大阪電気通信大学　自由工房
マイコンカーラリーオープンセミナー

Basic　Class（基礎編）

群馬県立前橋工業高等学校
電気科　阿佐美　斉

マイコンカーで心掛けていること

- 他人に任せない
- 調整箇所を少なくする
- いつでも完全動作
- 真っ直ぐに走る車を製作する
- オーバーアクションは控えて最短距離を走る

Basicのイメージ

- デジタルセンサ
- エントリーモデルのラジコンサーボ
- 遅い
- フラフラと走る
- 落ちる
- 制御する楽しみが無い
- キットと解説書そしてサンプルプログラムがあるので指導者は楽？

講習会の流れ

- 車体の製作とプログラムについて（基礎編）
  - 製作上の注意事項
  - タイヤの製作とセンサ基板の改造
  - 完走できない理由
  - Rを滑らかに走行をさせる方法1
- 改造とプログラムについて（実践編）
  - ドライバ基板の改造ポイント
  - Rを滑らかに走行をさせる方法2
  - クランクと車線変更の攻略
  - 電池の管理

本日の実技ポイント

- モータの測定
- ギヤボックスの調整
- タイヤの製作方法
- 車体材料の紹介
- センサの改造
- 完走できない理由を考える
- Rの走り方と練習方法

昨年製作した車体の特徴

- FREEモード搭載ドライバ基板
- 接触箇所の少ない電池ボックス
- コネクタの交換
- シンプルな車体
- 小径軽量な駆動輪
- 極小軽量な前輪
- 調整したギヤボックス
- 選別したモータと電池の使用

Basicの魅力

- 限られた条件の中で走る
- 自分の目と耳で感じながらプログラムできる
- Basicで得たことはAdvansへフィードバックできる

1.車体の製作

ハードウェア編

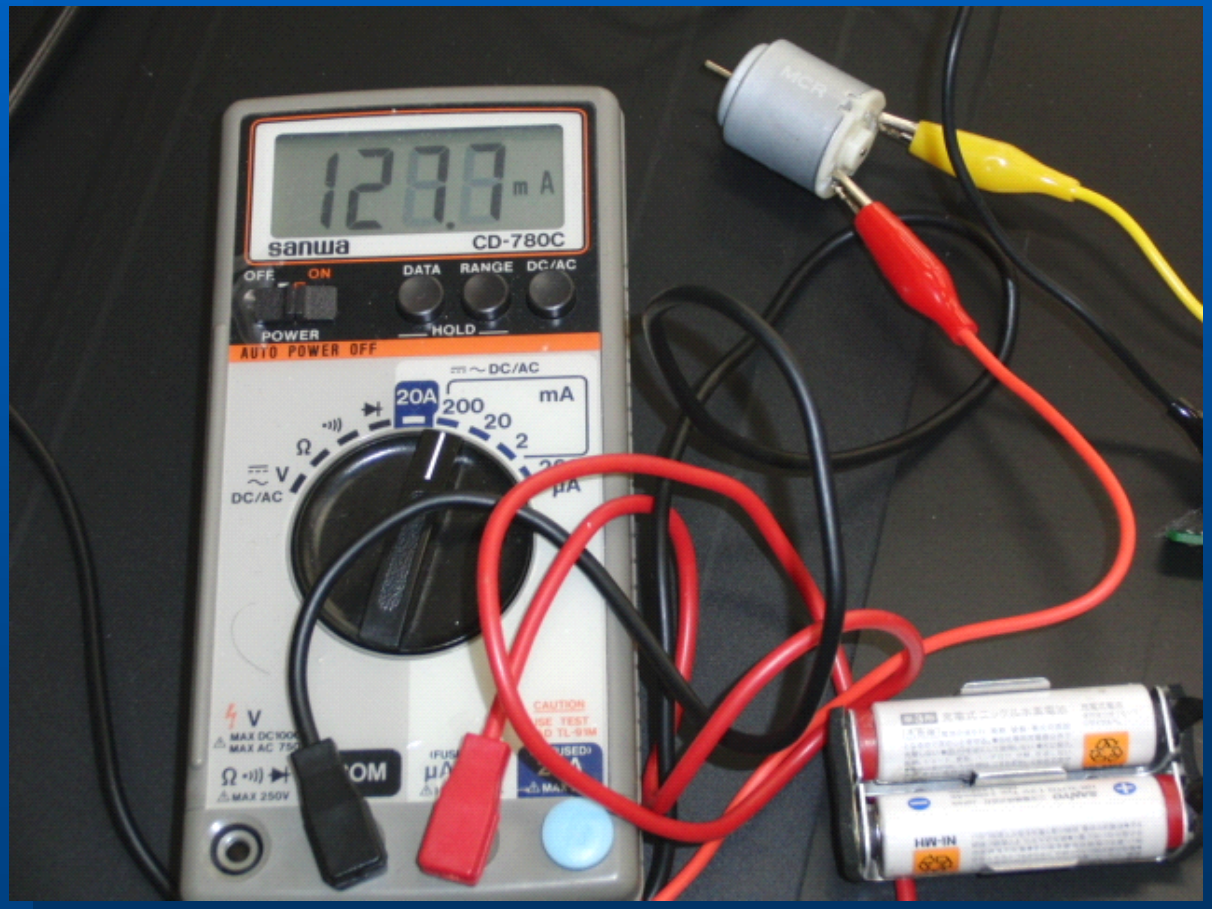
指定モータ

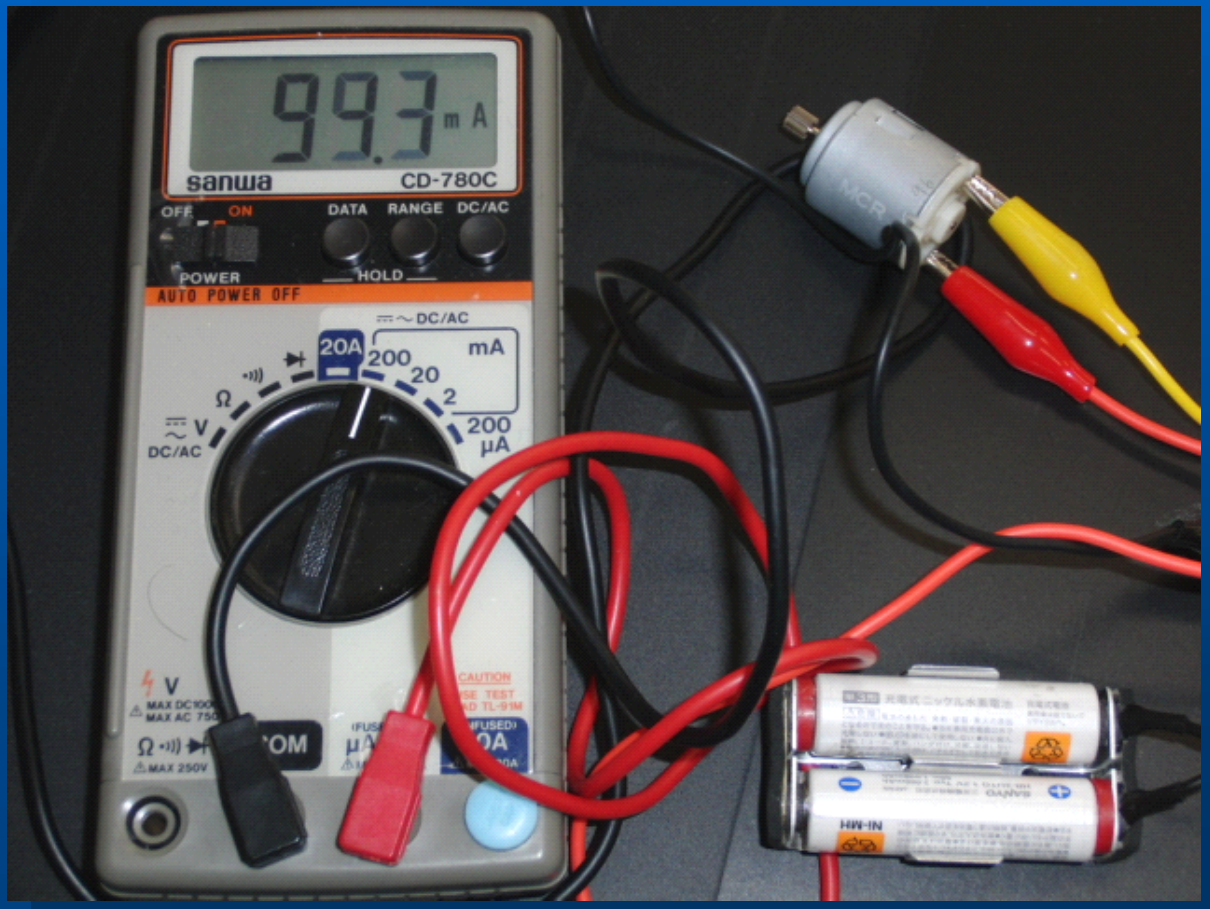
使用には注意が必要

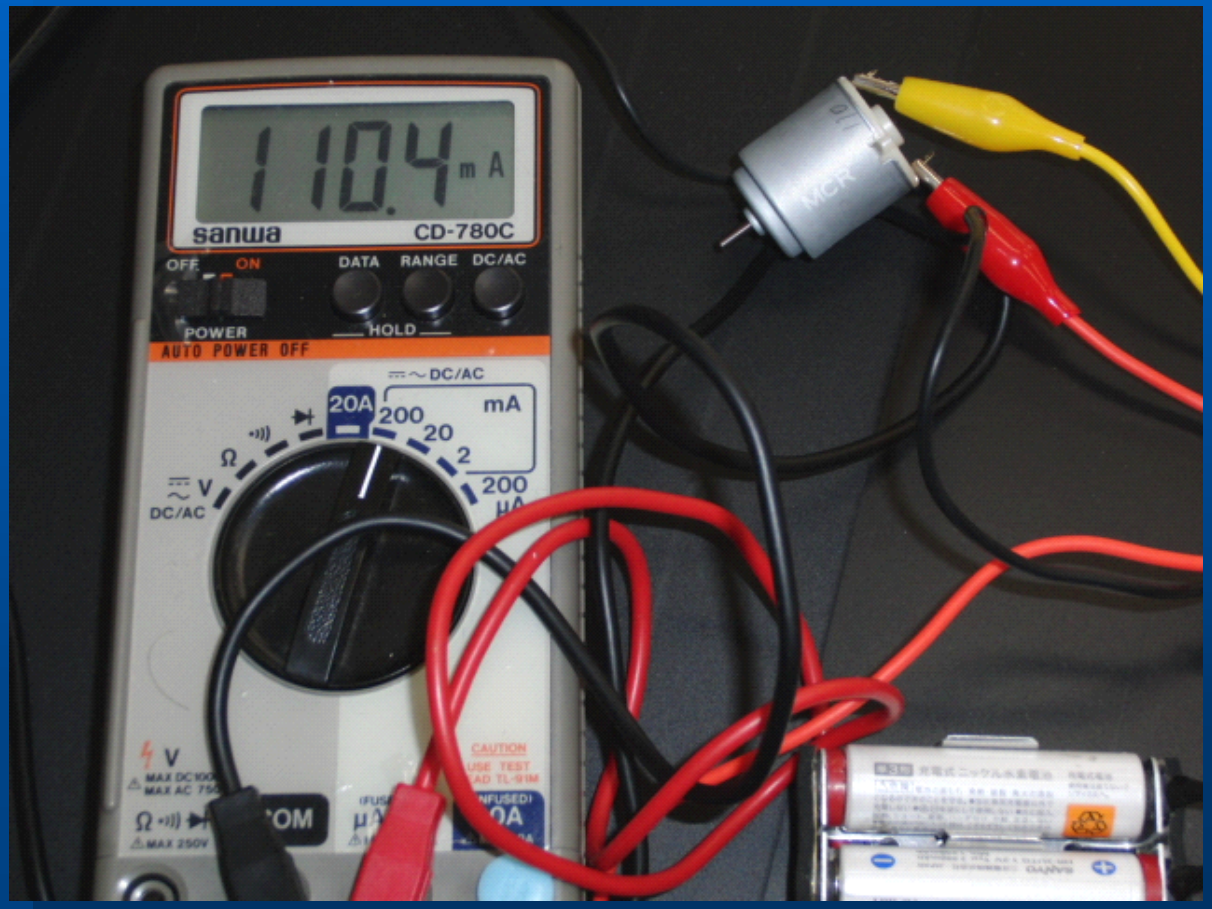
モータの選別

- 約2．8V（電池2本）で回して、異音や振動があるものは使わない。約半分が使用できない。
- 正逆転それぞれ30分程度、慣らし運転後に電流を測定。100～130mA程度。
- ほぼ同じ電流値のモータを組み合わせる。
- Basicでは1ヶ月ぐらいは交換しないで使えると思う。（それ以上は試していない）

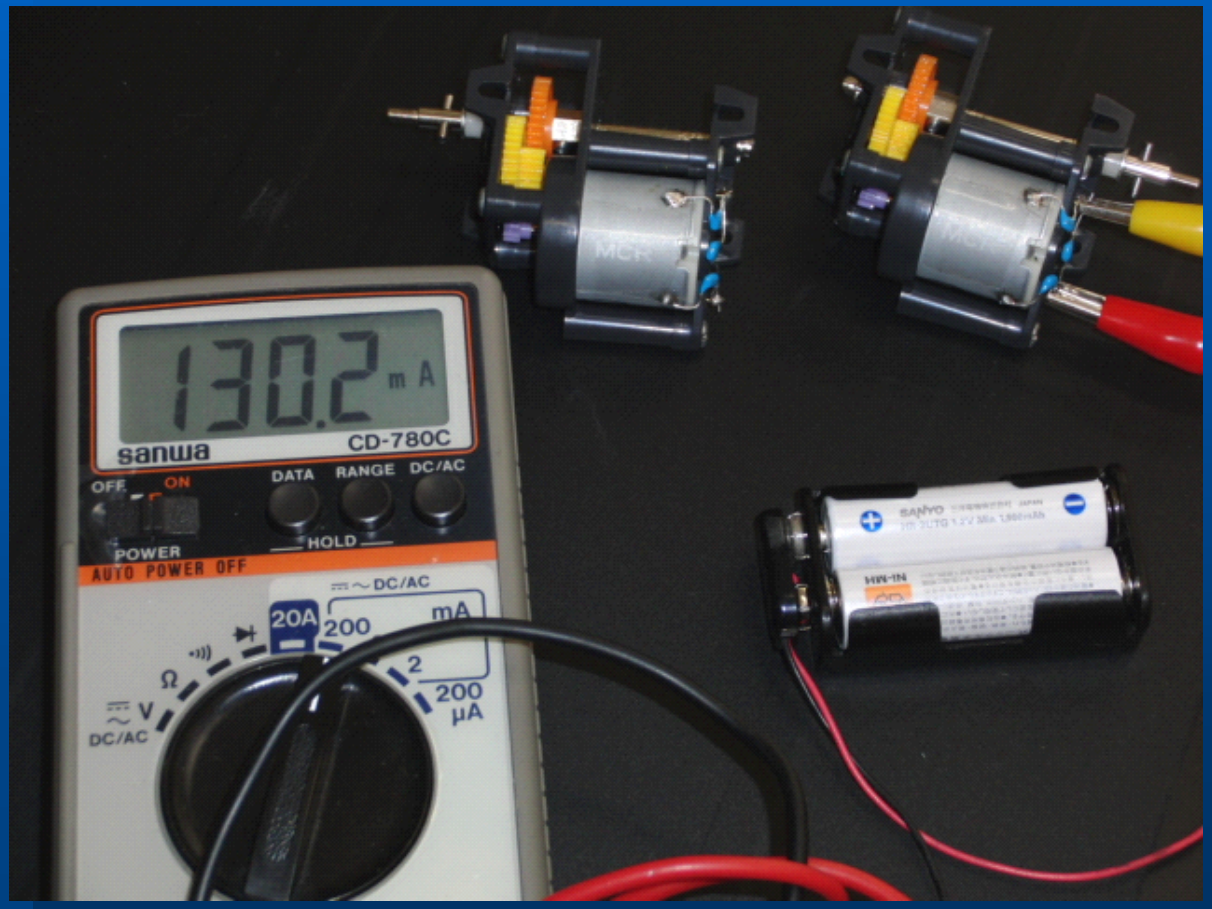
モータ単体での測定

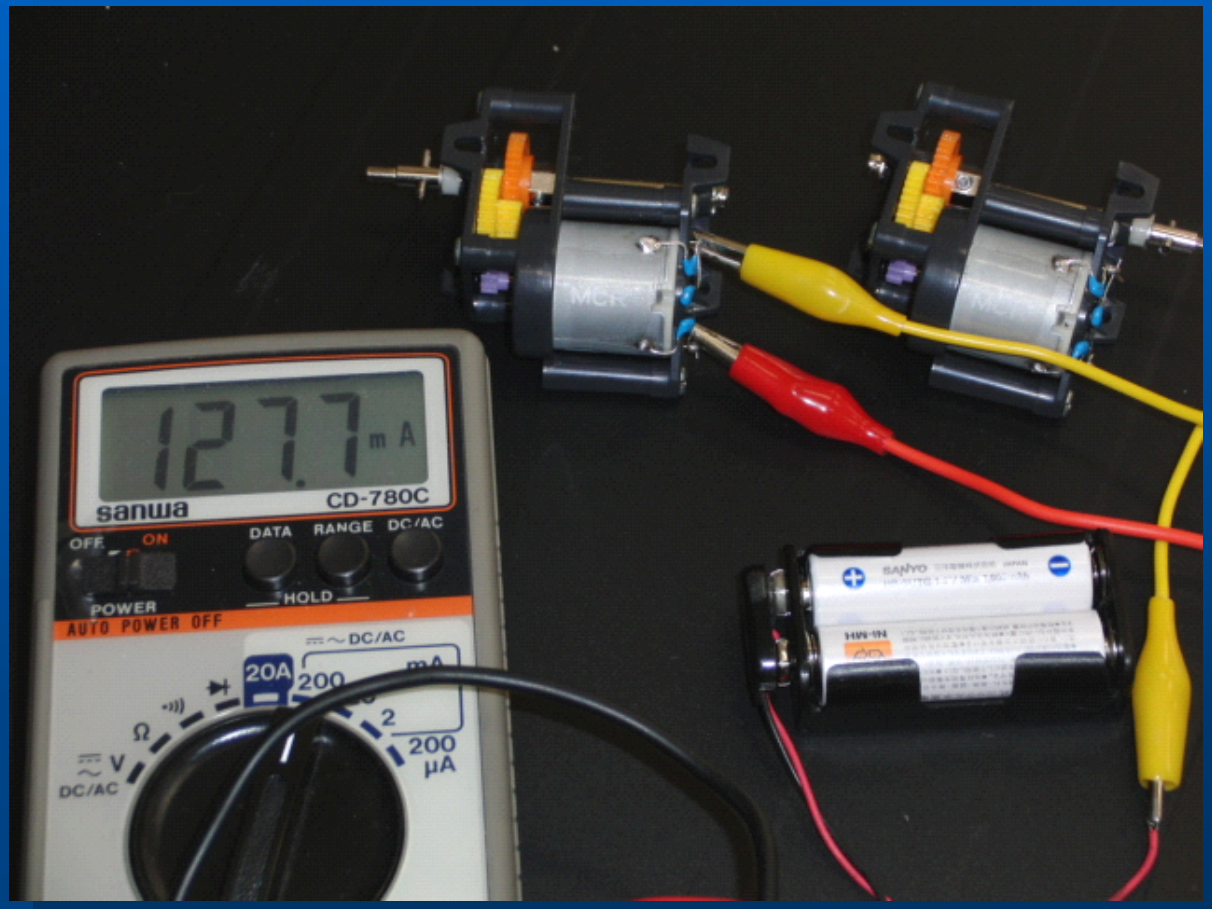

128.8 mA
sanwa
CD-780C
OFF ON
POWER
DATA
RANGE
DC/AC
HOLD
AUTO POWER OFF
DC/AC
20A
200
20
2
mA
μA
Ω
V
DC/AC
COM
CAUTION


99.3 m A
sanwa
CD-780C
OFF ON
POWER
DATA RANGE DC/AC
HOLD
AUTO POWER OFF
DC/AC
20A
200
20
2
mA
200
µA
Ω
V
DC/AC
HW-IN


110.4 mA
sanwa
CD-780C
OFF
ON
POWER
DATA
RANGE
DC/AC
HOLD
AUTO POWER OFF
DC/AC
20A
200
20
2
200
mA
μA
Ω
V
DC/AC
COM
CAUTION

ギヤボックスに取り付けて測定




127.7 mA
sanwa
CD-780C
OFF
ON
POWER
DATA
RANGE
DC/AC
HOLD
AUTO POWER OFF
DC/AC
20A
200
mA
2
200
µA
Ω
V
DC/AC
SANYO

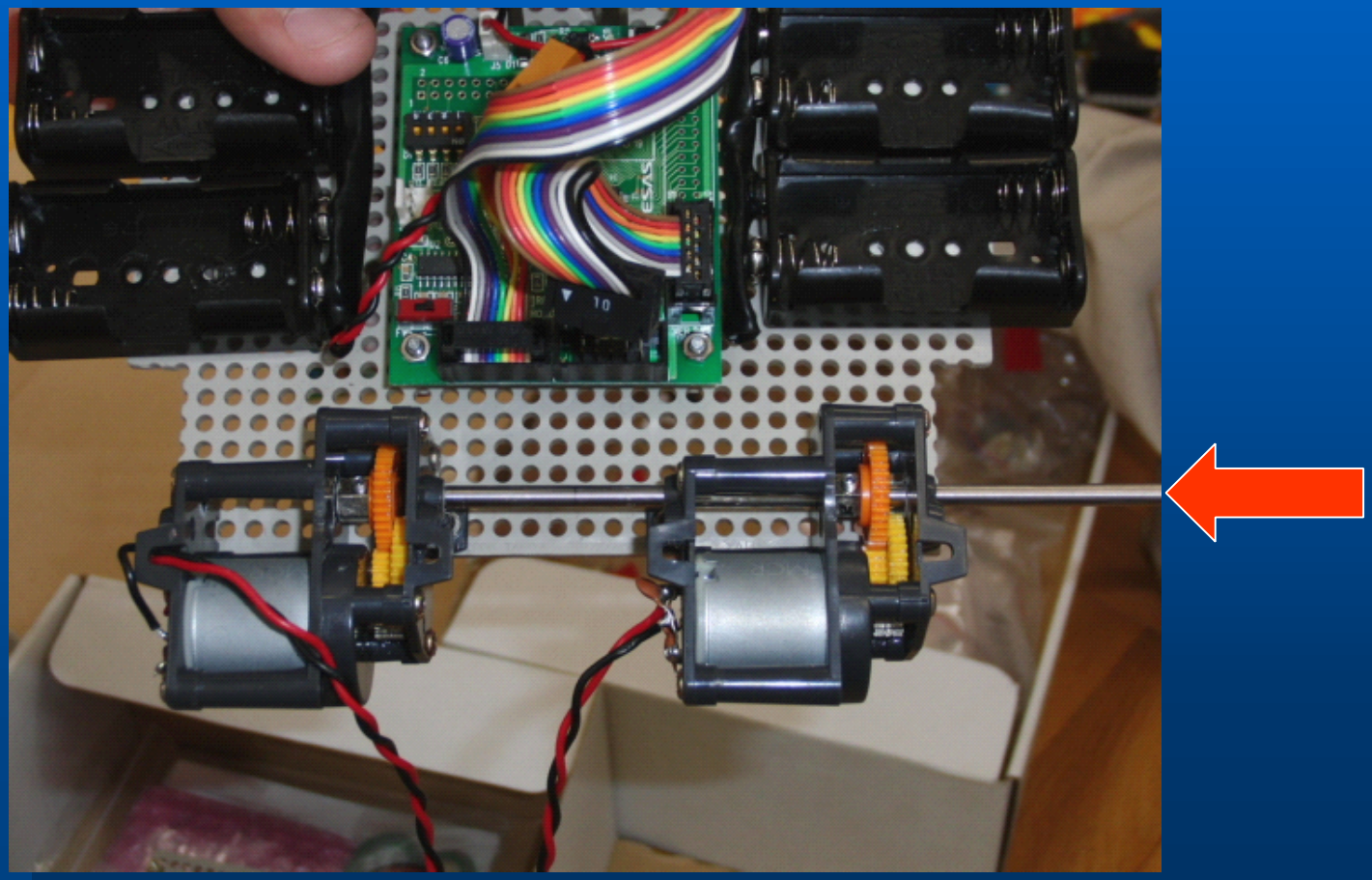
ギヤボックスの組み立て

- 必ず机の上などの平らな上に置いて組み立てる。グリスを忘れない。
- 一度にネジを全て締めないで、モータを回して電流の値が大きくならないように締める。タミヤのギヤボックスに取り付けても電流値はあまり変わらない。
- 締まらないネジがあったり、壊れている物は使用しない。

ギヤボックスの取り付け

- シャーシにギヤボックスを取り付けても電流値が変化しないか確認する。
- 2つのギヤボックスの位置が曲がっていないか確認する
- 2つのギヤボックスにシャフトを貫通させてみる

滑らかに貫通できるか



指定サーボについて

非力なサーボを生かす

- トルクが無いので前輪はコンパクトに仕上げ
  センサアームも軽く仕上げる
- なるべく軸にストレスを与えないようにする
- 無理に制御周期を短くしない
  8msは危ないと思う
- 意外と個体差が大きい
  左右の切れ角や制御周期など
- そのサーボ専用のプログラムとなる

駆動方式と前後のバランス

前輪駆動か後輪駆動か

前輪駆動

- 最初の優勝マシンは前輪駆動であった
- 駆動力で非力なサーボをアシストできる
- 正確な制御をしないとステアリングが暴れる
- 加速時に駆動輪（前輪）に加重がかかりにくい
- ブレーキが強い

後輪駆動

- 2回目以降の優勝マシンは後輪駆動である
- 滑らかで素直な走りが期待できる
- 加速時に加重が駆動輪（後輪）にかかりやすい
- ブレーキが弱い

前後の重量バランス

- 駆動輪側にやや多めに重量がかかるように電池ボックス等を配置する
- 後輪駆動であれば
  前：後＝４：６から４．５：５．５ぐらいか
- 間違えると、加減速に乱れが出たり、カーブで曲がれなくなる

駆動輪について

車はタイヤが勝負

駆動輪の直径

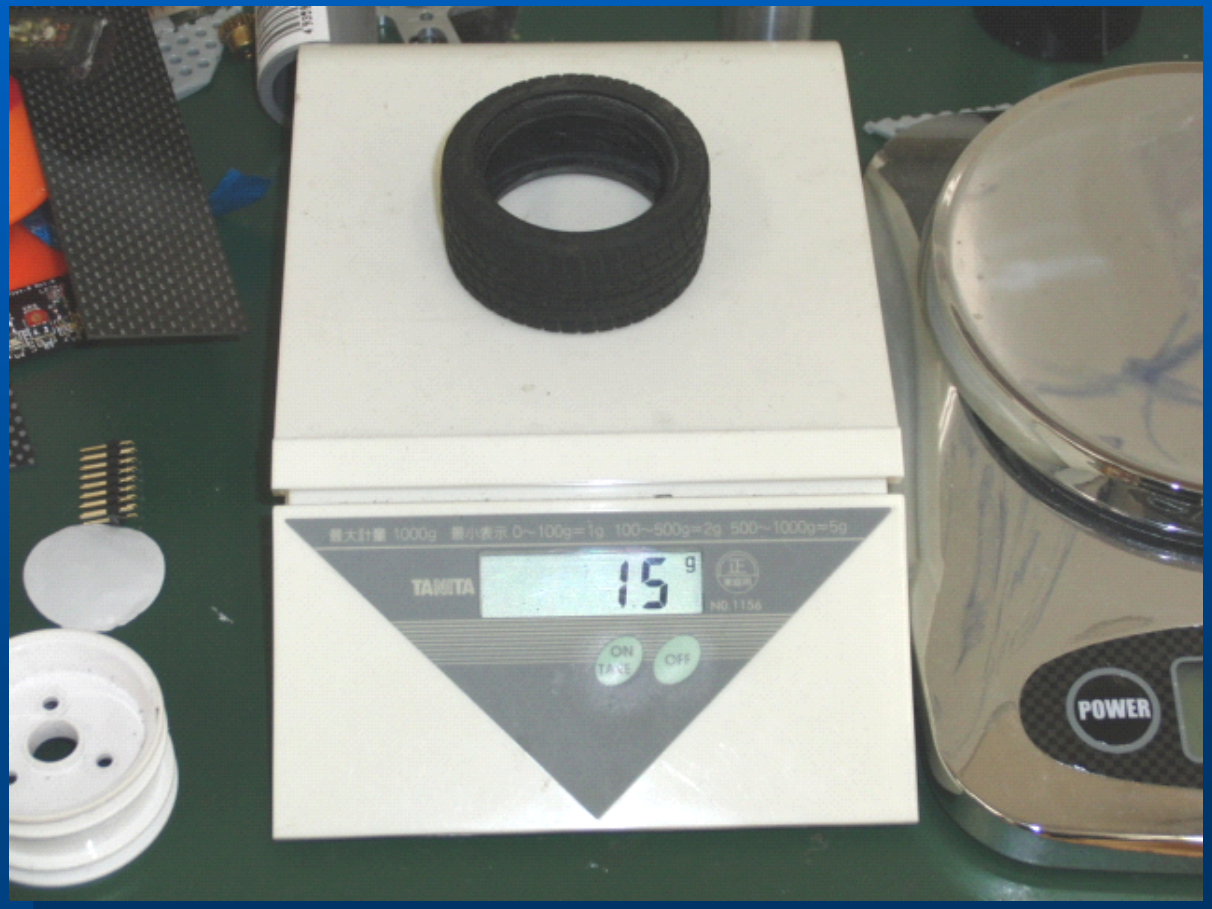
- 直径が大きいとトップスピードの伸びは期待できるが、加速と減速に問題が出てくる。
- 直径が小さいとトップスピードの伸びは期待できないが、加速と減速で有利である。
- MCRのコースでは直線は6m程度であるので素早く加速し減速も可能なサイズを選定することになる。5～5．3cmぐらいがBESTであると思う。実験と体験での値である。

駆動輪の重量

- 重量は意外と加速と減速に影響を与える。現在の重量は1輪当たり12g以下である。
- 軽い車輪では驚くほどブレーキが効くようになる。


1000g 最小表示 0~100g=1g 100~500g=2g 500~1000g=5g
TANITA
12 g
NO.1156
ON TARE
OFF
hpi-racing
POWER
26g
RES
Max:2000g Min:1g

ゴムタイヤだけで15gある



製作したもの

タイヤの幅（考察）

- アドバンスでは幅広のタイヤが多い
- 非力なモータ（1輪1モータ）と軽量な車体では幅の狭いタイヤの方が加重がかかりやすい
- キット付属のホイールの25mm程度が良い感じ

トレッド（考察）

- Advansより遅いコーナーリングスピードとなるのでトレッドは狭くても大丈夫そうである
- 狭くすることでコーナーでの外輪はセンターライン寄りを走行できる
- 同じ回転で走行した場合センターライン寄りを走行した方がコーナーリングスピードが上がる

タイヤの製作方法

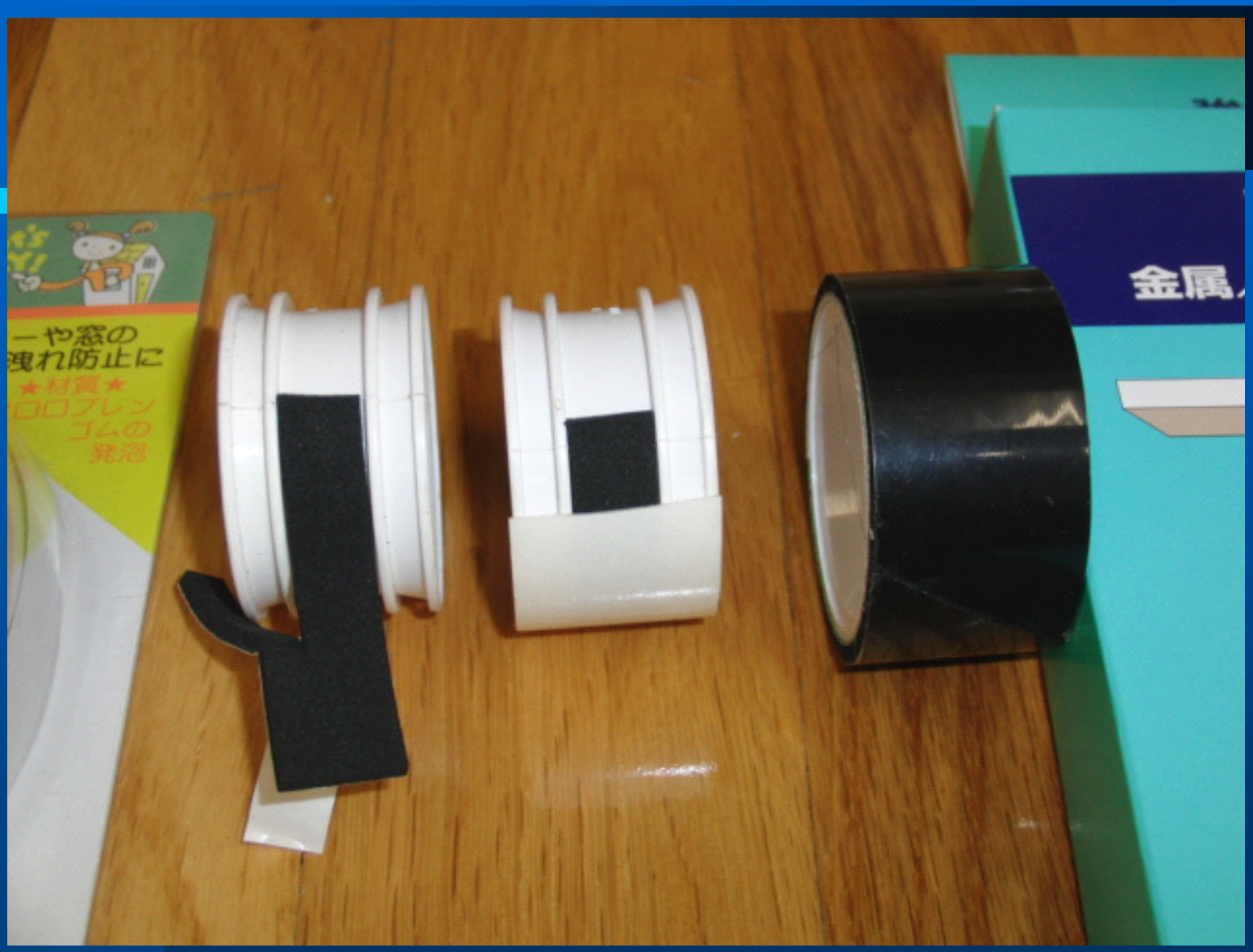
扱いにくいキットの付属品を生かす

- 準備するもの
  強力両面テープ（幅25mm）
  厚さ2．5mmの隙間テープ
  シリコンシート


防音戸当たり
●ドアーや窓の音洩れ防止に
2本入り
2.5mm×15mm×2m
強力両面テープ
業務用
強力両面テープ
金属／ガラス／プラスチック／木材用
業界初！ノントルエン
強ピタ
(厚)×25mm(幅)×10M(巻き)

製作手順

- キットのホイールにはリブがあるために直接スポンジを巻きにくい
- スポンジを巻きやすくするために溝にスポンジテープを貼る
- その上に強力両面テープを貼る
- その上にスポンジテープを貼る
- シリコンシートを貼って完成

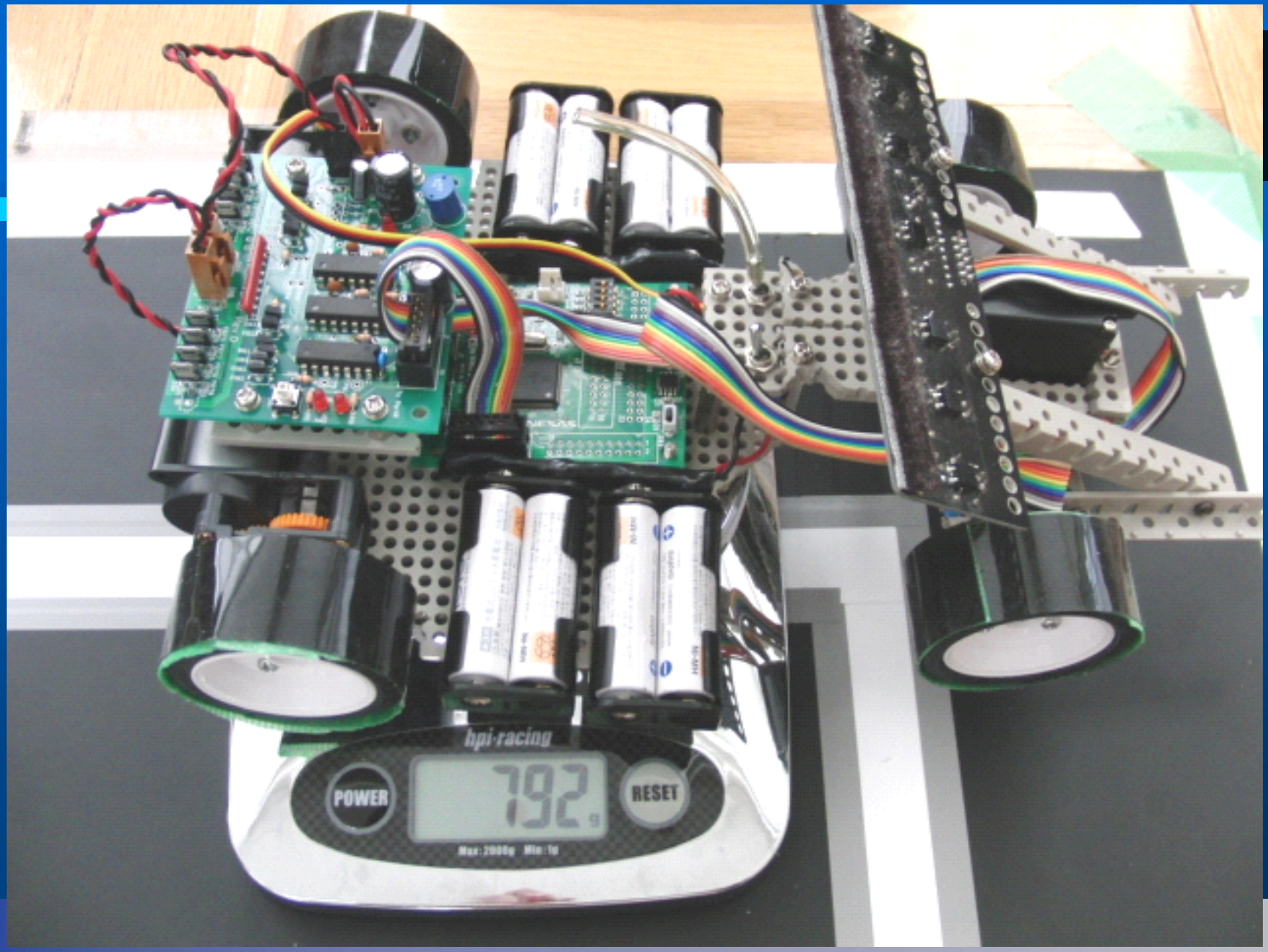

ーや窓の
魂れ防止に
★材質★
ロロプレン
ゴムの
発泡
金属

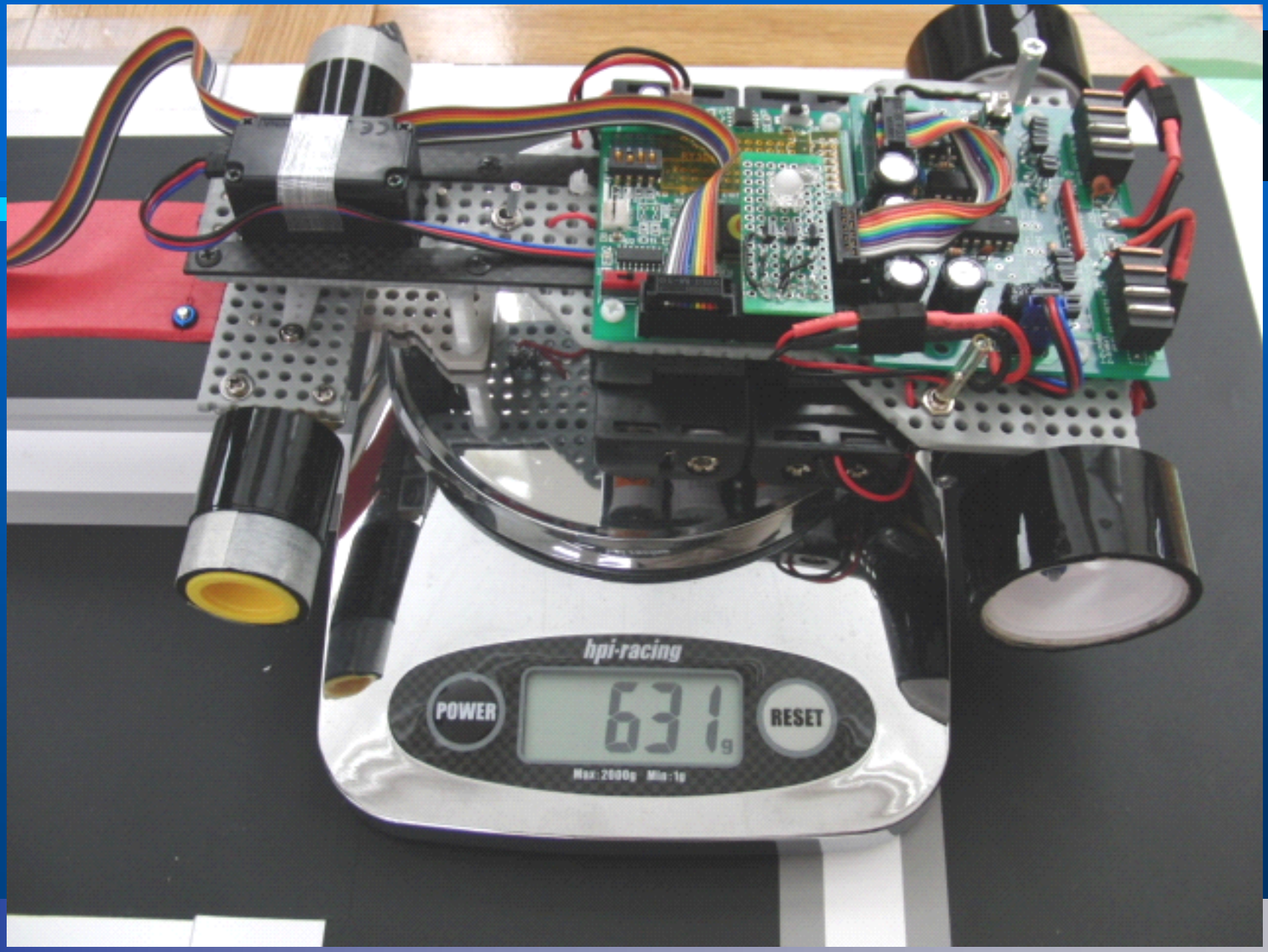
車体の製作について

強く軽く精度良く

ユニバーサルプレート

- 入手しやすく加工も楽である
- 穴が開いているため意外と精度良く組み立てられる
- 重量は1．5mmのFRPと同等
- 柔らかいので補強が必要


hpi·racing
POWER
792g
RESET


hpi-racing
POWER
631g
RESET
Max:2000g Min:1g

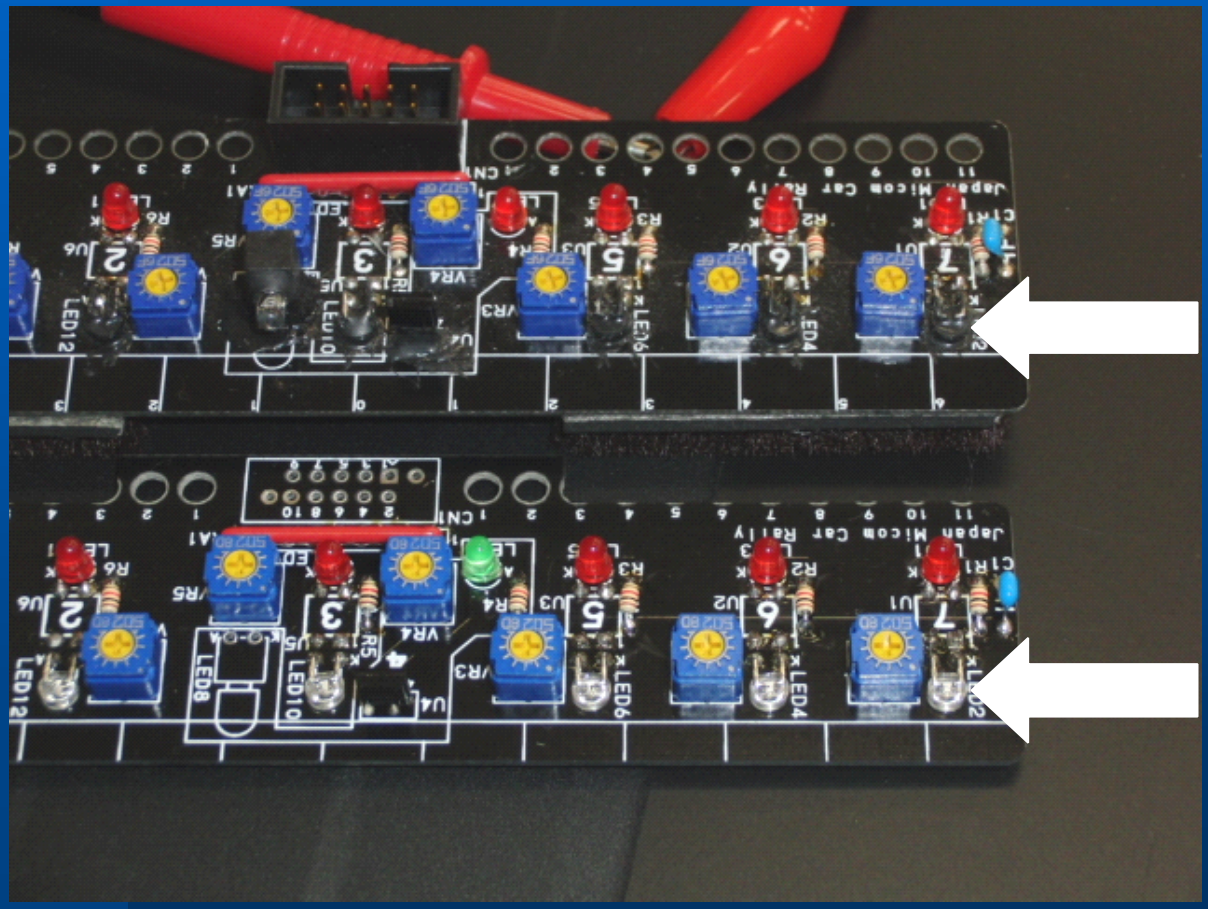
センサ基板

デジタルセンサでも外乱光防止

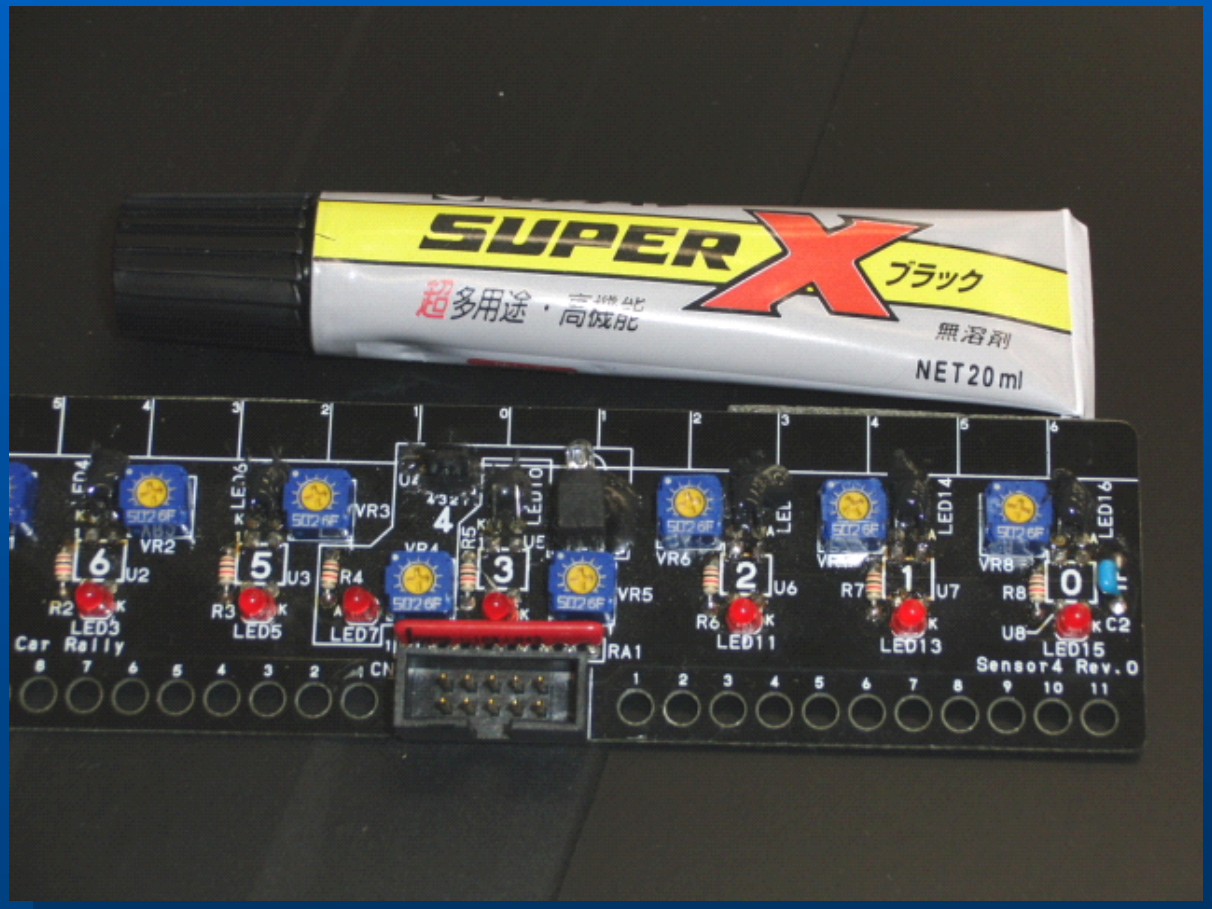
センサ基板の改造

- 赤外LEDを通して外乱光が進入する
- LEDの上部を黒くする
- センサの感度が安定して調整が楽になる

改造前（下）と改造後（上）



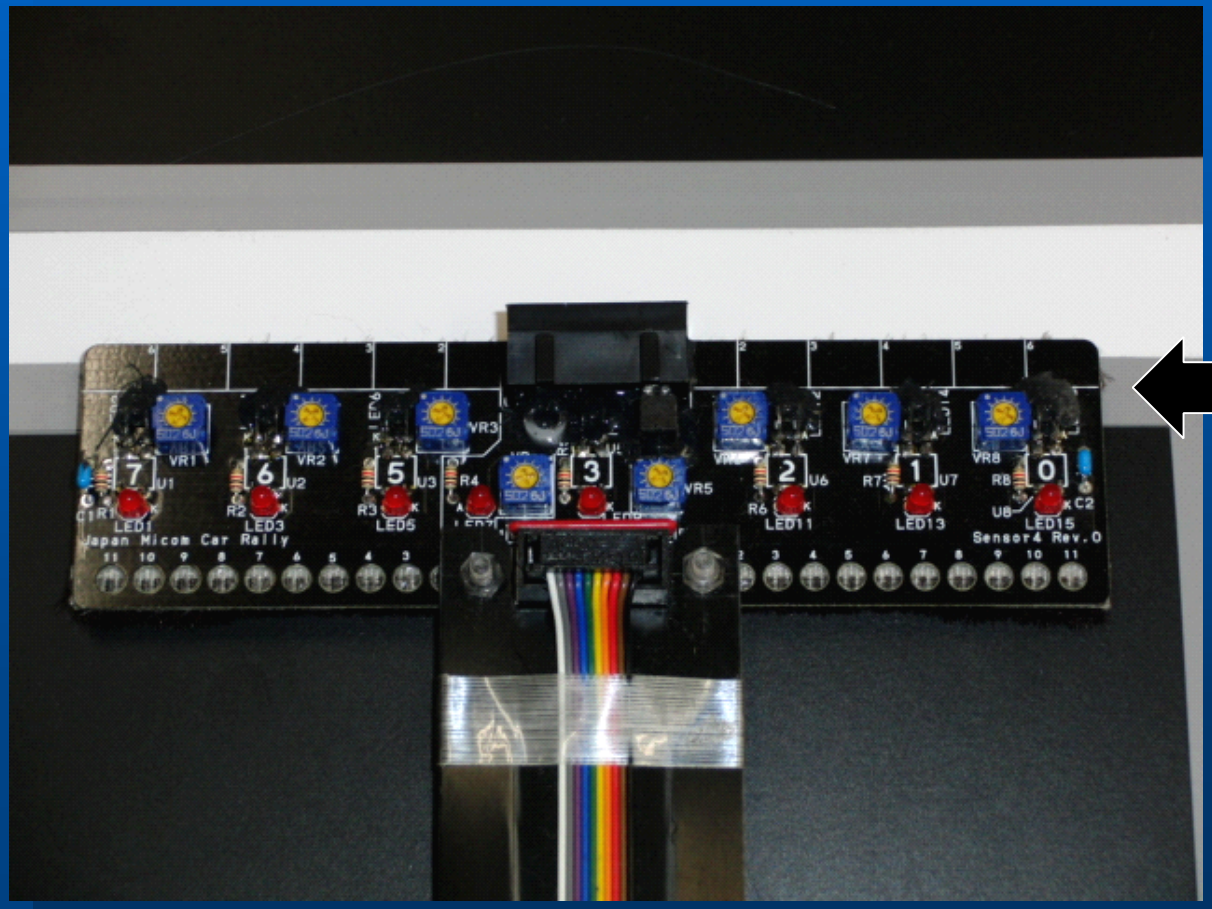
セメダインSUPERXを塗っただけ



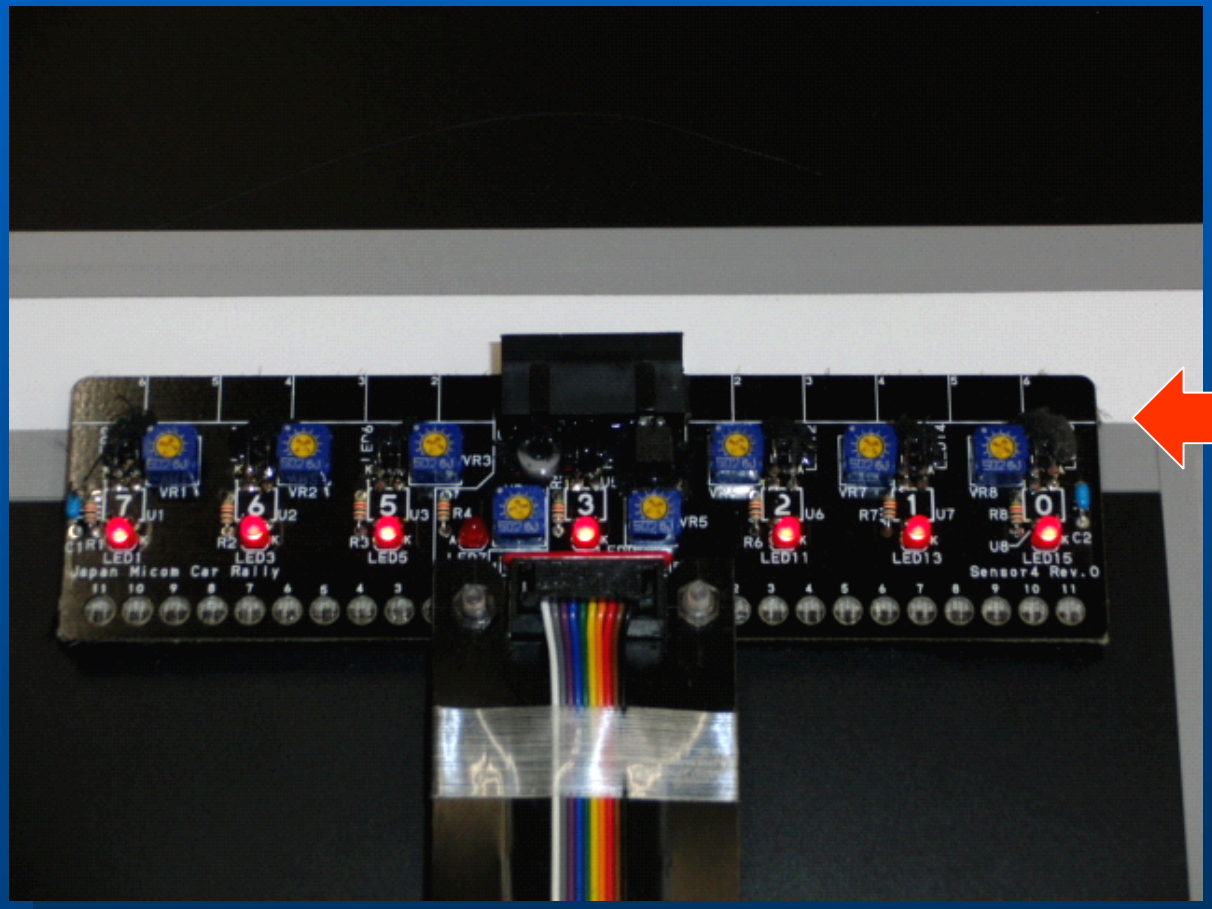
センサ調整のポイント

- 必ず走行させる場所で調整する
  車検場所と走行場所が異なる場合は要注意
- コースアウトしたら必ず確認をする
- 走行後センサに埃が付着していないか
- 黒でLEDが消灯するようにしてから合わせると楽に合わせられる
- 灰色ではLEDが点灯するようにする

ここでは点灯しない


LED1
Japan Micom Car Rally
LED3
LED5
LED11
LED13
LED15
Sensor4 Rev.0

ここで全て点灯する


LED1
Japan Micom Car Rally
LED3
LED5
LED11
LED13
LED15
Sensor4 Rev.0

2.プログラムの調整

ソフトウェア編

走行プログラムについて

マイコンカーの醍醐味

ライバルと同じスピードで走れない

Rの走り方（1．8m／s）

- プログラムの改造（センサパターン等）
- 左右の内輪差を計算する
- 計算が面倒であればdiffを入れる
- 走行スピードを上げる

Rの練習方法

- 450Rの円形コースを滑らかに走行できるように調整する
- プログラムで詰められないところはセンサアームの長さを変えることで対応する

完走できない理由を考える

マイコンカーは
プログラム通りに動作する

マイコンカーは悪くない

- プログラム通りに動作する。
- プログラムに記述されていないことはやってくれません。
- 壊れたり、性能が異なった部品を使用していないか。

kitプログラム

- 理想的な走行をすることを前提として記述されている。
- 理想的ではない場面を想定して自分で修正しなければ完走できない。
- 理想的でない場面の代表的なものがクロスラインとハーフラインの見極め。

制御の基準がずれていないか

- 左右のモータの特性は揃っているか。
- ギヤボックスの噛み合わせはどうか。
- ギヤボックスの組み付け状態はどうか。
- サーボのセンターは合っているか。
- センサの感度調整はどうか。

真っ直ぐ走る車を作る

- 車体に歪みが無いか。
- ハンドル0度の状態で押して60cm以上真っ直ぐ走るか。
- モータを駆動させてどのぐらい真っ直ぐ走るか。
- 左右前後のタイヤの接地状態はどうか。

話は最後まで聞く？

- センサ状態の確認を１度だけで実行しない。
- 進入角度の大きさやセンサ感度のバラつきによって誤ったセンサ状態がありうる。
- 起こりうる状態を考慮してプログラムを作る。

クロスライン検出

```
/*****************************************************/
/* クロスライン検出処理                              */
/* 戻り値 0:クロスラインなし 1:あり                  */
/*****************************************************/
    b = sensor_inp(MASK3_3);
    if( b==0xe7 || b==0x67 || b==0xe6 || b==0x66 ) {
```

右車線変更検出

```
/*****************************************************/
/* 右ハーフライン検出処理                            */
/* 戻り値 0:なし 1:あり                              */
/*****************************************************/
    b = sensor_inp(MASK4_4);
    if( b==0x1f || b==0x3f || b==0x7f ) {
```

新しいspeed関数の追加

- クランクではcnt1の時間のタイミングのみで走行している。
- 完走させようとスピードを落とすとタイミングがずれてコースアウトしてしまう。
- DIPSWに関係なくspeedが設定できる新しいspeed関数を追加することで解決できる。

通常のspeed関数

```
/*****************************************************************************/
/* 速度制御                                                                  */
/* 引数    左モータ:-100～100 , 右モータ:-100～100                           */
/*         0で停止、100で正転100%、-100で逆転100%                            */
/*****************************************************************************/
void speed( int accele_l, int accele_r )
{
    unsigned char   sw_data;
    unsigned long   speed_max;

    sw_data  = dipsw_get() + 5;         /* ディップスイッチ読み込み */
    speed_max = (unsigned long)(PWM_CYCLE-1) * sw_data / 20;
```

追加するspeed関数

```
/*****************************************************************************/
/* 速度制御2　ディップスイッチは関係ない                                      */
/* 引数　左モータ:-100～100 , 右モータ:-100～100                              */
/*       0で停止、100で正転100%、-100で逆転100%                               */
/*****************************************************************************/
void speed2( int accele_l, int accele_r )
{
    unsigned long   speed_max;

    speed_max = (unsigned long)(PWM_CYCLE-1);
```

プロトタイプ宣言にも登録を忘れないで

```
        void speed2( int accele_l, int accele_r );
```

使い方

- speed2(100 ,100);とすればDIPSWに関係なく常に100%の出力となる。
- クランクの中では40～50ぐらいで調整すると良いと思います。

- 講師ブログ
  http://mmc07.blog95.fc2.com/

- 講師動画ページ
  http://www.youtube.com/user/aiMMC07