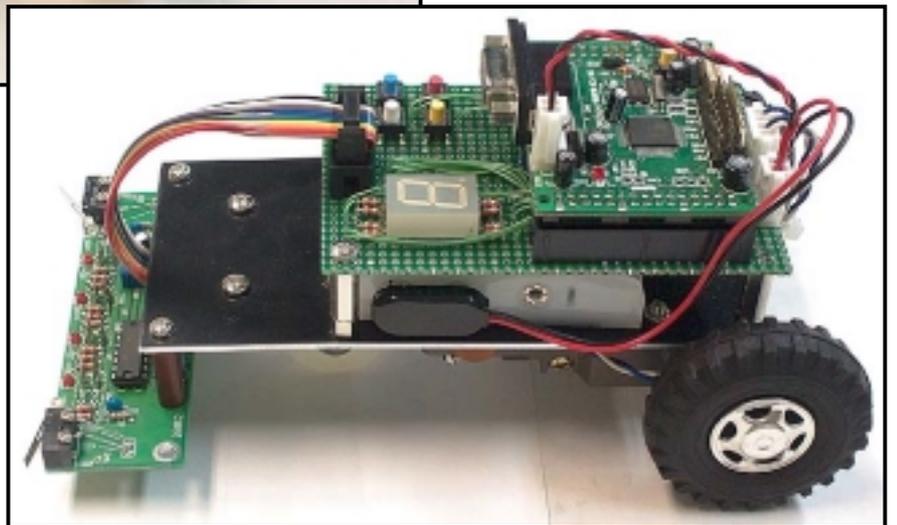
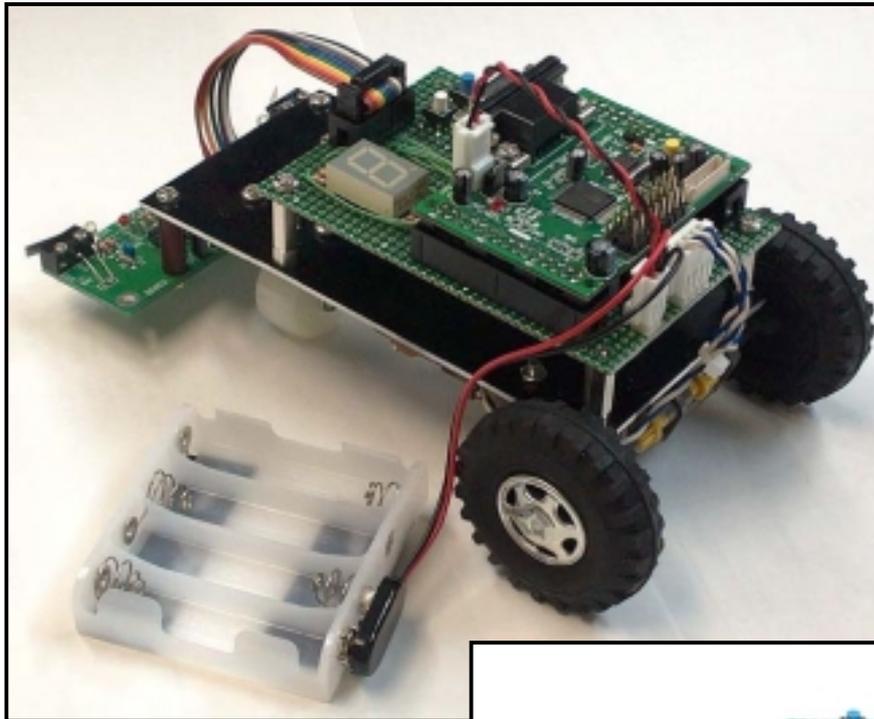
#### 4-7. センサ基板の取り付け

センサ基板は 20 mmのスペーサを介して 25mm のネジとナットでアルミ板に取り付けます。アルミ板に取り付ける前に付属の 10ピンコネクタをあらかじめ挿しておいてください。(アルミ板に取り付けてからではスペースの関係で挿さりません)



#### 4-8. TK-3687mini と電池ボックスの取り付け

TK-3687mini をドライバ基板に接続し、電池ボックスをアルミ板とドライバ基板の間に取り付けます。最後に二股電源ケーブルをドライバ基板とTK-3687mini に接続すれば、組み立ては完了です。(下の写真を参照)



# 第5章

## プログラムリスト“f1mini.c”

5-1. プログラムリスト“f1mini. c”

5-2. プログラムリスト“intprg. c”

### 5-1. プログラムリスト“f1mini. c”

```
/*
 *
 * FILE      : f1mini.c
 * DATE      : Wed, Apr 26, 2006
 * DESCRIPTION : Main Program
 * CPU TYPE  : H8/3687
 *
 * This file is programmed by TOYO-LINX Co.,Ltd. / yKikuchi
 *
 */
*****

プログラムの内容
*****
/*
電源オン後，SW2を押すとトレース開始。
黒字に白のラインをトレースする。
ラインから外れたときは後退，ラインを探して再びトレースする。
タッチスイッチオンで停止。
SW3を押しながら電源オン 回転方向チェックプログラム。
SW4を押しながら電源オン PWMチェックプログラム。
*/

*****

インクルードファイル
*****
#include <machine.h> // H8特有の命令を使う
#include "iodefine.h" // 内蔵I/Oのラベル定義

*****

定数の定義（直接指定）
*****
#define OK 0 //戻り値
#define NG -1 //戻り値

#define TL3LP 8 //左旋回,急,左モータ,%
#define TL3RP 20 //左旋回,急,右モータ,%
#define TL2LP 12 //左旋回,中,左モータ,%
#define TL2RP 20 //左旋回,中,右モータ,%
#define TL1LP 16 //左旋回,緩,左モータ,%
#define TL1RP 20 //左旋回,緩,右モータ,%

#define CW4LP 20 //直進,4速,左モータ,%
#define CW4RP 20 //直進,4速,右モータ,%
#define CW3LP 16 //直進,3速,左モータ,%
#define CW3RP 16 //直進,3速,右モータ,%
#define CW2LP 12 //直進,2速,左モータ,%
#define CW2RP 12 //直進,2速,右モータ,%
```

```

#define CW1LP      8      //直進,1速,左モータ,%
#define CW1RP      8      //直進,1速,右モータ,%

#define TR1LP     20     //右旋回,緩,左モータ,%
#define TR1RP     16     //右旋回,緩,右モータ,%
#define TR2LP     20     //右旋回,中,左モータ,%
#define TR2RP     12     //右旋回,中,右モータ,%
#define TR3LP     20     //右旋回,急,左モータ,%
#define TR3RP     8      //右旋回,急,右モータ,%

/*****
    定数エリアの定義(ROM)
*****/
// 7セグメントデータ変換テーブル -----
const unsigned char  SegTable[] = {0x3f,0x06,0x5b,0x4f,    //0,1,2,3
                                   0x66,0x6d,0x7d,0x07,    //4,5,6,7
                                   0x7f,0x6f,0x77,0x7c,    //8,9,A,B
                                   0x39,0x5e,0x79,0x71    //C,D,E,F
                                   };

/*****
    グローバル変数の定義とイニシャライズ(RAM)
*****/
// プログラム制御に関する変数 -----
unsigned char  ProgNo      = 0;  //プログラムナンバー
unsigned char  ProgStage  = 0;  //プログラムステージナンバー

// スイッチ入力に関係した変数 -----
unsigned char  SwData1     = 0;  //ファーストリード
unsigned char  SwData2     = 0;  //ダブルリードにより決定したデータ
unsigned char  SwData3     = 0;  //前回のダブルリードで決定したデータ
unsigned char  SwData4     = 0;  //0 1に変化したデータ
unsigned char  SwStatus    = 0;  //スイッチ入カステータス
                                   // 0:ファーストリード
                                   // 1:ダブルリード

// ライントレースに関する変数 -----
unsigned char  LineData    = 0;  //ライン情報
unsigned char  OldLineData = 0;  //前回のライン情報

//左旋回,急,左モータ
unsigned int  TurnLeft3L   = 0xffff - ((unsigned long)0xffff * TL3LP / 100);
//左旋回,急,右モータ
unsigned int  TurnLeft3R   = 0xffff - ((unsigned long)0xffff * TL3RP / 100);
//左旋回,中,左モータ
unsigned int  TurnLeft2L   = 0xffff - ((unsigned long)0xffff * TL2LP / 100);
//左旋回,中,右モータ
unsigned int  TurnLeft2R   = 0xffff - ((unsigned long)0xffff * TL2RP / 100);
//左旋回,緩,左モータ
unsigned int  TurnLeft1L   = 0xffff - ((unsigned long)0xffff * TL1LP / 100);
//左旋回,緩,右モータ
unsigned int  TurnLeft1R   = 0xffff - ((unsigned long)0xffff * TL1RP / 100);

//直進,4速,左モータ
unsigned int  Cw4L         = 0xffff - ((unsigned long)0xffff * CW4LP / 100);
//直進,4速,右モータ
unsigned int  Cw4R         = 0xffff - ((unsigned long)0xffff * CW4RP / 100);
//直進,3速,左モータ

```

```

unsigned int Cw3L      = 0xffff - ((unsigned long)0xffff * CW3LP / 100);
//直進,3速,右モータ
unsigned int Cw3R      = 0xffff - ((unsigned long)0xffff * CW3RP / 100);
//直進,2速,左モータ
unsigned int Cw2L      = 0xffff - ((unsigned long)0xffff * CW2LP / 100);
//直進,2速,右モータ
unsigned int Cw2R      = 0xffff - ((unsigned long)0xffff * CW2RP / 100);
//直進,1速,左モータ
unsigned int Cw1L      = 0xffff - ((unsigned long)0xffff * CW1LP / 100);
//直進,1速,右モータ
unsigned int Cw1R      = 0xffff - ((unsigned long)0xffff * CW1RP / 100);

//右旋回,緩,左モータ
unsigned int TurnRight1L = 0xffff - ((unsigned long)0xffff * TR1LP / 100);
//右旋回,緩,右モータ
unsigned int TurnRight1R = 0xffff - ((unsigned long)0xffff * TR1RP / 100);
//右旋回,中,左モータ
unsigned int TurnRight2L = 0xffff - ((unsigned long)0xffff * TR2LP / 100);
//右旋回,中,右モータ
unsigned int TurnRight2R = 0xffff - ((unsigned long)0xffff * TR2RP / 100);
//右旋回,急,左モータ
unsigned int TurnRight3L = 0xffff - ((unsigned long)0xffff * TR3LP / 100);
//右旋回,急,右モータ
unsigned int TurnRight3R = 0xffff - ((unsigned long)0xffff * TR3RP / 100);

/*****
関数の定義
*****/
void ccw1(void); //後退,1速
void ccw2(void); //後退,2速
void ccw3(void); //後退,3速
void ccw4(void); //後退,4速
void chk_cw_ccw(void); //回転方向確認プログラム
void chk_pwm(void); //PWM確認プログラム
void cw1(void); //前進,1速
void cw2(void); //前進,2速
void cw3(void); //前進,3速
void cw4(void); //前進,4速
void init_io(void); //I/Oポートイニシャライズ
void init_tmb1(void); //タイマB1イニシャライズ
void init_tmz(void); //タイマZイニシャライズ
void intprog_tmb1(void); //タイマB1割込み
void main(void); //メインプログラム
void stop(void); //停止
void switch_in(void); //スイッチ入力
void trace(void); //ライントレース
void turn_left_1(void); //左旋回(大)
void turn_left_2(void); //左旋回(中)
void turn_left_3(void); //左旋回(小)
void turn_right_1(void); //右旋回(大)
void turn_right_2(void); //右旋回(中)
void turn_right_3(void); //右旋回(小)

/*****
メインプログラム
*****/
void main(void)
{

```

```

// イニシャライズ -----
init_io();
init_tmb1();
init_tmz();

// チェックルーチンにジャンプ -----
if (IO.PDR5.BIT.B2==0){
    while(1) {chk_cw_ccw();}
}
if (IO.PDR5.BIT.B3==0){
    while(1) {chk_pwm();}
}

// メインループ -----
while(1){
    trace();
}
}

/*****
    ライントレース
*****/
void trace(void)
{
    switch (ProgStage){
        // スタート待ち -----
        case 0:
            if ((SwData4 & 0x02)!=0){ //SW2が押されたらラインレース走行スタート
                ProgStage = 1;
                SwData4 = 0;
            }
            IO.PDR3.BYTE = ~SegTable[0];
            break;
        // ライントレース走行 -----
        case 1:
            LineData = ~IO.PDR1.BYTE & 0xf0;
            switch (LineData){
                case 0x80: //左寄り(大) 右旋回(大)
                    turn_right_3();
                    IO.PDR3.BYTE = ~SegTable[1];
                    break;
                case 0xc0: //左寄り(中) 右旋回(中)
                    turn_right_2();
                    IO.PDR3.BYTE = ~SegTable[2];
                    break;
                case 0x40: //左寄り(小) 右旋回(小)
                    turn_right_1();
                    IO.PDR3.BYTE = ~SegTable[3];
                    break;
                case 0x60: //中央 前進
                    cw4();
                    IO.PDR3.BYTE = ~SegTable[4];
                    break;
                case 0x20: //右寄り(小) 左旋回(小)
                    turn_left_1();
                    IO.PDR3.BYTE = ~SegTable[5];
                    break;
                case 0x30: //右寄り(中) 左旋回(中)

```

```

        turn_left_2();
        IO.PDR3.BYTE = ~SegTable[6];
        break;
    case 0x10: //右寄り(大) 左旋回(大)
        turn_left_3();
        IO.PDR3.BYTE = ~SegTable[7];
        break;
    default: //それ以外 後退
        ccw2();
        IO.PDR3.BYTE = ~SegTable[8];
        break;
    }

    if ((IO.PDR1.BIT.B0==0)|| (IO.PDR1.BIT.B1==0)){
        stop(); //ぶつかったら停止
        ProgStage = 0; //スイッチ待ちに戻る
    }

    break;
}
}

/*****
    モータスピードセット
*****/
// 前進,1速 -----
void cw1(void)
{
    TZ0.GRC = Cw1L;
    TZ0.GRD = 0xffff;
    TZ1.GRC = Cw1R;
    TZ1.GRD = 0xffff;
}

// 前進,2速 -----
void cw2(void)
{
    TZ0.GRC = Cw2L;
    TZ0.GRD = 0xffff;
    TZ1.GRC = Cw2R;
    TZ1.GRD = 0xffff;
}

// 前進,3速 -----
void cw3(void)
{
    TZ0.GRC = Cw3L;
    TZ0.GRD = 0xffff;
    TZ1.GRC = Cw3R;
    TZ1.GRD = 0xffff;
}

// 前進,4速 -----
void cw4(void)
{
    TZ0.GRC = Cw4L;
    TZ0.GRD = 0xffff;
    TZ1.GRC = Cw4R;

```

```

    TZ1.GRD = 0xffff;
}

// 後退,1速 -----
void ccw1(void)
{
    TZ0.GRC = 0xffff;
    TZ0.GRD = Cw1L;
    TZ1.GRC = 0xffff;
    TZ1.GRD = Cw1R;
}

// 後退,2速 -----
void ccw2(void)
{
    TZ0.GRC = 0xffff;
    TZ0.GRD = Cw2L;
    TZ1.GRC = 0xffff;
    TZ1.GRD = Cw2R;
}

// 後退,3速 -----
void ccw3(void)
{
    TZ0.GRC = 0xffff;
    TZ0.GRD = Cw3L;
    TZ1.GRC = 0xffff;
    TZ1.GRD = Cw3R;
}

// 後退,4速 -----
void ccw4(void)
{
    TZ0.GRC = 0xffff;
    TZ0.GRD = Cw4L;
    TZ1.GRC = 0xffff;
    TZ1.GRD = Cw4R;
}

// 右旋回(小) -----
void turn_right_1(void)
{
    TZ0.GRC = TurnRight1L;
    TZ0.GRD = 0xffff;
    TZ1.GRC = TurnRight1R;
    TZ1.GRD = 0xffff;
}

// 右旋回(中) -----
void turn_right_2(void)
{
    TZ0.GRC = TurnRight2L;
    TZ0.GRD = 0xffff;
    TZ1.GRC = TurnRight2R;
    TZ1.GRD = 0xffff;
}

// 右旋回(大) -----

```

```

void turn_right_3(void)
{
    TZ0.GRC = TurnRight3L;
    TZ0.GRD = 0xffff;
    TZ1.GRC = TurnRight3R;
    TZ1.GRD = 0xffff;
}

// 左旋回 (小) -----
void turn_left_1(void)
{
    TZ0.GRC = TurnLeft1L;
    TZ0.GRD = 0xffff;
    TZ1.GRC = TurnLeft1R;
    TZ1.GRD = 0xffff;
}

// 左旋回 (中) -----
void turn_left_2(void)
{
    TZ0.GRC = TurnLeft2L;
    TZ0.GRD = 0xffff;
    TZ1.GRC = TurnLeft2R;
    TZ1.GRD = 0xffff;
}

// 左旋回 (大) -----
void turn_left_3(void)
{
    TZ0.GRC = TurnLeft3L;
    TZ0.GRD = 0xffff;
    TZ1.GRC = TurnLeft3R;
    TZ1.GRD = 0xffff;
}

// 停止 -----
void stop()
{
    TZ0.GRC = 0xffff;
    TZ0.GRD = 0xffff;
    TZ1.GRC = 0xffff;
    TZ1.GRD = 0xffff;
}

/*****
  回転方向の確認プログラム
  *****/
void chk_cw_ccw(void)
{
    IO.PCR6 = 0xcc; //ポート6, P62, 63, 66, 67出力, P60, 61, 64, 65入力

    if (IO.PDR5.BIT.B0==0) {IO.PDR6.BIT.B7 = 1;} //SW1-On : 右後退
    else {IO.PDR6.BIT.B7 = 0;}

    if (IO.PDR5.BIT.B1==0) {IO.PDR6.BIT.B6 = 1;} //SW2-On : 右前進
    else {IO.PDR6.BIT.B6 = 0;}

    if (IO.PDR5.BIT.B2==0) {IO.PDR6.BIT.B3 = 1;} //SW3-On : 左後退

```

```

else                                     {IO.PDR6.BIT.B3 = 0;}

if (IO.PDR5.BIT.B3==0)   {IO.PDR6.BIT.B2 = 1;}   //SW4-0n : 左前進
else                       {IO.PDR6.BIT.B2 = 0;}
}

/*****
PWMの確認プログラム
*****/
void chk_pwm(void)
{
    unsigned long i;

    if (IO.PDR5.BIT.B0==0){           //SW1-0n : 右減速
        if (TZ1.GRC==0xffff){
            if (TZ1.GRD!=0x0000) {TZ1.GRD = TZ1.GRD - 1;}
        }
        else{
            TZ1.GRC = TZ1.GRC + 1;
        }
    }

    if (IO.PDR5.BIT.B1==0){           //SW2-0n : 右加速
        if (TZ1.GRD==0xffff){
            if (TZ1.GRC!=0x0000) {TZ1.GRC = TZ1.GRC - 1;}
        }
        else{
            TZ1.GRD = TZ1.GRD + 1;
        }
    }

    if (IO.PDR5.BIT.B2==0){           //SW3-0n : 左減速
        if (TZ0.GRC==0xffff){
            if (TZ0.GRD!=0x0000) {TZ0.GRD = TZ0.GRD - 1;}
        }
        else{
            TZ0.GRC = TZ0.GRC + 1;
        }
    }

    if (IO.PDR5.BIT.B3==0){           //SW4-0n : 左加速
        if (TZ0.GRD==0xffff){
            if (TZ0.GRC!=0x0000) {TZ0.GRC = TZ0.GRC - 1;}
        }
        else{
            TZ0.GRD = TZ0.GRD + 1;
        }
    }

    IO.PDR3.BYTE = ~SegTable[TZ1.GRC / 0x1000]; //右前進の状態を表示する
// IO.PDR3.BYTE = ~SegTable[TZ1.GRD / 0x1000]; //右後退の状態を表示する
// IO.PDR3.BYTE = ~SegTable[TZ0.GRC / 0x1000]; //右前進の状態を表示する
// IO.PDR3.BYTE = ~SegTable[TZ0.GRD / 0x1000]; //右後退の状態を表示する

    for (i=0; i<200; i++){
}

/*****

```

## I/Oポート イニシャライズ

\*\*\*\*\*/

```
void init_io(void)
```

```
{
    IO.PMR1.BYTE = 0x00; //ポート1,汎用入出力ポート
    IO.PUCR1.BYTE = 0x00; //ポート1,内蔵ブルアップオフ
    IO.PCR1      = 0x00; //ポート1,P10-17入力

    IO.PDR3.BYTE = 0xff; //ポート3,初期出力
    IO.PCR3      = 0xff; //ポート3,P30-37出力

    IO.PMR5.BYTE = 0x00; //ポート5,汎用入出力ポート
    IO.PUCR5.BYTE = 0x0f; //ポート5,P50-53内蔵ブルアップオン
    IO.PDR5.BYTE = 0x10; //ポート5,初期出力
    IO.PCR5      = 0x10; //ポート5,P54出力,P50-53,P55-57入力

    IO.PCR6      = 0x00; //ポート6,P60-67入力
}
```

\*\*\*\*\*

## タイマZ イニシャライズ

\*\*\*\*\*/

```
void init_tmz(void)
```

```
{
    TZ.TSTR.BYTE = 0x00; //TCNT0,1 停止

    TZ0.TCR.BYTE = 0x20; //GRAのコンペアマッチでTCNT=0, /1
    TZ1.TCR.BYTE = 0x20; //GRAのコンペアマッチでTCNT=0, /1

    TZ.TPMR.BYTE = 0x66; //FTI0C0,FTI0D0,FTI0C1,FTI0D1 PWMモード
    TZ.TOCR.BYTE = 0x00; //初期出力,All Low

    TZ0.POCR.BYTE = 0x06; //FTI0C0,FTI0D0 出力ハイアクティブ
    TZ1.POCR.BYTE = 0x06; //FTI0C1,FTI0D1 出力ハイアクティブ

    TZ0.GRA = 0xfffe; //PWM 周期(65534*(1/20MHz)=3.2767ms)
    TZ1.GRA = 0xfffe; //PWM 周期(65534*(1/20MHz)=3.2767ms)
    TZ0.GRC = 0xffff; //FTI0C0=Low
    TZ1.GRC = 0xffff; //FTI0C1=Low
    TZ0.GRD = 0xffff; //FTI0D0=Low
    TZ1.GRD = 0xffff; //FTI0D1=Low

    TZ.TOER.BYTE = 0x33; //FTI0C0,DTI0D0,FTI0C1,FTI0D1 出力許可

    TZ0.TCNT = 0x0000; //TCNT0=0
    TZ1.TCNT = 0x0000; //TCNT1=0
    TZ.TSTR.BYTE = 0x03; //TCNT0 カウントスタート
}
```

\*\*\*\*\*

## タイマB1 イニシャライズ

\*\*\*\*\*/

```
void init_tmb1(void)
```

```
{
    TB1.TMB1.BYTE = 0xf9; //オートリロード,内部クロック /2048
    TB1.TLB1      = 0-97; //周期=10ms(100Hz)
    IRR2.BIT.IRRTB1 = 0; //タイマB1割込み要求フラグ クリア
    IENR2.BIT.IENTB1 = 1; //タイマB1割込み要求イネーブル
}
```

```

}

/*****
   タイマB1 割込み(10ms)
*****/
#pragma regsave (intprog_tmb1)
void intprog_tmb1(void)
{
    //タイマB1割込み要求フラグ クリア
    IRR2.BIT.IRRTB1 = 0;

    //スイッチ入力
    switch_in();
}

/*****
   スイッチ入力
*****/
void switch_in(void)
{
    switch(SwStatus){
        case 0:
            SwData1 = ~IO.PDR5.BYTE & 0x0f;
            if (SwData1!=0) {SwStatus = 1;}
            else          {SwData2 = SwData3 =0;}
            break;
        case 1:
            if (SwData1==(~IO.PDR5.BYTE & 0x0f)){
                SwData2 = SwData1;
                SwData4 = SwData4 | (SwData2 & (~SwData3));
                SwData3 = SwData2;
            }
            SwStatus = 0;
            break;
    }
}
}

```

## 5-2. プログラムリスト“intprg. c”

```
/*
*****
/* FILE      :intprg.c
/* DATE      :Wed, Apr 26, 2006
/* DESCRIPTION :Interrupt Program
/* CPU TYPE   :H8/3687
/*
/* This file is generated by Renesas Project Generator (Ver.4.0).
/*
*****
#include <machine.h>

extern void intprog_tmb1(void);

#pragma section IntPRG
// vector 1 Reserved

// vector 2 Reserved

// vector 3 Reserved

// vector 4 Reserved

// vector 5 Reserved

// vector 6 Reserved

// vector 7 NMI
__interrupt(vect=7) void INT_NMI(void) {/* sleep(); */}
// vector 8 TRAP #0
__interrupt(vect=8) void INT_TRAP0(void) {/* sleep(); */}
// vector 9 TRAP #1
__interrupt(vect=9) void INT_TRAP1(void) {/* sleep(); */}
// vector 10 TRAP #2
__interrupt(vect=10) void INT_TRAP2(void) {/* sleep(); */}
// vector 11 TRAP #3
__interrupt(vect=11) void INT_TRAP3(void) {/* sleep(); */}
// vector 12 Address break
__interrupt(vect=12) void INT_ABRK(void) {/* sleep(); */}
// vector 13 SLEEP
__interrupt(vect=13) void INT_SLEEP(void) {/* sleep(); */}
// vector 14 IRQ0
__interrupt(vect=14) void INT_IRQ0(void) {/* sleep(); */}
// vector 15 IRQ1
__interrupt(vect=15) void INT_IRQ1(void) {/* sleep(); */}
// vector 16 IRQ2
__interrupt(vect=16) void INT_IRQ2(void) {/* sleep(); */}
// vector 17 IRQ3
__interrupt(vect=17) void INT_IRQ3(void) {/* sleep(); */}
// vector 18 WKP
__interrupt(vect=18) void INT_WKP(void) {/* sleep(); */}
```

```
// vector 19 RTC
__interrupt(vect=19) void INT_RTC(void) { /* sleep(); */}
// vector 20 Reserved

// vector 21 Reserved

// vector 22 Timer V
__interrupt(vect=22) void INT_TimerV(void) { /* sleep(); */}
// vector 23 SCI3
__interrupt(vect=23) void INT_SCI3(void) { /* sleep(); */}
// vector 24 IIC2
__interrupt(vect=24) void INT_IIC2(void) { /* sleep(); */}
// vector 25 ADI
__interrupt(vect=25) void INT_ADI(void) { /* sleep(); */}
// vector 26 Timer Z0
__interrupt(vect=26) void INT_TimerZ0(void) { /* sleep(); */}
// vector 27 Timer Z1
__interrupt(vect=27) void INT_TimerZ1(void) { /* sleep(); */}
// vector 28 Reserved

// vector 29 Timer B1
__interrupt(vect=29) void INT_TimerB1(void) { intprog_tmb1(); }
// vector 30 Reserved

// vector 31 Reserved

// vector 32 SCI3_2
__interrupt(vect=32) void INT_SCI3_2(void) { /* sleep(); */}
```

# 付録

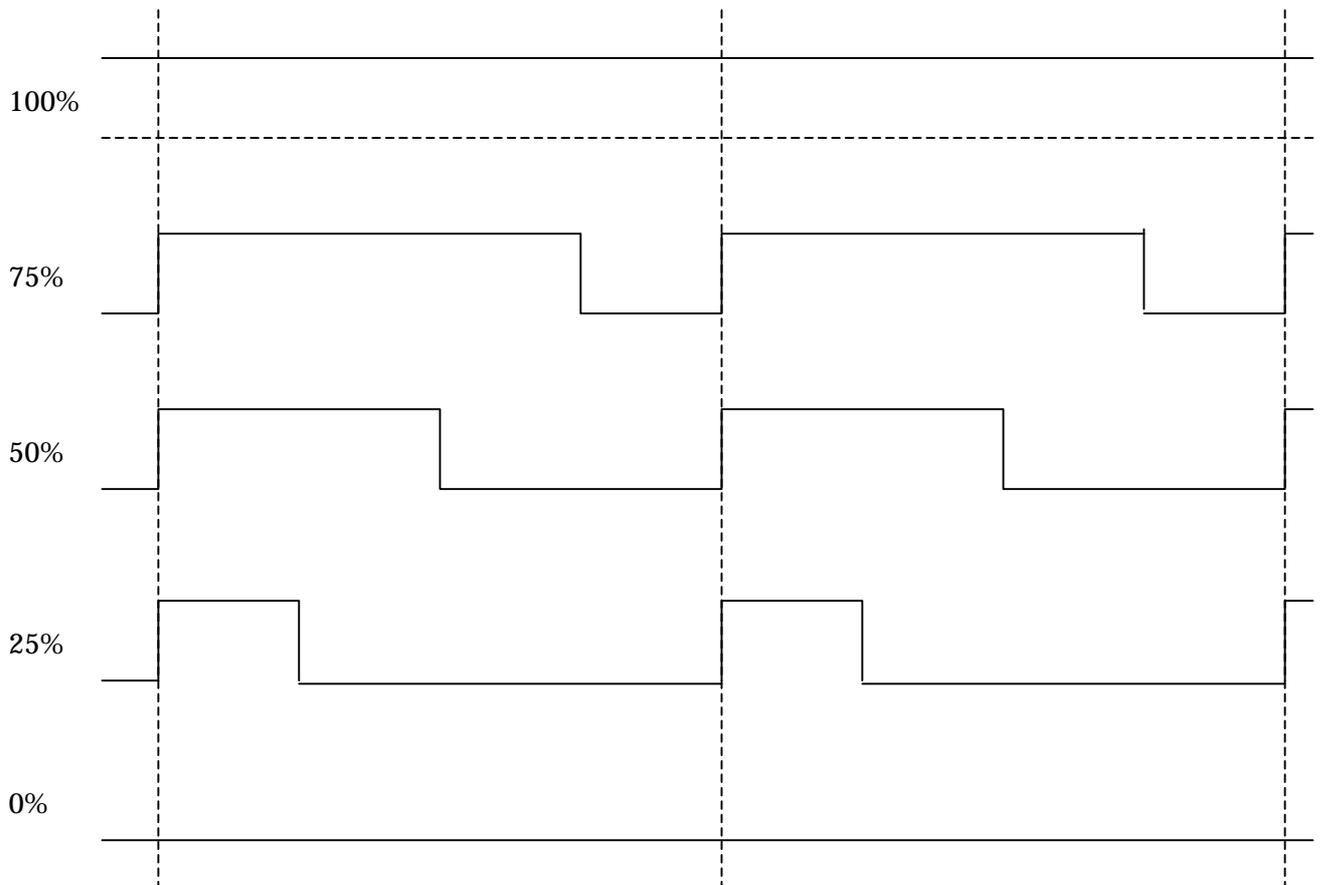
## モータ制御の考え方

## モータ制御の考え方

DC モータは電圧をかけると回転し、電圧をかけないと止まります。しかし、これでは回転するかしないかの 2 種類で、モータの回転数を変化させることはできません。では、どのようにして変化させているのでしょうか。

DC モータは電圧をかけたからといってその瞬間すぐに 100%で回転するわけではなく、また、電圧をかけるのをやめたからといってすぐに止まるものでもありません。徐々に回転が変化していきます。では、電圧のオン・オフをすばやく繰り返したらどうなるでしょうか。電圧が加わると、徐々にモータが回りだします。100%に向かって回転数が上がっていきませんが、すぐに電圧がオフになります。すると今度は徐々にモータが止まろうとします。止まる前に再び電圧が加わるので、また回りだします。これを繰り返せば一定の回転数でモータを回すことができます。また、オン・オフの時間を変化させれば任意の回転数でまわすこともできます。

通常、オンとオフを足した時間を一定にし、オンとオフの比(デューティ)を変化させます。このような制御方法を PWM(Pulse Width Modulation:パルス幅変調)といいます。



‘F1 ミニ’の場合、左右のモータを別々に PWM で制御しています。また、設定メニューでセットする数値は、PWM のデューティをあらわしています。

## 株式会社 東洋リンクス

ユーザサポート係(10:00~17:00, 土日祭は除く)

〒102-0093 東京都千代田区平河町 1-2-2, 朝日ビル

TEL:03-3234-0559 / FAX:03-3234-0549

E-mail : toyolinx@va.u-netsurf.jp

<http://www2.u-netsurf.ne.jp/~toyolinx>(最新情報はこちらからダウンロードできます)

- ★ 本書の内容は将来予告なしに変更することがあります。(2006年8月作成)
- ★ 本書の著作権は(株)東洋リンクスが所持しています。
- ★ 万一、不足部品や破損部品があった場合は(株)東洋リンクスまでお問い合わせください。
- ★ 掲載された回路、プログラム等を利用した結果生じたトラブルについて弊社は責任を負いかねますので、あらかじめご了承ください。
- ★ 本書で用いている固有名詞は一般に各メーカーの商標、もしくは登録商標です。

20060904