

# プログラミング あ・そ・ぼ



ジュニア・プログラミング・ワールド2024

いまをつくる!

未来を創る!

NPO法人 NEXTDAY

TEAM IchigoJam  
ほっかいどう

## Mission CARD in POSSIBLE

micro:bit ロボットカーを  
プログラミングして  
自動運転に挑戦!

2024年11月14,15日 札幌コンベンションセンター

12/14(土) 10:15~ 11:15~ 13:15~ 14:15~

12/15(日) 10:15~ 11:15~ 13:15~ 14:15~

このテキストは  
持ち帰ってね!



わくわく  
ドキドキ

Works with  
micro:bit

V2  
only



DFROBOT®  
DRIVE THE FUTURE




小学生4年生~中学3生 各回10組 40分講座

20241208v6a JPW2024

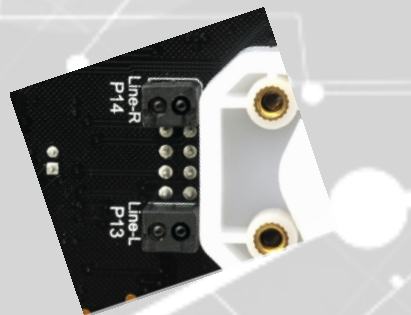
# マイクロマックイーン



ロボットカー・マイクロマックイーンを  
使って、どうろを外れずに走る  
プログラム  を作ります



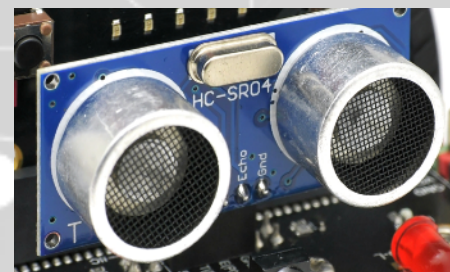
マックイーン



ラインセンサー  
(うら面にある!)



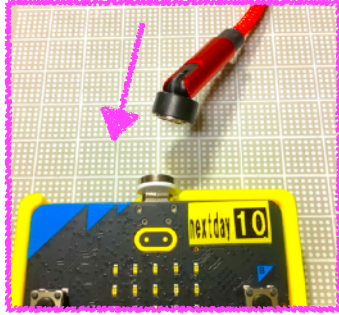
モーター



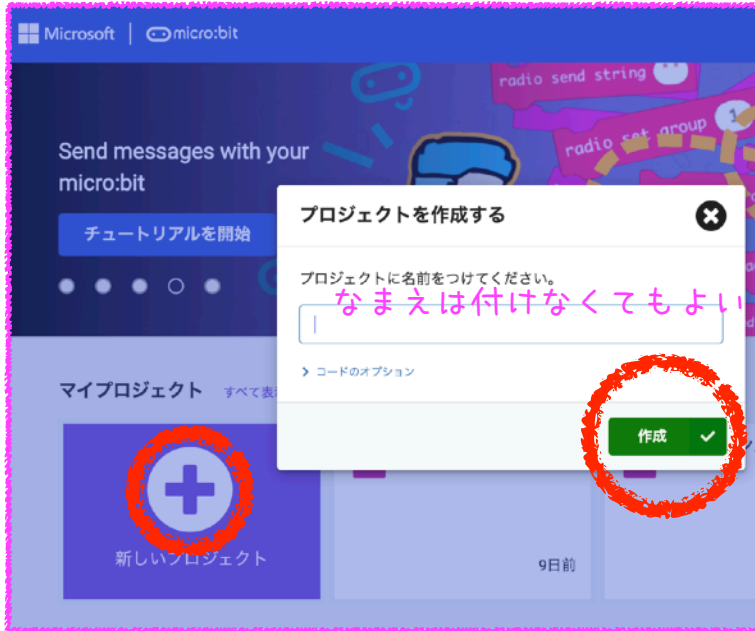
きょりセンサー

※micro:MaqueenはDFRobot社のmicro:bit用プログラミング・ロボットカーで、制御プログラムの学習に必要な、現実環境とPLCの関係がわかりやすく学べる、最適の教材です。  
今回はロボット制御の基本である、ライントレース・ロボットカーを、5つの「ステップ」で走らせます。

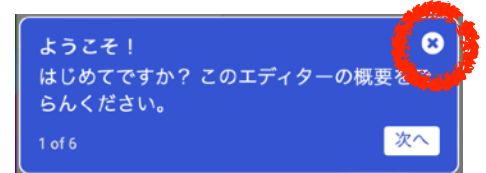
1-1. マックイーンの電源をOFFにして、マイクロビットにケーブルをつなげる



1-2. ⊕ をクリックして、あたらしいプロジェクトをつくる※




もし、「ようこそ！」の  
ポップアップが出たら、  
⊗ をクリックして消す




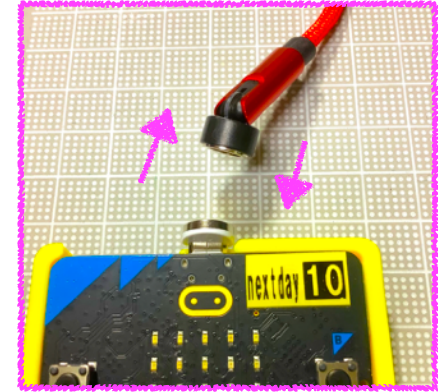
※micro:bitのプログラム(プロジェクト)は、MakeCodeエディターを使って作成します。ここでは必要最小限の作業を説明しています。詳細については、BBC公式ホームページのクイックスタートを参照してください。 <https://archive.microbit.org/ja/guide/quick/>




1-3. ロゴがになっているかな？※



もし、ロゴが表示されていなかったり、  
ロゴがのままなら、ケーブルをつなぎなおそう！



1-4. ロゴのてんめつが **あわって** から、  
「**ダウンロード**」をクリックする

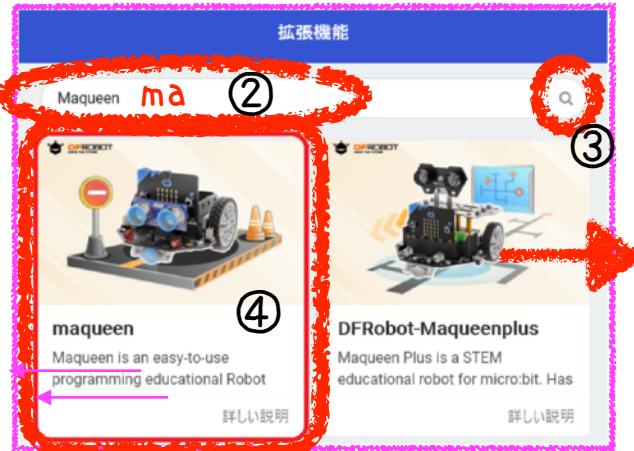
マイクロビットにプログラムをてんそうしよう！



1-5. プログラムがうごきだす  
でも、まだプログラムを  
作っていないので、ここ  
では何もおきない  
つぎのページでプログラム  
を作ろう！！



2-1. マックイーン（かくちょうきのう）の拡張機能（かくちょうきのう）ブロックをよういする。maでけんさくする



2-2. 変数（へんすう）をよういする。なまえは、douroにする



検索...



基本

入力

音楽

LED

IR

Maqueen

無線

ループ

論理

変数

計算

拡張機能

高度なブロック

左 のモーターを 前 へ 速さ 0 で回す

左 のモーターを止める

左 のラインセンサーの値

もし 真 なら

でなければ くらべる

+

変数 douro を 0 にする

douro

0

+

0

0

X

0

今回使うブロックを用意する

3-1. 左の絵とおなじブロックをツールボックスからさがす

3-2. 見つけたブロックをクリックする

3-3. プログラミングエリアにブロックがあらわれる

＋をクリックすると、ブロックがのびるヨ！



3-4. 今回使うブロックをがめんにぜんぶならべよう

The block palette contains the following items:

- Control:**
  - 最初だけ (Initially)
  - ずっと (Always)
- Sensors:**
  - 左 のラインセンサーの値 (Left line sensor value)
  - 右 のラインセンサーの値 (Right line sensor value)
- Motor Control:**
  - 両方 のモーターを 前 へ 速さ 0 で回す (Set both motors to speed 0)
  - 右 のモーターを止める (Stop right motor)
  - 左 のモーターを止める (Stop left motor)
  - 両方 のモーターを止める (Stop both motors)
- Logic:**
  - もし 真 なら (If true)
  - でなければもし 偽 なら (If false)
  - でなければもし 偽 なら (If false)
  - でなければ (If false)
- Variables:**
  - 変数 douro を 0 にする (Set variable douro to 0)
  - 0 + 0 (Addition)
  - 0 × 2 (Multiplication)
- Other:**
  - 0 = 1 (Equality)
  - 0 = 2 (Equality)
  - 0 = 0 (Equality)
  - douro (Variable dropdown)
  - douro (Variable dropdown)
  - douro (Variable dropdown)

※実際の画面ではブロックを置いただけでは色はありませんが、ブロックを繋げると色がつきます



4-1. ブロックをつなげて、プログラムをかんせいさせよう

ずっと ①

つぎのページにブロックのつなげ方が書いてあるよ

② 変数 **douro** を **左** のラインセンサーの値 **+** **右** のラインセンサーの値 **×** **2** にする

③ もし **douro** **=** **0** なら

④ **両方** のモーターを **前** へ 速度 **40** で回す

でなければもし **douro** **=** **1** なら **-**

⑤ **右** のモーターを止める

でなければもし **douro** **=** **2** なら **-**

⑥ **左** のモーターを止める

でなければ **-**

⑦ **両方** のモーターを止める

+

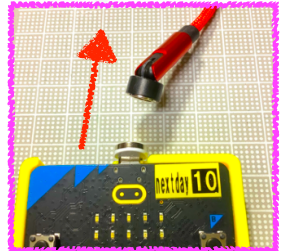
A. プログラムがかんせいしたら  
マックイーンのおしりスイッチ  
がOFFであることをたしかめる  
マイクロビットのUSBケーブル  
は**つなげたまま**



B. プログラムを、**ダウンロード** (転送) する



C. USBケーブルをマイクロビット  
からは**はずす**

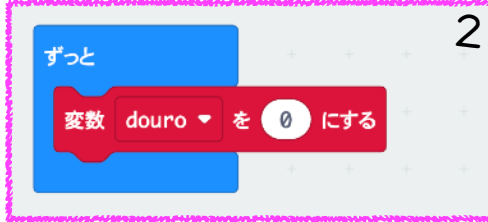


D. では **すてっぷ 5** で走らせよう

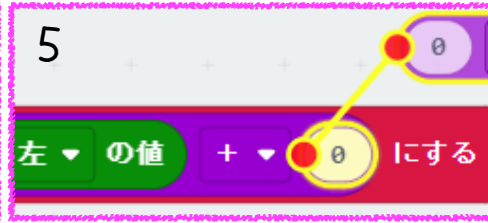
の**数**を変えて、はしる速さをかえてみよう！

このページはブロックのつなげ方が書いてある プログラムが完成してれば、ステップ5にいこう

## 4-2. ブロックをどうしを近づけると、じどうでつながる (プログラムが作られる)



赤丸と黄色線でつなげる目じるしがあるので、じょうずにブロックをつなげる



ひとつ、かんせい



## 4-3. どんどんブロックをつなげて、プログラムをかんせいさせよう

計算式を作るときは、四則演算の優先順序に注意しましょう。  
ブロックの式はわかりづらいので、メニューから[Python]言語などに  
切り替えて確認すると、計算式の間違ひがありません。

```
douro = maqueen.read_patrol(maqueen.Patrol.PATROL_LEFT) + maqueen.read_patrol(maqueen.Patrol.PATROL_RIGHT) * 2
```



# Run Maqueen!

GoGo!

- 5-1. コースに**乗せる**!
- 5-2. おしりのスイッチを**ON**!
- 5-3. GoGo**スタート**!



うまく動いたな?

かいりょうの必要はないかな?

 思うように動かないときは 

- でんげんは**ON**かな? **でん**ちはあるかな?
  - micro:bitが、Maqueenから**抜けて**いないかな?
  - プログラムは、**すてっぷ4**のとおりだよな?
  - ちゃんと**ダウンロード**できたかな?
- **すてっぷ1**の1-3.をみよう。💬を忘れていないかな?

もういちどダウンロードして、**再チャレンジ**しよう 10



マックイーンのしくみ  
*Honw Maqueen works*

⚙️ **黒せんを走るしくみ** ⚙️

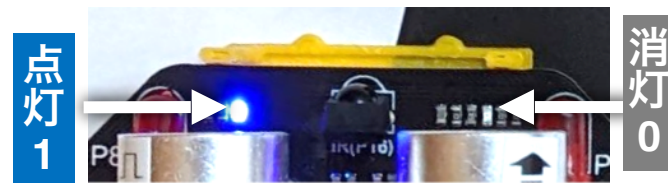
Maqueenは、車体裏面のラインセンサーを使い、赤外線反射を利用して道路を見ます。その様子は、Maqueenの表面にある青色LEDでもわかります（右図）。プログラムでは計算式によって、道路の方向を判断しています。**douro** = ラインセンサー左 + (ラインセンサー右 × 2)

ラインセンサー 左 ▾ の値

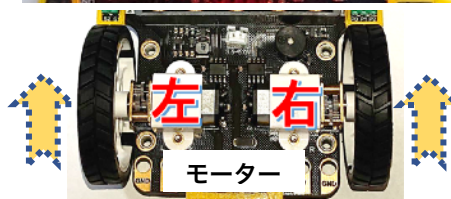
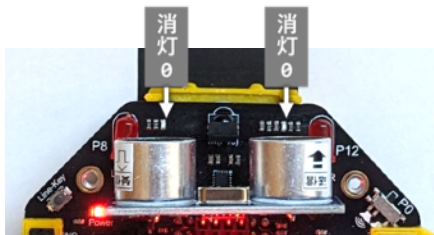
ラインセンサー 右 ▾ の値

(白の上：1)

(黒線の上：0)



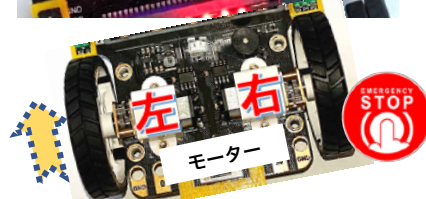
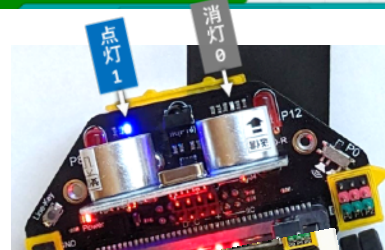
もし **douro** = 0 なら  
 ④ すべてのモーターを 前へ 進さ 60 で回す



③もし、道路のじょうたいが0(黒：直線)なら

④すべてのモーターを前へ回す  
 直進して前へすすむ

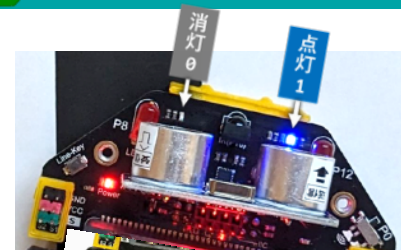
でなければもし **douro** = 1 なら  
 ⑤ 右のモーターを止める



③でなければもし、道路のじょうたいが1  
 (左にはずれた)なら

⑤右モーターを止める  
 左モーターは回ったままなので、  
 右方向前へすすむ

でなければもし **douro** = 2 なら  
 ⑥ 左のモーターを止める



③でなければもし、道路のじょうたいが2  
 (右にはずれた)なら

⑥左モーターを止める  
 右モーターは回ったままなので、  
 左方向前へすすむ

