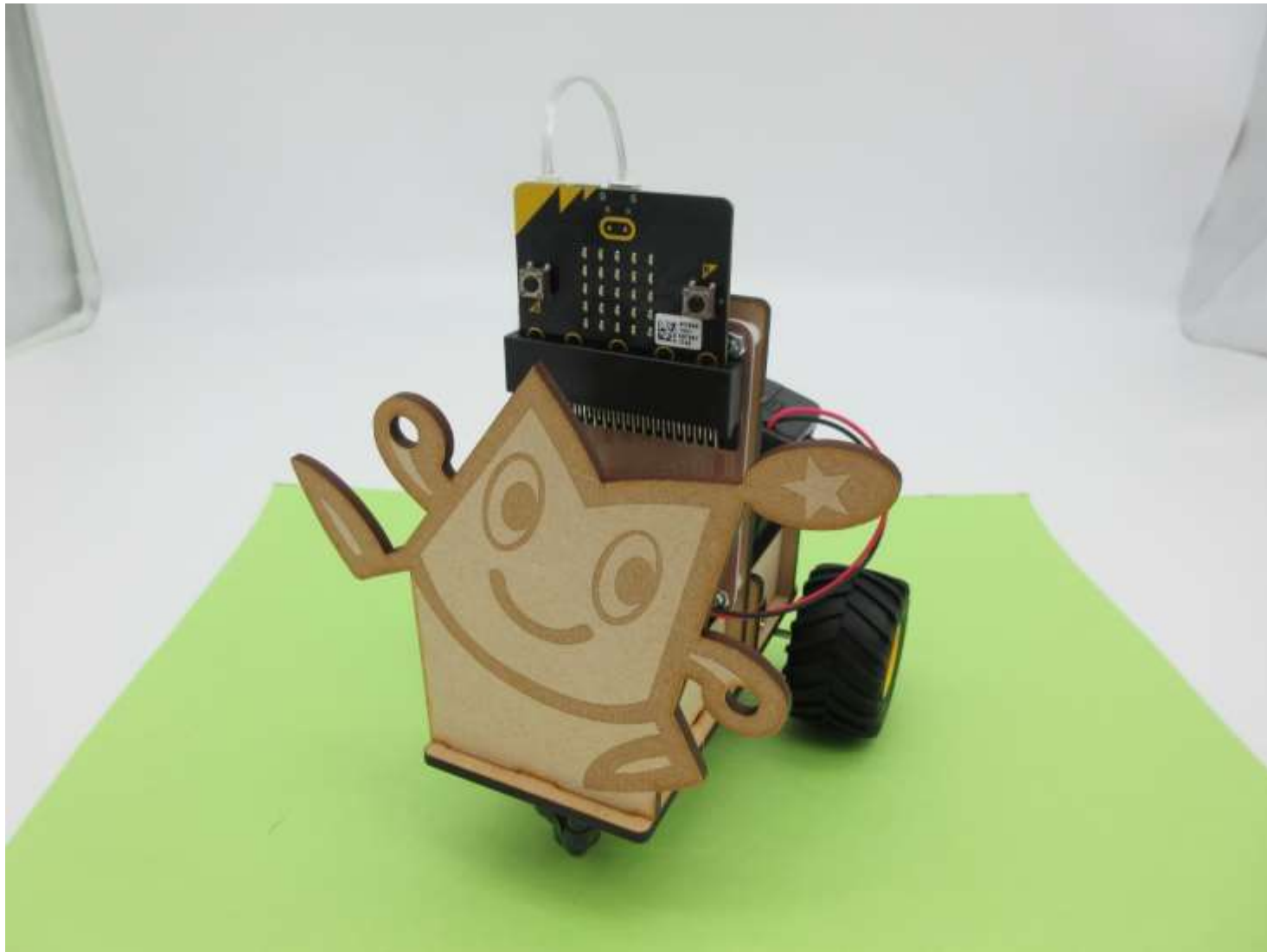


動くロボットプログラミング教室 ～マイクロビットで動く車を作ろう～

Apple版







ロボットの5大要素

コンピュータ



センサ



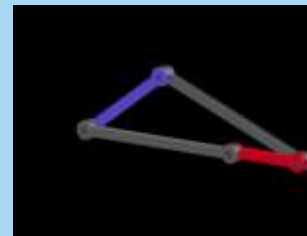
アクチュエータ



エネルギー源



機構(きこう)





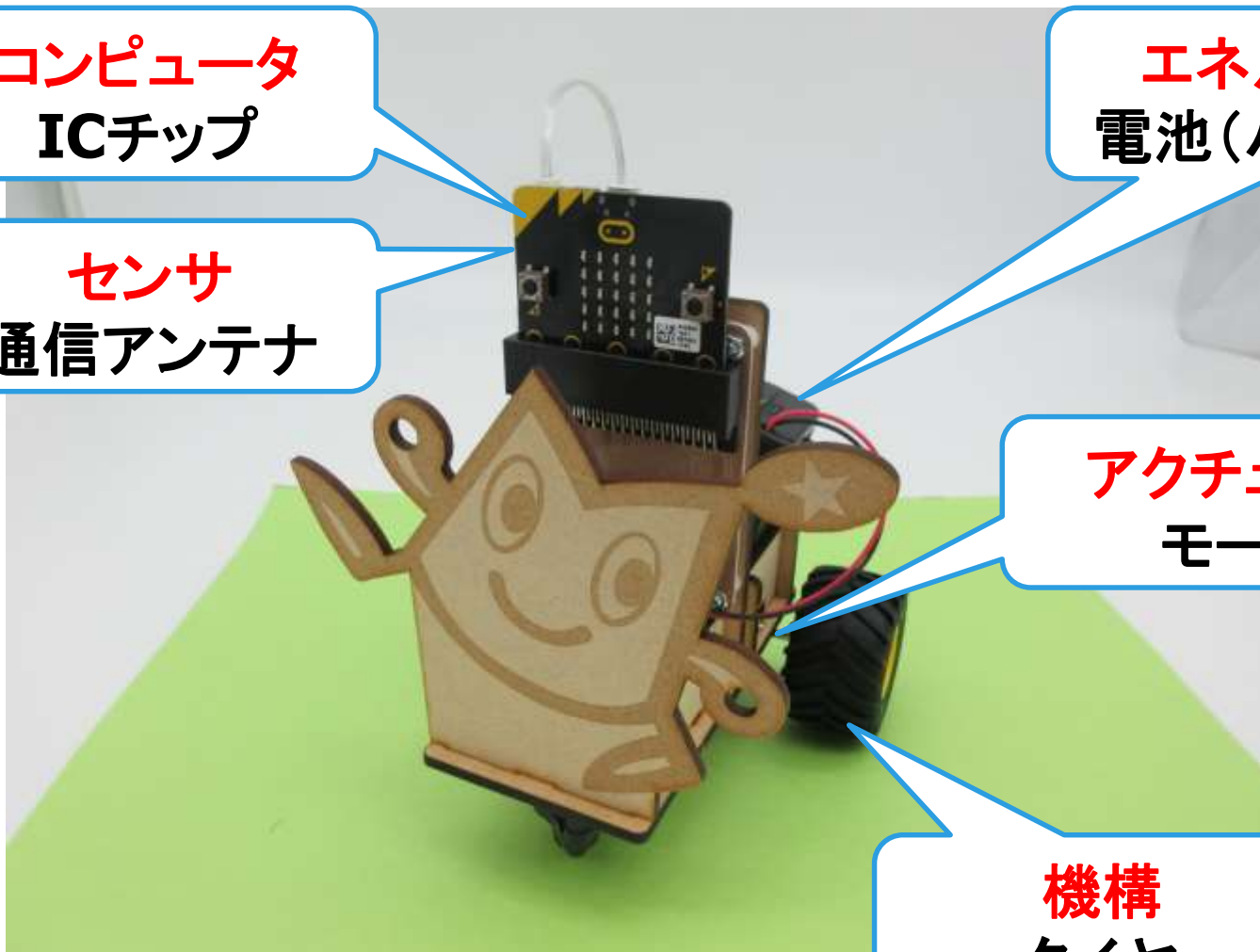
コンピュータ
ICチップ

センサ
通信アンテナ

エネルギー源
電池(バッテリー)

アクチュエータ
モーター

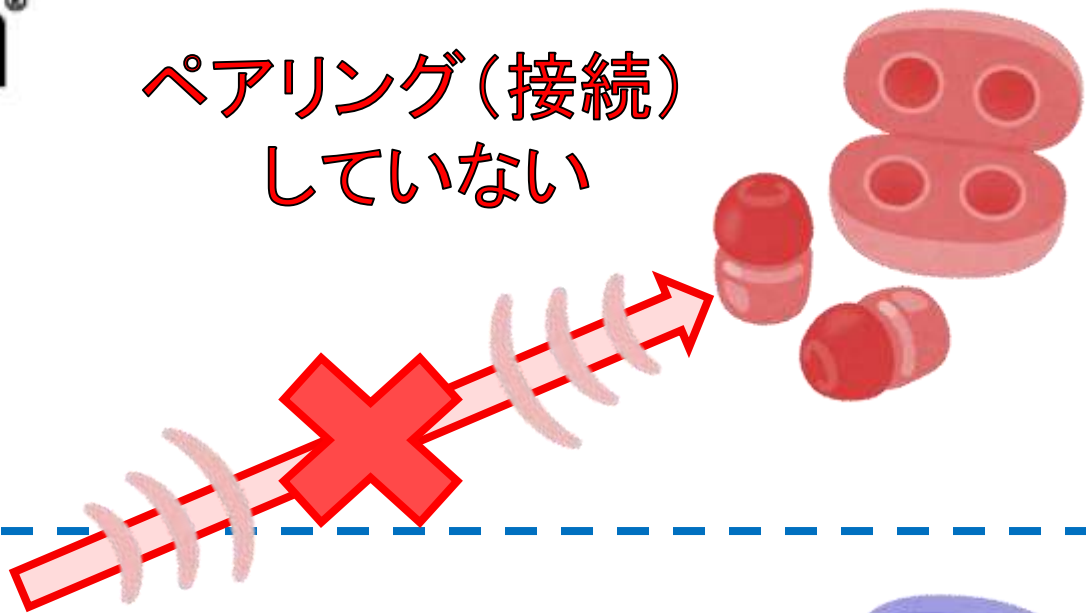
機構
タイヤ





ペアリング(接続)
していない

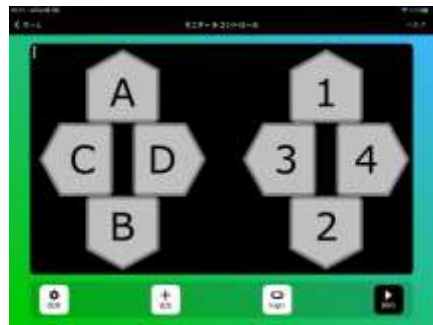
A



B

ペアリング(接続)済み

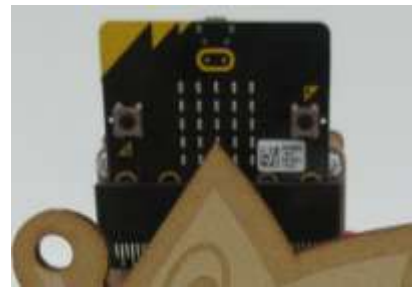
動作するための流れ



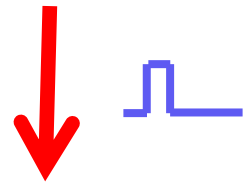
スマート
フォン



通信
アンテナ



マイクロビット



モータ
ドライバ



モータ



走る





ケガをしないように

- 集中する
- つかれたら休む
- まわりをかたづける
- あわてない



<https://www.irasutoya.com/>



作成のポイント

- 工具を正しく使う



- 説明書の写真や図をしっかりと見る

- わからないところはよく見る



<https://www.irasutoya.com/>



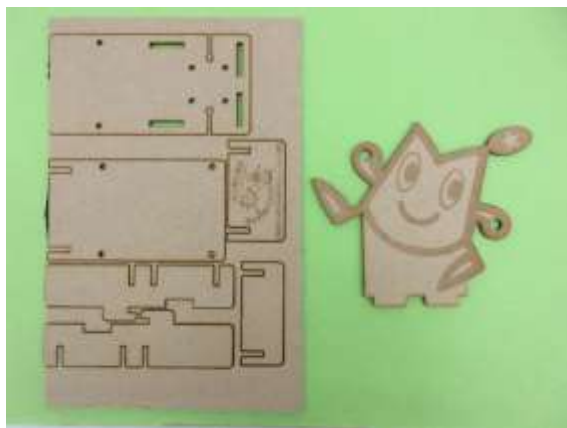
■ 道具の確認

- 紙皿
- ラジオペンチ
- ドライバー
- えんぴつ





■ 全部あるか確認しよう



車体パーツセット



ギアボックス



キャスター



基板セット



Micro:bit セット



タイヤ



■ 全部あるか確認しよう



短いネジ
6本



長いネジ
4本



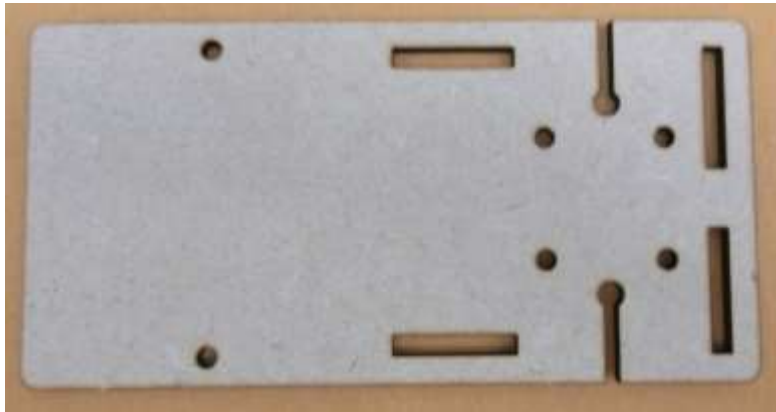
ナット
14個



スペーサー
4個



■ 用意するもの



土台パーツ



短いネジ
2本



ナット
2個



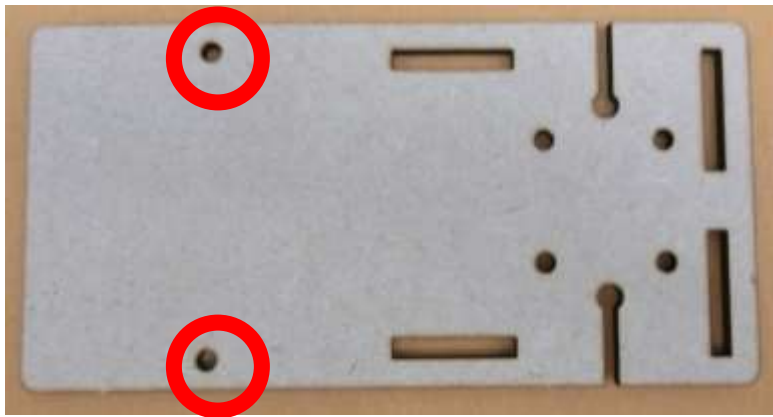
ギアボックス

①車体パーツセットから
土台を取り外す

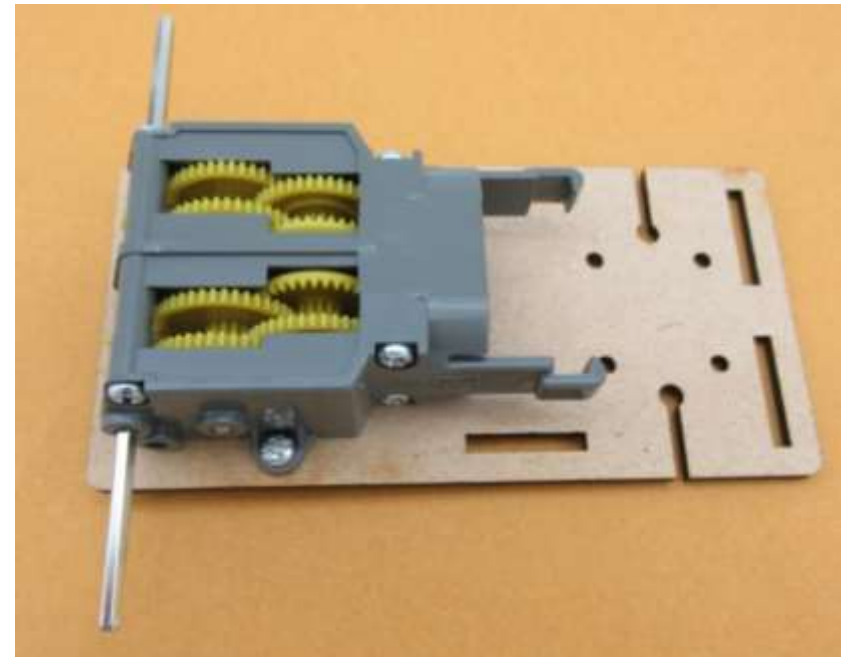
ねじの長さ注意



- 車体パーツセットから取り外した土台にギアボックスを取り付ける



丸の位置に短いねじで固定するよ



ギアボックス側からねじをさしこみ
うら側をナットでとめるよ



■ 用意するもの



基板セット



仕切り大



長いネジ



スペーサー
各 4 個

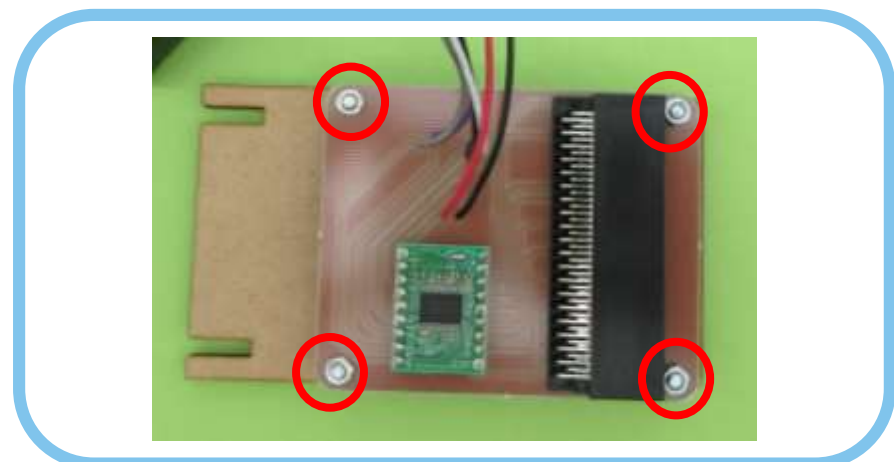


ナット
8 個

■ 仕切り大に基板セットを ねじとナットで取り付ける(4か所)

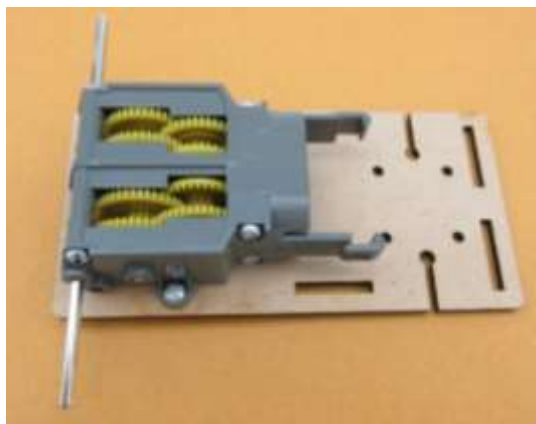


- ① **スペーサー**をナットでとめる
- ② **基盤**をナットでとめる

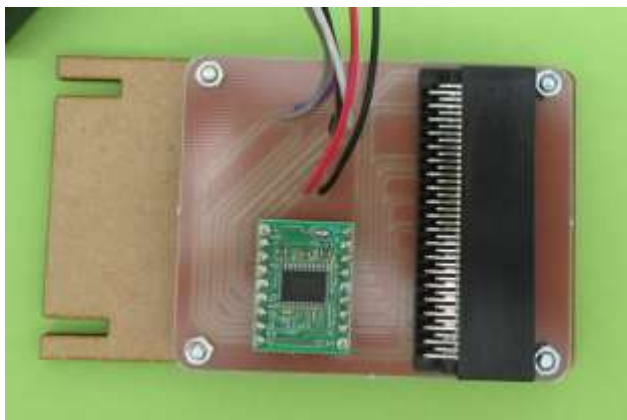




■ 用意するもの

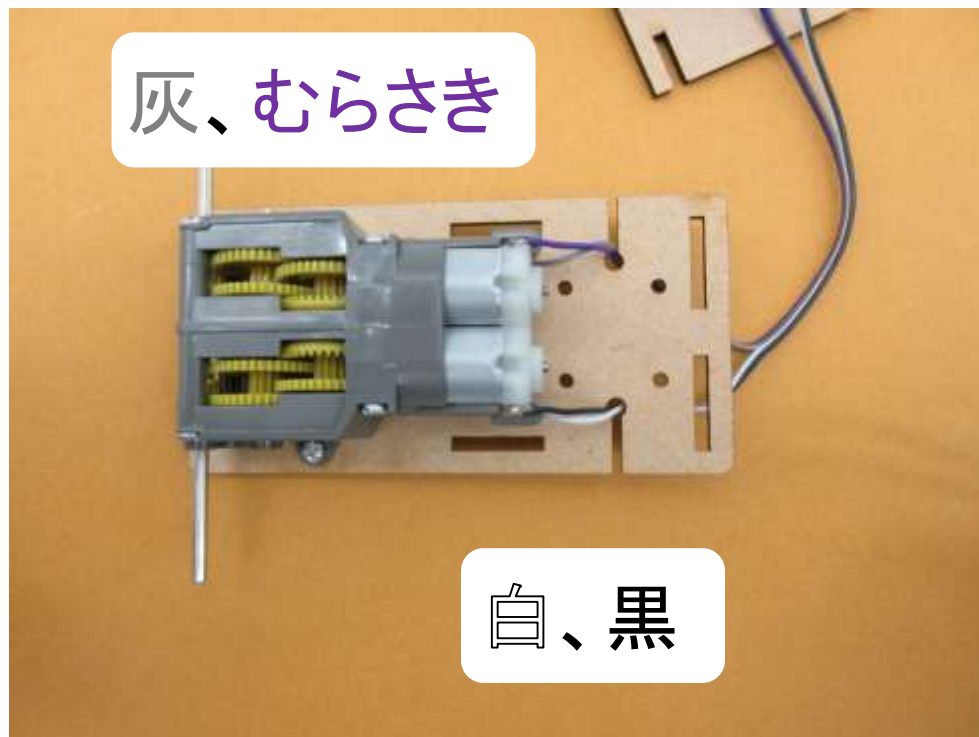


ギアボックス



基板セット

基板についているモーターを土台の切れ込みを通してギヤボックスにさしこむ線の色に注意しよう



灰、むらさき

白、黒



■ 用意するもの



キャスター



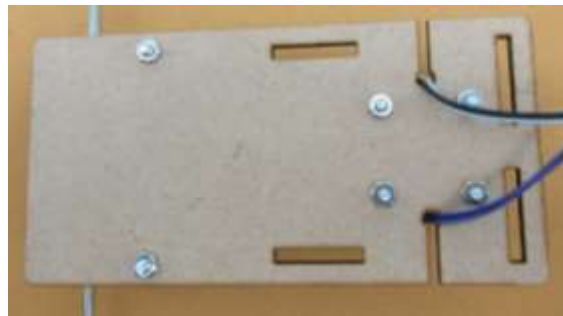
短いねじ



ナット

各 4 個

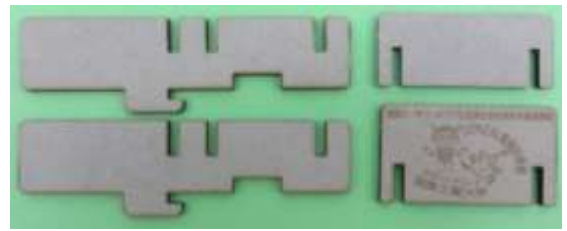
■ 土台にキャスターを取り付ける



裏側をナットで
とめよう

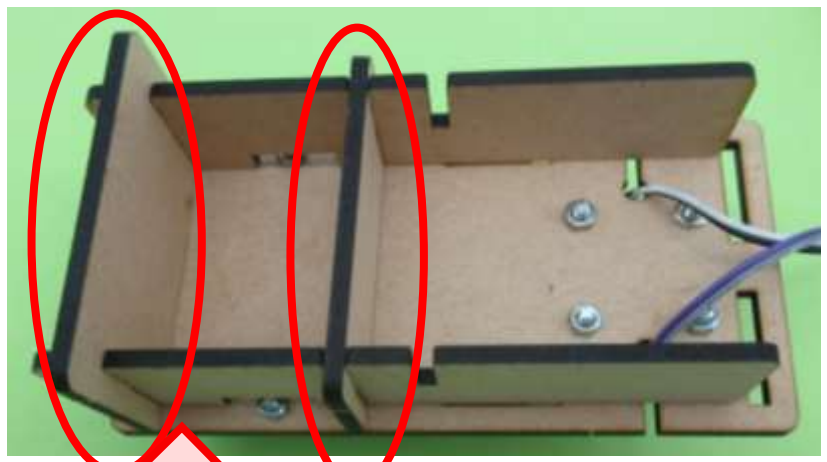


■ 用意するもの

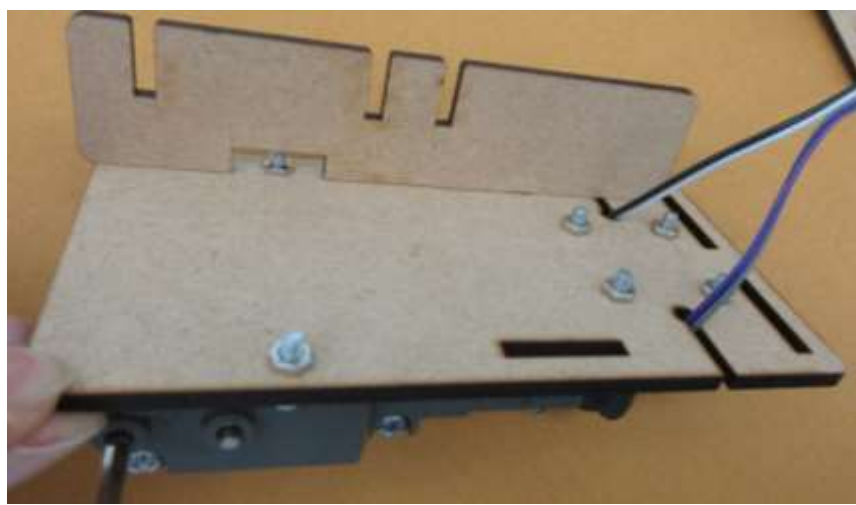


サイド 仕切り小

② 仕切り小をさしこむ

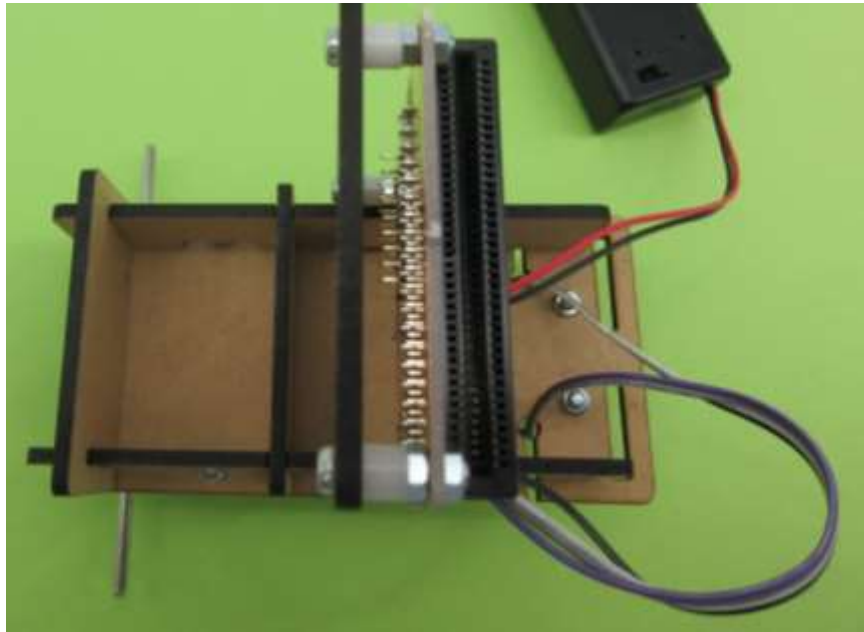


① 土台にさしこみスライドする

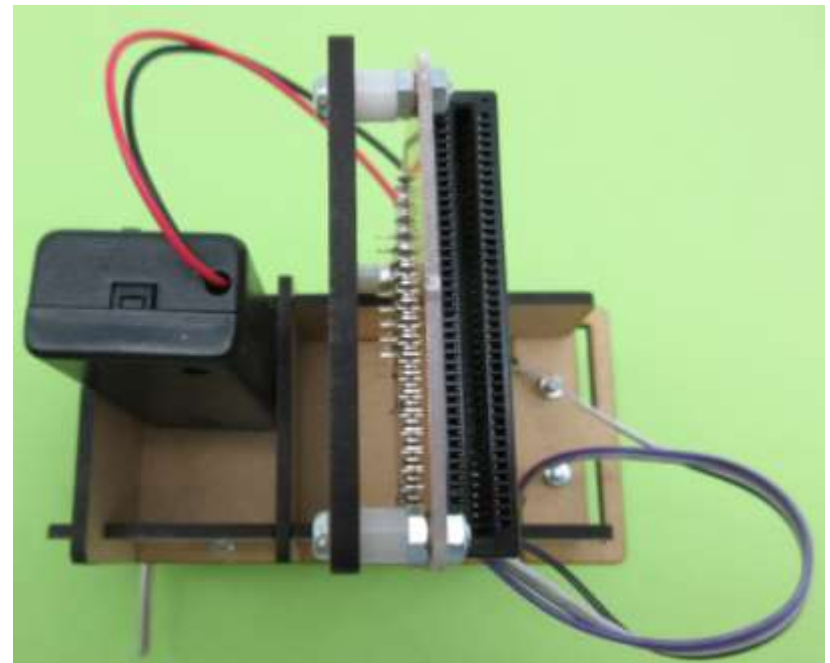




- 組み立てた土台に
基板をさしこむ



- 電池ボックスは左の
スペースにしまう





■ 用意するもの



タイヤ

タイヤを取り付けよう
しっかり奥までさしこもう





- 車体の前の穴にムロピョンを差し込む



コンピュータは
そのままでは動かないよ！

プログラムを書きこんで
ロボットが動くようにしよう

組み立て完成！！



10分



きゅうけい



micro:bit プログラム編





■ 道具の確認

□ パソコン

□ インターネット

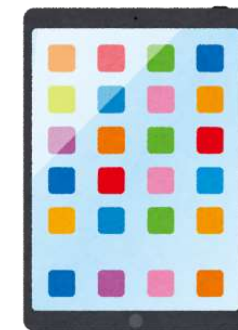
□ Webブラウザ

Google Chrome など

□ micro:bit

□ USBケーブル

□ スマホ／タブレット

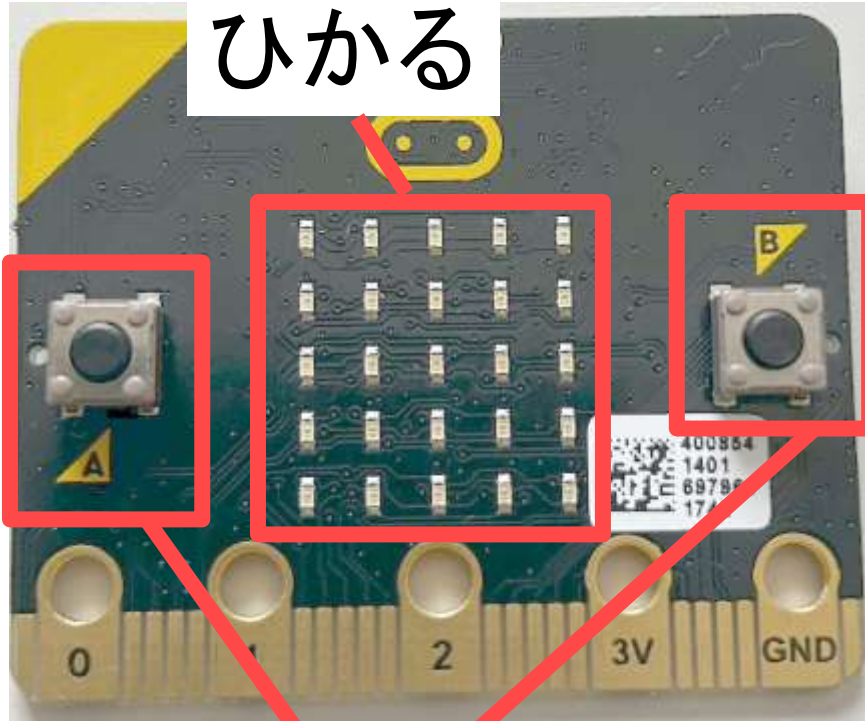




マイクロビット

PC

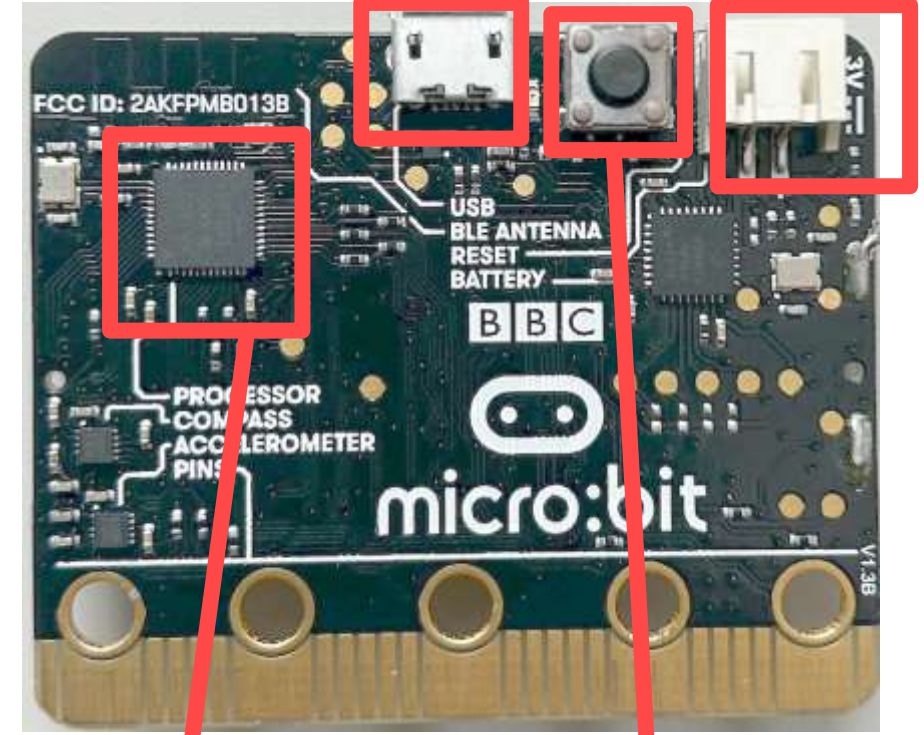
でんち



ひかる



ボタン

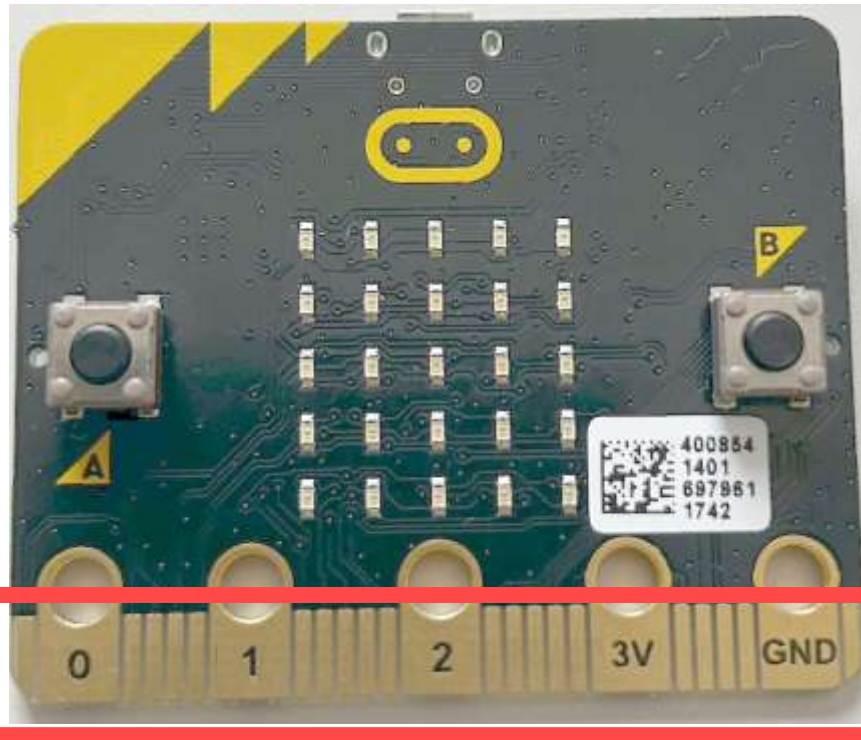


マイコン

リセット



マイクロビット



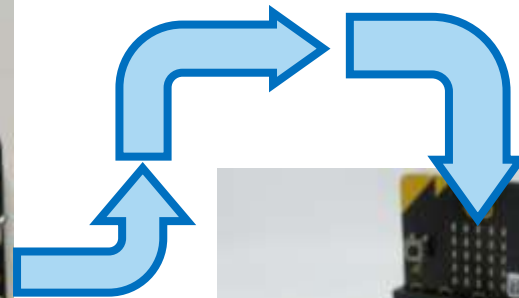
ロボットへ



パソコン



<https://www.irasutoya.com/>より転載



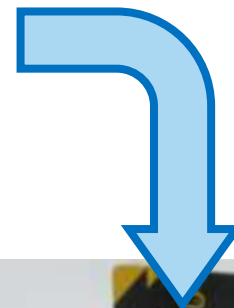
ケーブル: ユーエスビー (USB)



スマホ
タブレット



<https://www.irasutoya.com/>より転載

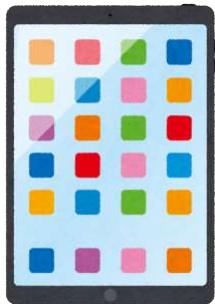


むせん: ブルートゥース (Bluetooth)



<https://www.irasutoya.com/>より転載

メイクコード
MakeCode



<https://www.irasutoya.com/>より転載

マイクロビット
micro:bit



<https://www.apple.com/jp/app-store/>



<https://play.google.com/>



シミュレータ

ぶひん

プログラム



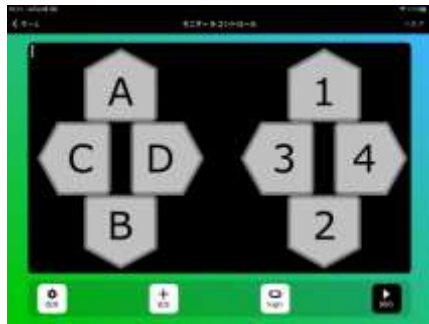
ぶひんを
くみあわせる

なまえ



プログラムのながれ

スマホからロボットをうごかす



スマート
フォン

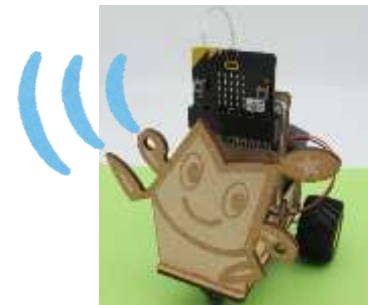


めいれい



はしる

ひかる

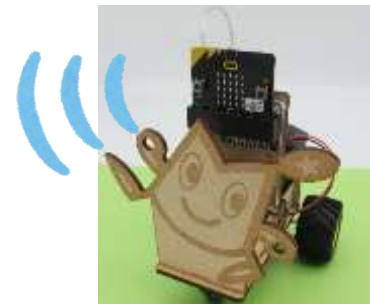
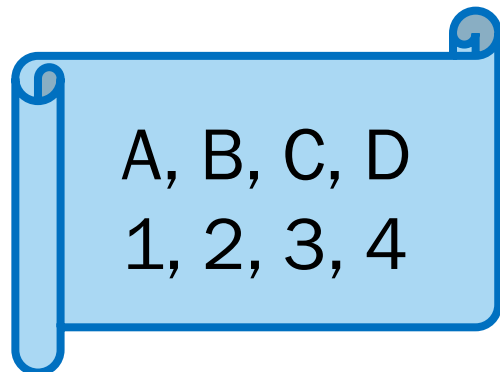


- 「A」 → 「まえに, はしる」
- 「B」 → 「うしろに, はしる」
- ⋮
- 「1」 → 「1 と, ひかる」
- 「2」 → 「2 と, ひかる」
- ⋮



- 【1】 スマホから, めいれいを, うけとる
- 【2】 どのめいれいか, はんだんする
- 【3】 タイヤをまわす, LEDをひからせる

8このめいれい

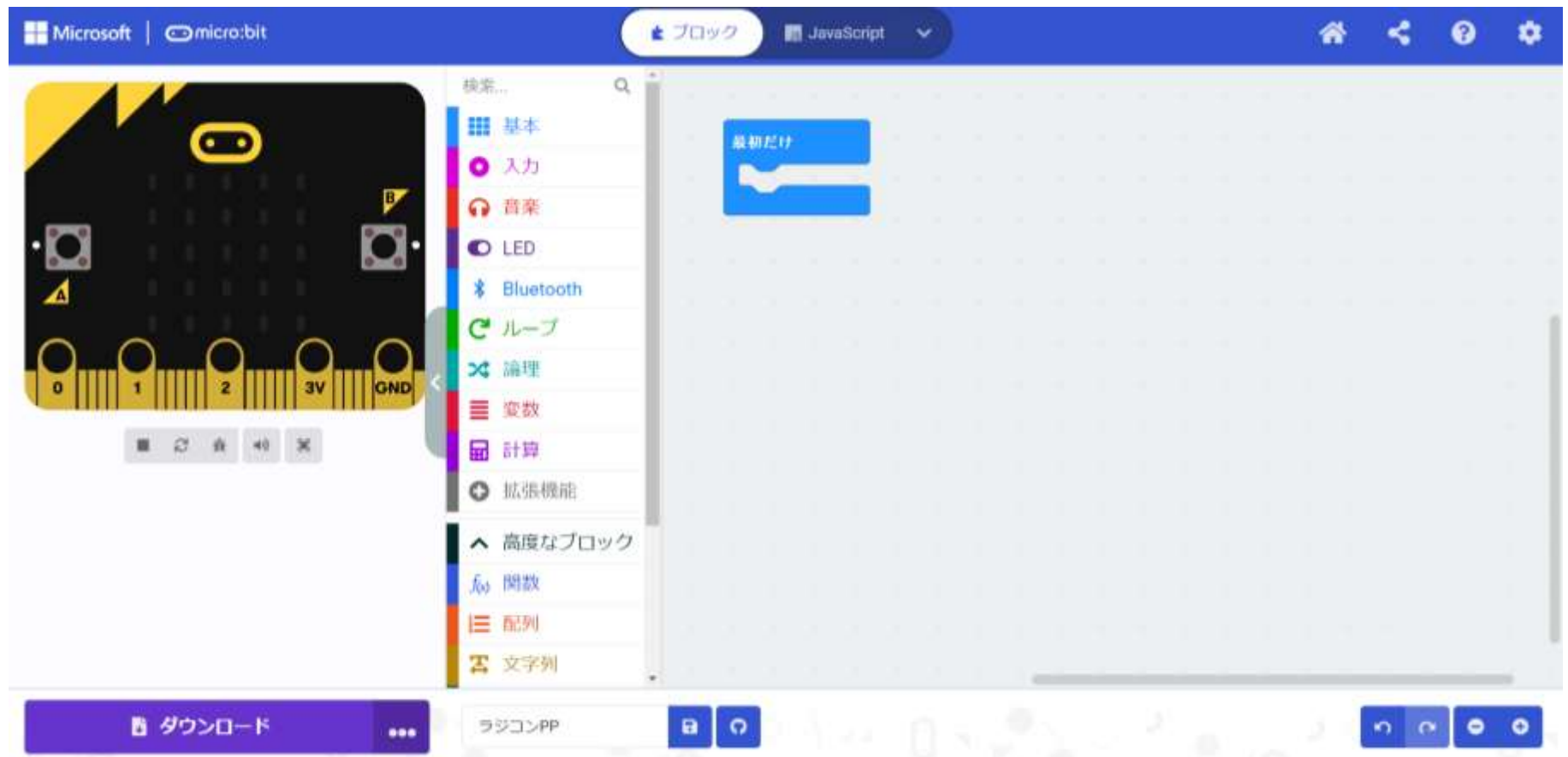


はしる

ひかる



プログラムのソフトをひらく





- 【1】 スマホから, めいれいを, うけとる (6)
- 【2】 どのめいれいか, はんだんする (12)
- 【3】 タイヤをまわす, LEDをひからせる (17)



【1】 スマホから、めいれいを、うけとる (1/6)



1 クリック

2 クリック

③ “last value” と入力。



3 last value

4 クリック



【1】 スマホから、めいれいを、うけとる (2/6)

The image shows a step-by-step guide for adding a variable block to a Scratch script. The background is a screenshot of the Scratch interface. On the left, a sidebar lists various block categories, with '変数' (Variables) highlighted in red. The main workspace shows a 'When green flag clicked' block (blue) containing a '変数 lastValue を 0 にする' (Set variable lastValue to 0) block (red). Three numbered callouts with arrows indicate the steps: 1. 'クリック' (Click) points to the '変数 lastValue を 0 にする' block in the 'Your Variables' section. 2. 'ドラッグ' (Drag) points to the same block being moved from the 'Your Variables' section to the workspace. 3. 'ドロップ' (Drop) points to the block being placed inside the 'When green flag clicked' block.

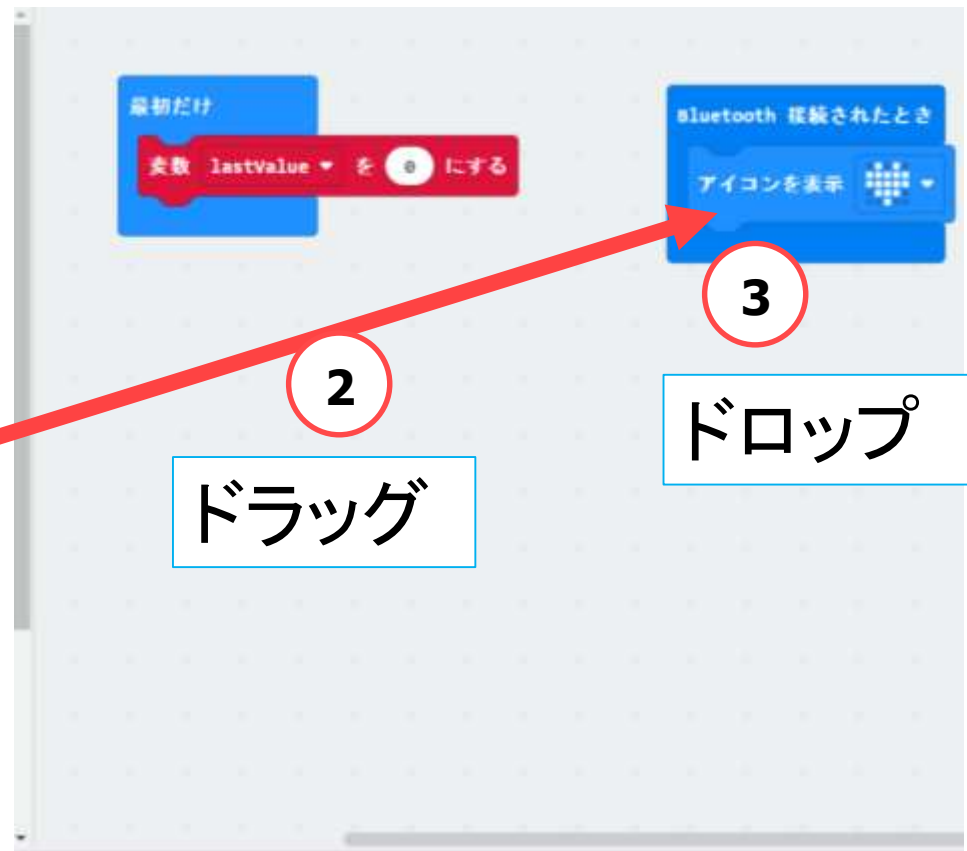
1 クリック

2 ドラッグ

3 ドロップ



【1】 スマホから、めいれいを、うけとる (4/6)





【1】 スマホから、めいれいを、うけとる (5/6)

The image shows a Scratch script area with several blocks: 'バックグラウンドで実行する', '時間 (ミリ秒)', 'リセット', '全体を一時停止 (マイクロ秒)', 'イベントを発生させる', and 'イベントのタイムスタンプ'. A yellow box highlights the 'イベントが届いたとき' block. A red arrow points from this block to the '高度なブロック' category in the left sidebar, labeled '1 クリック'. Another red arrow points from the '高度なブロック' category to the 'イベントが届いたとき' block, labeled '2 クリック'. A third red arrow points from the 'イベントが届いたとき' block to the '最初だけ' block in the script area, labeled '3 ドラッグ'. A fourth red arrow points from the '最初だけ' block to the 'イベントが届いたとき' block, labeled '4 ドロップ'.

1 クリック

2 クリック

3 ドラッグ

4 ドロップ



【1】 スマホから、めいれいを、うけとる (6/6)

イベントが届いたとき 発生源 **MES_DPAD_CONTROLLER_ID** 直 MICROBIT_EVT_ANY ▼

クリック

- MICROBIT_ID_IO_P12
- MICROBIT_ID_IO_P13
- MICROBIT_ID_IO_P14
- MICROBIT_ID_IO_P15
- MICROBIT_ID_IO_P16
- MICROBIT_ID_IO_P19
- MICROBIT_ID_IO_P20
- MES_DEVICE_INFO_ID
- MES_SIGNAL_STRENGTH_ID
- ✓ MES_DPAD_CONTROLLER_ID
- MES_BROADCAST_GENERAL_ID

2 クリック

「MES_DPAD_CONTROLLER_ID」を選択



- 【1】 スマホから, めいれいを, うけとる (6)
- 【2】 どのめいれいか, はんだんする (12)
- 【3】 タイヤをまわす, LEDをひからせる (17)



【2】 どのめいれいか, はんだんする (1/12)

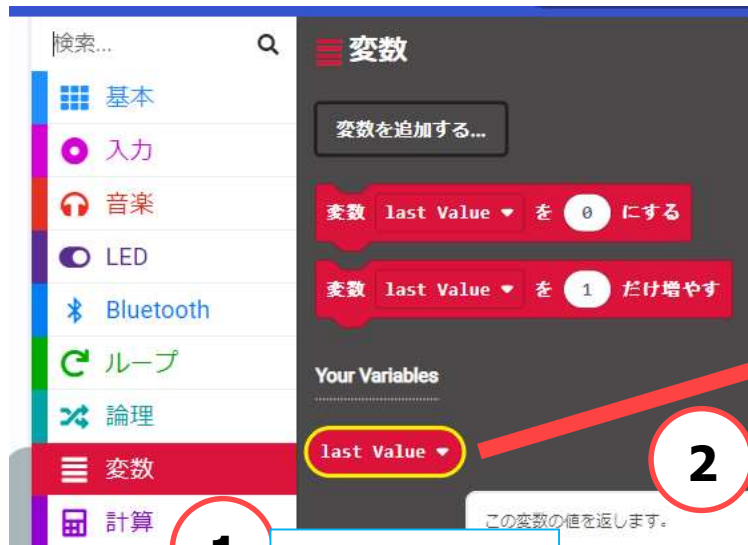
The image shows the Scratch programming environment. On the left is the 'Logic' block palette, and on the right is the workspace. Red circles and arrows indicate the following steps:

- 1 クリック**: Clicking the 'Logic' block in the palette.
- 2 ドラッグ**: Dragging a 'もし 真 なら' (If true) block from the palette into the workspace.
- 3 ドロップ**: Dropping the 'もし 真 なら' block into the workspace.
- 4 ドラッグ**: Dragging a '0 = 0' (0 equals 0) block from the 'Logic' palette into the workspace.
- 5 ドロップ**: Dropping the '0 = 0' block into the workspace, nested under the 'もし 真 なら' block.

The workspace shows the final result: an event handler block 'イベントが届いたとき 発生源 MES_DPAD_CONTROLLER_ID 値 MICROBI' with a nested 'もし 0 = 0 なら' block.



【2】 どのめいれいか, はんだんする (2/12)



1 クリック

2 ドラッグ



ポイント!

ガイド ●
がでた
ところでドロップ





【2】 どのめいれいか, はんだんする (3/12)

1 クリック

2 クリック

3 ドラッグ

⑤ \neq にする

4 ドロップ

5



【2】 どのめいれいか, はんだんする (4/12)

検索...

基本
入力
音楽
LED
Bluetooth
ループ
論理
変数
計算
拡張機能

変数

変数を追加する...

変数 last Value を 0 にする

変数 last Value を 1 だけ増やす

Your Variables

last Value

1 クリック

2 ドラッグ

イベントが届いたとき 発生源 MES_DPAD_CONTROLLER_ID 値

もし last value ≠ イベントの値 なら

変数 last value を 0 にする

3 ドロップ



【2】 どのめいれいか, はんだんする (5/12)

1 みぎクリック

2 クリック

3 コピーができる

4 ドラッグ

5 ドロップ

複製する

コメントを追加する

ブロックを削除する

ヘルプ

変数 last value を 0 にする

変数 last value を イベントの値 にする



【2】 どのめいれいか, はんだんする (6/12)

検索...

- 基本
- 入力
- 音楽
- LED
- Bluetooth
- ループ
- 論理**
- 変数
- 計算
- 拡張機能
- 高度なブロック
- 関数

1 クリック

2 ドラッグ

値が真を動かかたま

イベントが届いたとき 発生源 MES_DPAD_CONTROLLER_ID 値 MICR

もし last value ≠ イベントの値 なら

変数 last value を イベントの値 にする

もし 真 なら

でなければ

3 ドロップ



【2】 どのめいれいか, はんだんする (7/12)

1 クリック

2 ドラッグ

3 ドロップ

基本

- LED
- Bluetooth
- ループ
- 論理
- 変数
- 計算
- 拡張機能

論理

条件判断

もし 真 なら

でなければ

くらべる

0 = 0

0 < 0

0 = 0

イベントが届いたとき 発生源 MES_DPAD_CONTROLLER_ID 値

もし last value ≠ イベントの値 なら

変数 last value を イベントの値 にする

もし 真 なら

0 = 0

でなければ



【2】 どのめいれいか, はんだんする (8/12)

① (5)とおなじほうほうで,
コピーして, ドラッグ&ドロップ



② キーボードから
1 を入れる



【2】 どのめいれいか, はんだんする (9/12)

イベントが届いたとき 発生源 MES_DPAD_CONTROLLER_ID 値

もし last value ≠ イベントの値 なら

変数 last value を イベントの値 にする

もし イベントの値 = 1 なら

でなければ

1

+

+

+

値が真の場合には、最初のかたまりのブロックを動かします。そうでない場合には、二番目のかたまりのブロックを動かします。

＋を1かい
クリックすると

イベントが届いたとき 発生源 MES_DPAD_CONTROLLER_ID 値

もし last value ≠ イベントの値 なら

変数 last value を イベントの値 にする

もし イベントの値 = 1 なら

でなければもし なら

でなければ

+

+

2

「もし～なら」が1つ, 増える



【2】どのめいれいか, はんだんする (10/12)

ポイント!

めいれいは
8こある

A, B, C, D
1, 2, 3, 4



「もし～なら」を8こにする. あと, 6かい, クリックする



【2】 どのめいれいか, はんだんする (11/12)

イベントが届いたとき 発生源 MES_DPAD_CONTROLLER_ID 値 MICROBIT_EV

もし last value イベントの値 なら

変数 last value を イベントの値 にする

もし イベントの値 1

複製する

コメントを追加する

ブロックを削除する

ヘルプ

でなければもし なら

でなければもし なら

でなければもし なら

でなければもし なら

でなければもし なら

でなければもし なら

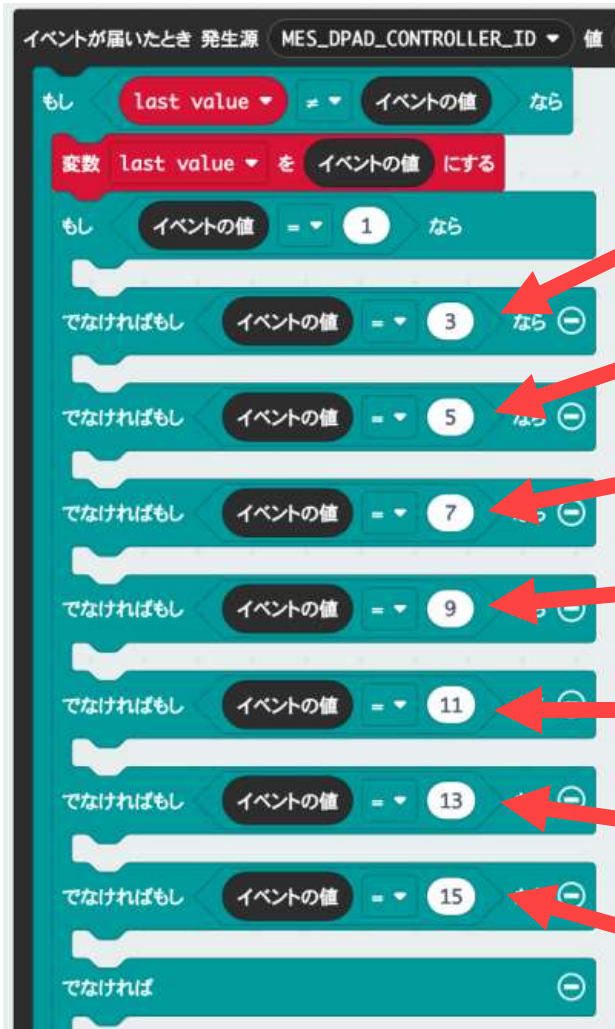
でなければもし なら

でなければ

- 1 みぎクリック
- 2 クリック
- 3 コピーができる
- 4 ドラッグ
- 5 ドロップ
- 6 ぜんぶ, おなじに



【2】 どのめいれいか, はんだんする (12/12)



① すうじをかえる

3 にする

5 にする

7 にする

9 にする

11 にする

13 にする

15 にする



- 【1】 スマホから, めいれいを, うけとる (6)
- 【2】 どのめいれいか, はんだんする (12)
- 【3】 タイヤをまわす, LEDをひからせる (17)



【3】 タイヤをまわす, LEDをひからせる (1/17)

イベントが届いたとき 発生源 MES_DPAD_CONTROLLER_ID 値

もし last value = イベントの値 なら

変数 last value を イベントの値 にする

もし イベントの値 = 1 なら

でなければもし イベントの値 = 3 うしろ

でなければもし イベントの値 = 5 ひだり

でなければもし イベントの値 = 7 なら

でなければもし イベントの値 = 9 なら

でなければもし イベントの値 = 11 なら

でなければもし イベントの値 = 13 なら

まえ

みぎ

タイヤをまわしてすすむ

もし イベントの値 = 1 なら

文字列を表示 "A"

デジタルで出力する 端子 P8 値 1

デジタルで出力する 端子 P1 値 0

デジタルで出力する 端子 P12 値 1

デジタルで出力する 端子 P2 値 0

アナログで出力する 端子 P0 値 300

アナログで出力する 端子 P16 (出力のみ) 値 300

ポイント!
ブロックのかたまりをコピー



【3】 タイヤをまわす, LEDをひからせる (2/17)



2 ドラッグ

3 ドロップ



このなかに, ブロックのかたまりをつくる



【3】 タイヤをまわす, LEDをひからせる (3/17)

The image shows the Scratch programming environment. On the left is the 'Basic' palette with various block categories. A red circle labeled '1' highlights the 'Click' block. A red box labeled '2' highlights a 'Hello!' block on the stage. A red arrow labeled '3' points from the 'Hello!' block to a 'Hello!' block being dropped into a 'When clicked' block on the stage.

1 クリック

2 ドラッグ

3 ドロップ

基本

入力

音楽

LED

Bluetooth

ループ

論理

変数

計算

拡張機能

高度なブロック

数を表示 0

LED画面に表示

アイコンを表示

文字列を表示 "Hello!"

表示を消す

ずっと

もし 真 なら

文字列を表示 "Hello!"



【3】 タイヤをまわす, LEDをひからせる (4/17)

1 クリック

2 クリック

3 ドラッグ

4 ドロップ

5 4かい, くりかえす



【3】 タイヤをまわす, LEDをひからせる (5/17)

1 クリック

2 クリック

3 ドラッグ

4 ドロップ

5 2かい, くりかえす



【3】 タイヤをまわす, LEDをひからせる (6/17)



1 すうじをかえる

300 にする



【3】 タイヤをまわす, LEDをひからせる (7/17)

1 クリック

3 ほかのポートもかえる

もし 真 なら

文字列を表示 "Hello!"

デジタルで出力する 端子 P0 値 0

デジタルで出力する

デジタルで出力する

デジタルで出力する

アナログで出力する

アナログで出力する

P0 P1

P4 P5 P6 P7

P8 P9 P10 P11

P12 P13 P14 P15

P16

P8を
クリック

もし 真 なら

文字列を表示 "Hello!"

デジタルで出力する 端子 P8 値 P8 にする

デジタルで出力する 端子 P1 値 P1 にする

デジタルで出力する 端子 P12 値 P12 にする

デジタルで出力する 端子 P2 値 P2 にする

アナログで出力する 端子 P0 値 P0 にする

アナログで出力する 端子 P16(出力のみ) 値 300

P16 にする



【3】 タイヤをまわす, LEDをひからせる (8/17)

もし 真なら

- 文字列を表示 Hello!
- デジタルで出力する 端子 P8 値 0
- デジタルで出力する 端子 P1 値 0
- デジタルで出力する 端子 P12 値 0
- デジタルで出力する 端子 P2 値 0
- アナログで出力する 端子 P0 値 300
- アナログで出力する 端子 P16(出力のみ) 値 300

複製する

コメントを追加する

ブロックを削除する

ヘルプ

みぎ
クリック

クリック

コピー

3

4かい, くりかえして,
5つに



【3】 タイヤをまわす, LEDをひからせる (9/17)

クリック

1



2 ドラッグ



3 ドロップ





【3】 タイヤをまわす, LEDをひからせる (10/17)

The image consists of three sequential screenshots of the Scratch programming environment, illustrating the steps to add a block to a script:

- 1 クリック**: A yellow box highlights the 'Hello!' block in the 'Text to say' block palette.
- 2 ドラッグ**: A red arrow points from the highlighted block in the palette to the 'Hello!' block being dragged in the script area.
- 3 ドロップ**: A red arrow points from the dragged block to its final position in the script area.

4 「イベントの値」が 1, 3, 5, 7 のところに4つとも

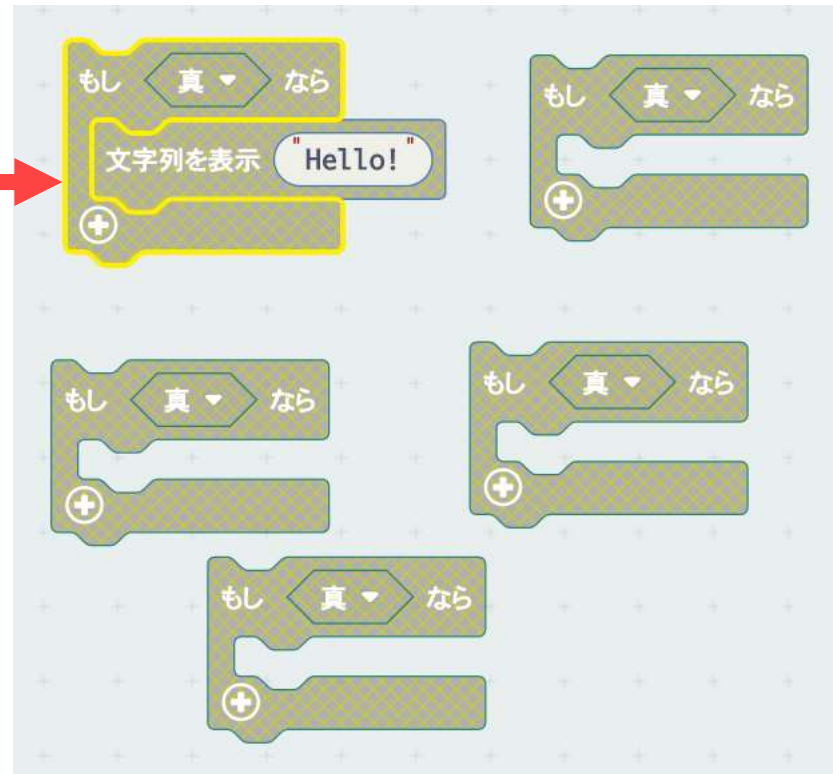


【3】 タイヤをまわす, LEDをひからせる (11/17)

1 クリック

2 del キー

ざんがいを,
さくじょ



3 くりかえして, ほかの4つも, さくじょする



【3】 タイヤをまわす, LEDをひからせる (12/17)

はじめは, とまるので, ぜんぶ 0 にする

もし last value \neq イベントの値 なら

- デジタルで出力する 端子 P8 値 0
- デジタルで出力する 端子 P1 値 0
- デジタルで出力する 端子 P12 値 0
- デジタルで出力する 端子 P2 値 0
- アナログで出力する 端子 P0 値 0
- アナログで出力する 端子 P16(出力のみ) 値 0

変数 last value を イベントの値 にする

P8 → 0

P1 → 0

P12 → 0

P2 → 0

P0 → 0

P16 → 0



【3】 タイヤをまわす, LEDをひからせる (13/17)

イベントの値が 1 (まえ)

赤のところをかえる

もし イベントの値 = 1 なら

文字列を表示 "A"

デジタルで出力する 端子 P8 値 1

デジタルで出力する 端子 P1 値 0

デジタルで出力する 端子 P12 値 1

デジタルで出力する 端子 P2 値 0

アナログで出力する 端子 P0 値 300

アナログで出力する 端子 P16 (出力のみ) 値 300

文字 → A

P8 → 1

P1 → 0

P12 → 1

P2 → 0

P0 → 300

P16 → 300



【3】 タイヤをまわす, LEDをひからせる (14/17)

イベントの値が 3 (うしろ)

赤のところをかえる

でなければもし イベントの値 = 3 なら

文字列を表示 "B"

デジタルで出力する 端子 P8 値 0

デジタルで出力する 端子 P1 値 1

デジタルで出力する 端子 P12 値 0

デジタルで出力する 端子 P2 値 1

アナログで出力する 端子 P0 値 300

アナログで出力する 端子 P16 (出力のみ) 値 300

文字 → B

P8 → 0

P1 → 1

P12 → 0

P2 → 1

P0 → 300

P16 → 300



【3】 タイヤをまわす, LEDをひからせる (15/17)

イベントの値が 5 (ひだり)

赤のところをかえる

でなければもし イベントの値 = 5 なら

文字列を表示 "C"

デジタルで出力する 端子 P8 値 1

デジタルで出力する 端子 P1 値 0

デジタルで出力する 端子 P12 値 0

デジタルで出力する 端子 P2 値 1

アナログで出力する 端子 P0 値 300

アナログで出力する 端子 P16 (出力のみ) 値 300

文字 → C

P8 → 1

P1 → 0

P12 → 0

P2 → 1

P0 → 300

P16 → 300



【3】 タイヤをまわす, LEDをひからせる (16/17)

イベントの値が 7 (みぎ)

赤のところをかえる

でなければもし イベントの値 = 7 なら

文字列を表示 "D"

デジタルで出力する 端子 P8 値 0

デジタルで出力する 端子 P1 値 1

デジタルで出力する 端子 P12 値 1

デジタルで出力する 端子 P2 値 0

アナログで出力する 端子 P0 値 300

アナログで出力する 端子 P16 (出力のみ) 値 300

文字	→	D
P8	→	0
P1	→	1
P12	→	1
P2	→	0
P0	→	300
P16	→	300



【3】 タイヤをまわす, LEDをひからせる (17/17)

The image shows the Scratch programming environment. On the left, the 'Basic' block palette is visible, with the 'Click' block highlighted and circled with a red '1'. A red arrow points from this block to the script area on the right. In the script area, four 'Click' blocks are stacked, each with a 'Show text' block containing a number (1, 2, 3, 4) and a 'Make text read' block with the number circled in red. A red arrow points from the 'Click' block in the palette to the first 'Click' block in the script area. Another red arrow points from the 'Click' block in the palette to the 'Show text' block of the first 'Click' block. A third red arrow points from the 'Click' block in the palette to the 'Make text read' block of the first 'Click' block. A fourth red arrow points from the 'Click' block in the palette to the 'Click' block in the script area. A red circle with the number '3' is placed over the 'Click' block in the palette, and a red arrow points from it to the 'Click' block in the script area. A white box with the text 'ドラッグ&ドロップ' (Drag & Drop) is positioned below the 'Click' block in the palette.

検索... 基本

1 クリック

LED画面に表示

アイコンを表示

文字列を表示 "Hello!"

表示を消す

ドラッグ&ドロップ

でなければもし イベントの値 = 9 なら

文字列を表示 "1" 1 にする

でなければもし イベントの値 = 11 なら

文字列を表示 "2" 2 にする

でなければもし イベントの値 = 13 なら

文字列を表示 "3" 3 にする

でなければもし イベントの値 = 15 なら

文字列を表示 "4" 4 にする

表示を消す



パソコンと micro:bit をケーブルでつなぐ

パソコン



<https://www.irasutoya.com/>より転載

ブラウザが
Google chrome, Edge など
グーグル クローム エッジ



(2) へ

それ以外

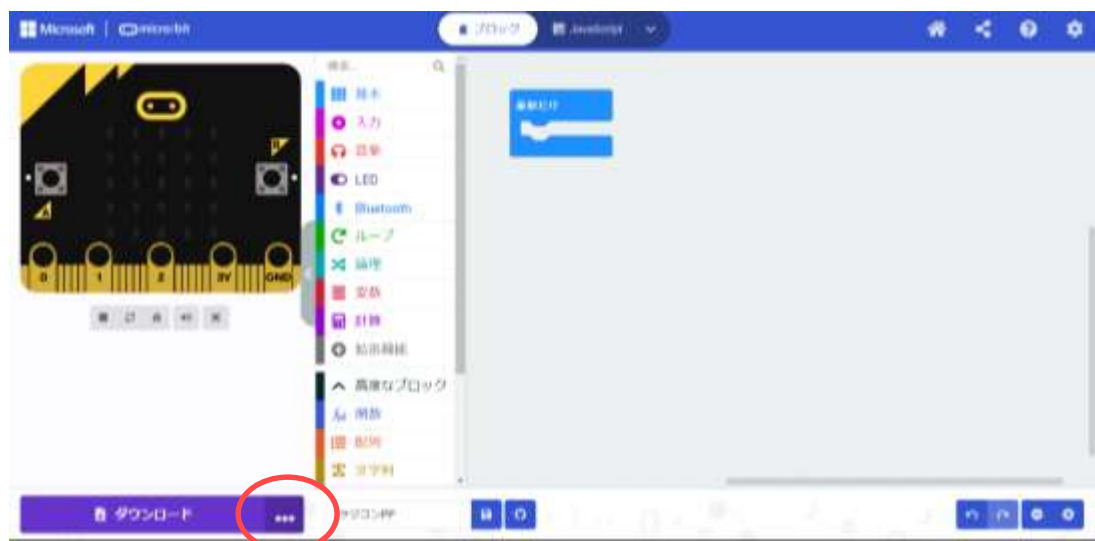


つなげる . (5) へ

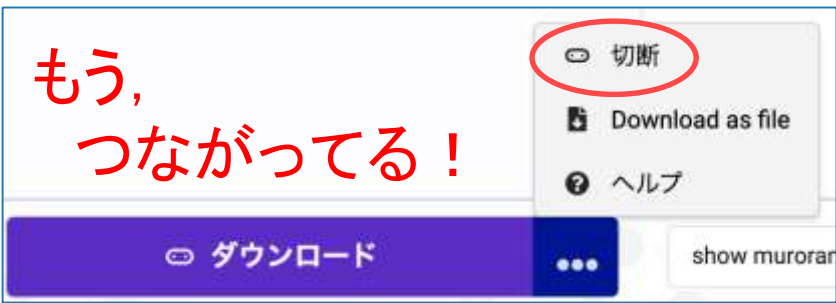
micro:bit ヘブプログラムをおくる(2)



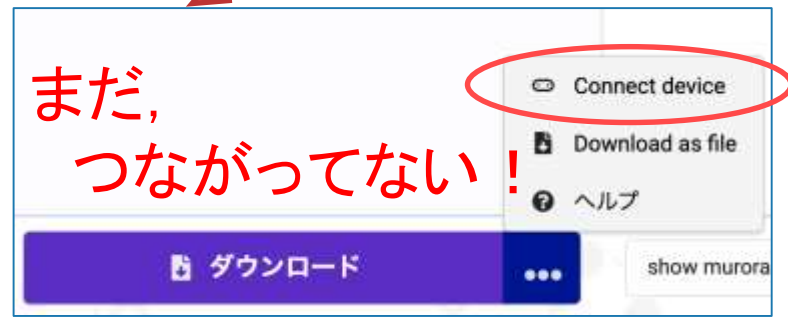
ブラウザと
つながっているか、
たしかめる



クリック



(4)へ



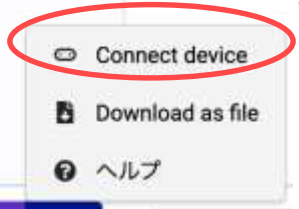
つなげる。(3)へ

micro:bit ヘブプログラムをおくる(3)



ブラウザとつなげる

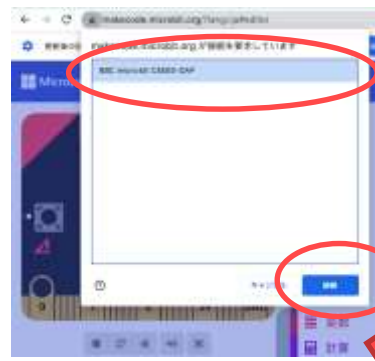
クリック



クリック



クリック

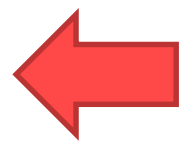


クリックでえらぶ

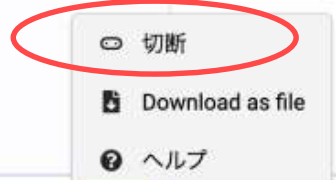
クリック



クリック

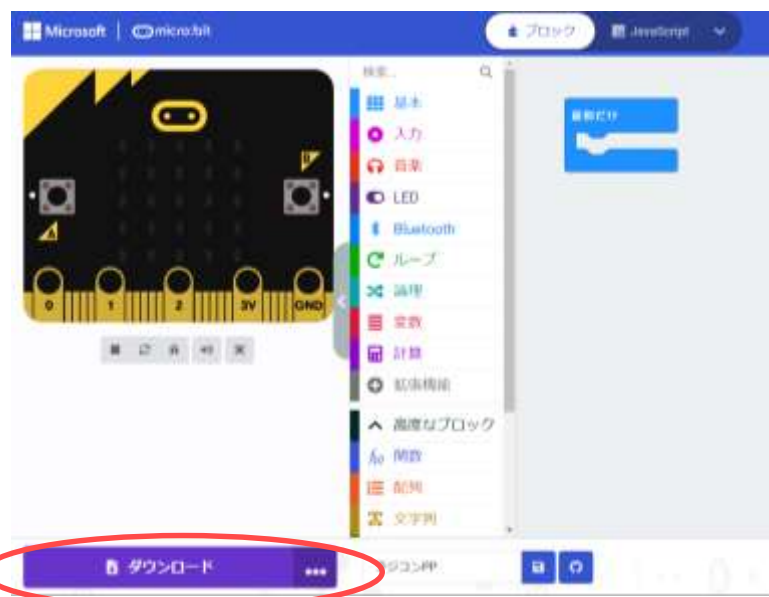


おわったら、
たしかめ！





プログラムをおくる



クリック



<https://www.irasutoya.com/>より転載

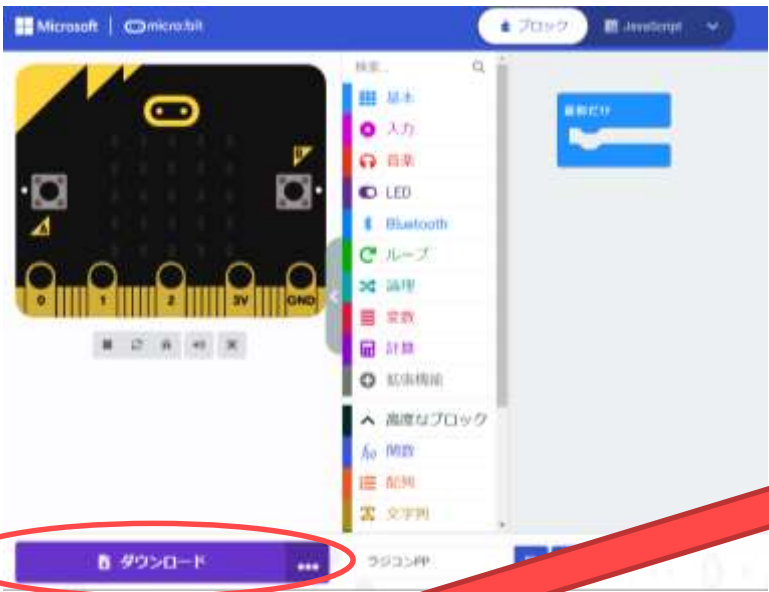
オレンジに
ピカピカ
ひかる

おわり

micro:bit ヘブプログラムをおくる(5)



Google Chrome や Edge などではないとき



クリック



クリック

プログラムはパソコンの中にダウンロードしたよ。
パソコンで micro:bit にコピーしてね



プログラムをコピー

MICROBIT ドライブに
ドラッグ & ドロップ



プログラムのなまえは
microbit- ではじまるよ



<https://www.irasutoya.com/>より転載



オレンジに
ピカピカ
ひかる

おわり

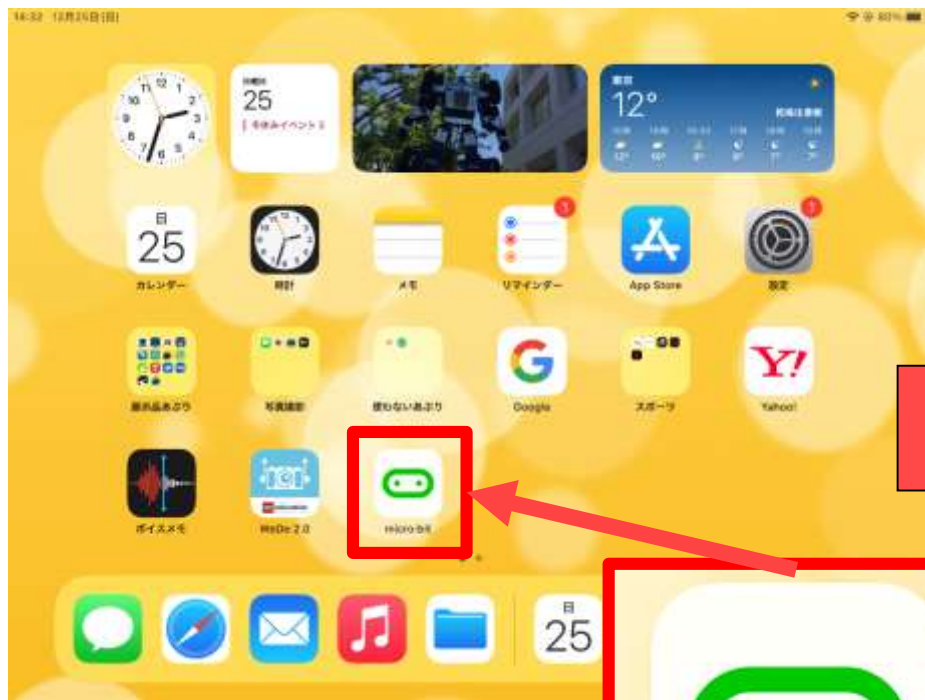


- 設定のアプリからBluetoothの設定がオンであることを確認する



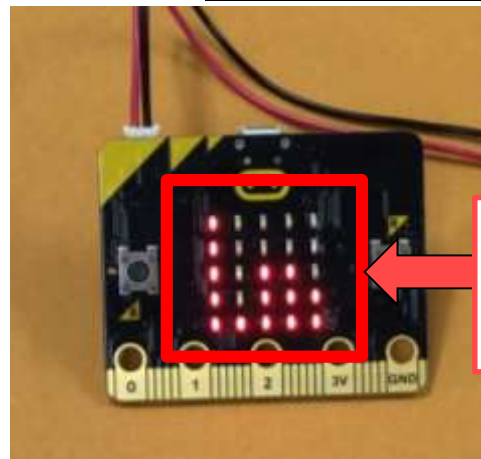
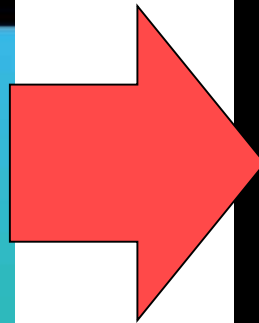


- ホーム画面に戻り、「micro:bit」というアプリを起動する





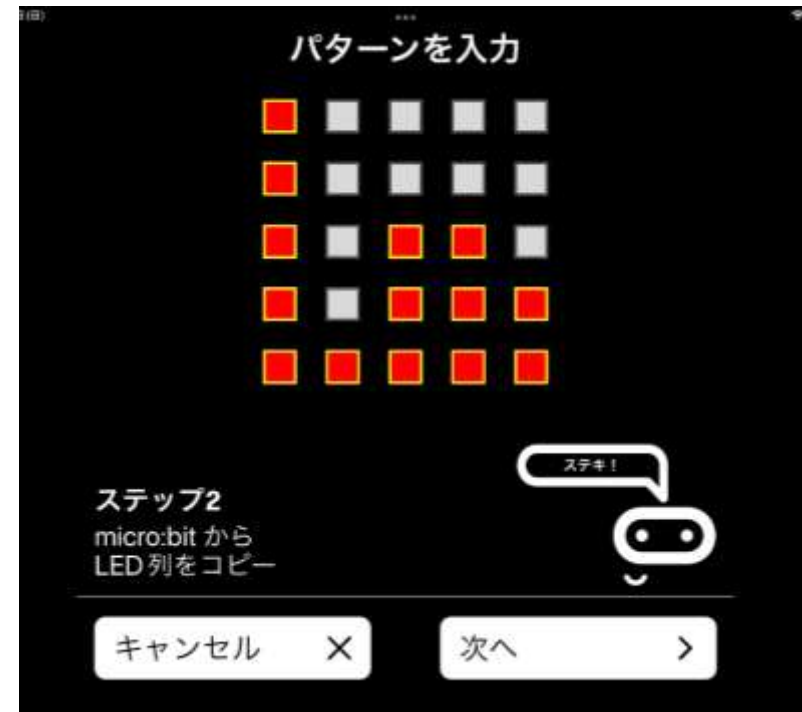
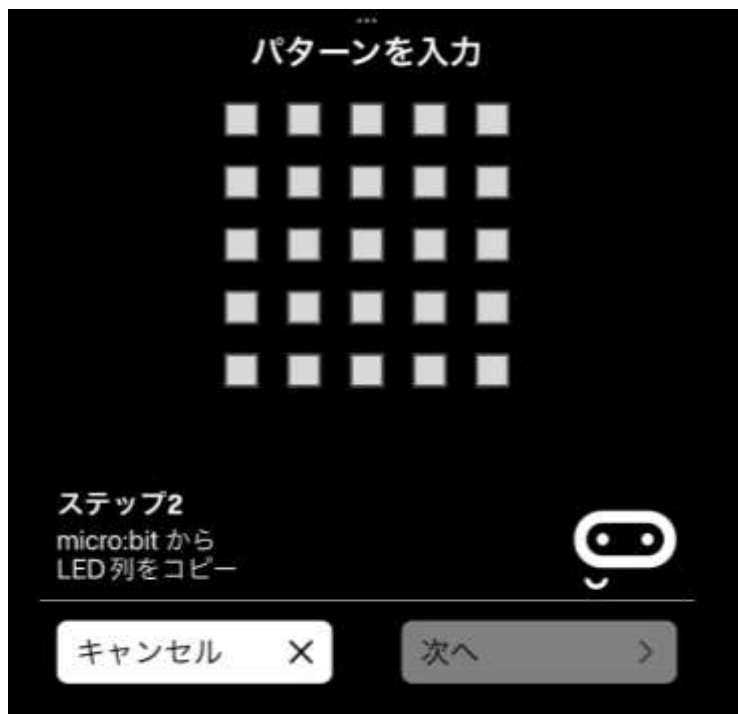
- 「micro:bitを選ぶ」から「新しいmicro:bitをペアリング」をタップして、画面に従ってmicro:bitをペアリングモードにする



microbitにこんなパターンが出たらOK!



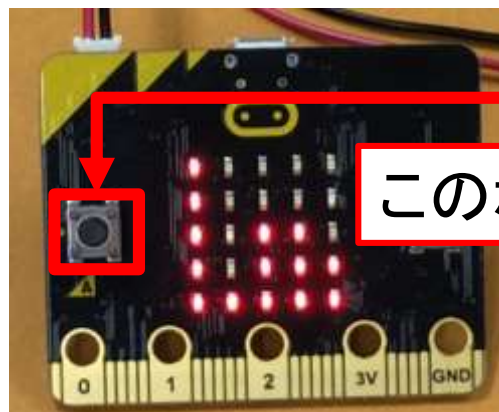
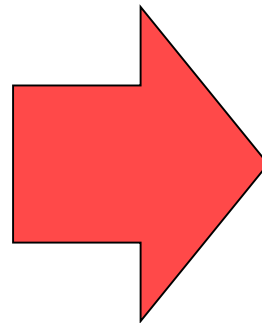
- 「次へ」からパターン入力画面へ移動し、手元の micro:bit の発光パターンを入力する



各列の一番上の光っている場所と同じ場所をタップしよう！



- 「次へ」からペアリング準備完了画面へ行き、micro:bitの左側のボタンを押したら「次へ」をタップ



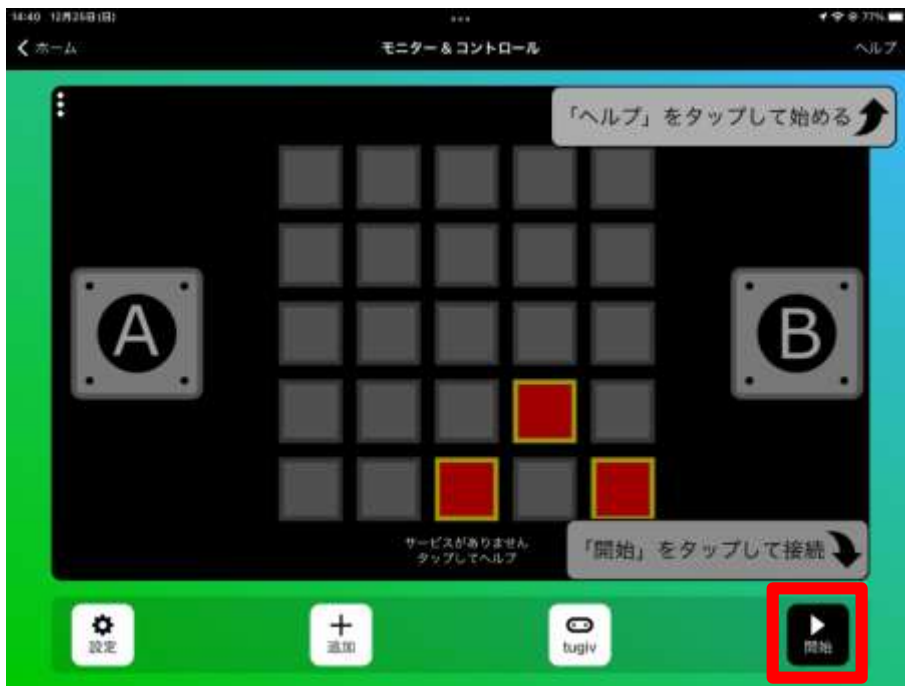
このボタンを押す

端末がこの画面になったら
OK！

※もし繋がらなかったら指導員に聞こう！



- アプリのホーム画面へ戻り、「モニター&コントロール」から開始をタップして、ペアリングしたmicrobitと接続する



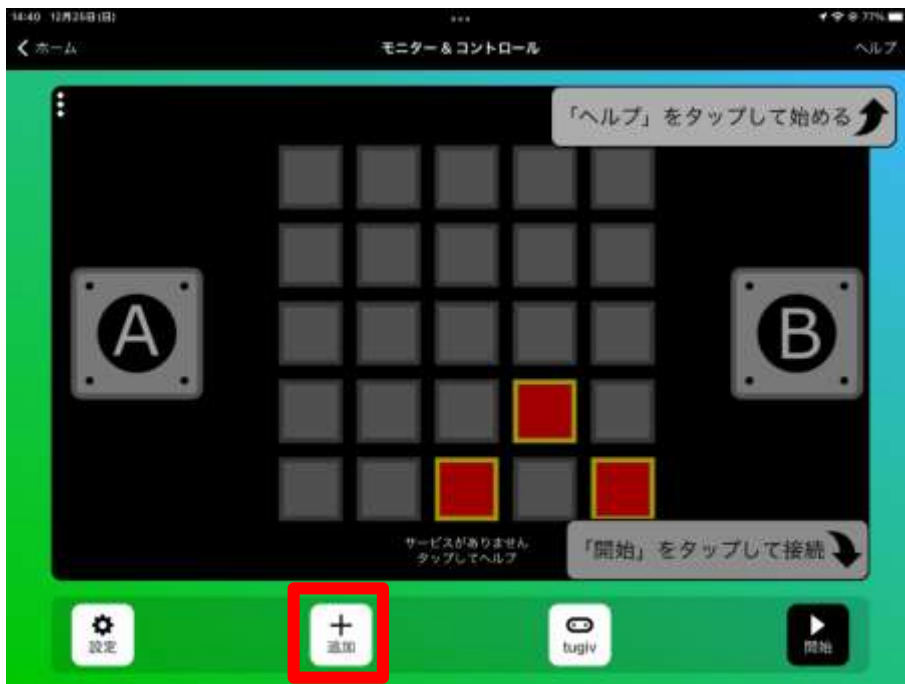
microbitの方もペアリングモードにしておこう！



「成功しました」と表示されたら接続完了



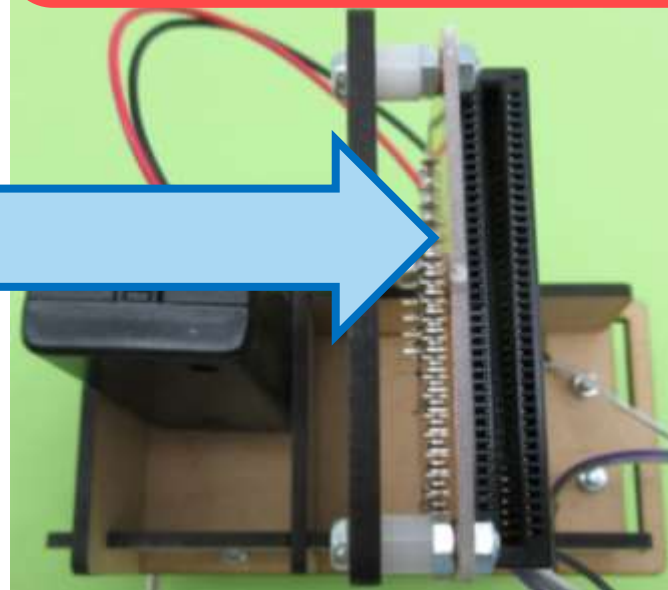
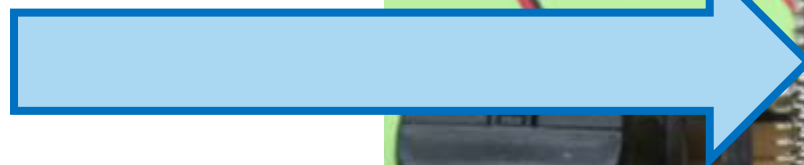
- 「モニター&コントロール」の「+ 追加」で「ゲームパッド」をえらぶ





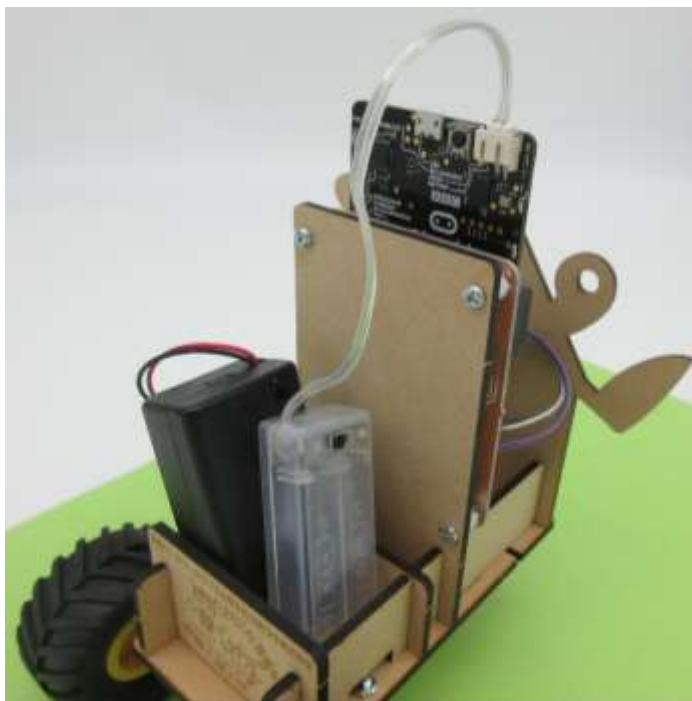
- 黒色のソケットにmicro:bitを差し込む

差し込むのが難しい場合は
指導員に声掛けて下さい。



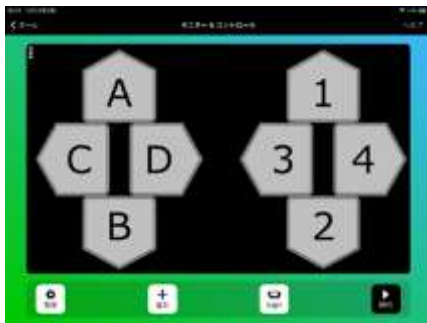


- 電池ボックス端子を接続して、後ろに入れる



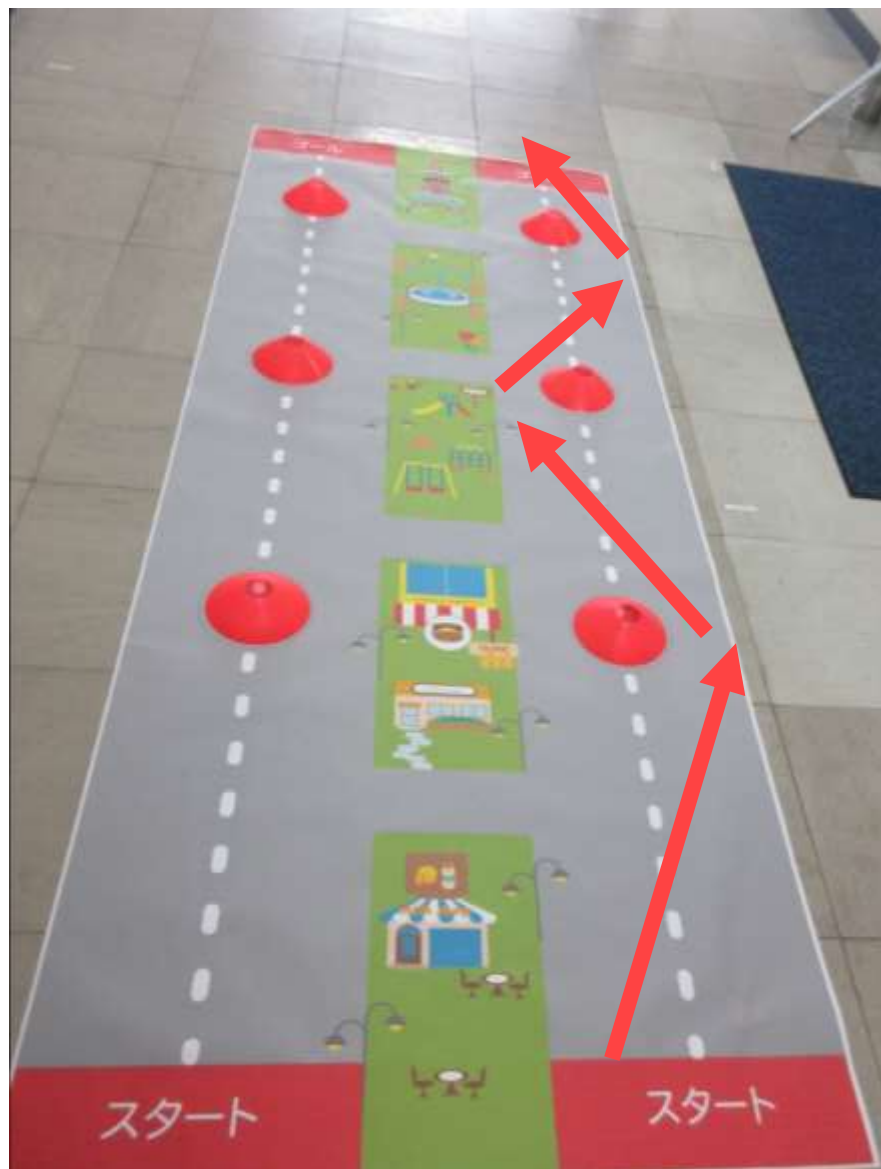
完成！！

じっさいにうごかしてみよう！





チャレンジしてみよう！



自動でギザギザ道を進む
プログラムを考えよう。



■ 真っ直ぐ進むために補正する



左のプログラムを動かして真っ直ぐ進むか確認する。
真っ直ぐ進むようにP0とP16の値を変える。

左 P0 300 — =

右 P16 300 — =



- 秒数と角度の関係を求める。



左のプログラムを動かして
何ミリ秒で角度が何度変わる
か確認する。

100ミリ秒 → °

↓

5° → ミリ秒

45° → ミリ秒

90° → ミリ秒



45° 右に回転

初めは車体が真っ直ぐ
なので秒数を変える

前進

初めは距離が長いので
秒数を変える



くりかえし 2 回

90° 左に回転

3

デジタルで出力する 端子 P8 値 1

デジタルで出力する 端子 P1 値 0

デジタルで出力する 端子 P12 値 0

デジタルで出力する 端子 P2 値 1

アナログで出力する 端子 P16 (出力のみ) 値 300

アナログで出力する 端子 P0 値 300

一時停止 (ミリ秒) 250

デジタルで出力する 端子 P8 値 1

デジタルで出力する 端子 P1 値 0

デジタルで出力する 端子 P12 値 1

デジタルで出力する 端子 P2 値 0

アナログで出力する 端子 P0 値 300

アナログで出力する 端子 P16 (出力のみ) 値 300

一時停止 (ミリ秒) 1600

前進

2

一時停止 (ミリ秒) 1600

デジタルで出力する 端子 P8 値 0

デジタルで出力する 端子 P1 値 1

デジタルで出力する 端子 P12 値 1

デジタルで出力する 端子 P2 値 0

アナログで出力する 端子 P0 値 300

アナログで出力する 端子 P16 (出力のみ) 値 300

一時停止 (ミリ秒) 250

デジタルで出力する 端子 P8 値 1

デジタルで出力する 端子 P1 値 0

デジタルで出力する 端子 P12 値 1

デジタルで出力する 端子 P2 値 0

アナログで出力する 端子 P0 値 300

アナログで出力する 端子 P16 (出力のみ) 値 300

一時停止 (ミリ秒) 1600

90° 右に回転

1

前進

2

繰り返すことでギザギザを進めるよ