

ロボット教室

「教育版レゴマインドストーム EV3 を使った ロボット制御プログラミング」

未来の科学者の皆さんへ

本教室は EV3 トレーニングロボットを使い、ロボット工学の基礎を学ぶとともに、超音波と色センサを理解し、プログラムによるロボット制御の基礎を学びます。難しいことが多いかもしれませんが、頑張ってみてください。

内 容

1 基本ロボットセット組立

トレーニングロボットの組立て
(「組み立てガイド」41P.まで)

2 プログラミング開発環境確認

3 ロボットを思い通りに

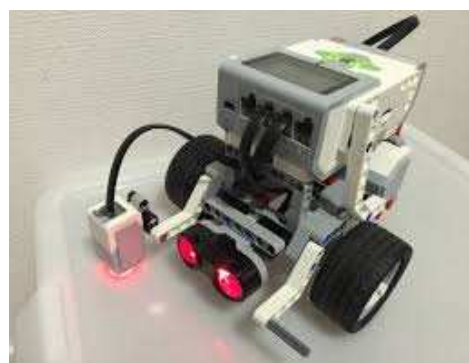
動かしてみよう!!

- ・前進, 後進, 90 度, 180 度旋回



4 基本動作プログラミング実習

- ・超音波センサーとは, 超音波センサを用いたプログラミング実習
- ・色センサとは, 色センサを用いたラインレースプログラミング実習



EV3 システムを使って

出られるロボコンがWRO です。(今日からスタート、来年の世界大会は「タイ」?)

NPO法人 WRO Japan World Robot Olympiad  WRO Japan サイト

NPO法人 WRO Japan WROってなに? WRO Japan 2017 国内大会 WRO 2017 コスタリカ国際大会

2017年大会の情報を随時お知らせしていきます。メディア掲載実績はこちら

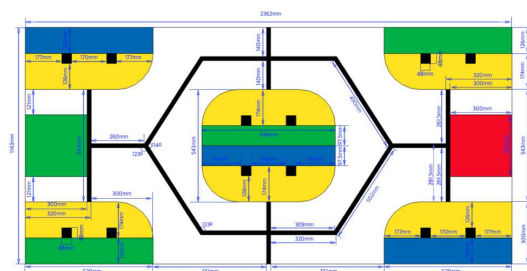


小中高大生 だれでも参加できる
自律型ロボットによる国際的なロボットコンテスト

WRO World Robot Olympiad とは
WROは自律型ロボットによる国際的なロボットコンテストです。
世界中の子どもたちが各々ロボットを製作し、プログラムにより自動制御する技術を競うコンテストで、市販ロボットキットを利用することで、参加しやすく、科学技術を身近に体験できる機会を提供するとともに、国際交流も行われます。

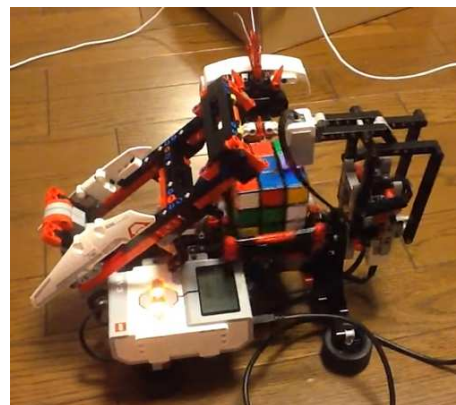
8月地区大会、9月全国大会そして
11月世界大会が開催されます。
競技会は小学生、中学生さらに
高校生とそれぞれの部門毎に開催され
ます。

2017WRO 小学生エキスパート部門 競技コース



競技ロボットばかりではありません

日本中に EV3 でいろいろなロボットを作っている方が大勢います。



EV3 システムを使ったプログラミング

1. プログラムを作る

(1) EV3 ソフトウェアを起動

デスクトップ上のショートカットまたはスタートメニューから、“LEGO MINDSTORMS Education EV3” を起動する。



ショートカット



システム起動画面

(2) ファイル→新規プロジェクト→プログラムで、編集画面を開く



プログラムは PC 側で作ります。

(3) EV3 電源 ON

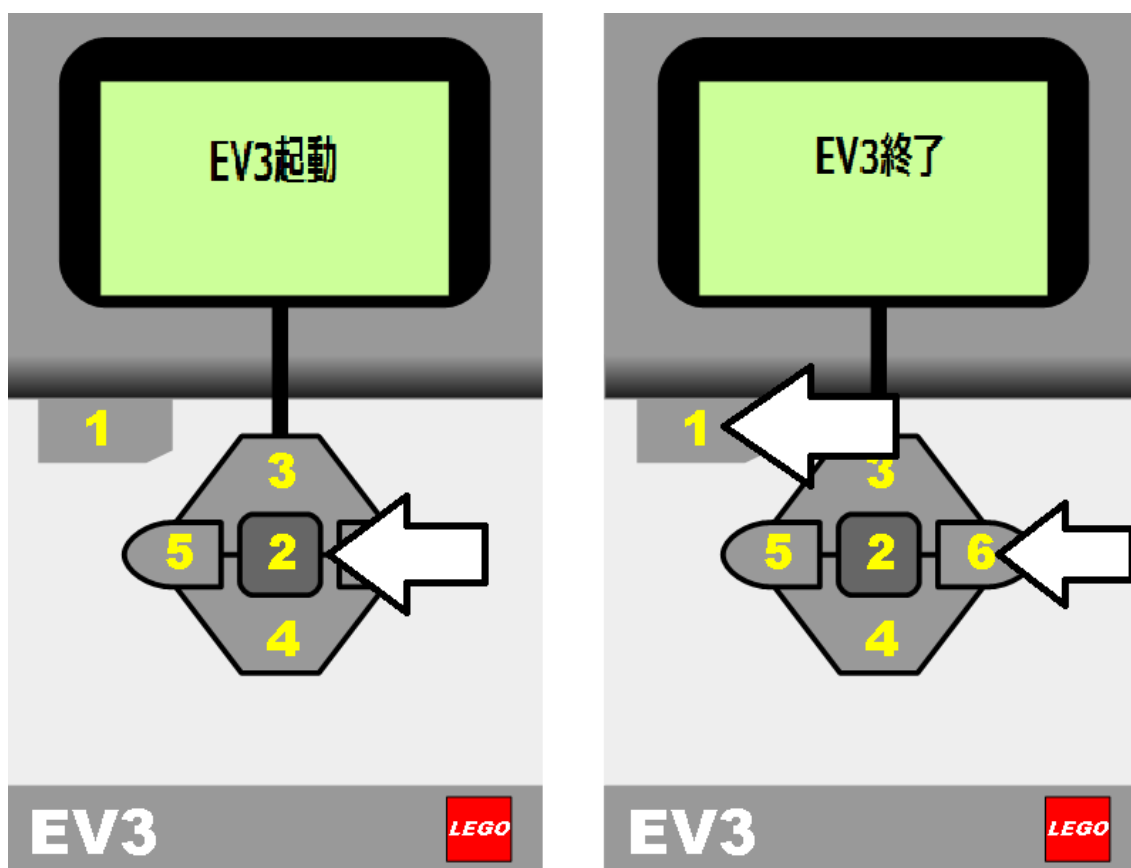
プログラムが完成したら EV3 にプログラムを転送します。(ダウンロードと呼びます)

本体中央ボタンを押し、EV3 起動します。

LED が赤く点灯し、起動画面が表示されます。

赤点灯から緑点灯に変わったら、準備完了です。





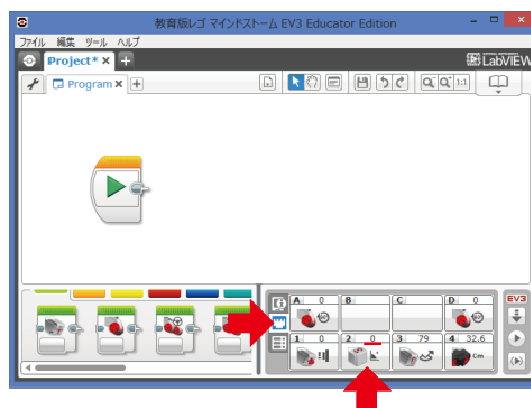
(ここで先に)

EV3 本体の電源を切るには、終了画面が表示されるまで戻るボタン【1】を押します。中断の X が選択されています。右ボタン【6】を使って同意のチェックマークを選択し、中央ボタン【2】を押して OK します。これで EV3 本体の電源が切れます。X の選択中に OK を押した場合、最近使用したファイル画面に戻ります。

(4) EV3 と PC を USB ケーブルでつなぐ

PC で作成したプログラムを EV3 に転送するため、PC とロボットを USB ケーブルでつなぎます。

少し時間がかかる場合もありますが、接続がうまくいくと各種センサの状態をモニタすることができます。これは大変便利な機能です。



(5) プログラムをダウンロードし EV3 で実行

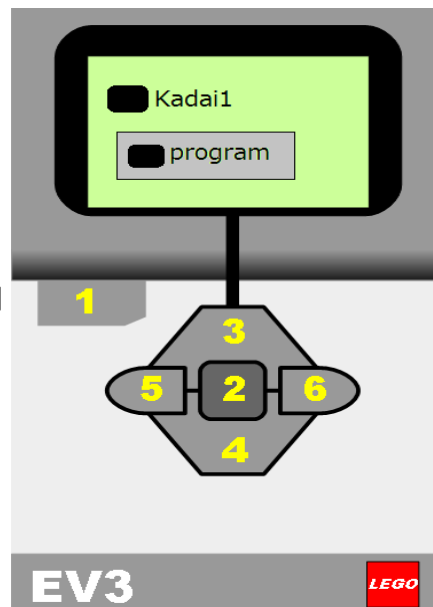
PCで作ったプログラムをロボットに転送(ダウンロード)します。ダウンロードが正常に終了したら(確認音あり)、EV3 のボタン【6】

【4】を操作、

ファイルナビゲーションタブ→「プログラム名」
→「program」を実行します。

EV3 の画面に「MINDSTORMS program」と表示しながらロボットが動きます。

ダウンロードしたプログラムは全て EV3 に残ります。たくさんダウンロードした際は、実行するプログラム名間違わないよう指定して下さい。




2.いよいよプログラミング。

(1) ロボットを前進・後進させる。

簡単に前進させるのは右のコマンドで十分ですが、きちんとロボットを動かすためには、**タイヤの回転数を指定し動かす**ようにします。



タイヤ 1 回転の【1】を【ステアリング】コマンドの、 に設定すると

タイヤ 1 回転すなわち円周分だけ走行します。

タイヤの円周

$$\begin{aligned} &= \text{タイヤ直径} \times \pi \\ &= \phi 56\text{mm} \times 3.14 \\ &= 175.84 \text{ mm} \end{aligned}$$

このプログラム(コマンド)を実行するとロボットは **175.84mm** 前進します。

課題 1 ロボットを 100mm 前進し、その後停止させなさい。

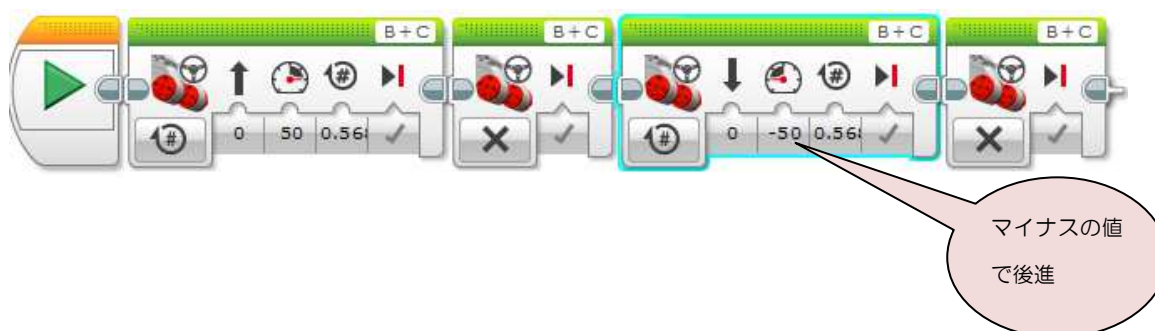
タイヤ 1 回転で 175.84mm 進むので、

100mm 前進させるには $100 \div 175.84 = 0.568$

タイヤを 0.568 回転させれば良いわけです。



課題 2 ロボットを 100mm 後進し、その後停止させなさい。



課題 3 ロボットを前進・後進を繰り返すプログラムを作成しなさい。

プログラムの流れ(順番)を動作確認しながら一緒に作成します。

使うコマンドはこの3つ

ロボットを前進させる

ロボットを停止する

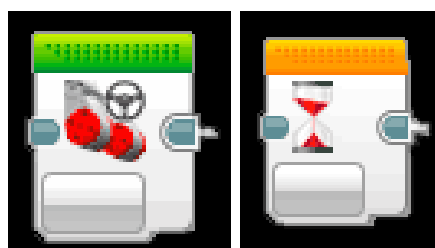
0.5 秒待つ

ロボットを後進させる

ロボットを停止する

0.5 秒待つ

上記を繰り返します。





課題 4 ロボットを 180 度,90 度回転させる。
次の流れに従い、動作確認しながら順番に作成します。

- ロボットを前進させる
- ロボットを右 180 度回転させる
- ロボットを前進させる
- ロボットを左 180 度回転させる
- ロボットを前進させる
- ロボットを右 90 度回転させる
- ロボットを前進させる
- ロボットを左 90 度回転させる
- ロボットを停止する



旋回はタイヤの汚れ、回転スピード、ロボット搭載のボールキャスターの不具合などで誤差が発生します。

マウスでは 4 回同じ動作を繰り返してもピッタリいくパラメータを設定しています。各パラメータを入力し、動作確認しそれぞれのロボットの最適値を設定して下さい。

左 180 度 信地旋回

ロボット左旋回
 パワー:50
 走行距離:タイヤ0.95回転
 (おおよそ180度旋回)



右 180 度 信地旋回

ロボット右旋回
 パワー:50
 走行距離:タイヤ0.95回転
 (おおよそ180度旋回)



左 90 度 信地旋回

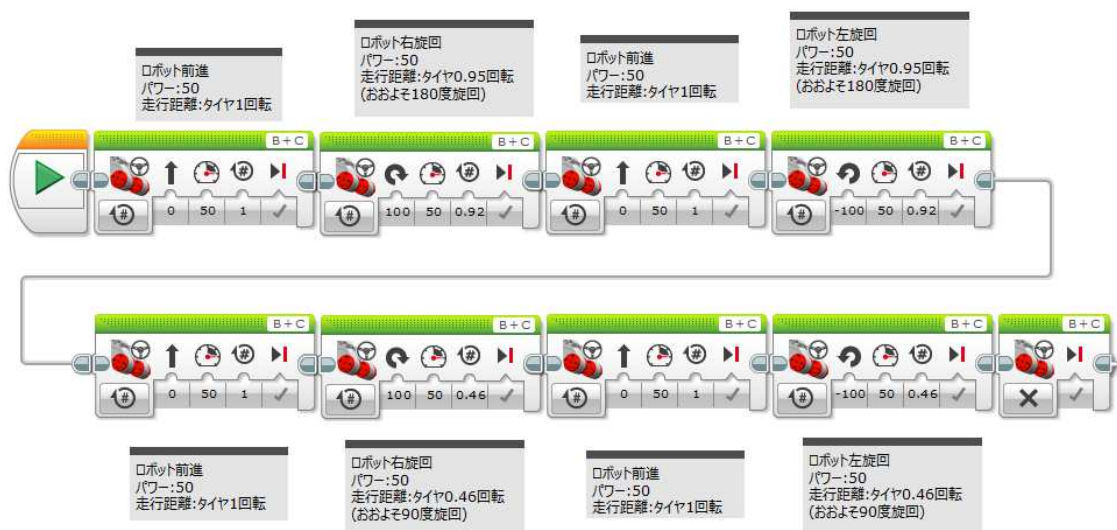


ロボット左旋回
 パワー:50
 走行距離:タイヤ0.46回転
 (おおよそ90度旋回)

右 90 度 信地旋回

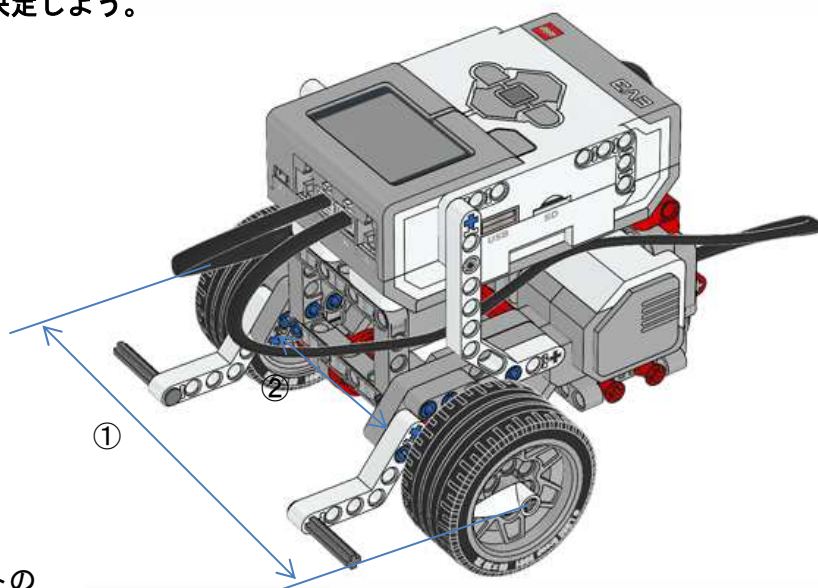


ロボット右旋回
 パワー:50
 走行距離:タイヤ0.46回転
 (おおよそ90度旋回)



実習 ここまで学んだことを組み合わせ、自分の思う通りにロボットを動かしてみよう。(各自プログラム実習)

各自のロボットを実測し
 各自のパラメータを決定しよう。

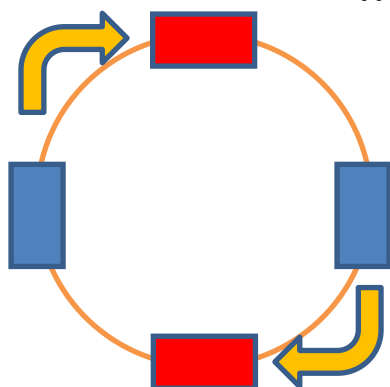


トレーニングロボットの

- ①左右タイヤの外側間隔
- ②左右タイヤの内側間隔
- ∴③左右タイヤの中心間隔

青のタイヤが赤の位置まで走行すれば 90 度回転、
 青のタイヤが逆側の青の位置まで走行すれば 180 度回転することができる。

90 度, 180 度回転するためには何 mm 走れば良いか



90 度回転は $③ \times \pi / 4$

180 度回転は $③ \times \pi / 2$

で求めるところができる。

理論値を計算しよう。

①147.5mm ②91.6mm ∴③119.5mm \approx 120mm

90 度回転 $120 \times 3.14 / 4 = 94.2\text{mm}$

180 度回転 $120 \times 3.14 / 2 = 188.4\text{mm}$

タイヤの回転数はいくつになるか?

90 度回転 $94.2 / 175.84 = 0.535$

180 度回転 $188.4 / 175.84 = 1.07$

実際にはターンの速度, ボールキャスターの問題, タイヤの汚れなどによって誤差が出る。
 カット&トライし、最適な回転数を決めて下さい。(ロボット固有)

3.超音波センサとは

ロボット前方に手をかざし近づけたり遠ざけたりして下さい。

手までの距離が「ハードウェアページ」に表示されることが確認できます。

超音波センサを利用すると「距離計測」ができます。

課題 11 超音波センサを使う。

ロボットを障害物 10cm手前で旋回する。

ロボットを前進させる

超音波センサで距離を計測する

障害物との距離が 10cm より小さくなったら停止する

左 90 度旋回する

ロボットを前進させる 上記を繰り返します。



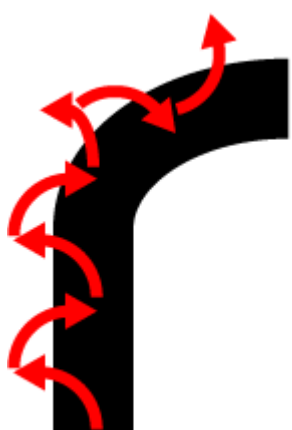
実習 超音波センサの能力を把握しよう。

次の実験をして下さい。(大会出場を考えているなら必ず実習すること)

- (1) 障害物の色により停止位置に変化あるか。
- (2) 障害物に斜めに近づいた場合、停止位置に変化はあるか。
- (3) 障害物が円柱の場合、停止位置に変化はあるか。

4. ラインレースする

ロボコンの基本中の基本はラインレース（ラインに沿って走行）です。
既に無人搬送車として実用化されています。まずは最も安全走行を学習します。



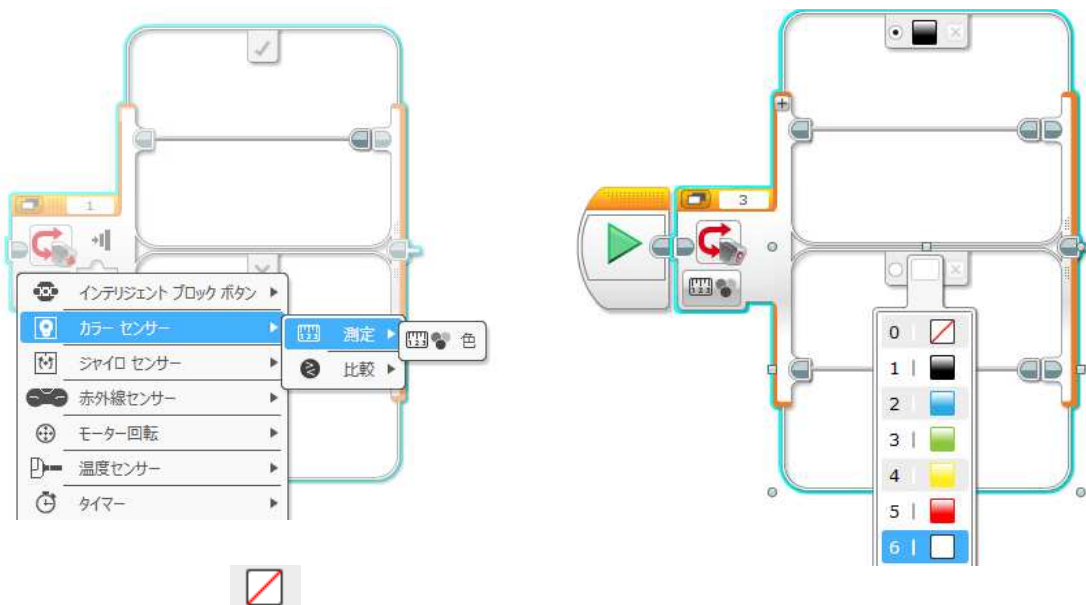
床、ラインテープの色は競技によってそれぞれです。
多くは白地の床に黒テープ、あるいは白地の床に黒
テープが一般的です。最近では他の色のテープの場合も
ある様です。

ロボットが床に張ったテープに沿って走行するには
左図のように右ターンと左ターンを繰り返し走行す
るのが基本です。

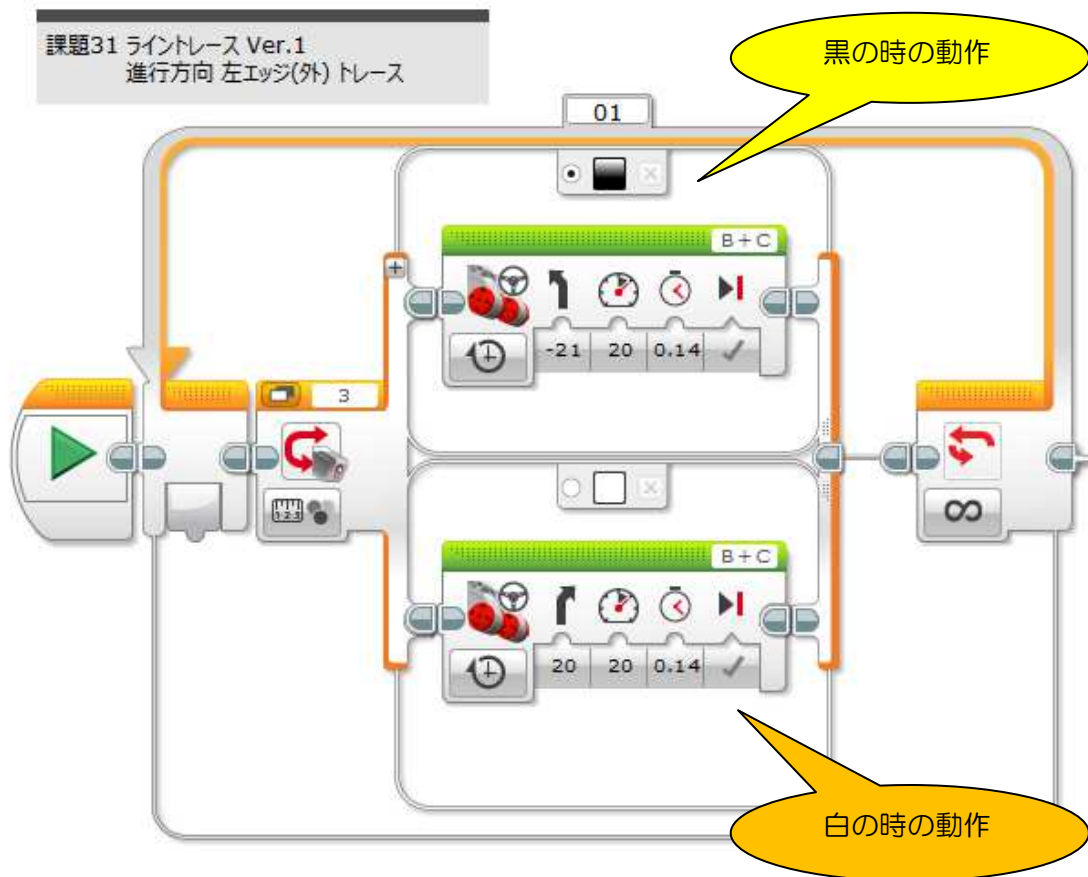
今回は白地に黒テープでプログラミングします。

簡易ラインレース方法

オレンジ色のタブより【スイッチ】を選び貼り付けます。
【カラーセンサ】【測定】【色】を選択します。



を【白】に変更します。（この印は透明を意味します。）
その後【黒】【白】それぞれに【ステアリング】を貼り付け、各パラメータを変
更します。今回はゆっくりですが確実にトレースするプログラムにします。



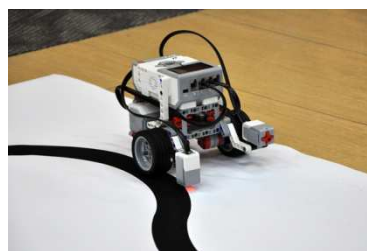
さっそくライントレースしてみましょう。

実習 各自床にテープを貼り、自由に走行してみよう。

光センサの赤スポットが進行方向ラインの左エッジの上に来るようにロボットを置き、プログラムをスタートします。

センサが白(床)にある場合は右旋回、センサが黒(ライン上)の場合は左旋回しながら走行します。

“じくざく走行” となりますが、これがライントレースの基礎・基本です。



※このプログラムでは低速(ゆっくり)走行しかできません。

速度を上げるとコースアウトしてしまいます。

また右カーブは大丈夫ですが、左カーブはコースアウトしてしまいます。

どうすれば高速にかつ安全走行ができるか、各自考えプログラムを修正してみよう。改善の余地はたくさんあります。

5 ライントレース応用

大会では距離を指定してライントレースしたり、超音波センサーで障害物が見つかるまで走行するなどのことがよくあります。次の課題を実習下さい。

課題 36 距離を指定してライントレースする方法

左タイヤが 6.5 回転するまでライントレース下さい。

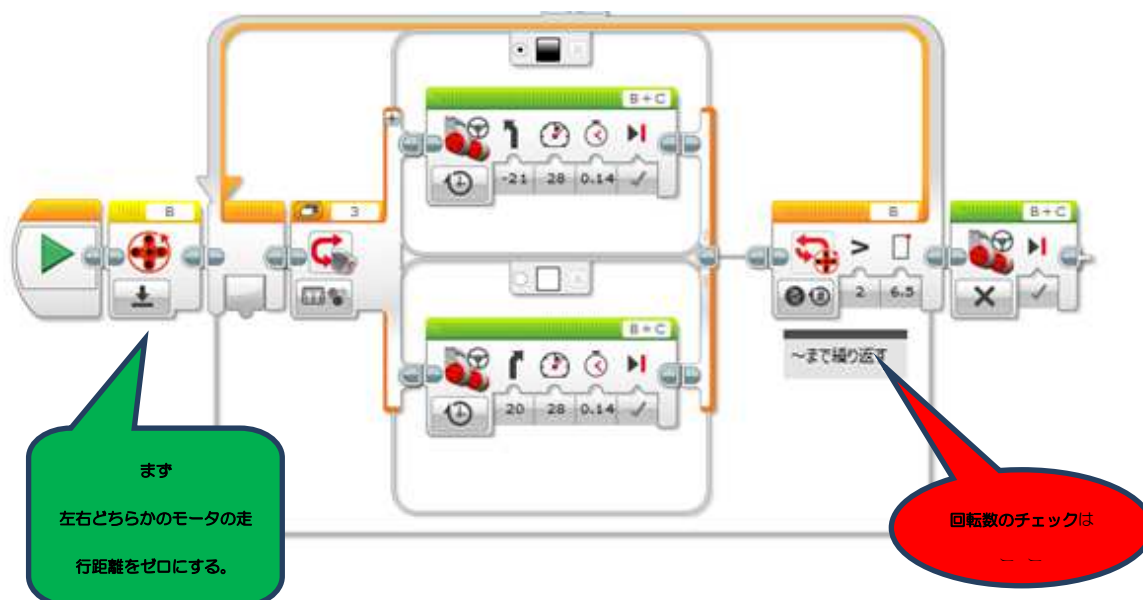
タイヤ 6.5 回転とは？

タイヤ 1 回転で 175.84mm 前進するので、6.5 回転するとロボットは

$$175.84 \times 6.5 \text{ 回転} \div 1143\text{mm}$$

進むみます。(前進する。)

このように走行距離を設定しライントレースすることができます。



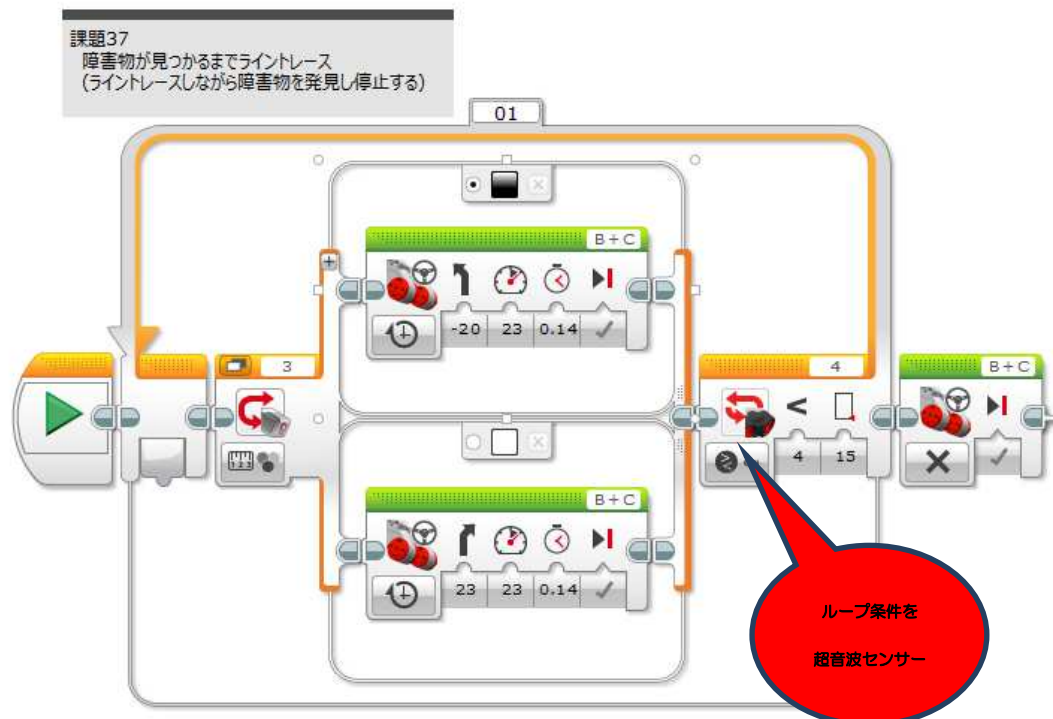
では 1m(100cm,1000mm)ライントレースするには？

$$1000 \div 175.84 = 5.686 \text{ 回転}$$

このプログラムの 6.5 を 5.6 に修正すると約 1m ライントレース走行となります。

課題 37 障害物が見つかるまでライントレース

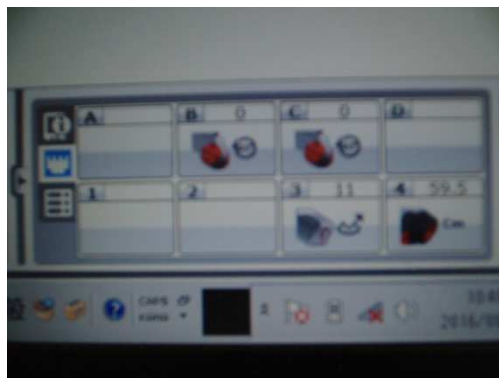
ライントレース中に障害物を発見したらロボットを停止するプログラムを作りなさい。



5.ラインレースする(2)

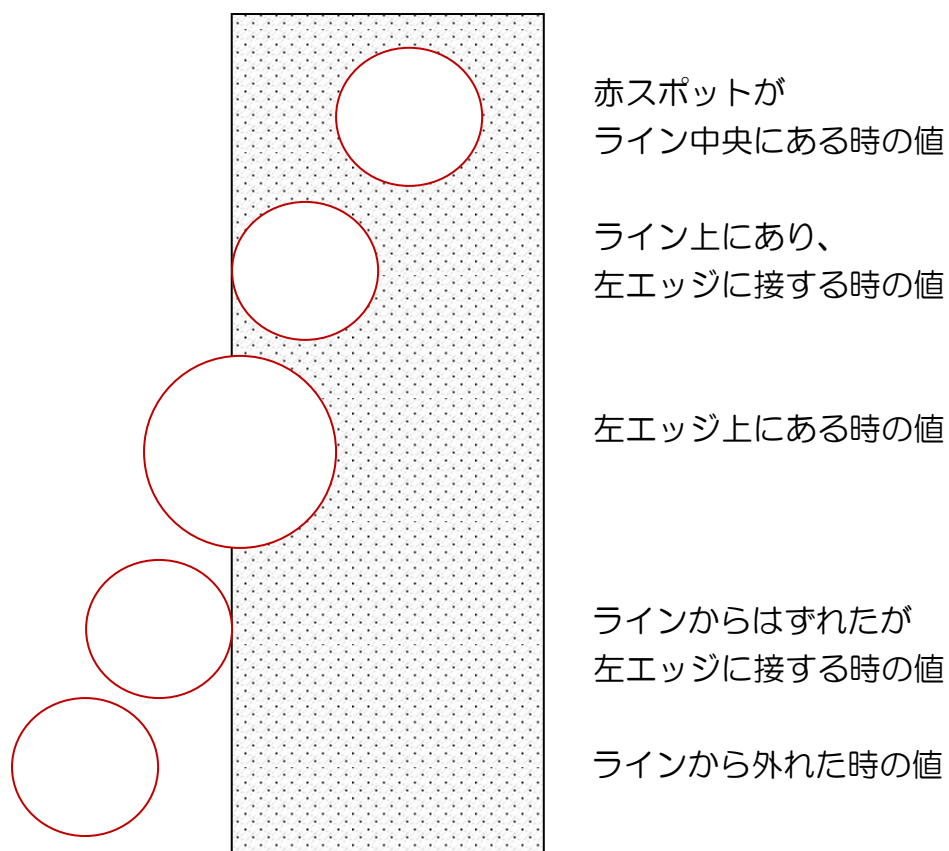
(1) 光(カラー)センサ値の計測

黒テープの真上に光センサの赤スポットが来るようにロボットを置きます。黒テープ上でロボットを手で移動し、光センサの値が変わることを確認します。光センサの値により「ロボットの位置」を知ることができることを理解できましたか。



各自光センサの値を画面で確認し、○の中にその値を記入してみよう。

これは重要な作業です。



(2) プログラミングする

ロボットを前進させる



光センサでラインからの反射光の強さを計測する

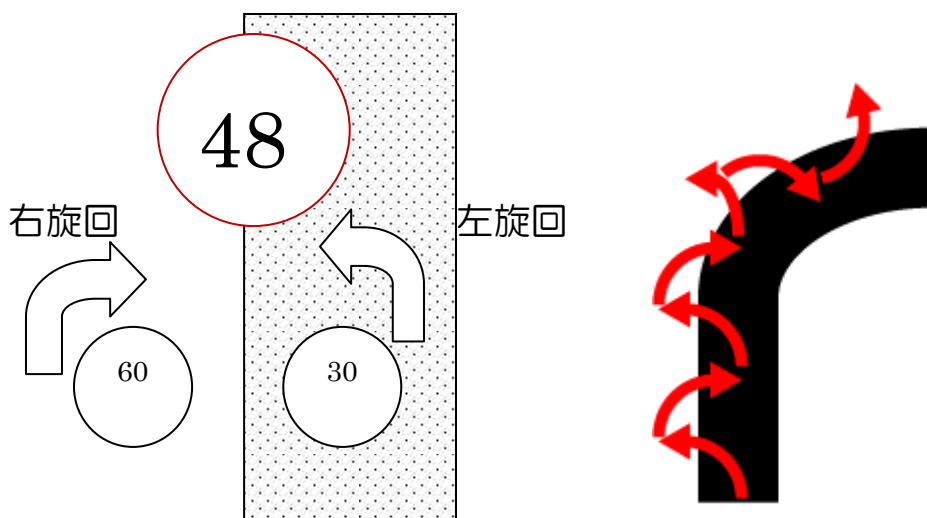
反射光の強さ 48 より大きいならば、左旋回する

反射光の強さ 48 以下ならば右旋回する

上記動作を繰り返す



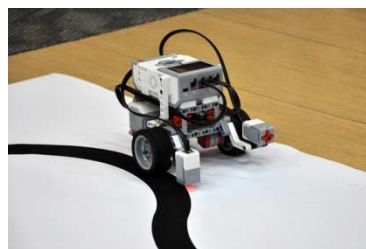
注意 48 は各自測定した値を入力する



光センサの赤スポットが進行方向ラインの左エッジの上にくるようにロボットを置き、プログラムをスタートします。

センサの値が 48 を越した場合、すなわちロボットがラインから外に外れた場合は右旋回、センサ値が 48 以下の場合、すなわちロボットライン上の場合は左旋回しながら走行します。

(4. ライントレースで紹介した簡易走行と同じです。)



(3) 滑らかなライントレースに挑戦(考えよう)

センサ値	処理
8	左折 強
18	左折 中
28	左折 弱
38	直進
48	直進
58	右折 弱
68	右折 中
78	右折 強
88	

まずロボットの横ずれの確認から

すぐにできる改善方法として「センサ場合分けを増やす」方法があります。センサの状況に応じて旋回の強さ変化させる方法です。別の方法として少し勉強しなければなりませんが、ロボコンの世界で良く用いられる PID 制御によるトレースがあります。

次にロボット縦のずれの確認

1 つ前のセンサ値と今のセンサ値を比較することで、
どちらの方向にどれだけずれたのかを認識することができます。

$$\boxed{\text{1 つ前のデータ}} - \boxed{\text{今のデータ}} > 0 \quad \text{右にずれた}$$
$$= 0 \quad \text{状態は変わらない}$$

$$< 0 \quad \text{左にずれた}$$

値(絶対値)が大きいほど急激な変化(ずれ)が生じたと認識できます。

例) 1 つ前のデータ 84(大きくコースアウトしていた。)

今のデータ 48(左エッジ上にある)

$$\boxed{\text{1 つ前のデータ}} - \boxed{\text{今のデータ}} = 84 - 48 = 36$$

ロボットは右方向に 36 ずれた

例) 1 つ前のデータ 9(ライン中央にいた)

今のデータ 84(大きくコースアウトした)

$$\boxed{\text{1 つ前のデータ}} - \boxed{\text{今のデータ}} = 9 - 84 = -75$$

ロボットは左方向に 75 ずれた

ここまで理解できましたか。

今回作成したラントレースプログラムの考えとは明らかに違いますね。

次に PID 制御とは

ロボットがラインの左エッジにいるとき(データ 48)がベストの位置、
すなわち目標値とすることができます。

現在の値と目標値を比較すると現在のずれを計算できます。

$$\text{現在値} - \text{目標値} = \text{現在のずれ}$$

ずれが大きければターンの量大きく、ずれが小さければターンの量小さく
することができます。(≡細かい場合分け)

これを P 制御と言います。

$$\text{P 制御} \quad (\text{現在値} - \text{目標値}) \times K_p \quad K_p \text{ は比例係数}$$

次にずれに着目します。

今回のずれと前回のずれを比較することで、ずれの大きさを計算できます。

$$\text{今回のずれ} - \text{前回のずれ} = \text{ずれの変化}$$

ずれの変化が大きければターンの量大きく、ずれの変化が小さければ
ターンの量小さくすることができます。

これが D 制御です。

$$\text{D 制御} \quad (\text{今回のずれ} - \text{前回のずれ}) \times K_d \quad K_d \text{ は微分係数}$$

最後に i 制御ですが、これは実際にはプログラム化せず、通常 PD 制御で走
行しています。今回は簡単な説明にとどめます。

$$(\text{今回の残ったずれ} - \text{前回の残ったずれ}) \times K_i \quad K_i \text{ は積分係数}$$

この求められた制御量を基にターン量を決めるのが PiD 制御です。

ターン量=P 制御 + D 制御 + I 制御

このようにターン量をコースの状況から判断するのが PID 制御です。

最も苦勞するのは各係数を決めることです。

そのためには地道なカット&トライが必要となります。

6.知っておくと便利なコマンド

複雑なプログラムを作成するのに必要となります。

この教室が終わったら各自勉強して下さい。



表示(画面出力)

EV3 の画面にセンサの値等を表示することができます。

サンプルプログラム kadai15.ev3

「超音波センサの値を画面に出力する。」



変数設定

EV3 の中にデータを格納する”箱”を確保し、データの値を保存(記憶)しておくことができます。プログラム作成にはほとんどの場合使用します。



定数設定

決まった値を変数と同様に保存しておくことができます。



算術演算, 論理演算

プログラムで電卓と同じように計算することができます。



ロジックによるループ終了

変数の値によってループを終了したい場合使用します。

(使い方は少し面倒ですが、シングルタスクで開発するには覚えておくと便利です。)



データロギング

センサの値やモータの回転数などをまとめて EV3 の中に保存しておくことができます。保存後はPCにアップロードして使用します。

ロボットの細かい状況などを解析することができます。

サンプルプログラム kadai19.ev3

「超音波センサのロギング」

7. ミニ競技会



ここまで紹介したプログラムを参考に「障害物回避ラインレース」するプログラムを作ってみましょう。内容は「ライン上に置かれたペットボトルにぶつからないようコースを周回する」です。目指せ完走!!

8. 最後に

WRO 世界大会出場を目指すために

山形県大会（宮城県大会）、全国大会そして世界大会とつながる WRO 大会は小中高一般とカテゴリが分かれます。カテゴリ毎の勝負です。小中部門は光センサによるラインレース、超音波による対物認識が必須。~~さらにジャイロセンサによる角度検出もできれば安心して大会参加できるでしょう。~~

ラインレースは滑らか走行がタイム短縮につながりますが、小中学生の皆さんに PID 制御はちょっと難しいかもしれません。PID 制御に固執するのではなく、左エッジ走行と右エッジ走行の切り替えや色センサをもう一つ購入し、色センサ 2 個を用いてレースする参加者が多いようです。各自ちょっとした工夫を考えてみて下さい。

今年度の競技規則を見ても、「距離を指定しての走行，正確な 90 度,180 度旋回」が最も大切のようです。今回学習したことをよく復習し来年の全国大会，世界大会出場を目指し、これから継続し頑張ってください。今回の教室がロボットエンジニアへの第一歩になれば幸いです。

やまがたロボットクラブ

代表 齋藤薫(山形県立霞城学園高等学校勤務)