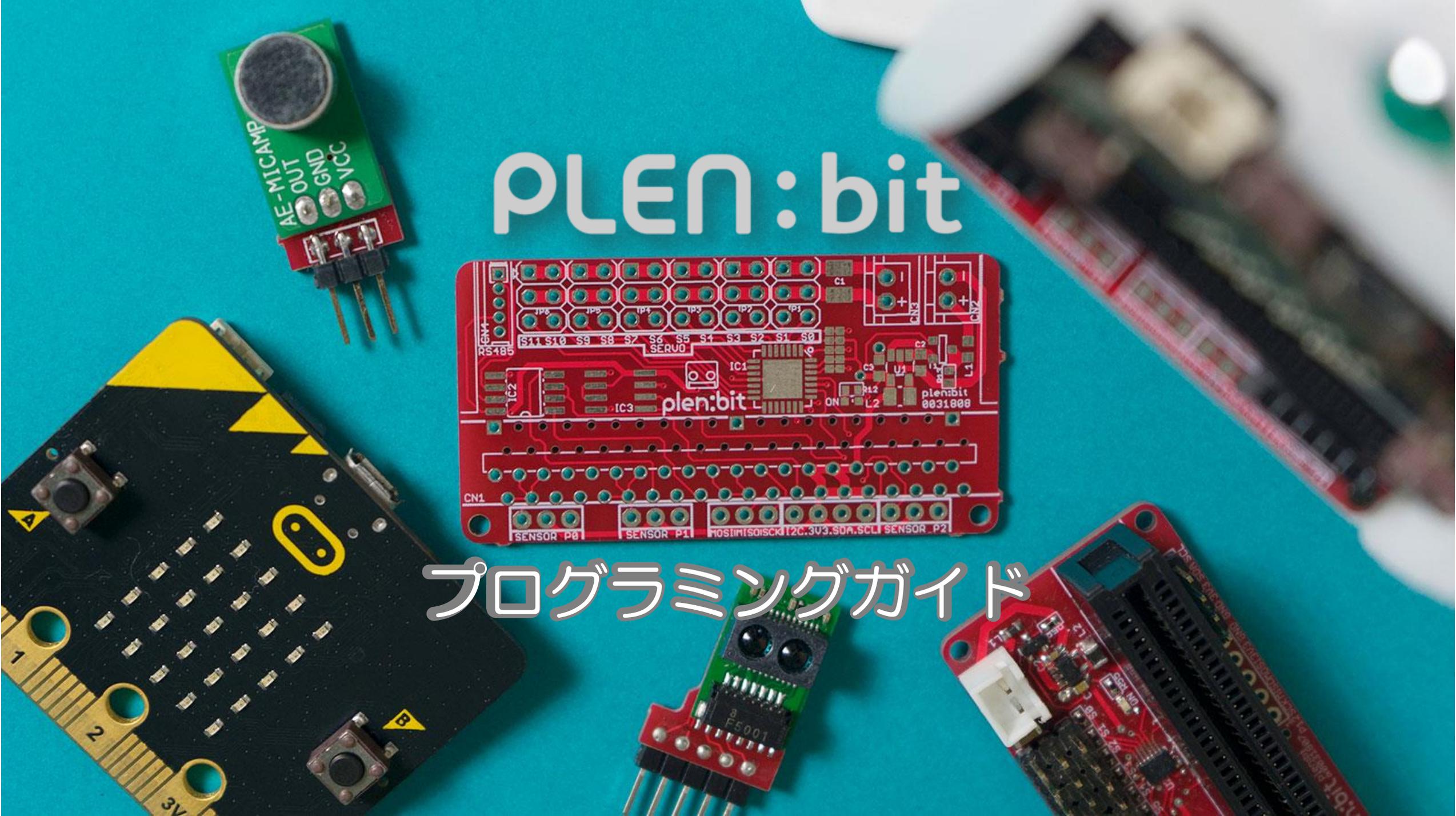


# PLEN:bit

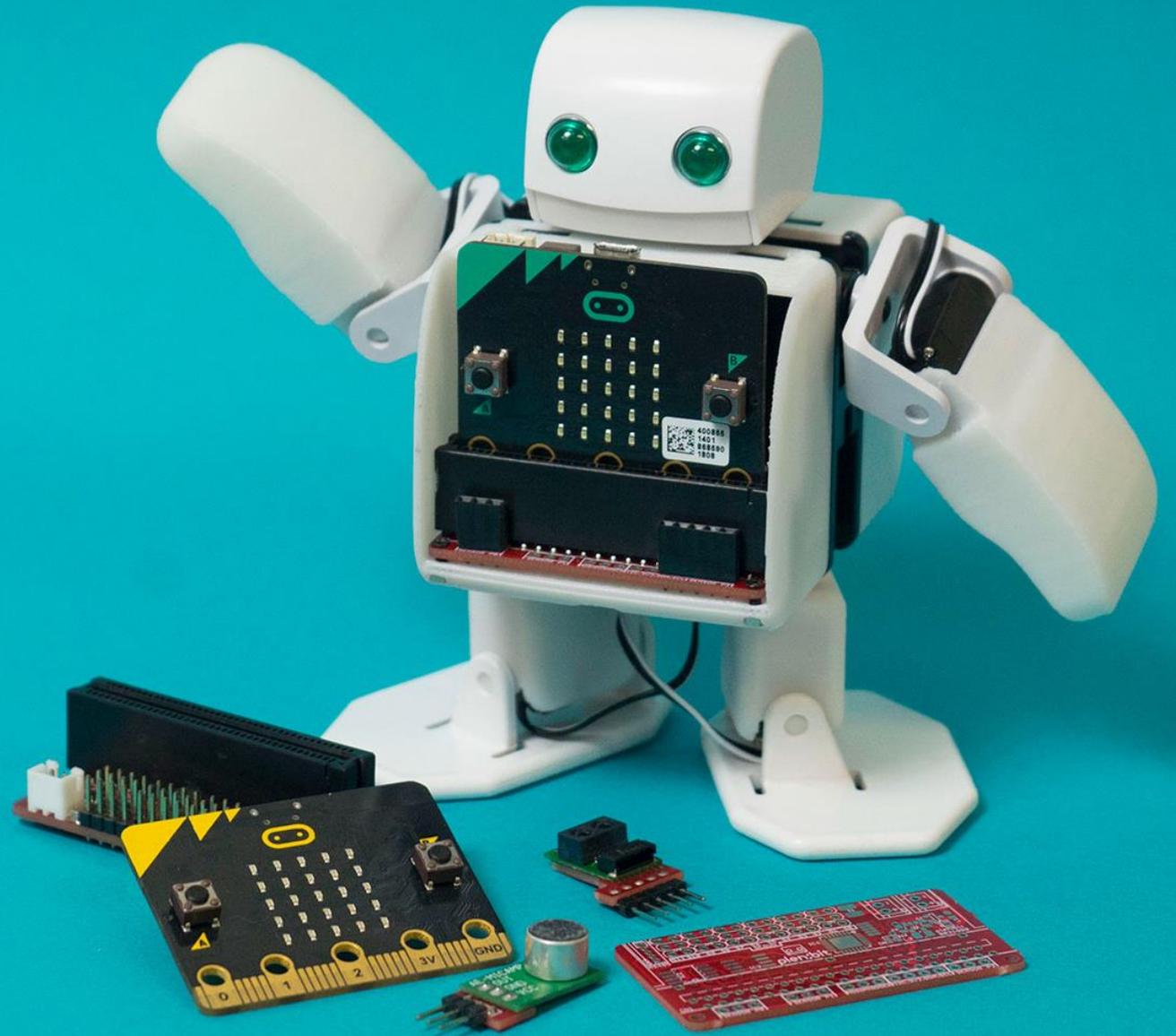
## プログラミングガイド



# PLEN:bit とは？

PLEN:bit は、**micro:bit** と  
**二足歩行ロボット**を組合わせた  
**教育向けロボット**です。

専門的なロボットの知識が  
無くても動かすことができ、  
教育者も理解しやすく教えやすい  
**ブロックエディタ** (makecode) を  
使ってプログラミングできます。



# PLEN:bit の使い方

## 目次

- ・プログラムの作成から転送
- ・ **PLEN:bit** のブロック一覧
- ・ **PLEN:bit** のプログラミング
- ・ **PLEN:bit** のプログラミング  
-SPKIT用
- ・ **PLEN:bit** のHELPコーナー
- ・ その他

**PLEN:bit** を動かすにはプログラミングが必要です。

プログラムを作成し、micro:bit に転送することにより **PLEN:bit** が動作します。

プログラムの作成にはmakecodeを使います。

※推奨環境

OS: **Windows10, macOS**

ブラウザ: **Google Chrome, Microsoft Edge**

※これ以外の環境では、本資料の手順でプログラムを転送できない可能性があります。

下記サイトもご参考ください

<https://microbit.org/ja/guide/quick/>

# PLEN:bit の使い方 ～プログラムの作成から転送～

① makecodeへ  
アクセスする

② PLEN:bit専用  
ブロックを追加

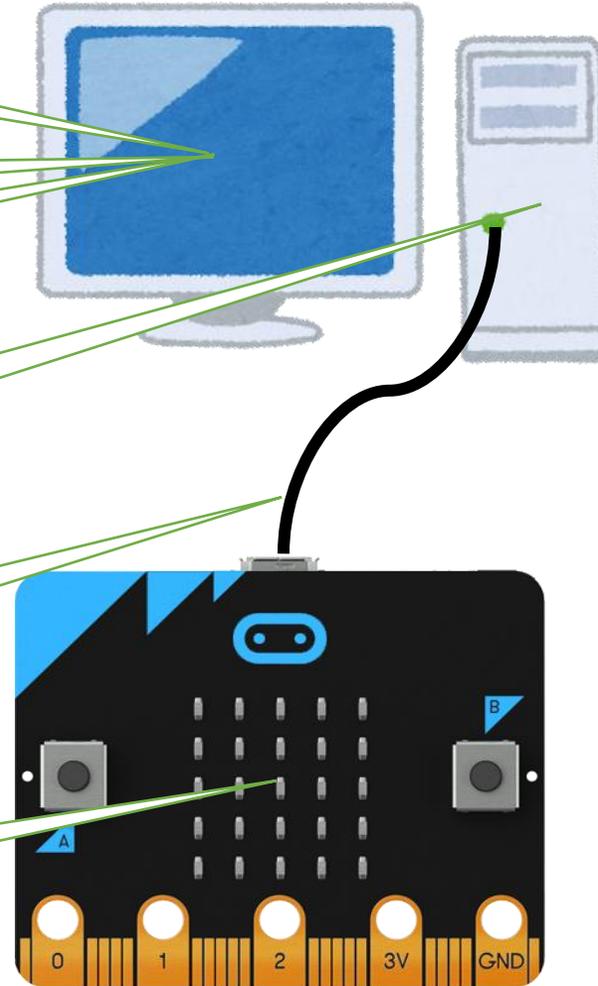
③ makecodeで  
プログラムを作成

④ 作成したプログラムをダ  
ウンロードする

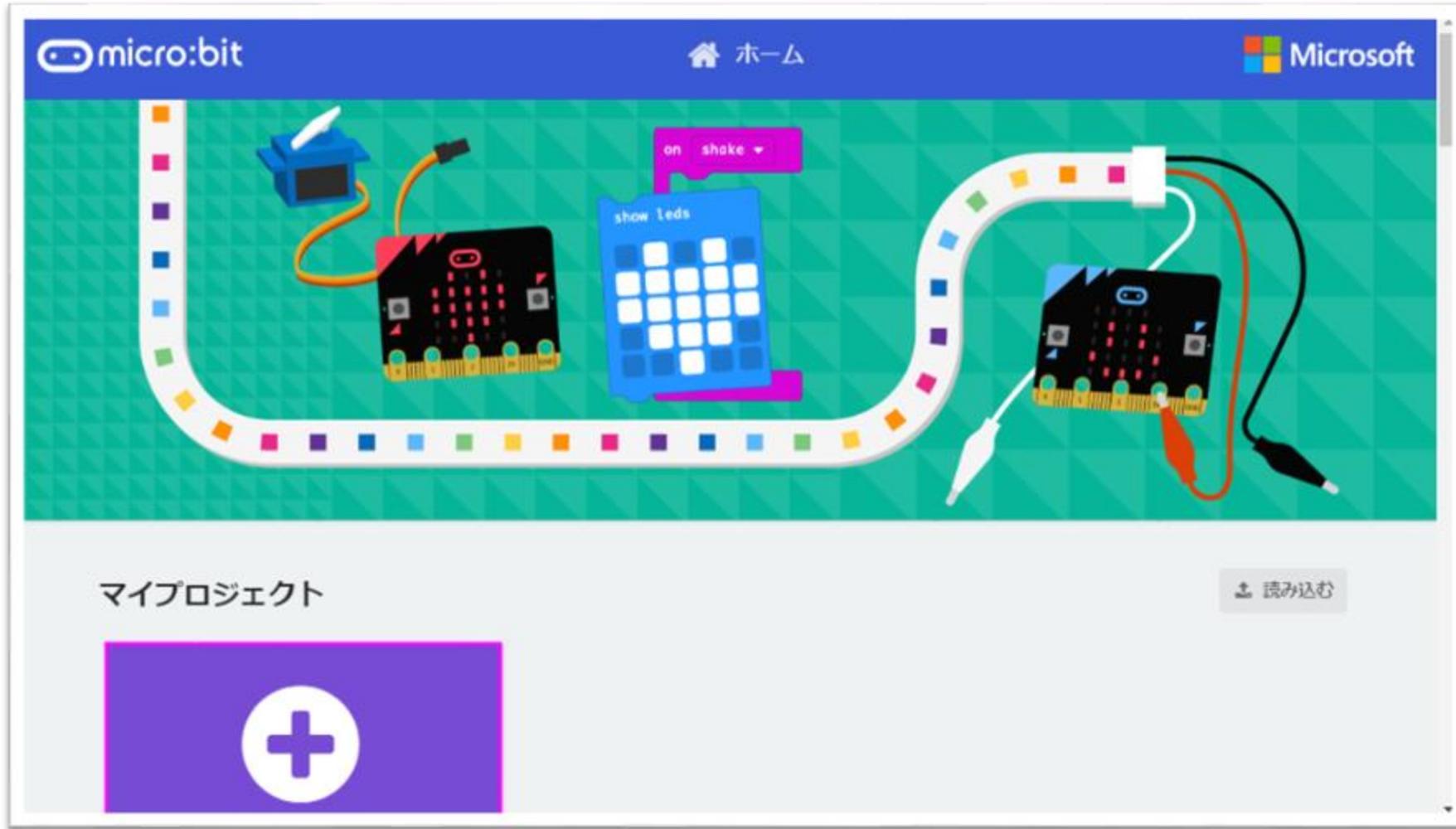
⑤ PCとmicro:bitを接続し、  
ダウンロードした  
プログラムを転送する

micro:bitが動く！

chromeと  
Edgeを使っ  
た2種類の  
方法があり  
ます。



# PLEN:bit の使い方 ～makecodeへアクセスする～



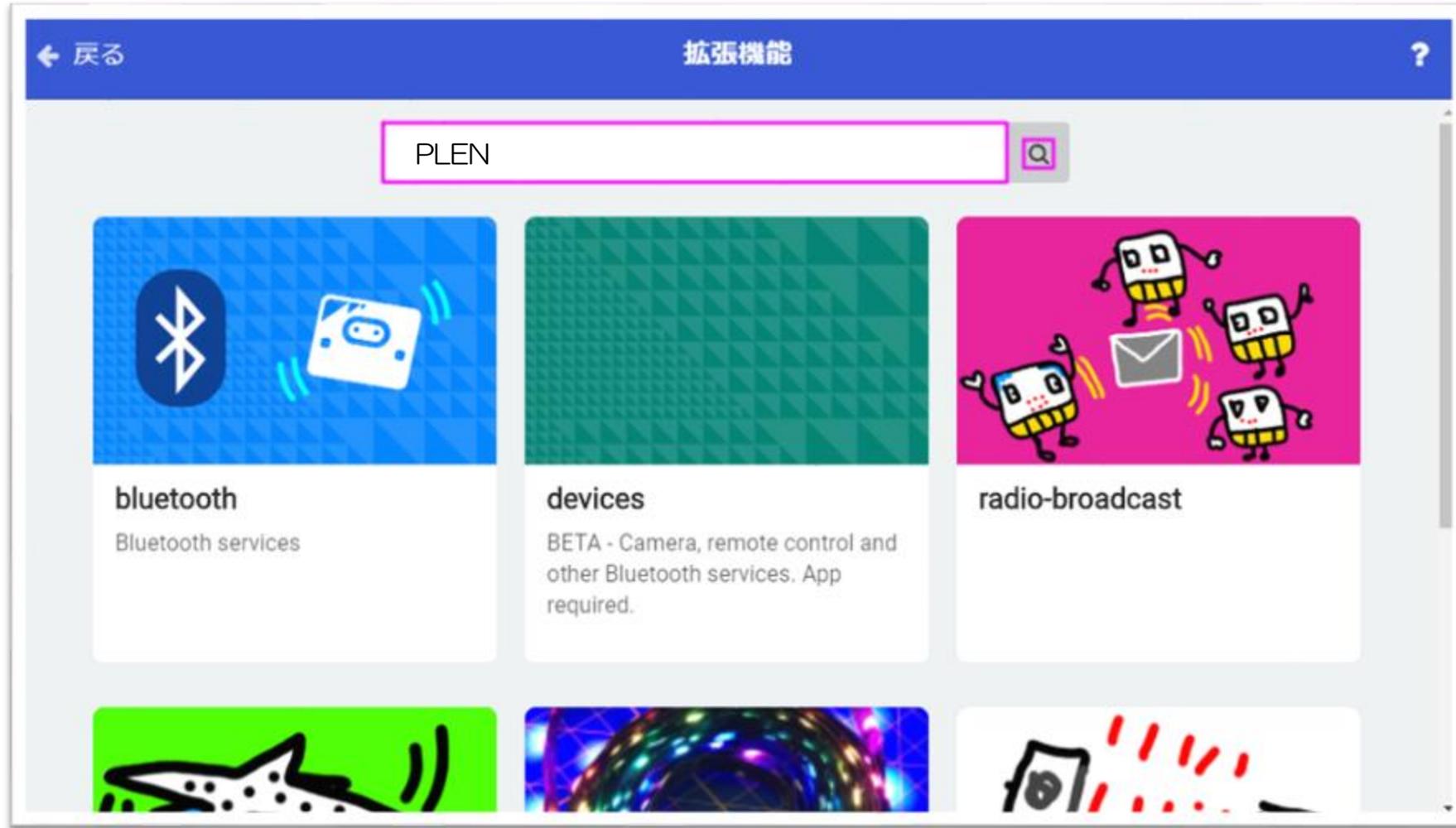
<https://makecode.microbit.org/> へアクセスし「+」をクリックします

# PLEN:bit の使い方 ～PLEN:bitブロックの追加～



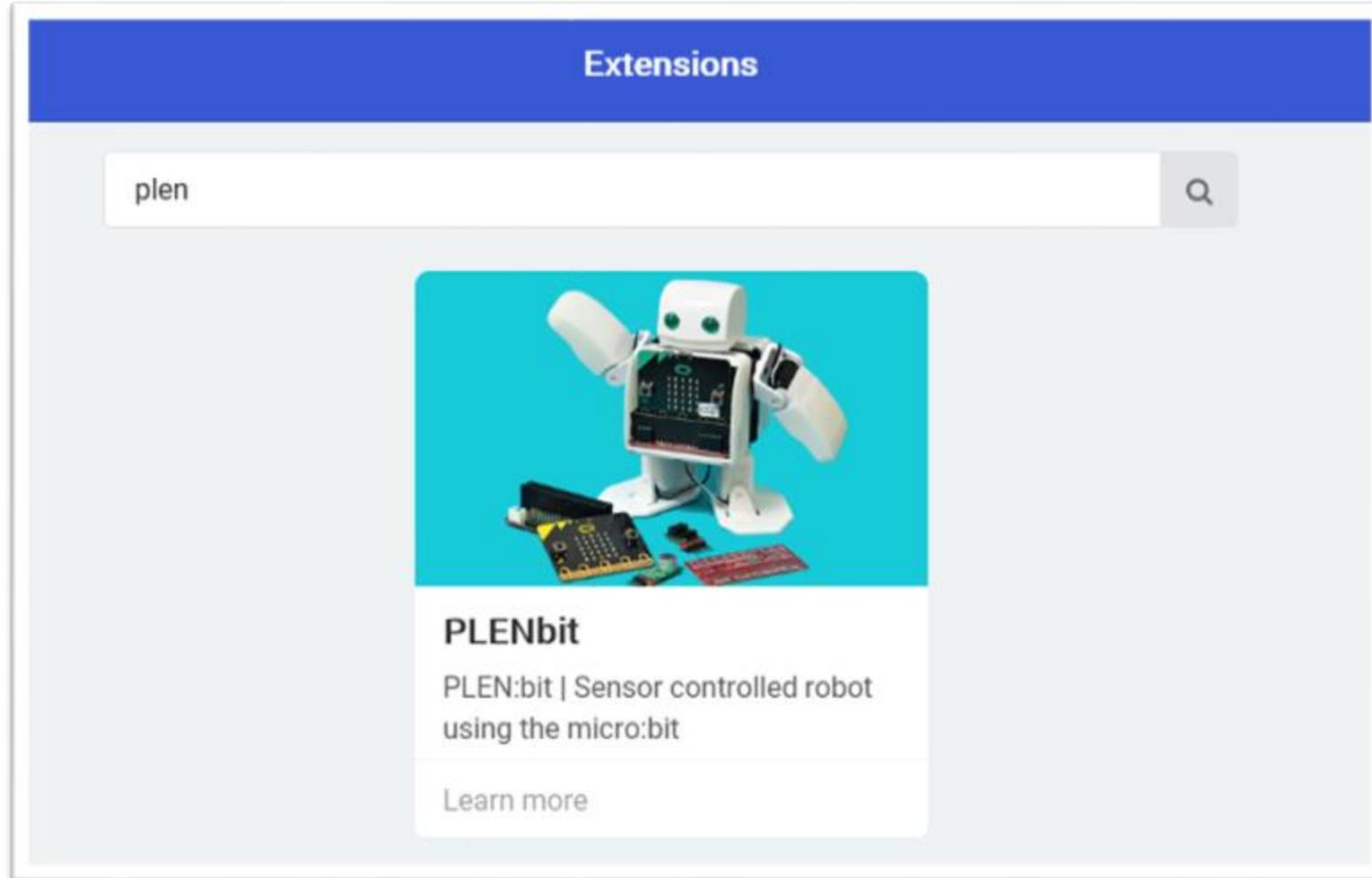
右上の歯車ボタン(⚙️)をクリックし、「拡張機能」を押す

# PLEN:bit の使い方 ～PLEN:bitブロックの追加～



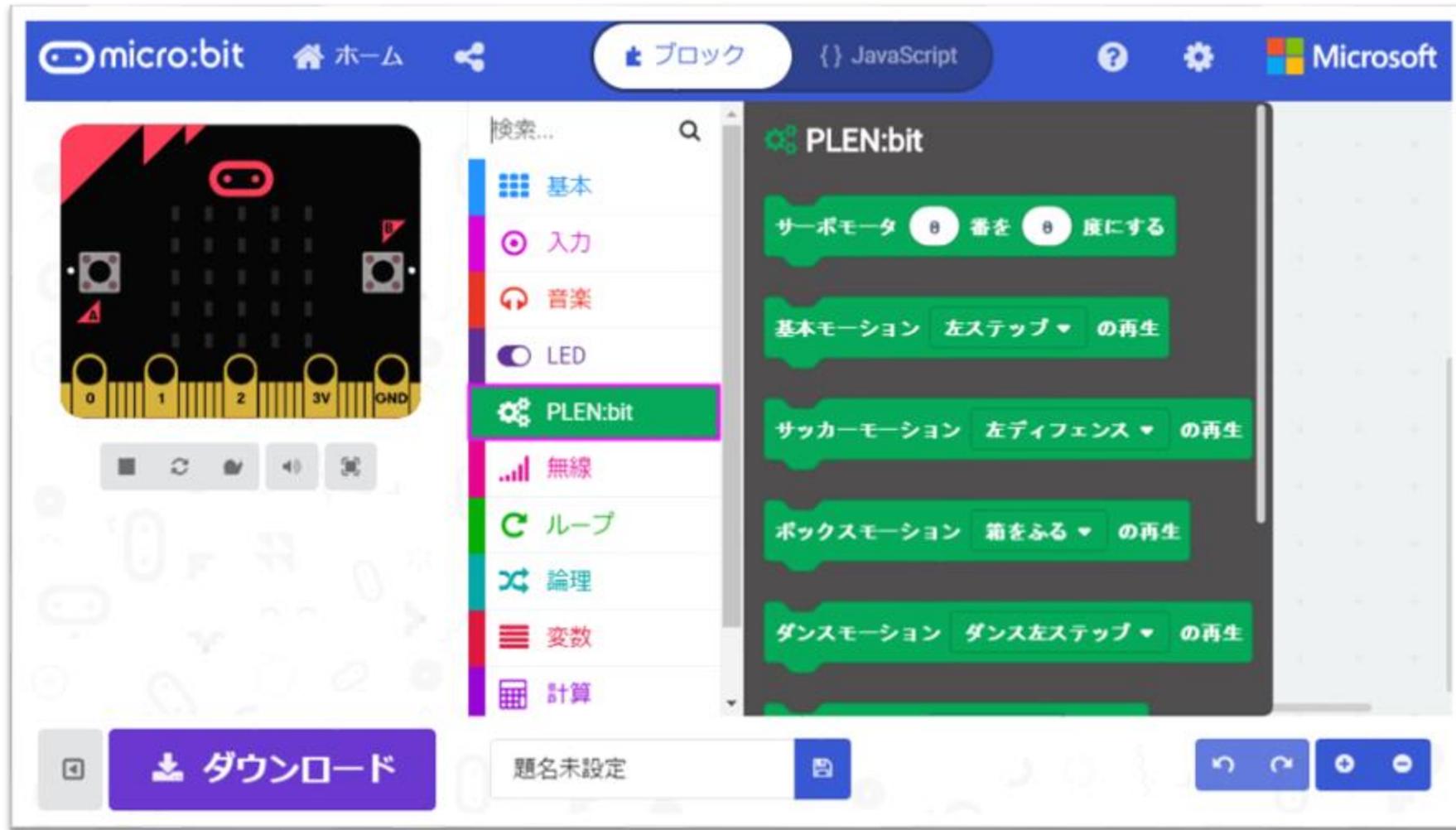
検索画面から『PLEN』と入力し、検索（🔍）する

# PLEN:bit の使い方 ～PLEN:bitブロックの追加～



検索結果から「PLENbit」を選択します

# PLEN:bit の使い方 ～PLEN:bitブロックの追加～



「PLEN:bit」というブロック一覧が追加されます

# PLEN:bit の使い方 ～プログラムの作成～

micro:bit ホーム ブロック JavaScript Microsoft

検索...

- 基本
- 入力
- 音楽
- LED
- PLEN:bit
- 無線
- ループ
- 論理
- 変数
- 計算

ボタン A が押されたとき

基本モーション 前に進む の再生

ボタン B が押されたとき

基本モーション 腕をパタパタする の再生

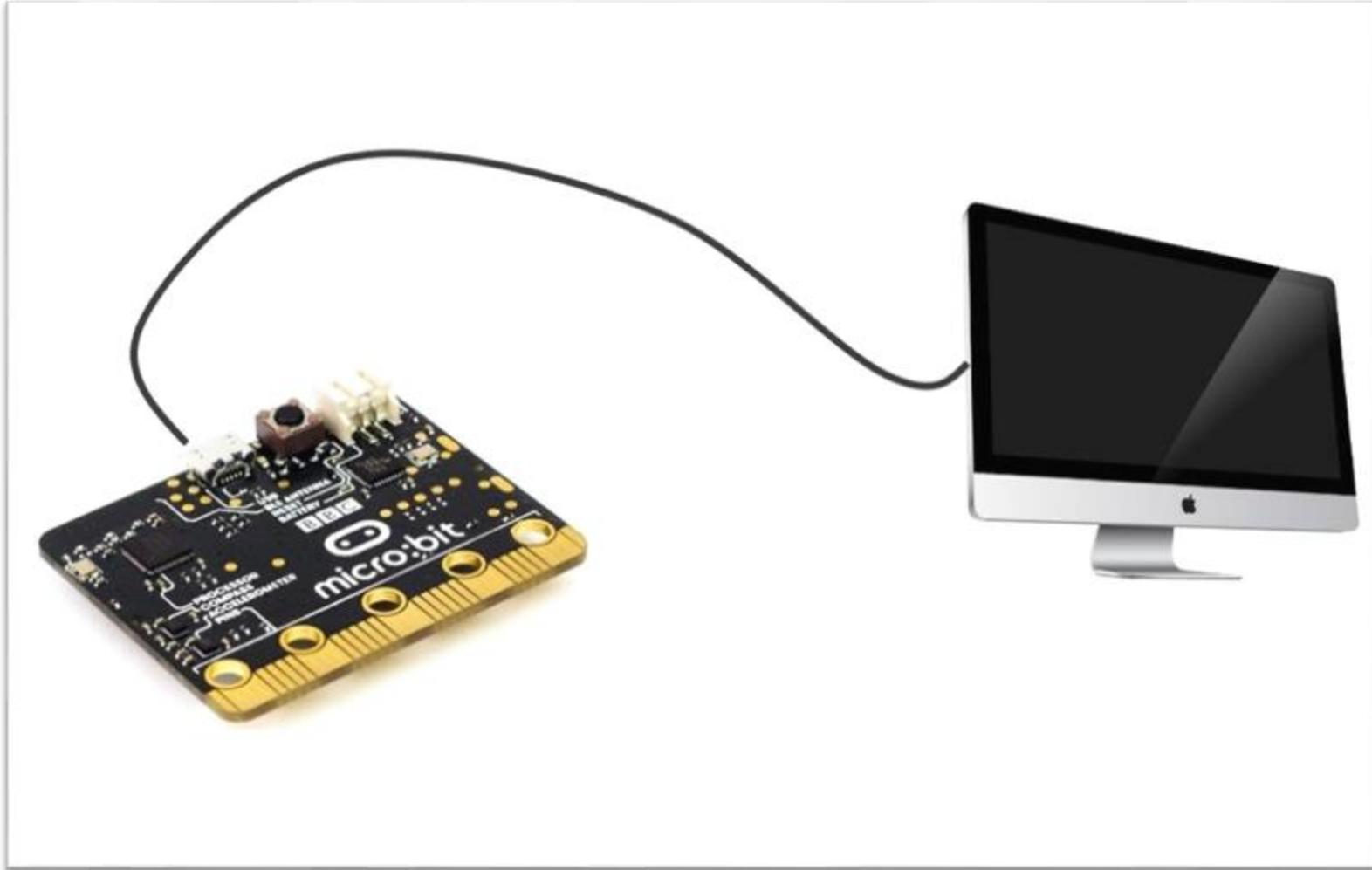
ダウンロード sample1

※Aボタンを押すと前に歩き、Bボタンを押すと腕をパタパタするプログラム

※プログラム名をつけておくと後で見分けやすくなります

画面を参考にブロックを置いてみよう

# PLEN:bit の使い方 ～デバイス接続～



micro:bitとPCを、USBケーブルで接続します

# PLEN:bit の使い方 ～プログラムのダウンロード～

Microsoft Edge  
の場合



※Chromeの場合は後述

画面左下にある「ダウンロード」ボタンを押す

# PLEN:bit の使い方 ～プログラムのダウンロード～

Microsoft Edge  
の場合



画面下に現れる操作で「保存」を選ぶ

# PLEN:bit の使い方 ～プログラムのダウンロード～

Microsoft Edge  
の場合



画面下に現れる操作で「フォルダーを開く」を選ぶ

# PLEN:bit の使い方 ～プログラムのダウンロード～



Microsoft Edge  
の場合

※ファイルエクスプローラが開かない場合は、補足資料に示した手順で起動可能です

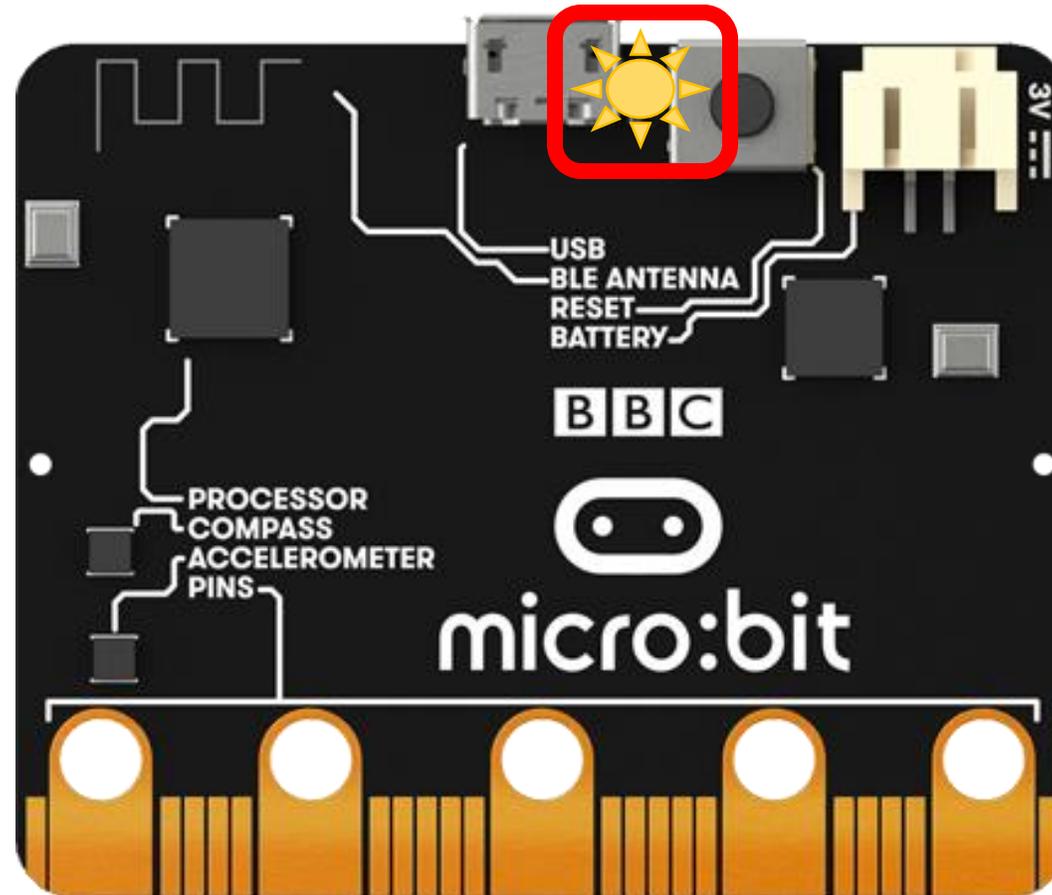
# PLEN:bit の使い方 ～プログラムのダウンロード～



転送したいプログラムを選んで「MICROBIT」にドラッグ&ドロップする<sup>16</sup>

# PLEN:bit の使い方 ～プログラムのダウンロード～

Microsoft Edge  
の場合

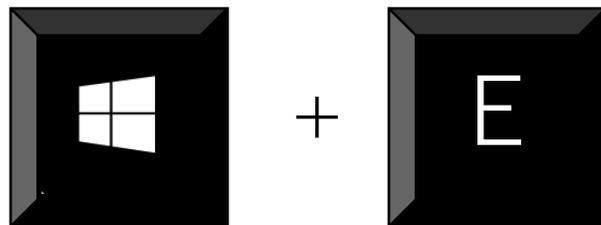


プログラムの転送中は、micro:bitの黄色LEDが点滅し、転送が完了すると消灯する

# PLEN:bit の使い方 ～補足1～

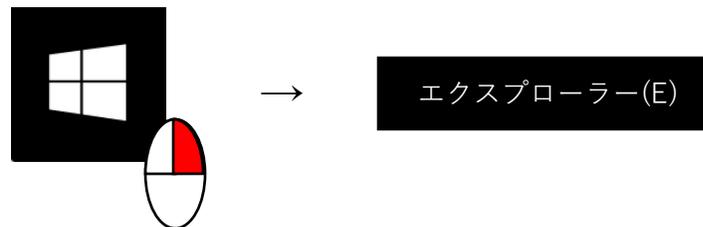
Microsoft Edge  
の場合

下記のどちらかの方法でファイルエクスプローラを起動する



※Windowsキーを押しながら、「E」キーを押す

もしくは

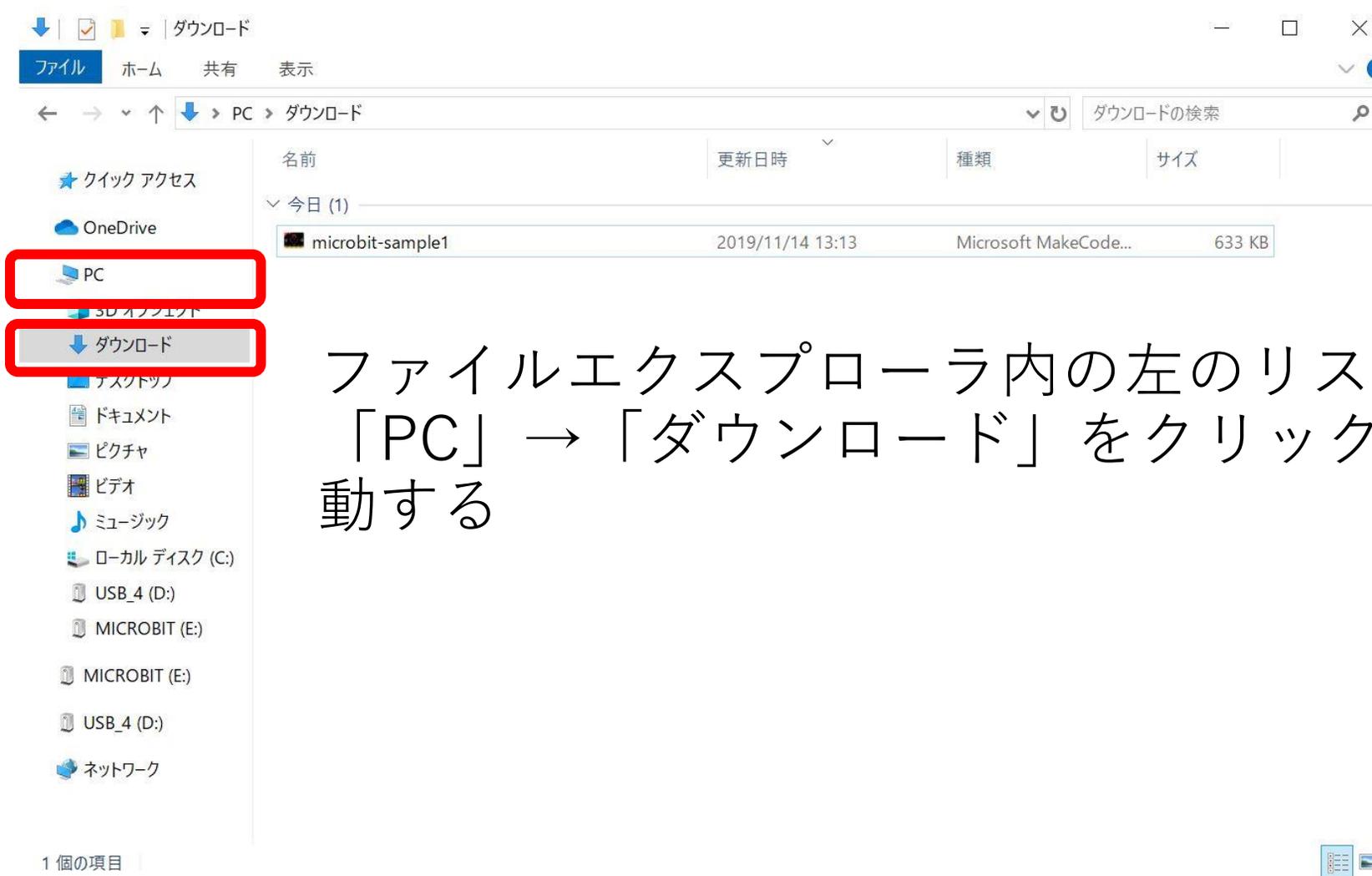


スタートボタンを右クリックして、「エクスプローラー(E)」

## ファイルエクスプローラを起動する方法

# PLEN:bit の使い方 ～補足2～

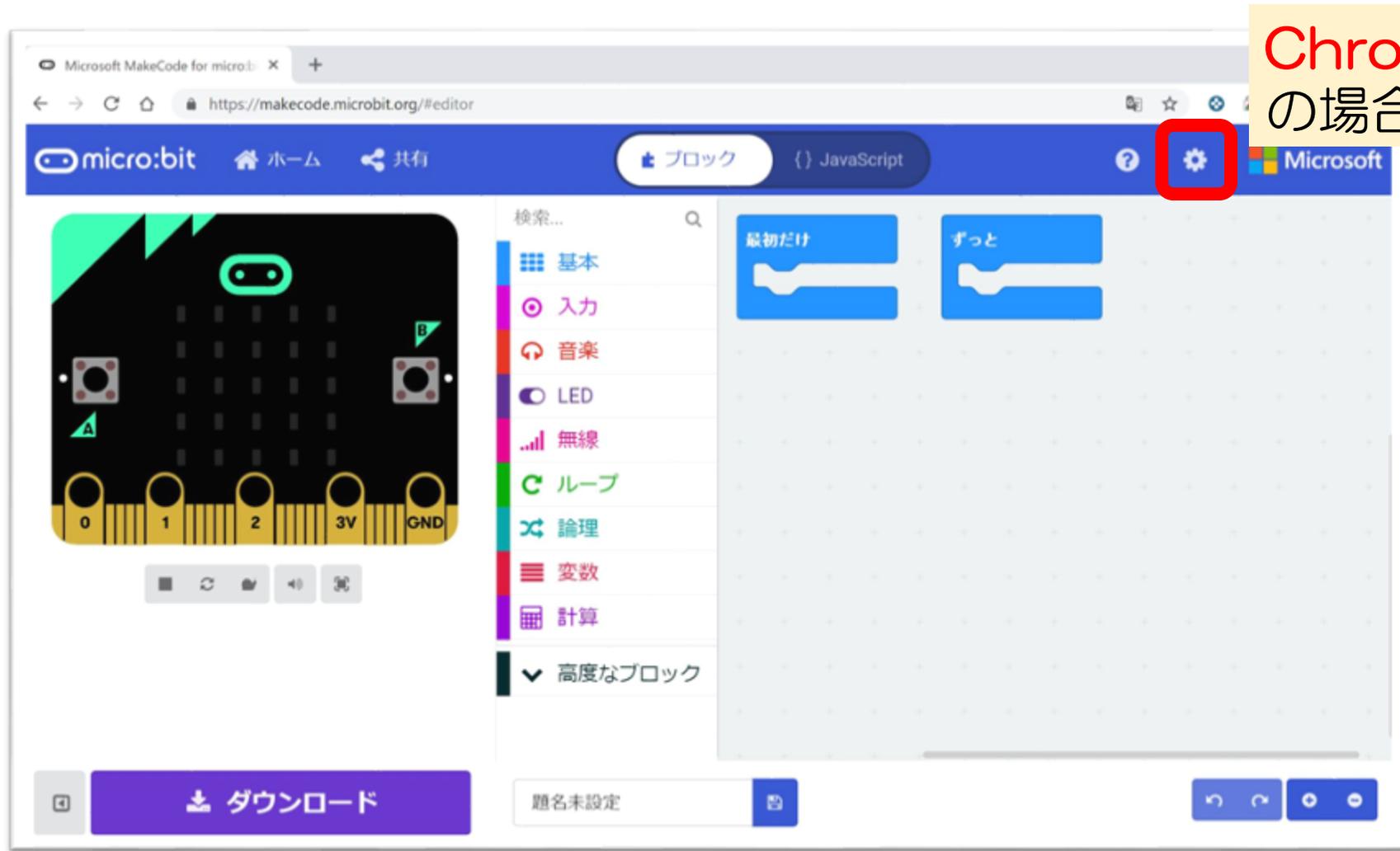
Microsoft Edge  
の場合



ファイルエクスプローラ内の左のリストから、「PC」→「ダウンロード」をクリックして移動する

ファイルエクスプローラを起動する方法

# PLEN:bit の使い方 ～プログラムのダウンロード～



画面内右上「歯車マーク」をクリックします

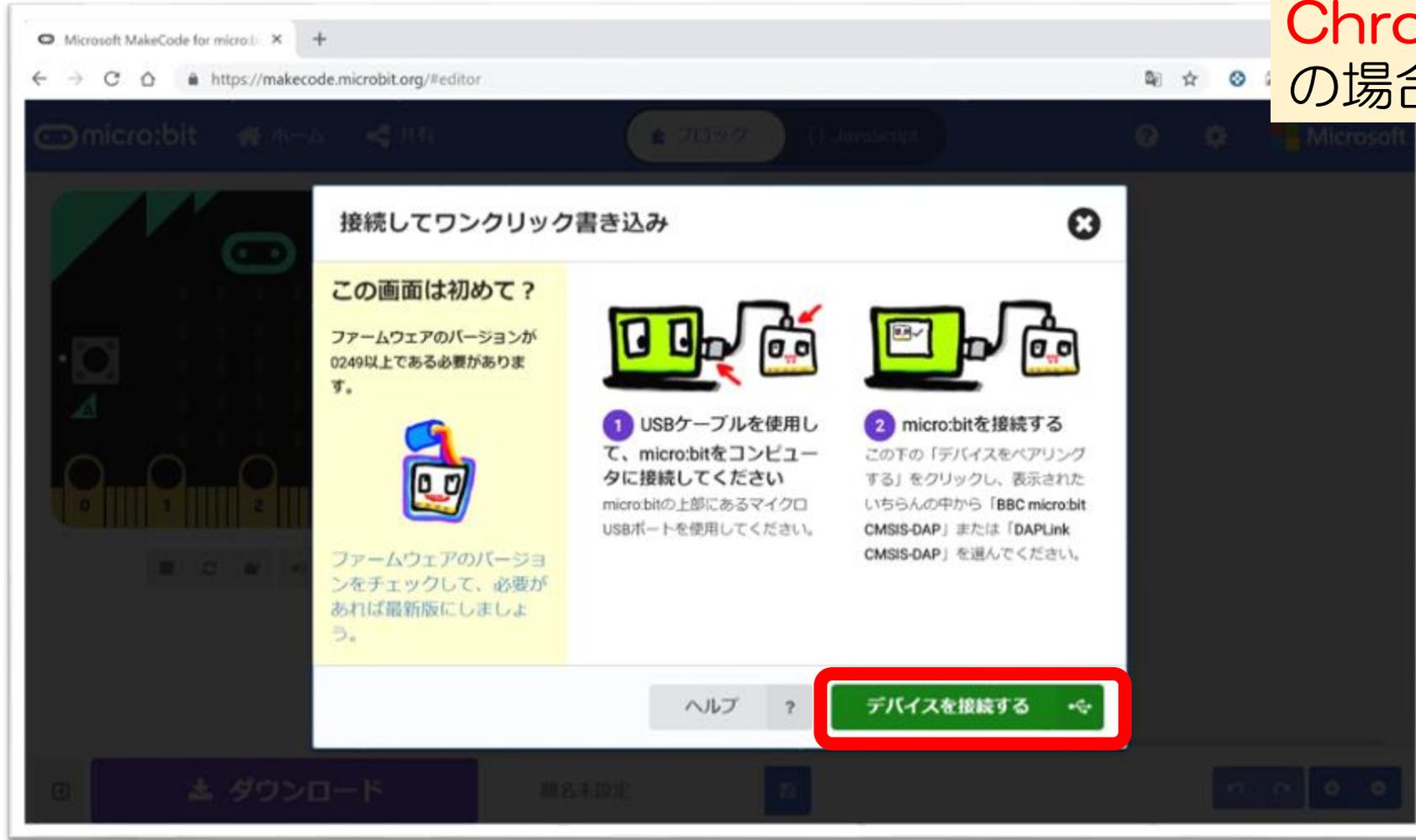
# PLEN:bit の使い方 ～プログラムのダウンロード～



「デバイスを接続する」をクリックします

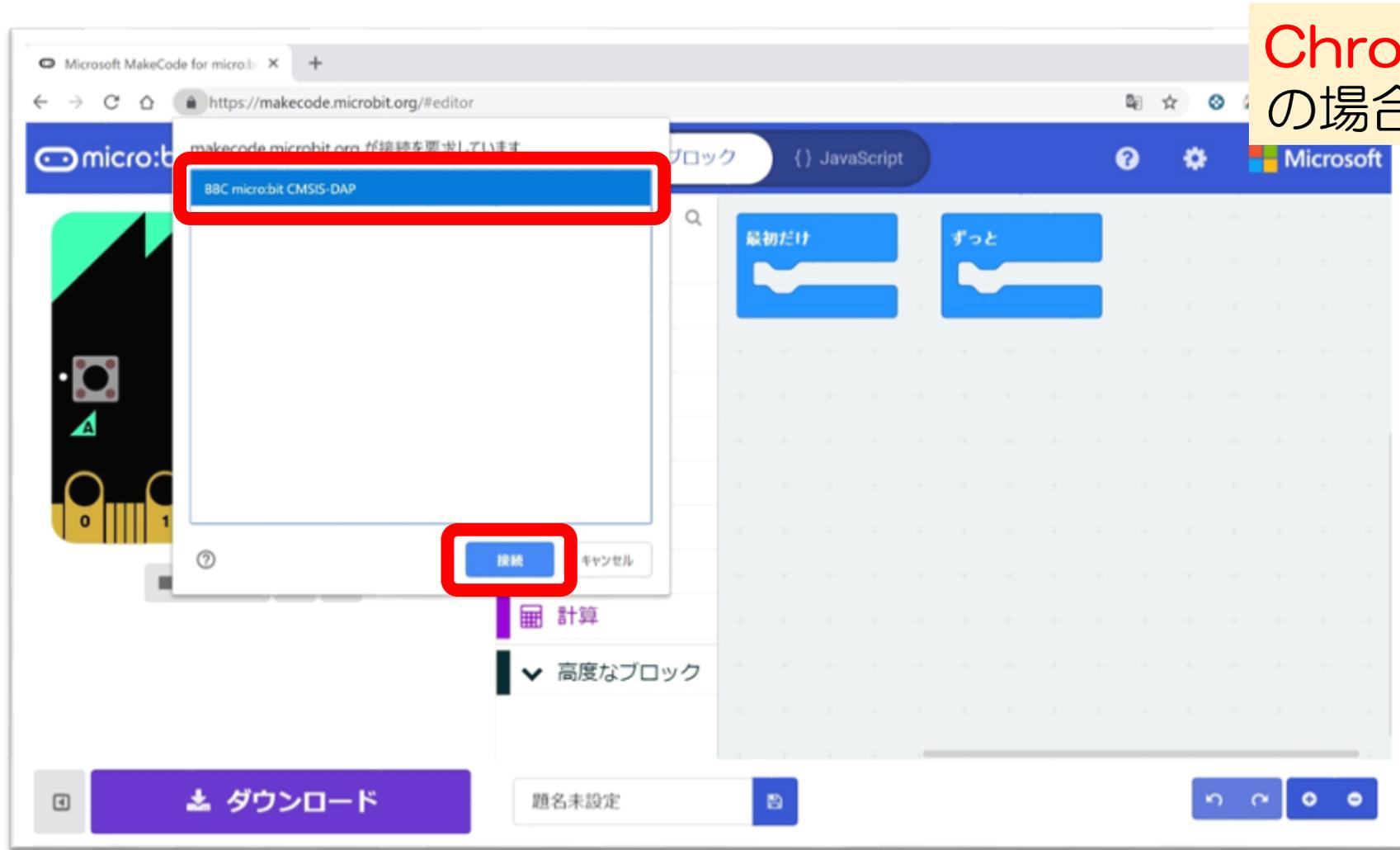
# PLEN:bit の使い方 ～プログラムのダウンロード～

Chrome  
の場合



「デバイスを接続する」ボタンをクリックします

# PLEN:bit の使い方 ～プログラムのダウンロード～



micro:bitを選択して「接続」ボタンをクリックします

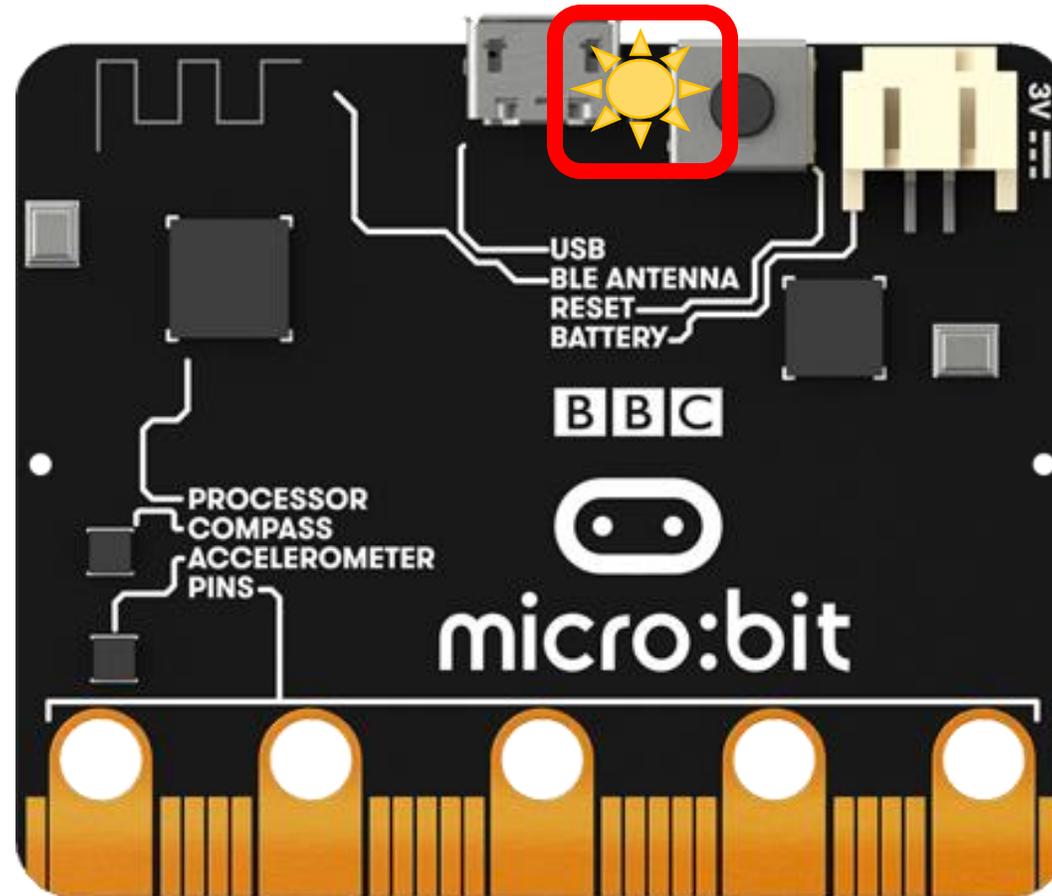
# PLEN:bit の使い方 ～プログラムのダウンロード～



「ダウンロード」を押してプログラムが転送されることを確認する

# PLEN:bit の使い方 ～プログラムのダウンロード～

Chrome  
の場合



プログラムの転送中は、micro:bitの黄色LEDが点滅し、転送が完了すると消灯する

# PLEN:bit のブロック一覧

目のLEDを オン ▾ にする

サーボモータ初期値設定

前進を する ▾

ダンスモーション ダンス左ステップ ▾ をする

サッカーモーション 左ディフェンス ▾ をする

基本モーション 前に進む ▾ をする

サーボモータ 0 番を 0 度にする

方角 (°)

Aボタン側 ▾ 距離が 600 以上

Aボタン側 ▾ マイクが 100 以上 (基準値 550 )

Aボタン側 ▾ マイク初期化

センサー Aボタン側 ▾ の値を読み取る

- ロボットの目のLEDを点灯させる
- サーボモータを初期位置に移動
- 連続歩行用。前進を「する」「しない」を切替できる
- ロボットに「ダンスモーション」の動きをさせる
- ロボットに「サッカーモーション」の動きをさせる
- ロボットに「基本モーション」の動きをさせる
- ロボットの関節を1つ指定し、角度を変える
- 方角を角度で取得
- 距離センサでの条件分岐に使用
- 音センサでの条件分岐に使用
- 音センサで使用する基準値を取得
- センサの値を読み取る (Aボタン側 or Bボタン側)

# PLEN:bit のプログラミング | PLEN:bit基本

最初だけ

- サーボモータ初期値設定
- アイコンを表示 

ボタン A が押されたとき

- 基本モーション 前に進む の再生

ボタン B が押されたとき

- 基本モーション 腕をパタパタする の再生

ずっと

- 目のLEDを オン にする
- 一時停止 (ミリ秒) 100 ▼
- 目のLEDを オフ にする
- 一時停止 (ミリ秒) 100 ▼

ボタン A+B が押されたとき

- サッカーモーション 右キック の再生

組立のときに  
使ってる  
プログラム  
だよ!

上のようにブロックを組んでプログラミングしてみましょう

# PLEN:bit のプログラミング | 距離センサ基本

[プログラムのリンク](#)



上のようにブロックを組んでプログラミングしてみましょう

# PLEN:bit のプログラミング | 音センサ基本

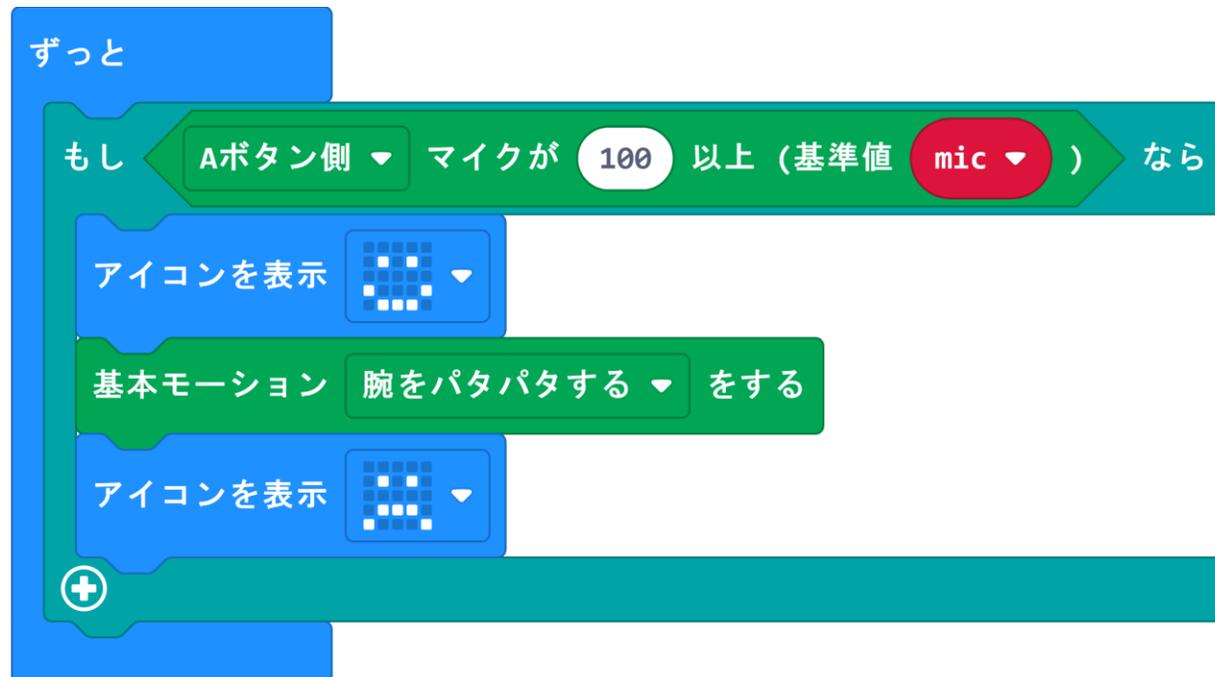
[プログラムのリンク](#)



Scratch script for microphone calibration:

- Block: 最初だけ (Initially)
- Block: 変数 mic を Aボタン側 マイク初期化 にする (Set variable mic to A button side microphone initialization)
- Block: アイコンを表示 (Show icon)

マイクのキャリブレーション (Microphone Calibration)



Scratch script for microphone sensitivity:

- Block: ずっと (Forever)
- Block: もし Aボタン側 マイクが 100 以上 (基準値 mic) なら (If A button side microphone is 100 or more (reference value mic))
- Block: アイコンを表示 (Show icon)
- Block: 基本モーション 腕をパタパタする をする (Basic motion: flap the arm)
- Block: アイコンを表示 (Show icon)

上のようブロックを組んでプログラミングしてみましょう

# PLEN:bit のプログラミング | 連続歩行

最初だけ

アイコンを表示



[プログラムのリンク](#)

ボタン **A** ▼ が押されたとき

基本モーション 前に進む ▼ をする

基本モーション 前に進む ▼ をする

基本モーション 前に進む ▼ をする

ボタン **B** ▼ が押されたとき

前進を する ▼

前進を する ▼

前進を する ▼

前進を 止める ▼

一歩ずつ  
止まって進む

止まらず  
連続で3歩進む

上のようブロックを組んでプログラミングしてみましょう

# PLEN:bit のプログラミング | 転ぶと助けをもとめる

プログラムのリンク

最初だけ

アイコンを表示

ずっと

もし 加速度 Z < -512 なら

アイコンを表示

基本モーション 腕をパタパタする の再生

でなければ

アイコンを表示

上のようにブロックを組んでプログラミングしてみましょう

# PLEN:bit のプログラミング | 壁をよけて歩く

[プログラムのリンク](#)



上のようにブロックを組んでプログラミングしてみましょう

# PLEN:bit のプログラミング | 壁をよけて連続歩行

最初だけ

アイコンを表示 

ずっと

もし **Bボタン側** 距離が **600** 以上 なら

前進を **止める**

アイコンを表示 

くりかえし **3** 回

基本モーション **右に回る** をする

アイコンを表示 

でなければ 

前進を **する**



[プログラムのリンク](#)

連続歩行したい  
場合の参考に！

上のようブロックを組んでプログラミングしてみましょう

# PLEN:bit のプログラミング | 北に向けて歩く

[プログラムのリンク](#)

The image shows a sequence of code blocks for a navigation program. It starts with a 'ずっと' (Forever) loop block. The first block is '変数 direction を 方角 (°) にする' (Set variable direction to direction (°)). This is followed by a 'もし' (If) block with the condition 'direction ≤ 20 または direction ≥ 340 なら' (If direction ≤ 20 or direction ≥ 340 then). Inside this block, there are three sub-blocks: '矢印を表示 上向き ↑' (Show arrow pointing up), '基本モーション 前に進む の再生' (Repeat basic motion 'move forward'), and 'でなければもし' (Else if) block with the condition 'direction > 20 かつ direction ≤ 180 なら' (If direction > 20 and direction ≤ 180 then). This block contains '矢印を表示 右向き →' (Show arrow pointing right), '基本モーション 左に回る の再生' (Repeat basic motion 'turn left'), and another 'でなければもし' (Else if) block with the condition 'direction > 180 かつ direction < 340 なら' (If direction > 180 and direction < 340 then). This final block contains '矢印を表示 左向き ←' (Show arrow pointing left) and '基本モーション 右に回る の再生' (Repeat basic motion 'turn right'). The code ends with a '+' sign in a small blue block.

上のようにブロックを組んでプログラミングしてみましょう

# PLEN:bit のサーボモータの初期位置調整

```

最初だけ
  サーボモータ初期値設定
  アイコンを表示
ずっと
  もし ボタン A が押されている なら
    関数呼び出す servoAdjust
  でなければもし ボタン B が押されている なら
    基本モーション 前に進む の再生
  
```

```

関数 servoAdjust
  変数 adjNum を 0 にする
  変数 servoNum を 0 にする
  数を表示 servoNum
  変数 loop を 真 にする
  もし loop ならくりかえし
    もし ボタン A+B が押されている なら
      save positon servoNum adjNum
      変数 servoNum を 1 だけ増やす
      変数 adjNum を 0 にする
      数を表示 servoNum
    でなければもし ボタン A が押されている なら
      変数 adjNum を 1 だけ増やす
      変数 adjNum を servo adjust servoNum adjNum 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 adjNum を -1 だけ増やす
      変数 adjNum を servo adjust servoNum adjNum にする
    でなければもし servoNum > 7 なら
      アイコンを表示
      一時停止 (ミリ秒) 2000
      変数 loop を 偽 にする
  
```



[プログラムのリンク](#)

使い方

- 1.Aボタンを押すと調整開始
- 2.Aボタン または Bボタンでサーボモータを移動
- 3.A+Bボタンで決定。次のサーボモータに
- 4.繰り返す
- 5.8個目が決定したらにっこりマークが表示される。
- 6.電源を入れ直し、Bボタンを押すと歩行するので、こけずに歩けると成功！！

※起動時にBボタンで保存されている初期位置を削除

リンクからプログラムをダウンロードして使おう



# PLEN:bit のプログラミング | モーションを作ろう2

The screenshot shows a programming block titled "関数 R\_Punch" (Function R\_Punch). The block contains the text "サーボを 300 ミリ秒で各角度を" (Move servo for 300 milliseconds to each angle). To the right of this text is a vertical list of input fields for servo angles. The first field is set to "0", and the fourth field is set to "-900". The other fields are empty. A blue arrow points from the "0" field to a yellow callout box on the right.

この要素のサーボ番号

0  
1  
2  
3  
4  
5  
6  
7

この要素の配列

8個の数字は8個のサーボモータの角度を入力します。  
(-1800 ~ 1800)  
サーボの番号は上から順に0,1,2,3,4,5,6,7です。  
動かさないサーボモータは0のままにします。

ブロックの組み合わせによりオリジナルモーションが作れます！

# PLEN:bit のプログラミング | リモートコントロール

[プログラムのリンク](#)

リモコン側に必要なブロック

PLEN:bit側に必要なブロック

```
最初だけ  
無線のグループを設定 0  
アイコンを表示  
  
ボタン A が押されたとき  
無線で文字列を送信 "A"  
  
ボタン B が押されたとき  
無線で文字列を送信 "B"  
  
ボタン A+B が押されたとき  
無線で文字列を送信 "C"  
  
無線で受信したとき receivedString  
もし receivedString = "A" なら  
基本モーション 前に進む をする  
でなければもし receivedString = "B" なら  
基本モーション 右に回る をする  
でなければもし receivedString = "C" なら  
基本モーション ハイタッチ をする
```

PLEN:bit側とリモコン側で必要なブロックは違いますが、このプログラムをそのまま両方のmicro:bitに入れても動作します。また、相互通信も可能です。

micro:bitを2つ用意してPLEN:bitを遠隔操作！

# PLEN:bit のプログラミング | 距離センサをcmに変換

[プログラムのリンク](#)

```
最初だけ
  変数 誤差調整 を 20 にする
  アイコンを表示

ずっと
  変数 dis を センサー Bボタン側 の値を読み取る にする
  変数 dis を 数値をマップする dis 元の下限 0 元の上限 1023 結果の下限 0 結果の上限 330 誤差調整 にする
  変数 dis を 数値をマップする dis 元の下限 60 元の上限 220 結果の下限 50 結果の上限 4 にする
  シリアル通信 名前と数値を書き出す "CM" = dis
  もし dis ≤ 6 なら
    基本モーション 腕をパタパタする をする
  一時停止 (ミリ秒) 50
```

1. 距離センサの値を電圧に換算  
0~1023 → 0~3.3V  
2. センサーの出力する電圧(0.6~2.2V)と長さ(4~50cm)を当てはめてcmに換算

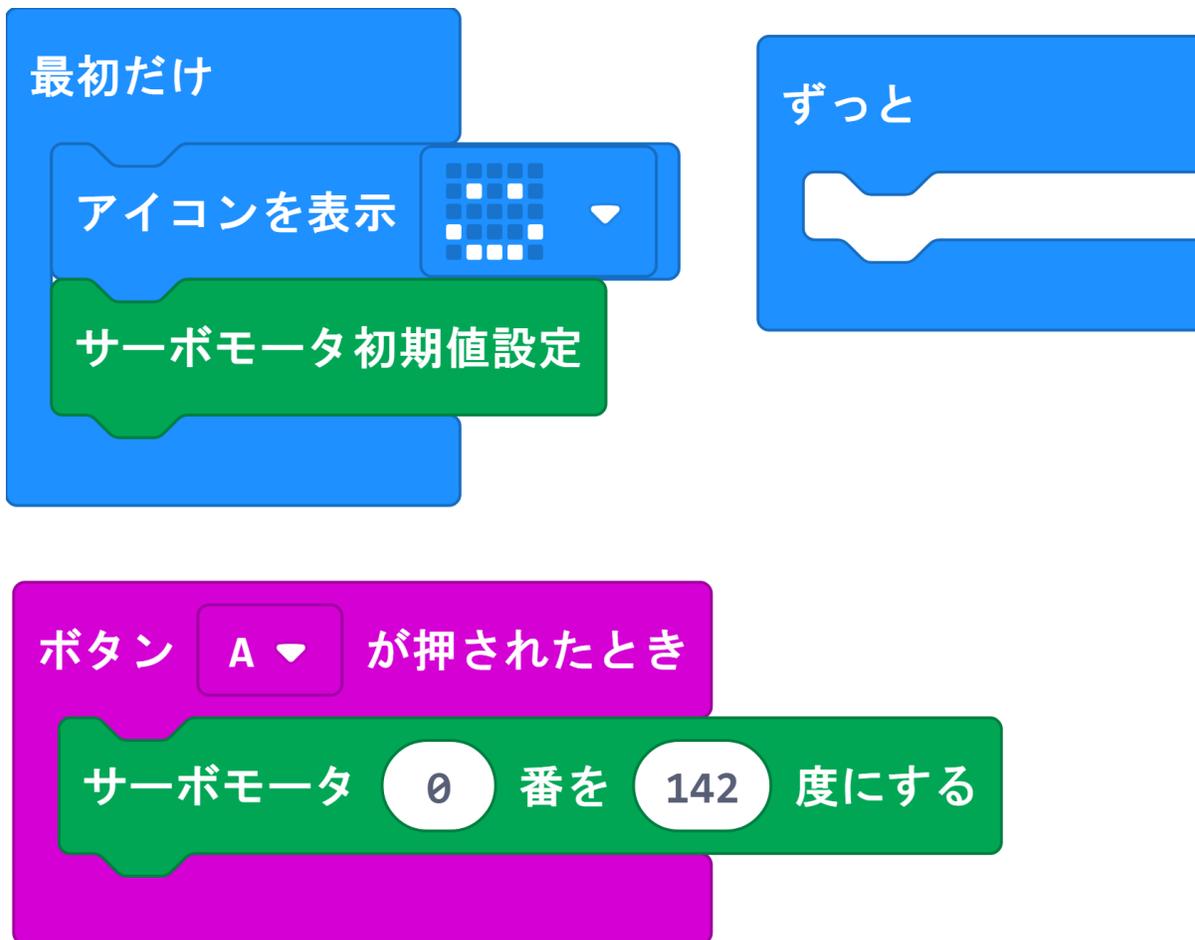
←センサからの距離が6cm以内なら腕をパタパタする

上のようにブロックを組んでプログラミングしてみましょう

# PLEN:bit のプログラミング | サーボモータを動かそう

[サーボモータ]ブロックを使う場合は、「最初だけ」のなかに、[サーボモータ初期位置設定]ブロックを必ず入れてください。

無ければ、サーボモータが変な角度に動いてしまいます。



サーボモータをPLEN:bit以外の用途で使う場合に便利です

# PLEN:bit のプログラミング | センサの値を知ろう

[プログラムのリンク](#)

最初だけ

アイコンを表示

ずっと

シリアル通信 名前と数値を書き出す "mic" = センサー A ▼ ボタン側 の値を読み取る

シリアル通信 名前と数値を書き出す "dis" = センサー B ▼ ボタン側 の値を読み取る

一時停止 (ミリ秒) 100 ▼

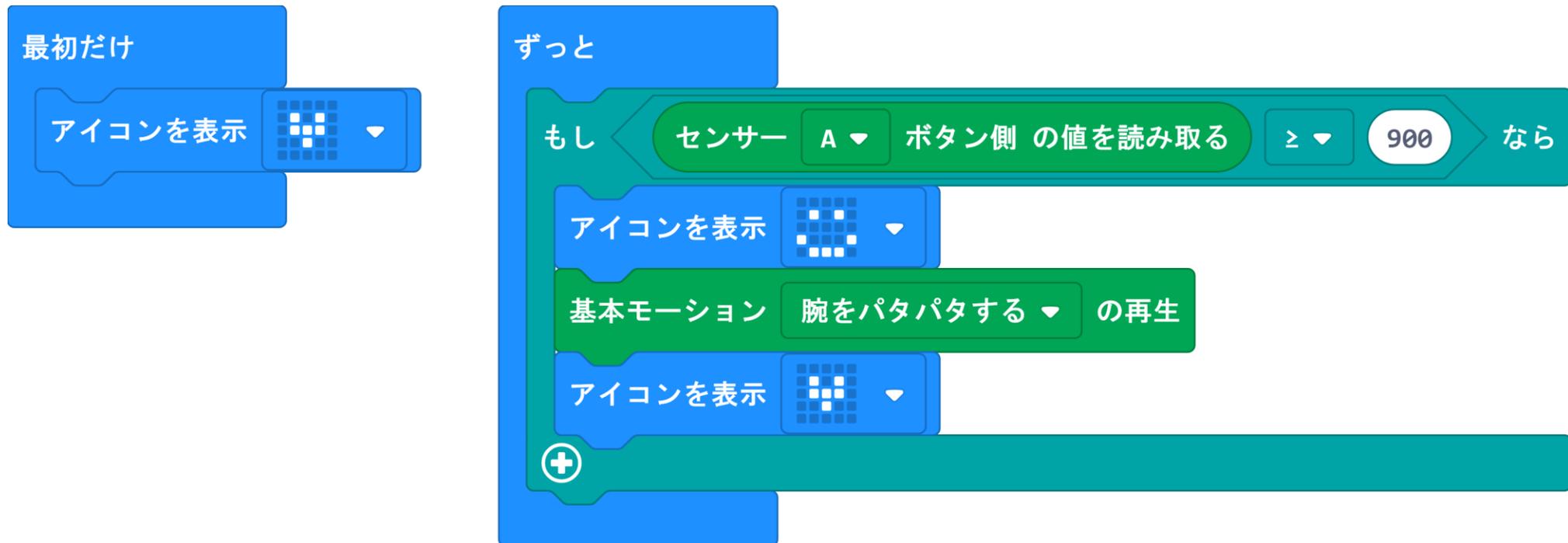
上のようにブロックを組んでプログラミングしてみましょ

# PLEN:bit のプログラミング | 旧距離センサ基本



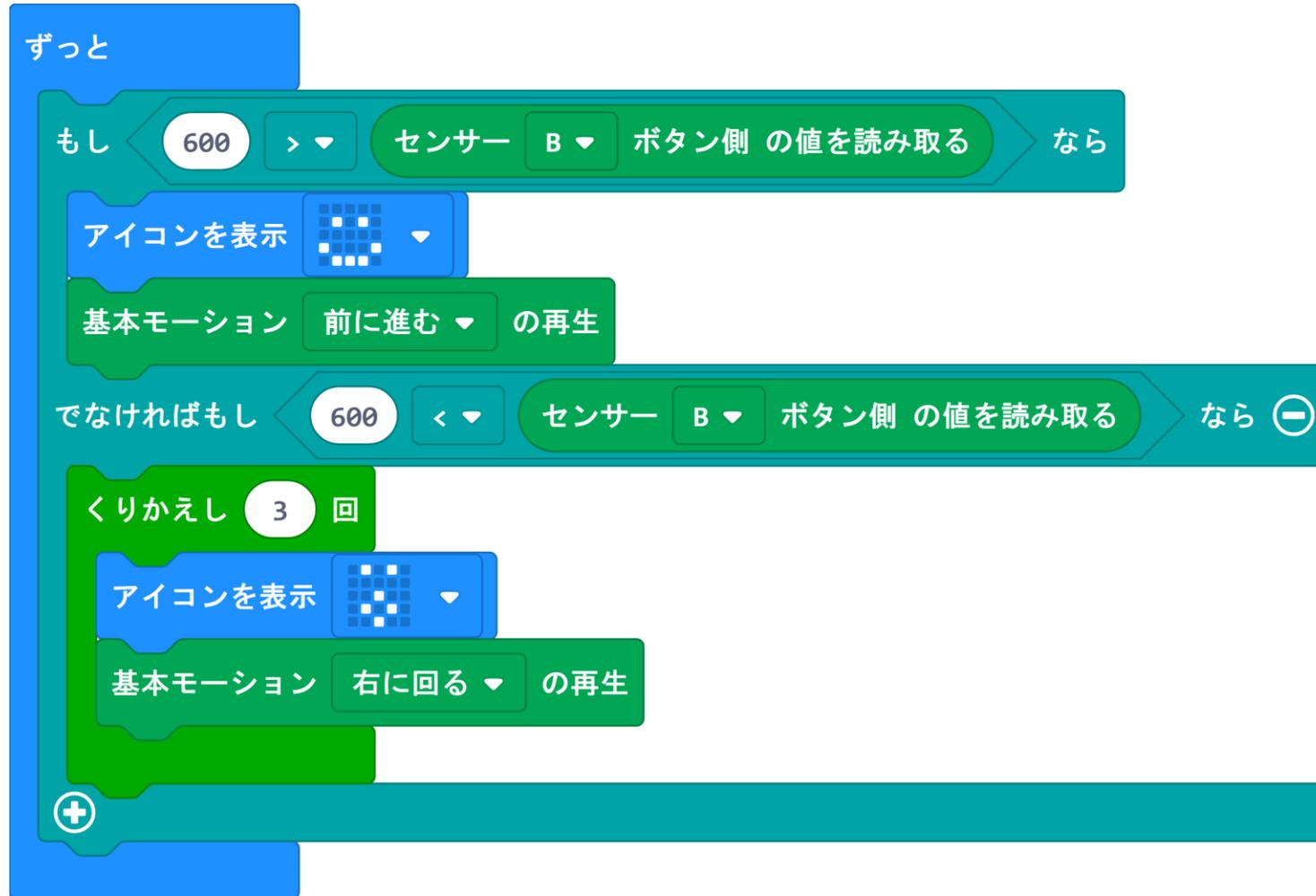
上のようにブロックを組んでプログラミングしてみましょう

# PLEN:bit のプログラミング | 旧音センサ基本



上のようにブロックを組んでプログラミングしてみましょう

# PLEN:bit のプログラミング | 旧壁をよけて歩く



上のようにブロックを組んでプログラミングしてみましょう

# PLEN:bit のプログラミング | 赤外線を受信



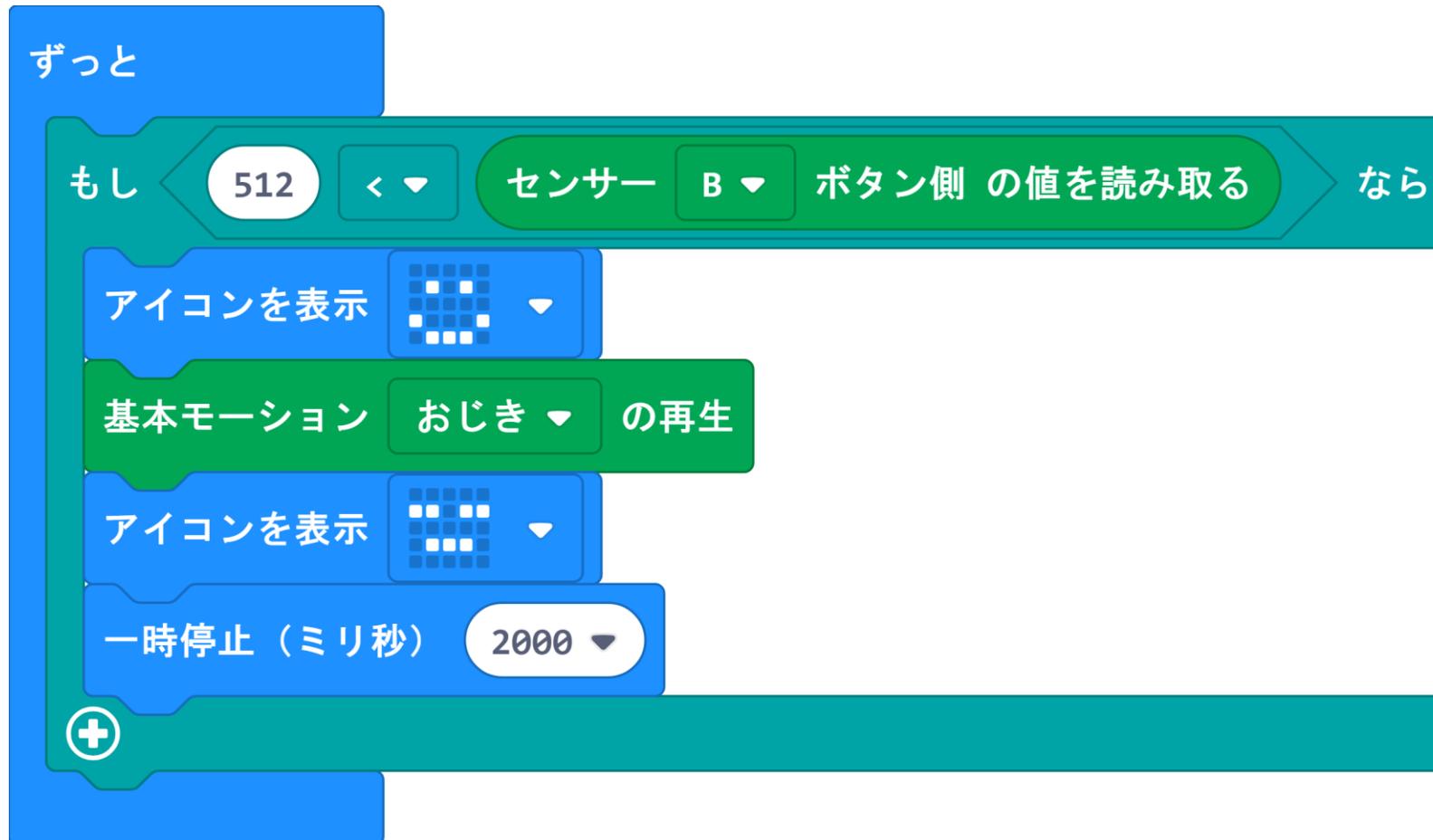
```
ずっと
もし < センサー A ▼ ボタン側 の値を読み取る < ▼ 512 > なら
  アイコンを表示 [アイコン] ▼
  基本モーション こくはく ▼ の再生
  アイコンを表示 [アイコン] ▼
  +
```

赤外線リモコンなら  
なんでも使えます

赤外線センサーを使った赤外線検知プログラムです。  
赤外線リモコンのボタンを押すと腕をパタパタします。

# PLEN:bit のプログラミング | いらっしゃい!

SPKIT



人感センサーを使った人感知プログラムです。  
PLEN:bitの前で人が動くと腕をパタパタします。

# PLEN:bit のBLE版目玉基板ってどうするの？



最初だけ

アイコンを表示



アプリをダウンロードして使ってね

**PLEN Connect**

[iPhone用](#)

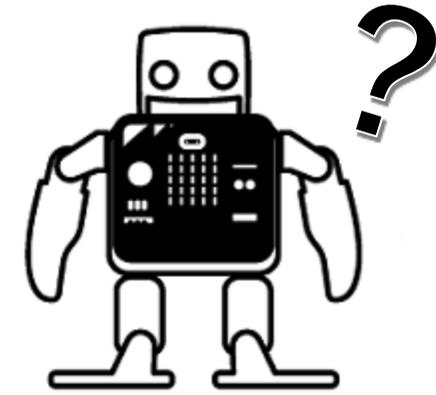
[Android用](#)

ずっと

スマホからコントロールできるようにする

BLE版目玉基板を通常の日玉基板と付け替えることでスマートホンのアプリから操作することが可能になります。

# PLEN:bit の HELP コーナー



## 1. PLEN:bitのブロックが違う??

PLEN:bitの拡張機能は日々更新しています。  
なにかおかしい場合は、追加しなおしてみてください。

## 2. わからないことがありましたら、下記までご連絡ください。

- PLENサポート <https://plen.jp/wp/contact/>

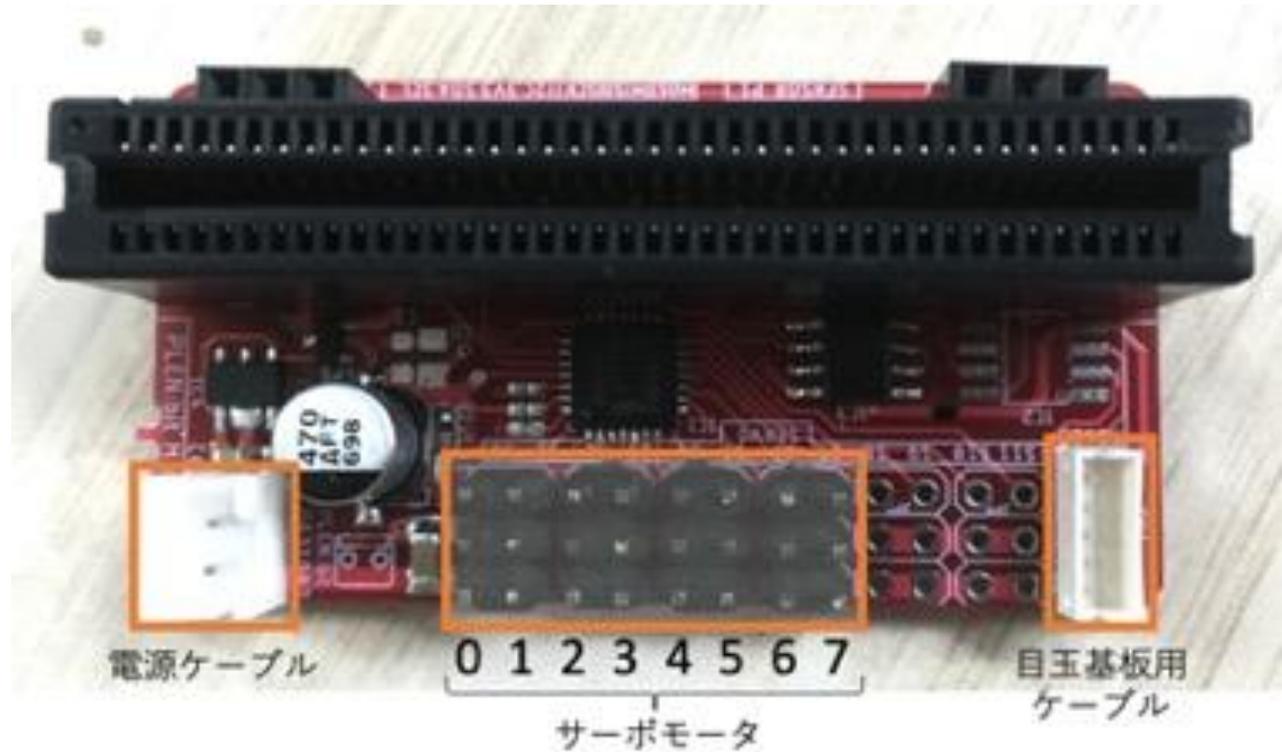
# PLEN:bit をタブレットでプログラミング



アプリをダウンロードして使ってね  
**micro:bit 公式アプリ**  
[iPhone用](#)  
[Android用](#)

タブレットまたはスマートフォンで、micro:bit公式アプリを使う事で  
micro:bitにプログラムを書き込むことができます。  
※対応していない機種もございます。

# PLEN:bit のサーボモータ番号



# PLEN:bit の活用事例

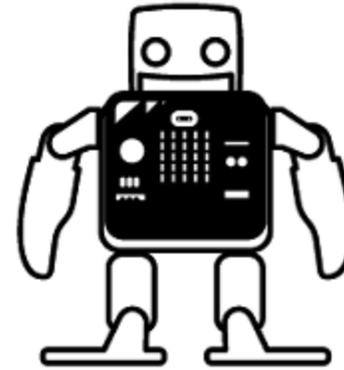


関東圏 アフタースクール「習い事教室」にて



関西圏私立大学 初等部「クラブ活動」にて

# PLEN:bit 変更履歴



発行日	変更内容
2019/12/4	
2020/10/1	新ブロック説明追記
2021/01/15	初期位置調整プログラムのリンク修正