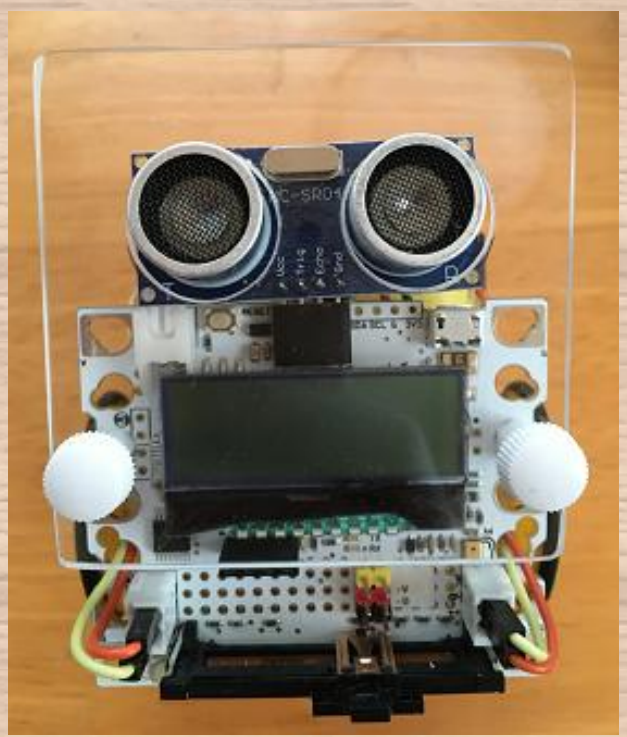
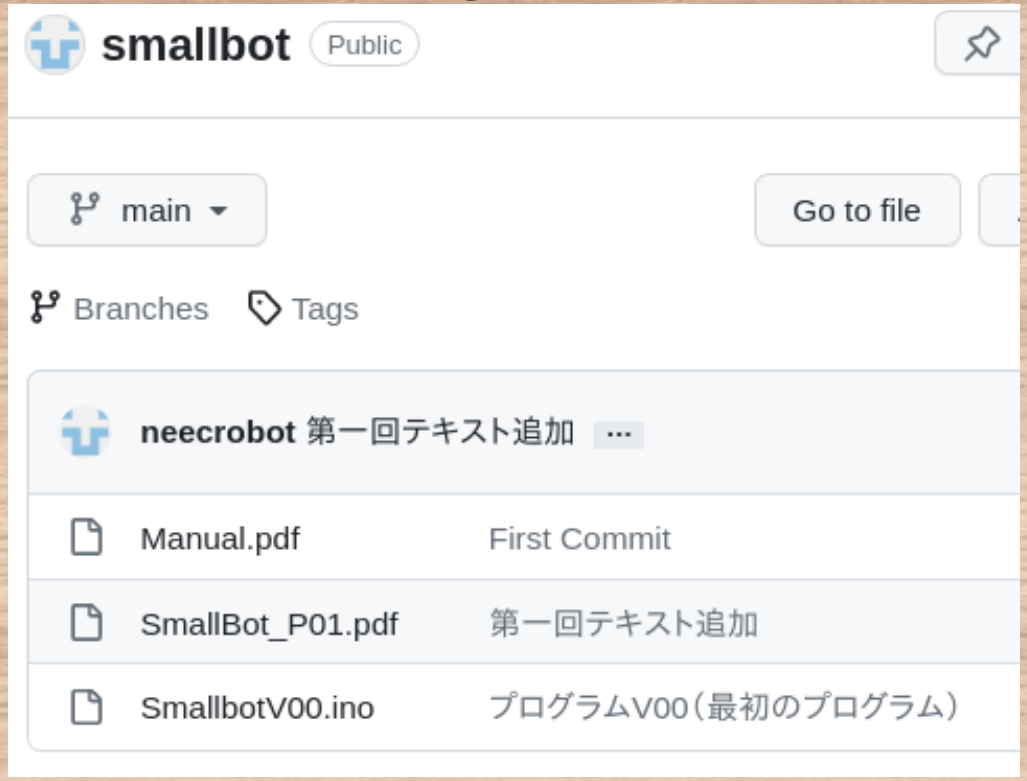
ロボットプログラミング#2



SmallBotを動かそう

- 前回の復習
- マクロ(#define)の利用
- ライントレースさせてみよう

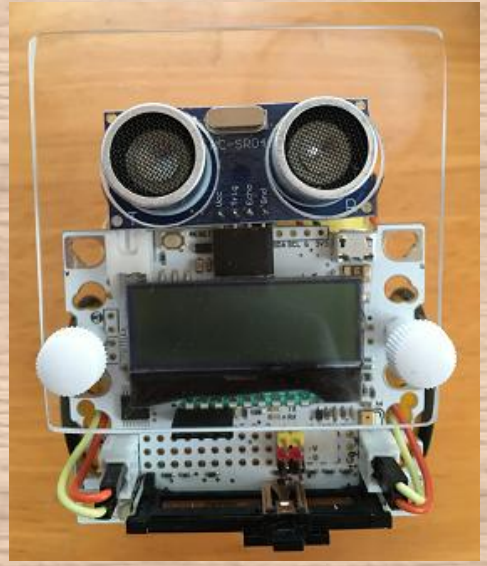
マニュアルやプログラム類は githubに置いてあります

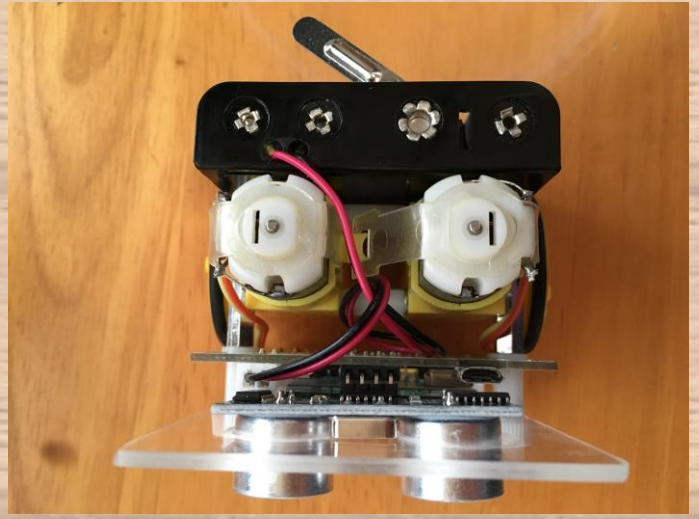


置いてあるファイル
はクリックして
ダウンロード可

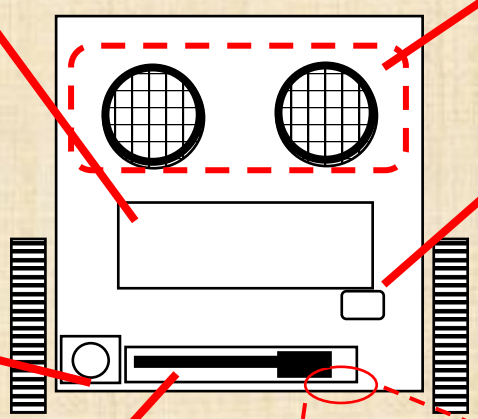
https://github.com/neecrobot/smallbot

SmallBotについて





SmallBotの主要部品配置



キャラクタLCD

超音波距離センサ

右モーター

マイク

押しボタンスイッチ

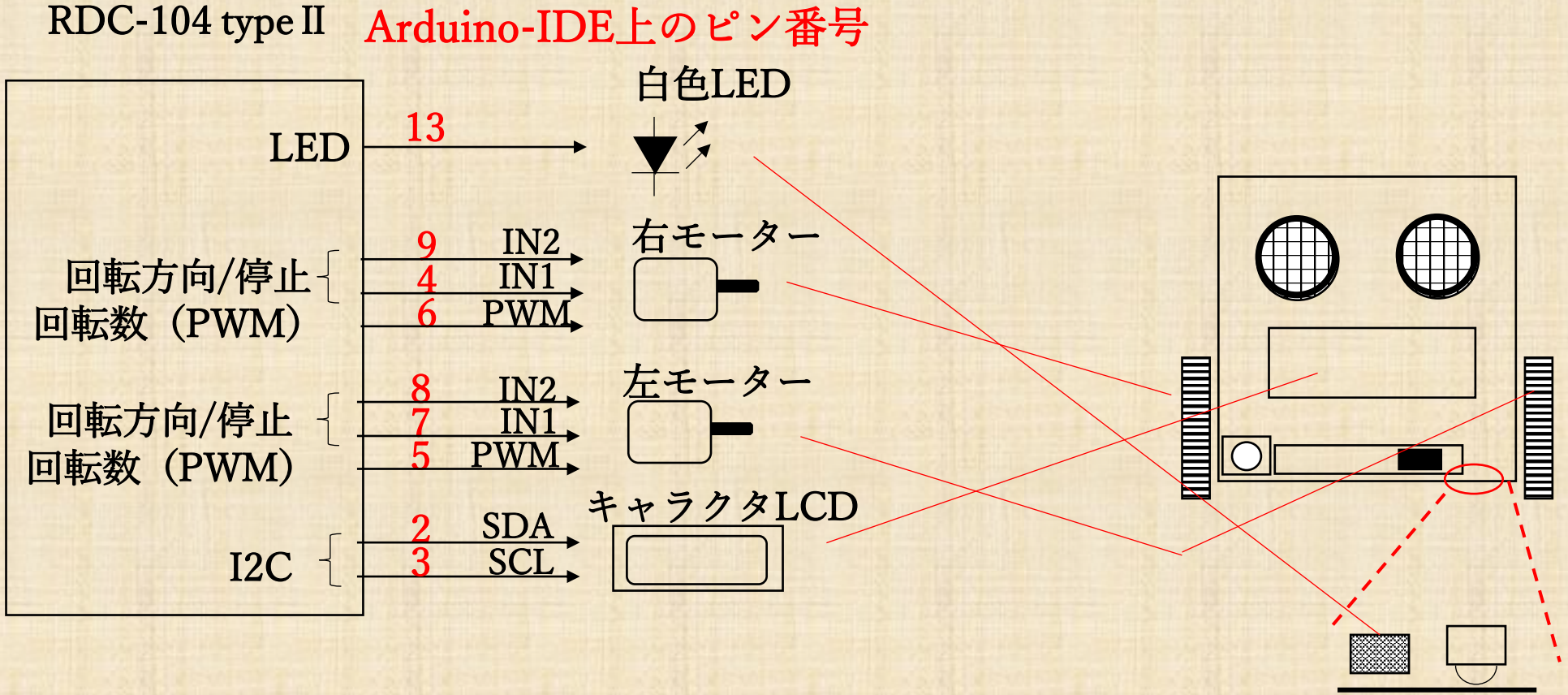
左モーター

スライドボリューム

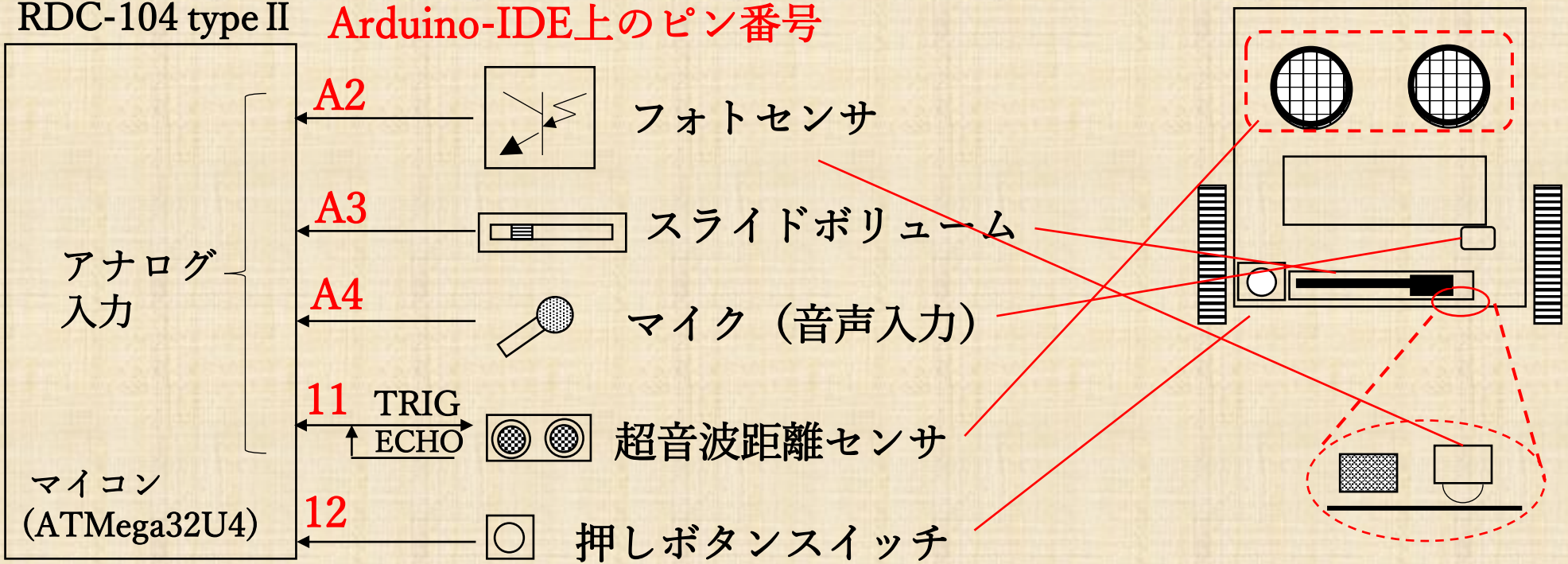
白色LED

フォトセンサー

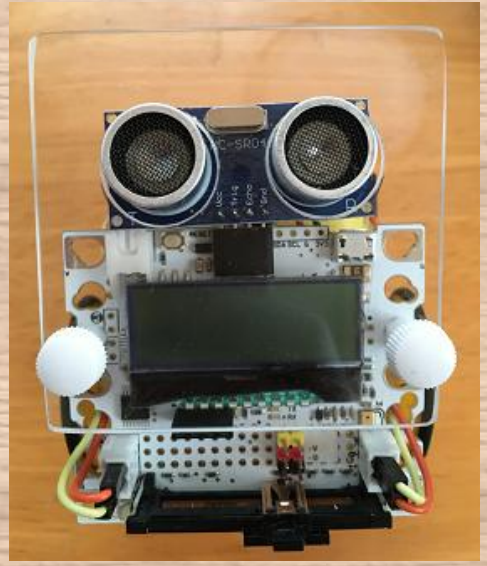
出力信号の接続（参考）

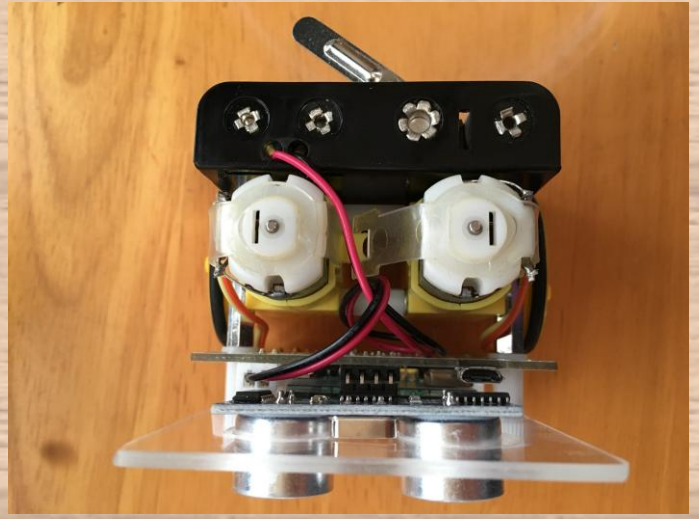

RDC-104 type II
Arduino-IDE上のピン番号
白色LED
LED
13
右モーター
回転方向/停止
回転数（PWM）
9 IN2
4 IN1
6 PWM
左モーター
回転方向/停止
回転数（PWM）
8 IN2
7 IN1
5 PWM
キャラクタLCD
I2C
2 SDA
3 SCL

入力信号の接続（参考）

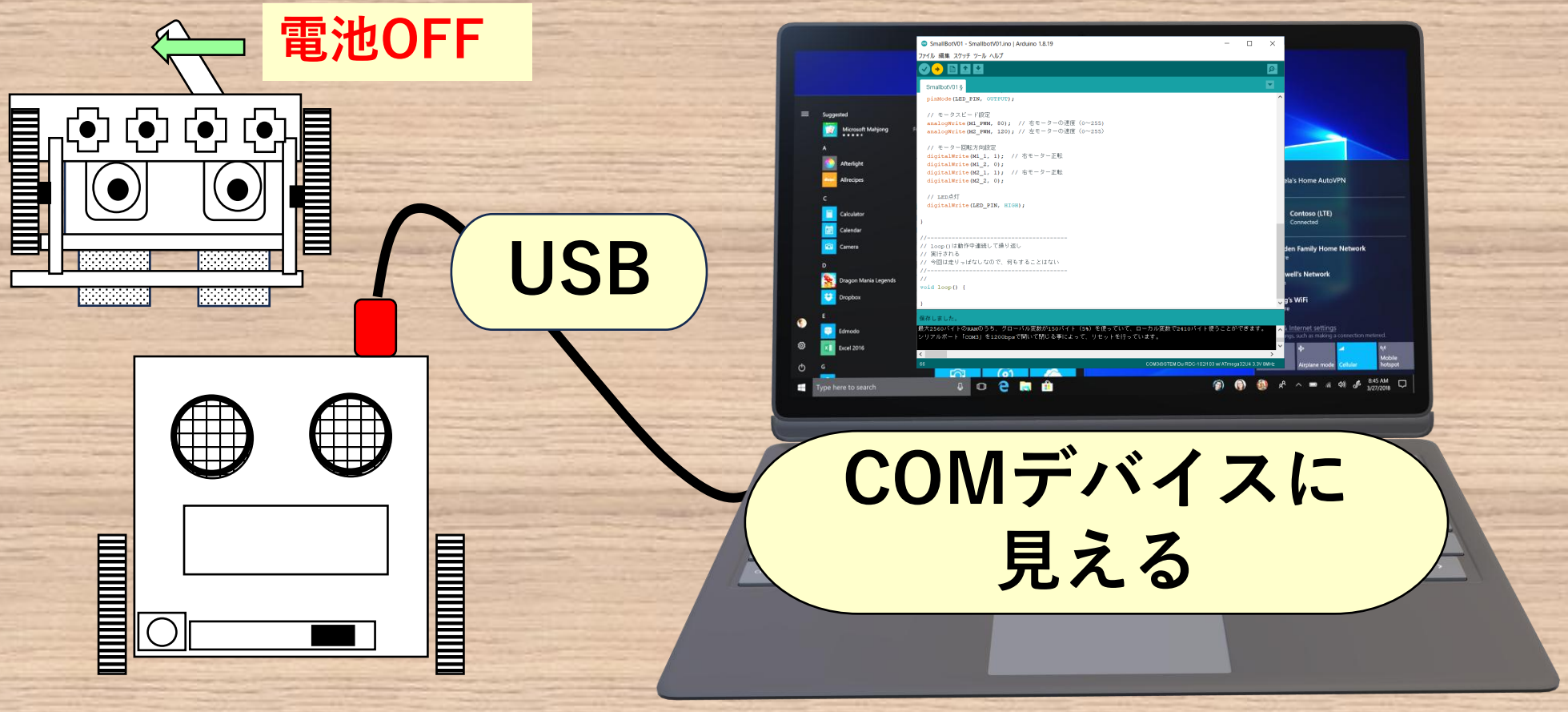

RDC-104 type II
Arduino-IDE上のピン番号
A2
フォトセンサ
A3
スライドボリューム
アナログ
入力
A4
マイク（音声入力）
11
TRIG
ECHO
超音波距離センサ
マイコン
(ATMega32U4)
12
押しボタンスイッチ

書き込んで動かすまでの手順のおさらい

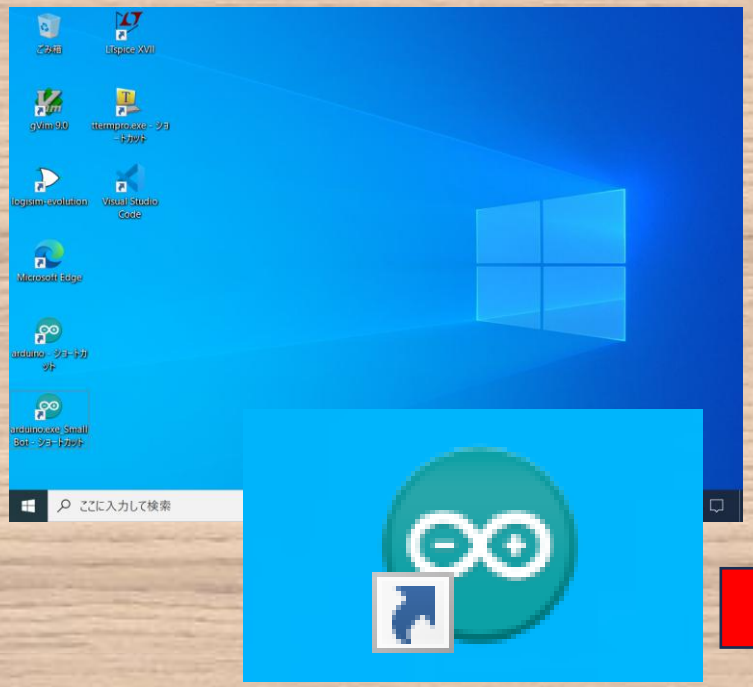




PCと接続



Arduino IDEの起動



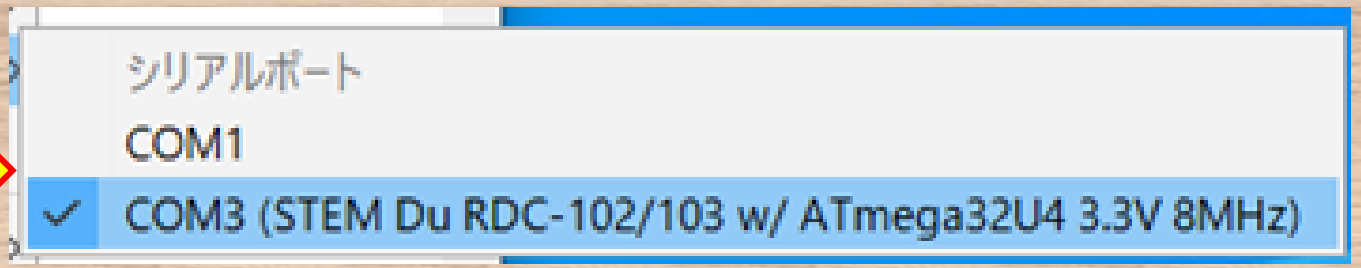
ターゲットの選択

ツール =>ボード =>Arduino AVR Boards
=>STEM Du RDC-104.... を選択

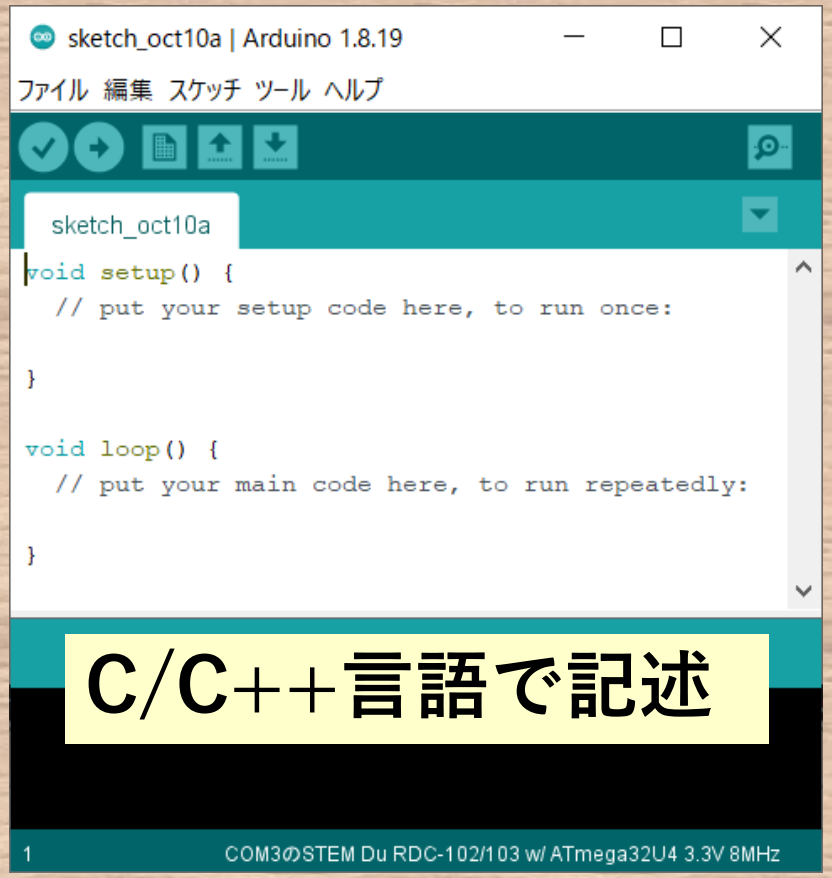
COM（シリアル）ポートの選択

COM(シリアル）ポートの番号はPC環境によって異なる

ツール
=>シリアルポート
=>STEM Du RDC-102/103 w/……



プログラムの記述




```
void setup() {
```

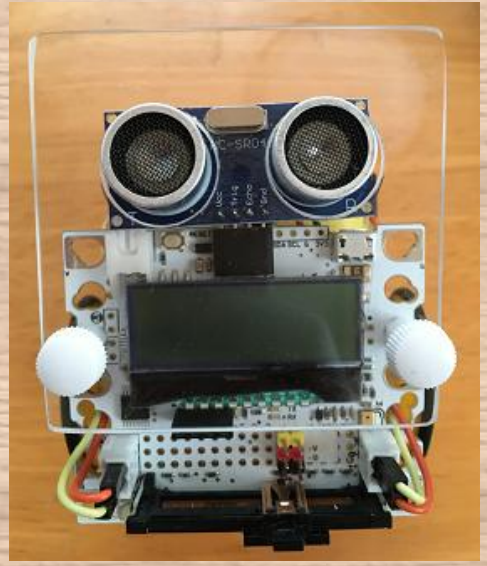
起動時最初に
一回だけ実行される

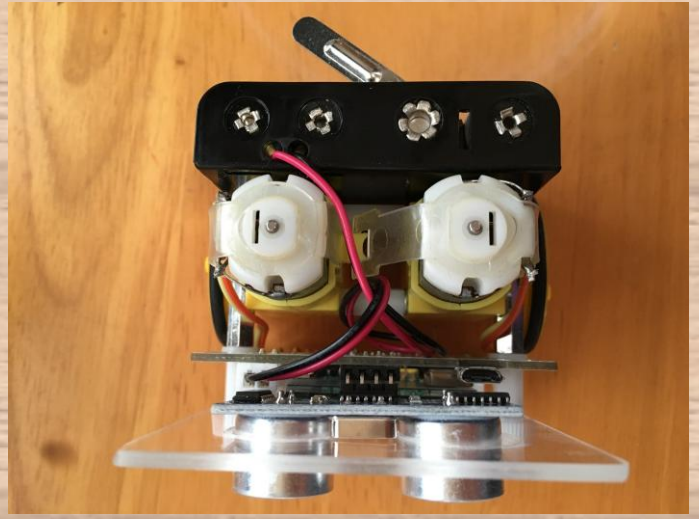
```
}

void loop() {
```

繰り返し実行される

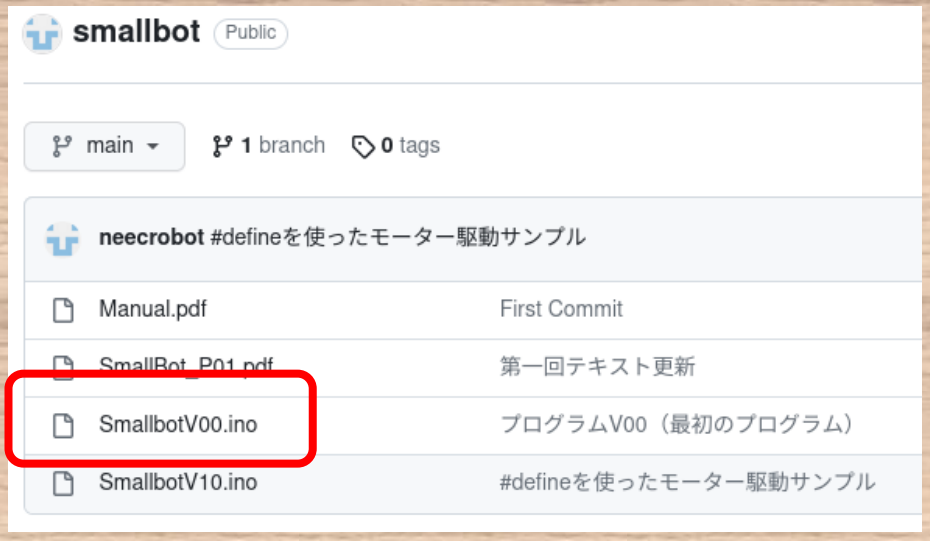
```

}
```

モーターを動かそう

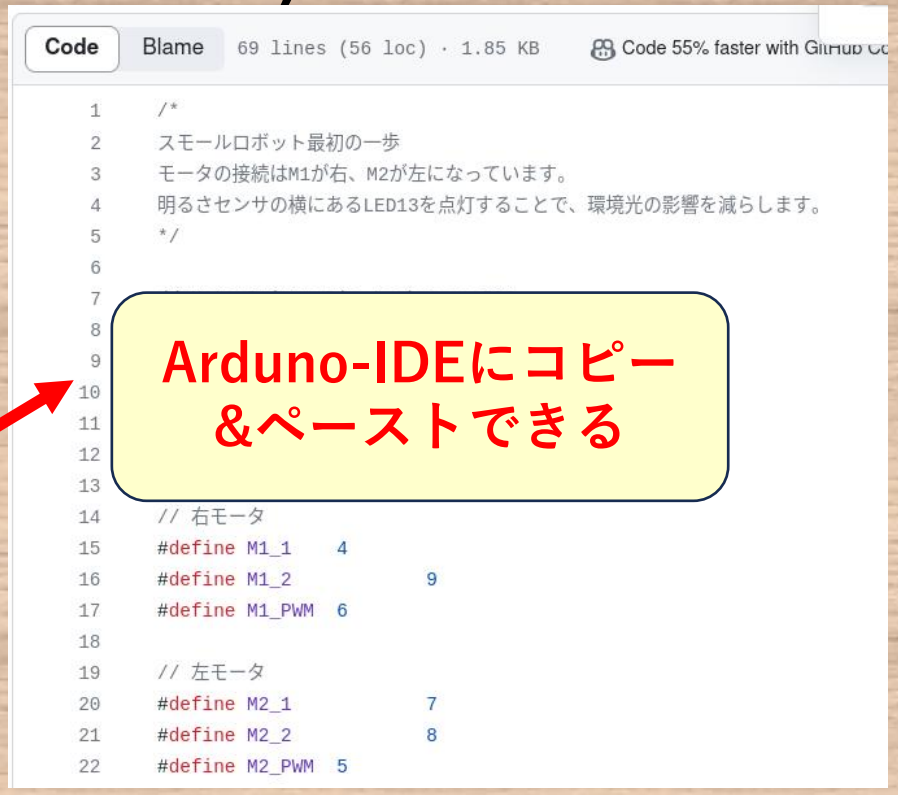




モータ駆動サンプル（SmallbotV00.ino）



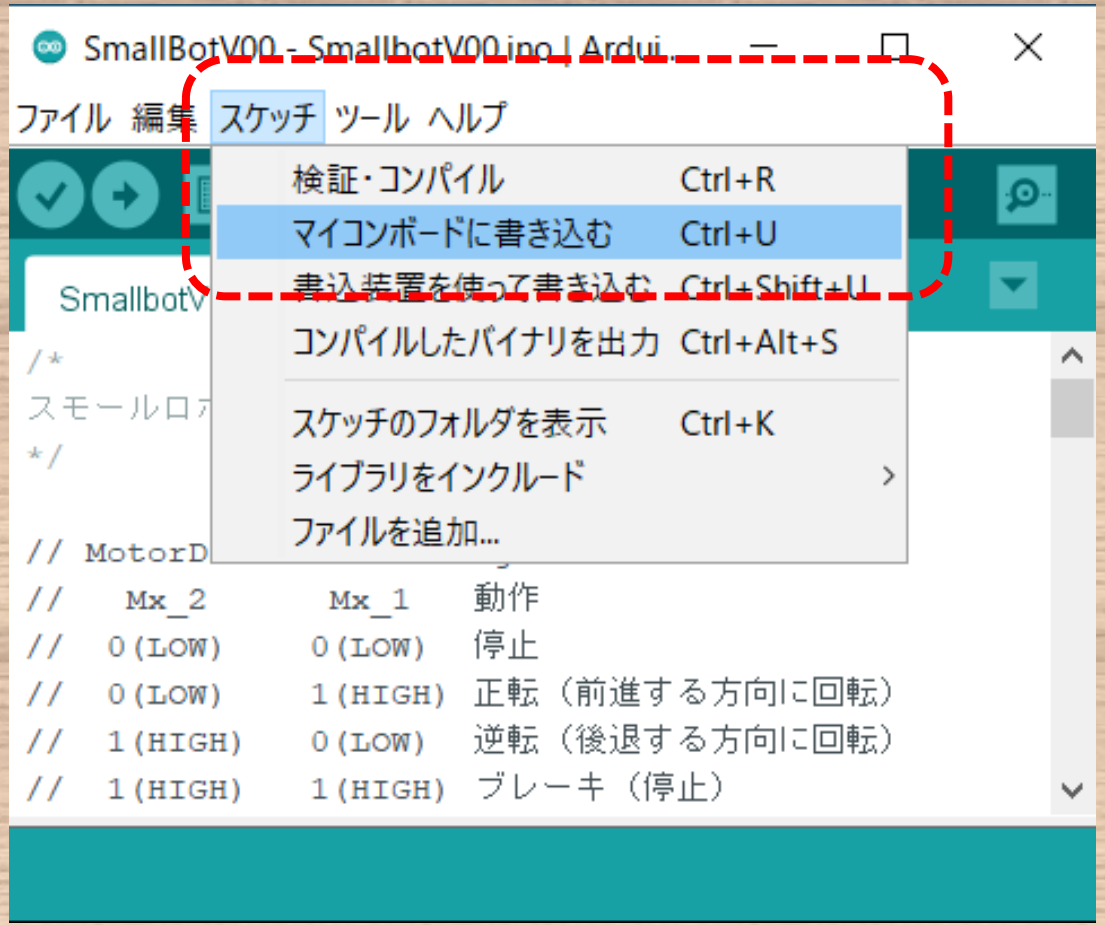
開く



Arduno-IDEにコピー
&ペーストできる

https://github.com/neecrobot/smallbot

ビルド＆書き込み＆実行

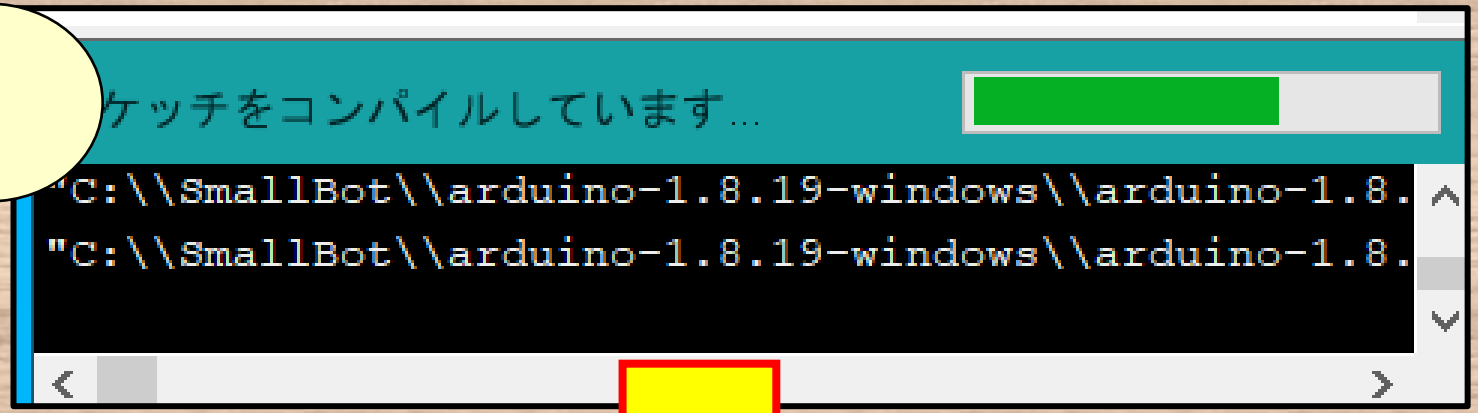


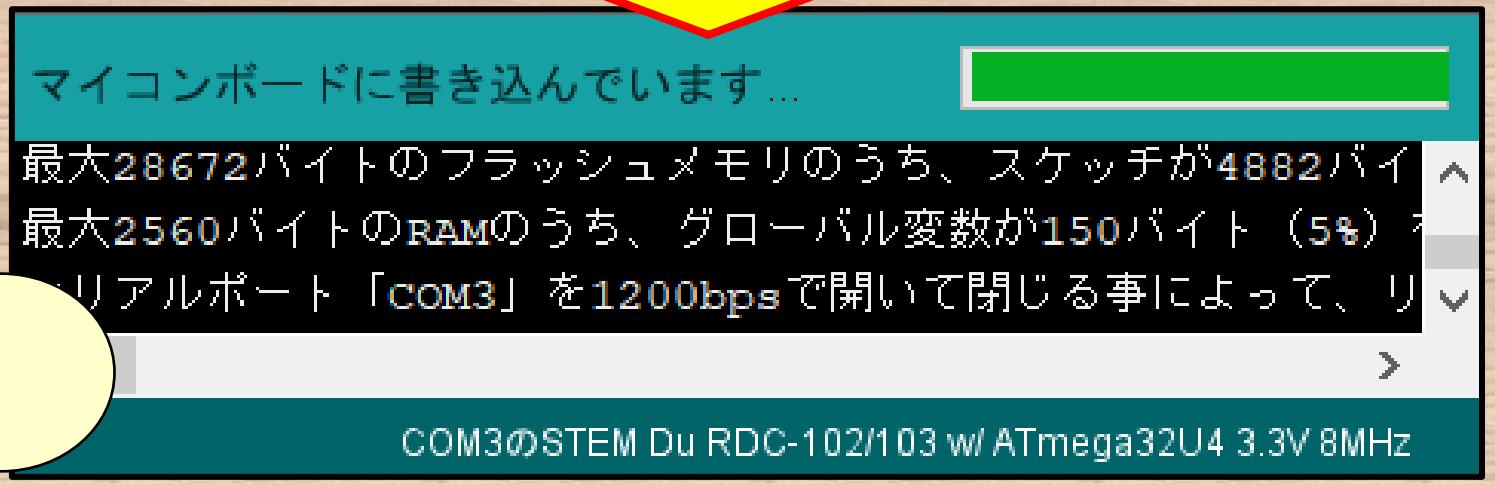
接続したSmallBotに書き込みを行う

ビルドも自動的に行われる

書き込み実行

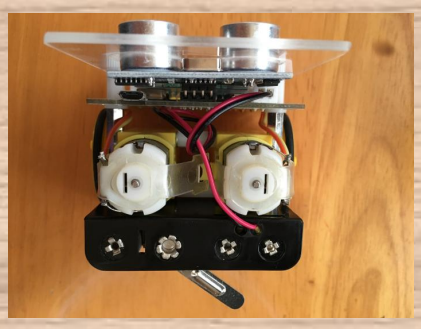
ビルド中

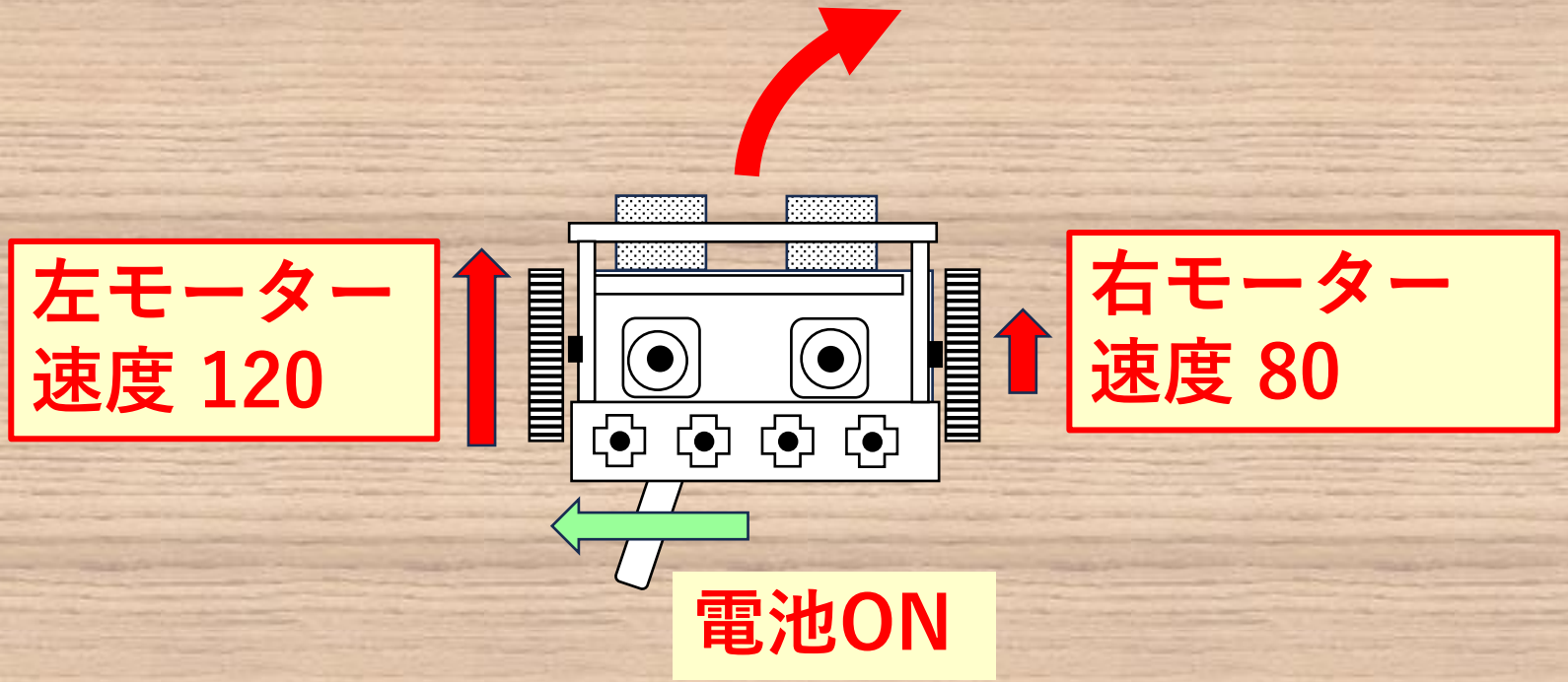




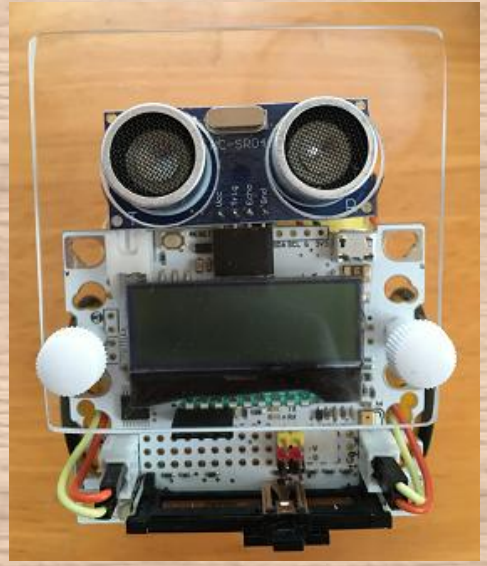
書込完了

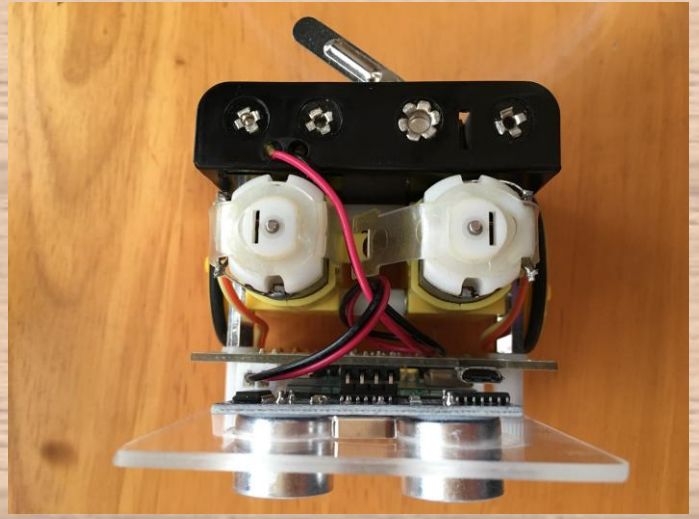
時計回りに回転する





文字列の置き換えマクロ（#define）の使い方





#defineの動作

```
pinMode(4, OUTPUT);
pinMode(9, OUTPUT);
pinMode(6, OUTPUT);
```

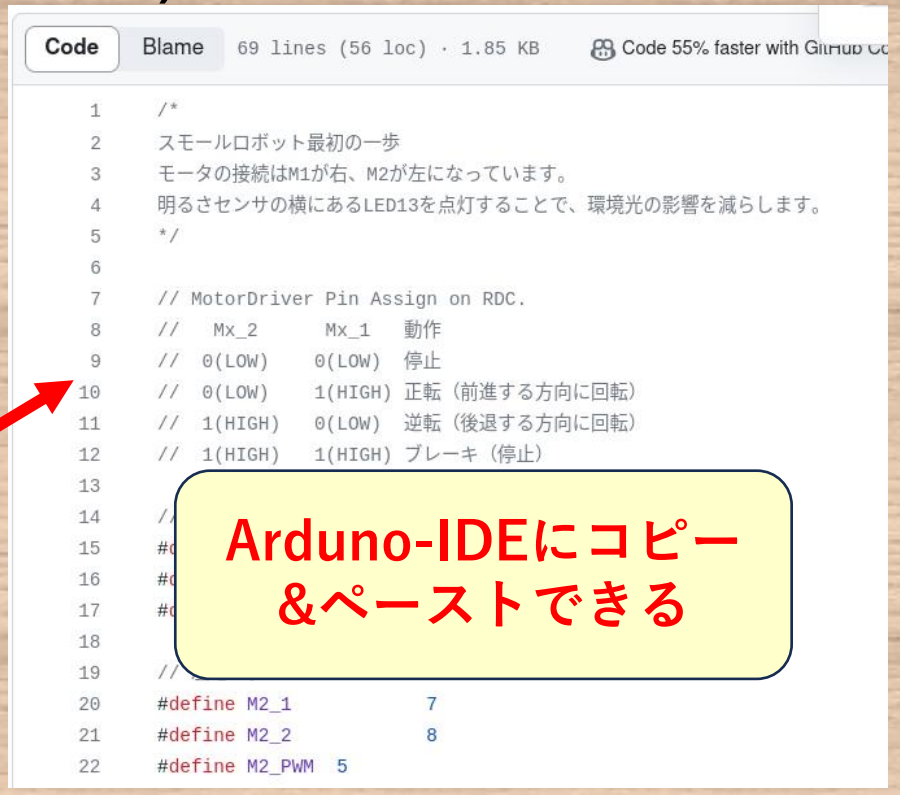
define 置換前　置換後

```
#define M1_1          4
#define M1_2          9
#define M1_PWM        6
pinMode(M1_1, OUTPUT);
pinMode(M1_2, OUTPUT);
pinMode(M1_PWM, OUTPUT);
```

"M1_1"が”4”に置き換えられ、
PinMode(4,OUTPUT)になる

#defineを使ったサンプル
(SmallbotV10.ino)

開く



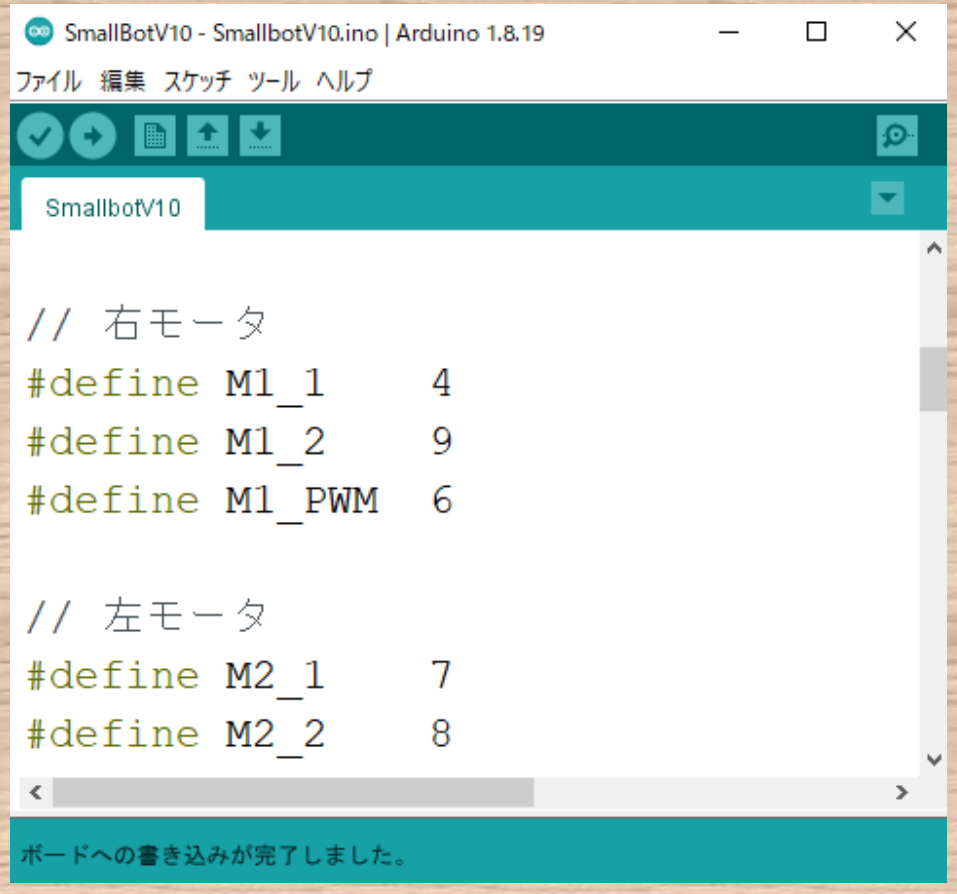
Arduno-IDEにコピー
&ペーストできる

https://github.com/neecrobot/smallbot

コピー＆ペーストして動かしてみよう

Code　Blame　69 lines (56 loc) · 1.85 KB　Code 55% faster with GitHub Co

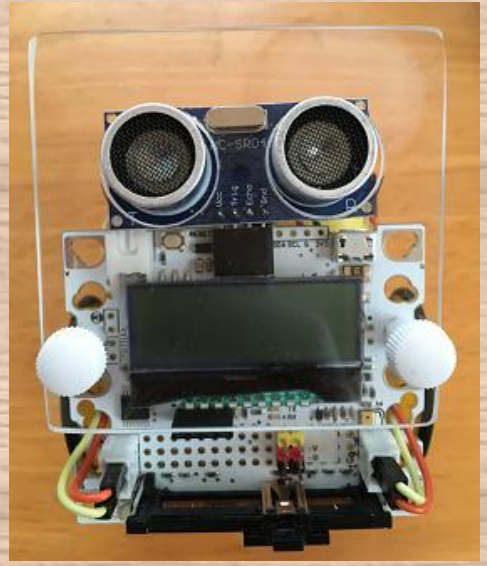
```
/*
スモールロボット最初の一歩
モータの接続はM1が右、M2が左になっています。
明るさセンサの横にあるLED13を点灯することで、環境光の影響を減らします。
*/

// MotorDriver Pin Assign on RDC.
//   Mx_2      Mx_1   動作
//  0(LOW)    0(LOW)  停止
//  0(LOW)    1(HIGH) 正転（前進する方向に回転）
//  1(HIGH)   0(LOW)  逆転（後退する方向に回転）
//  1(HIGH)   1(HIGH) ブレーキ（停止）

// 右モータ
#define M1_1    4
#define M1_2            9
#define M1_PWM  6

// 左モータ
#define M2_1            7
#define M2_2            8
#define M2_PWM  5
```

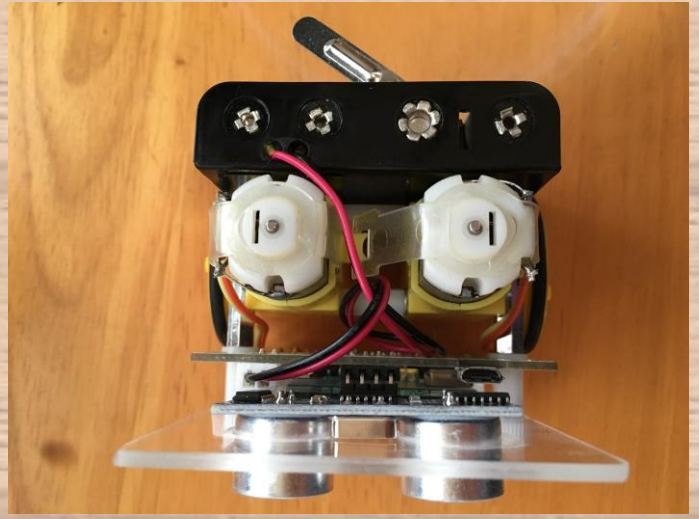


SmallBot用の定義例

```
#define M1_1          4      // 右モータ回転方向
#define M1_2          9
#define M1_PWM        6      // 右モーター速度
#define M2_1          7      // 左モータ回転方向
#define M2_2          8
#define M2_PWM        5      // 左モーター速度
#define PING_PIN      11     // 超音波測距センサ
#define BUTTON_PIN    12     // 押しボタンスイッチ
#define LED_PIN       13     // LED
#define PHOTO         A2     // フォトセンサ
#define SLIDER        A3     // スライダ
```

C言語をちょっとだけ





まずはこれだけ

・整数型変数（int型）

<宣言>

```
int data;
int data = 0; // 初期化あり
```

<代入>

```
data = 3;
data = data+4;
```

・条件判断と分岐

< if 文>

```
 if (data == 3) {
    条件成立時の処理
} else {
   条件不成立時の処理
}
```

==	等しい
!=	等しくない

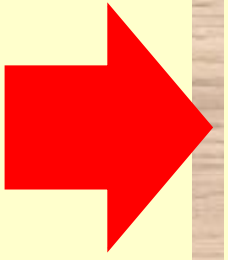
＞	左辺＞ 右辺
＞＝	左辺≧ 右辺

＜	左辺＜ 右辺
＜＝	左辺≦ 右辺

if() ... else if()... else

```
if (data == 3) {
  data = 2;
}
if (data == 2) {
   data = 3;
}
```

=> 3の時も3になってしまう



```
if (data == 3) {
   data = 2;
 } else if (data == 2) {
    data = 3;
 } else {
    data = 0;
 }
```

3のときは　2
2のときは　3
それ以外なら 0

動かしてみよう
(SmallbotV20.ino)

SmallBotV20 | Arduino 1.8.19

ファイル　編集　スケッチ　ツール　ヘルプ

SmallBotV20

```
void loop() {
  if (dir == 0) {     // dirが0だった
    // モータスピード設定
    analogWrite(M1_PWM,  80); // 右
    analogWrite(M2_PWM, 120); // 左
  } else {
    // モータスピード設定
    analogWrite(M1_PWM, 120); // 右
    analogWrite(M2_PWM,  80); // 左
```

GithubのV20.ino
をコピー&ペースト
して実行してみよう

変数とif文の利用（V20.inoから抜粋）

左右交互に繰り返し

```
int dir = 0;
void loop() {
  if (dir == 0) {  // dirが0
    analogWrite(M1_PWM,  80);
    analogWrite(M2_PWM, 120);
    dir = 1;   // 次は左回転
  }
```

変数dirが0なら
　右回転に設定
　dirの値を1にする

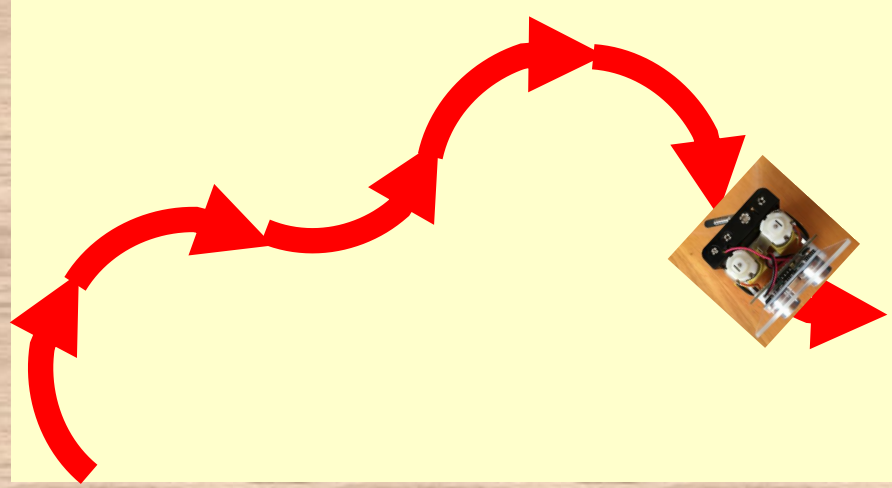
```
else {
    analogWrite(M1_PWM, 120);
    analogWrite(M2_PWM,  80);
    dir = 0;   // 次は右回転
  }
  delay(1000);       // １秒待つ
}
```

変数dirが0でないなら
　左回転に設定
　dirの値を0にする

改造してみよう

1：右に2秒回転
　　左に1秒回転
を繰り返すようにしてみよう

ヒント：”＞” や ”＞＝” の利用



2：　1の動きを4回行った
　　後、停止するように
　　してみよう

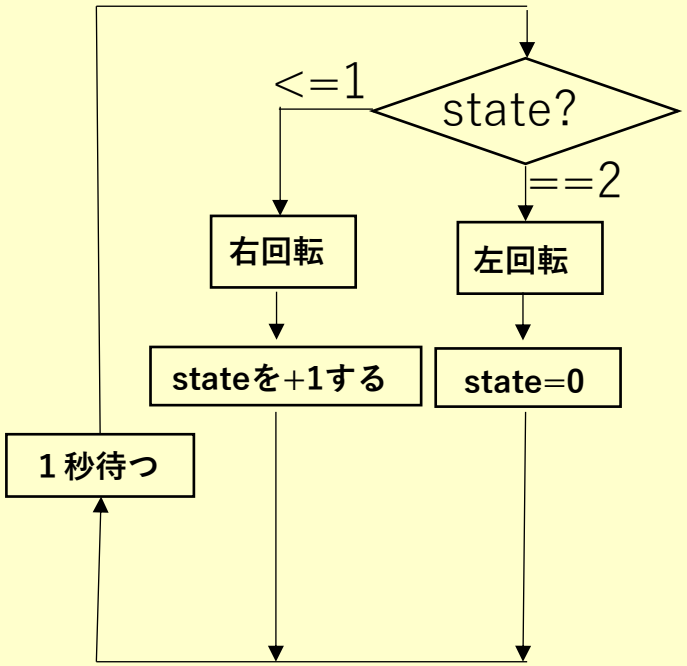
ヒント：0,1,2, 3,4,5, ... ,12

3：　変数を二つ使って2と
　　同じ動きにしてみよう

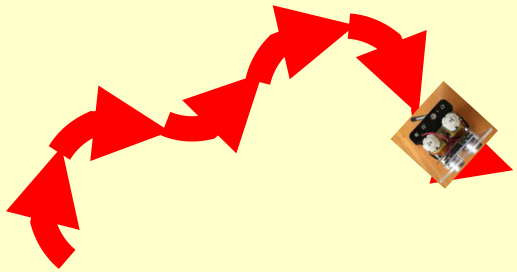
ヒント：　0 : 0,1,2
　　1 : 0,1,2
　　...

考え方の例－1

1：右に2秒回転
　　左に1秒回転
を繰り返すようにしてみよう




```
int state = 0;
void loop() {
  if (state <= 1) { // < 2 でもいい
     <右回転>
    state = state + 1;
  } else {
    ＜左回転＞
    state = 0;
  }
  delay(1000);
}
```



考え方の例－２

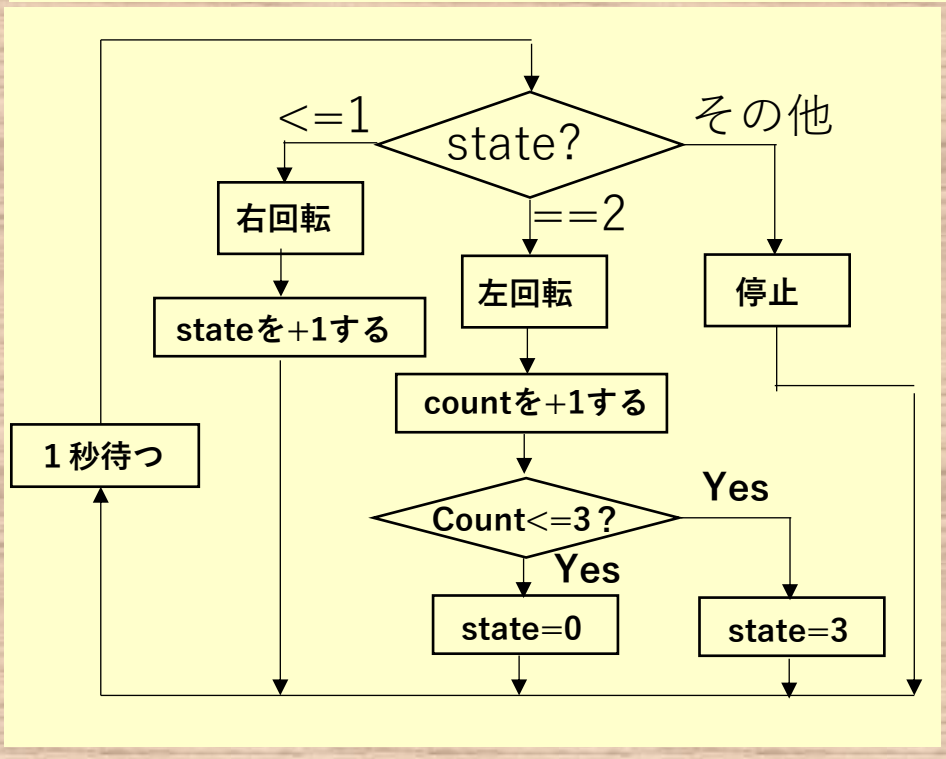
２： 1の動きを4回行った後、停止するようにしてみよう

ヒント：0,1,2, 3,4,5, ... ,12
右右左 右右左 ... ,停止

```
int state = 0;
void loop() {
  if (state <= 1)  //0,1の時
    <右回転>
    state = state + 1;
  }
  else  if (state == 2) { // 2の時
    ＜左回転＞
    state = state + 1;
  } else if (state <= 4) { // 3,4の時
    ＜右回転＞
    state = state + 1;
  } else if・・・・
    ・・・・
  } else if (state == 12) { //終了
    ＜停止＞
}
  delay(1000);
}
```

考え方の例ー３-1

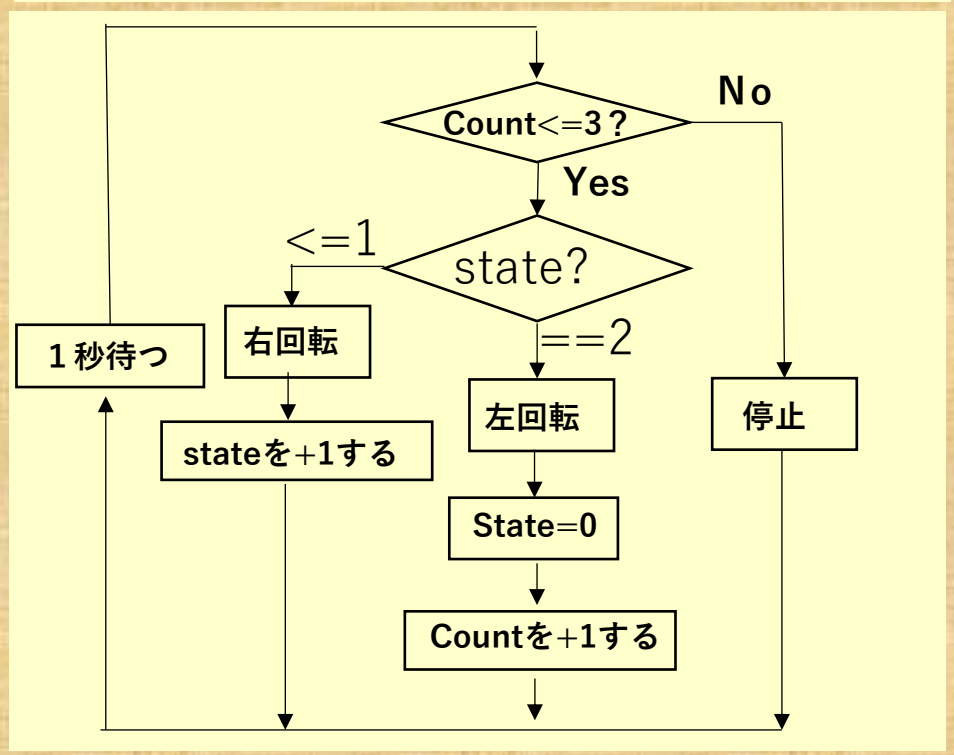
３： 変数を二つ使って2と同じ動きにしてみよう




```
int state = 0;
int count = 0;
void loop() {
   if (state <= 1) { // 0,1なら右回転
    <右回転>
    state = state + 1; // 1増やす
   else  if (state == 2) { //2なら左回転
   ＜左回転＞
      count = count + 1; // 回数カウントを増やす
      if (count <= 3) // 回数カウントが1,2,3なら
         state = 0;    // もう一回
      else             // 4になったら
         state = 3;    // 終わり
   } else {  // 4だったら
    ＜モータ停止＞
   }
    delay(1000); //  １秒待つ
}
```

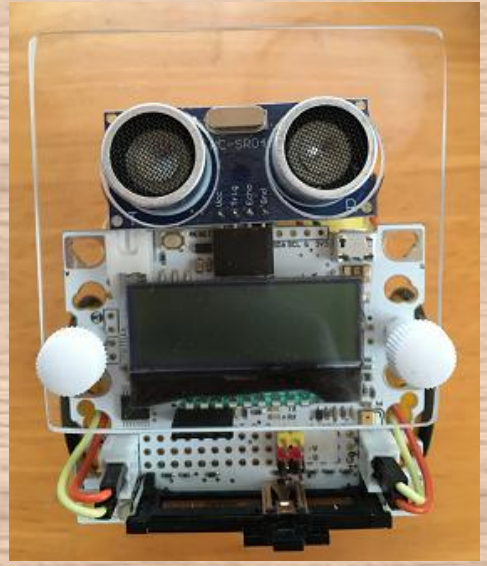
考え方の例ー３-2

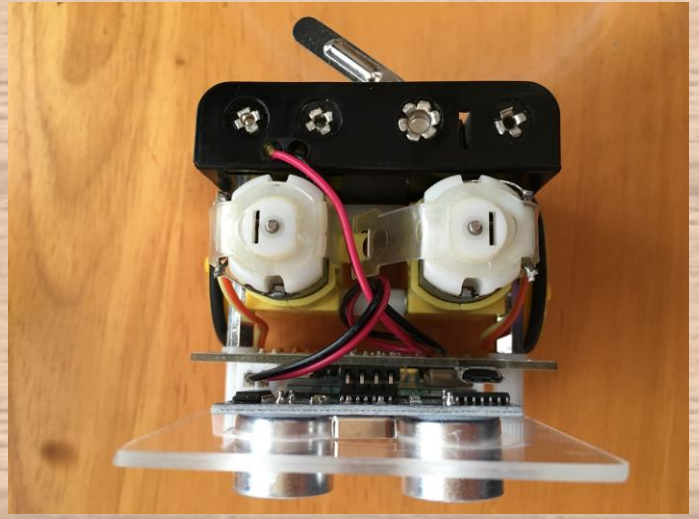
３： 変数を二つ使って2と同じ動きにしてみよう



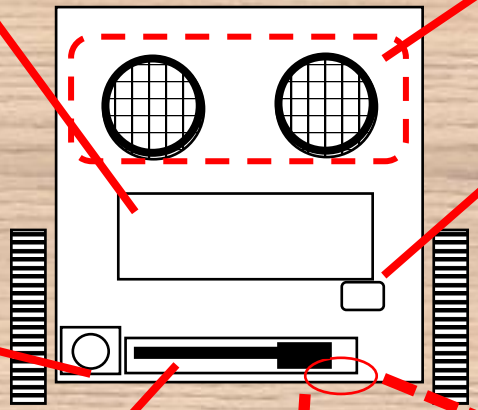

```
int state = 0;
int count = 0;
void loop() {
   if (count <= 3) { //0,1,2,3なら
      if (state <= 1) { // 0,1なら右回転
       <右回転>
        state = state + 1;
      else  if (state == 2) { //2なら左回転
         ＜左回転＞
         state=0;
         count = count + 1; //  １回分終わり
      }
       delay(1000);
   } else { 4回繰り返したら
    ＜モータ停止＞
   }
}
```

フォトセンサを使って
ライントレースさせてみよう





使用するセンサ類



キャラクタLCD

超音波距離センサ

右モーター

マイク

押しボタンスイッチ

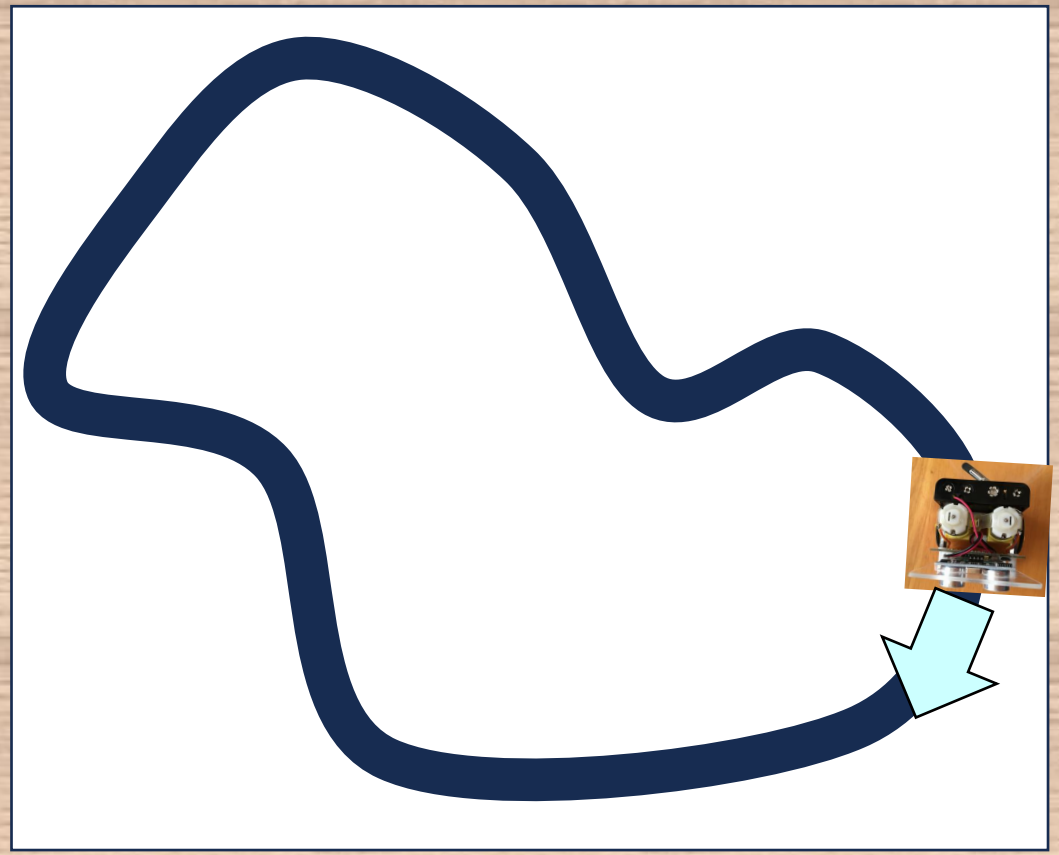
左モーター

スライドボリューム

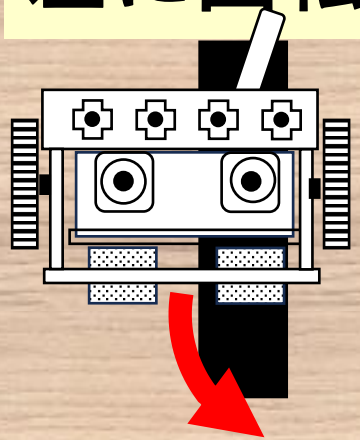
白色LED

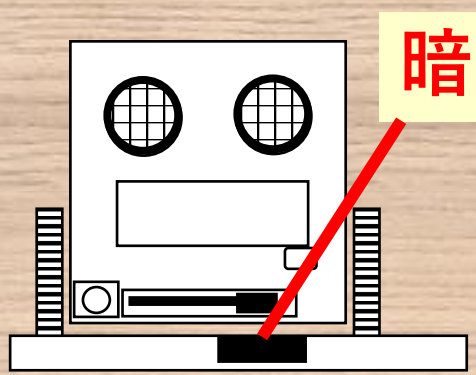
フォトセンサー

ライントレースロボット

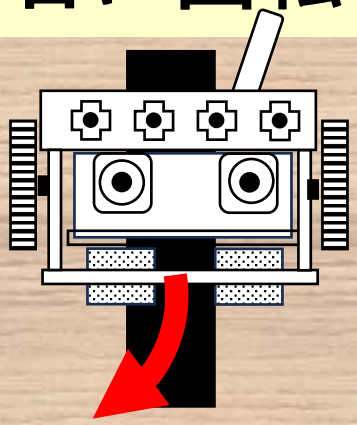


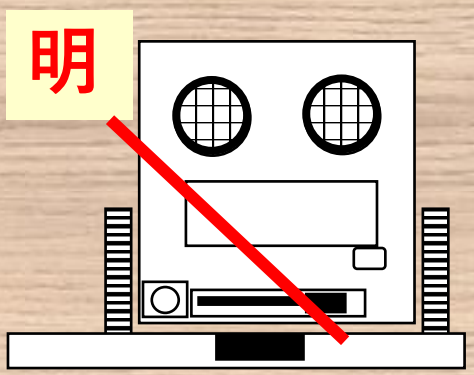
左に回転





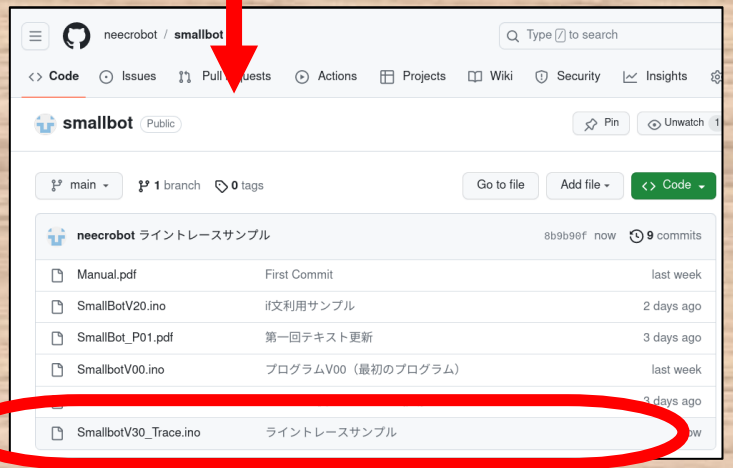
右に回転

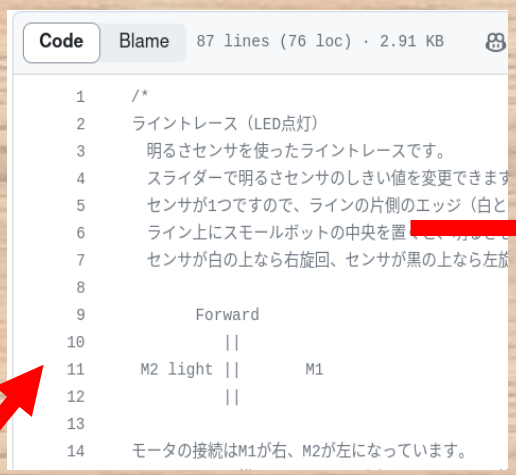


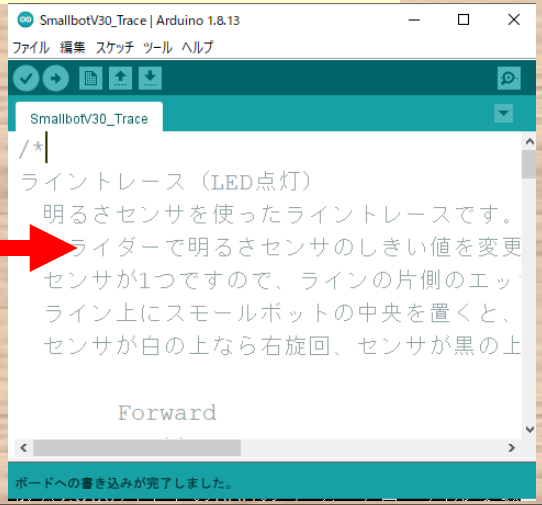


ライントレースサンプル
（SmallbotV30_Trace.ino）

https://github.com/neecrobot/smallbot





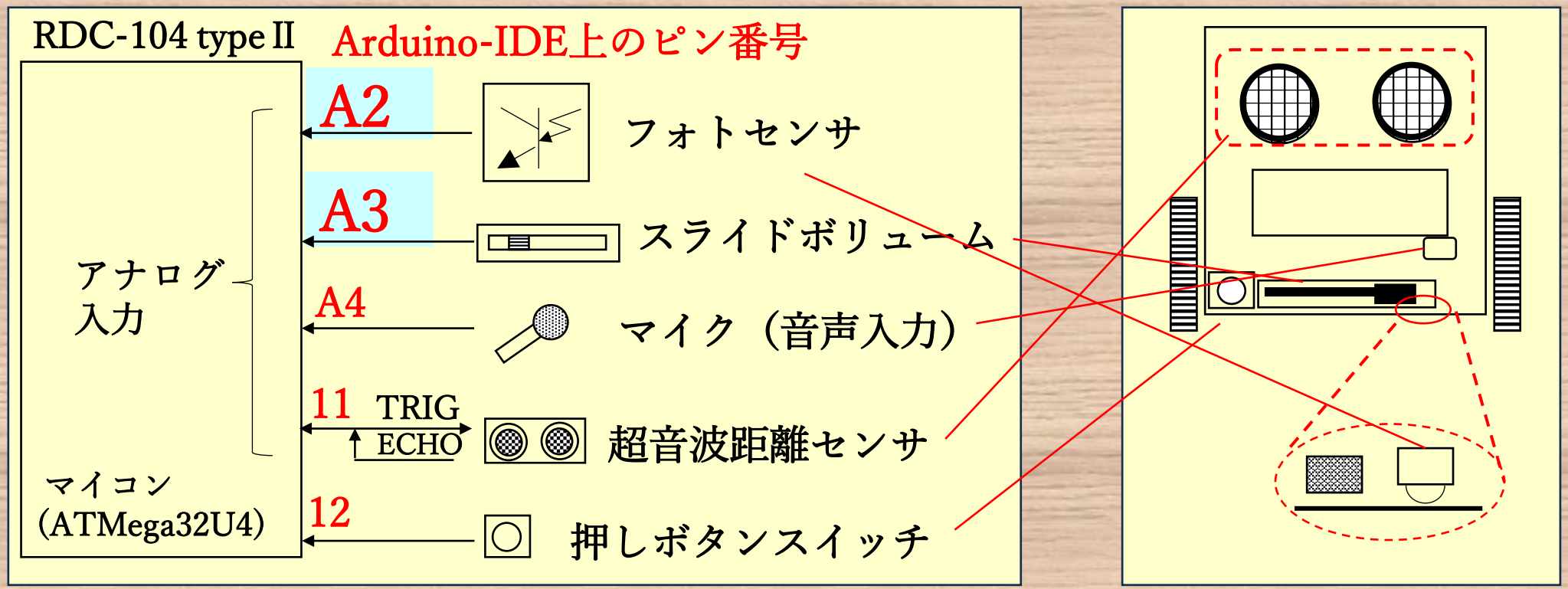


ライントレースのポイント
(SmallbotV30_Trace.inoから抜粋)

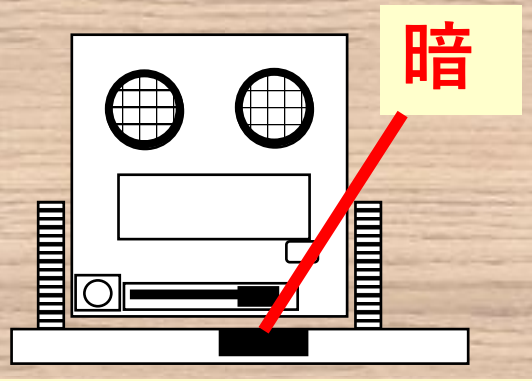
```
int sliderval = 0; // スライダーの値
int photoval = 0; // フォトセンサの値
void loop() {
  sliderval = analogRead(SLIDER);
  photoval = analogRead(PHOTO);
  if (photoval > sliderval) { // 右旋回
    analogWrite(M1_PWM,  80);
    analogWrite(M2_PWM, 120);
  }
  else {  // 左旋回
    analogWrite(M1_PWM, 120);
    analogWrite(M2_PWM,  80);
  }
  delay(100); // 0.1秒現状維持
}
```

フォトセンサの値をスライダーの値と比較して、線の上か否かを決めている

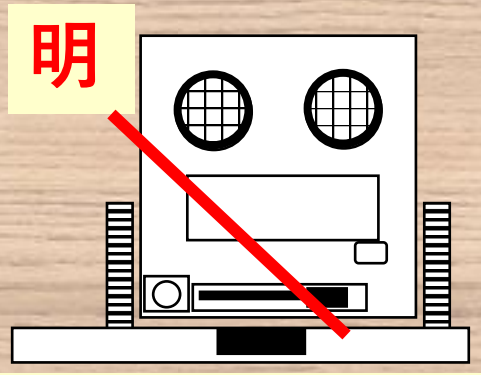
入力信号の接続


RDC-104 type II
Arduino-IDE上のピン番号
A2
フォトセンサ
A3
スライドボリューム
アナログ
入力
A4
マイク（音声入力）
11
TRIG
ECHO
超音波距離センサ
マイコン
(ATMega32U4)
12
押しボタンスイッチ

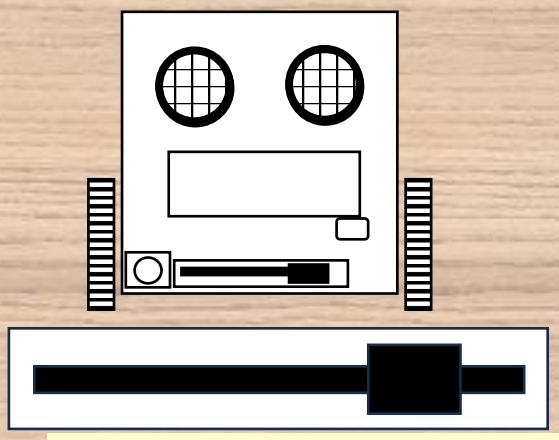
センサ類の値の取得と大小
(analogRead()の値は0～1023)



フォトセンサ値小



フォトセンサ値大



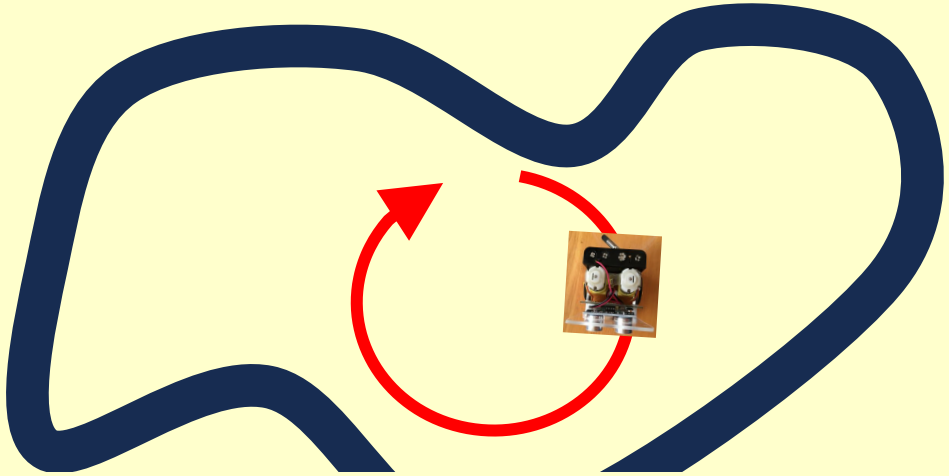
スライダ値 小

スライダ値 大

```
#define PHOTO A2
photoval = analogRead(PHOTO);
#define SLIDER A3
sliderval = analogRead(SLIDER);
```

改造してみよう

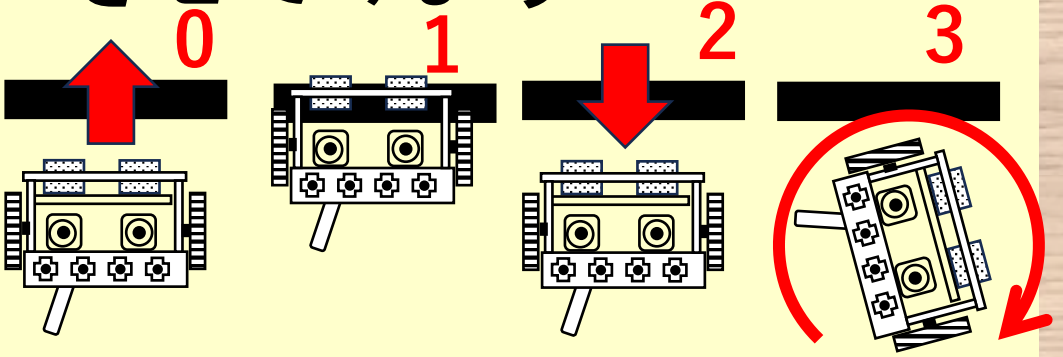
1：線が見つからない時
回転し続けずに線を
探しに行くには？



ヒント：途中で方向を変える

2：通常は直進

黒い線(停止線)を検出
=>1秒間後退して
1秒間右回転
させてみよう

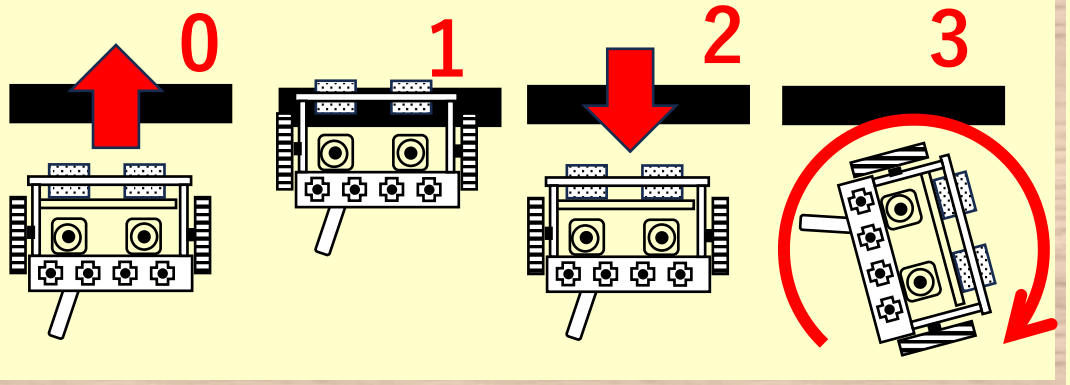


考え方の例－１

１：線が見つからない時
回転し続けずに線を
探しに行くには？

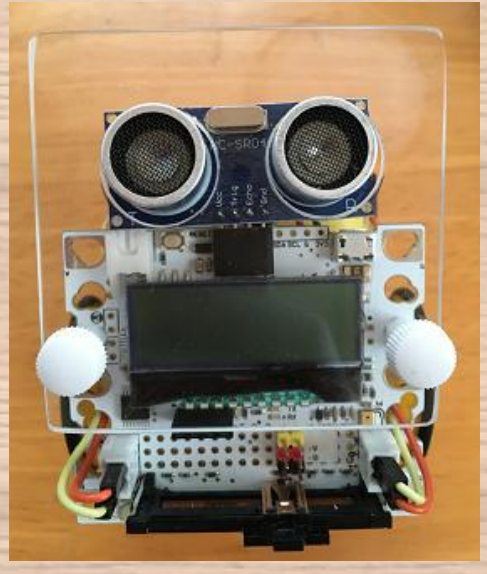
```
count = 0;
void loop() {
....
if (photoval>sliderval) { //白地
    count = count+1;
    if (count <= 20) { //0.1×20= ２秒
        ＜右旋回＞
    }
    else if (count < 40) {//0.1×40= 4秒
        ＜左旋回＞
    } else {
        count = 0; // 右旋回やりなおし
    }
  } else { // 黒地
    count = 0;
   ＜左旋回＞
  }
  delay(100);   //0.1秒
}
```

考え方の例－２




```
count = 0;
void loop() {
if (state == 0) { //直進中
  if (photoval＞sliderval) { //白線
    ＜直進＞
  } else { // 黒線
    count = 10; // タイマ
     state = 2;    // 後退
  }
} else if (state == 2) {
    ＜後退＞
   count = count-1;
    if (count == 0) {
        count = 10;  state = 3;
    }
   . . . . .
}
```

お疲れ様でした



次回はマイクと超音波センサを使ってみましょう