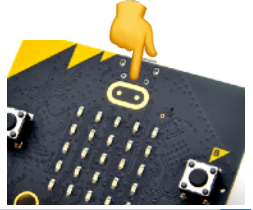




1. 傾けるだけでMaqueenを操縦する！

n:bitの傾きセンサーだけを使って、動きをプログラムします。

「ロゴ」はココ



ずっと

もし **ロゴが下になった** ▼ **動き** **なら**

無線で送信 **"D" = 1** n:bitを向こう側へ倒すと、前へすすむ

でなければもし **左に傾けた** ▼ **動き** **なら** ⊖

無線で送信 **"D" = 2** 左に傾けると、ゆっくり左にまがる

でなければもし **右に傾けた** ▼ **動き** **なら** ⊖

無線で送信 **"D" = 3** 右に傾けると、ゆっくり右にまがる

でなければもし **ロゴが上になった** ▼ **動き** **なら** ⊖

無線で送信 **"D" = 4** 手前に起こすと、後ろへすすむ

でなければもし **画面が上になった** ▼ **動き** **なら** ⊖

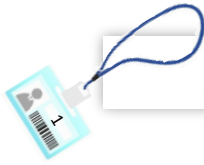
無線で送信 **"D" = 0** なにもしなければ、とまれ

+

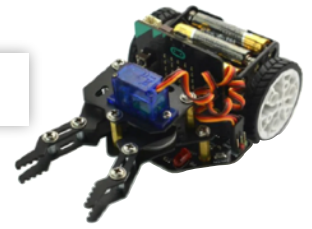
 ゆさぶら...	 ロゴが上...	 ロゴが下...	 画面が上...
 画面が下...	 左に傾けた	 右に傾けた	 落とした
 3G	 6G	 8G	

ボタンだけのプログラムよりも、多くの動きができます。

※「最初だけ」ブロックも必要です
テキストのプログラムと同じものを使います



2. ボタンと傾きでMaqueenを操縦する！



ずっと

もし ボタン A+B ▼ が押されている なら

もし ログが下になった ▼ 動き なら

無線で送信 "D" = 9 ABボタンが押されていて、さらに、n:bitを向こうへ倒すと、最高速で前進する

でなければもし ログが上になった ▼ 動き なら ⊖

無線で送信 "D" = 4 手前に起こすと、後ろへすすむ

でなければ ⊖

無線で送信 "D" = 1 ABボタンを押すだけならゆっくり前にすすむ

でなければもし ボタン A ▼ が押されている なら ⊖

無線で送信 "D" = 2

でなければもし ボタン B ▼ が押されている なら ⊖

無線で送信 "D" = 3

でなければ ⊖

無線で送信 "D" = 0

ボタンと傾きセンサーを組み合わせて、さらに多くの動きをプログラムできます。

ボタンAとBが同時に押されている時に、さらにn:bitを傾けることで、3種類の動きをします。

1. ゆっくり前に進む (n:bitを水平にする)
2. 最高速で前に進む (n:bitを前に倒す)
3. バックする (n:bitを手前に起こす)

コンピュータ (機械) に人間の意思を伝える設計のことを、「マンマシンインタフェース (HMI)」といいます。

人がなにかを操作したときには、機械が人の感覚に合った動きになるようにします。

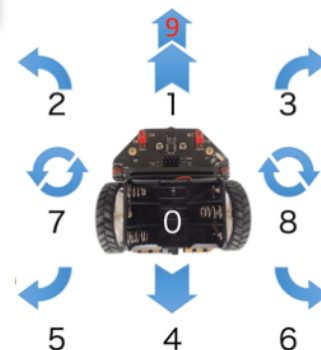
こうして、操作ミスや事故が起きないようにつくり出すことが、プログラミングの技なのです。

※「最初だけ」ブロックも必要です
テキストのプログラムと同じものを使います



3. ボタンと傾きでMaqueenを操縦！

チャレンジ2と同じ動きをするプログラムです。
ブロックの作り方が異なります。
さらに、いまどの方向に動かしているのか、画面のLEDに表示する機能を追加しました。



最初だけ

無線のグループを設定 5
変数 command を 1 にする

ずっと

アイコンを表示 
数を表示 command

画面のLEDに方向を表示します。「ずっと」ブロックをふたつ使うことで、「並列処理」ができます。

ずっと

もし ボタン A+B が押されている なら
無線で送信 "D" = command
でなければもし ボタン A が押されている なら ⊖
無線で送信 "D" = 7
でなければもし ボタン B が押されている なら ⊖
無線で送信 "D" = 8
でなければ ⊖
無線で送信 "D" = 0

画面が上になった とき

変数 command を 1 にする

ロゴが下になった とき

変数 command を 9 にする

ロゴが上になった とき

変数 command を 4 にする



4. 全方向フル操縦！！

チャレンジ3のプログラムを改良して、0から9の動きをすべて行います！

最初だけ

```
無線のグループを設定 5
変数 AB_forward を 1 にする
変数 A_left を 2 にする
変数 B_right を 3 にする
```

ずっと

```
アイコンを表示
数を表示 command
```

ずっと

```
もし ボタン A+B が押されている なら
  変数 command を AB_forward にする
でなければもし ボタン A が押されている なら -
  変数 command を A_left にする
でなければもし ボタン B が押されている なら -
  変数 command を B_right にする
でなければ
  変数 command を 0 にする
無線で送信 "D" = command
```

[無線で送信]ブロックは、1個だけ！

画面が上になった とき

```
変数 AB_forward を 1 にする
変数 A_left を 2 にする
変数 B_right を 3 にする
```

ロゴが下になった とき

```
変数 AB_forward を 9 にする
変数 A_left を 7 にする
変数 B_right を 8 にする
```

ロゴが上になった とき

```
変数 AB_forward を 4 にする
変数 A_left を 5 にする
変数 B_right を 6 にする
```

左に傾けた とき

```
変数 A_left を 7 にする
```

右に傾けた とき

```
変数 B_right を 8 にする
```