

鎌ヶ谷市 中学校様向け

micro:bit(マイクロビット)を使用した
プログラミング実習マニュアル



ＩＴサポートありのみ 編集

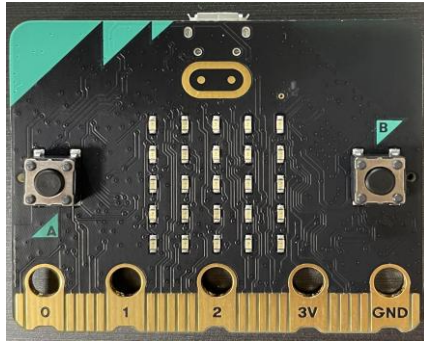
2022/4/11

目次

(下線のある項目はリンクしています)

micro:bit (マイクロビット) とは	3
ボタンと各種センサー	4
Chromebookとマイクロビットの接続	5
接続の確認 (USB接続)	6
マイクロビットに付属しているセンサー	7
MakeCodeの使い方	
準備：Make code (プログラミング作成ツール) を起動する	9
補足説明：MakeCodeの画面について	10
TRY1：プログラミングをする	11
TRY2：micro:bitに転送する	12
TRY3：プログラミングの保存する	13
キットを使用したプログラミング① (信号機)	
準備1.2：キットを利用するためのブロックを追加する	15
準備3：コントロールボックスにマイクロビットをセットする	16
TRY4：信号機の動きを考え、キットを使ってシュミレーションしよう	17
TRY5：関数を使ってプログラミングを作成しよう	18
TRY6：「左折可」の矢印を表示する信号機にしよう	19
TRY7：「左折可」の矢印を表示と音を鳴らそう	20
TRY8：分岐処理を使ったプログラミングを作成しよう	21
双方向性のあるコンテンツ	
TRY9：無線を使うプログラミングをしよう	23
TRY10：双方向性のある信号機をプログラミングしよう	24
TRY11：A信号とX信号についても、動きを考えよう	25
計測制御のあるコンテンツ	
キットを使用したプログラミング② (自動車)	
TRY12：自動車を前進させよう	28
TRY13：障害物を感知させた自動車をプログラミングしよう	29
プログラミング例・目次	30

micro:bit(マイクロビット)とは

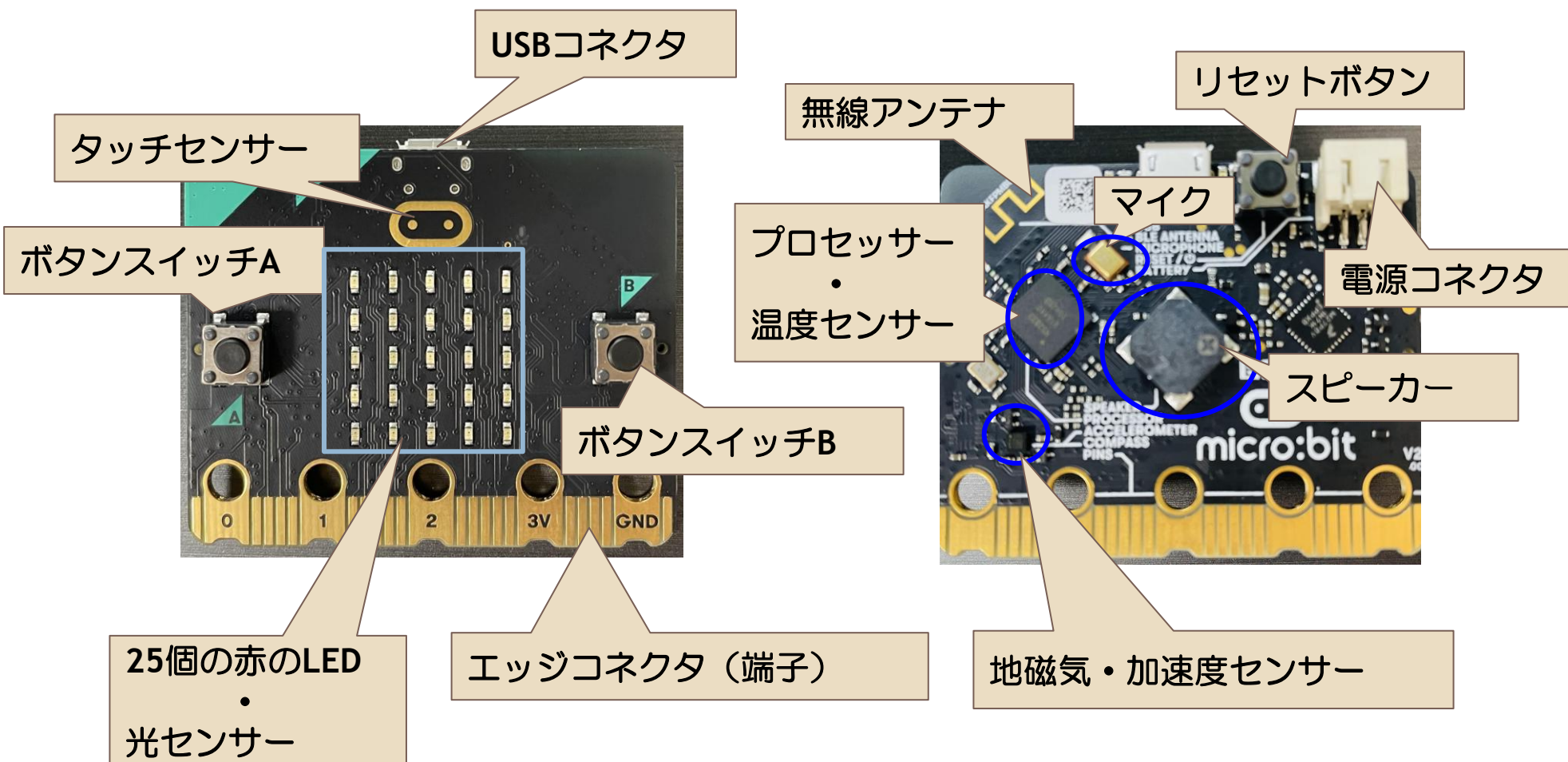


マイコンボード

マイクロコンピュータと入出力回路などを1枚の基盤に乗せたもの。

・教科書P205参照

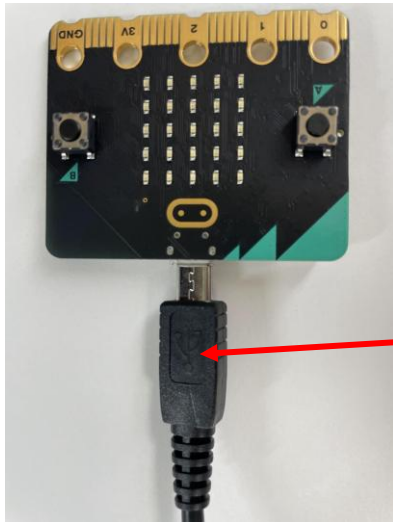
ボタンと各種センサー



Chromebook と micro:bit の接続

1. USBケーブルを準備します。
2. マイクロビットとUSBケーブルを接続します

3. ChromebookとUSBケーブルを接続します。

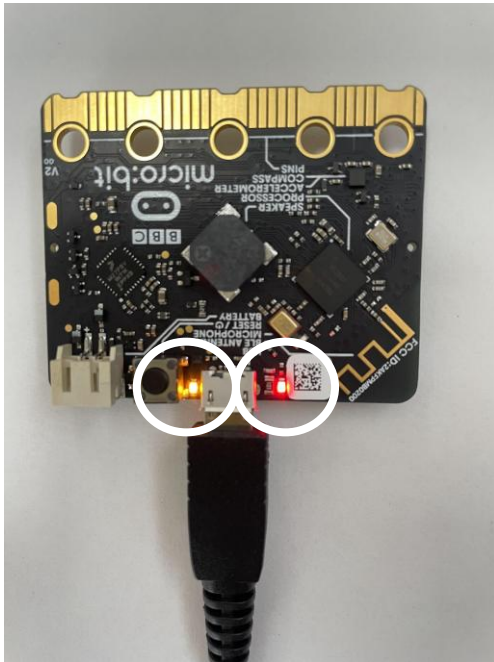


USBマーク
が上向き



接続の確認(USB接続)

micro:bitの2つのライトが点滅



※ファイルを開く



MICROBITの表示があれば接続中

MICROBITを抜くときは、▲をタップ。表示が消えてから抜くこと。

micro:bit に付属しているセンサー類

- 【温度センサー】

基板上のICチップ周辺の温度を計測する。

- 【加速度センサー】

基盤の動きを計測する。例) micro:bitが揺さぶられたときに特定の動作を実行させる。

- 【地磁気センサー（デジタルコンパス）】

地球の磁場（磁界）の方向を計測する。例) micro:bitの頭が北を向いたら「N」と表示する。

- 【タッチセンサー】

マイクロビットのロゴマークを触ると感知する。

- 【光センサー】

明るさのレベルを測定する。

- 【無線アンテナ】

無線機能を使うことで、micro:bit間でメッセージの送受信ができる。

MakeCodeの使い方

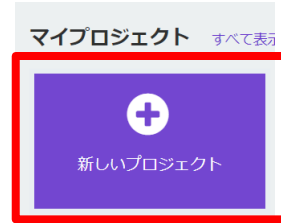
MakeCode（メイクコード）を使ってプログラミングする



準備：MakeCode(プログラミング作成ツール)を起動する

1. MakeCodeをWEBブラウザから起動 <https://makecode.microbit.org/>

2. 「新しいプロジェクト」を選択



3. 「プロジェクトを作成する」から、
プロジェクト名（ファイル名）を入力し、
【作成】ボタンを選択



4. 左がmicro:bitのシミュレーション画面、右がブロックプログラミング作業領域になる。



補足説明: MakeCodeの画面について



TRY1: プログラミングをする

★LEDディスプレイに大きなハートマークと小さなハートマークを交互にずっと表示するプログラミングを作成してみる。

使用するブロック

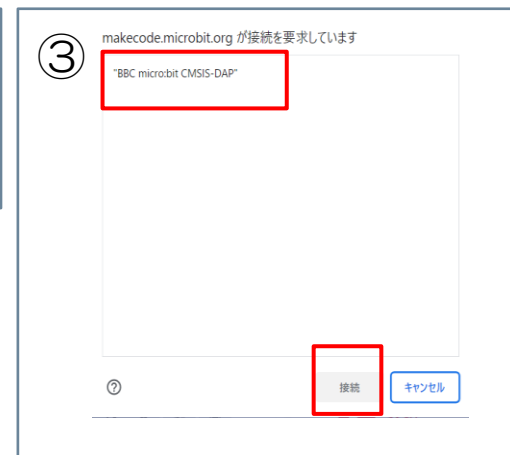
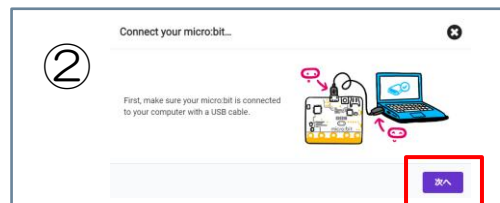
- 「ずっと」
- 「アイコンを表示」



TRY2: micro:bitに転送する

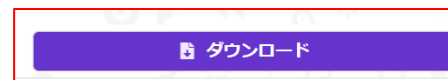
①ペアリング作業

- ① ダウンロードボタン横の3点リーダーボタンから「Connect device」を選択
- ② 接続の画面がでたら、【次へ】を選択
- ③ 接続を要求している画面から、「BBC micro:bit CMSIS-DAP」を選択し、【接続】を選択
- ④ 「micro:bitに接続しています」の画面がでたら、【完了】を選択



②プログラミング転送作業

- ① ダウンロードボタンを選択
- ② ダウンロードしていますと表示され、下記の表示が一瞬出たら正常にダウンロードできています。

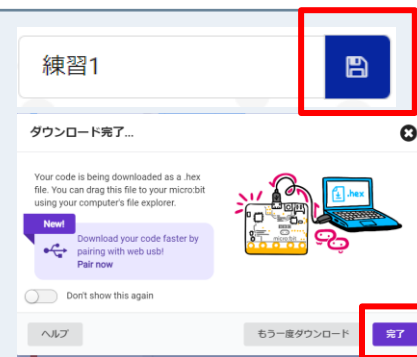


※USBを抜くときは、URL先頭の鍵マークを選択し、micro:bitの欄の×を選択してから、USBケーブルをぬきます。（ペアリング解除）

TRY3: プログラミングを保存する

※プログラミングを保存するには、ダウンロードを、ファイルとして保存しておきます。
オープニング画面の過去のプロジェクトの履歴からも呼び出すことができますが、そちらからは最後に閉じた画面が上書きされた状態になっています。

- ① ファイル名をつけてフロッピーマークを選択
(プロジェクト名のままでよければそのままでも良い)
- ③ ダウンロード完了の画面が出ますので、完了を選択
- ④ ダウンロードフォルダに保存されます



ファイルを読み込むには？

- ② 「ファイルを読み込む」を選択



- ① オープニング画面から「読み込む」ボタンを選択



- ③ 「ファイルを開く」を選択し、ダウンロードフォルダ内のファイルを選択

キットを使用したプログラミング①

信号機の動きをプログラミングする



準備1: キットを利用するためのブロックを追加する

※キットは組み立てておきましょう

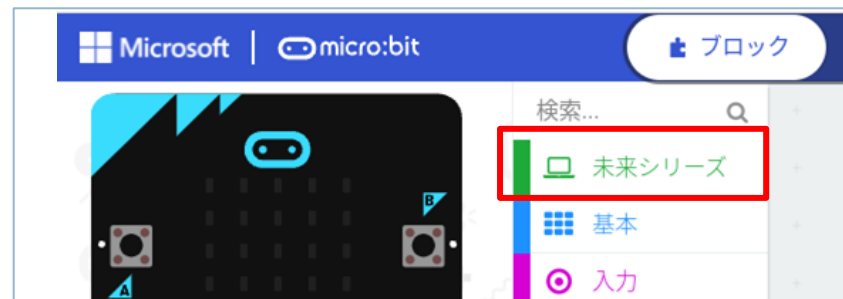
操作手順

1. MakeCode起動
2. マイプロジェクトの右上、
【読み込む】を選択
3. 「URLから読み込む」を選択

指示されたURLを入力（下記URLを貼り付ける）

https://makecode.microbit.org/_2fDWzfKJfD2b

4. 未来シリーズのブロックが
ツールボックスに追加される



準備2: 作成したプログラムをマイクロビットに転送する

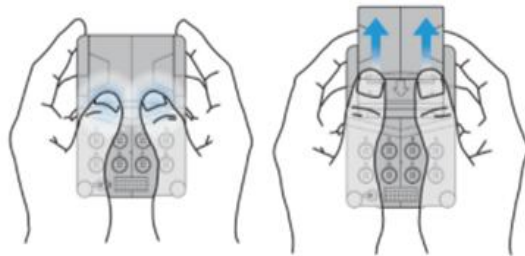
※ スライド P.13 「TRY2: micro:bitに転送する」 参照

準備3: コントロールボックスにマイクロビットをセットする

※転落防止のため、micro:bit ボードをセットし、キットを床に置いてからコントロールボックスのスイッチをオンにする。動きを確認する。

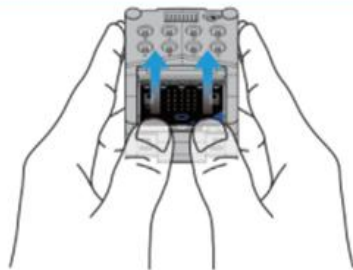
1. カバーを外します

親指でカバーの下部を軽く押しなが
ら矢印の方向にスライドさせるとカ
バーが外れます。



2. micro:bit ボードを セットします

micro:bit ボードをコントロー
ルボックスにセットします。



3. 正しくセットできているか 確認しましょう

micro:bit ボードは左の写真
のように正しくセットできて
いるか確認しましょう。右の
4 つの写真のようになってい
ませんか？

正しくセット
できている状態



間違ったセットの例



完全に挿入
できていない



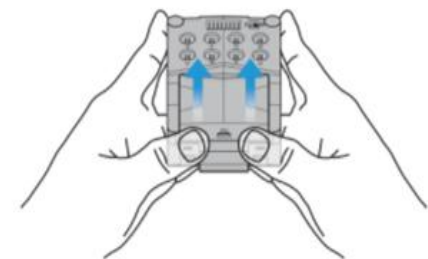
裏表が逆



斜め挿入

4. カバーを取り付けましょう

カバーをスライドさせて取り
付けたら完成です。

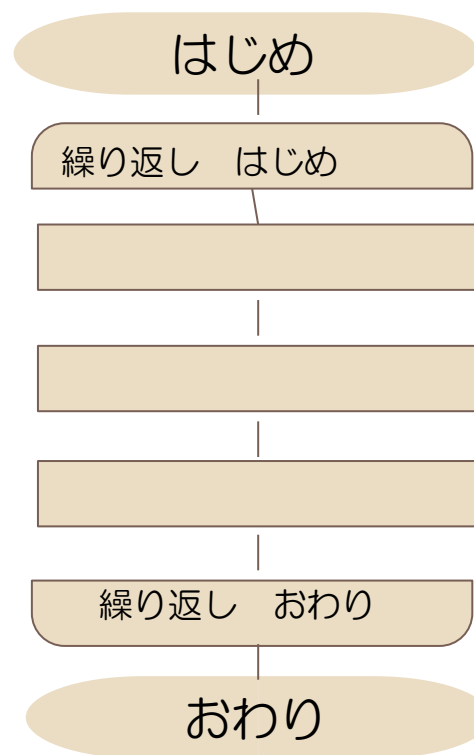


TRY4：信号機の動きを考え、キットを使ってシミュレーションしよう

操作手順

1. 信号機の動きをフローチャートで作ってみよう
(教科書P208参照)
2. MakeCodeを使ってプログラミングする
3. 作成したプログラムをmicro:bitに転送する
4. micro:bitをコントロールボックスにセットし、信号機の動きを確認する

手順1：信号機のフローチャートを書く

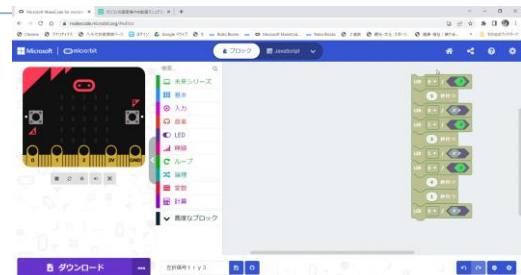


TRY5: 関数を使ってプログラミングを作成しよう

処理をそのまま並べず、まとめたものを作るとわかりやすくなります。
これを**サブルーチン（関数）**といいます。同じ処理を何度も利用するときには便利です。
ここでは、前に作った「青」「黄」「赤」の点灯を関数を使ってプログラミングしてみます。

操作手順

1. 高度なブロック内の「関数」選択
2. 関数から「関数を作成する」選択
3. 関数の編集画面が出たら、完了ボタン
4. 「doSomething」の文字をわかりやすい関数名に変更する。
ここでは「青」
5. 先に作成した信号機の青の点灯動作のブロックを挿入
6. 同じように「黄」「赤」も作成する
7. 実行するためのブロックは、「関数」から呼び出しを使用する



TRY6:「左折可」の矢印を表示する信号機にしよう

★micro:bitのLED画面を利用して、左折の矢印を表示してみましょう。



正方形を塗りつぶして、矢印の絵にしよう！
(正方形を選択するとLEDを光らせたいところが白くなります。)

TRY7:「左折可」の矢印を表示と音を鳴らそう

★micro:bitの「音楽」から「音を鳴らす」ブロックを使ってみよう。
音用の関数を作ると、プログラムがわかりやすくなります。



TRY8：分岐処理を使ったプログラミングを作成しよう

☆「もし～なら～して」という条件を使った処理になります。

ここでは、夜中の信号を再現してみましょう。

ずっと黄色の点滅信号が続いている信号機から、Aボタンが押されたら、通常の信号の動作を1回行う処理を作ってみましょう。

TRY: 双方向性のあるシステムを作ろう

双方向性のあるコンテンツ

教科書P238参照



双方向性とは？

何かの働きかけ（入力）によって、何かを応答（出力）すること。

相互にやりとりすることができることを「双方向性がある」と定義しています。

micro:bit の無線機能を使って、双方向性のある信号機システムを作成してみます。

TRY9: 無線を使うプログラミングをしよう

複数の micro:bit 間で、無線で数値や文字を送受信できます。

☆数値や文字は半角英数文字で入力したものです。

2つの micro:bit を使って、数字や文字を送信してみましょう。

無線ブロックの扱い方

- 無線を行うときは無線グループを設定します (0~255)
- 無線で数字を送信するときは、受信者側は `receivedNumber`
- 無線で文字を送信するときは、受信者側は `receivedString`

無線練習 1

ボタンAを押したとき、相手に数字のメッセージを送る

無線練習 2

ボタンBを押したとき、相手に文字を送る

注意！！

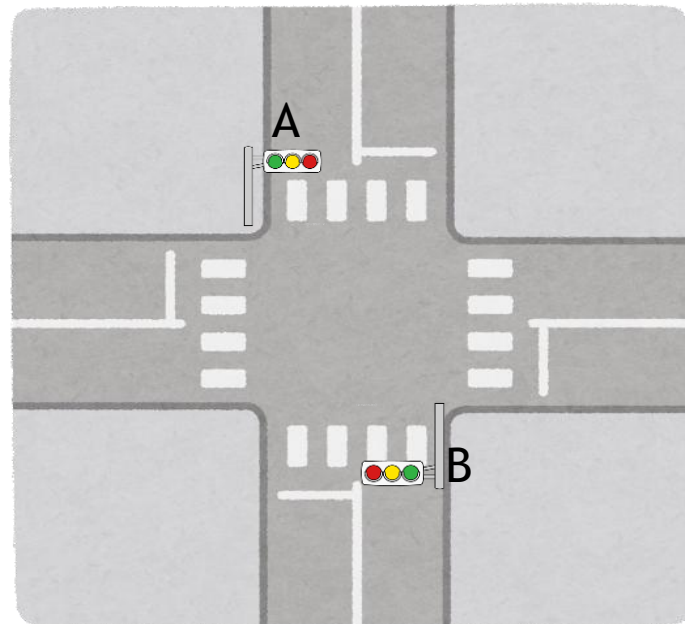
※マイクロビットは半角英数のみしか表示できません。

※グループごとに同じグループ番号を設定しましょう。

TRY10: AとBの信号機が同時に動く信号機をプログラミングしよう

無線を使って、Aと双方向性のある信号機をプログラミングしてみよう。

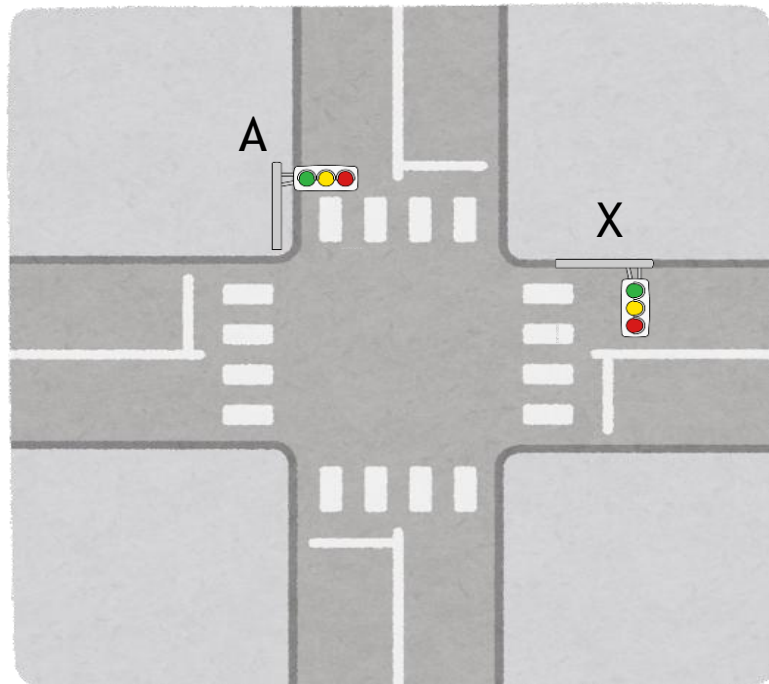
☆ヒント：数字を送受信することで、同じプログラムを開始するプログラムを考えてみよう。



TRY11:A信号とX信号についても、動きを考えよう

無線を使い、Aの信号とXの信号が正しく動くようにプログラミングする。

☆ヒント：X信号が数字を受信することで、A信号とは違うプログラムを開始するプログラムを考えてみよう。



計測制御のあるコンテンツ

教科書P194・P248参照



計測・制御システムとは？

コンピュータを使って、自動的に目的とする動きを機器にさせるためには、必要な情報を得ること（制御）が必要です。計測・制御システムは、周囲の状況を計測する部分（センサ）、入力された情報から処理の方法を判断し、命令を出す部分（コンピュータ）、実際に出力や動作などをする部分（アクチュエータなど）を組み合わせることで構成されています。

（教科書P204参照）

キットを使用したプログラミング②

自動車の動きをプログラミングする



TRY12：自動車を前進させる

手順

1. 自動車を組み立てよう（組み立て方参照）
2. フローチャートを書いて、流れを確認しよう
3. MakeCode（未来シリーズ追加版）を使ってプログラミングをしよう
4. micro:bit にプログラムを転送しよう
5. 転送したmicro:bit をコントロールボックスにセットし、動きを確認しよう
6. うまく動かなかったら原因を考えて修正しよう

TIPS：

- ・回転用サーボモーターの回転速度：60～100が適正值
- ・まっすぐに進まないときは、重心が一定になっていない状態です。

右と左のモーターの速度出力を「右側速度100、左側速度80」というように真っすぐ走る条件を探して調整します。

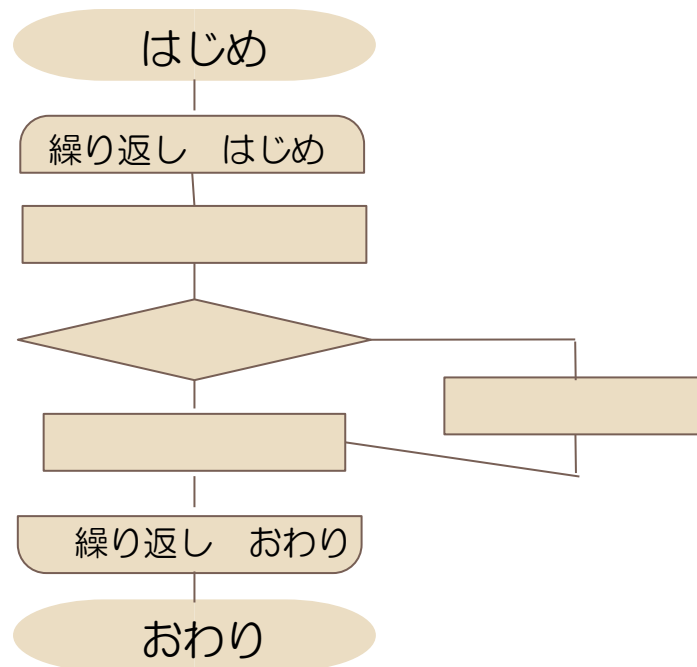
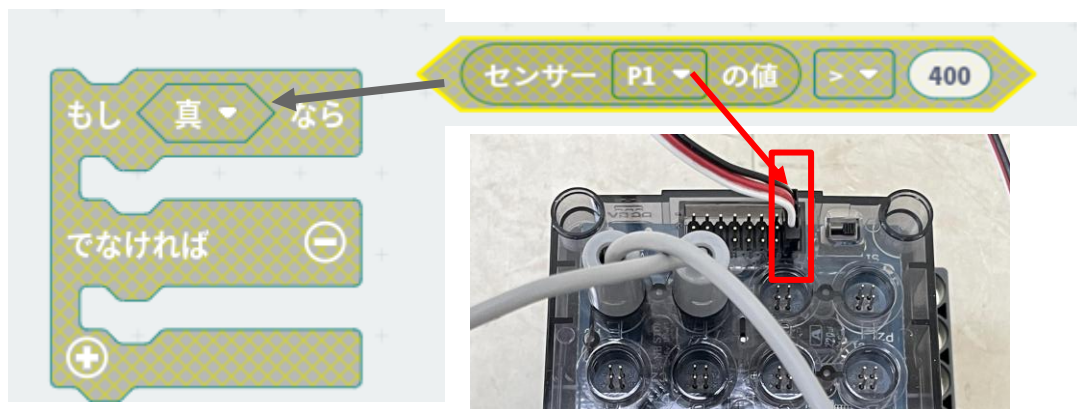
TRY13:障害物を感知させるプログラミングしてみよう

※障害物を感知したら、一度停止し、バックをしてから動くプログラミングを考えよう。

ヒント

1. センサーをP1に取り付けよう
2. 論理式を使ってプログラミングをしてみよう(下図参照)
3. 作成したプログラムを信号機に転送して動作を確認しよう

フローチャートをかいてみよう



※【赤外線センサー】参考値
最小値(遠)：85、最大値(近)：947

プログラミング例・目次

セクション	該当ページ	プログラミング項目	ページ
TRY4	17	信号機点灯プログラミング	31
TRY5	18	関数を使った信号機点灯プログラミング	32
TRY6	19	「左折可」の矢印が表示するプログラミング	33
TRY7	20	「左折可」および「音」が出るプログラミング	34
TRY8	21	分岐処理が使われたプログラミング	35
TRY9	22	無線を使ったプログラミング練習	36
TRY10	24	A信号とB信号が同時に動くプログラミング	37
TRY11	25	A信号とX信号が動くプログラミング（受信側）	38
TRY12	28	自動車を前進させるプログラミング	39
TRY13	29	障害物を感知しバックさせるプログラミング	40



TRY4) 信号機点灯プログラミング作成例

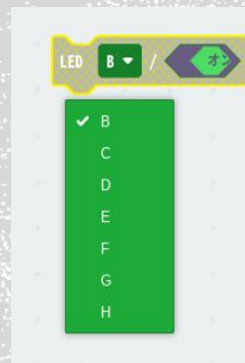
31



使用するブロック

信号機を点灯させるブロック

- LED B C D オン・オフ を利用
- ○秒待つ



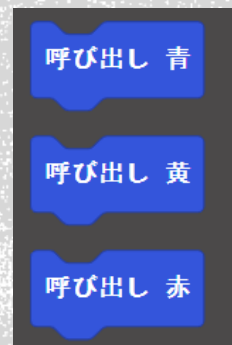
TRY5) 関数を使った信号機点灯プログラミング作成例

32



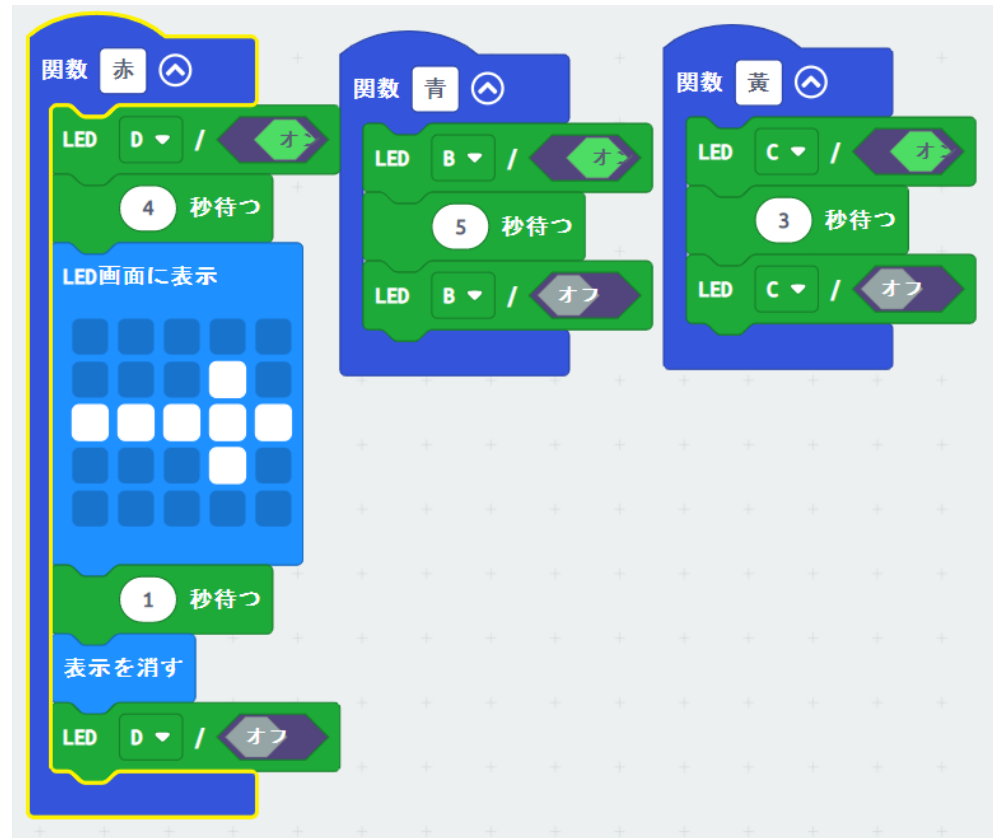
新たに使用するブロック

- 関数
- 関数-呼び出し



TRY6)「左折可」の矢印が表示するプログラミング作成例

33



新たに使用するブロック

- LED画面に表示



TRY7)「左折可」および「音」が出るプログラミング作成例

34



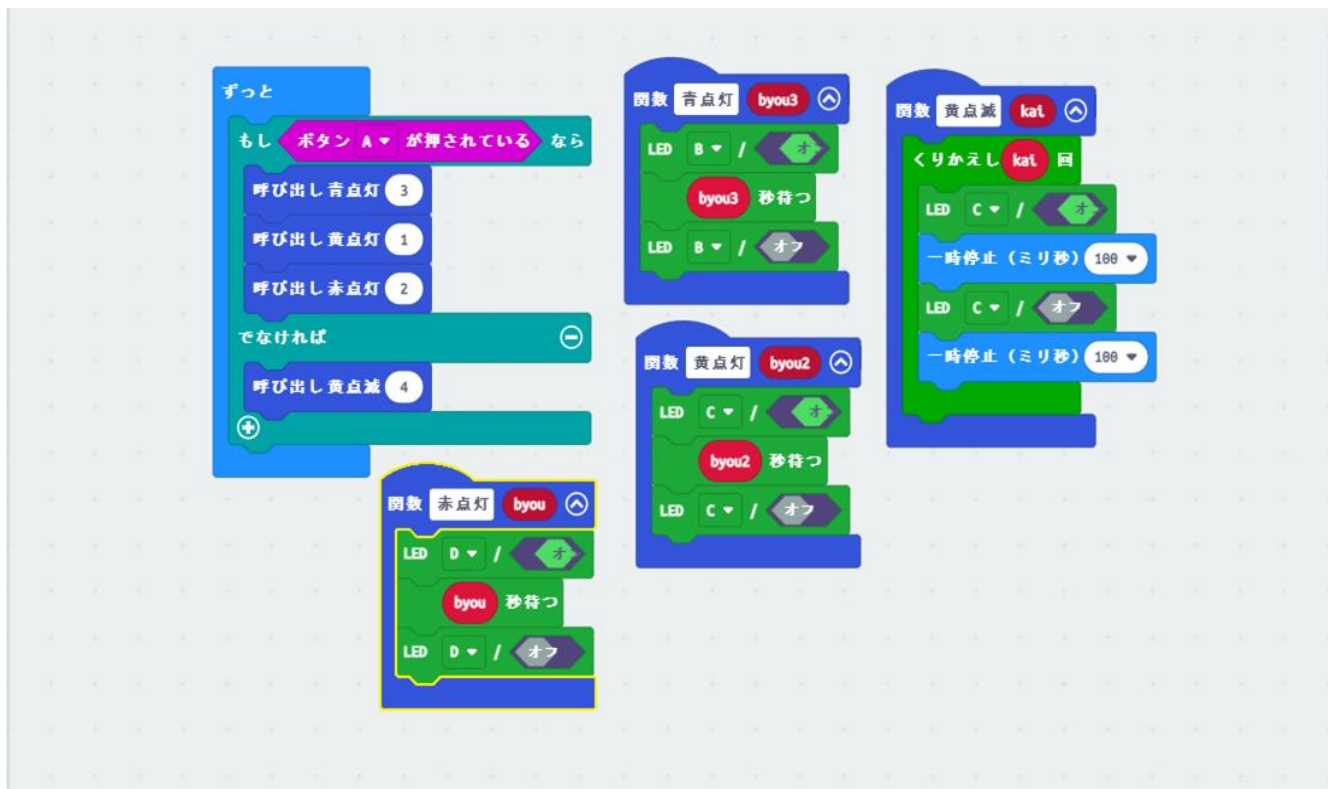
新たに使用するブロック

- 音楽-音を鳴らす



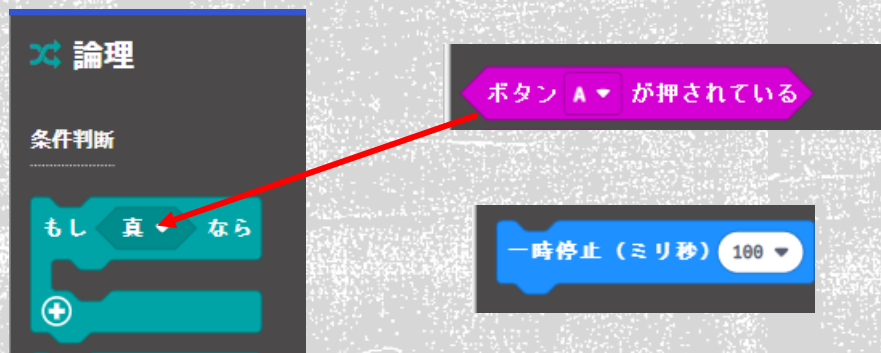
TRY8) 分岐処理が使われたプログラミング作成例

35



新たに使用するブロック

- もし～なら
- ボタンAが押されている
- 一時停止



TRY9) 無線を使ったプログラミング練習作成例

Aさん

36



Bさん



新たに使用するブロック

- 最初だけ
- 最初のグループを設定
(無線グループごとに同一のもの) ここでは1を設定
- 無線で数値を送信 • 無線で文字列を送信

使用するブロック

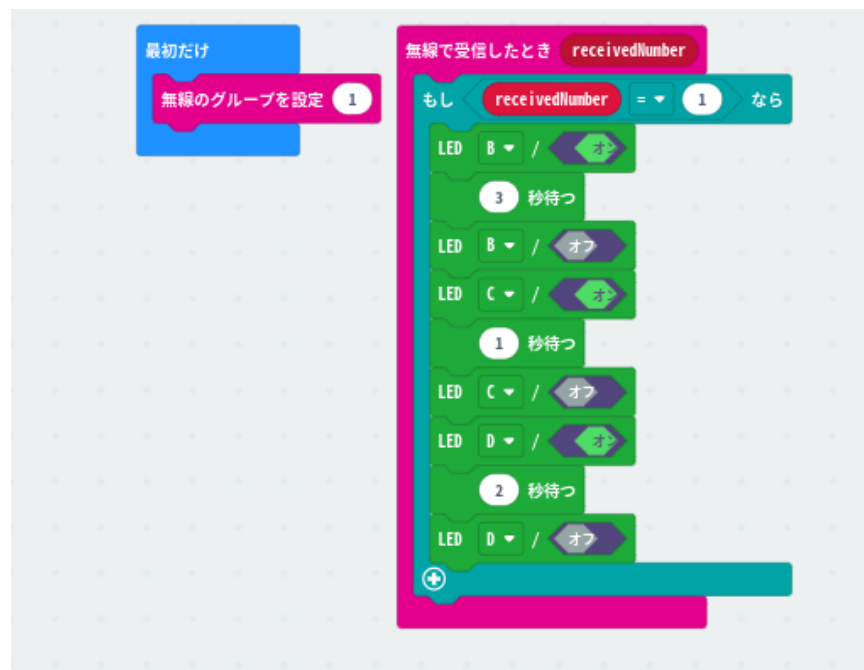
- 最初だけ
- 無線で受信したとき receivedNumber
- 無線で受信したとき receivedString
- 数を表示 (receivedNumber)
- 文字列を表示(receivedString)

TRY10) A信号とB信号が同時に動くプログラミング作成例

Aさんのプログラム



Bさんのプログラム



37

Aさんの使用するブロック

- 無線のグループを設定
- 無線で数値を送信
(信号のプログラムに必要なブロック)

Bさんの使用するブロック

- 無線で受信したとき
- もし～なら
- $\bigcirc = \bigcirc$

※真のところに数式をブロックをあてはめる

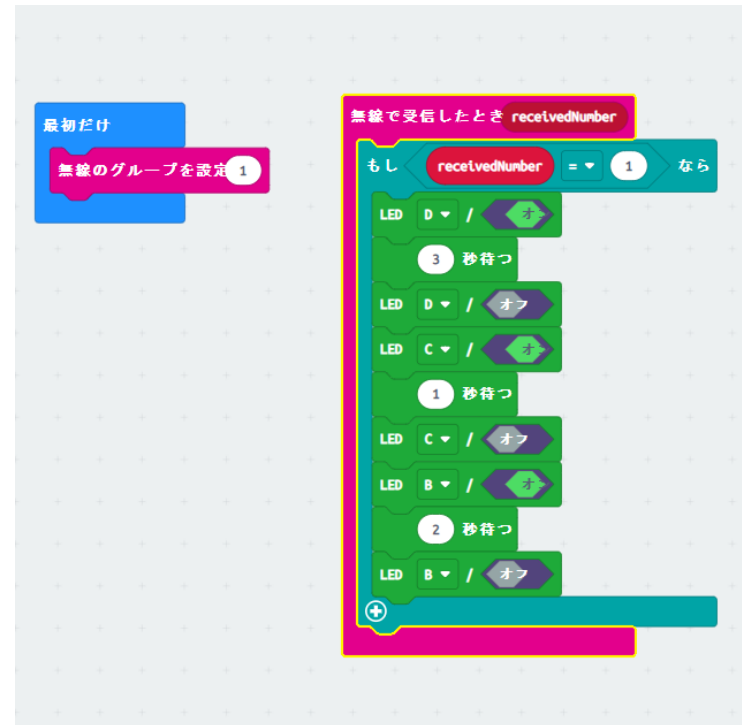


※両者、同じ無線のプログラムを設定すること。ここでは1を設定。

TRY11) A信号とX信号が動くプログラミング作成例

受信側のBさんのプログラムを変更してみよう

38



※通常の信号とは逆の動き(赤⇒黄⇒青)になるように、LEDのプログラムを変更してみる

TRY12) 自動車が前進するプログラミング作成例

39



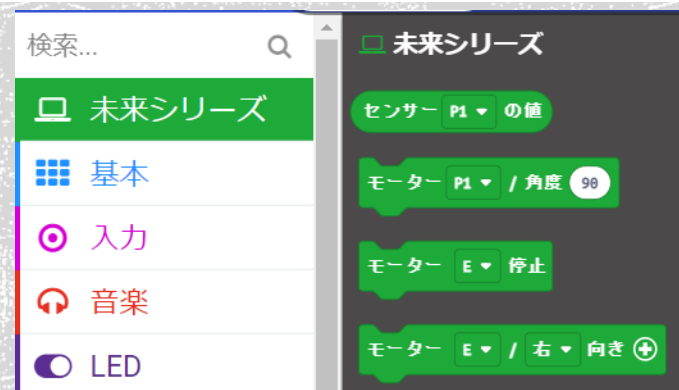
HINT

本体の重心が少しでもずれてしまうと、スピードの設定が左右同じでもまっすぐに走らないことがあります。
その時は左右のバランスをみながら、速度を調整してみましょう。

新たに使用するブロック

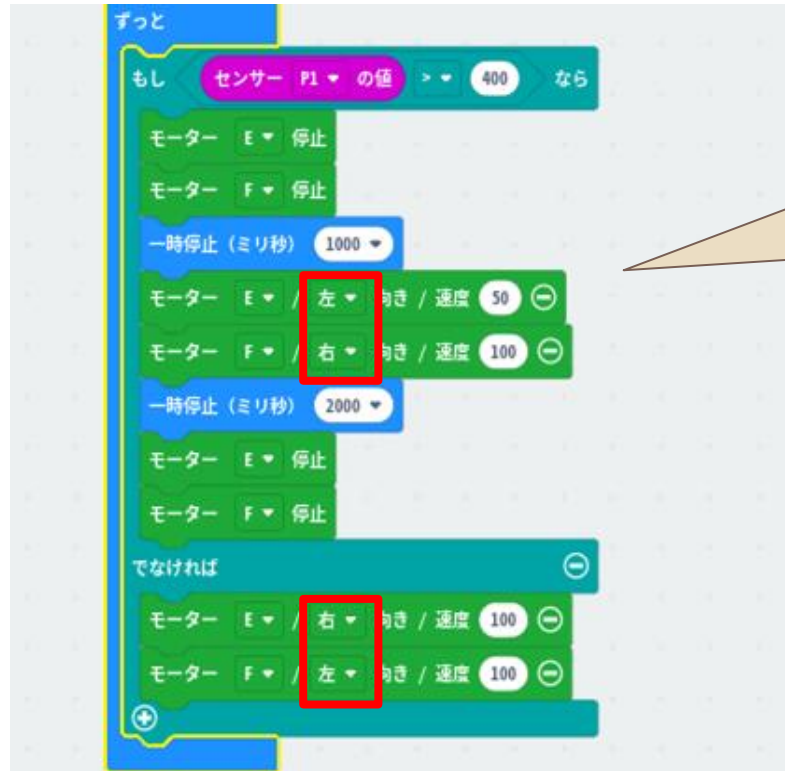
未来シリーズカテゴリより選択

• モーター 向き • 速度



TRY13) 障害物を感知し、バックするプログラミング作成例

40



車をバックさせるためには、
モーターの右と左を変えてみよう！
速度は適正な値になるまで、
試してみよう！

新たに使用するブロック

- 入力-センサーP1の値

(※センサーがP1につけられている時)

