



実践ロボットプログラミング

LEGO Mindstorms NXT で目指せロボコン!

WEB : <http://robot-programming.jp/>

著者 : 藤吉弘亘, 藤井隆司, 鈴木裕利, 石井成郎

E-mail : support@robot-programming.jp

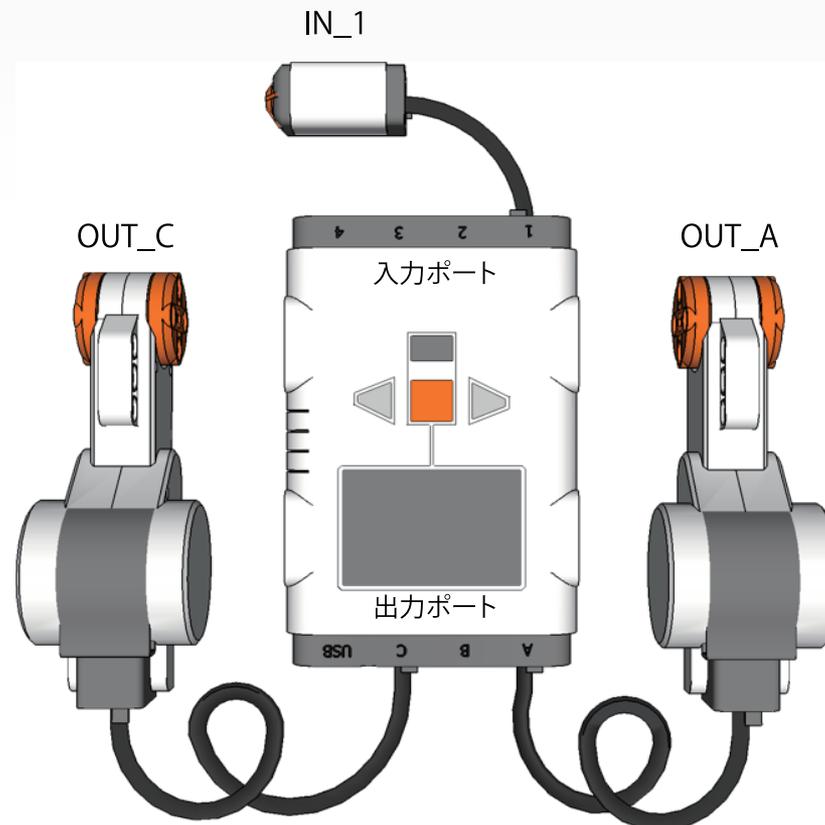


■障害物回避(タッチセンサ)



タッチセンサの接続

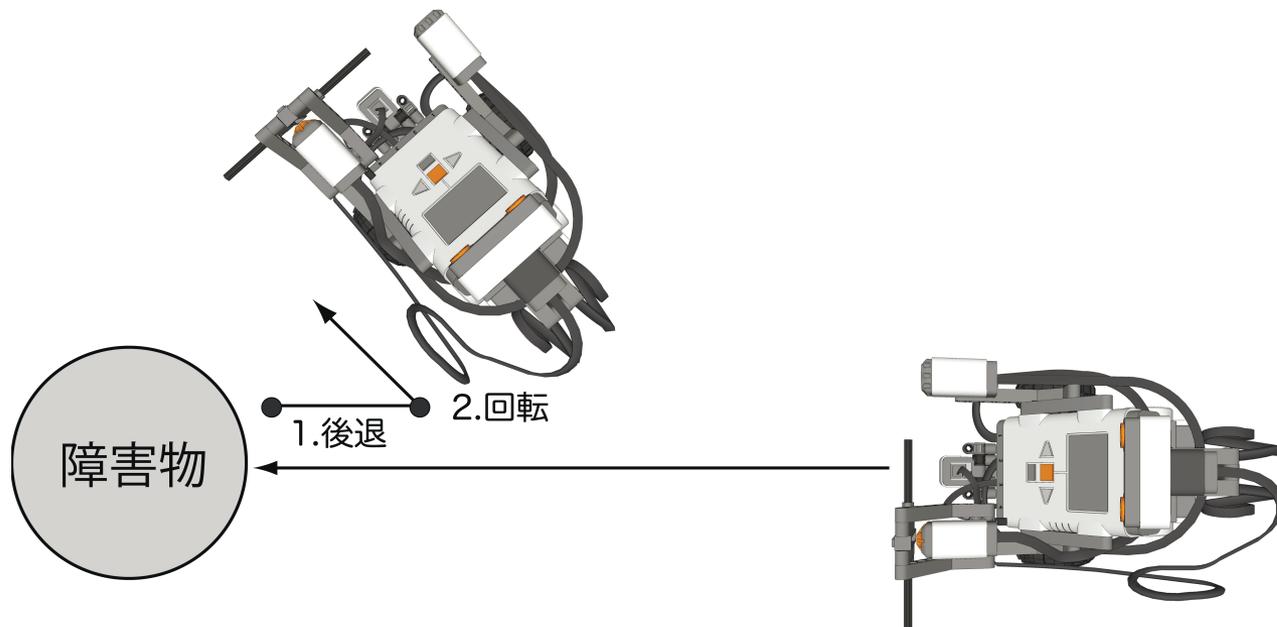
- NXTの入力ポート 1 タッセンサを接続

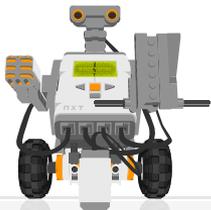




タッチセンサによる障害物回避

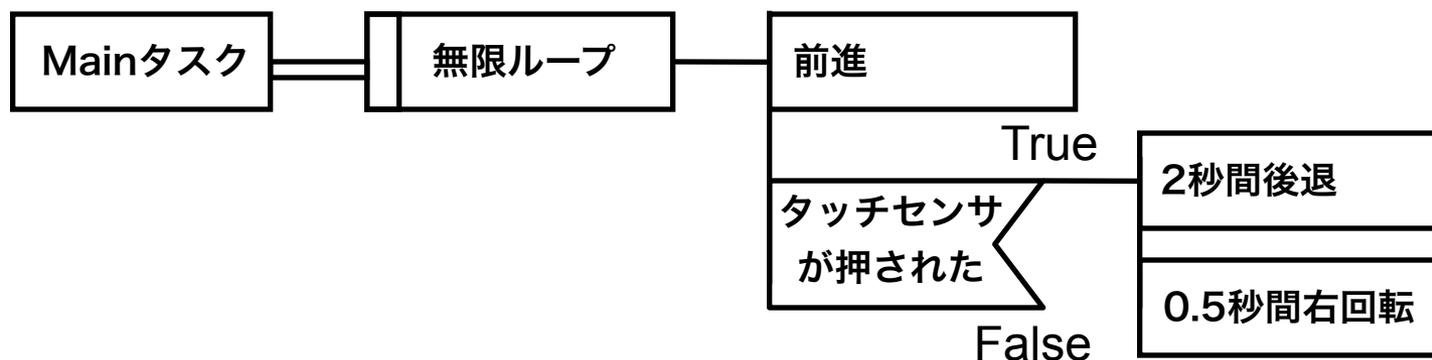
1. 常にロボットを前進 → 無限ループの利用
2. タッチセンサが押されたら，障害物と判定 → 条件分岐
3. 衝突と判定したら，一定時間後退し，右回転.
その後1. に戻る

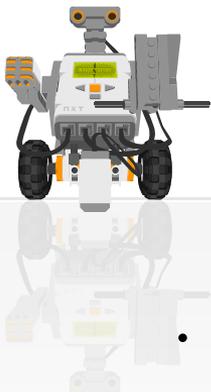




PADによるアルゴリズムの図示 (p.50)

- PAD(Problem Analysis Diagram)
 - プログラムは必ずPADを用いて図示
 - 制御の流れを理解すること





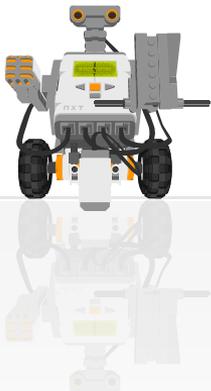
if文による選択構造

- ・ 「もし～ならば・・・しなさい」という場合、if文を用いる

```
if (Sensor(IN_1) == 1 ) {  
    OnFwd(OUT_AC);  
}  
  
else{  
    Off(OUT_AC);  
}
```

センサIN_1が1(押された)のとき
→ロボットを前進

それ以外するとき
→停止



等号と代入

- 等号 (==)

左辺と右辺が等しい

例： $A == 10$ (Aと10が等しい)

- 代入 (=)

左辺に右辺を代入する

例： $A = 10$ (Aに10を代入する)



タッチセンサによる障害物回避 (p.51: touch.nxc)

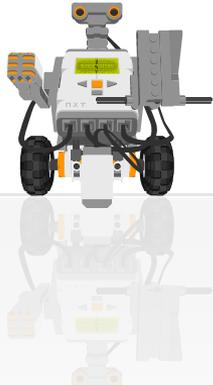
touch.nxc

```
task main()
{
  SetSensorTouch(IN_1);           // ポート 1 をタッチセンサに設定

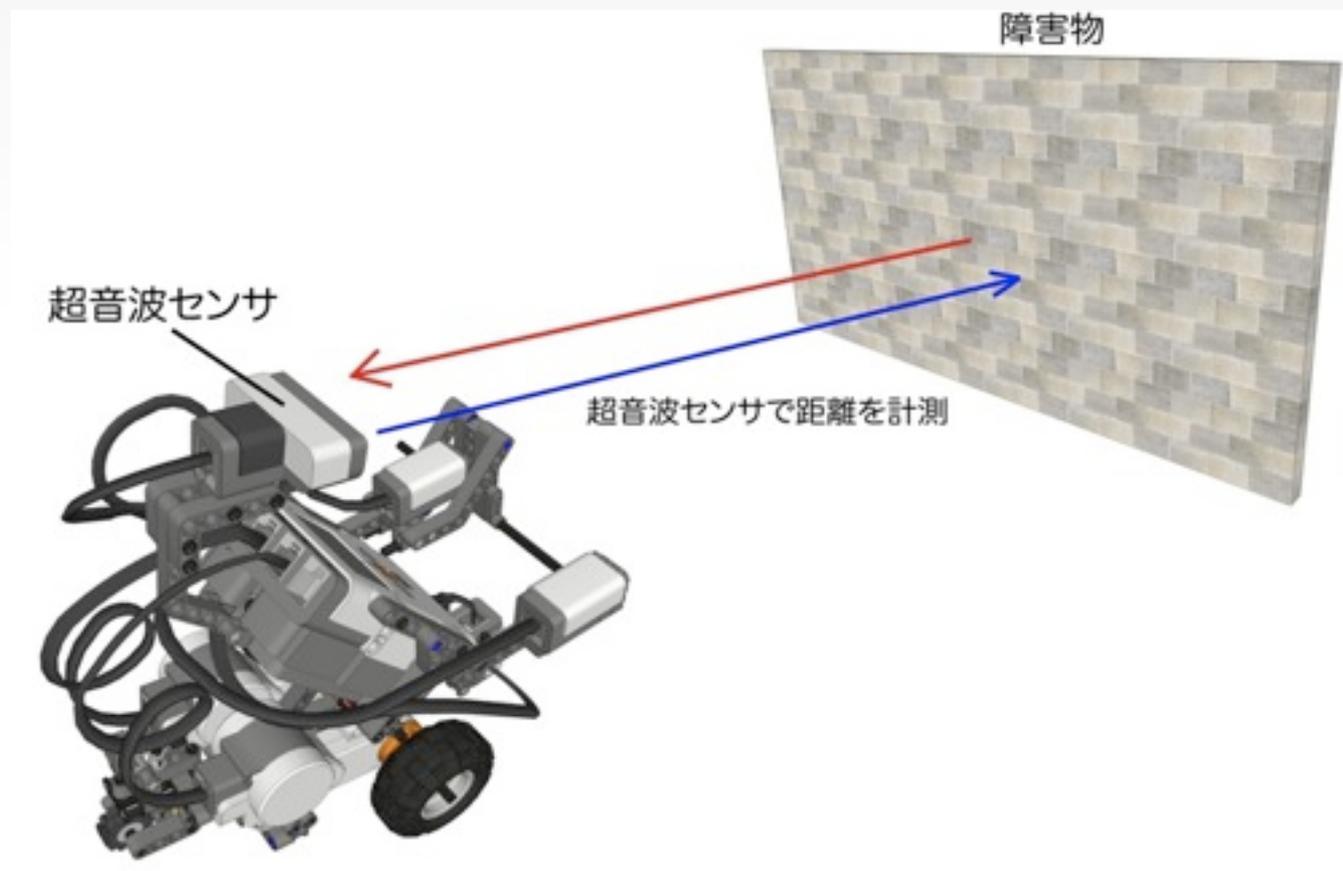
  ① while(true) {
    ② OnFwd(OUT_AC, 75);
    ③ if(Sensor(IN_1) == 1){
      ④ OnRev(OUT_AC, 75);         // 2 秒間後退
        Wait(2000);
      ⑤ OnFwd(OUT_C, 75);         // 1.2 秒間回転
        OnRev(OUT_A, 75);
        Wait(1200);
      }
    }
}
```

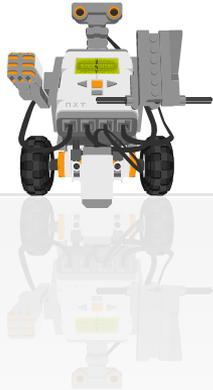


■ 障害物回避(超音波センサ)

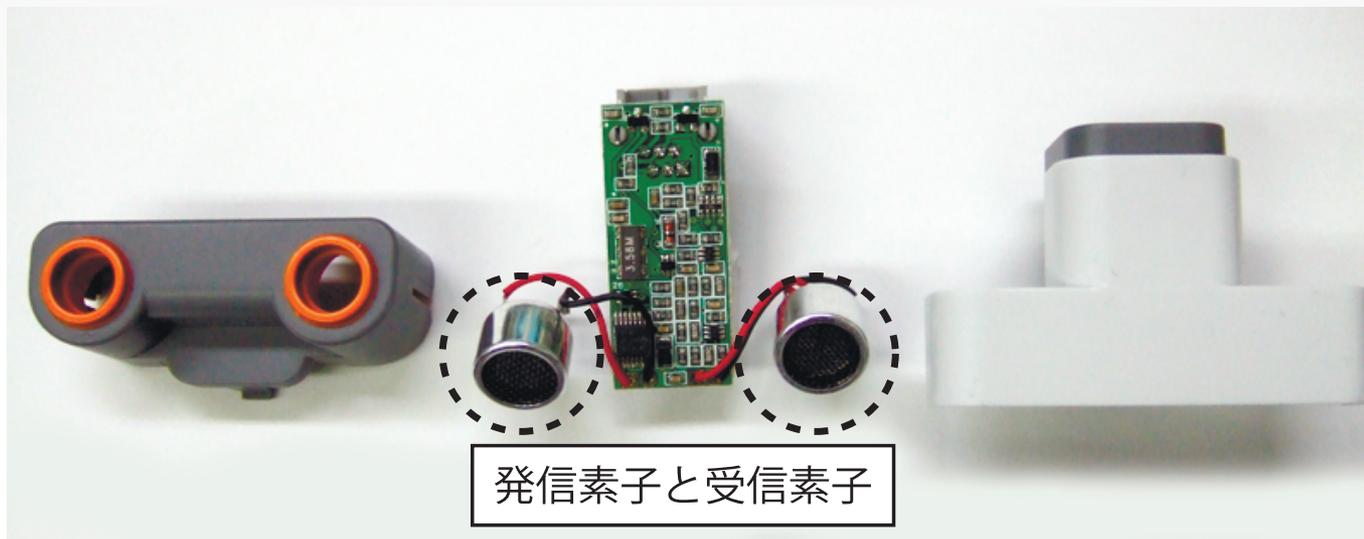


超音波センサによる障害物回避





超音波センサ



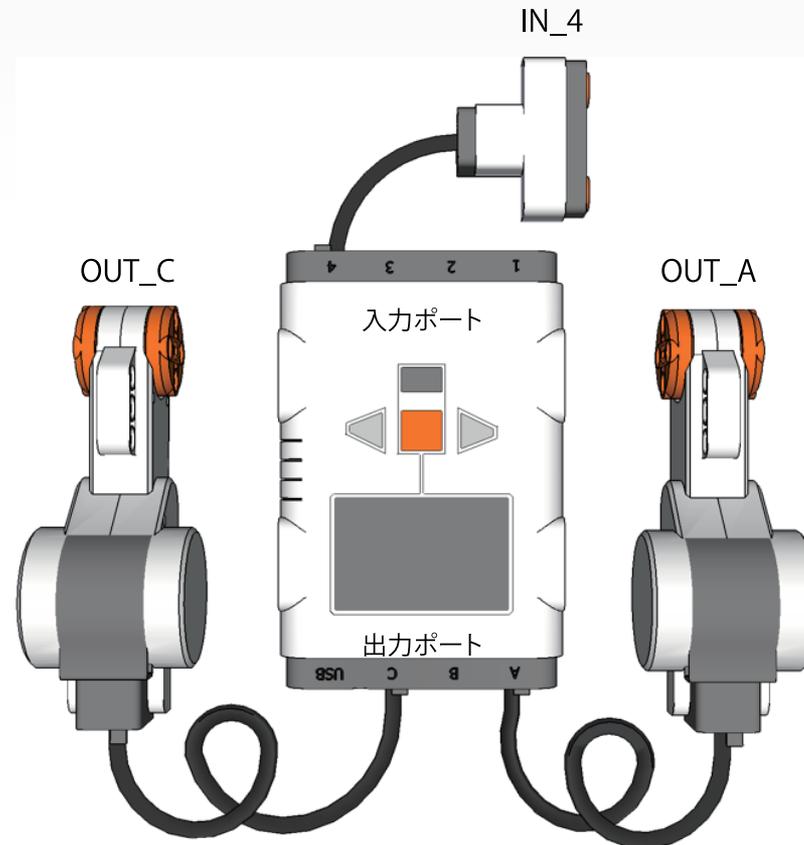
超音波センサの測定原理

超音波を発信し、対象物で反射した超音波を受信し、この音波の発信から受信までの時間を計測することで対象物までの距離を計測



超音波センサの接続

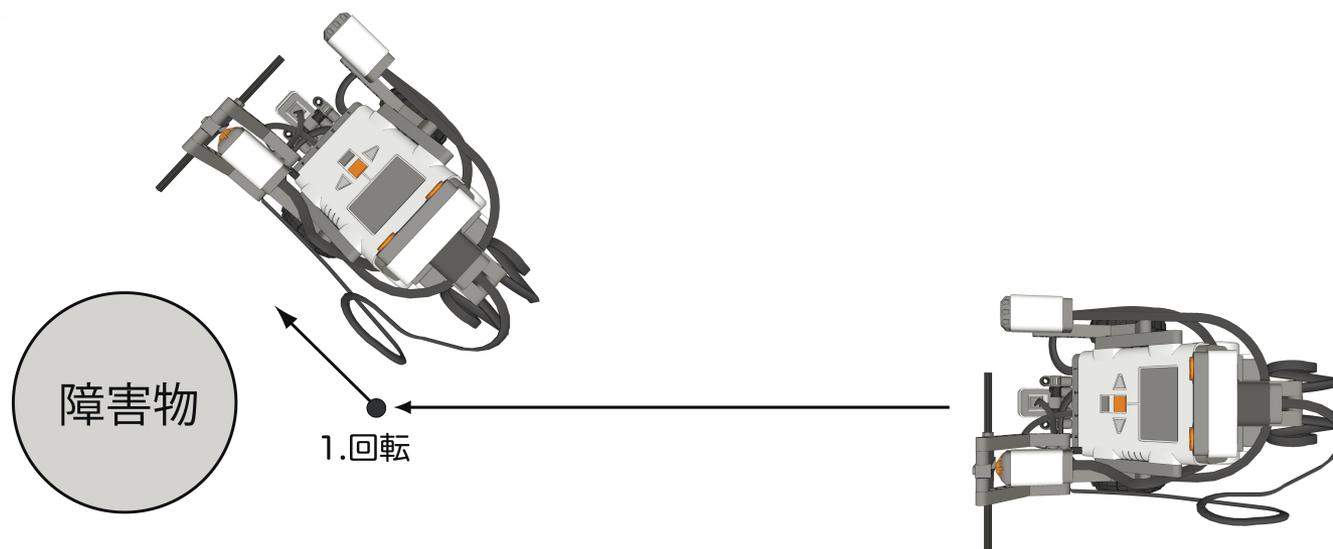
- NXTの入力ポート4に超音波センサを接続



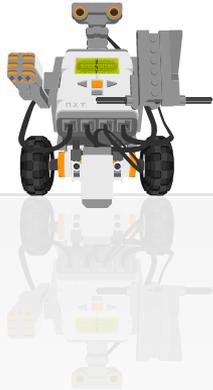


超音波センサによる障害物回避

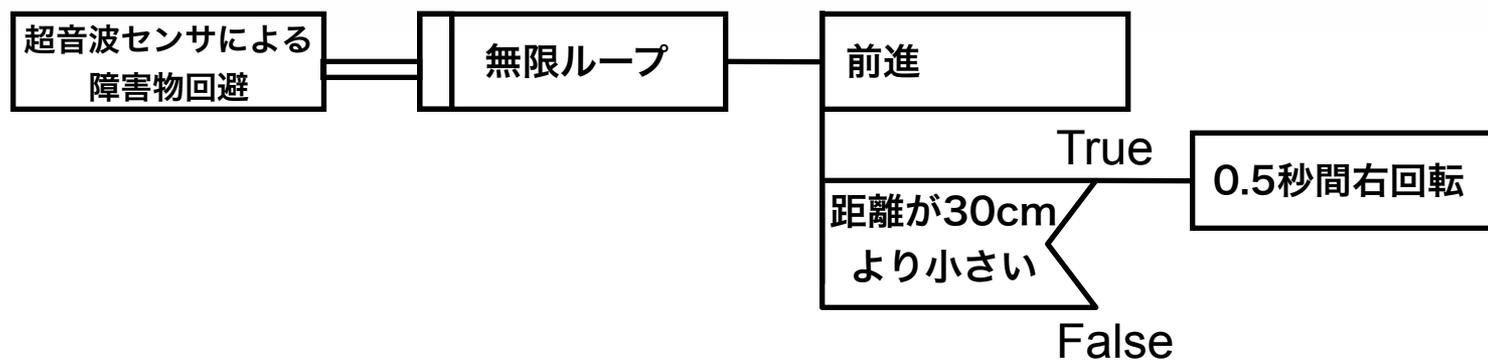
1. 常にロボットを前進 → 無限ループの利用
2. 障害物との距離が30cmより小さいとき, 右回転, その後
1. に戻る

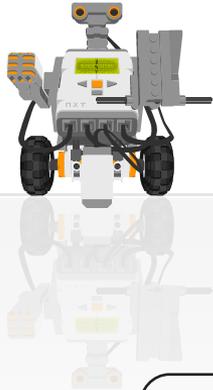


→後退する動作を必要としない



超音波センサによる障害物回避のPAD





超音波センサによる障害物回避 (p.55: usonic.nxc)

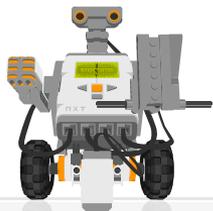
usonic.nxc

```
task main()
{
  SetSensorLowspeed(IN_4); // 入力ポート 4 を超音波センサに設定

  ① while(true){
    ② OnFwd(OUT_AC, 75);
    ③ if(SensorUS(IN_4) < 30){ // 障害物との距離が 30cm より小さいならば実行
      ④ OnRev(OUT_A, 75); // 1.2 秒間右回転
        OnFwd(OUT_C, 75);
        Wait(1200);
      }
    }
  }
```

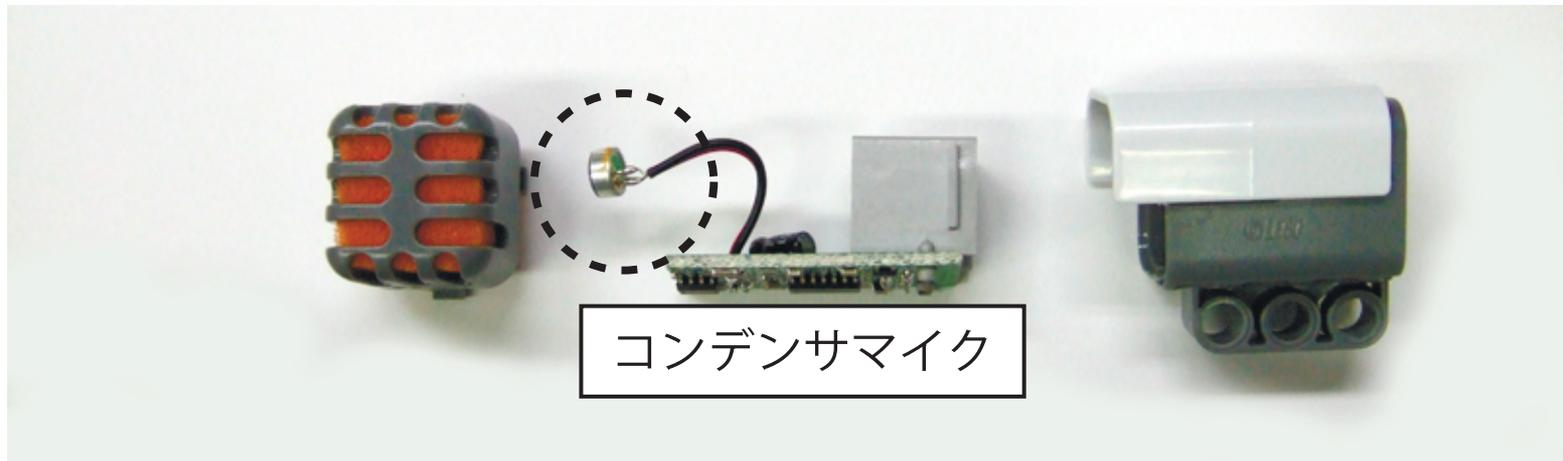


■ サウンドセンサによる制御



サウンドセンサ

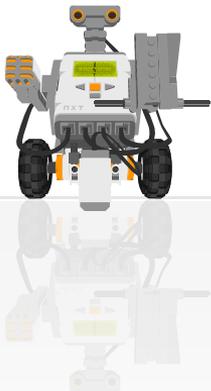
- ・ サウンドセンサの出力
 - 音のない状態を0として0~100までの値





サウンドセンサによる制御のPAD





サウンドセンサによる制御 (p.57: mic.nxc)

mic.nxc

```
task main()
{
    SetSensorSound(IN_2);           // 入力ポート 2 をサウンドセンサに設定

    ① while(true){
        ② until(Sensor(IN_2) > 40); // 音のレベルが 40 より大きくなるまで待機
        ③ OnFwd(OUT_AC, 75);       // 0.5 秒間前進
        Wait(500);
    }
}
```

待機にはuntil文を利用→通常のC言語にはない



until文による待機をwhile文で実現すると

```
task main()
{
  SetSensorSound(IN_2);

  while(true) {
    until(Sensor(IN_2) > 40);
    OnFwd(OUT_AC, 75);
    Wait(500);
  }
}
```

until文の場合

=

```
task main()
{
  SetSensorSound(IN_2);

  while(true) {
    while(Sensor(IN_2) > 40){
      ;
    }
    OnFwd(OUT_AC, 75);
    Wait(500);
  }
}
```

while文の場合