

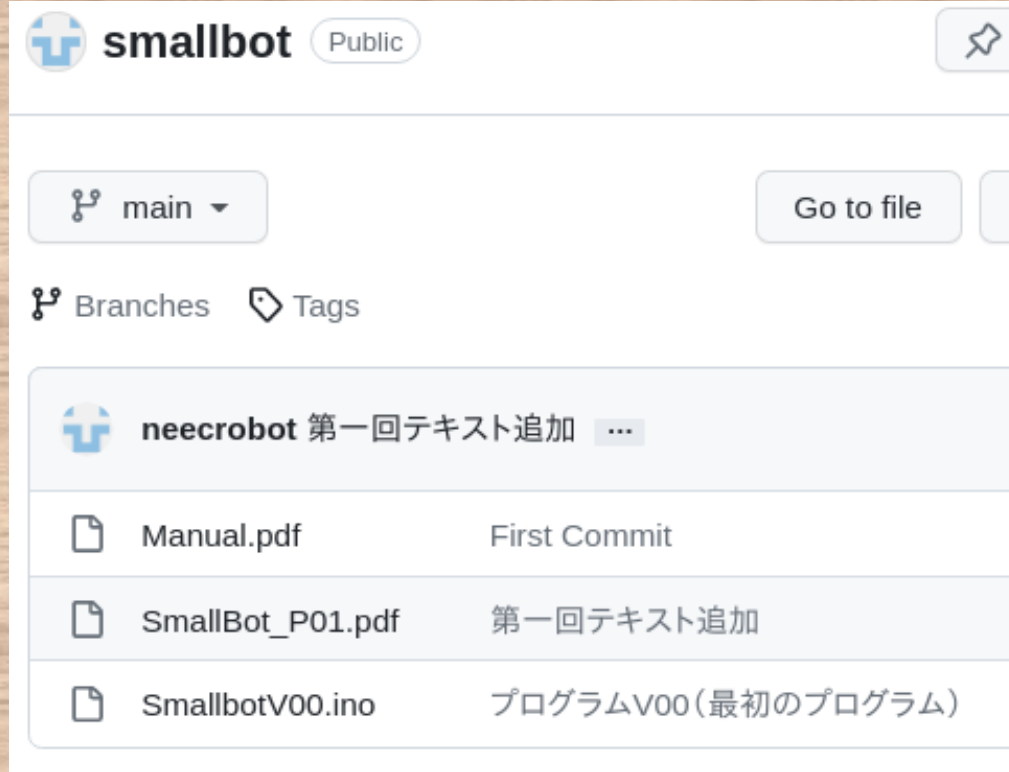
ロボットプログラミング#3



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
- ・ 超音波／音声の利用

マニュアルやプログラム類は githubに置いてあります

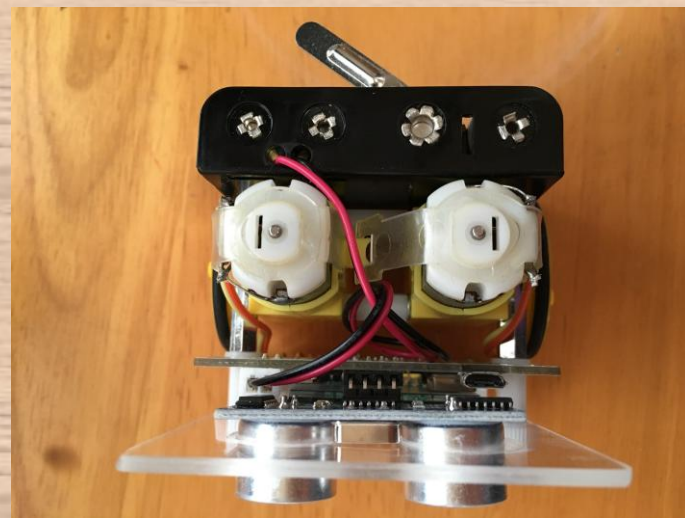


置いてあるファイル
はクリックして
ダウンロード可

<https://github.com/neecrobot/smallbot>

前回のおさらい

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

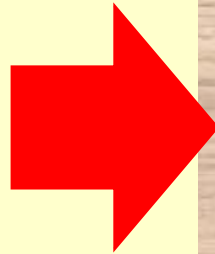
> 左辺 > 右辺
> = 左辺 ≥ 右辺

< 左辺 < 右辺
< = 左辺 ≤ 右辺

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

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

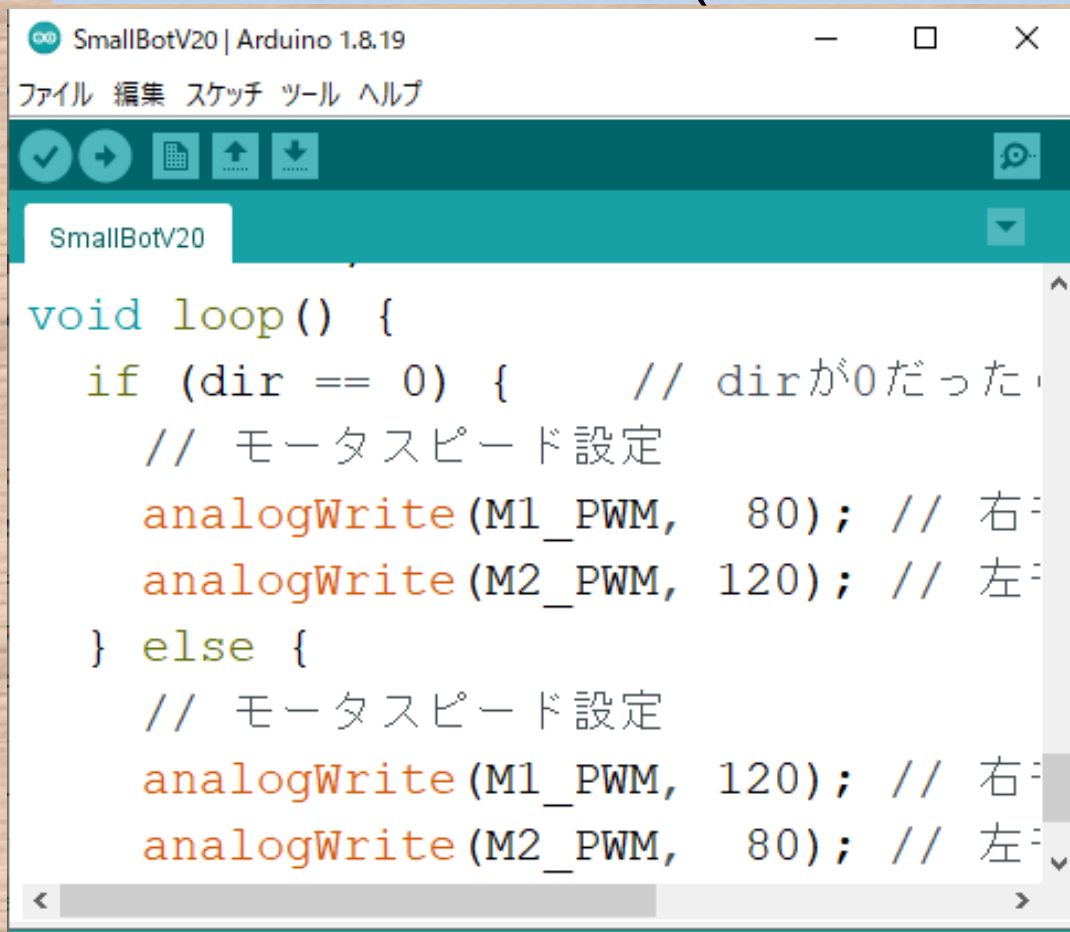
=> 3の時も3になってしまう



```
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

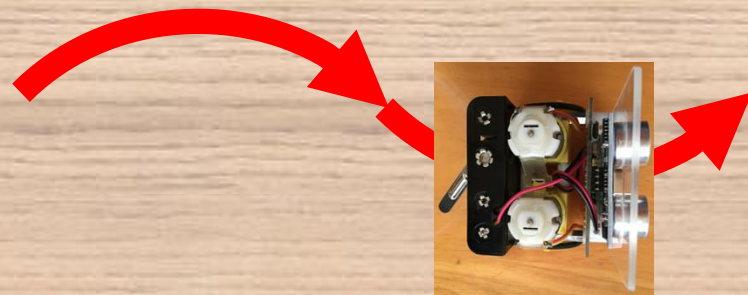
動かしてみよう (SmallbotV20.ino)



The screenshot shows the Arduino IDE interface with the file 'SmallBotV20' open. The code in the main editor area is as follows:

```
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

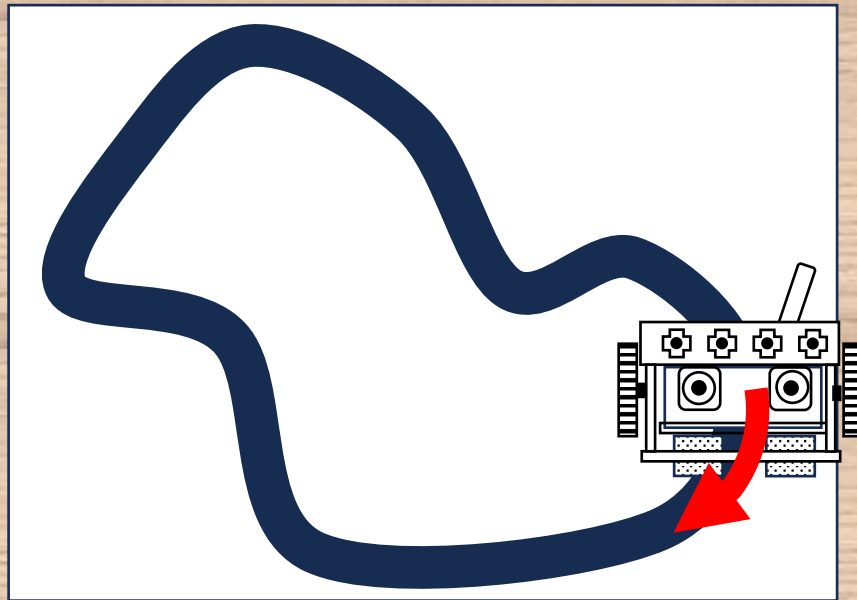
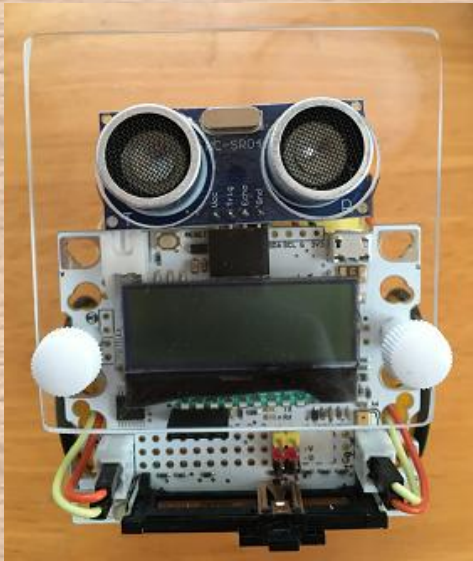
```
else {
  analogWrite(M1_PWM, 120);
  analogWrite(M2_PWM, 80);
  dir = 0; // 次は右回転
}
delay(1000); // 1秒待つ
}
```

変数dirが0でないなら
左回転に設定
dirの値を0にする

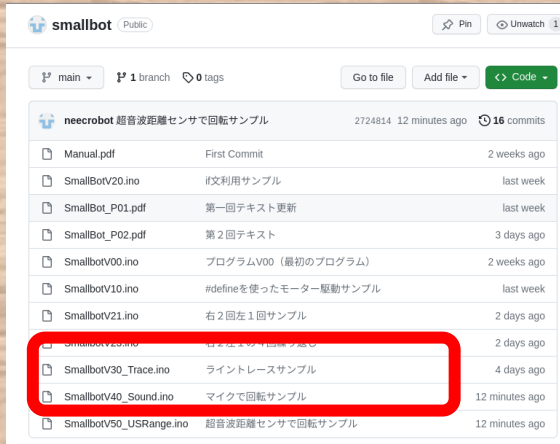
前回のおさらいー 2

ライントレース

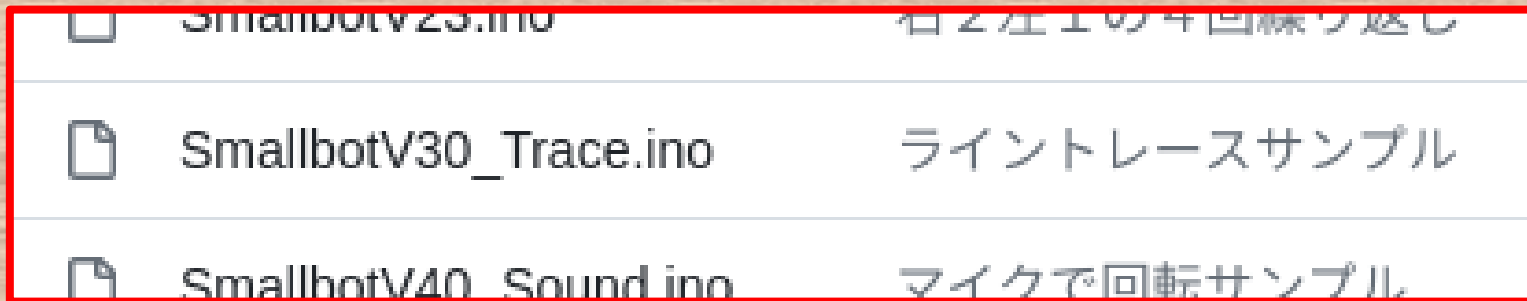
- ・ フォトセンサを使った
ライントレース



動かす & 調整してみよう

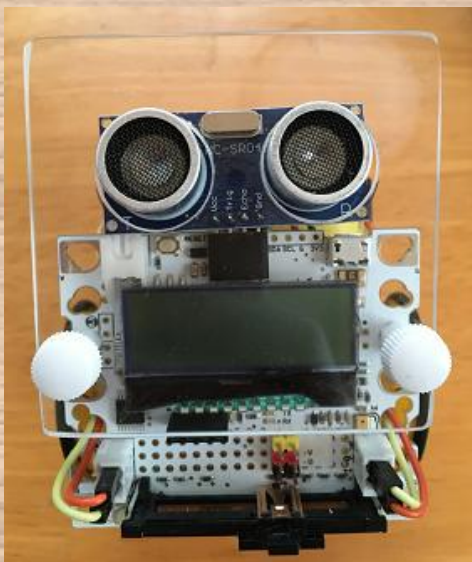


- SmallBotV30_Trace.ino
をベースにする
=> 回転半径を小さくする
=> モーターの回転速度の調整



<https://github.com/neecrobot/smallbot>

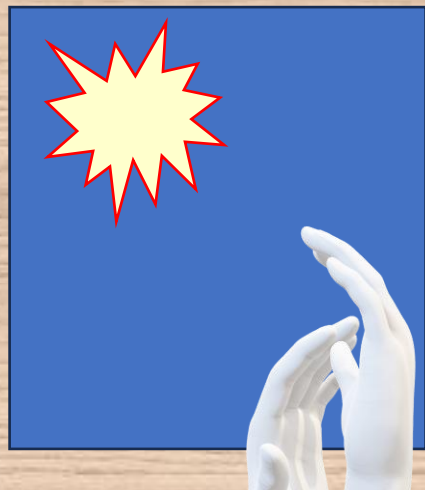
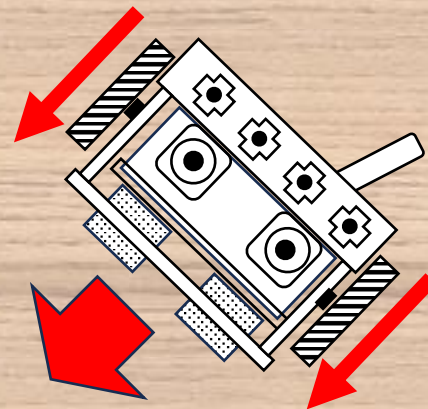
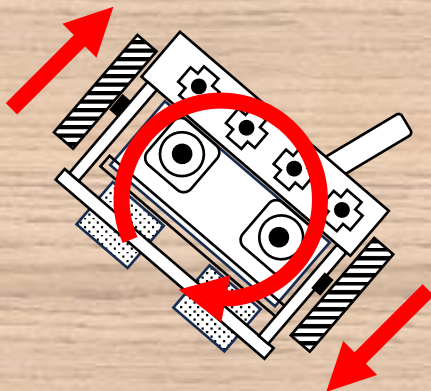
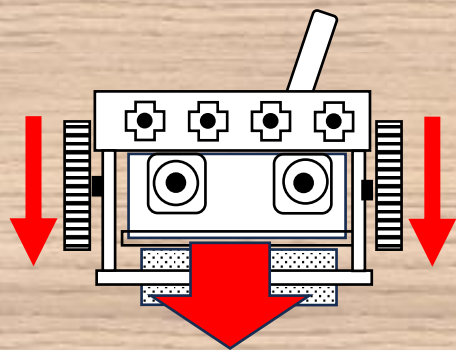
今回のネタ



- ・ 音声入力（拍手）で方向転換
- ・ 超音波センサ（壁検出）で方向転換

本日のネタ（その1）

音声入力（拍手）で方向転換



その場で方向転換
（超信地旋回）

再び直進

動かしてみよう (SmallbotV40_Sound.ino)

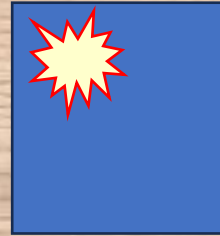
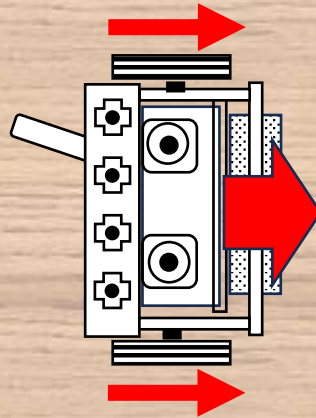
smallbot Public

main 1 branch 0 tags

Go to file Add file <> Code

neecrobot 超音波距離センサで回転サンプル 2724814 12 minutes ago 16 commits

Manual.pdf	First Commit	2 weeks ago
SmallBotV20.ino	#文利用サンプル	last week
SmallBot_P01.pdf	第一回テキスト更新	last week
SmallBot_P02.pdf	第2回テキスト	3 days ago
SmallbotV00.ino	プログラムV00 (最初のプログラム)	2 weeks ago
SmallbotV10.ino	#defineを使ったモーター駆動サンプル	last week
SmallbotV21.ino	右2回左1回サンプル	2 days ago
SmallbotV23.ino	右2左1の4回繰り返し	2 days ago
SmallbotV30_trace.ino	ライトレースサンプル	4 days ago
SmallbotV40_Sound.ino	マイクで回転サンプル	12 minutes ago
SmallbotV50_USRange.ino	超音波距離センサで回転サンプル	12 minutes ago



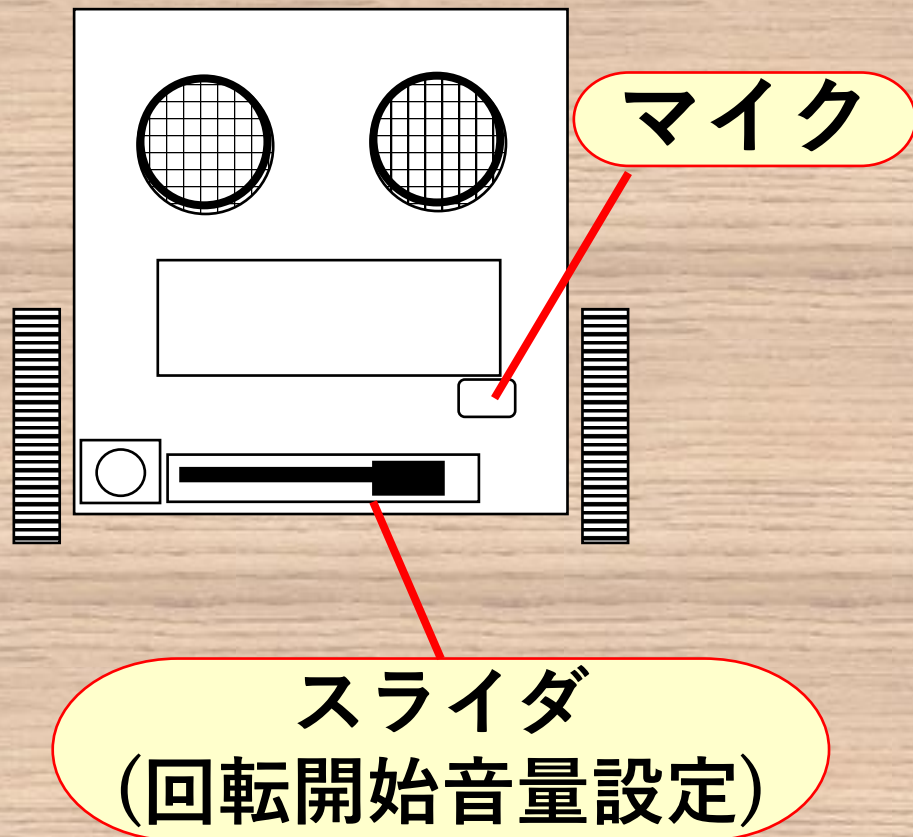
SmallbotV40_Sound.ino

マイクで回転サンプル

<https://github.com/neecrobot/smallbot>

動作のポイント

(SmallbotV40_Sound.ino)



マイクから入力された音声の大きさとスライダで設定したレベルを比較

マイク音量が小さい
=>直進

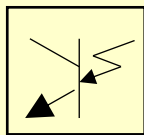
マイク音量が大きい
=>方向転換（右旋回）

入力信号の接続（参考）

RDC-104 type II

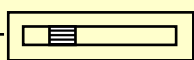
Arduino-IDE上のピン番号

A2



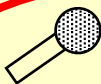
フォトセンサ

A3



スライドボリューム

A4



マイク（音声入力）

11

TRIG

ECHO



超音波距離センサ

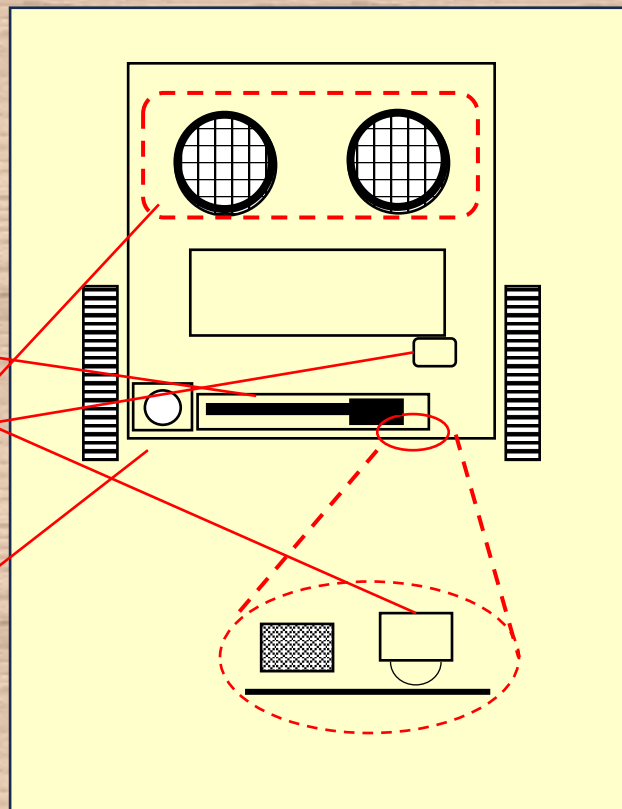
12



押しボタンスイッチ

アナログ
入力

マイコン
(ATMega32U4)



プログラムのポイント

```
insound = analogRead(A4);  
if (SoundLevel < insound) {  
    SoundLevel = insound;  
}
```

```
// 音声レベルを取得  
// 音が今までより大きかったら  
// レベル更新
```

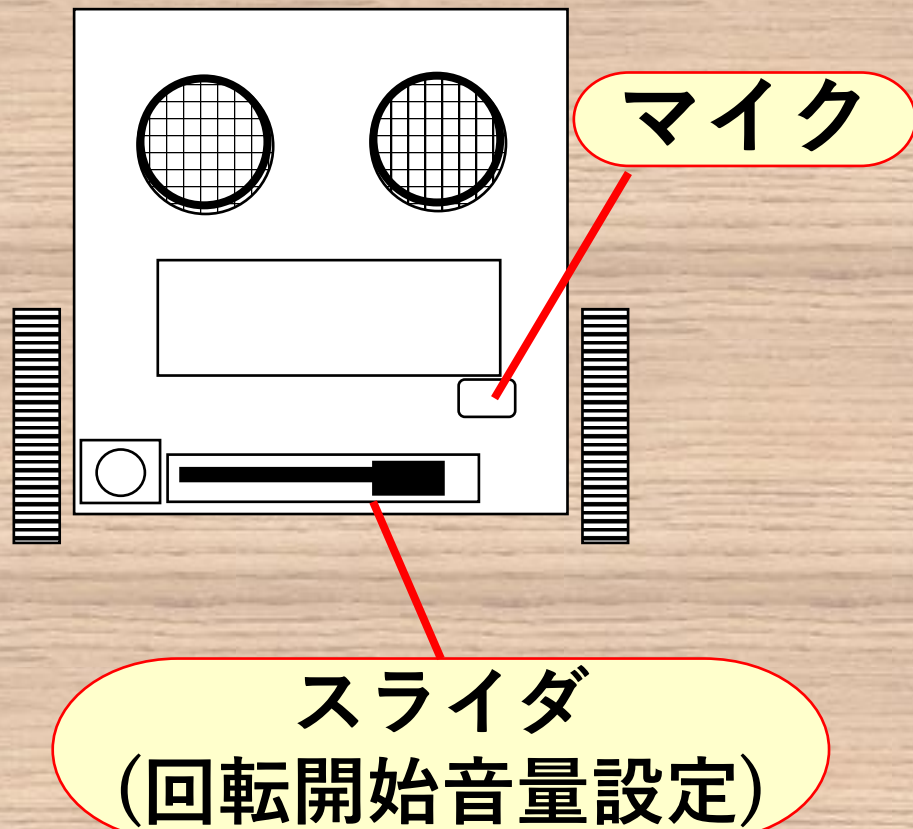
....

```
} else if (RState == RBT_MEASURE) { // 音声を計測  
    sliderValue = analogRead(A3)/3; // スライダー値を取得して  
    if (sliderValue < SoundLevel) { // マイクの音声大きい  
        RState = RBT_ROT; // 回転  
    } else { // マイクの音声小さい  
        RState = RBT_FORWARD; // 前進  
    }
```

```
SoundLevel = 0;
```

```
// 最大音量をクリア
```

改造してみよう



マイクから入力された音が

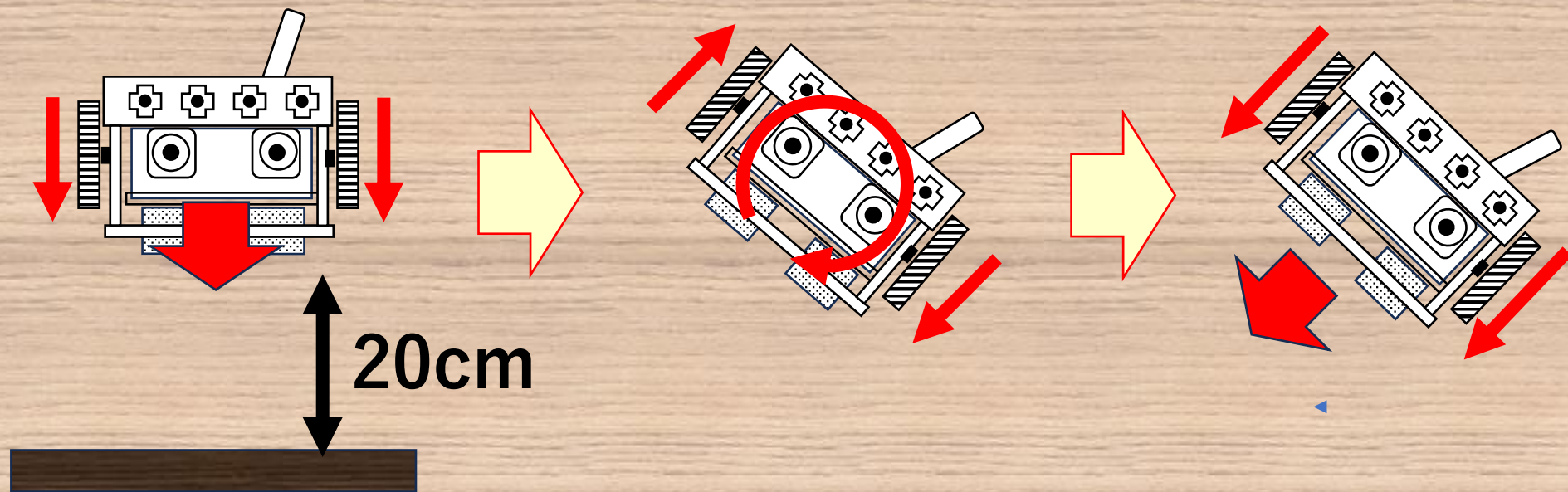
- ・ 小さい時：回転
- ・ 大きい時：直進

(手を叩かれると進む)

前に進む距離をTimer変数で調整しよう

本日のネタ（その2）

超音波距離センサで方向転換



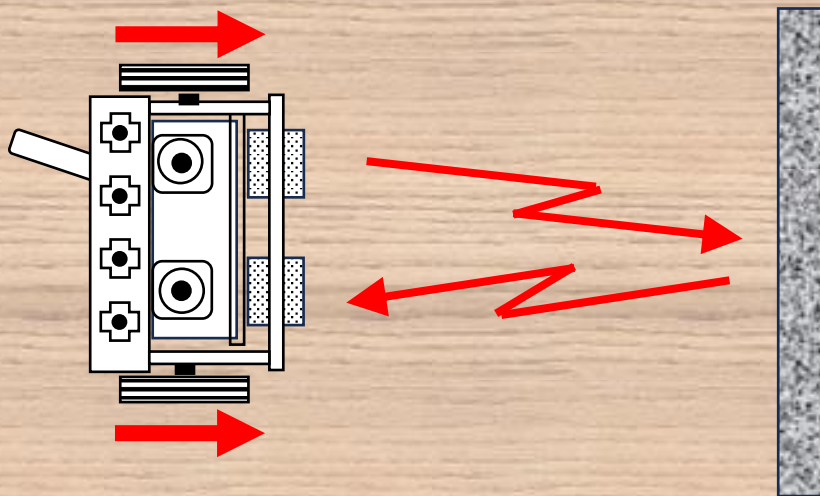
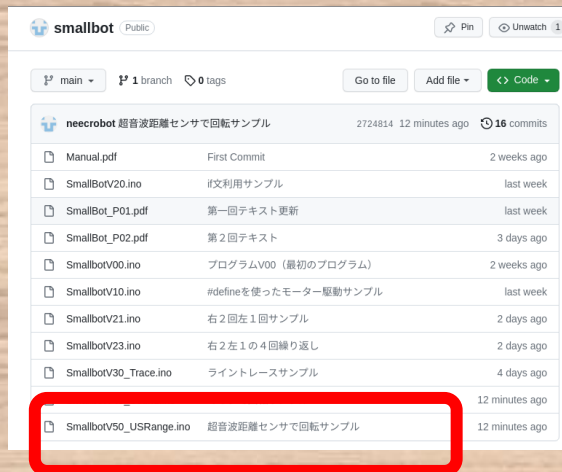
障害物に20cm
まで接近

その場で方向転換
(超信地旋回)

再び直進

動かしてみよう

(SmallbotV50_USRange.ino)



SmallbotV50_USRange.ino

超音波距離センサで回転サンプル

<https://github.com/neecrobot/smallbot>

動作のポイント

(SmallbotV50_USRange.ino)

超音波距離センサ

超音波距離センサで反射物
までの距離を測定

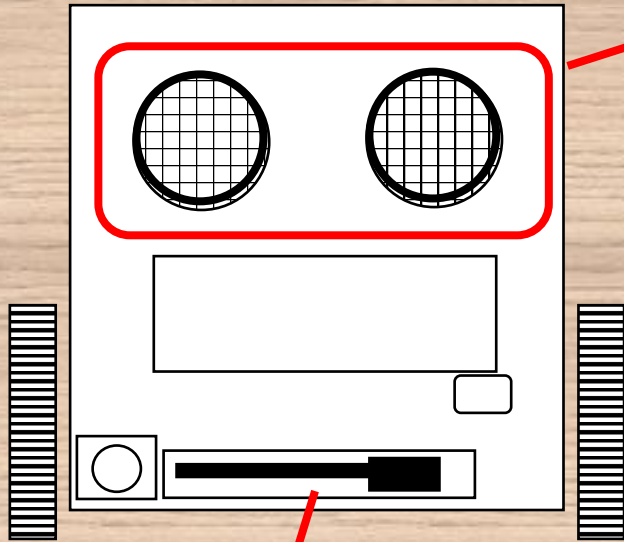
20cm以上 :

=>直進

20cm以下 :

=>旋回

旋回時間はスライダで調整



旋回時間調整

距離計測

```
pinMode(PING_PIN, OUTPUT);    // 出力モードに設定して
```

```
digitalWrite(PING_PIN, LOW);    // 'L'にして  
delayMicroseconds(2)           // 2  $\mu$ s待つ  
digitalWrite(PING_PIN, HIGH);   // 'H'にして  
delayMicroseconds(5);           // 5  $\mu$ s待つ  
digitalWrite(PING_PIN, LOW);    // 'L'に戻す
```

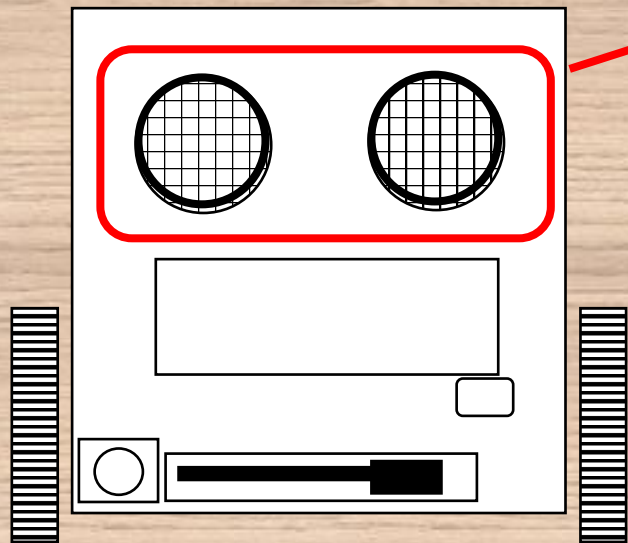
```
pinMode(PING_PIN, INPUT);      // 入力にして  
duration = pulseIn(PING_PIN, HIGH); // パルス時間計測  
cm = duration / 29 / 2;        // 距離をcmに直す(1cmあたり約29  $\mu$ s)
```


測定結果で分岐

```
if (cm < 20) {           // 障害物まで20cm未満
    RState = RSTATE_ROT;  // なら方向転換
} else {                 // 20cm以上ある
    RState = RSTATE_FORWARD; // なら直進
}
```

改造してみよう

超音波距離センサ



手を近づけると寄ってくる

20cm以上：旋回

20cm以下：直進

1回あたりの旋回時間を変え
るとどうなるでしょう？

来週は自由製作

(複数台使うの也可)

[たとえば]

1) ライントレースタイムアタック

コースをスムーズに曲がれるように調整

2) ロボット双六?

`n = random(1,6);` // 1~6の乱数生成
で、n本目の黒線やn回目の拍手で停止

3) 迷路脱出(フォトセンサ/超音波センサ)

4) 先頭のロボットを見つけて追いかける など

※うまくいかなくても頑張ればOK※

