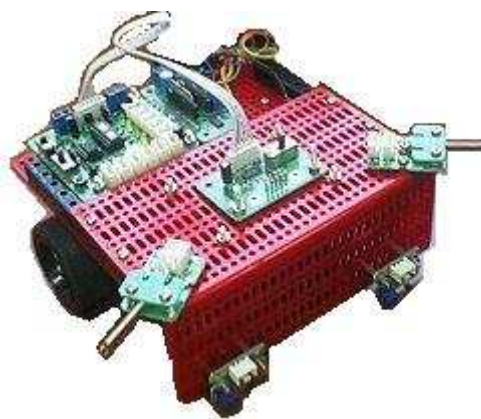


言語を使っ



ロボットを動かそう!

！ 目次 ！

- ・ TiColla_C の起動方法
 - ・ ダウンロードと実行の方法
-
- 1 . ロボットを動かそう
 - 2 . タイヤの回転を制御しよう
 - 3 . ロボットを 5 秒前に進ませよう
 - 4 . ロボットを動かし続ける (繰り返し)
 - 5 . タッチセンサーを使ってみよう
 - 6 . コマンドを自分で作ろう
 - 7 . ライトセンサーを使ってみよう
 - ライントレースをしよう
 - グレースケールシート (切り取り用)
 - テストライン (切り取り用)
 - 8 . 指定回数繰り返しそう

1. ロボットを動かそう

これから書くプログラムはロボットを前進させるためのものです。まず下の例にならってプログラムを書いてみてください。

```
int main() {  
  
    init();  
  
    MOTOR_SPD(1,2);  
    MOTOR_SPD(0,2);  
    M1_IN1 = 0;  
    M1_IN2 = 1;  
    M2_IN1 = 1;  
    M2_IN2 = 0;  
    DELAY_SEC(100);  
}
```

プログラム説明

```
int main()  
{  
  
    init();  
  
}
```

まず、左に挙げた文はプログラムに必要ですので、必ず書くようにしてください。

これらはタイルプログラムの



や



にあたります。

```
MOTOR_SPD(X,Y);
```

このコマンドはモーターのスピードを設定するためのものです。

- ・ X が左右の指定
左側 (M 1) なら 0
右側 (M 2) なら 1 を書く

- ・ Y がパワーの設定
高速なら 0
中速なら 1
低速なら 2
超低速なら 3 を書く

4 . ロボットを動かし続ける (繰り返し)

今までは『DELAY_SEC』コマンドを使って、動作を何秒間継続させるのかを指定しました。
では、今度は動作をずっと続けるようなプログラムを作ってみましょう。

```
int main()
{
    init();

    MOTOR_SPD(1, 2);
    MOTOR_SPD(0, 2);
    M1_IN1 = 1;
    M1_IN2 = 0;
    M2_IN1 = 1;
    M2_IN2 = 0;
    DELAY_SEC(500);

    while(1){

        MOTOR_SPD(1, 2);
        MOTOR_SPD(0, 2);
        M1_IN1 = 1;
        M1_IN2 = 0;
        M2_IN1 = 0;
        M2_IN2 = 1;

        COPCTL = 0;
    }
}
```

実行すると5秒間直進した後に、ずっと右回転を続けるはずですが。

！ ロボットの動作を止めるためにはロボット本体のスイッチを切ること ！

ロボットは何故止まらなくなってしまったのでしょうか。

while コマンド

```
while(1){  
  
    copctl = 0;  
}
```

このコマンドは{ } (中括弧) の中に書かれた命令を繰り返すためのものです。『copctl = 0;』はプログラムを正常に繰り返すために必要なもので必ず書くようにしてください。

このコマンドは、タイルプログラムの



にあたります。

4 . のプログラム例では、最初に直進 5 秒の命令を行った後に『while』のル - プに入ります。今回の『while』は無限ループ (永遠に繰り返しを行う) を行う設定にしてあるので、中括弧内に書かれている回転の命令を繰り返し続けます。

！ ポイント ！

『while』ループの中に書いてある回転の命令に注目してください。以前まで入っていた『DELAY_SEC』という命令がありません。回転命令は『while』によって無限に繰り返されるので、秒数を指定することに意味が無いのです。

今は回転という単一の動作なので『DELAY_SEC』が必要ありませんが“ 回転 直進 ” など複数の動作を組み合わせる場合はそれぞれの動作に『DELAY_SEC』が必要になります。

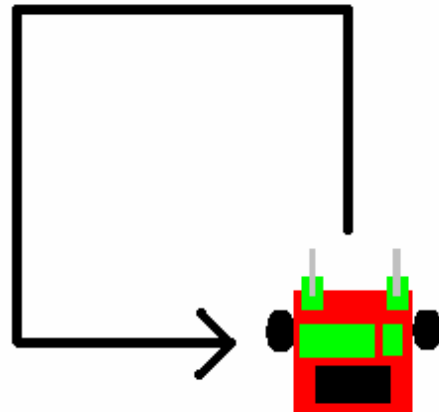
< 問 2 >

右の図のような動作をずっと繰り返すようなプログラムを書きなさい (ターンはできるだけ 90 ° になるように調整すること) 。

！ ヒ ン ト ！

- ・ 繰り返しを有効に使って、命令をなるべく少なくする

うまくいかない時は説明やポイントなどを見直しましょう。



5 . タッチセンサーを使ってみよう

皆さんの作ったロボットにはタッチセンサーがついています。このセンサーは、主にロボットが壁にぶつかったことを検知する為に使用します。

このセンサーを使ってステップアップしたプログラミングに挑戦してみましょう。下に示すプログラムは、『直進中、もしロボットが壁にぶつかったら後退した後に回転して壁から離れる』というものです。実際に打ってみましょう。

```
int main() {
    init();
    while(1){
        if(CN1 == 1){
            MOTOR_SPD(1,0);
            MOTOR_SPD(0,0);
            M1_IN1 = 0;
            M1_IN2 = 1;
            M2_IN1 = 0;
            M2_IN2 = 1;
            DELAY_SEC(250);

            MOTOR_SPD(1,0);
            M1_IN1 = 0;
            M1_IN2 = 1;
            M2_IN1 = 1;
            M2_IN2 = 0;
            DELAY_SEC(250);
        }
        else{
            MOTOR_SPD(1,0);
            MOTOR_SPD(0,0);
            M1_IN1 = 1;
            M1_IN2 = 0;
            M2_IN1 = 1;
            M2_IN2 = 0;
        }
        copctl = 0;
    }
}
```

5のプログラムでは『if』と『else』が出てきます。これらをセットでif文と呼び、『TiColla C』に限らずC言語では極めて重要な命令文となっています。

if コマンド

```
if(条件)
{
    . . .
    命令文
    . . .
}
else
{
    命令文
}
```

ifとは『もし指定された条件を満たしたら、中括弧の中に書かれた命令を実行する』と言うものです。まずifの横にある括弧内に条件を記入します。その後、中括弧内に条件を満たしたときに行う命令を書き込みます。

elseとは『ifに書かれた条件を満たさなかった場合、中括弧の中に書かれた命令を実行する』というものです。ifとelseはセットですから、書き方は左側にある例のように、まずifを書いてから、下にelseを書きます。(ifとelseはセットですが、場合によってはelseを省略することも可能です)

プログラム例では、ifの条件文に『CN1が“1”になった時(タッチセンサーに反応があったら)』という指定をし、命令文には『後退した後、その場で回転』という書き込みをしました。そしてelseの命令文(タッチセンサーに反応が無いとき)には『直進』と書きました。実際にそうなっているか、前のページに戻って確かめてください。

このコマンドはタイルプログラムの



と同じ働きをします。

< 問 3 >

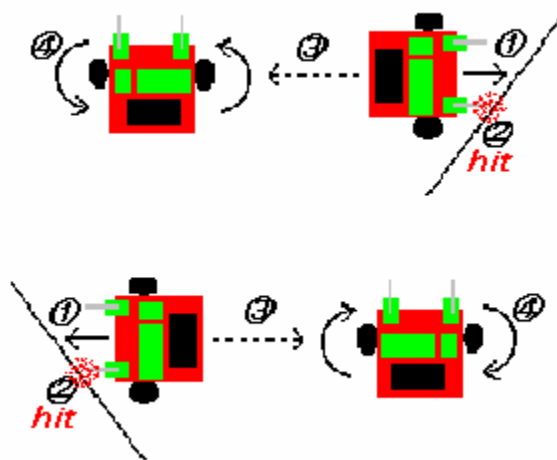
右の図のように

- ・『右側のタッチセンサーが壁に当たった場合はバックして左回転』
- ・『左側のタッチセンサーが壁に当たった場合はバックして右回転』

するようなプログラムを書きなさい。タッチセンサーに反応がないときは常に直進の動作をさせなさい。

！ ヒ ント ！

- ・CN1だけでなくCN2も使うこと
- ・もし…だったら をする という考え方が重要



6 . コマンドを自分で作ろう

プログラムが複雑になってくると自然に行数も多くなっていきます。複雑な動作をしようとするれば命令数が増えるため仕方無い事なのですが、行数の多さの原因として一つの命令にかかる行数の多さがあげられます。

特に1の項目の『MDRIV_X_Y = Z;』コマンドは前進・後退などの命令に必要なので、高い頻度で使用するわりに1つの命令(前進など)を行うのに4行も書かなくてははいけません。プログラムは誰が見ても理解しやすいように『短く・簡潔』に書くことが重要です。

そこで、ここでは行数を省略して命令する方法があります。下のプログラムを必ず『int main()』よりも上に書いてください。

！ 但し、/* main */よりも上に命令を書いてはいけない ！
まずは実際に記述してください。

```
int stop() {
    init();
    M1_IN1 = 0;
    M1_IN2 = 0;
    M2_IN1 = 0;
    M2_IN2 = 0;
    return;
}

int fw() {
    init();
    M1_IN1 = 1;
    M1_IN2 = 0;
    M2_IN1 = 1;
    M2_IN2 = 0;
    return;
}

int bk() {
    init();
    M1_IN1 = 0;
    M1_IN2 = 1;
    M2_IN1 = 0;
    M2_IN2 = 1;
    return;
}
```

7. ライトセンサーを使ってみよう

このロボットのライトセンサーは赤外線を検知するためについています。
使用する場面は色々考えられますが、このロボットを使ってできることは主に『赤外線を放つボールを追いかけること』と『ライントレース』の2つです。

！ ライントレース ！

床に引いてある線を検知して、その線に沿ってロボットを動かすこと

以下のプログラム例はボールを見つけた時にそれを追いかける動作をするものです。
(このプログラムを書く前提として6.の例にあるコマンドの準備が必要です。Fw();は前進、le();は左回転を示しています)

```
int main() {
    init();

    while(1){
        AD_GETDATA(AD_CN4, &adValue);
        if(adValue >= 190){
            fw();
            MOTOR_SPD(1,2);
            MOTOR_SPD(0,2);
        }
        else{
            le();
            MOTOR_SPD(1,2);
            MOTOR_SPD(0,2);
        }
        COPCTL = 0;
    }
}
```

今回のプログラムはこれだけでは動きません。

プログラム中に出てくる『adValue』という変数を定義しなければ正常な動作ができません。

次に変数を定義する方法を示します。

GETDATA (代入) 命令

```
AD_GETDATA(AD_CN04, &adValue);
```

これはライトセンサーから得られた値を変数『adValue』に代入するという命令です。括弧中にある『AD_CN04』は CN (チャンネル) 4 につながっているライトセンサーから値をとるということを表しています。

今回のプログラム

```
AD_GETDATA(AD_CN04, &adValue);
```

```
if (adValue>=X)
```

```
{  
}
```

```
else
```

```
{  
}
```

という一連の命令群はタイルプログラムの



と同じです。

動作とプログラムの説明

まずライトセンサーは赤外線強度を検知します。検知された赤外線は強度に応じた値に変換 (『全く検知しない』 '0' ~ '255' 『強い赤外線が検知』) され、その値は先ほど説明した『AD_GETDATA(AD_CN04, &adValue);』という命令で『adValue』に代入されます。代入された値は if 文の (条件) の中で比較されます。つまり、

```
if (adValue>=190)
```

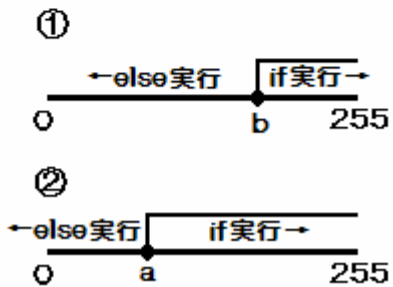
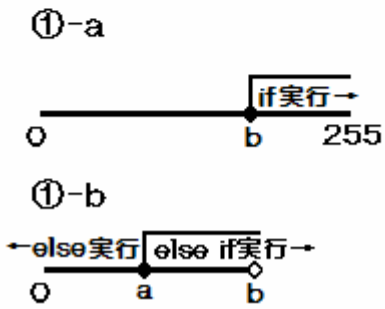
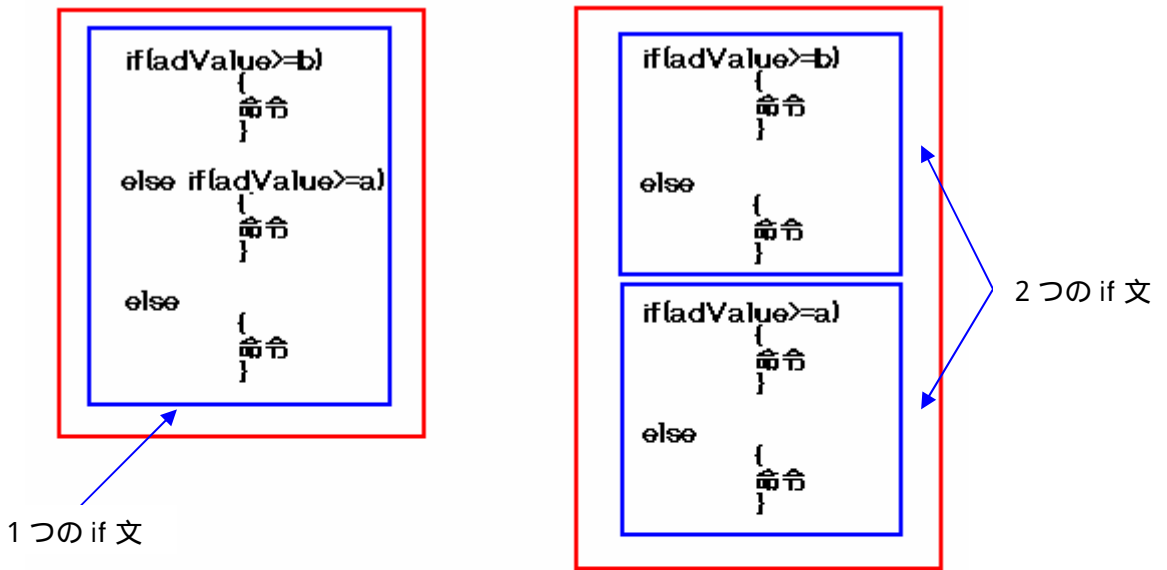
という条件文は『もし adValue に代入された値が 190 より大きいなら (簡単に言えば、『強い赤外線を検知したとき』) if 文 { } 内の命令を実行する』ということを示しています。

・ if 文の説明（応用）

タッチセンサーで if 文は 1 であるかどうか（入力があったかどうか）を比較するために使用しました。そのため if と else 意外は必要になりませんでした。しかし、if 文には if と else の他に『else if』という要素があります。

else if というのは、最初の if で除かれなかった条件の中でさらに複数の条件を設定するために用いられます。『条件を複数設定するなら今までの if 文を何個も書けば良い』と思った人もいると思います。しかし if 文を複数書くのと、else if を使用した if 文を書くのでは意味が違います。

ライトセンサーを例に else if を使用した if 文について検証してみましょう。まずライトセンサーではタッチセンサーのときとは違い、条件として得られる数値は 0 ~ 255 の範囲で変わります。そして得られた数値は if 文の条件と比較されます。先ほども書きましたが、else if を使用すると比較の方法が変わります。これから示す例では『センサーの値が a ~ b の範囲にある時、指定の動作をさせる』という想定をしています。下の図を見てください。



左右のプログラムには大きな違いがあります。左側が『else if を使用した if 文で書い