

ライトレーサー

これから作成するキットは、決められたコースを自動的に走行するライトレーサーという車です。いくつかの部品をハンダ付けして組み立てた後、コースを判断して走行するためにプログラムを作成します。プログラムを実行するためのコンピューターには micro:bit を使用します。車に取り付けられたセンサーがコースを読み取り、micro:bit がプログラムにしたがってモーターを回転させることで自動走行ができます。



micro:bit ってなに？

micro:bit (マイクロビット) とはイギリスの BBC (英国放送協会) が主体となって作った教育向けマイコンボードです。手のひらに乗る小さなコンピューターですが、小さくても LED やボタン、センサーなどいろいろな機能が搭載されています。パソコンやタブレット、さまざまな環境でプログラミングができます。また、拡張パーツを追加することでいろいろな機能を追加できます。本キットも micro:bit にセンサーとモーターを追加することで動作します。



開発環境 (ブラウザ)

<https://makecode.microbit.org/>



使い方

<https://learn.switch-education.com/microbit-tutorial/>

micro:bitをはじめよう

- 開発環境
- 初めてのプログラミング (PC版)
- 初めてのプログラミング (Android版)
- 初めてのプログラミング (iOS版)
- プログラムの書き込み方法
- プログラムの共有方法
- LEDの点滅
- ボタンスイッチ
- タッチセンサー
- 加速度センサー
- コンパス

micro:bit

できること：micro:bitには何がついているのかが分かります

micro:bitの表

ロゴマークが描かれているほうが表です。※画像のmicro:bitのバージョンは2です。

ボタン

LED (明るさセンサー)

タッチ検出 (ロゴマーク)

ボタン

端子

電子部品の確認

電子部品には極性があるものとないものがあります。極性があるものは取り付ける向きに気を付ける必要があります。発光ダイオード（LED）は代表的な極性あり部品で、取り付ける向きを間違えると光りませんので注意しましょう。

極性なし



抵抗器 半固定抵抗器 モーター スイッチ

極性あり

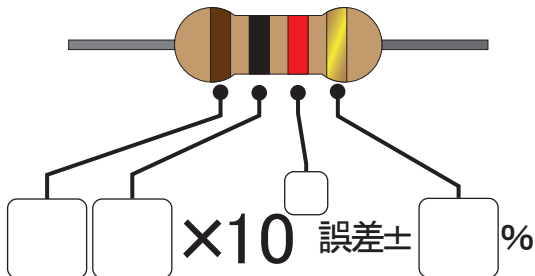


LED フォトリフレクター 電池ボックス

抵抗値の読み方

抵抗器は電流を流れにくくするための部品です。抵抗値が大きいほど電流は流れにくくなります。抵抗値はカラーコードという色の帯で記されています。また、同じ抵抗値の抵抗器でも若干の誤差（バラつき）があります。どの程度の誤差に収まるかも帯の色で記されています。本キットには3種類の抵抗値を持った抵抗器が入っていますので、必ず色を確認して間違えないように気をつけましょう。

黒	0
茶	1
赤	2
橙	3
黄	4
緑	5
青	6
紫	7
灰	8
白	9



金の帯は誤差±5%

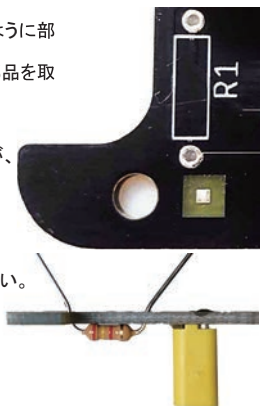
例

$56 \times 10^2 = 5600 \Omega = 5.6k \Omega$

ハンダ付け

部品の足を基板の穴に挿入します。基板には「R1」のように部品番号が記されています。次のページ以降登場する順に部品を取り付けます。

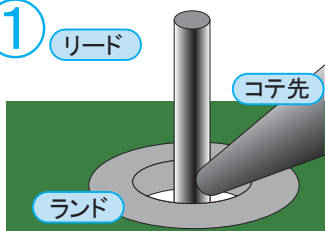
シャーシ基板にはあらかじめ黄色い部品がついていますが、この部品があるため他の部品の固定が少し難しくなっています。部品の足を基板の穴に通した後少し足を曲げる等対策をして部品が落下しないようにしてハンダ付けしてください。



ハンダ付け

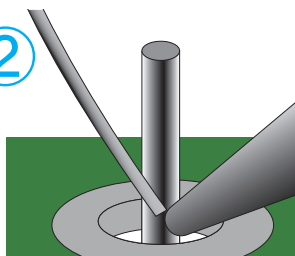
①

リード



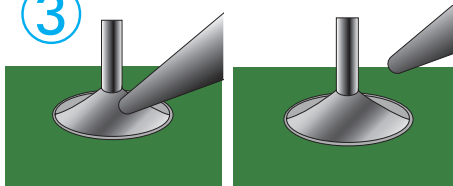
リード（部品の足）とランド（ドーナツ状の部分）にコテ先を3秒程度あててしっかりと熱を伝えます。

②



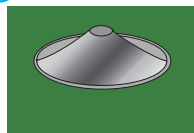
温められた部分に糸ハンダを送りハンダを溶かします。

③



2秒程度ハンダを温め続けハンダをなじませたら、コテ先をはずします。ハンダが富士山のような形になるのが理想的です。

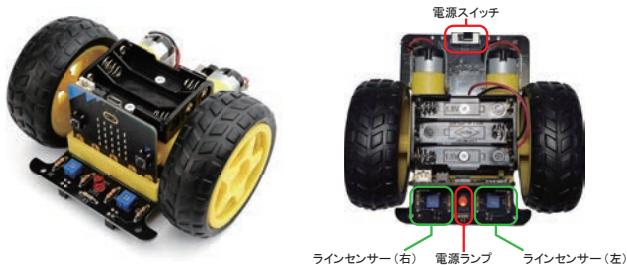
④



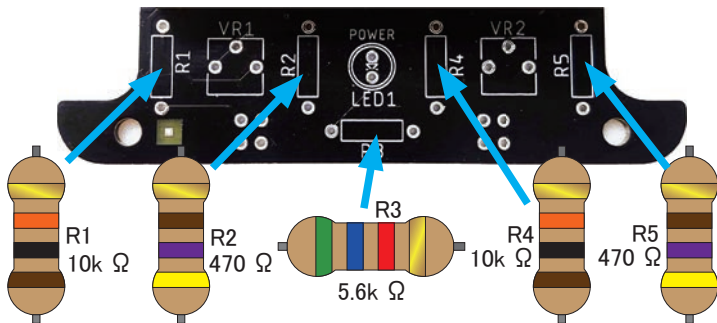
余分なリードをニッパーでカットします。

組立

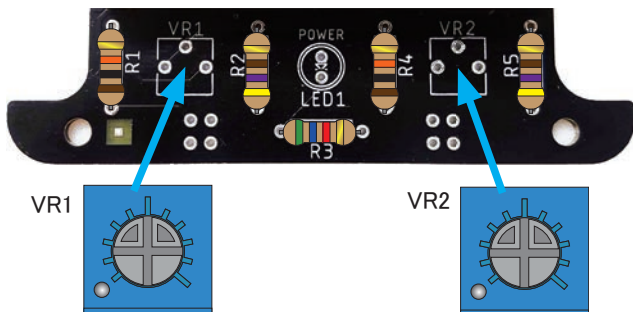
完成図



抵抗器のハンダ付け



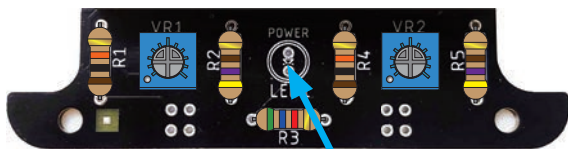
半固定抵抗器のハンダ付け



LEDのハンダ付け



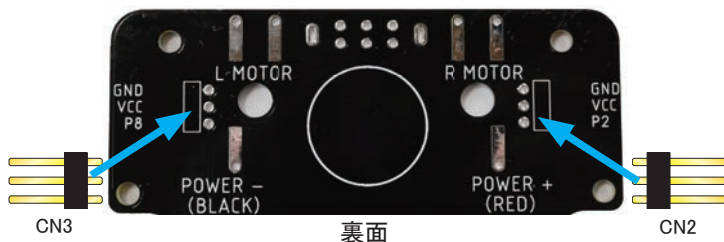
極性注意



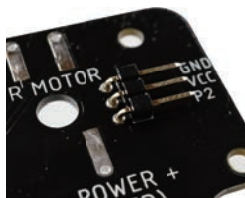
LEDの足の長さをよく見て、上図のように向きを合わせます。

ピンヘッダーのハンダ付け

この部品はオプションパーツ（詳しくは p.24）を接続するための部品です。
そのため、オプションパーツを使用しない場合はつける必要はありません。



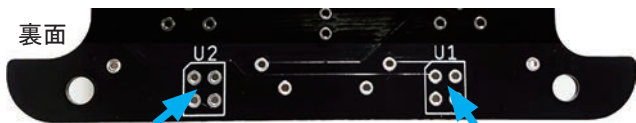
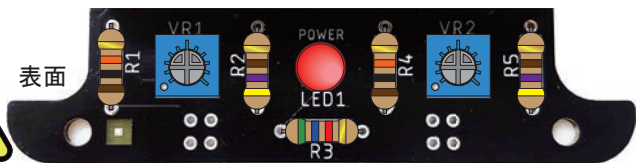
この部品は基板の表裏どちらに付けても構いません。
裏面に取付けた方が部品の固定が容易で、ハンダ付けが簡単ですのでおすすめです。



フォトリフレクターのハンダ付け



極性注意



U2

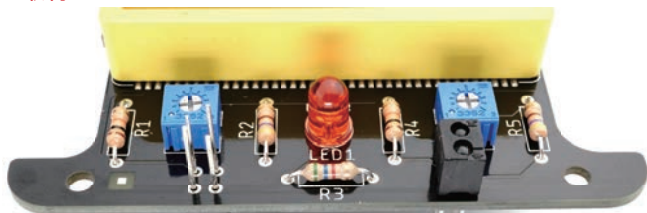
基板の裏面につけます。
取り付ける向きを図に
あわせます。

U1

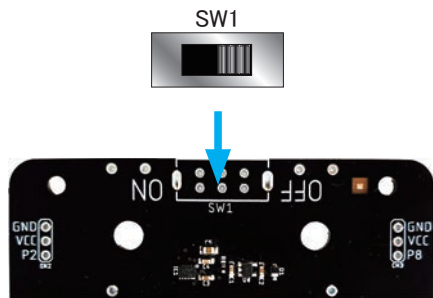


取付向き注意

裏面に取付

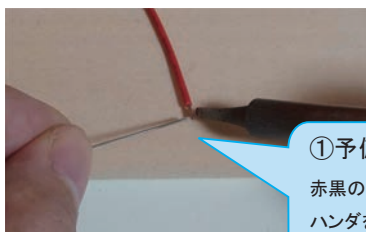


スイッチのハンダ付け



8ヶ所すべてハンダ付け

配線材のハンダ付け



①予備ハンダ

赤黒の配線材 4本の両端（被覆がむけた所）にハンダを溶かして導線にハンダを吸い込ませます。



②モーターの端子にハンダを溶かしてつけます。

この時、モーターを固定している透明の部品を溶かさないように気を付けましょう。

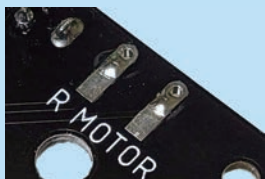
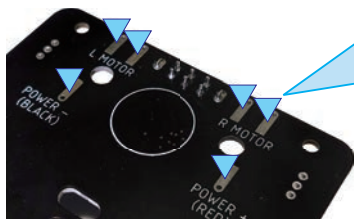


③ 予備半田をした配線材をモーターの端子に当てた状態で再びハンダゴテをあててハンダを溶かしモーターの端子と配線材を接合します。配線材の色は図に合わせてください。(逆にしてもプログラムで修正可)



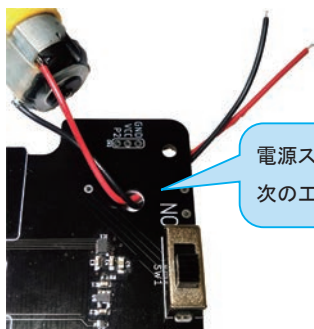
④ タイヤをモーターに取り付けます。取り付け向きも図に合わせてください。

モーターのハンダ付け



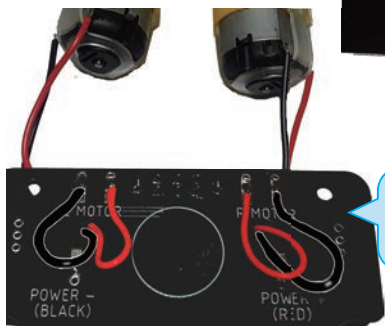
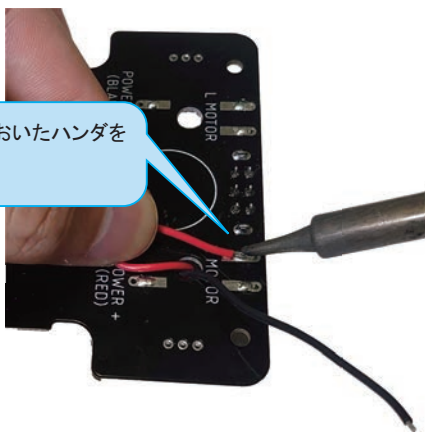
6 か所の▼部分にハンダを少量流しておきます。

指定した箇所にハンダ付けできていれば大丈夫ですので、必ずしも下の図の通りに作らなくてもかまいません。ハンダ付けしやすいように配線材のルートを確認してください。断線した場合などに修理ができるよう、配線材は少し長めになっています。



電源スイッチ横の穴に配線材を通します。
次の工程が難しい場合は穴に通さなくても大丈夫です。

シャーシ基板を裏返し、先につけておいたハンダを溶かして配線材をとりつけます。

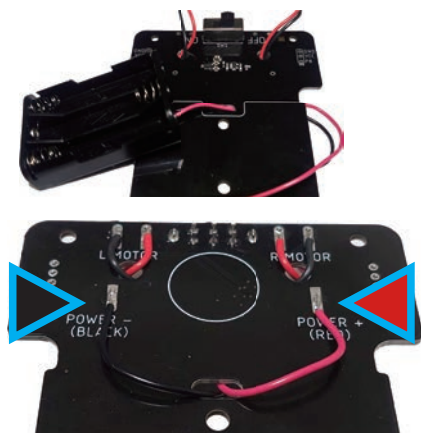


同様に反対側も取り付けます。
外側が黒、内側が赤になるようにします。

電池ボックスのハンダ付け



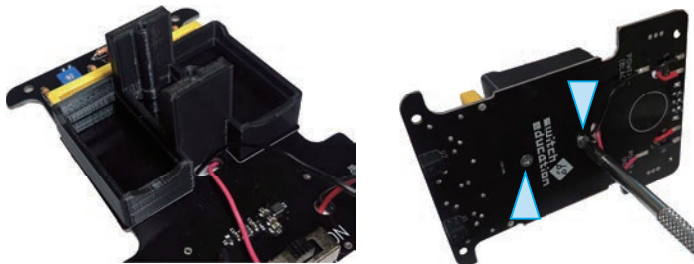
極性注意



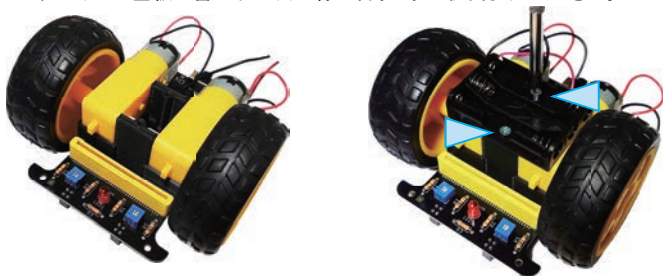
シャーシ基板の長穴から電池ボックスの配線を通し、裏面でハンダ付けします。
赤と黒の配線は必ずシャーシ基板に書いてある通りにつなげてください。

間違えると故障の原因になります。

モーターの固定



モーター台をシャーシ基板にねじ止めします。モーター台には向きがありますので、シャーシ基板に書かれた白い線に合うように取り付けてください。



モーター台にモーターをセットして上から電池ボックスを重ねてねじ止めします。

キャスターの取り付け



キャスターについている両面テープを使ってシャーシ基板裏面の白丸の位置にキャスターを取り付けます。

完成！



最終チェック

- 全ての部品がハンダ付けされているか。
- ハンダがあふれて本来つながるべきでない部分と接触していないか。
- 電池ボックスの赤・黒の線は正しくハンダ付けされているか。p.11
- モーター、電池ボックスの配線は外れていないか。
- スイッチを OFF 側にしてあるか。

チェックが終わったら単 4 アルカリ電池 3 本をセットしましょう。

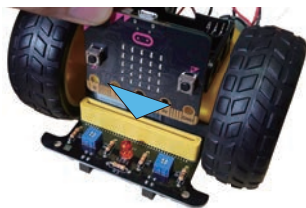
電池は必ず正しい向きに入れてください。逆向きに入れると故障の原因になります。



電源スイッチを ON にして LED が光ればひとまず OK です。

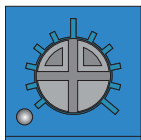
光らない場合、主に次のような原因が考えられます。

- LED の向きが逆。
- LED もしくは R3 の抵抗が正しくハンダ付けされていない。
- 電池ボックスの配線が切れてしまった。ハンダ付けに失敗している。
- 電池ボックスの配線が赤黒逆。(故障の恐れあり)
- 本来つながってはいけない箇所が接触している。(ショート)
- LED の破損・初期不良。



問題がなければ micro:bit をシャーシ
基板の黄色いソケットに差し込みましょう。

プログラミング前に



【半固定抵抗器】

この部品はつまみをドライバーで回転させることで抵抗値が変化する部品です。抵抗値を変えることでセンサーの感度を変えられます。初期状態では図のようにつまみを中央にセットしておきます。うまくコースの白黒を判定できないとき、この部品を調整することで白黒の判断基準を変える事ができます。

使い方は p.18 を参照してください。

【LEDの点灯】

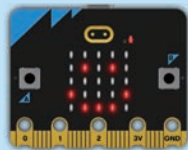


micro:bit を PC と USB ケーブルで接続した場合、電源スイッチを ON にしていなくても LED が光ります。この状態でも micro:bit とセンサーは動作しますが、モーターは動きません。モーターを動かすためにはシャーシ基板の電源スイッチを ON にする必要があります。

動作の仕組み

制御・演算・記憶

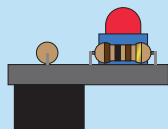
センサーから入力された情報をプログラム通りに処理して出力します。micro:bit 下部の金色の部分が入出力端子です。0、1、2というのが端子番号です。その間にある細い端子にも番号が割り当てられています。P0（0番の端子）とP1にはそれぞれフォトリフレクターが接続されます。P13、P14は右のモーター、P15、P16は左のモーターに接続されます。



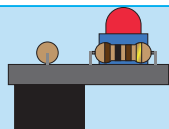
入力



フォトリフレクターは光がどの程度反射して戻ってくるかを調べるセンサーです。白丸の部分から目には見えない赤外線的光を出しています。黒丸の部分は光を受け取るセンサーで受け取った光の量を電圧に変換します。



白は光を反射する



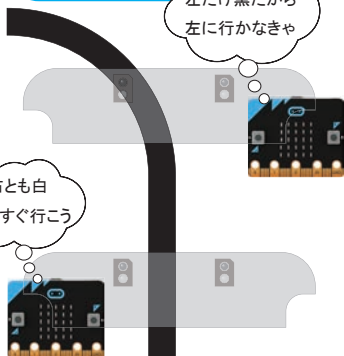
黒は光を吸収する

出力

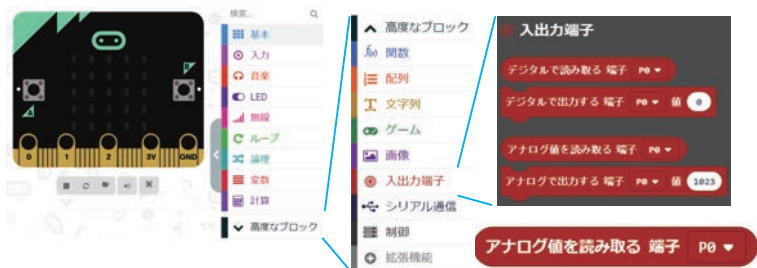
モーターを回転させるためには大きな電流が必要ですが、micro:bit の出力端子は大きな電流を流すことができないため、micro:bit とモーターの間にモータードライバーという部品を中継して micro:bit の命令でモーターを回しています。

左だけ黒だから
左に行かなきゃ

左右とも白
まっすぐ行こう



センサーの確認



使用ブロック： 高度なブロック → 入出力端子 → アナログ値を読み取る端子

センサーの値を取得するためには【アナログ値を読み取る 端子P0】ブロックを使います。P0が右センサー、P1が左センサーの値です。センサーの値は白い時に小さく、黒い時に大きくなります。実際にセンサーの出力を数値で確認してみましょう。

使用ブロック： 基本 → 数を表示

数を表示 0

micro:bit のLED 画面に数字を表示するためには【数を表示 0】ブロックを使います。このブロックの数字部分に【アナログ値を読み取る 端子P0】を入れることで、センサーの値を表示することができます。白と黒の紙を用意して実際にセンサーの値がどう変化するか確認してみましょう。数値は0～1023の範囲で変化します。【数を表示 0】ブロックは同時に複数の桁の数字を表示できませんので、スクロールして複数の桁が表示されます。

色だけでなく材質によっても数値が変わります。またセンサーとの距離によっても数値が変化しますので、紙がたわんだりしないようにしましょう。

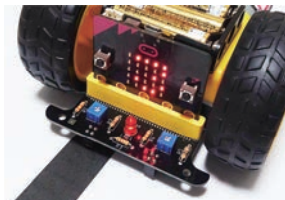
プログラム STEP1

ずっと

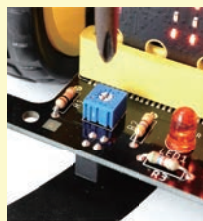
数を表示

アナログ値を読み取る 端子 P0

P0が右側のセンサー（micro:bit のAボタン側）
 P1が左側のセンサー（micro:bit のBボタン側）
 P0をP1に変えて同じように白と黒が変化するか確認しましょう。また、数値がスクロールしている間は色を変えても反映されませんので、次のスクロールを待ちましょう。



実際は黒色でも全く光が反射しないわけではなく、白に比べて少量ですが反射します。環境によっては黒の時と白の時のセンサーの値に大きな違いが出ない場合もあります。その場合半固定抵抗を回してセンサーの感度を調整しましょう。例えば黒の反射量が多い時、時計回りに回すと感度が低下し、より多くの反射がないと反応しなくなり、白と黒の区別がつきやすくなります。



しきい値の設定

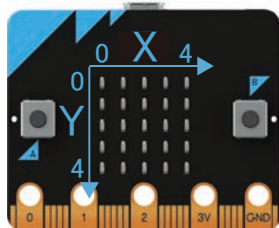
白と黒でセンサーの値が変化の様子は確認できました。この数値からセンサーの下の色を判断します。この数値より下だったら白、上だったら黒とみなすというように、判断のしきいになる値を設定しましょう。

例：白が 100 黒が 900 だった場合、この間のどこかにしきい値を設定します。この後実際に試してみても変更できますので、とりあえずは 500 くらいにします。

使用ブロック： LED → 点灯 x 0 y 0 LED → 消灯 x 0 y 0



micro:bit の LED を個別に制御するためのブロックです。X、Y の座標で制御する LED を指定します。ここではセンサーの状態によって LED をつけたり消したりします。



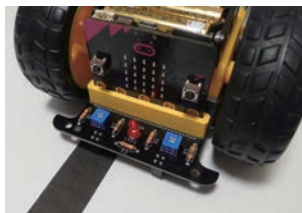
使用ブロック： 論理 → 0<0

論理 → もし真なら でなければ

条件に応じてプログラムを分岐させるためのブロックです。今回はセンサーの値が黒の時と白の時で処理を変えるために使います。



まずは右のセンサー（P0）に注目したプログラムをつくります。ブロックを下図のように組み合わせて数値を変更してください。下図のプログラムでは P0 のセンサーのアナログ値が 500 より小さかった場合（白だったら）X0,Y2 の LED を点灯させ、そうでなければ（黒だったら）X0,Y2 の LED を消灯します。



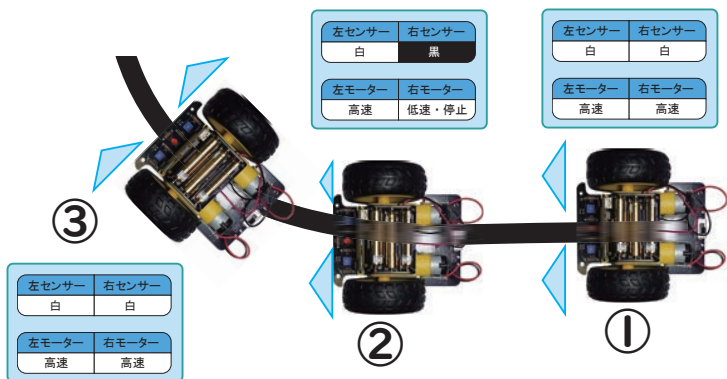
黒だと消灯



白だと点灯

これで右側のセンサーの動きが確認できましたので、左側も作ってみましょう。
答えは22ページにあります。

ライトレースの仕組みを考える



上図のような場合について考えます。まず①の状態ではコースは直線で左右のセンサーは両方とも白になっています。この場合は左右のモーターの速さを合わせて直進します。しばらく直進すると右カーブにさしかかります。この状態②の時、右のセンサーが黒の状態になります。その場合車を右の方に向けないとコースからそれてしまいますので、左右のモーターの速さに差をつけて直進ないようにします。すると状態③になり再び左右のセンサーが白になります。この動作を繰り返すことでコースに沿って進んでいきます。

ここで右センサーと右モーターの関係に注目してみましょう。右センサーが白の時右モーターは高速、右センサーが黒の時右モーターは低速もしくは停止です。これをプログラムで表現したのが下図です。これに左センサー、左モーターの関係も追加すれば完成です。

プログラム STEP3

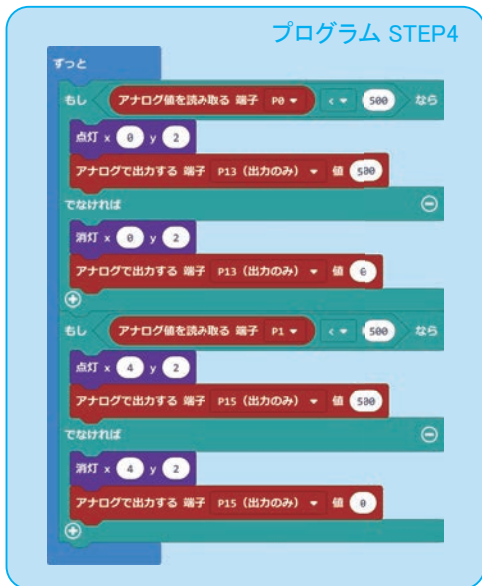
ずっと

```

もし アナログ値を読み取る 端子 P0 < 500 ならば
    点灯 x 8 y 2
    アナログで出力する 端子 P13 (出力のみ) 値 500
でなければ
    消灯 x 8 y 2
    アナログで出力する 端子 P13 (出力のみ) 値 0
    
```

プログラム完成

これまでは右センサーと右モーターだけのプログラムでしたので、左側も追加すれば完成です。ただし、このプログラムで正しく走るとは限りません。急カーブがあるコースなどではスピードが速すぎると脱線してしまうかもしれません。センサーのしきい値もこのまでは白・黒を判別できないかもしれません。実際に走らせてみてプログラムを工夫してみましょう。



さらに詳しい情報はこちら

<https://switch-education.com/products/microbit-linetracer/>

(うまくいかない時の対処方法、STEP ごとのプログラムコード、サンプルコースなど)



オプション

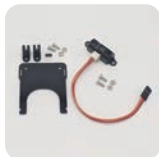
ハンダづけライトレーサー用バンパーセット



前方の衝突検知ができるようになります。
物体と衝突すると micro:bit のボタンが押される
仕組みになっています。

<https://sedu.link/mb-lt-bumper>

ハンダづけライトレーサー用距離センサーセット



前方の障害物との距離を10cm～80cmの
範囲で計測します。接続にはCN2(P2)にピンヘッ
ダーのハンダ付けが必要です。(p.7 参照)

<https://sedu.link/mb-lt-distance>

ライトレーサー用パネルセット



直線と曲線のパネルを組み合わせて簡単な
コースが作成できます。

<https://sedu.link/mb-lt-course>

micro:bit を使った作例集

<https://switch-education.com/examples/>

micro:bit の商品ページ

<https://switch-education.com/products/microbit/>