

親子ロボット制御プログラム制御講座 ～mBotで遊ぼう～

清水南公民館 小澤晴樹



この講義の流れ

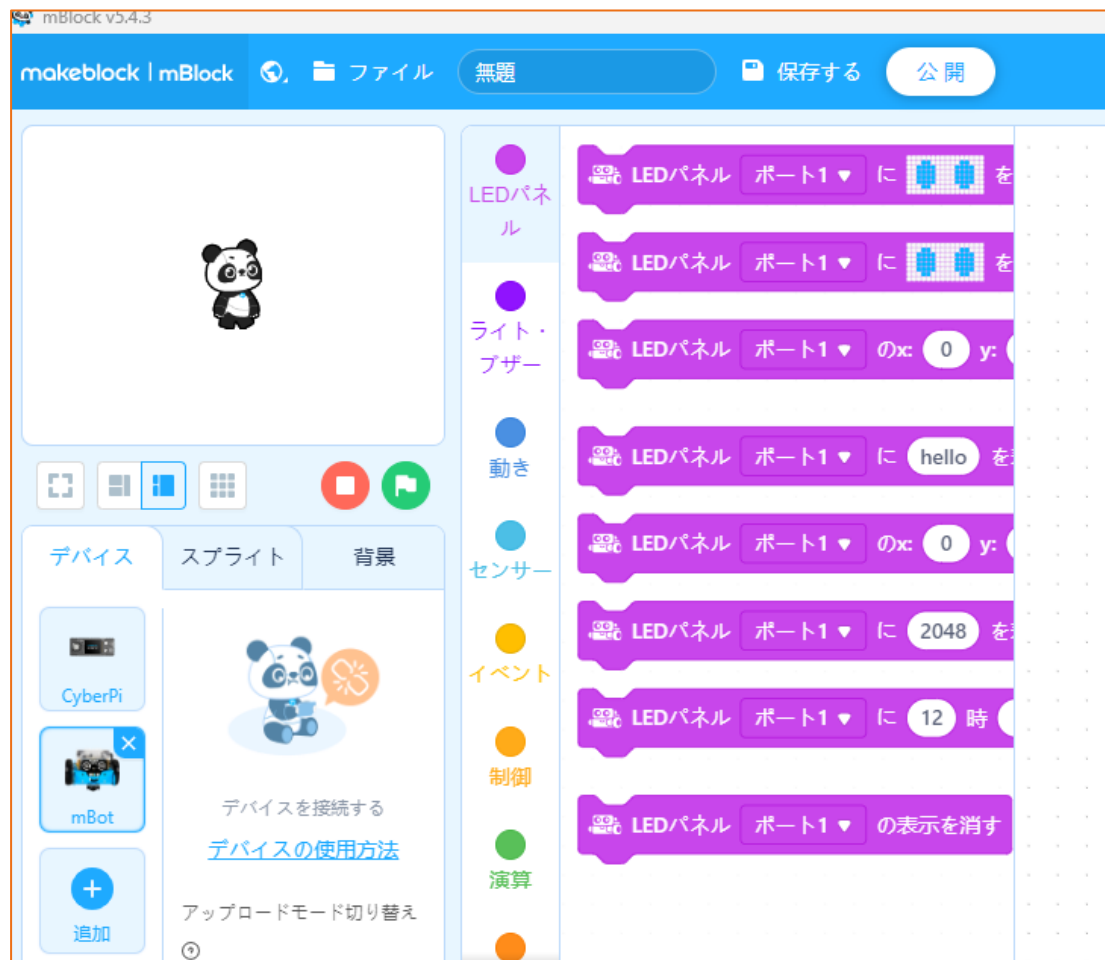
- ① mBot（エムボット）とは
- ② mBotとPCを連動する
- ③ mBotとPCを連動させよう（LED、ブザー）
- ④ mBotとPCを連動させよう（モーター）
- ⑤ mBotとPCを連動させよう（各センサー）

mBotとは

モーター（車輪）、LED（光）、ブザー（音）をプログラムで制御する。

mBot

mBlockで簡単にロボット制御のプログラムが使える。



mBot (モーター駆動)



モーター駆動
(前進、後退、右折、左折)

mBot (LEDライト)

LED(赤、緑、青色)



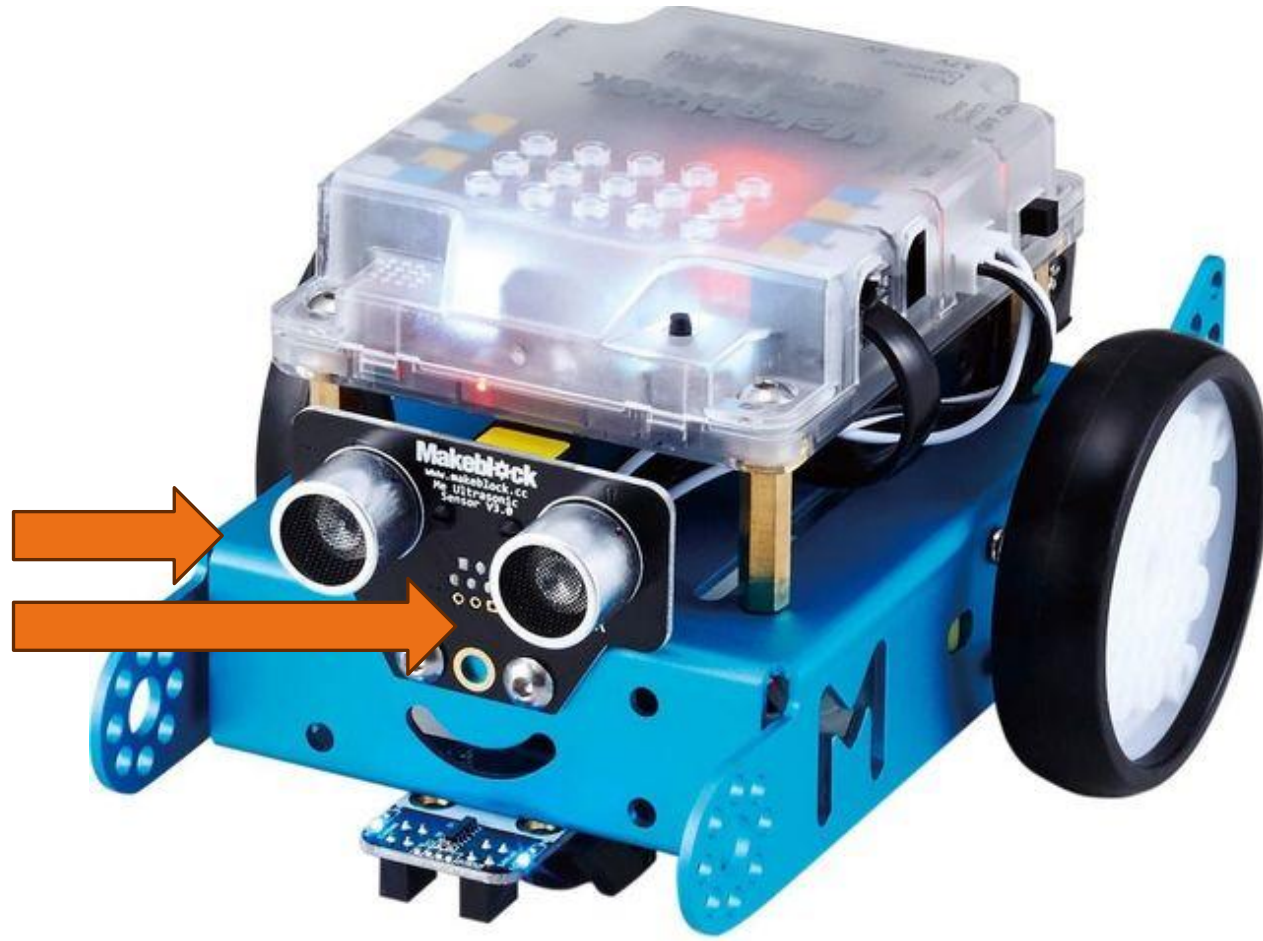
mBot (ブザー：音階を鳴らす)

ブザー

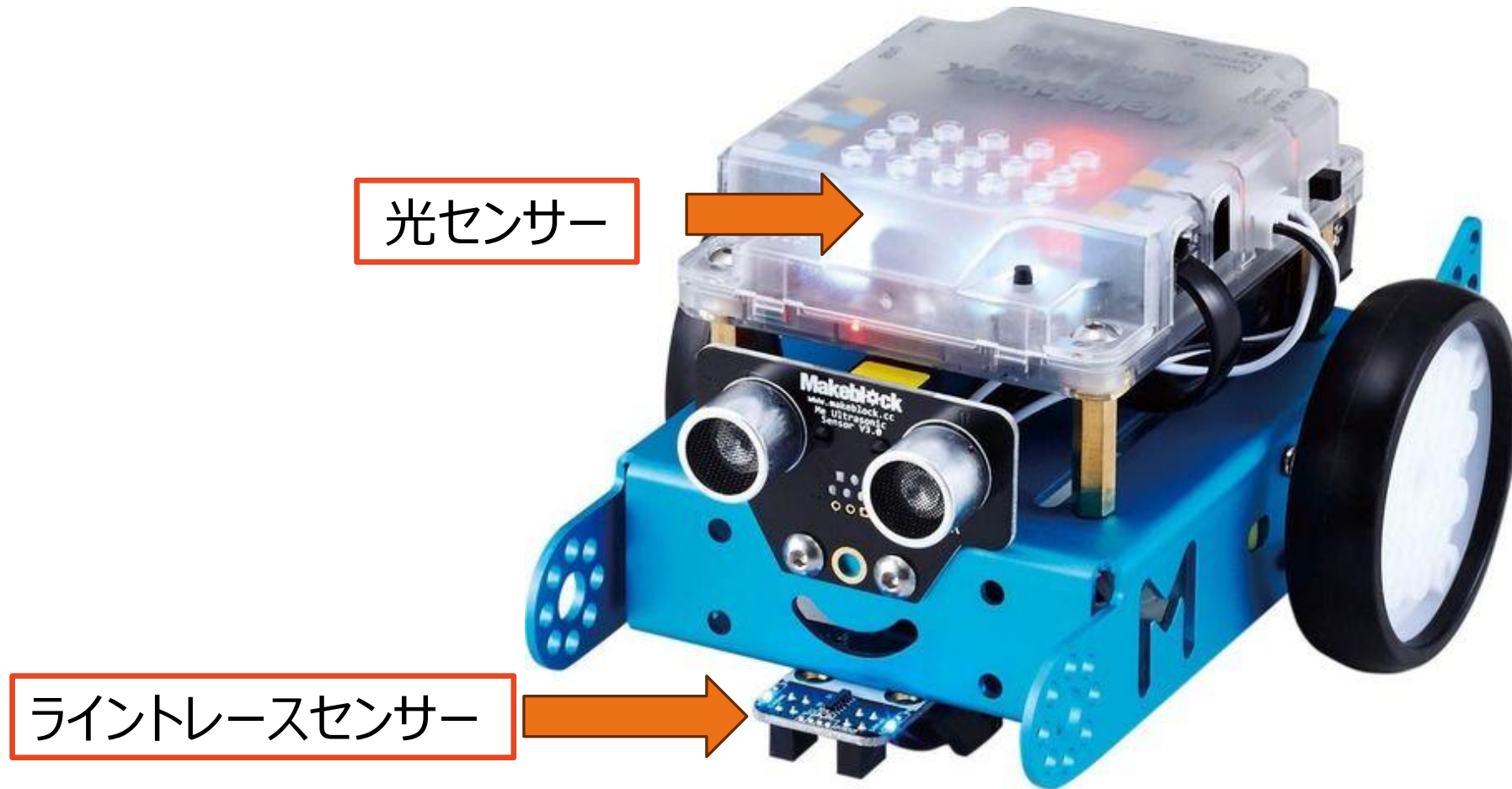


mBot（超音波センサー：障害物検知）

超音波センサー



mBot（光センサー：ライントレース）



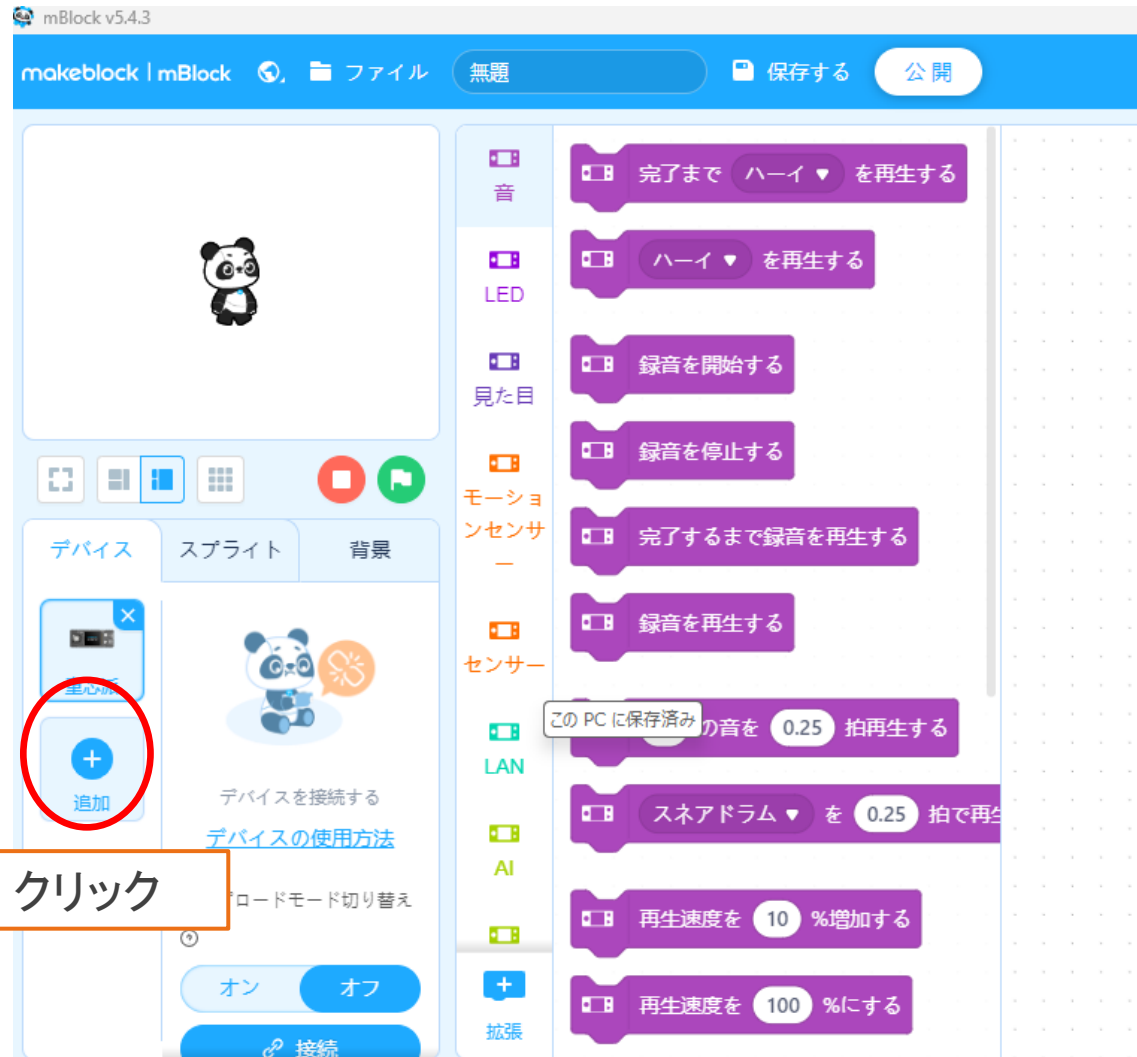
mBotとPCを連動する

mBlock（PC）とmBotを dongle（Bluetooth）で連動する。

mBlockを起動する



このアイコンをクリック



クリック

mBlockを起動する



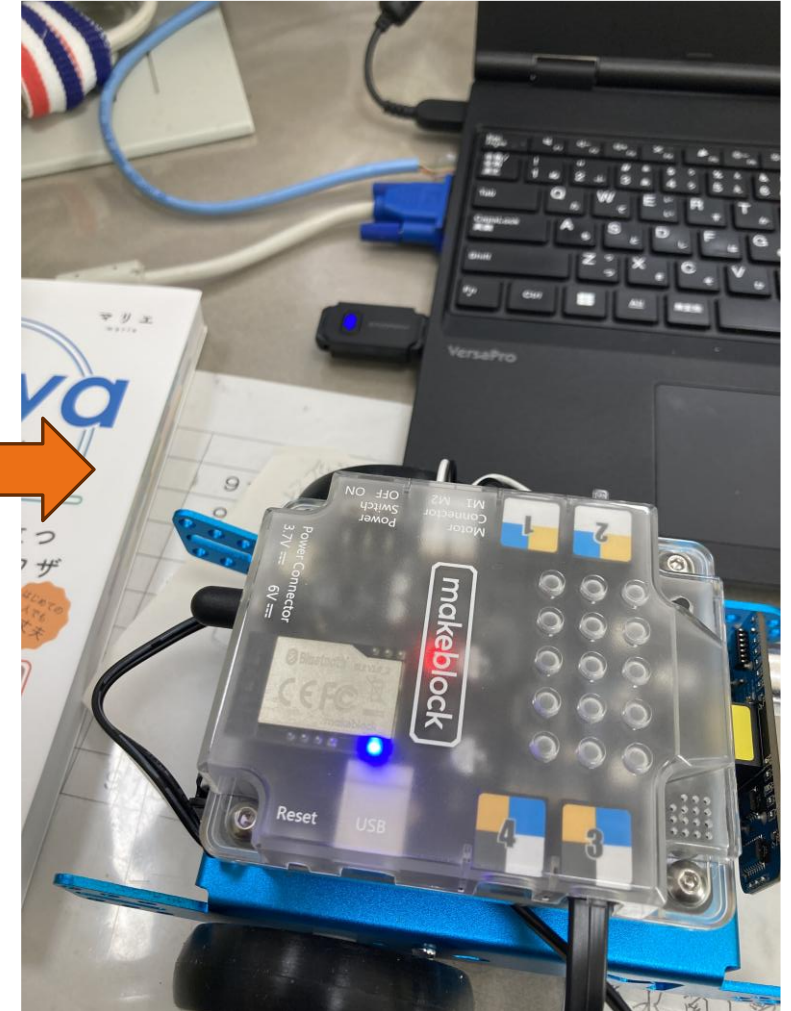
クリック

mBotが追加されていることを確認



ドングル（mBlockとmBotを連動）

PCのUSBポートに差し込んで、
mBotとBluetoothで連動する。



mBlockとmBotを接続（※ライブモード：PCでmBotを制御）

（※アップロードモード：PCで作成したプログラムをmBotに書き込み、mBot単体で動作する。）



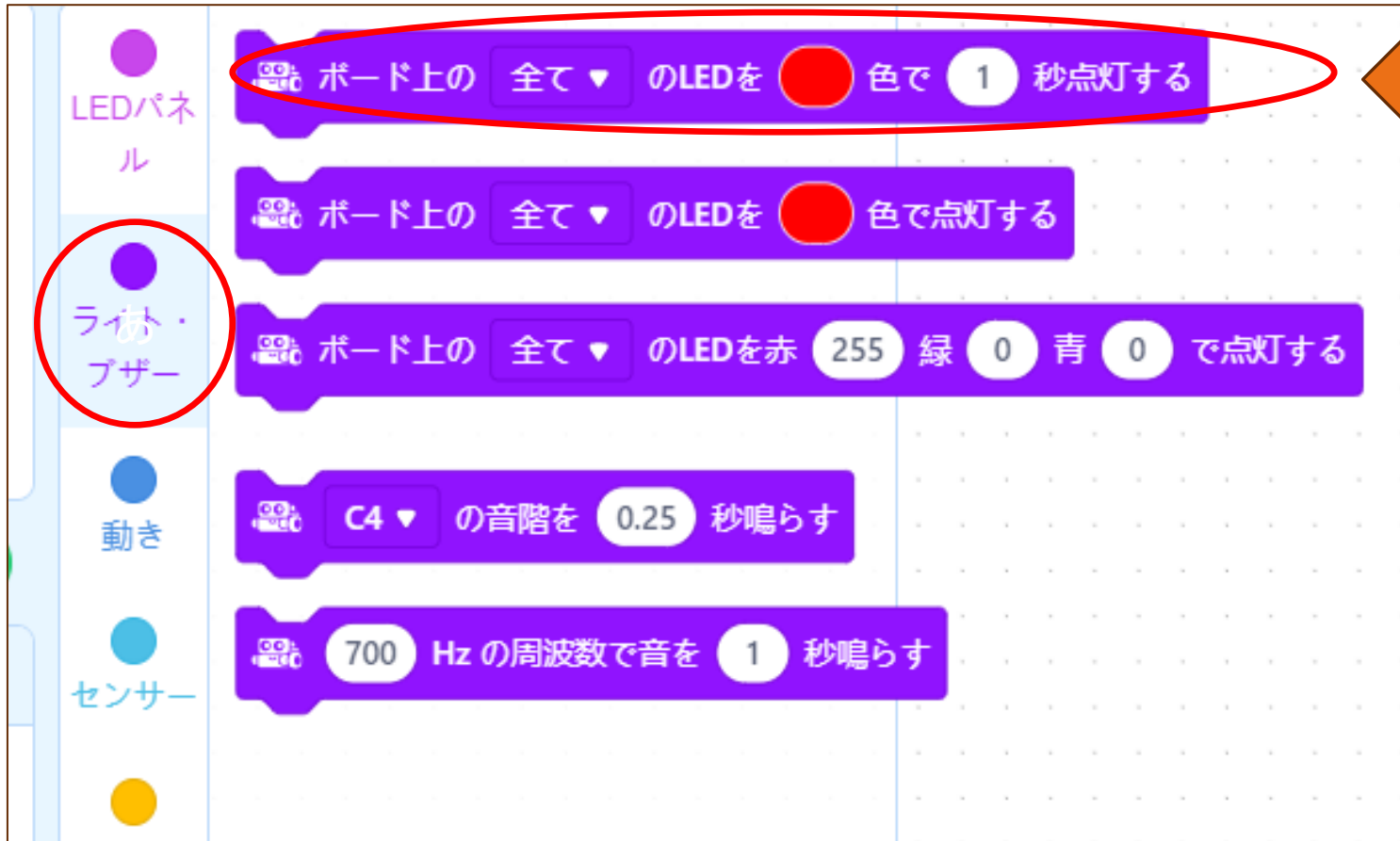
ライブモード

mBotを動作させよう

LED（光）、ブザー（音階）

LEDを点灯・消灯させる。

① 「ライト・ブザー」を選択（「LEDパネル」では無いです。）



② クリックして赤く光ることを確認しよう

③ 「全て」⇒「左」や「右」
「赤色」⇒「いろんな色」
「1秒」⇒「10秒」
に试着てみよう

ブザー（音階）を鳴らす。

① 「ライト・ブザー」を選択



② 「ドレミ。。。ド」を鳴らす

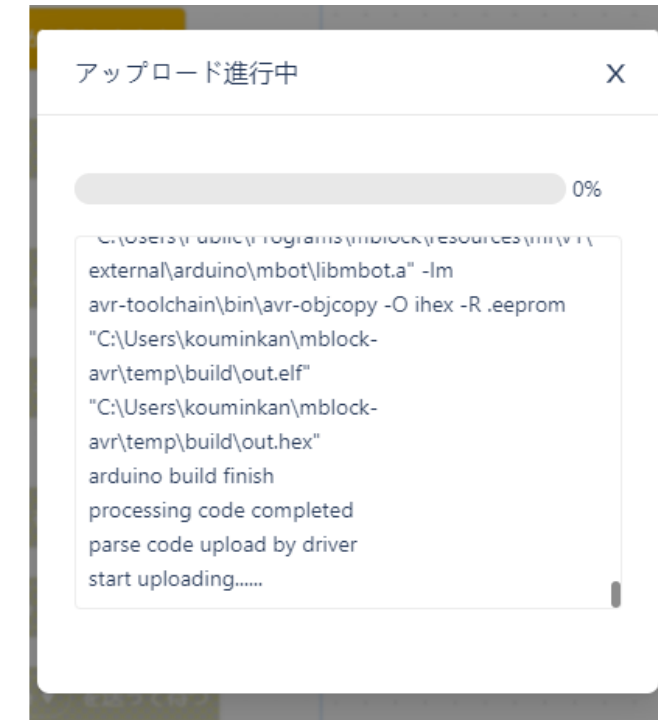


アップロードモードにして、mBot単体で動かす

① 「mBotが起動したとき」(イベント)を追加

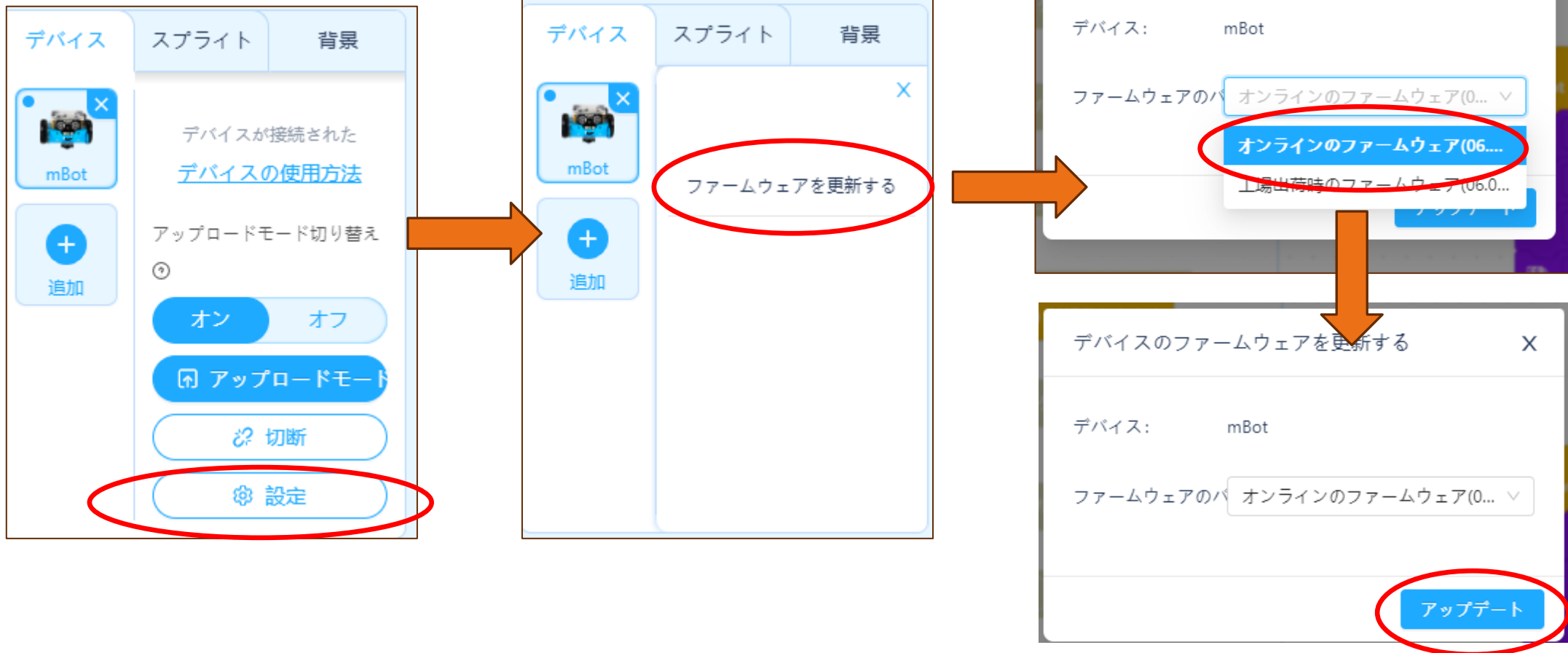
アップロードモードをオンにすると選択できる。

mBot (mcore) が起動したとき



アップロードモードからライブモードに戻す

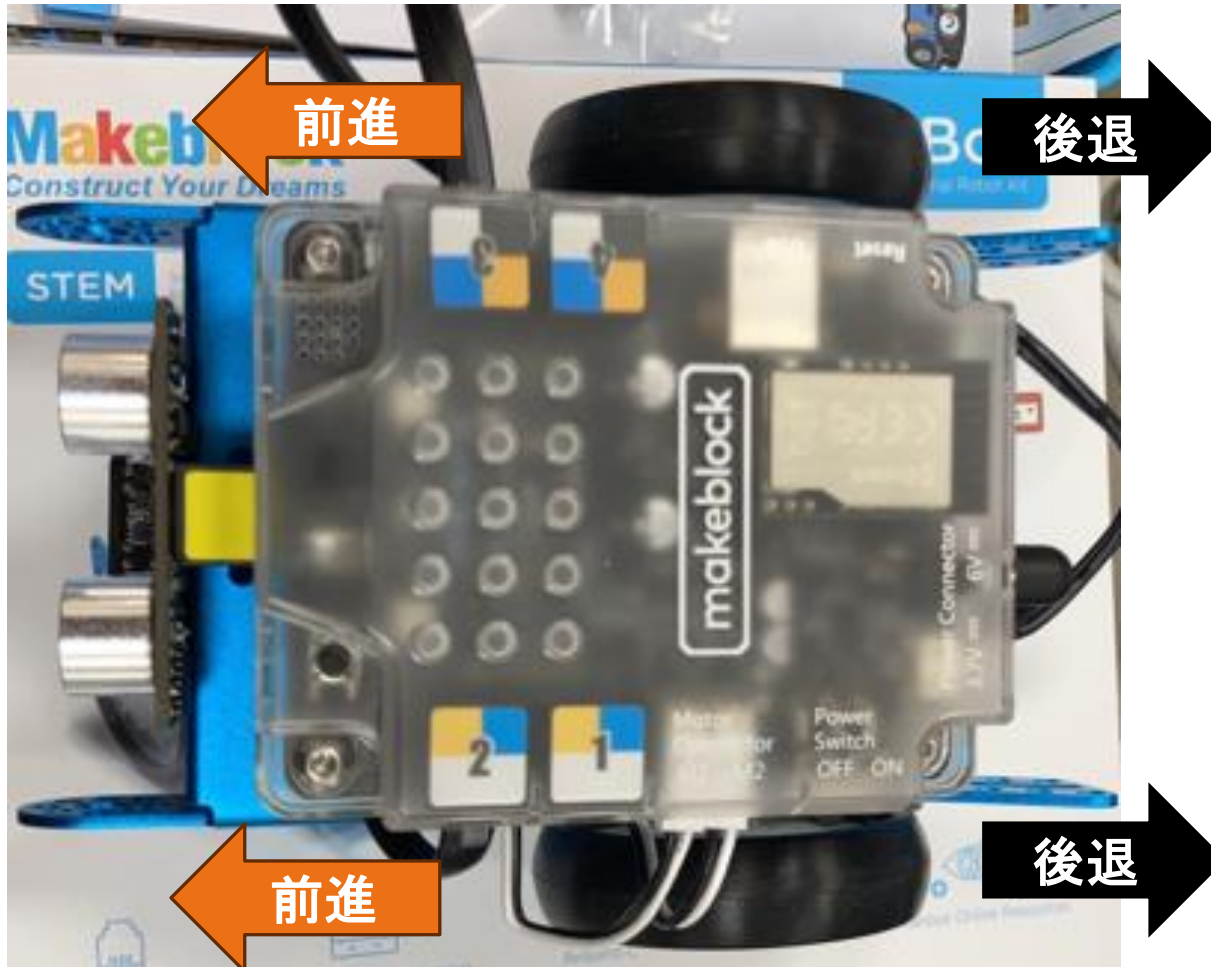
アップロードしたプログラムを前の状態に戻す



mBotを動作させよう

モーター（前進・後退）

モーターの説明



前進・後退・右折・左折

① 落下して壊れないために、mBotを床においてください。

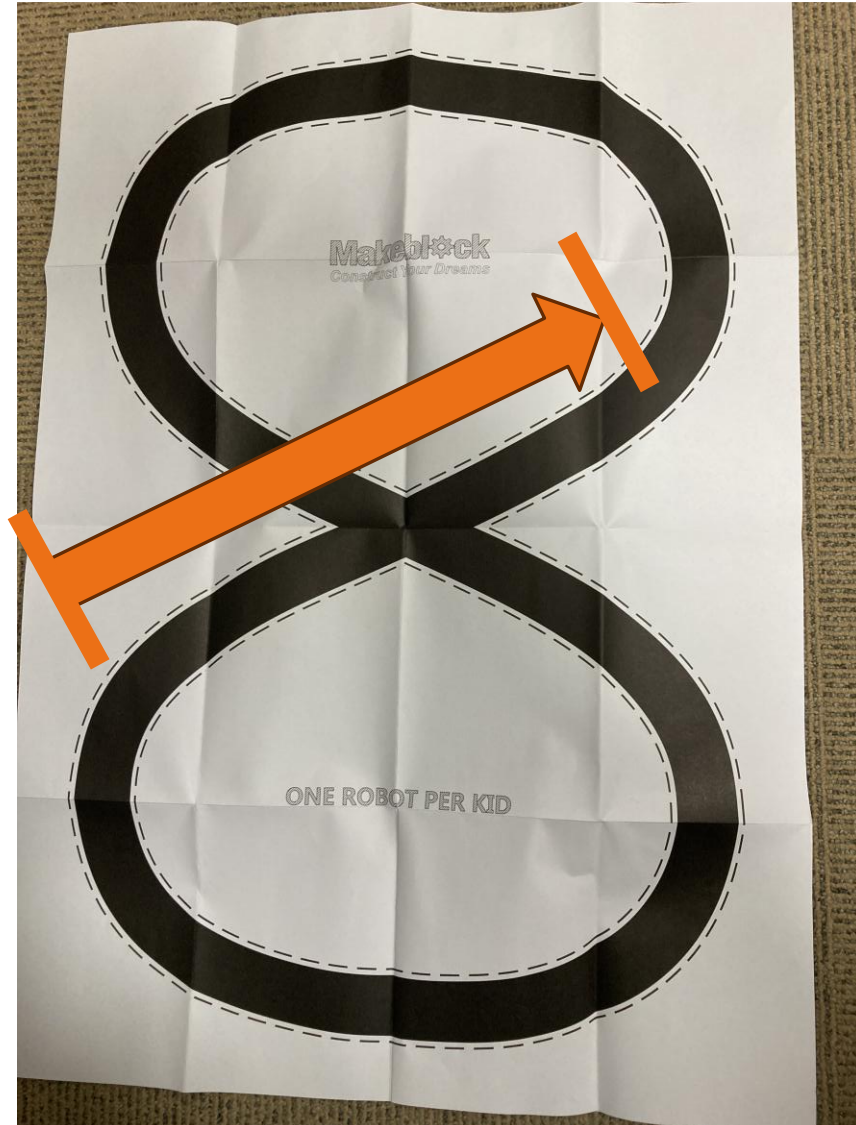


上から順番に動かしてみよう

暴走したら、「動きを止める」を押すか、mBotの電源を切ろう

動きを止める

コースを走ってみよう！

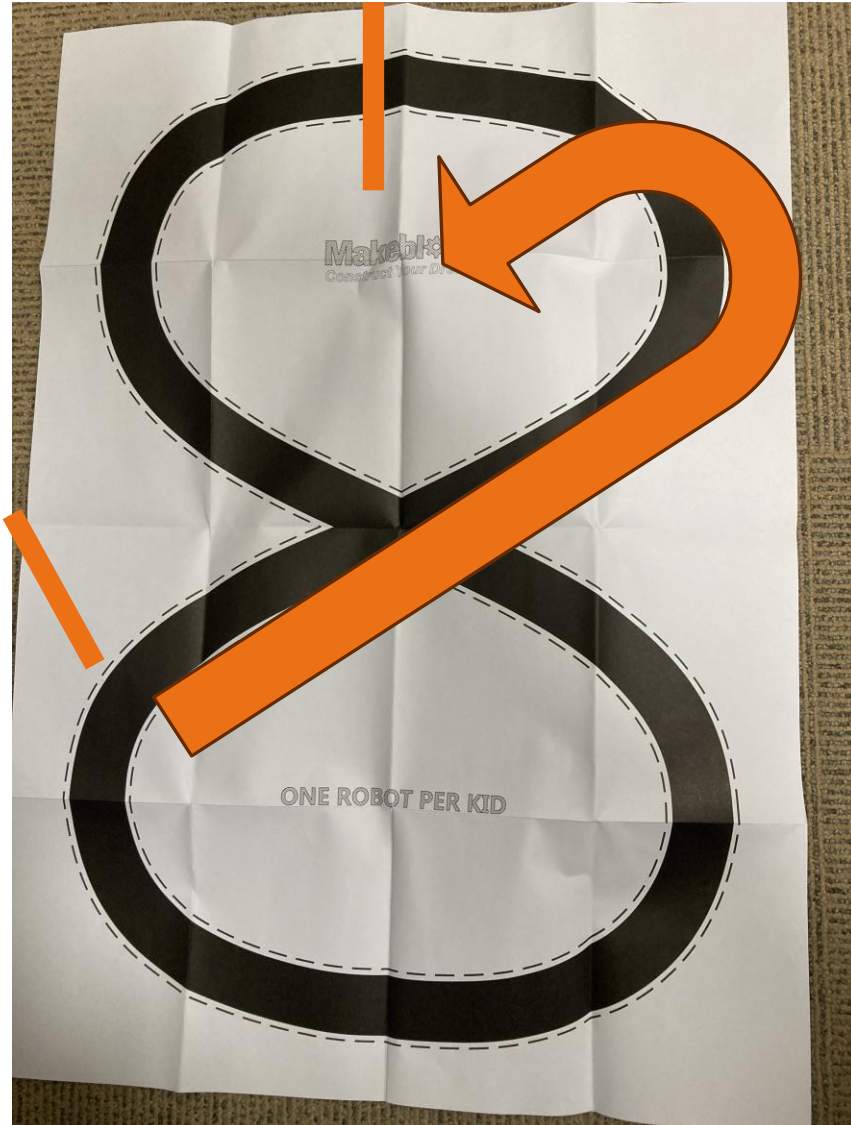


① 直線のコースを走ってみよう



ここが旗です

コースを走ってみよう！



① 直線＋カーブのコースを走ってみよう

プログラム例

が押されたとき

前向きに 50 %の速さで 2 秒動かす

左向きに 50 %の速さで 0.5 秒動かす

前向きに 50 %の速さで 0.5 秒動かす

左向きに 50 %の速さで 0.5 秒動かす

前向きに 50 %の速さで 0.5 秒動かす

左向きに 50 %の速さで 0.5 秒動かす

前向きに 50 %の速さで 0.5 秒動かす

mBotを動作させよう

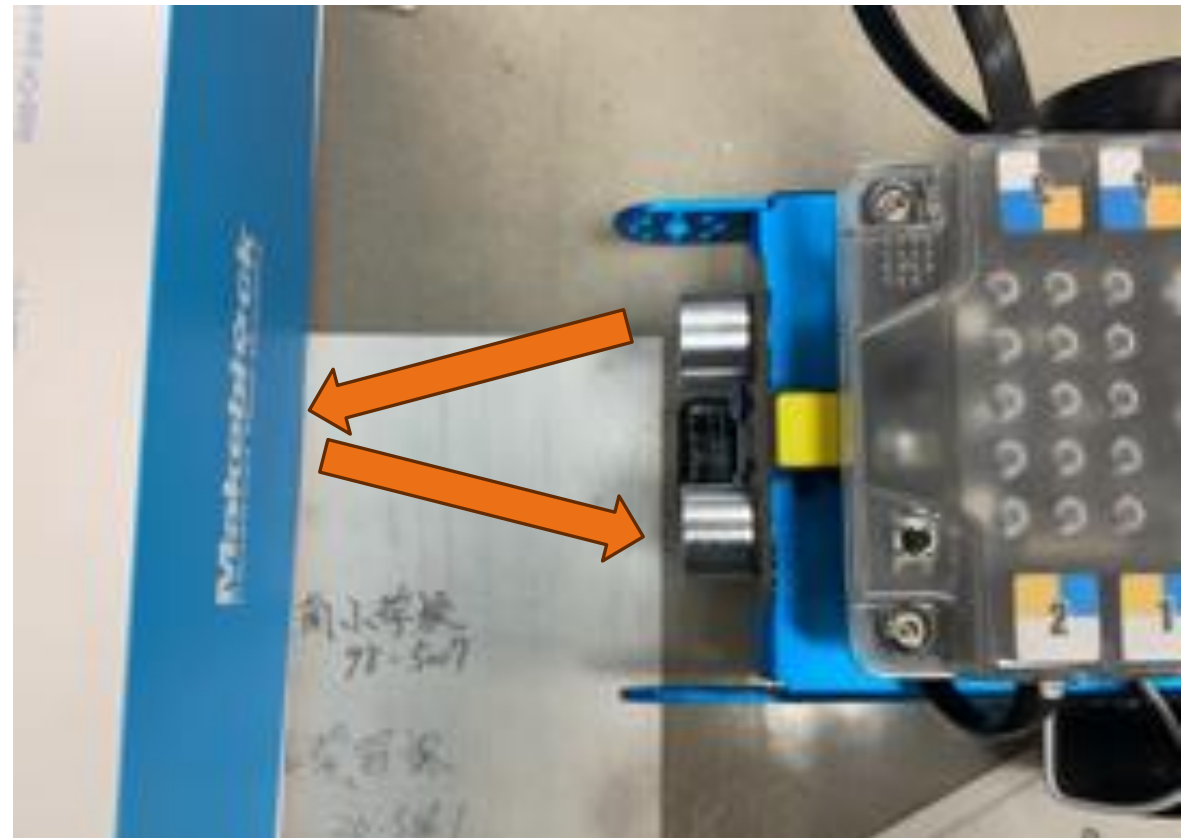
超音波センサー、ライントレースセンサー、光センサー

超音波センサーの説明

超音波センサーの位置



右目から超音波を発射して、左目で受けるまでの時間で距離を測定する。



超音波センサー（障害物との距離）を使う。

① 変数を選択

② 変数を作るをクリック

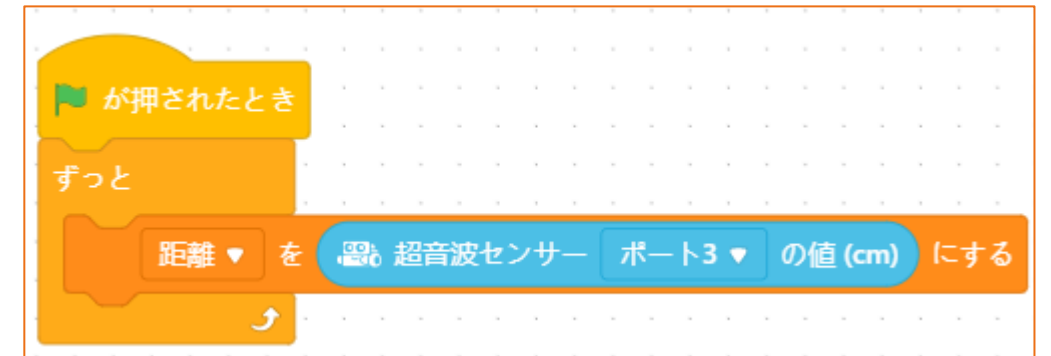
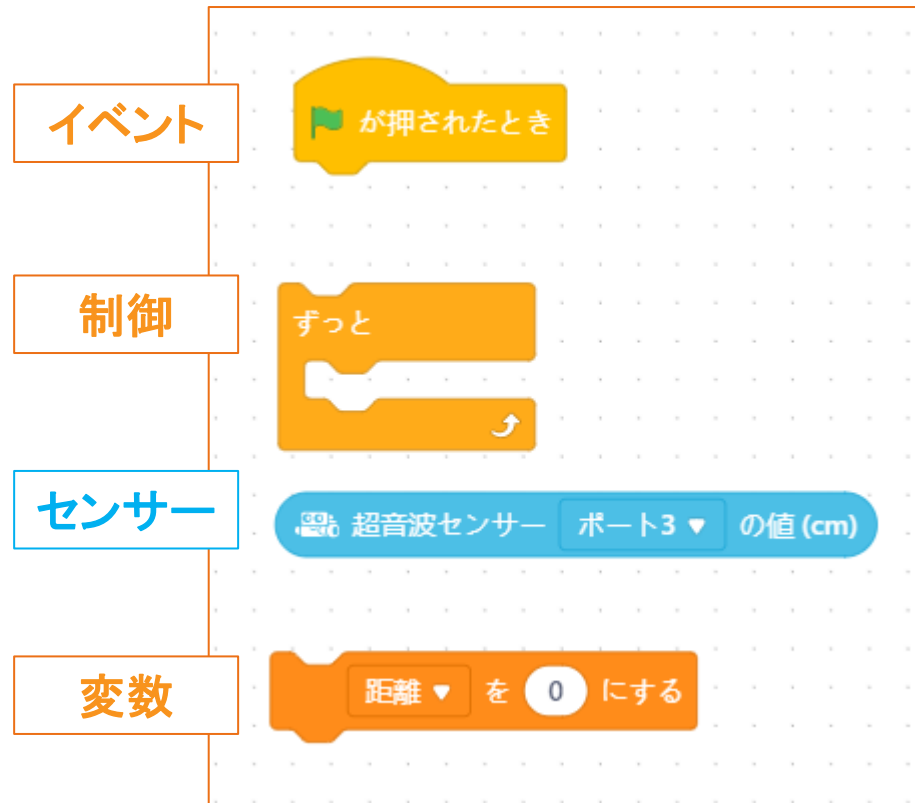
③ 変数名を「距離」(すべてのスプライト用)を作成



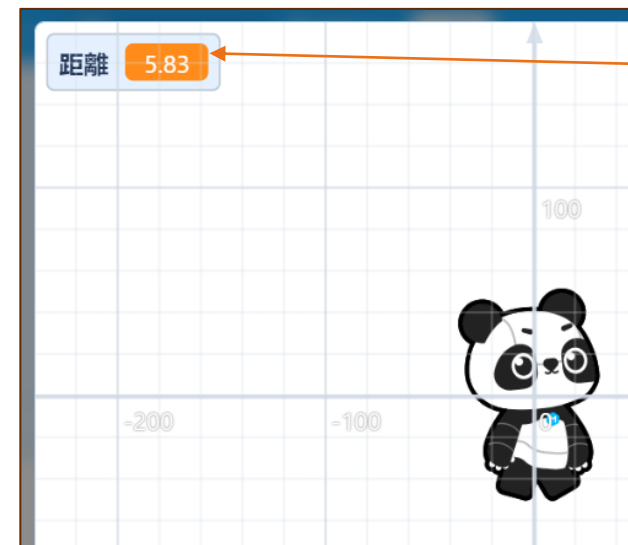
The image illustrates the steps to create a variable named 'Distance' in Scratch. It shows the 'Variables' menu, the 'New Variable' dialog, and a list of variables.

The 'Variables' menu shows the '変数' (Variables) option selected. The 'New Variable' dialog shows the variable name '距離' (Distance) and the scope 'すべてのスプライト' (All Sprites). The list of variables shows '距離' (Distance) as the first variable.

超音波センサー（障害物との距離）を使う。

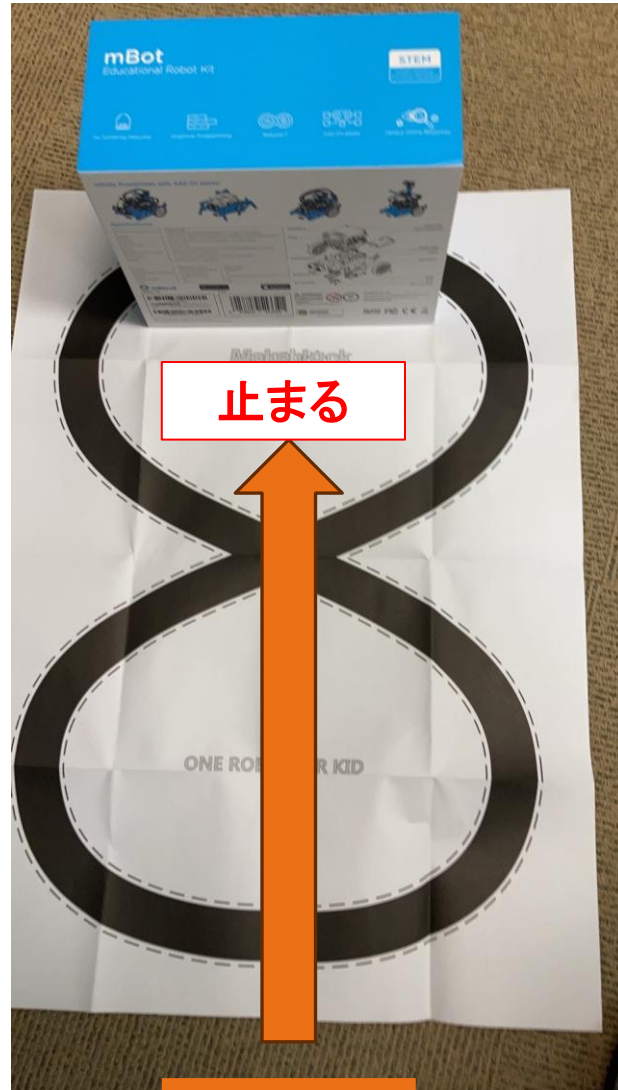


センサーの値を、ずっと変数「距離」に読み込んでいる。

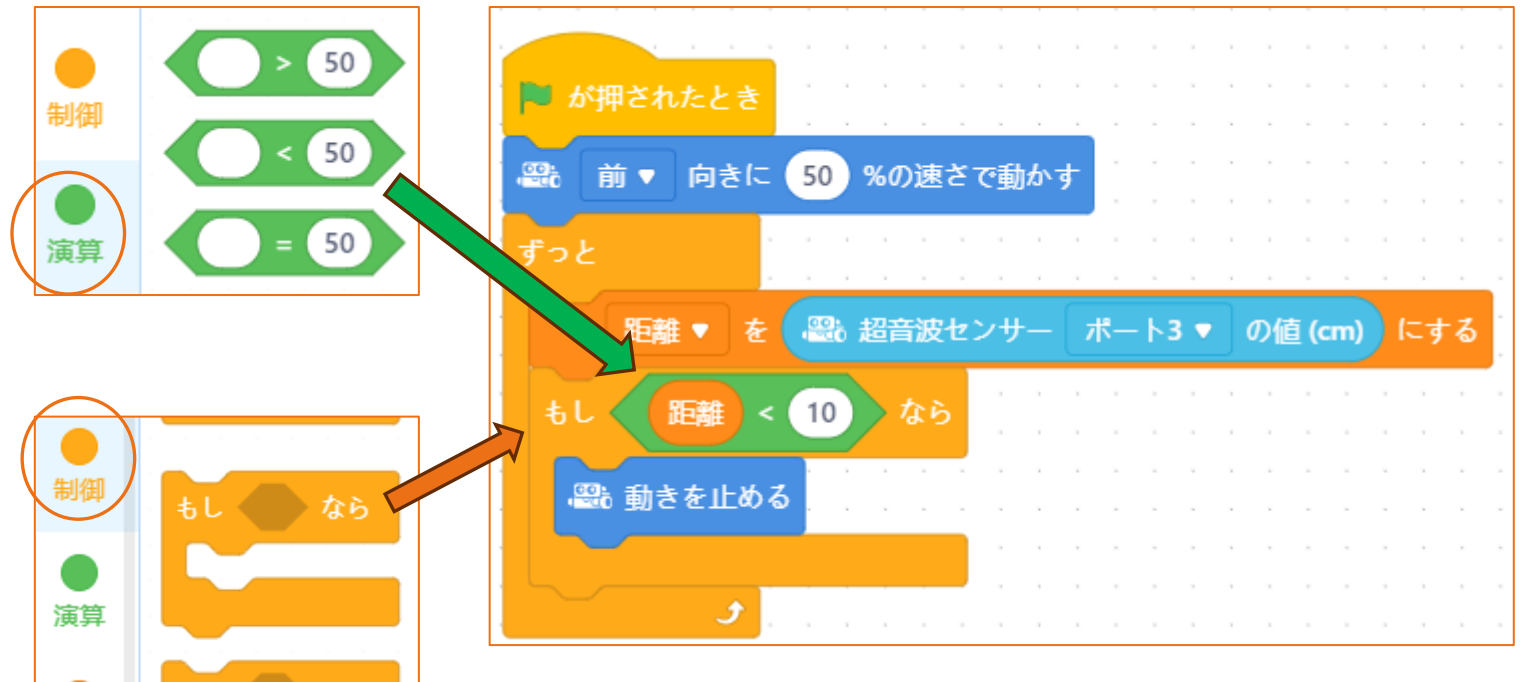


ここの値が変わることを確認しよう

センサーで障害物の前で止まる



プログラム例



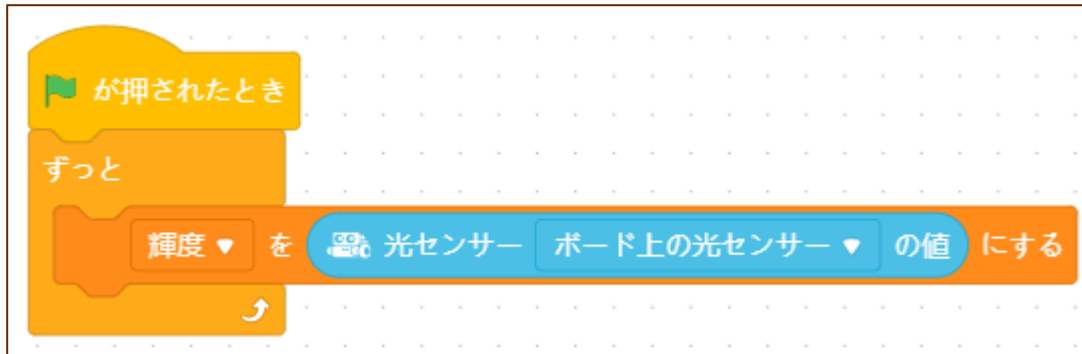
前方に進んで、「距離」が10cm以下なら止まる

光センサーの説明

光センサーの位置



光センサー（輝度の計測）を使う。



mBotの上部を手などでふさいで(光を遮る)、値が変わることを確認しよう！

暗くなったら、「蛍の光」を鳴らす。

変数

ブロック
定義

ブロック定義を使おう

ブロック定義名を入力



蛍の光

引数を追加
テキスト

引数を追加
真偽値

ラベルのテキストを追加

取り消し

OK

定義 蛍の光

G4 ▼ の音階を 0.25 秒鳴らす
C5 ▼ の音階を 0.375 秒鳴らす
C5 ▼ の音階を 0.125 秒鳴らす
C5 ▼ の音階を 0.25 秒鳴らす
E5 ▼ の音階を 0.25 秒鳴らす
D5 ▼ の音階を 0.375 秒鳴らす
C5 ▼ の音階を 0.125 秒鳴らす
D5 ▼ の音階を 0.25 秒鳴らす
E5 ▼ の音階を 0.25 秒鳴らす
C5 ▼ の音階を 0.25 秒鳴らす
C5 ▼ の音階を 0.25 秒鳴らす
E5 ▼ の音階を 0.25 秒鳴らす
G5 ▼ の音階を 0.25 秒鳴らす
A5 ▼ の音階を 0.75 秒鳴らす

ブロック定義に
プログラムを設定

光センサーの値が暗く
なったら、ブロック定義の
プログラムを呼び出す

が押されたとき

ずっと

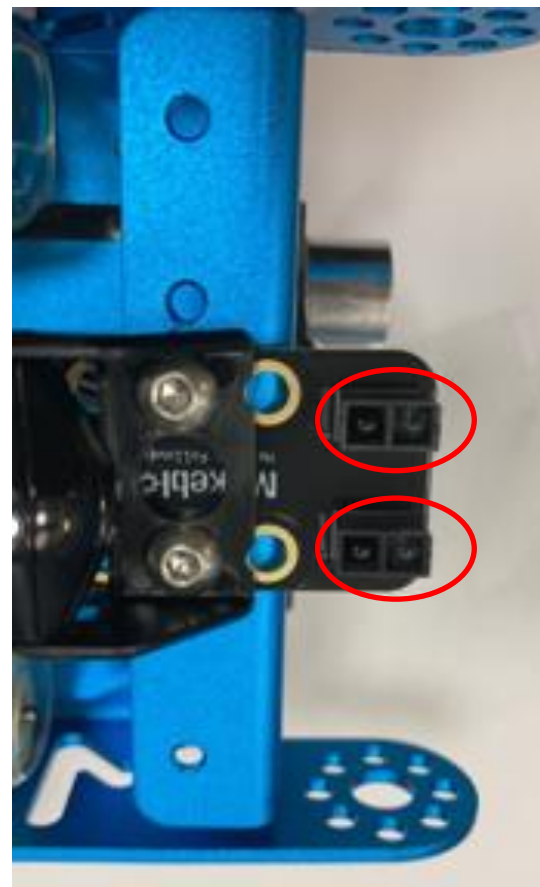
輝度 ▼ を 光センサー ボード上の光センサー ▼ の値 にする

もし 輝度 < 150 なら

蛍の光

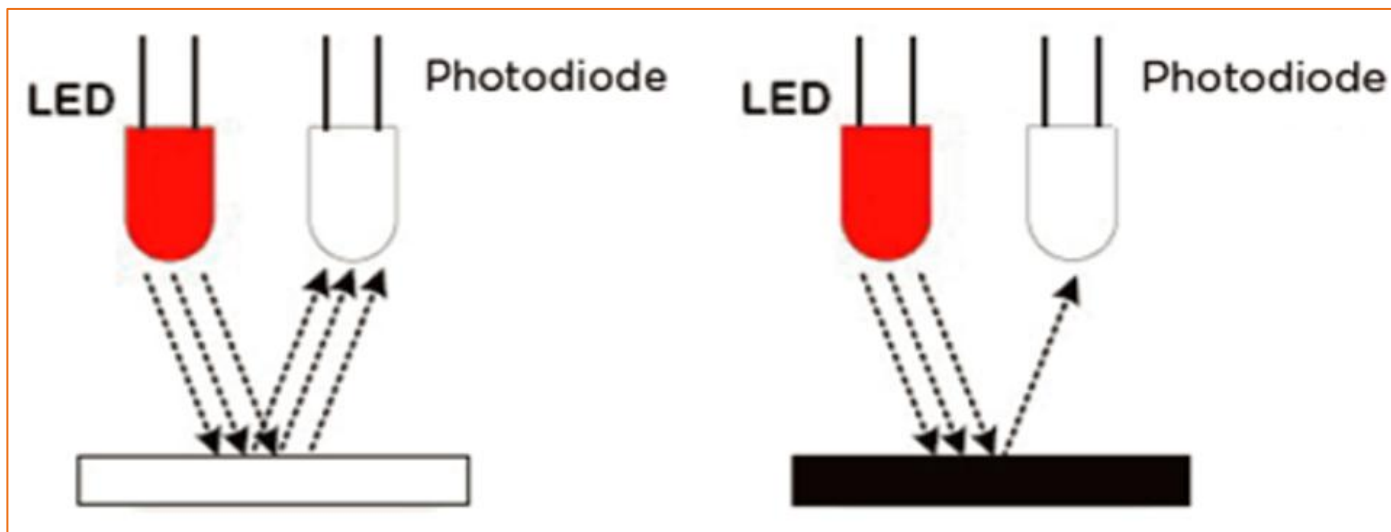
ライントレースセンサーの説明

ライントレースセンサーの位置



反対側からの写真

ライントレースセンサーの説明



ライントレースセンサーは、LEDから発光された赤外線が、白い線や、黒い線にあたり、反射してくる光の量によって、白か黒かを見分けます。

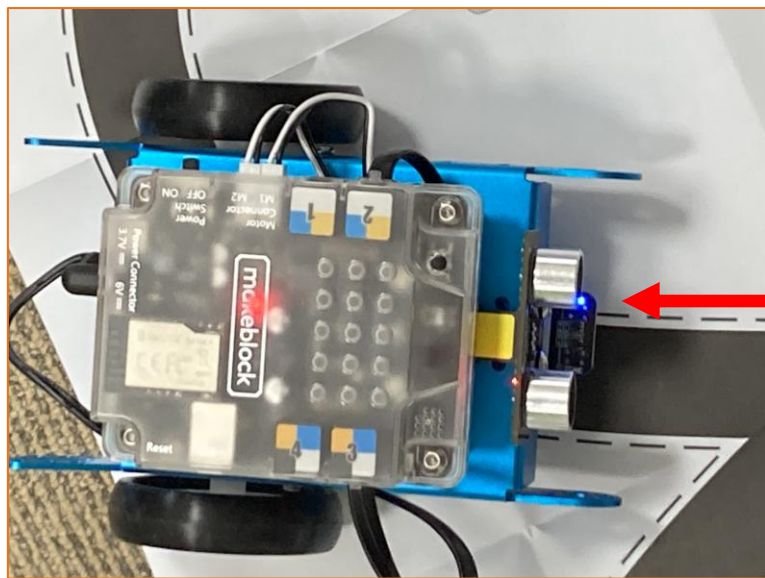
図のように、白い線に赤外線をあてると、反射する光の量が多く、黒い線に赤外線をあてると、反射する光の量が少ないのです。

図 ライントレースセンサーの原理

出典：www.makeblock.com

センサーの値	判定種類
0	両方のセンサーが黒
1	左のセンサーが黒、右のセンサーが白
2	左のセンサーが白、右のセンサーが黒
3	両方のセンサーが白

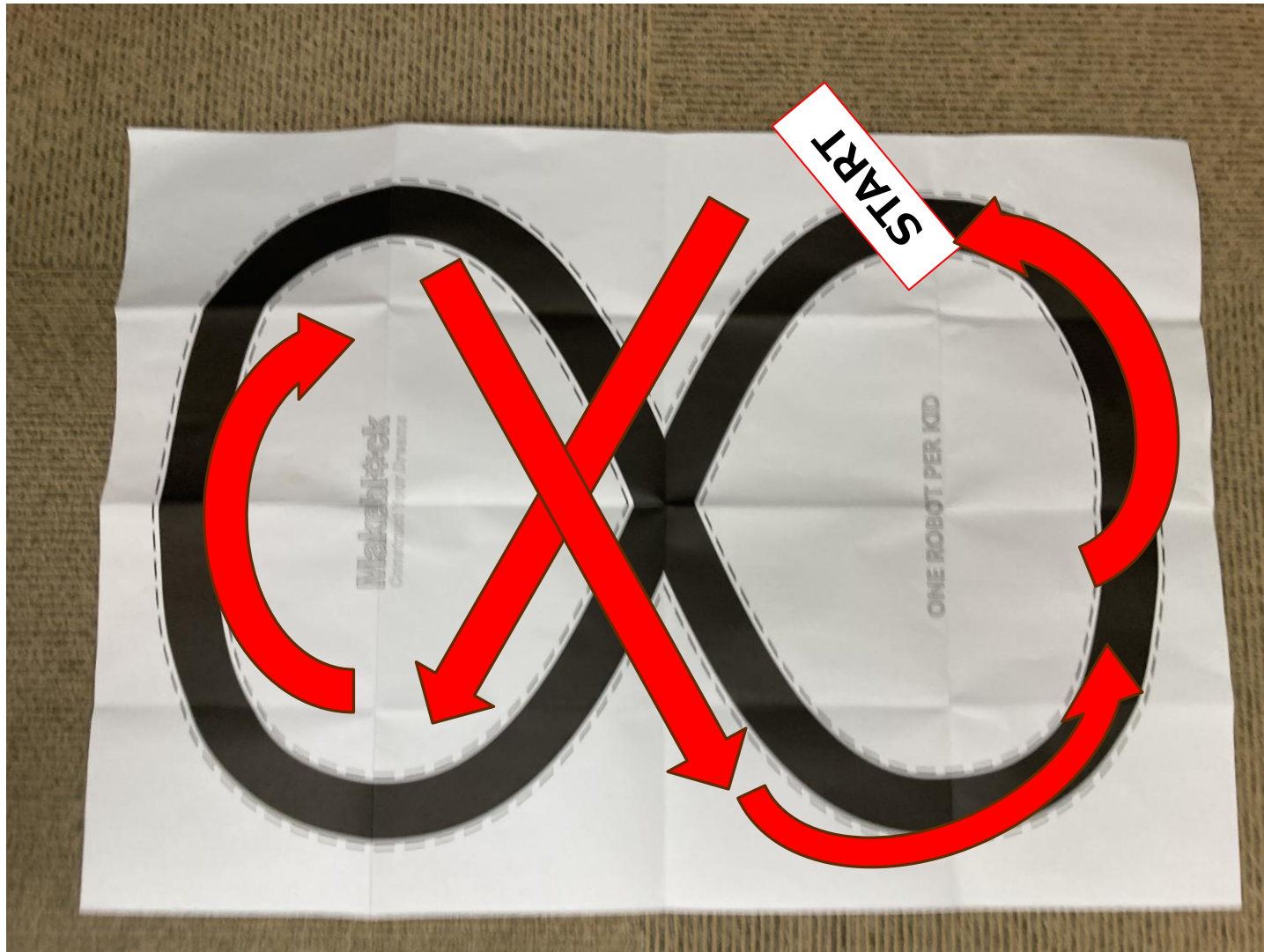
ライントレースセンサー（ラインの色）を使う。



白いと青く光る

mBotの白黒のコース上の位置を変えて、値が変わることを確認しよう！

ライトレースで一周する。



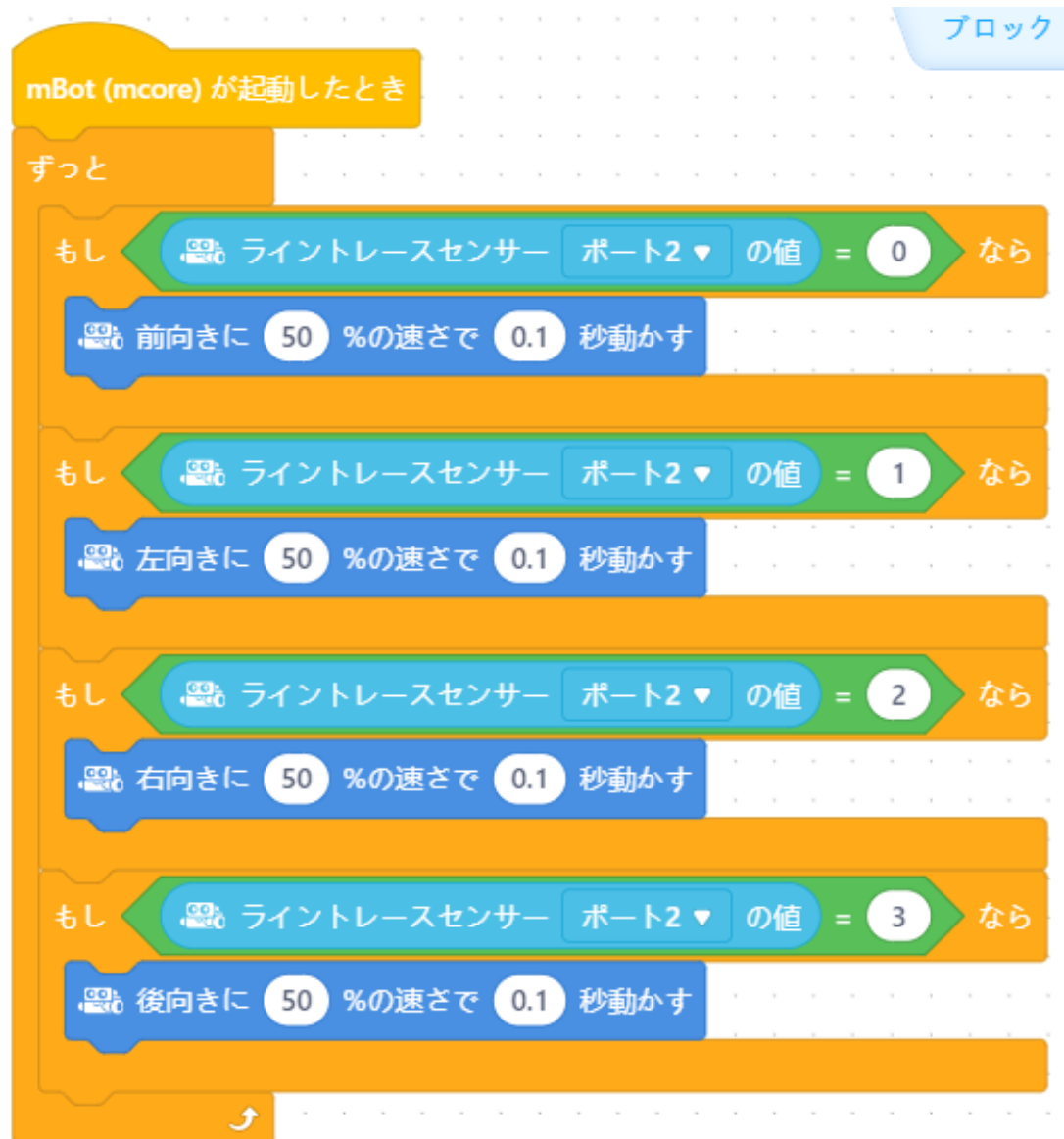
作成するプログラムの内容

① STARTから黒線に沿って
コースを周回する。

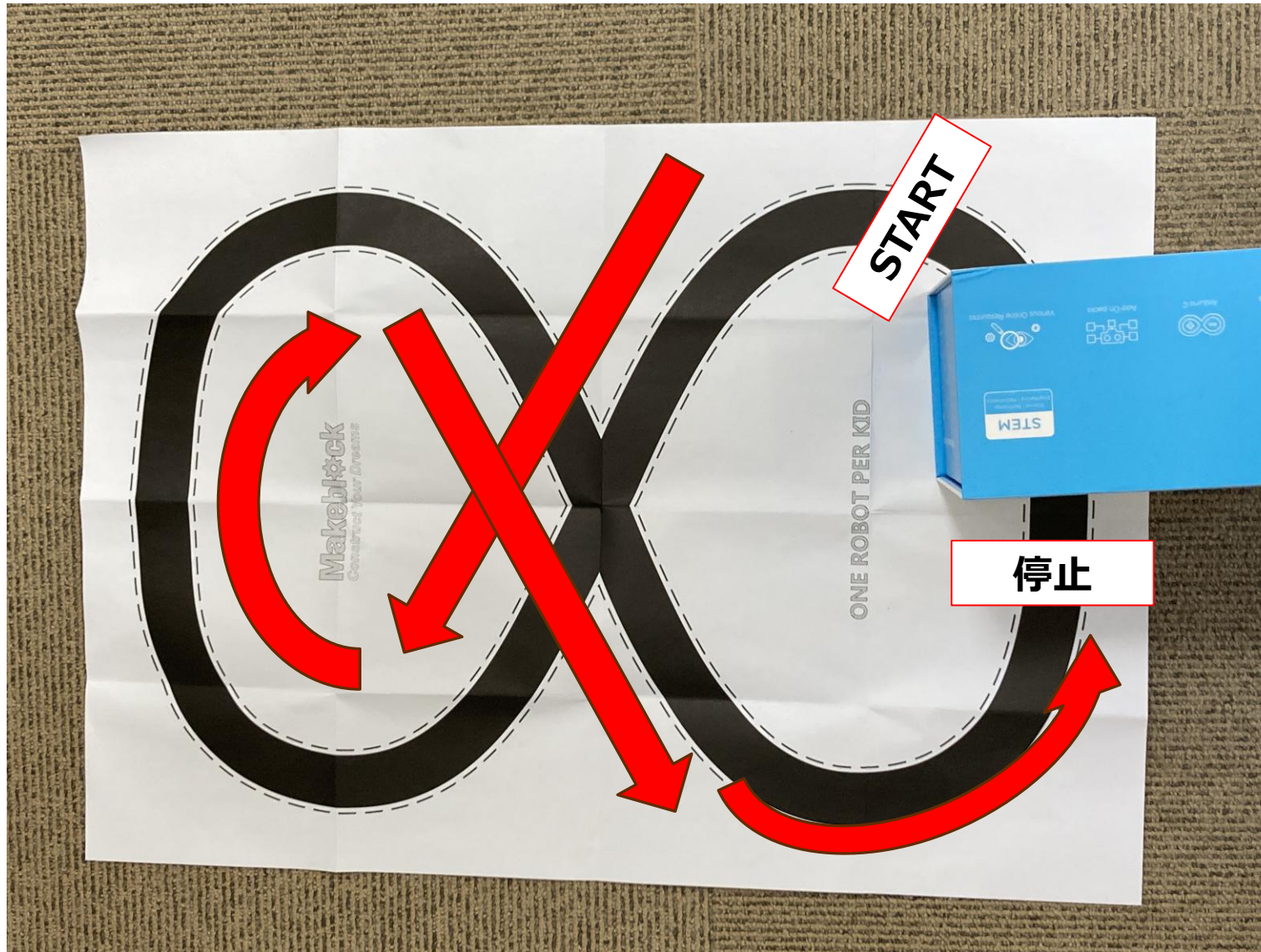
アップロードモード

でプログラムを作成して動作しましょう。

プログラム例



ライントレースで一周、超音波センサーで停止



作成するプログラムの内容

- ① STARTから黒線に沿ってコースを周回する。
- ② 障害物があれば、停止する。

アップロードモード

でプログラムを作成して動作しましょう。

プログラム例



障害物があれば、止まる

ライトレース、超音波センサーで一周する。



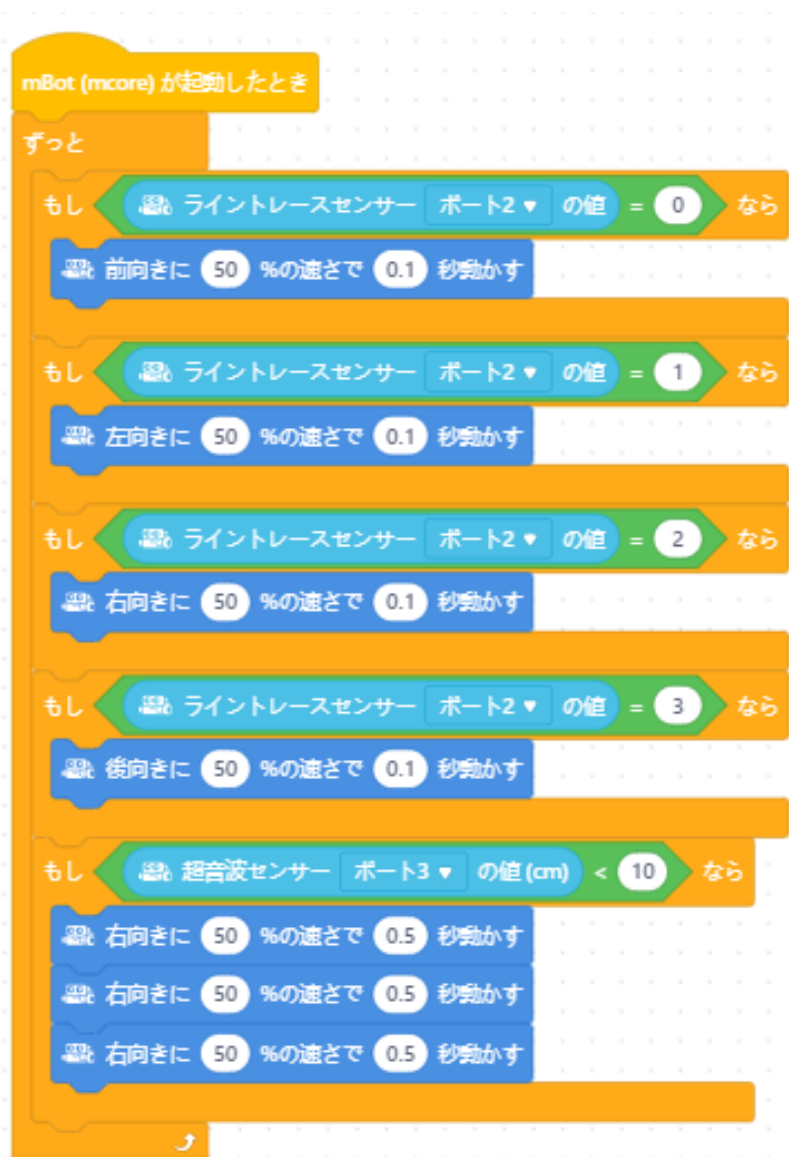
作成するプログラムの内容

- ① STARTから黒線に沿ってコースを周回する。
- ② 障害物があれば、反転してコースを逆の方向に戻る

アップロードモード

でプログラムを作成して動作しましょう。

プログラム例



障害物があれば、反転する。

お疲れさまでした。

