

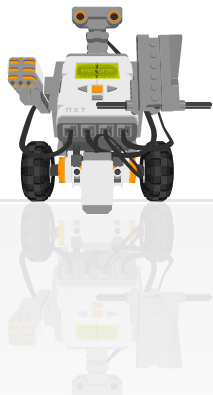
実践ロボットプログラミング

LEGO Mindstorms NXT で目指せロボコン!

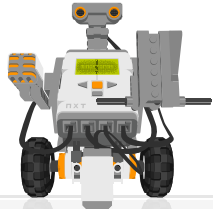
WEB : <http://robot-programming.jp/>

著者 : 藤吉弘亘, 藤井隆司, 鈴木裕利, 石井成郎

E-mail : support@robot-programming.jp

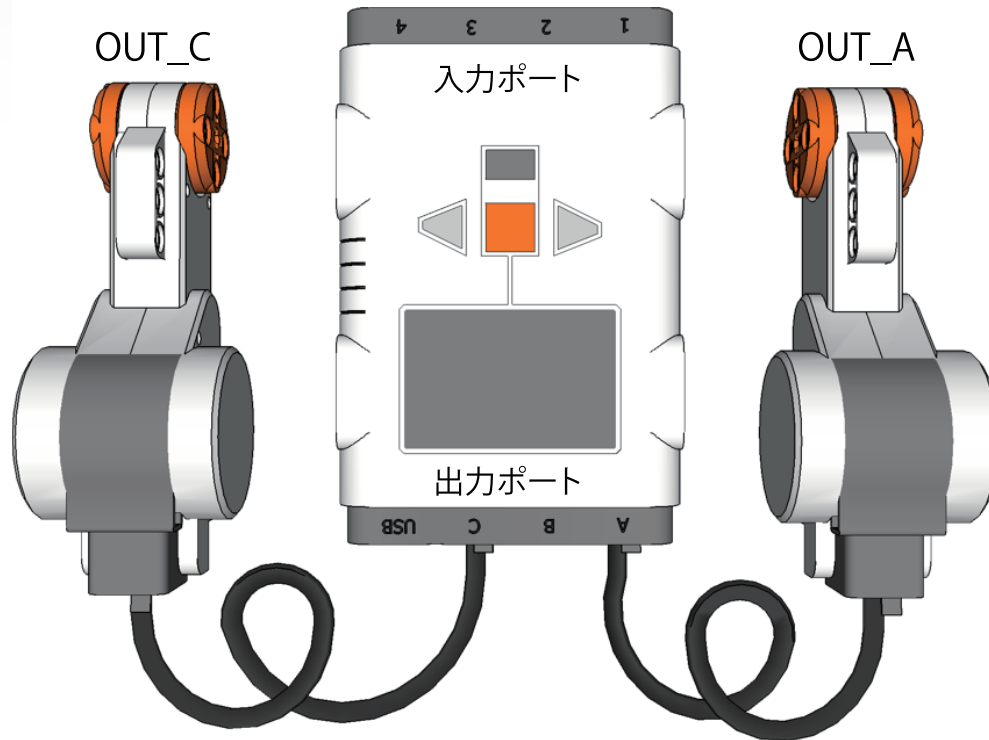


■ ロボットを前進させるには(モータ制御1)

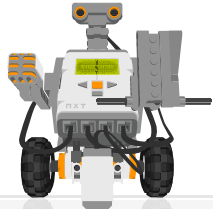


モータの接続

- ・ NXTの出力ポートAとCにモータを接続

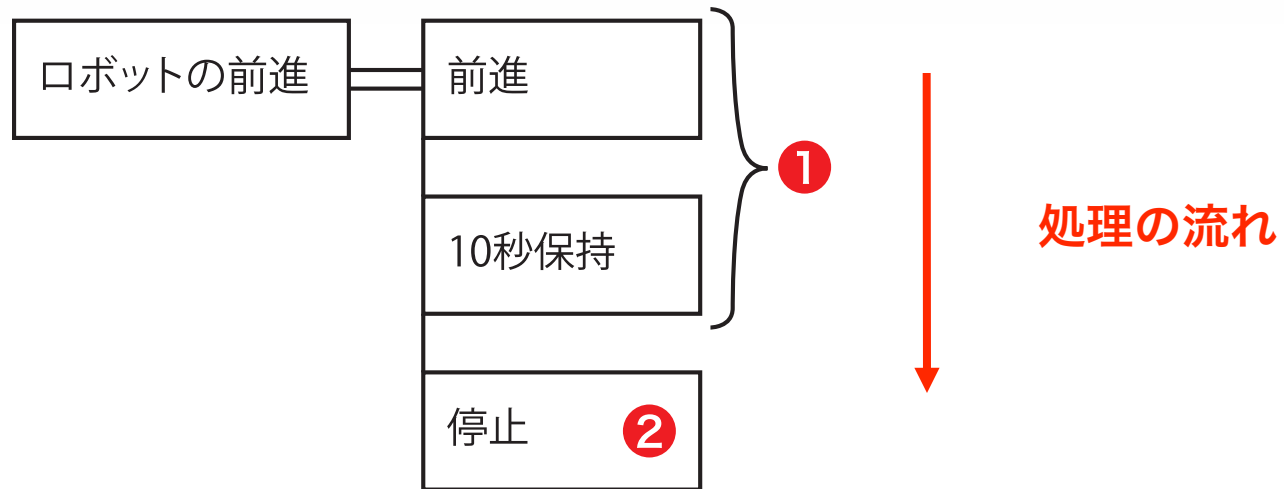


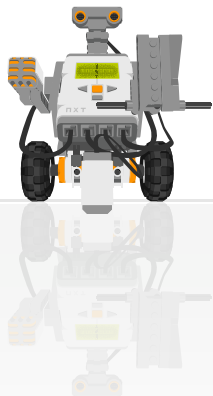
左のモータ : OUT_A
右のモータ : OUT_C



前進プログラムのPAD

- 10秒前進するアルゴリズム

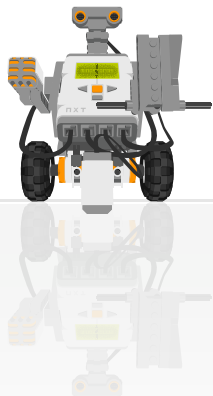




モータ制御によるロボットの前進 (p.42: forward.nxc)

forward.nxc

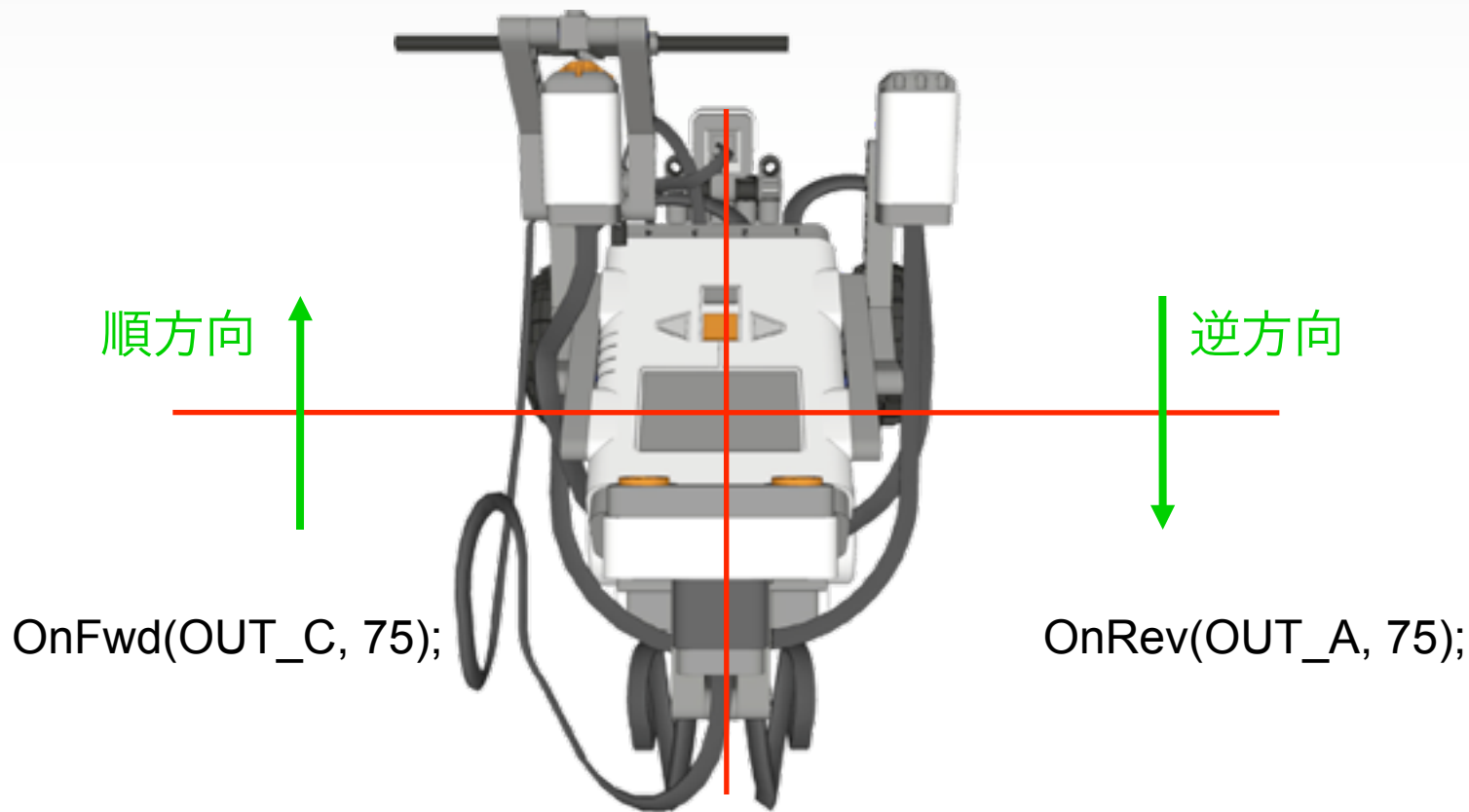
```
task main()
{
  ① OnFwd(OUT_AC, 75);           // 75% のパワーで前進しなさい
  Wait(10000);                  // 10 秒間保持しなさい
  ② Off(OUT_AC);                // 停止しなさい
}
```

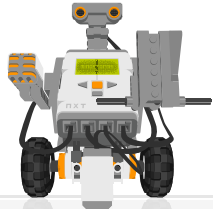


■ ロボットを回転させるには (モータ制御2)

回転

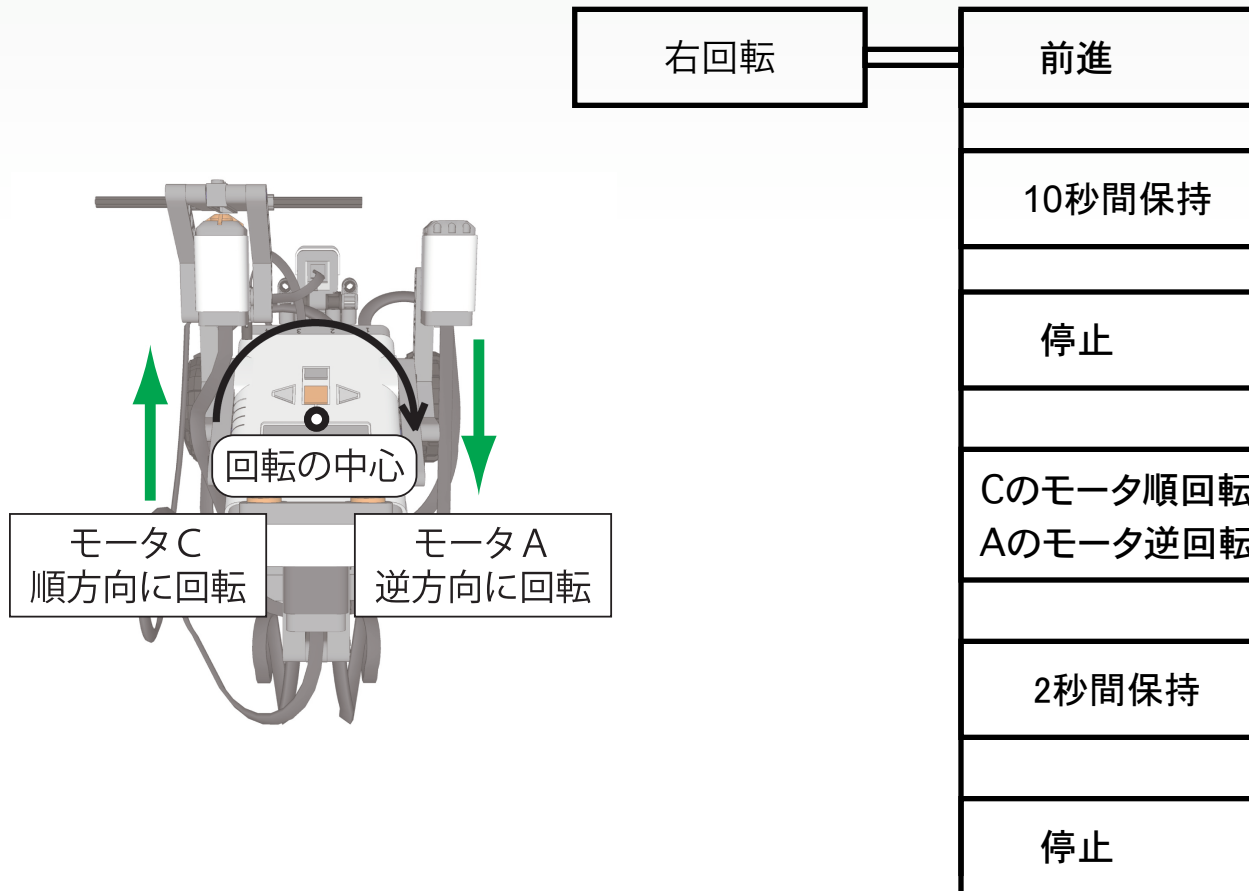
- ・ ロボットを右に回転させるには





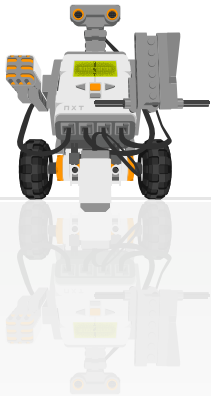
右回転プログラムのPAD

- 右回転するアルゴリズム



処理の流れ



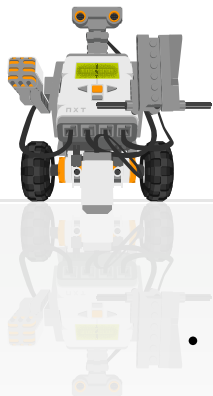


モータの制御 2 (p.45: rotation.nxc)

rotation.nxc

```
task main()
{
  1 OnFwd(OUT_AC, 75);
    Wait(10000);           // 10 秒間前進し停止
    Off(OUT_AC);

  2 OnFwd(OUT_C, 75);     // C のモータを順回転
    OnRev(OUT_A, 75);     // A のモータを逆回転
    Wait(2000);           // 2 秒間保持しなさい
    Off(OUT_AC);         // 停止しなさい
}
```



ロボットを90度回転させるには

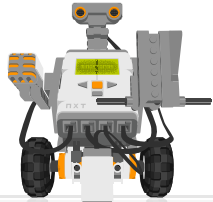
- 90度回転させるには？

→保持させる時間とモーターパワーを調節する

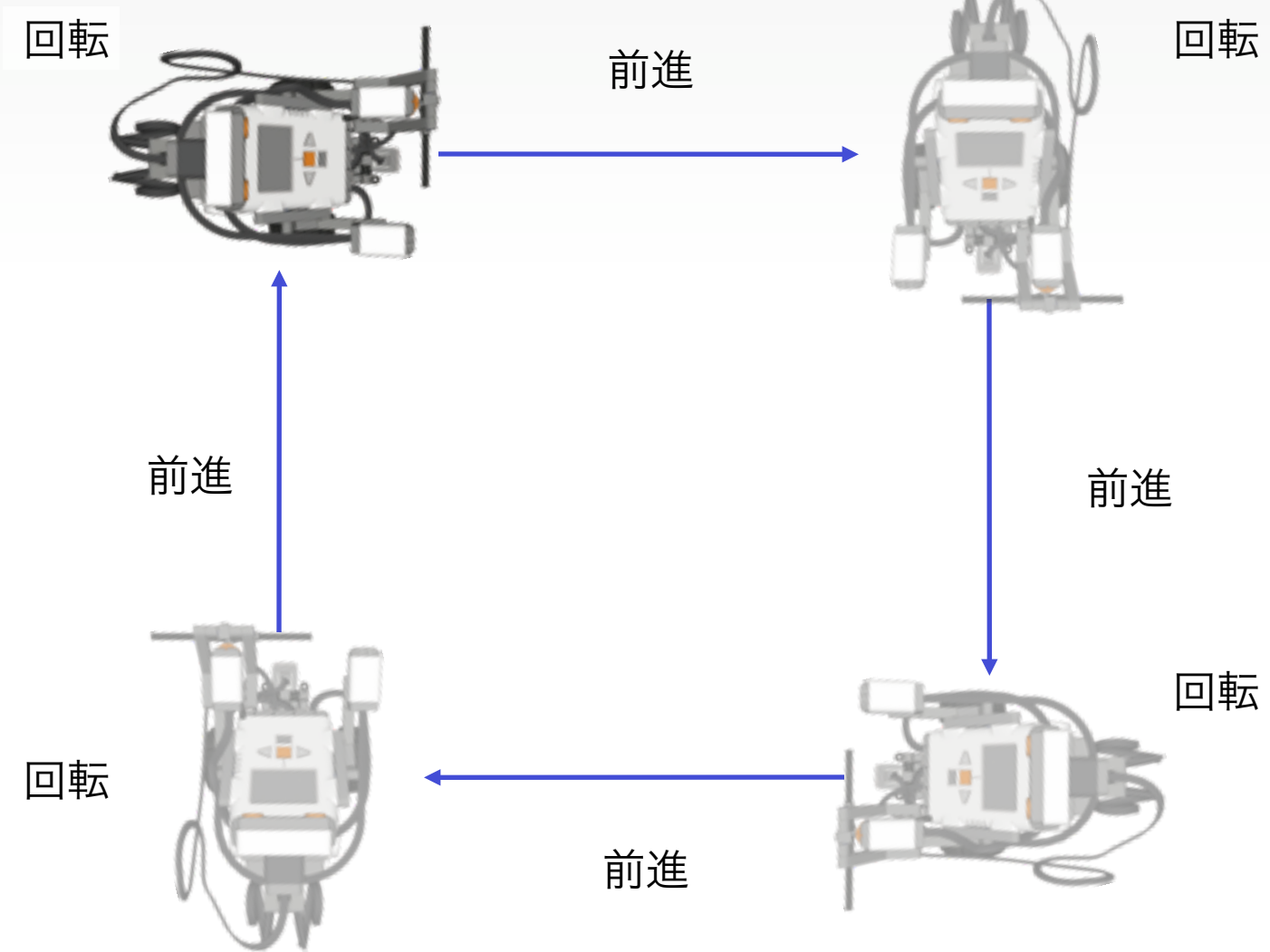
```
task main()
{
    OnFwd(OUT_AC, 75);
    Wait(10000);
    Off(OUT_AC);

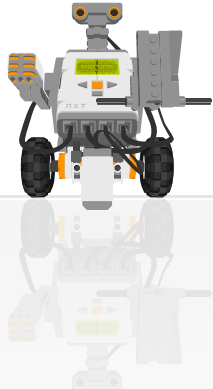
    OnFwd(OUT_C, 75);
    OnRev(OUT_A, 75);
    Wait(500);
    Off(OUT_AC);
}
```

0.5秒保持しなさい

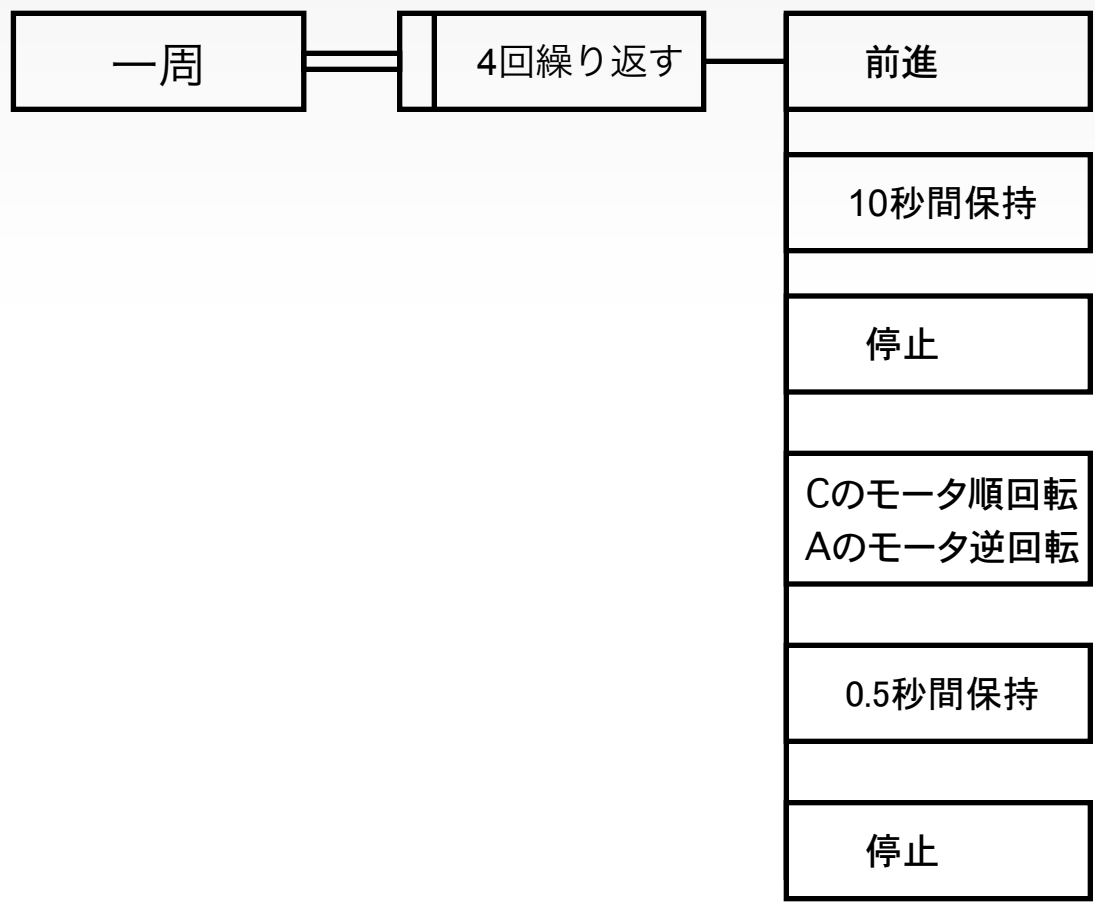


一周するには？

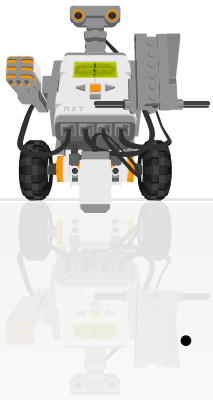




一周するプログラムのPAD (p.47)



処理の流れ



プログラム (一周)

- 一周するには
→前進と回転を4回繰り返せばよい

100周するには

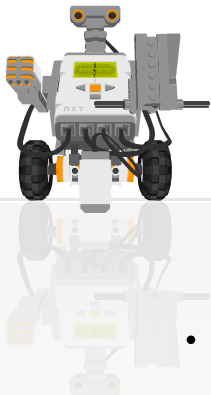
- 前進と回転を400回繰り返えす？
- 2800行のプログラムを書く？
- とても面倒！

```
task main()
{
    OnFwd(OUT_AC, 75);
    Wait(10000);
    Off(OUT_AC);
    OnFwd(OUT_C, 75);
    OnRev(OUT_A, 75);
    Wait(500);
    Off(OUT_AC);

    OnFwd(OUT_AC, 75);
    Wait(10000);
    Off(OUT_AC);
    OnFwd(OUT_C, 75);
    OnRev(OUT_A, 75);
    Wait(500);
    Off(OUT_AC);

    OnFwd(OUT_AC, 75);
    Wait(10000);
    Off(OUT_AC);
    OnFwd(OUT_C, 75);
    OnRev(OUT_A, 75);
    Wait(500);
    Off(OUT_AC);

    OnFwd(OUT_AC, 75);
    Wait(10000);
    Off(OUT_AC);
    OnFwd(OUT_C, 75);
    OnRev(OUT_A, 75);
    Wait(500);
    Off(OUT_AC);
}
```



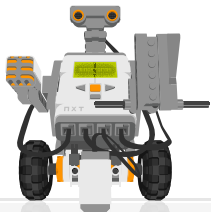
繰り返し

- ・ 同じ処理を繰り返すときはfor文を用いる

初期値 条件 増分処理

```
for ( i=0; i<4; i++ ) {  
    繰り返す内容を記述する  
}
```

変数 i が4より大きいと終了→4回繰り返す ($i=0, 1, 2, 3$)



プログラム (一周 : for文)

```
task main()
{
    OnFwd(OUT_AC, 75);
    Wait(10000);
    Off(OUT_AC);
    OnFwd(OUT_C, 75);
    OnRev(OUT_A, 75);
    Wait(500);
    Off(OUT_AC);

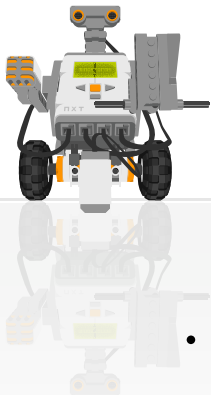
    OnFwd(OUT_AC, 75);
    Wait(10000);
    Off(OUT_AC);
    OnFwd(OUT_C, 75);
    OnRev(OUT_A, 75);
    Wait(500);
    Off(OUT_AC);

    OnFwd(OUT_AC, 75);
    Wait(10000);
    Off(OUT_AC);
    OnFwd(OUT_C, 75);
    OnRev(OUT_A, 75);
    Wait(500);
    Off(OUT_AC);

    OnFwd(OUT_AC, 75);
    Wait(10000);
    Off(OUT_AC);
    OnFwd(OUT_C, 75);
    OnRev(OUT_A, 75);
    Wait(500);
    Off(OUT_AC);
}
```



```
task main()
{
    int i;
    for(i=0; i<4; i++) {
        OnFwd(OUT_AC, 75);
        Wait(10000);
        Off(OUT_AC);
        OnFwd(OUT_C, 75);
        OnRev(OUT_A, 75);
        Wait(500);
        Off(OUT_AC);
    }
}
```

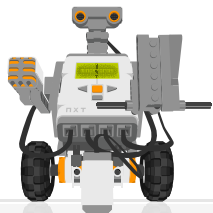


無限ループ (while文)

- ・ while文の条件をtrueにする
 - 無限ループ：終わりのないくり返し

```
task main()
{
    int i;

    while(true) {
        OnFwd(OUT_AC, 75);
        Wait(10000);
        Off(OUT_AC);
        OnFwd(OUT_C, 75);
        OnRev(OUT_A, 75);
        Wait(500);
        Off(OUT_AC);
    }
}
```

■■ 演習問題4-5 (p.49) ■■

- ・ スパイラルや星形の軌跡を描くロボットの動きを実現

