

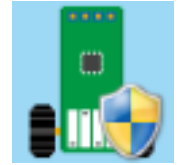
## マイコンレーサーを動かす (1)

「ブロックコマンダー」(Block Commander)でマイコンレーサーを動かすプログラムを作ります。

### ●ブロックコマンダー

マイコンレーサーを動かすプログラムを作ります。

ブロックコマンダー(Block Commander)のアイコンをダブルクリックします。



ブロックコマンダーの画面が表示されます。



上の配置エリアに、下にあるブロックを並べて、プログラムを作ります。

## ●車輪を回す

車輪を回して、マイコンレーサーを走らせてみましょう。

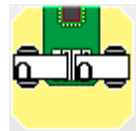
「▶」マークの「スタートブロック」を、配置エリアの左上へドラッグして配置します。



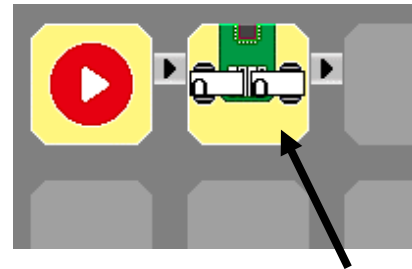
ここからプログラムがスタートします。



次に、車輪を回す「モーターブロック」を、スタートの右側に配置します。



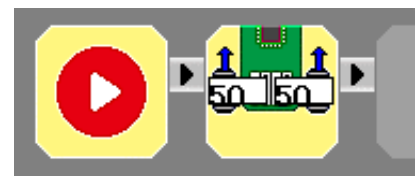
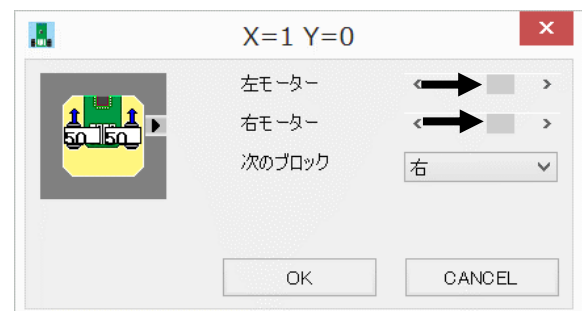
2つのブロックが矢印「▶」でつながった状態になります。



このままだと左右のモーターのパワーが「0」になっていて回りません。  
モーターブロックをダブルクリックして、設定画面を出します。

「左モーター」「右モーター」のスライダーをドラッグして、それぞれの値を「50」にします。

できたら、下の「OK」ボタンをクリックします。



この後はプログラムを終了したいのですが、ブロックコマンダーには「終了」のブロックがありません。

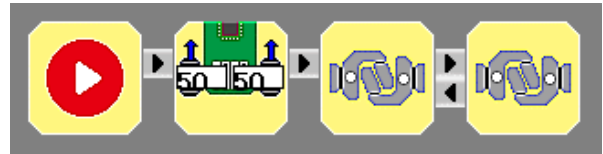
接続ブロックを使って、ぐるぐるのループするようにしましょう。

接続ブロックは、何もせずにとりなるブロックへつなげるブロックです。



モーターブロックの右側に、接続ブロックを2個置きます。

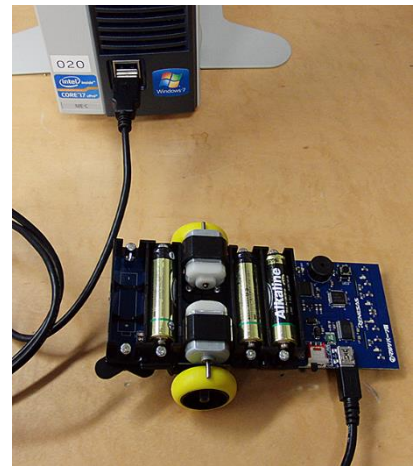
接続ブロックをダブルクリックすると、次に行く方向を変えられます。左側の接続ブロックは右、右側の接続ブロックは左へ行くようにして、2個のブロックでぐるぐる回るようにします。



## ●プログラムを転送

作ったプログラムを、マイコンレーサーへ転送します。

まず、マイコンレーサーとパソコンを USB ケーブルでつなぎます。



ブロックコマンダーで、「転送」ボタンをクリックします。



マイコンレーサーへプログラムが転送されます。しばらく時間がかかるので、待っててください。

```

C:\Users\shiro\Desktop\block_commander.lnk
C:\Users\shiro\Desktop\block_commander.lnk
test.c
MIC/80/30/20/10/Tiny_R8C/Tiny Series Compiler V.5.44 Release 00
Copyright (C) 2011,2008 Renesas Technology Corp.
and Renesas Solutions Corp. All rights reserved.
test.c
C:\PROGRA~2\BLOCK_1\SOURCE\1>IF ERRORLEVEL 1 GOTO C_ERROR
C:\PROGRA~2\BLOCK_1\SOURCE\1>ECHO スタートアッププログラムのアセンブル
スタートアッププログラムのアセンブル
C:\PROGRA~2\BLOCK_1\SOURCE\1>aa30 ncr10.a30 -finfo -0 *.Release -LM -D_R8C_
1 -R8C -l>err.txt
C:\PROGRA~2\BLOCK_1\SOURCE\1>IF ERRORLEVEL 1 GOTO A_ERROR
C:\PROGRA~2\BLOCK_1\SOURCE\1>ECHO リンク
リンク
C:\PROGRA~2\BLOCK_1\SOURCE\1>In30 -L "r8lib" -0 -k6 -0 *.Release\test_x30 -f
80 *.Release\ncr10_r30 *.Release\test_r30 -l>err.txt

```



最初に黒い画面が表示された後、転送画面が出ます。

転送できたら、マイコンレーサーからケーブルを外して、電源スイッチを ON にします。

スタートボタンを押すと、スタート音がして、車輪が回って走り出します。

今はモーターを回転させるだけのプログラムなので、走り出すといつまでも止まりません。手で止めて、電源スイッチを OFF にしましょう。

**●モーターの調整**

今は左右のモーターのパワーを「50」にしていますが、真っ直ぐ進まない場合があります。これはそれぞれのモーターで「くせ」(個性差)があって、同じ「50」のパワーを設定しても同じ回転数にならないためです。

そんな時は、モーターの回転数を調整します。

ブロックコマンドのメニューの「ツール」→「オプション」をクリックします。

オプション設定画面が開きます。



「最大回転数設定」の項目で、「左」「右」のモーターのパワーを設定できます。弱い側のモーターの回転数を上げてみてください。(1~5%くらい変えれば十分でしょう)

変更したら、右下の「OK」ボタンをクリックします。

もう一度プログラムをマイコンレーサーへ転送して、走らせてみましょう。

ちゃんと真っ直ぐ走るように、回転数を調整してください。

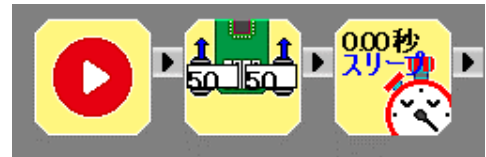
## ●タイマーを使う

今のプログラムは、マイコンレーサーが走り出すとずっと止まりません。  
一定時間走ったら止まるプログラムにしてみましょう。

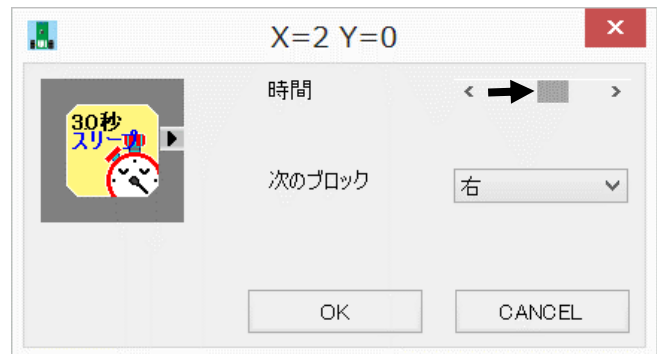
2つの接続ブロックをドラッグして、右へ2つずらして、ブロック2個分のスペースを空けます。



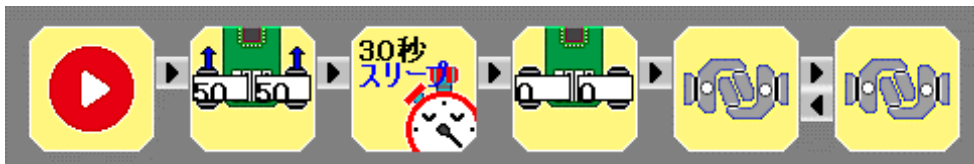
モーターブロックの右どなりに、タイマースリープブロックを配置します。



タイマースリープブロックをダブルクリックして、時間を設定します。  
今回は3秒にしましょう。



さらに右側にモーターブロックを置いて、パワーを「0」にします。



これで、3秒間前進するプログラムができました。  
プログラムを転送して、走らせてみましょう。

スリープの時間(秒数)をいろいろ変えて、その秒数だけ走るかどうか試してみましょう。

## ●いろいろな走らせ方

モーターブロックとタイマースリープブロックの組み合わせで、いろいろな走らせ方をすることができます。

まずは、途中で速度を変えてみましょう。



このプログラムは、「パワー50 で1秒前進」→「パワー100 で1秒前進」→「パワー50 で1秒前進」→「停止」という動きをします。

モーターのパワーやスリープの時間をいろいろ変えて、試してみましょう。

また、左右のモーターのパワーを違う値にすると、マイコンレーサーが左右へカーブします。



このプログラムは、「右へカーブして1秒前進」→「左へカーブして1秒前進」→「停止」という動きをします。

モーターのパワーやスリープの時間をいろいろ変えて、試してみましょう。

※いらないブロックを消す時は、消したいブロックをクリックして、「削除」ボタンをクリックしてください。

削除  
Del

## ●くり返し

これまではプログラムを最初から最後まで 1 回だけ実行していましたが、同じ動作をくり返し実行することができます。

左右へカーブするプログラムを改造して、ずっとくり返すようにしてみましょう。

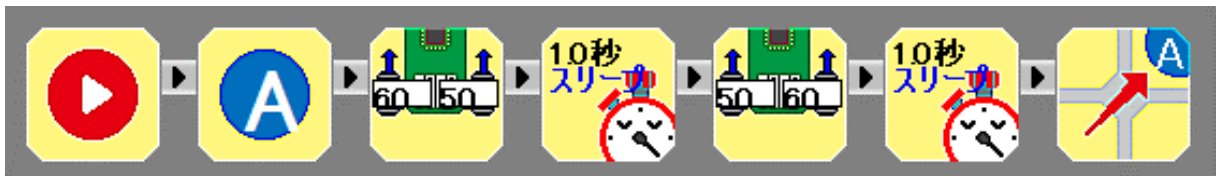
まず、スタートブロックの右側にラベルブロック(Aのマーク)を入れます。

(2 個目のモーターブロック以後のブロックを、全て 1 個右にずらして、スペースを空けてください)



ラベルブロックは、ジャンプする時の飛び先になるブロックです。

そして、最後の停止する部分のブロックを消して、代わりにジャンプブロックを入れます。



ジャンプブロックは、指定したラベルへジャンプするブロックです。

これで、「右へ1秒カーブ」→「左へ1秒カーブ」→「最初に戻ってくり返し」というプログラムになります。

プログラムを転送して走らせてみましょう。左右へカーブしながら前進していきます。

モーターのパワーやスリープの時間をいろいろ変えて、くり返しの動作を試してみましょう。

※ラベルは「A」～「P」まで 16 個使えます。ラベルブロックやジャンプブロックをダブルクリックすると、ラベルのアルファベットを変えられます。

作ったプログラムは、メニューの「ファイル」→「名前を付けて保存」で、デスクトップに保存しておきましょう。

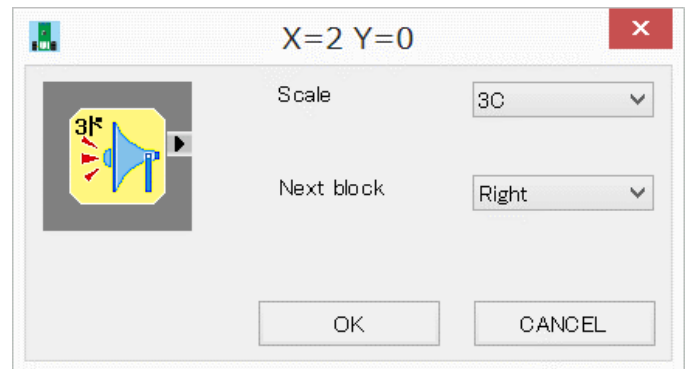
## ●音を出す

モーターを回して走らせる他に、ブザーで音を鳴らすことができます。  
左右カーブのプログラムを改造して、音を鳴らしてみましょう。



モーターブロックの代わりに、ブープブロック(スピーカーのマーク)を配置します。  
ブープブロックをダブルクリックすると、音階が設定できます。

英語表記になっていますが、「Scale」を変更すると、音階が変更されます。  
「3C」が普通の「ド」です。  
普通の「ド」～2 オクターブ上の「ド」まで設定できます。  
「OFF」は音が停止します。



## ◇音階

Scale	3C	3D	3E	3F	3G	3A	3B	4C	...
音階	ド	レ	ミ	ファ	ソ	ラ	シ	ド(1オクターブ上)	...

1 個目のブープブロックの音階を「ド」(3C)にします。



プログラムを転送して実行してみましょう。1 秒ごとに「ド」の音を鳴らします。

音階やスリープ(音の長さ)をいろいろ変えて、試してみましょう。  
うまくやると、曲が奏でられます。



曲のプログラムの例です。

ラベルを使って小節ごとに分けると、わかりやすいと思います。



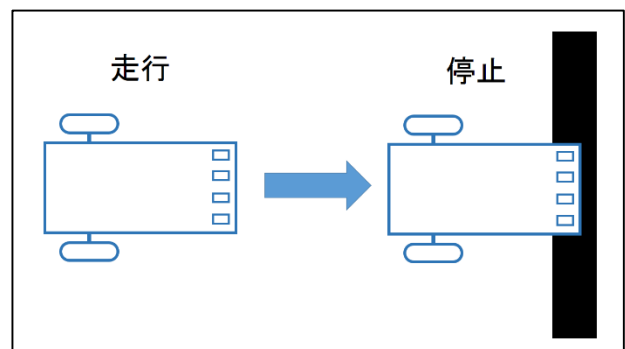
## ●センサーを使う

マイコンレーサーの先端の下側には、センサーが 4 つ付いています。

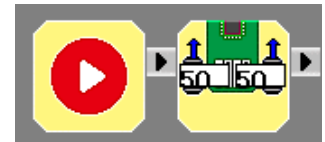
地面の明るさ(白い／黒い)を検知して、走り方などを変えることができます。



センサーを使って、黒い地面(線)に出会ったら走行を止めるプログラムを作ってみましょう。

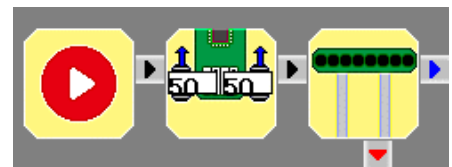


まず、左右のモーターパワー＝50 で走るプログラムを作ります。



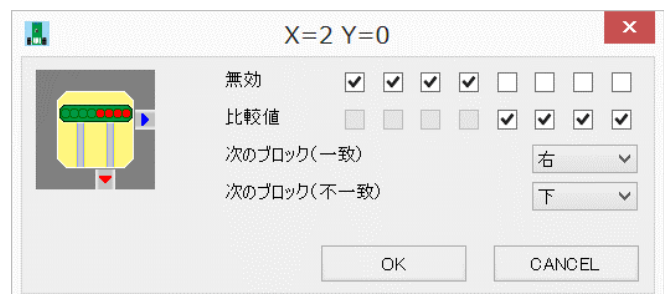
次に、センサーの値をチェックするセンサブロックを置きます。

センサブロックをダブルクリックして、設定画面を出します。

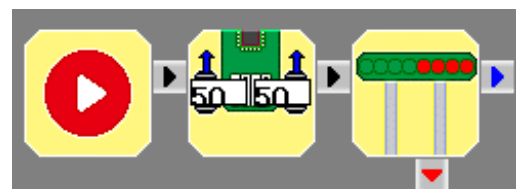


設定画面ではセンサーが 8 個ありますが、左側の4個は使わないので「無効」にチェックを入れます。

今回は、4個のセンサー全部が ON か OFF かで判断するので、「比較値」の4個全部にチェックを入れます。



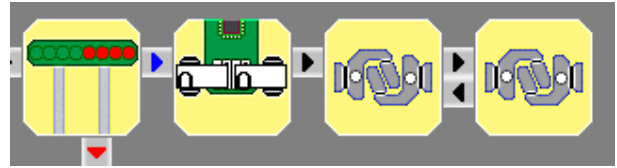
「OK」ボタンをクリックすると、その設定が反映されます。



センサーブロックから先は、条件によって2つの処理に分かれます。

まず、センサーが4つ共 ON の時(黒い地面に出会った時)は、ブロックの右へ処理が移動します。

この時は、モーターを止めましょう。

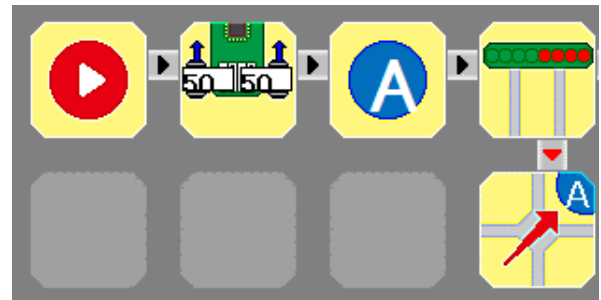


センサーが ON になっていない時(黒い地面に出会っていない時)は、処理がセンサーブロックの下へ移ります。

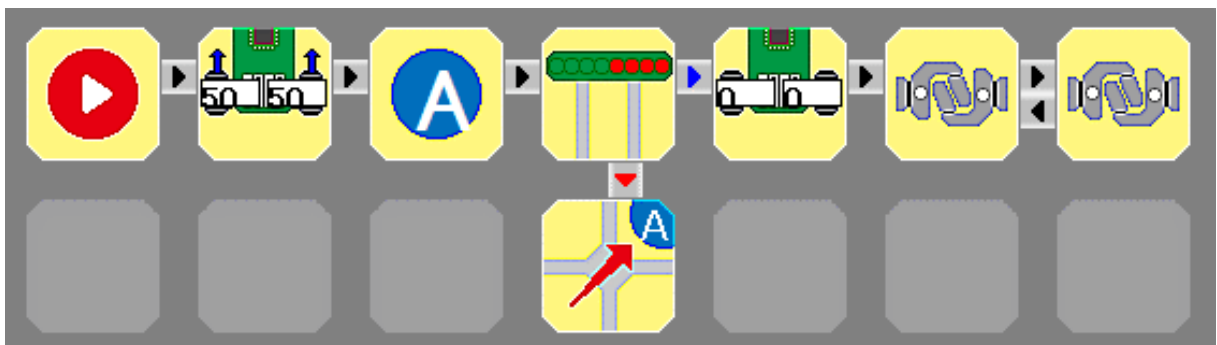
この時はさらに走り続けるといけません。

センサーブロックより前に処理を戻して、センサーのチェックをくり返します。

ラベルブロックとジャンプブロックを使いましょう。



プログラム全体はこのようになります。



プログラムを転送して、黒い線があるコースの上で走らせてみましょう。

地面に黒い線が描かれているところで、マイコンレーサーが止まりましたか？

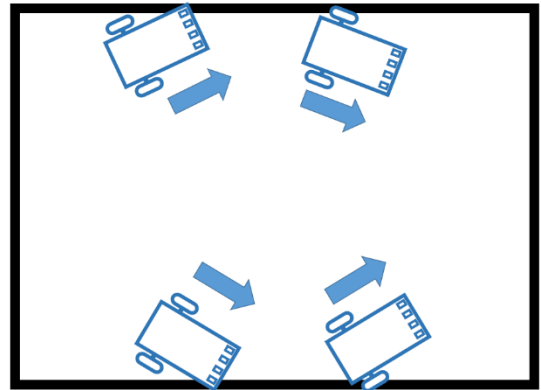
### ● 黒い線から外へ出ないようにする

今のプログラムは、黒い線に出会うと走行が止まります。

黒い線に囲まれたエリアの中を走って、線に出会ったら向きを変えて外へ出ないようにプログラムを考えてみましょう。

「黒い線から外へ出ない」動きを考えると、

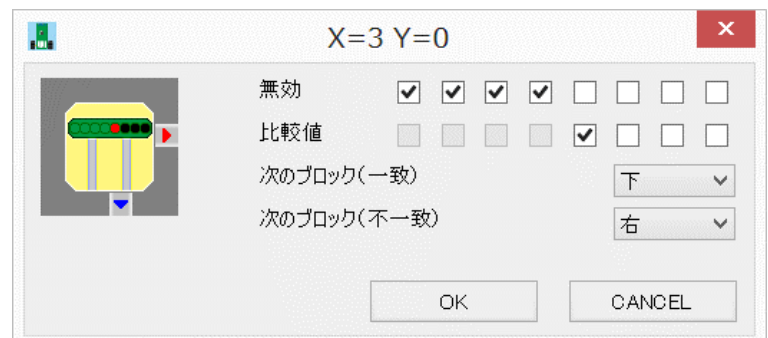
- 左側のセンサーが黒い線にのって ON になったら、マイコンレーサー全体を右へカーブさせる。
- 逆に、右側のセンサーが黒い線にのって ON になったら、マイコンレーサー全体を左へカーブさせる。
- それ以外は真っ直ぐ進むということになります。



黒い線で止まるプログラムを改造します。

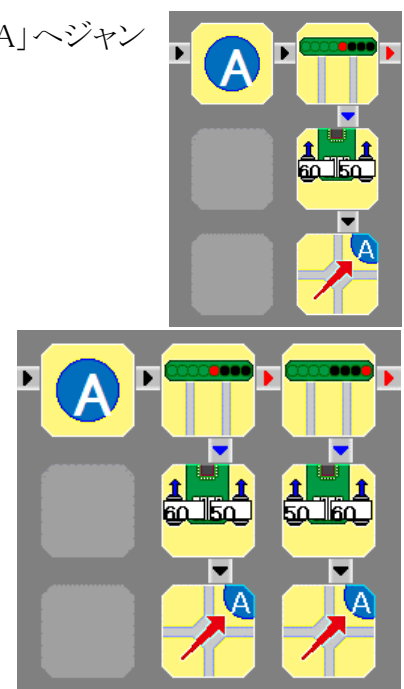
センサーブロックをダブルクリックして、条件を変更します。

一番左側のセンサーが ON だったら(一致)下のブロックへ、そうでなければ右のブロックへ行くようにします。

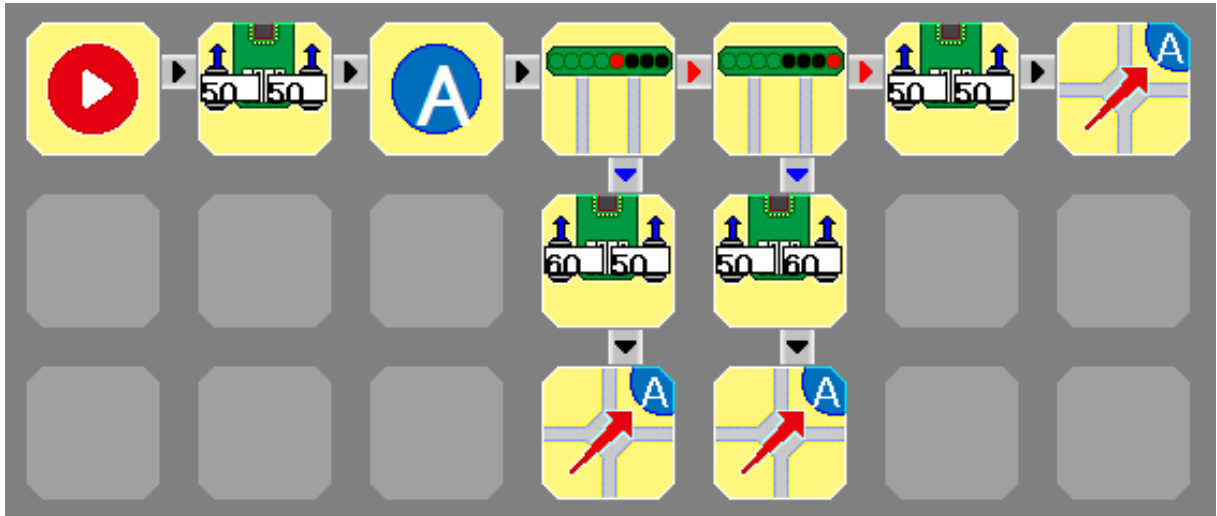


下には右へカーブさせるモーターブロックを置いて、ラベル「A」へジャンプして戻します。

同じように、一番右側のセンサーが ON だったら、左へカーブさせるプログラムを、続けて並べます。



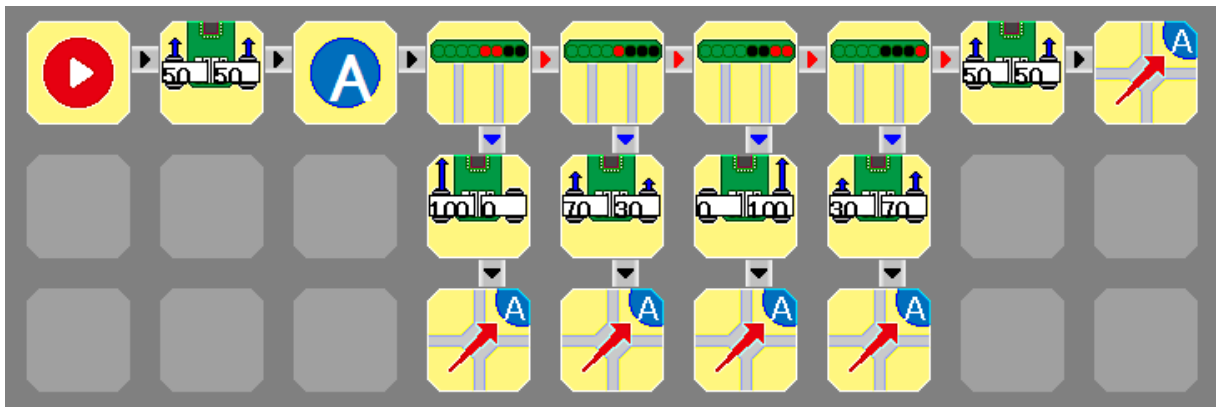
最後に、どちらの条件も成り立たない時は、真っ直ぐ前進します。  
プログラム全体ではこのようになります。



プログラムを転送して、テストコース上で走らせてみましょう。  
黒い線から外へ出ないようにうまく走るでしょうか。

黒い線に出会った時に、うまく中へ戻らない時は、カーブするモーターブロックのパワーの値を調節してみてください。

さらに「センサーの左側2つが ON になった場合」「センサーの右側2つが ON になった場合」を追加して、大きくカーブするようにすると、うまく行くかもしれません。



黒い線から外へ出ないプログラムの改良版。センサーの条件を4つにしています。