

# micro:bit 「サンプル プログラミング集」 (第5.0版)

【ブロック+ V2 分冊版】

大分県立芸術文化短期大学  
情報コミュニケーション学科 非常勤講師  
望月陽一郎  
[mochizuki@oita-pjc.ac.jp](mailto:mochizuki@oita-pjc.ac.jp)

作品のクレジットを表示すること・営利目的での利用をしないこと・元の作品を改変しないこと  
(作者の許可を得ている場合は除きます)



# 目次 1

- 001 温度を表示する
- 002 明るさを表示する
- 003 ボタンで温度を表示する
- 004 ボタンで明るさを表示する
- 005 ボタンで明るさ・温度を表示する
- 006 暗くなるとLEDが点く
- 007 Happy Birthdayを鳴らす
- 008 電子オルゴールを鳴らす
- 009 テスターで音を鳴らす
- 010 明るさで音が変わるテルミン
- 011 ストップウォッチ
- 012 明るさで秒数計測
- 013 コンパス (Compass)
- 014 2台で送受信
- 015 水準器
- 016 傾きメーター
- 017 方位磁針
- 018 方位磁針16方位
- 019 方位磁針16方位program分割
- 020 暗くなったことを検知
- 021 相互にストップウォッチ
- 022 通信で合奏
- 023 明るさ変化を送信
- 024 揺れたら送信
- 025 明るさ数値送信
- 026 ひらがな表示
- 027 傾き (ピッチ) 角度を表示
- 028 明るさにより発信音の高さを変える
- 029 太陽高度計
- 030 ふりこ送信
- 031 ふりこ送信 段階1
- 032 ふりこ送信 段階2
- 033 文字列を送信
- 034 接近したら点灯 (距離センサー)
- 035 スイッチを制御 (制御スイッチ)
- 036 リモコン・スイッチで点灯 (制御スイッチ)
- 037 人感センサー  
(距離センサー・制御スイッチ)
- 038 距離を表示 (距離センサー)
- 039 明るさグラフ
- 040 明るさグラフリアルタイム
- 041 距離を測定 (距離センサー)
- 042 micro:bitファームウェアアップデート
- 043 人感センサー
- 044 ランダム数値表示
- 045 じゃんけん
- 046 じゃんけん無線判定
- 047 磁気表示
- 048 磁力計
- 049 検流 (ペルチェ素子)
- 050 検流グラフ (ペルチェ素子)

# 目次 2

- 051 計算機
- 052 計算機（関数）
- 053 気温を小数表示
- 054 湿度を小数表示
- 055 気圧を小数表示
- 056 気温・湿度・気圧を小数表示
- 057 簡易熱中症計
- 058 気温数値を送信
- 059 気温数値を受信
- 060 気温・湿度を同時送信
- 061 気温・湿度を同時受信

## 【v2 編】

- 001 重ねて音を鳴らす
- 002 音がしたらビックリ
- 003 タッチしたらビックリ
- 004 音のしきい値
- 005 静かになったら
- 006 音量グラフ
- 007 音量を送信
- 008 音量を受信
- 009 タッチする楽器
- 010 メロディを打楽器に

# micro:bit プログラミングサイト

<https://makecode.microbit.org/#editor>

Python・JavaScript編集画面  
への移動

ホーム画面へ

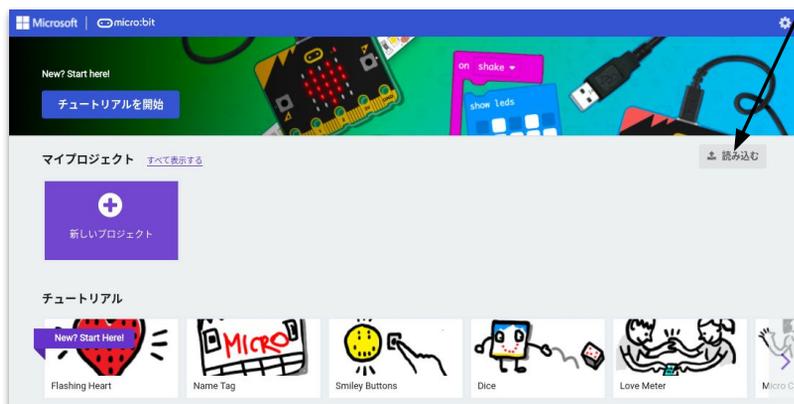


確認用シミュレータ

作成画面

作成したプログラムを  
USB接続した  
micro:bit にコピー

ローカル保存しておいた  
ファイル読込



## 【説明】

micro:bit プログラミングサイト

<https://makecode.microbit.org/#editor>

にアクセスしてプログラミングします。

- ①ブロックを組み合わせてプログラミング。
- ②確認用シミュレータで確認する。
- ③USB接続したmicro:bit にプログラムをコピー。
- ④micro:bit 本体で動作させる。

（プログラムをPCにダウンロードして保存することもできます。）

※ダウンロード保存しておいたプログラムを読み込んで修正することもできます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

## 001 温度を表示する

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

ずっと

数を表示 温度 (°C)

数値で表示

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

## 002 明るさを表示する

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

ずっと

棒グラフを表示する 値 明るさ

最大値 255

一時停止 (ミリ秒) 1000

LEDで表示

iii コンソールを表示 シミュレーター

## 【説明】

### 001 温度を表示する

micro:bit が持っているセンサー（温度・光・傾き・磁気）を使うベースサンプルとして作りました。温度センサーの数値をLEDに表示します。他のセンサーの数値ブロックに「置き換え」てみてもよいですね。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

- ・・・使用するのは、センサー・LED
- ・・・分類をわかりやすくするための表記

2018/4/18作成

### 002 明るさを表示する

LEDにグラフ表示する方法を調べて作りました。センサーの数値→グラフ化。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

- ・・・使用するのは、センサー・LED

2018/4/18作成

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

## 003 ボタンで温度を表示する

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

ボタン B が押されたとき

数を表示 温度 (°C)

ボタンで動作

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

## 004 ボタンで明るさを表示する

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

ボタン A が押されたとき

棒グラフを表示する 値 明るさ

最大値 255

一時停止 (ミリ秒) 1000

表示を消す

||| コンソールを表示 シミュレーター

## 【説明】

### 003 ボタンで温度を表示する

「001 温度を表示する」をボタンで動作するように「置き換え」たものです。

「ずっと」→「ボタンAが押されたとき」。ボタンの使い方のベースです。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

．．．ボタン・センサー・LED

2018/4/18作成

### 004 ボタンで明るさを表示する

「002 明るさを表示する」のボタン表示パターンです。このように、いろいろな組み合わせを考えることができますね。1000ミリ秒（1秒）で表示が消えるようにしています。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

．．．ボタン・センサー・LED

2018/4/25作成

## 005 ボタンで明るさ・温度を表示する

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

ボタン A が押されたとき

棒グラフを表示する 値 明るさ

最大値 255

一時停止 (ミリ秒) 1000

表示を消す

ボタン B が押されたとき

数を表示 温度 (°C)

## 【説明】

### 005 ボタンで明るさ・温度を表示する

「003 ボタンで温度を表示する」と「004 ボタンで明るさを表示する」を組み合わせたものです。

複数のプログラムを組み合わせることができることを示したものです。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・センサー・LED

2018/4/18作成

# 006 暗くなるとLEDが点く

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |



センサーで動作

## 【説明】

### 006 暗くなるとLEDが点く

ボタンではなくセンサーの数値をきっかけにLEDの点き方を変えています。（条件分岐）

設定している「センサーの数値」を変えることで、変化する明るさ（感度）を変えることができます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・センサー・LED

2018/4/22作成

## 007 Happy Birthday を鳴らす

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

ボタン が押されたとき

- 音を鳴らす 高さ (Hz) 真ん中のソ 長さ 750
- 音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1/4 拍
- 音を鳴らす 高さ (Hz) 真ん中のラ 長さ 1 拍
- 音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
- 音を鳴らす 高さ (Hz) 上のド 長さ 1 拍
- 音を鳴らす 高さ (Hz) 真ん中のシ 長さ 2 拍
- 音を鳴らす 高さ (Hz) 真ん中のソ 長さ 750
- 音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1/4 拍
- 音を鳴らす 高さ (Hz) 真ん中のラ 長さ 1 拍
- 音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
- 音を鳴らす 高さ (Hz) 上のレ 長さ 1 拍
- 音を鳴らす 高さ (Hz) 上のド 長さ 2 拍

スピーカーやイヤホンをつなぐ

## 008 電子オルゴールを鳴らす

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

端子 P1 がタッチされたとき

- 音を鳴らす 高さ (Hz) 真ん中のソ 長さ 750
- 音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1/4 拍
- 音を鳴らす 高さ (Hz) 真ん中のラ 長さ 1 拍
- 音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍
- 音を鳴らす 高さ (Hz) 上のド 長さ 1 拍
- 音を鳴らす 高さ (Hz) 真ん中のシ 長さ 2 拍

P1端子とGND端子の間に電流が流れると動作

## 【説明】

### 007 Happy Birthday を鳴らす

micro:bit が音を出力できることを示しているサンプルです。楽譜を見ながらブロックをつなげていきます。四分音符=1000なので、選択肢にない符点八分音符を750で入力（「計算」の数値入力ブロックで）しています。micro:bit 単体にはスピーカーがないので、P0端子とGND端子にスピーカーやイヤホンをつないで確認します。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

・・・出力（この場合は音声として）

2018/4/23作成

2019/1/3修正

### 008 電子オルゴールを鳴らす

P1端子とGND端子に導線をつなぎ、その間に「電流が流れたら」音を鳴らすように置き換えたものです。

P0端子とGND端子にスピーカーをつないでいても、P1端子とGND端子のタッチが別に動作するか確認したものです。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

・・・センサー・出力

2018/4/23作成

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

## 009 テスターで音を鳴らす



P1端子とGND端子の間に電流が流れると動作

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

## 010 明るさで音が変わるテルミン



明るさセンサーで動作

## 【説明】

### 009 テスターで音を鳴らす

「008 電子オルゴールを鳴らす」をシンプル化したものです。P1端子とGND端子に導線をつなぎ、その間に「電流がながれたら」音が鳴ります。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

．．．センサー・出力

2018/4/23作成

### 010 明るさで音が変わるテルミン

導線から明るさセンサーに置き換えたものです。初期の電子楽器「テルミン」のように手を明るさセンサーにかざすことで、音の高さが変わるようにしています。

明るさの数値を変えることで、変化の度合い（感度）を変えることができます。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

．．．センサー・出力

2018/4/23作成

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

## 011 ストップウォッチ



ボタンにより動作



## 【説明】

### 011 ストップウォッチ

ボタン操作で時間を測るようにしています。

変数「停止」「秒数」を作り、「停止」が1のときは秒数に加算、「停止」が0のときは加算しないようにしています。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・LED

2018/5/6作成

|   |   |   |   |
|---|---|---|---|
| ポ | セ | シ | 出 |
| 通 |   | 才 |   |

## 012 明るさで秒数計測

The screenshot shows a Scratch script with the following blocks:

- 最初だけ (Only at the beginning):**
  - 変数 秒数 を 0 にする
  - 変数 停止 を 1 にする
  - 文字列を表示 "Hello!"
  - 表示を消す
  - 文字列を表示 "Ready!"
  - 表示を消す
  - 数を表示 秒数
- ボタン A+B が押されたとき (When button A+B is pressed):**
  - 変数 秒数 を 0 にする
  - 変数 停止 を 1 にする
  - 文字列を表示 "Reset!"
  - 表示を消す
  - 数を表示 秒数
- ずっと (Forever loop):**
  - もし 停止 = 0 ならくりかえし
  - もし 明るさ < 最初の明るさ - 20 なら
    - 変数 停止 を 1 にする
    - 数を表示 秒数
  - でなければ
    - 一時停止 (ミリ秒) 1000
    - 変数 秒数 を 1 だけ増やす
    - 数を表示 秒数

This view shows a detailed look at the script blocks:

- ボタン A+B が押されたとき:**
  - 変数 秒数 を 0 にする
  - 変数 停止 を 1 にする
  - 文字列を表示 "Reset!"
  - 表示を消す
  - 数を表示 秒数
- ボタン A が押されたとき:**
  - 変数 最初の明るさ を 明るさ にする
  - 変数 停止 を 0 にする
- ボタン B が押されたとき:**
  - 変数 停止 を 1 にする

明るさ変化で動作

## 【説明】

### 012 明るさで秒数計測

懐中電灯の灯りをセンサーにあてておき変数「最初の明るさ」にしています。ボタンでスタートし、何かが灯りをさえぎる（前を横切る）までを計測できないかを考えたプログラムです。

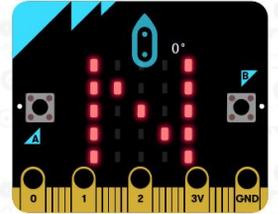
|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

．．．ボタン・センサー・LED

2018/5/6作成

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

## 013 コンパス (Compass)



検索...
🔍

- 🏠 基本
- 🎯 入力
- 🎵 音楽
- 💡 LED
- 📶 無線
- 🔄 ループ
- 🧠 論理
- 📄 変数
- 🧮 計算
- 📦 高度なブロック

```

ずっと
もし < 315 < 方角 (°) または < 方角 (°) < 45 < なら
  文字列を表示 "N"
でなければもし < 45 < 方角 (°) かつ < 方角 (°) < 135 < なら
  文字列を表示 "E"
でなければもし < 135 < 方角 (°) かつ < 方角 (°) < 225 < なら
  文字列を表示 "S"
でなければもし < 225 < 方角 (°) かつ < 方角 (°) < 315 < なら
  文字列を表示 "W"
でなければ
  表示を消す
          
```

磁気センサーで動作

## 【説明】

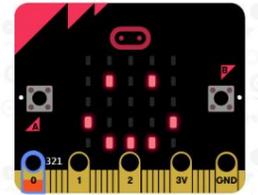
### 013 コンパス (Compass)

micro:bit の磁気センサーでコンパスを作りました。

NEWS (東西南北) をアルファベットで表示します。

注意 . . . 最初に磁気センサーの補正のため、LEDが全部点くようにぐるぐるまわします。 TILT TO FILL SCREEN→うまくいくと下のマークが出ます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |



. . . センサー・LED

2018/5/14作成

## 014 2台で送受信

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

The screenshot shows the Scratch environment with a Makeblock board on the left. The code blocks on the right are as follows:

- 最初だけ** (When first clicked):
  - 無線のグループを設定 (無線のグループを設定 1)
  - ボタン A が押されたとき (無線で数値を送信 0)
- 無線で受信したとき** (無線で受信したとき receivedNumber):
  - もし receivedNumber = 0 なら
    - 文字列を表示 "Receive!"
    - 一時停止 (ミリ秒) 1000
    - 表示を消す

Bluetoothによる通信

## 015 水準器

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

傾きセンサーで計測

The screenshot shows the Scratch environment with a Makeblock board on the left. The code blocks on the right are as follows:

- 最初だけ** (When first clicked):
  - 変数 x を 2 にする
  - 変数 y を 2 にする
- ずっと** (Forever loop):
  - 消灯 x x y y
  - 変数 x を -1 - 加速度 x + 1000 ÷ 400 にする
  - 変数 y を -1 - 加速度 y + 1000 ÷ 400 にする
  - 点灯 x x y y

## 【説明】

### 014 2台で送受信

micro:bit が持つBluetoothの通信機能をテストしたものです。通信を受け取ったら文字を表示します。（「receivedNumber」は、無線を受け取ったらのブロックからとり出します）

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・LED・通信

2018/5/17作成

### 015 水準器

micro:bit の傾きセンサーをテストしたものです。

最初x,yが2,2なのは、LEDの表示位置が0~4のためです（2,2が真ん中）。傾きが変わると、それに合わせて点灯する位置を変えるサンプルです。



・・・センサー・LED

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

2018/5/21作成

# 016 傾きメーター

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |

検索...

基本

入力

音楽

LED

無線

ループ

論理

変数

計算

高度なプロ...

ずっと

変数 Y を 0 にする

変数 X を -1 に設定し、加速度 X を 1000 で 400 で割る

くりかえし 5 回

点灯 x X y Y

変数 Y を 1 だけ増やす

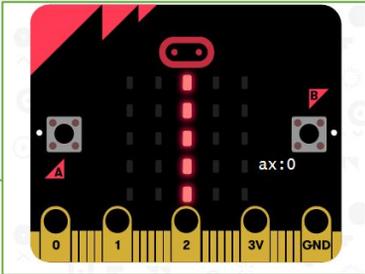
一時停止 (ミリ秒) 100

表示を消す

最初だけ

変数 X を 2 にする

傾きをLED表示



を -1 に設定し、加速度 X を 1000 で 400 で割る

## 【説明】

### 016 傾きメーター

micro:bit の傾きセンサーをテストしたものです。左右の傾きを検出してバーとして表示するようにしたものです。

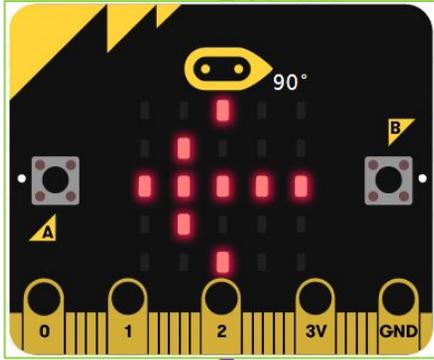
|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・LED・通信

2018/5/21作成

# 017 方位磁針

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |



検索...

基本

入力

ずっと

もし 315 < 方角 (°) または 方角 (°) < 45 なら

LED画面に表示

でなければもし 45 < 方角 (°) かつ 方角 (°) < 135 なら

LED画面に表示

でなければもし 135 < 方角 (°) かつ 方角 (°) < 225 なら

LED画面に表示

高度なブロック

でなければもし 135 < 方角 (°) かつ 方角 (°) < 225 なら

LED画面に表示

でなければもし 225 < 方角 (°) かつ 方角 (°) < 315 なら

LED画面に表示

でなければ

表示を消す

方角を → 表示

## 【説明】

### 017 方位磁針

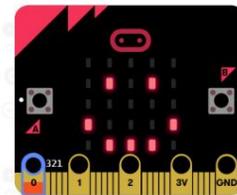
micro:bit の磁気センサーを使って、LED表示します。文字ではなくLEDのドットです。

5×5のLEDで、図形を表示する方法を模索したプログラムです。

注意・・・最初に磁気センサーの補正のため、LEDが全部点くようにぐるぐるまわします。 TILT TO FILL SCREEN →うまくいくと下のマークが出ます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・センサー・LED



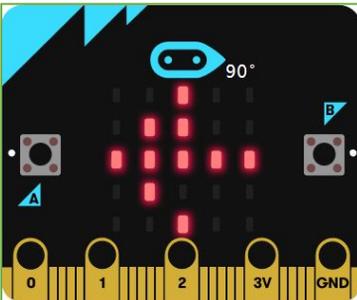
2018/5/23作成

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |

# 018 方位磁針16方位-1

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック



90°

16方位版

## 【説明】

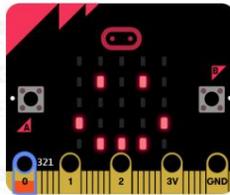
### 018 方位磁針16方位-1

micro:bit の磁気センサーを使って、LED表示します。文字ではなくLEDのドットです。

5×5のLEDで、図形を表示する方法を模索したプログラムですが、8方位版と違い、より矢印の図形を表すのが難しくなっています。

注意・・・最初に磁気センサーの補正のため、LEDが全部点くようにぐるぐるまわします。 TILT TO FILL SCREEN →うまくいくと下のマークが出ます。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |



・・・センサー・LED

(プログラムの続きがあります 018-2)

2018/5/24作成

# 018 方位磁針16方位-2

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

16方位版



**【説明】**

**018 方位磁針16方位-2**

018-1の続きになります。

2018/5/24作成

# 019 方位磁針16方位program分割-1

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

長いプログラムを分割

## 【説明】

### 019 方位磁針16方位program分割-1

micro:bit の磁気センサーを使って、LED表示します。文字ではなくLEDのドットです。

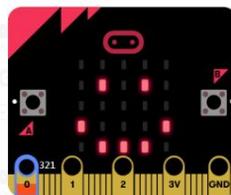
5×5のLEDで、図形を表示する方法を模索したプログラムですが、8方位版と違い、より矢印の図形を表すのが難しくなっています。

プログラムが長くなったため、分割ができないか工夫したものです。

「ずっと」を2つ作り、022「方位磁針16方位」を分割して入れました。「でなければ」に何も入れないことで、2つが成り立っているようです。

注意・・・最初に磁気センサーの補正のため、LEDが全部点くようにぐるぐるまわします。 TILT TO FILL SCREEN →うまくいくと下のマークが出ます。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |



・・・センサー・LED

(プログラムの続きがあります 019-2)

2018/5/24作成

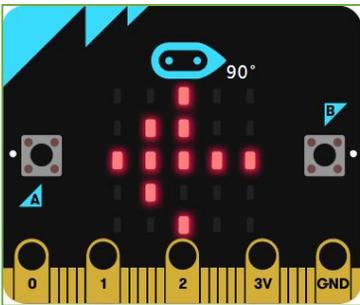
# 019 方位磁針16方位program分割-2

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

長いプログラムを分割



**【説明】**

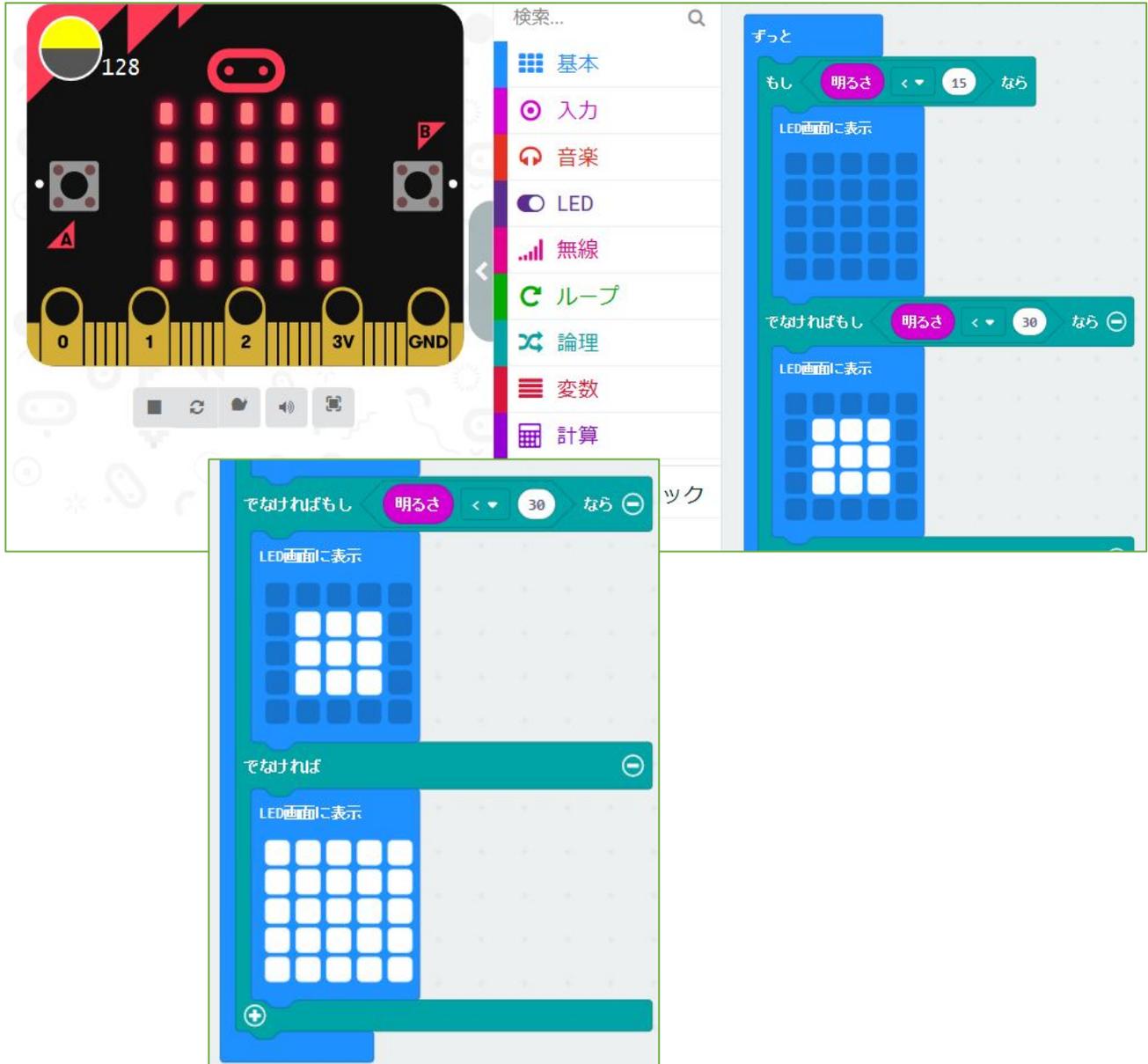
019 方位磁針16方位program分割-2

019-1の続きになります。

2018/5/24作成

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

## 020 暗くなったことを検知



## 【説明】

### 020 暗くなったことを検知

micro:bit の明るさセンサーを使って、明るさの変化によりLED表示を変化させます。

設定している明るさの値を変化させると、変化の度合い（感度）が変わります。

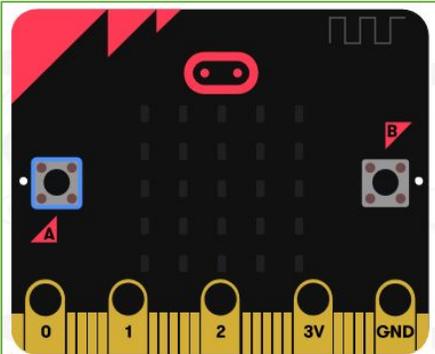
|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・センサー・LED

2018/5/28作成  
2019/1/6修正

## 021 相互にストップウォッチ

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |



受信したらスタート



Scratch-style block diagram for an Arduino Uno program:

- 最初だけ (Initial):**
  - 無線のグループを設定 (1)
  - 変数 秒数 を 0 にする
  - ボタン A が押されたとき: 無線で数値を送信 (0)
- ずっと (Forever):**
  - もし 進行 = 1 ならくりかえし
    - 一時停止 (ミリ秒) 1000
    - 変数 秒数 を 1 だけ増やす
    - 数を表示 秒数
  - もし 進行 = 0 なら
    - 表示を消す
    - 変数 秒数 を 0 にする
  - ボタン B が押されたとき: 無線で数値を送信 (1)
- 無線で受信したとき (receivedNumber):**
  - もし receivedNumber = 0 なら
    - 変数 進行 を 1 にする
  - でなければ
    - 変数 進行 を 0 にする

## 【説明】

### 021 相互にストップウォッチ

micro:bit のBluetooth通信機能を使って相互通信させるプログラムです。

ストップウォッチのプログラムを転用して試しています。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・LED・通信

2018/6/14作成

## 022 通信で合奏

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

The Scratch code for '022 通信で合奏' is as follows:

- 最初だけ** (Initially):
  - 無線のグループを設定 (1)
- ボタン A が押されたとき** (When button A is pressed):
  - 無線で数値を送信 (0)
  - 一時停止 (ミリ秒) (100)
  - 音を鳴らす 高さ (Hz) (真ん中のド) 長さ (1) 拍
- 無線で受信したとき receivedNumber** (When received wirelessly):
  - もし receivedNumber = 0 なら (If receivedNumber = 0 then):
    - 音を鳴らす 高さ (Hz) (真ん中のシ) 長さ (1) 拍

A callout box on the right states: **一対多送信可能** (One-to-many transmission possible).

## 023 明るさ変化を送信

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

The Scratch code for '023 明るさ変化を送信' is as follows:

- 最初だけ** (Initially):
  - 無線のグループを設定 (1)
  - LED画面に表示 (LED screen display)
- ずっと** (Forever loop):
  - もし 明るさ < 20 なら (If brightness < 20 then):
    - 無線で数値を送信 (0)
- 無線で受信したとき receivedNumber** (When received wirelessly):
  - もし receivedNumber = 0 なら (If receivedNumber = 0 then):
    - LED画面に表示 (LED screen display)
    - 一時停止 (ミリ秒) (100)
    - LED画面に表示 (LED screen display)

## 【説明】

### 022 通信で合奏

micro:bit のBluetooth通信機能が、一対多で動作するか確認するために作ったプログラムです。

無線のグループを同じにしたmicro:bit を多数配置しておき、一台をコンダクター（信号を送信する）とすることで、合奏もできるのではないかと考えました。スピーカーをP0端子とGND端子に接続しておきます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・出力・通信

2018/6/14作成

### 023 明るさ変化を通信

micro:bit のBluetooth通信機能を使って、一台の明るさセンサーの変化をもう一台に送信するテストプログラムです。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・センサー・LED・通信

2018/6/15作成

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

## 024 揺れたら送信

The Scratch code for '024 揺れたら送信' is as follows:

- 最初だけ (When first clicked):**
  - 無線のグループを設定 (1)
  - LED画面に表示 (Grid)
- ずっと (Forever loop):**
  - もし 加速度 X > 200 ならば
  - 無線で数値を送信 (0)
- 無線で受信したとき (When received number):**
  - もし receivedNumber = 0 ならば
  - LED画面に表示 (Grid)
  - 一時停止 (ミリ秒) 100
  - LED画面に表示 (Grid)

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

## 025 明るさ数値送信

The Scratch code for '025 明るさ数値送信' is as follows:

- 最初だけ (When first clicked):**
  - 無線のグループを設定 (1)
  - 変数 送信数値 を 0 にする
  - 変数 受信数値 を 0 にする
- ボタン A が押されたとき (When button A pressed):**
  - 変数 送信数値 を 明るさ にする
  - 無線で数値を送信 (送信数値)
- 無線で受信したとき (When received number):**
  - 変数 受信数値 を receivedNumber にする
  - 数を表示 (受信数値)

Two photographs of the micro:bit are included. The top one shows the '明るさ' (Brightness) sensor icon and a value of 128. The bottom one shows the LED display with a grid of lights.

## 【説明】

### 024 揺れたら送信

micro:bit のBluetooth通信機能を使って、一台の明るさセンサーの変化をもう一台に送信するテストプログラムです。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・センサー・LED・通信

2018/6/18作成

### 025 明るさ数値送信

micro:bit のBluetooth通信機能を使って、一台のセンサー数値自体をもう一台に送信できるかテストしたものです。

この場合は、ボタンを押したときの明るさセンサーの数値を送信しています。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・センサー・LED・通信

2018/6/18作成

# 026 ひらがな表示

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

ひらがなに見えるよう

## 【説明】

### 026 ひらがな表示

micro:bit は、日本語が表示できない、ということで、なんとか表示したように工夫できないかと考えたものです。

ドットで疑似的に表示しているだけですが、「できないことをなんとかできないか」と『工夫』した例」として作ってみました。

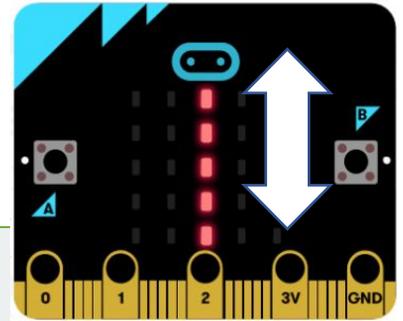
|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

．．．LED

2018/7/29作成

## 027 傾き（ピッチ）角度を表示

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |



検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

最初だけ

変数 高度 を 0 にする

ボタン A が押されたとき

変数 高度 を 90 - 傾斜 (°) ピッチ にする

くりかえし 3 回

数を表示 高度

文字列を表示 "/"

傾き（ピッチ）から表示

変数 高度 を 90 - 傾斜 (°) ピッチ にする

くりかえし 3 回

## 【説明】

### 027 傾き（ピッチ）角度を表示

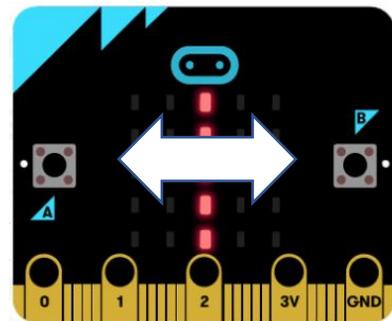
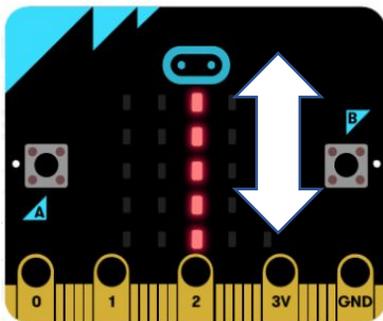
ボタンを押したときのmicro:bit 基板の角度（ピッチのほう。もう一つの角度がロール）を使って表示するプログラムです。

これは太陽の地面からの角度（南中高度）を測ることができないかというアイデアを実現するためのプログラムの一部です。

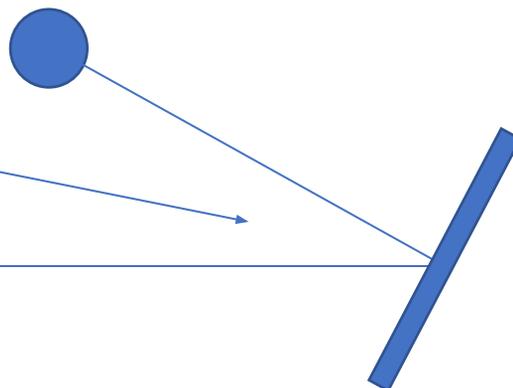
|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・センサー・LED

2018/7/29作成



この角度



## 028 明るさにより発信音の高さを変える

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |

The image shows a programming environment with a music block. The block is titled '音を鳴らす 高さ (Hz) 真ん中のド' and has a volume slider set to 50. A callout box points to the volume slider with the text '明るさで変化' (Change with brightness). The environment also shows a search bar and a list of categories: 基本, 入力, 音楽, LED, 無線, ループ, 論理, 変数, 計算.

## 【説明】

### 028 明るさにより発信音の高さを変える

ボタンを押したときの明るさセンサーの値により、音の高さを変えて出力するものです。

これは太陽の地面からの角度（南中高度）を測ることができないかというアイデアを実現するためのプログラムの一部です。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | ン | サ |
| ト |   | ー | ー |

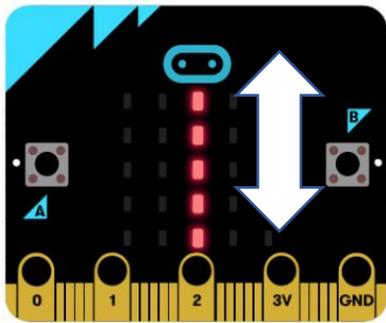
．．．ボタン・センサー・出力

2018/7/29作成

# 029 太陽高度計

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | 才 |   |

The screenshot shows the Scratch environment with a hardware setup on the left and code blocks on the right. The hardware is a microcontroller board with a potentiometer connected to pins 0, 1, 2, 3V, and GND. The code includes a 'when green flag clicked' event, a 'set height variable to 0' block, and a 'when button A+B is pressed' block with a 'wait 1 beat' block. A 'when button B is pressed' block sets the height variable to 90 and triggers a loop of 3 iterations. Each iteration displays the height variable and a sound block with a pitch of 77.



太陽に向けて使用

A close-up of the Scratch code blocks. It shows a series of 'if brightness is less than...' conditions. Each condition is linked to a specific sound block with a different pitch. The conditions are:
 

- 明るさ < 100 なら 真ん中のミ
- 明るさ < 125 なら 真ん中のファ
- 明るさ < 150 なら 真ん中のソ
- 明るさ < 200 なら 真ん中のラ
- 明るさ < 200 なら 真ん中のシ

A close-up of the Scratch code block: '変数 高度 を 90 に傾斜 (°) ピッチ を 77 にする'.

## 【説明】

### 029 太陽高度計

これは太陽の地面からの角度（南中高度）を測ることができないかというアイデアを実現するためのプログラムの一部です。

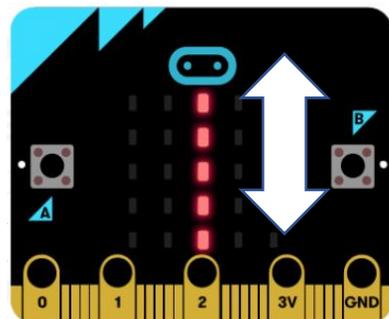
リアルタイムで明るさセンサーの値から音の高さを変えて出力するプログラムと、ボタンを押したときの micro:bit 基板の角度（ピッチのほう。もう一つの角度がロール）から太陽高度を表示するプログラムを組み合わせています。

太陽に向けて、一番明るい時を「太陽に向かった時」と考えて、そのときにボタンを押すことで測ろうとしています。

2つのプログラムが連動していないのもポイントです。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

- ・ ・ ・ ボタン ・ センサー
- ・ LED ・ 出力



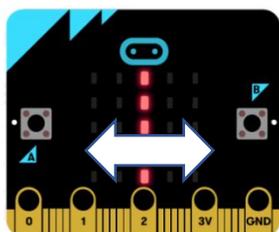
2018/8/6作成

## 030 ふりこ送信

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |



最下点に来た時をカウント



## 【説明】

### 030 ふりこ送信

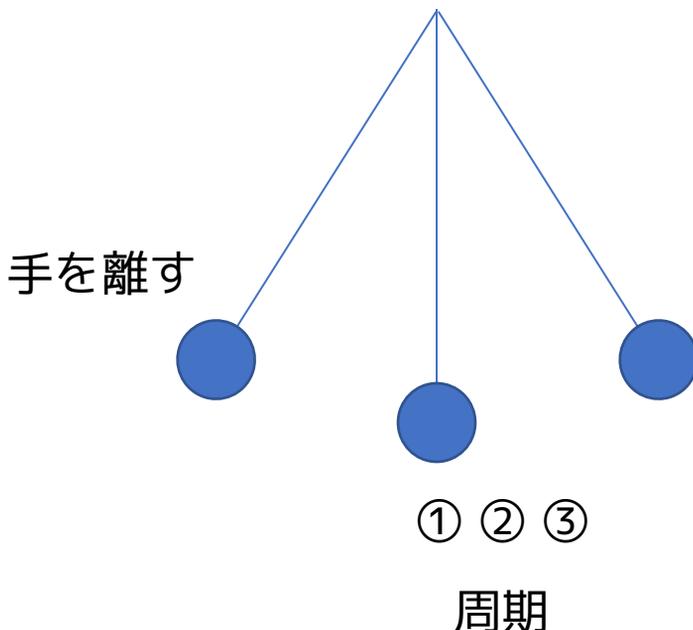
ふりこの周期を測るためのプログラムの一部です。「ふりこの最下点に来た時、ロールが0（水平）になる」ことを使っています。

水平になった回数をBluetoothで、もう一台に「0」を送信する部分です。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・センサー・LED・通信

2018/8/15作成



# 031 ふりこ受信 段階1

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

The top image shows a Scratch-like block editor with the following code:

- 最初だけ** (Initially):
  - 表示を消す (Clear display)
  - 無線のグループを設定 (1) (Set wireless group to 1)
  - 変数 時間 を 0 にする (Set time variable to 0)
  - 変数 Start を 0 にする (Set Start variable to 0)
  - 変数 回数 を 0 にする (Set count variable to 0)
- 無線で受信したとき receivedNumber** (When received wirelessly):
  - もし receivedNumber = 0 なら (If receivedNumber is 0):
    - 変数 回数 を 1 だけ増やす (Increase count by 1)
- ずっと** (Forever) loop:
  - もし 回数 = 1 なら (If count is 1):
    - 変数 Start を 1 にする (Set Start to 1)
  - でなければ (Otherwise):
    - もし Start = 1 なら (If Start is 1):
      - 一時停止 (ミリ秒) 100 (Pause 100 ms)
      - 変数 時間 を 100 だけ増やす (Increase time by 100)
      - 数を表示 時間 ÷ 100 (Show number: time / 100)

On the right, a breadboard is shown with a red LED and a push button. The LED is connected to pin 2, and the button is connected to pins 1 and 3. Pin 3V and GND are also labeled.

The bottom image shows a modified Scratch-like block editor with the following code:

- 最初だけ** (Initially):
  - 変数 回数 を 0 にする (Set count variable to 0)
  - ボタン A が押されたとき (When button A is pressed):
    - 表示を消す (Clear display)
    - 無線のグループを設定 (1) (Set wireless group to 1)
    - 変数 時間 を 0 にする (Set time variable to 0)
    - 変数 Start を 0 にする (Set Start variable to 0)
    - 変数 回数 を 0 にする (Set count variable to 0)
- ずっと** (Forever) loop:
  - もし 回数 = 1 なら (If count is 1):
    - 変数 Start を 1 にする (Set Start to 1)
  - でなければ (Otherwise):
    - もし Start = 1 なら (If Start is 1):
      - 一時停止 (ミリ秒) 100 (Pause 100 ms)
      - 変数 時間 を 100 だけ増やす (Increase time by 100)
      - 数を表示 時間 ÷ 100 (Show number: time / 100)
      - 表示を消す (Clear display)

## 【説明】

### 031 ふりこ受信 段階1

ふりこの周期を測るためのプログラムの一部です。

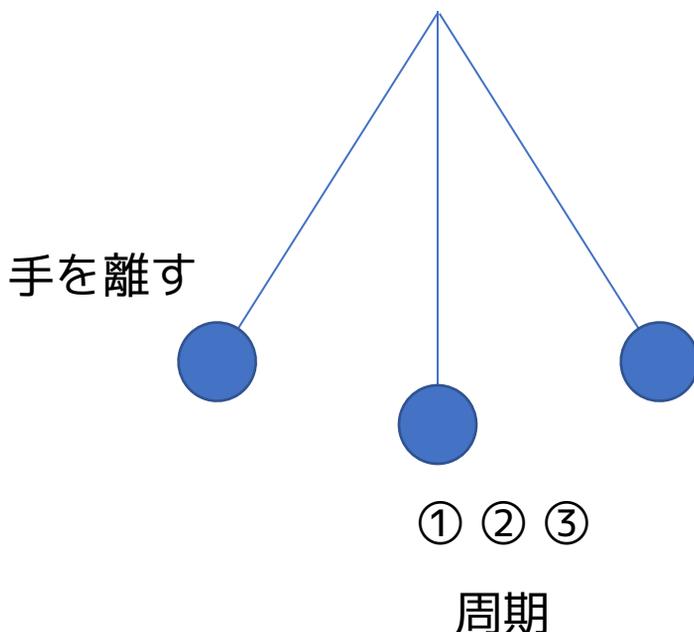
受信プログラムの原型で、受信したら秒数カウントが始まるようにしています。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・センサー・LED

(段階2があります)

2018/8/15作成



# 032 ふりこ受信 段階2

|   |   |   |   |
|---|---|---|---|
| ボ | セ | シ | 出 |
| 通 |   | オ |   |

The diagram shows the following block structure:

- 最初だけ (Initially):**
  - 表示を消す (Clear display)
  - 無線のグループを設定 1 (Set wireless group to 1)
  - 変数 時間 を 0 にする (Set time variable to 0)
  - 変数 Start を 0 にする (Set Start variable to 0)
  - 変数 回数 を 0 にする (Set count variable to 0)
- ボタン A が押されたとき (When button A is pressed):**
  - 表示を消す (Clear display)
  - 無線のグループを設定 1 (Set wireless group to 1)
  - 変数 時間 を 0 にする (Set time variable to 0)
  - 変数 Start を 0 にする (Set Start variable to 0)
  - 変数 回数 を 0 にする (Set count variable to 0)
- 無線で受信したとき receivedNumber (When received via wireless):**
  - もし receivedNumber = 0 なら (If receivedNumber is 0)
    - 変数 回数 を 1 だけ増やす (Increase count by 1)
- ずっと (Forever loop):**
  - もし 回数 = 1 なら (If count is 1)
    - 変数 Start を 1 にする (Set Start to 1)
  - でなければ (Otherwise)
    - もし Start = 1 なら (If Start is 1)
      - 一時停止 (ミリ秒) 100 (Pause 100 ms)
      - 変数 時間 を 100 だけ増やす (Increase time by 100)
      - 数を表示 時間 ÷ 100 (Show number: time / 100)
      - 表示を消す (Clear display)

This section provides a closer look at the logic within the loops:

- 無線で受信したとき receivedNumber:**
  - もし receivedNumber = 0 なら (If receivedNumber is 0)
    - 変数 回数 を 1 だけ増やす (Increase count by 1)
- ずっと (Forever loop):**
  - もし 回数 = 1 なら (If count is 1)
    - 変数 Start を 1 にする (Set Start to 1)
  - でなければ (Otherwise)
    - もし Start = 1 なら (If Start is 1)
      - 一時停止 (ミリ秒) 100 (Pause 100 ms)
      - 変数 時間 を 100 だけ増やす (Increase time by 100)
      - 数を表示 時間 ÷ 100 (Show number: time / 100)
      - 表示を消す (Clear display)

## 【説明】

### 032 ふりこ受信 段階2

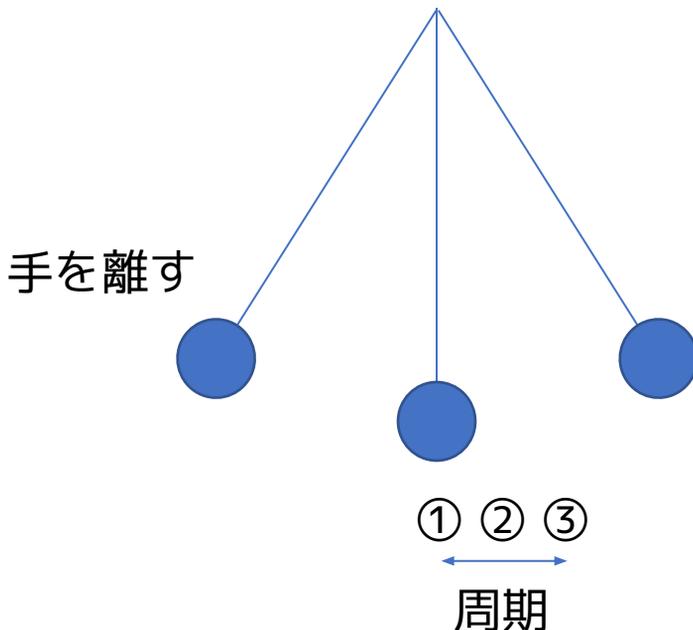
ふりこの周期を測るためのプログラムの一部です。  
030「ふりこ送信」と組み合わせます。

受信プログラムは、送られてきた「最下点に来た回数」が「1」から秒数をスタートして「3」になったらストップします。

(ふりこを離して、最下点1回目～3回目の間が周期になる)

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・センサー・LED・通信



2018/8/15作成

## 033 文字列を送信

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |



送信部のみの部分

受信部のみの部分

## 【説明】

### 033 文字列を送信

「文字列そのものを送信できませんか」という質問からつくってみたものです。

一台のmicro:bit から「文字列」を送信して、受信側のmicro:bit では、「送られてきた文字列」を表示します。

送受信部が両方入っていますが、分割して2台にそれぞれプログラムを入れても動作します。

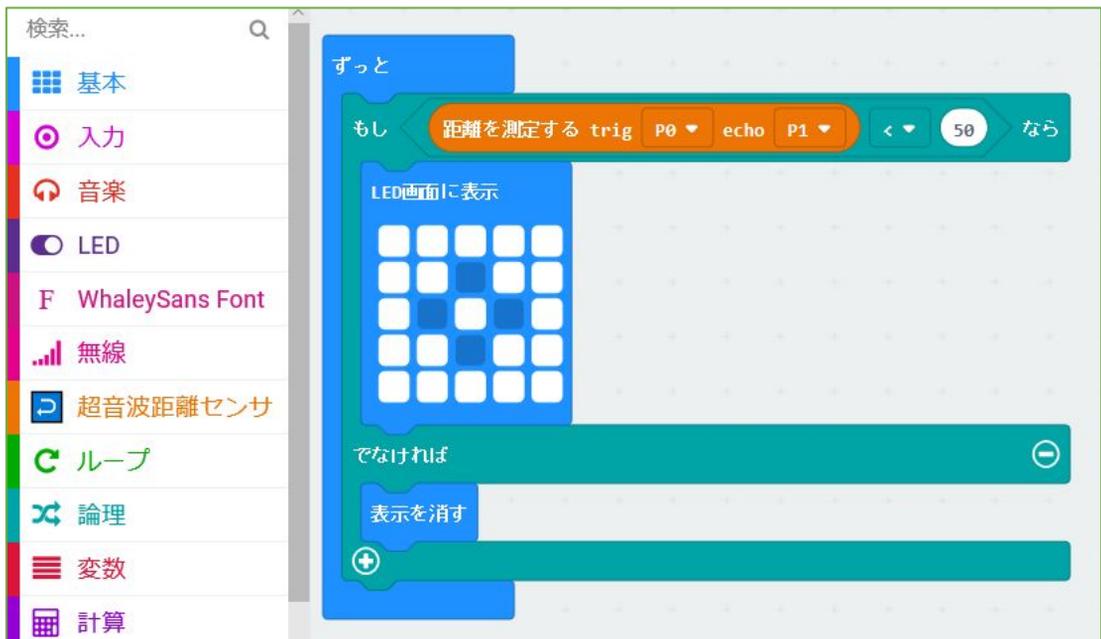
|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・LED・通信

2018/8/27作成

## 034 接近したら点灯（距離センサー）

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |



近づいたらLED表示

ワンタッチ距離センサー（TFabWorks）



## 【説明】

### 034 接近したら点灯（距離センサー）

<https://tfabworks.com/product/tfw-ds1/>

端子にワンタッチではめこむタイプのオプションを使ってみました。

<https://makecode.microbit.org/45600-49587-10845-13475>

（距離センサー(pxt-uds版)）を参照して変更しました。距離の数値を変更すると、反応する距離を変更することができます。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

．．．センサー・LED・オプション

2018/9/24作成

## 035 スイッチを制御（制御スイッチ）

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |



micro:bit用プログラム制  
御スイッチセット  
電磁石版（TFabWorks）

## 【説明】

### 035 スイッチを制御（制御スイッチ）

<https://tfabworks.com/product/mb-set-sw2/>

を使用

端子にワンタッチではめこむタイプのオプションを使ってみました。

フィルムターミナルを懐中電灯の乾電池の接触部分のすきまにはさみこむことで、P1端子から「オンオフ」を制御することができます。懐中電灯のスイッチを入れても、ターミナルが「オフ」なら点きません。

micro:bit のAボタンを押すと出力がターミナルを「オン」にするので、懐中電灯が点くわけです。

乾電池で動くものならいろいろ制御できそうですね。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

．．．ボタン・出力・オプション

2018/9/27作成  
2021/1/ 4修正

## 036 リモコン・スイッチで点灯（制御スイッチ）

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |



micro:bit用プログラム制御スイッチ  
セット  
電磁石版（TFabWorks）



## 【説明】

### 036 リモコン・スイッチで点灯（制御スイッチ）

<https://tfabworks.com/product/mb-set-sw2/>

端子にワンタッチではめこむタイプのオプションを使ってみました。

035「スイッチを制御」を通信機能で、もう一台のmicro:bit から「オンオフ」するように変えたものです。こうすることで離れた場所にあるものも制御することができます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

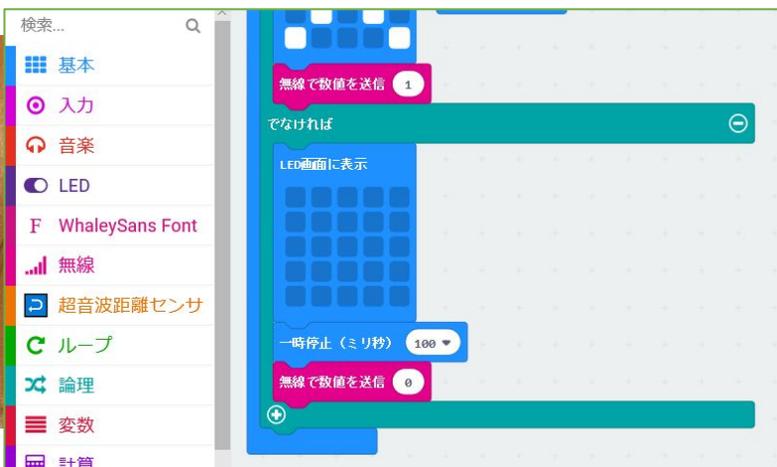
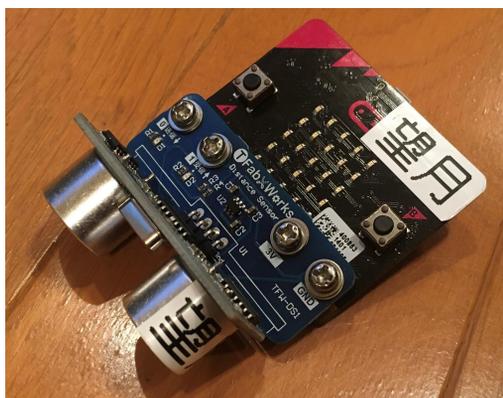
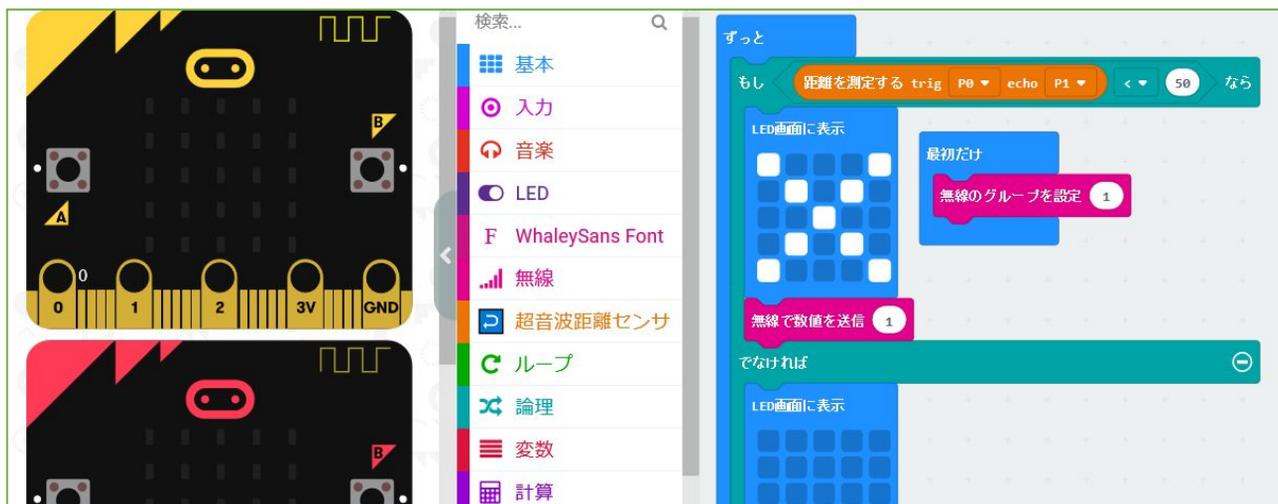
・・・ボタン・出力・通信・オプション

2018/9/27作成  
2021/1/ 4修正

# 037 人感センサー（距離センサー・制御スイッチ）-1

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |

送信部



ワンタッチ距離センサー（TFabWorks）

## 【説明】

### 037 人感センサー（距離センサー・制御スイッチ）-1

<https://tfabworks.com/product/tfw-ds1/>

一台のmicro:bit の端子にワンタッチではめこむタイプの距離センサーを使います。

(<https://makecode.microbit.org/45600-49587-10845-13475>  
(距離センサー(pxt-uds版)) を参照して変更)

もう一台のmicro:bit の端子にワンタッチではめこむタイプの制御スイッチ経由で懐中電灯をつなぎます。

036「リモコン・スイッチを制御」と距離センサーを組み合わせることで近づいたら懐中電灯が点くようになります。数値を変えると、感じる距離を変えることができます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

- ・ ・ ・ ボタン・センサー・LED・出力
- ・ 通信・オプション

(プログラムの続きがあります 043-2)

2018/10/1作成

# 037 人感センサー（距離センサー・制御スイッチ）-2

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | 才 |   |

受信部

```
最初だけ
無線のグループを設定 1
デジタルで出力する端子 P1 値 0

無線で受信したとき receivedNumber
もし receivedNumber = 1 ならば
  デジタルで出力する端子 P1 値 1
でなければ
  デジタルで出力する端子 P1 値 0
  一時停止 (ミリ秒) 100

ボタン A が押されたとき
無線で数値を送信 1

ボタン B が押されたとき
無線で数値を送信 0
```

micro:bit用プログラム制御  
スイッチセット  
電磁石版 (TFabWorks)



## 【説明】

### 037 人感センサー（距離センサー・制御スイッチ）-2

037-1の続きになります。

<https://tfabworks.com/product/mb-set-sw2/>

端子にワンタッチではめこむタイプのオプションを使っています。

2018/10/1作成

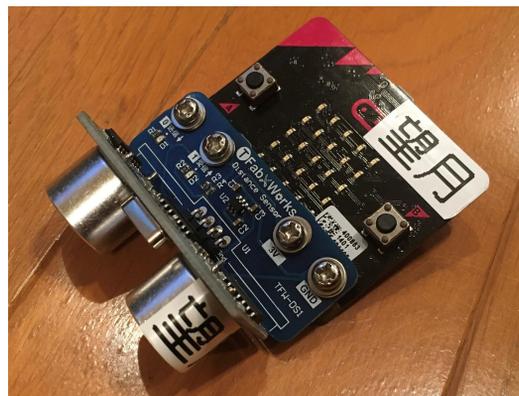
2021/01/4修正

## 038 距離を表示（距離センサー）

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |



ワンタッチ距離センサー（TFabWorks）



## 【説明】

### 038 距離を表示（距離センサー）

<https://tfabworks.com/product/tfw-ds1/>

一台のmicro:bit の端子にワンタッチではめこむタイプの距離センサーを使います。

距離センサーの数値自体を表示するようにしてみました。（

<https://makecode.microbit.org/45600-49587-10845-13475>（距離センサー(pxt-uds版)）を参照して変更）

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

．．．センサー・LED・オプション

2018/10/3作成  
2020/1/10修正

## 039 明るさグラフ

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

ずっと

変数 カウンター を 0 から 4 に変えてくりかえす

消灯 x 0 y カウンター

変数 カウンター を 0 から 明るさ ÷ 50 に変えてくりかえす

点灯 x 0 y カウンター

一時停止 (ミリ秒) 1000

バーの長さで表示

## 【説明】

### 039 明るさグラフ

センサーの値をバーの長さ（グラフ）表示（一番左の列に）させるテストとして作ったものです。値を他のセンサーに置き換えることもできますね。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

．．．センサー．LED

2018/10/3作成

2020/1/10修正

## 040 明るさグラフリアルタイム

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |



リアルタイムでバーの長さが変化



## 【説明】

### 040 明るさグラフリアルタイム

039「明るさグラフ」をリアルタイムで更新するようにしてみたものです。

別なセンサーの値にすれば、そのグラフを作ることもできます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

．．．センサー．LED

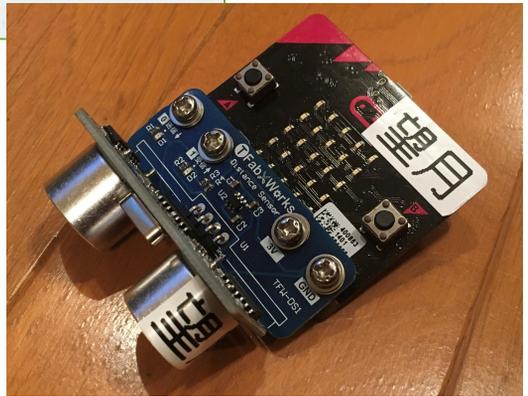
2018/10/3作成  
2020/1/10修正

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |

# 041 距離を測定（距離センサー）



ワンタッチ距離センサー（TFabWorks）



## 【説明】

### 041 距離を測定（距離センサー）

<https://tfabworks.com/product/tfw-ds1/>

一台のmicro:bit の端子にワンタッチではめこむタイプの距離センサーを使います。

Aボタンを押してから1秒間に近づいた距離を測るようになっています。（

<https://makecode.microbit.org/45600-49587-10845-13475>（距離センサー(pxt-uds版)）を参照して変更)

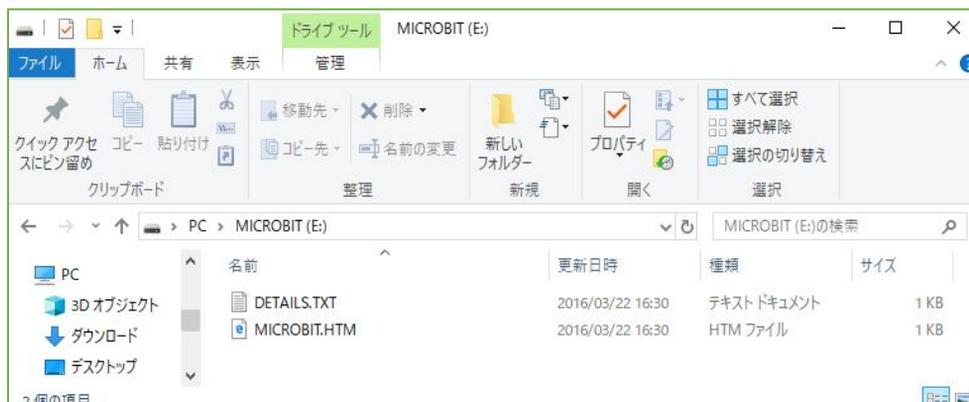
|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・センサー・LED・オプション

2018/10/4作成  
2020/1/11修正

## 042 micro:bit ファームウェアアップデート -1

USB接続して、DETAILS.TXTを確認



この場合 version 0241

```
# DAPLink Firmware - see https://mbed.com/daplink
Unique ID: 9900000041414e45002f000f000000200000000097969901
HIC ID: 97969901
Auto Reset: 1
Automation allowed: 0
Daplink Mode: Interface
Interface Version: 0241
Git SHA: fa4132987102c51c110751d8bdb8a74aeb7b071b
Local Mods: 1
USB Interfaces: MSD, CDC, HID
Interface CRC: 0xe369fb16
```

## 【説明】

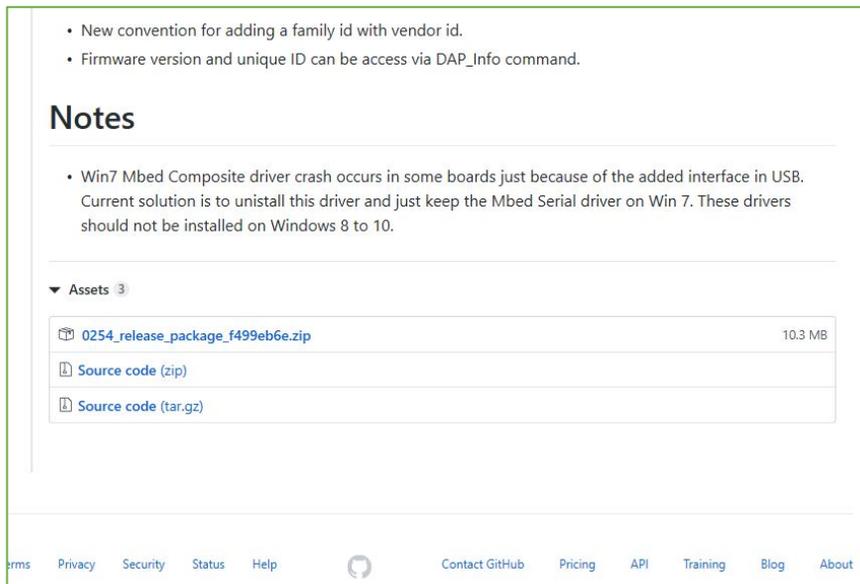
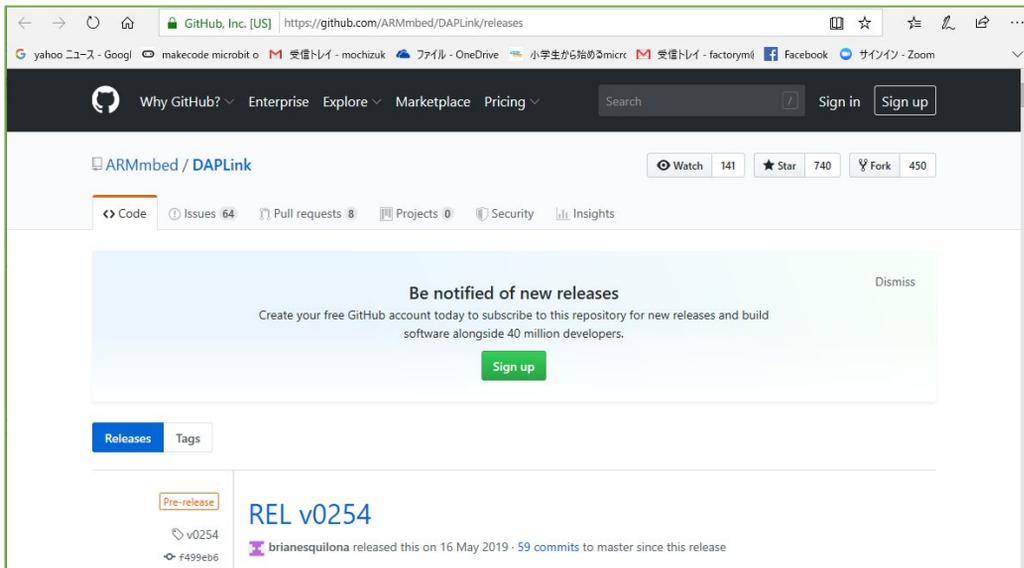
### 042 micro:bit ファームウェアアップデート

当時のベータサイトで作ったサンプルの表示が場合によって違っていたため、micro:bit 本体のファームウェアがアップデートされていないためかと考えて、アップデートした手順を記録したものです。

2018/10/8作成  
2020/1/11修正

## 042 micro:bit ファームウェアアップデート -2

<https://github.com/ARMmbed/DAPLink/releases>  
にアクセス



この場合 version 0254 のデータがあるので  
0254\_release\_package\_f499eb6e.zip をダウンロード

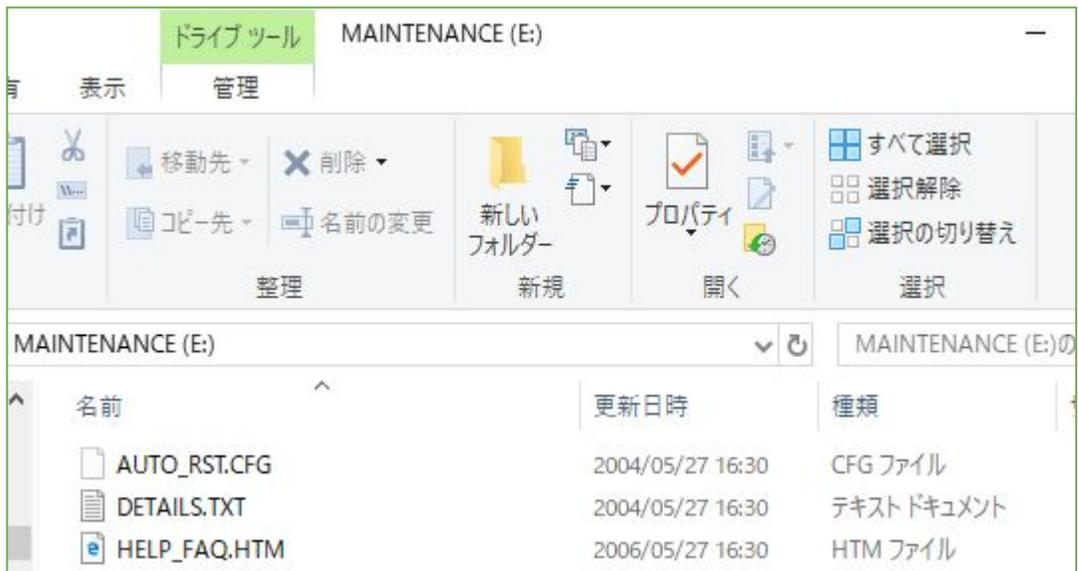
【メモ】

## 042 micro:bit ファームウェアアップデート -3

ZIPファイルを開いて、hexファイルを探す

|                                     |                                    |                            |        |   |
|-------------------------------------|------------------------------------|----------------------------|--------|---|
| <input type="checkbox"/>            | 0254_k20dx_rblnano_0x5000.bin      | BIN ファイル                   | 107 KB | 無 |
| <input type="checkbox"/>            | 0254_k20dx_twrke18f_0x8000.bin     | BIN ファイル                   | 95 KB  | 無 |
| <input type="checkbox"/>            | 0254_k20dx_twrkl28z72m_0x8000.bin  | BIN ファイル                   | 95 KB  | 無 |
| <input type="checkbox"/>            | 0254_k20dx_xdot_l151_0x8000.bin    | BIN ファイル                   | 95 KB  | 無 |
| <input type="checkbox"/>            | 0254_k26f_0x0000.bin               | BIN ファイル                   | 188 KB | 無 |
| <input type="checkbox"/>            | 0254_k26f_frdmk32w042_0x8000.bin   | BIN ファイル                   | 188 KB | 無 |
| <input type="checkbox"/>            | 0254_kl26z_0x0000.bin              | BIN ファイル                   | 95 KB  | 無 |
| <input checked="" type="checkbox"/> | 0254_kl26z_microbit_0x8000.hex     | Microsoft MakeCode bina... | 268 KB | 無 |
| <input type="checkbox"/>            | 0254_kl26z_nina_b1_0x8000.bin      | BIN ファイル                   | 95 KB  | 無 |
| <input type="checkbox"/>            | 0254_lpc11u35_0x0000.bin           | BIN ファイル                   | 60 KB  | 無 |
| <input type="checkbox"/>            | 0254_lpc11u35_6lowpan_borderrou... | BIN ファイル                   | 60 KB  | 無 |
| <input type="checkbox"/>            | 0254_lpc11u35_6lowpan_borderrou... | BIN ファイル                   | 60 KB  | 無 |
| <input type="checkbox"/>            | 0254_lpc11u35_6lowpan_borderrou... | BIN ファイル                   | 60 KB  | 無 |
| <input type="checkbox"/>            | 0254_lpc11u35_6lowpan_borderrou... | BIN ファイル                   | 60 KB  | 無 |

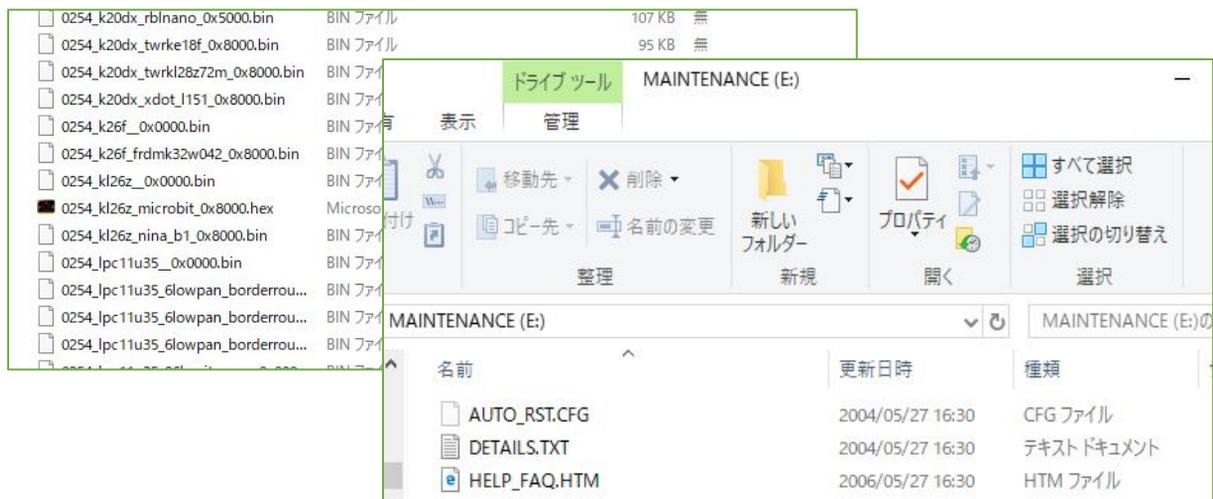
リセットボタンを押しながら基板をUSB接続すると  
メンテナンスモードが開く



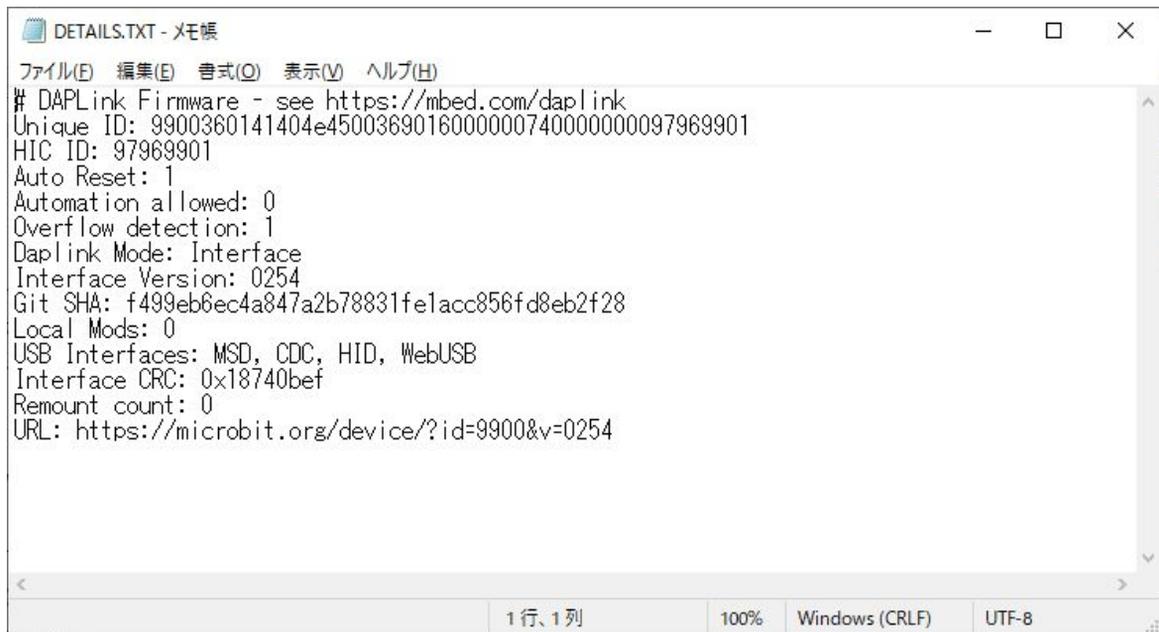
【メモ】

## 042 micro:bit ファームウェアアップデート -4

hexファイルをドロップすると再起動される



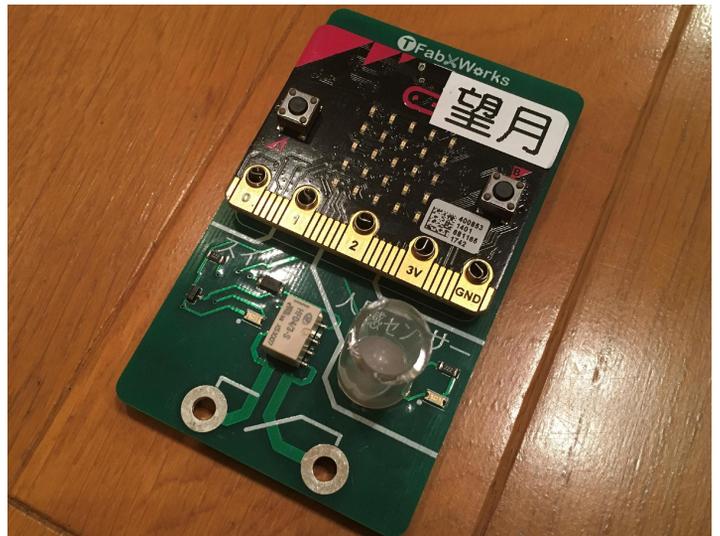
DETAILS.TXT でversionが0254になったことを確認



【メモ】

## 043 人感センサー

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | 才 |   |



## 【説明】

### 043 人感センサー

<https://tfabworks.com/product/tfw-rk2/>

を使用

<http://tfab.jp/rk2p>

を編集してつくりました。

人が近づいたら、チェックを表示  
遠ざかったら（いなかったら）×を表示します。  
Aボタン、Bボタンをチェック用にしています。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

．．．ボタン・センサー・LED・オプション

2019/1/16作成

## 044 ランダム数値表示

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 |   | オ |   |

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

ボタン A が押されたとき

数を表示 ランダムな数字を選択: 0 から 9 まで

## 【説明】

### 044 ランダム数値表示

ここまで「乱数」を使ってこなかったなので、初めて使ってみたものです。

ボタンを押すことで0～9の中から一つ表示します。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・LED

2019/1/17作成  
2020/8/25修正

# 045 じゃんけん

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

ボタン A が押されたとき

変数 乱数 を ランダムな数字を選択: 0 から 2 まで にする

数を表示 乱数

もし 乱数 = 0 なら

LED画面に表示

でなければもし 乱数 = 1 なら

LED画面に表示

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

LED画面に表示

でなければ

LED画面に表示

## 【説明】

### 045 じゃんけん

乱数を使ってじゃんけんをするものです。  
Aボタンを押したとき乱数で発生した0~2を使って、  
LEDで「パー」「チョキ」「グー」を表示します。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・LED

2019/1/17作成  
2020/1/11修正  
2020/8/25修正



パー



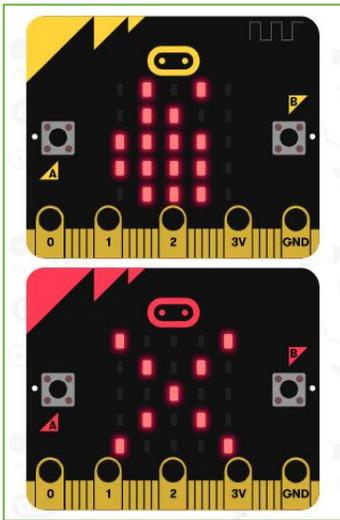
チョキ



グー

# 046 じゃんけん無線判定

|   |   |   |   |
|---|---|---|---|
| ボ | セ | シ | 出 |
| 通 |   | オ |   |



| 相手<br>自分 | 0 パー | 1 チョキ | 2 グー |
|----------|------|-------|------|
| 0 パー     | △    | ×     | □    |
| 1 チョキ    | □    | △     | ×    |
| 2 グー     | ×    | □     | △    |

□ 勝ち △ 引き分け × 負け

## 【説明】

### 046 じゃんけん無線判定

「045じゃんけん」を、相手の出した手を無線で読み取り、勝敗判定するようにしたものです。

乱数を発生させるときに、その数をBluetoothで送信するため、相手はその数値と自分が発生した数値（パー、チョキ、グーに該当）と比較して勝敗を判定します。二重の条件（自分の数値に対して送信されてきた相手の数値3パターンで判定）を設定することで、プログラムを短くしています。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

・・・ボタン・LED

2019/1/18作成

2020/1/12修正

2020/8/26修正



パー



チョキ



グー

## 047 磁気表示

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

The screenshot shows a block-based programming environment. On the left is a virtual breadboard with a yellow microcontroller and several red LEDs. A search bar at the top of the block palette contains the text "検索...". The block palette lists various categories: 基本 (Basic), 入力 (Input), 音楽 (Music), LED, 無線 (Wireless), ループ (Loop), 論理 (Logic), 変数 (Variables), 計算 (Math), and 高度なブロック (Advanced Blocks). On the right, a script area contains a blue "ずっと" (Forever) loop block. Inside the loop, there is a "数を表示" (Show Number) block with "磁力 (μT)" (Magnetic Force (μT)) selected and a multiplier of "x". Below it is a "一時停止 (ミリ秒)" (Pause (ms)) block set to "100".

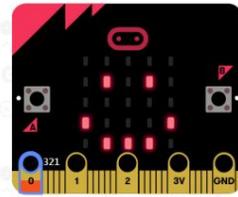
## 【説明】

### 047 磁気表示

理科の実験で作る電磁石の強さを測りたいと考えたものです。

磁気センサーでとらえた磁力の強さを数値表示します。  
注意・・・最初に磁気センサーの補正のため、LEDが全部点くようにぐるぐるまわします TILT TO FILL SCREEN →下のマークが出るようにします。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |



・・・ボタン・センサー・LED

2019/1/23作成



## 【説明】

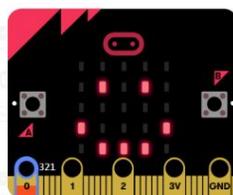
### 048 磁力計

「047磁気表示」を「002明るさを表示する」のグラフ表示のやり方を使って、作り変えたものです。

磁力が強くなるとLED上のグラフが明るくなるので、磁石を見つけたり、磁力の強さを測ることができるとかんがえました。

注意・・・最初に磁気センサーの補正のため、LEDが全部点くようにぐるぐるまわします TILT TO FILL SCREEN →下のマークが出るようにします。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 |   | オ |   |

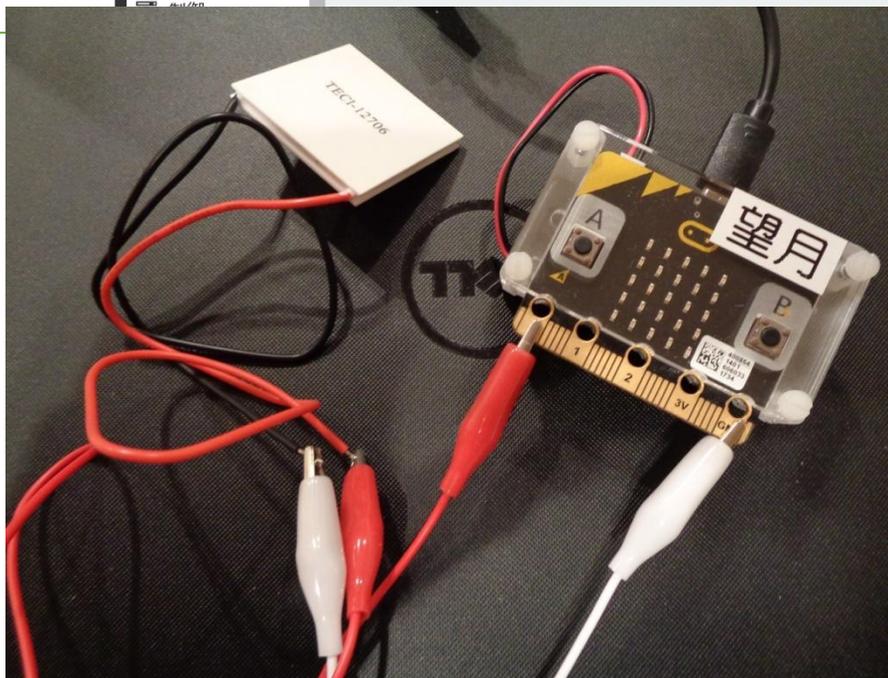


・・・ボタン・センサー・LED

2019/2/19作成

# 049 検流 (ペルチェ素子)

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |



## 【説明】

### 049 検流（ペルチエ素子）

端子P0に電流が入力されたかどうかを測るサンプルを作ってみました。

入力のためには、「ペルチエ素子（熱と電流を相互変換する）」を使用しました。

入力されない間は、×を表示し、ペルチエ素子の片面を指で温めると電流が発生し入力されるため、☑に表示が変わります。温度が下がると入力がなくなるため、×に戻ります。

※写真のようなペルチエ素子は数百円からあります。ペルチエ素子は、電流が流れることで片面から熱をもう片面に移動することで冷却加熱するものです。逆に温めたり冷やすことで電流を発生させます。

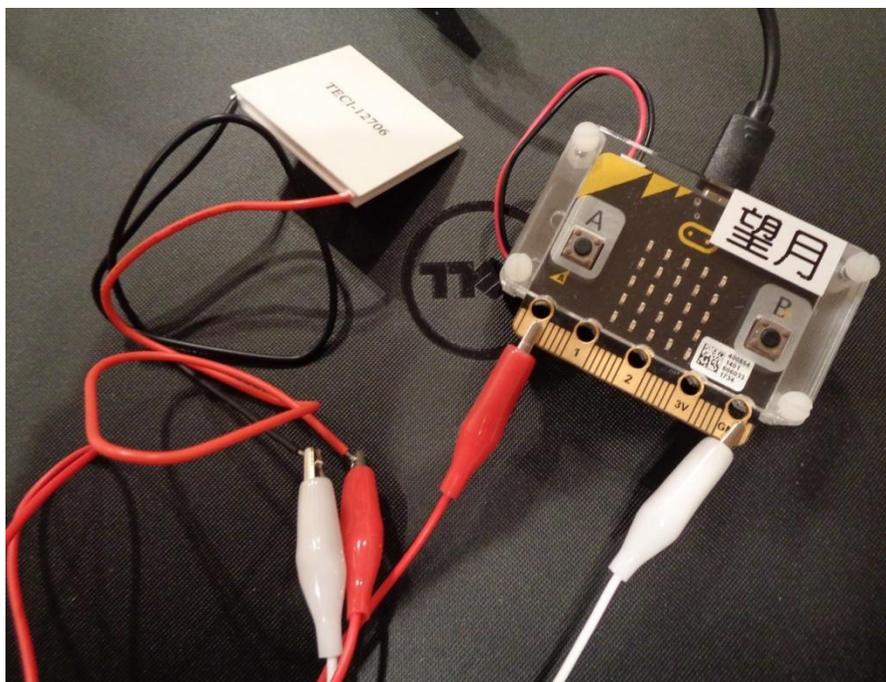
|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |

・・・LED・入力

2019/3/2作成

# 050 検流グラフ (ペルチェ素子)

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | 才 |   |



## 【説明】

### 050 検流グラフ（ペルチェ素子）

端子P0に電流が入力されたかどうかを測るサンプル049をグラフ表示にしたものです。感度調整は「最大値」で設定しています。

入力のためには、「ペルチェ素子（熱と電流を相互変換する）」を使用しました。

入力されない間は、×を表示し、ペルチェ素子の片面を指で温めると電流が発生し入力されるため、☑に表示が変わります。温度が下がると入力がなくなるため、×に戻ります。

※写真のようなペルチェ素子は数百円からあります。ペルチェ素子は、電流が流れることで片面から熱をもう片面に移動することで冷却加熱するものです。逆に温めたり冷やすことで電流を発生させます。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 | 入 | オ |   |

・・・LED・入力

2019/3/4作成  
2020/1/12修正

# 051 計算機-1

The diagram illustrates the logic of a calculator program in three stages:

- Initial State (Top):**
  - Initial values: 変数 ~する数 を 0 にする, 変数 ~される数 を 0 にする, 変数 計算履歴 を 0 にする, 変数 四則 を 0 にする.
  - Event: ボタン A が押されたとき.
    - Condition: もし 計算履歴 = 0 なら.
      - Action: 数を表示 ~される数.
      - Action: 変数 ~される数 を 1 だけ増やす.
    - Condition: もし 計算履歴 = 1 なら.
      - Condition: もし 四則 = 0 なら.
        - Action: LED画面に表示 (Calculator UI).
      - Condition: もし 四則 = 1 なら.
        - Action: LED画面に表示 (Calculator UI).
- Button A+B Pressed (Middle):**
  - Event: ボタン A+B が押されたとき.
    - Action: 変数 ~する数 を 0 にする.
    - Action: 変数 ~される数 を 0 にする.
    - Action: 変数 計算履歴 を 0 にする.
    - Action: 変数 四則 を 0 にする.
    - Action: 表示を消す.
- Button A Pressed (Bottom):**
  - Event: ボタン A が押されたとき.
    - Condition: もし 四則 = 1 なら.
      - Action: LED画面に表示 (Calculator UI).
    - Condition: もし 四則 = 2 なら.
      - Action: LED画面に表示 (Calculator UI).
    - Condition: もし 四則 = 3 なら.
      - Action: LED画面に表示 (Calculator UI).

※計算機-2に続きます。

## 【説明】

### 051 計算機-1

micro:bitで計算機をつくることができないか試したものです。

Aボタンの役割 . . .

数字を増やす、四則を変える

Bボタンの役割 . . .

今計算のどの段階か区切りをつける

A+Bボタン（同時押し）の役割 . . .

リセット

### 計算の進み方

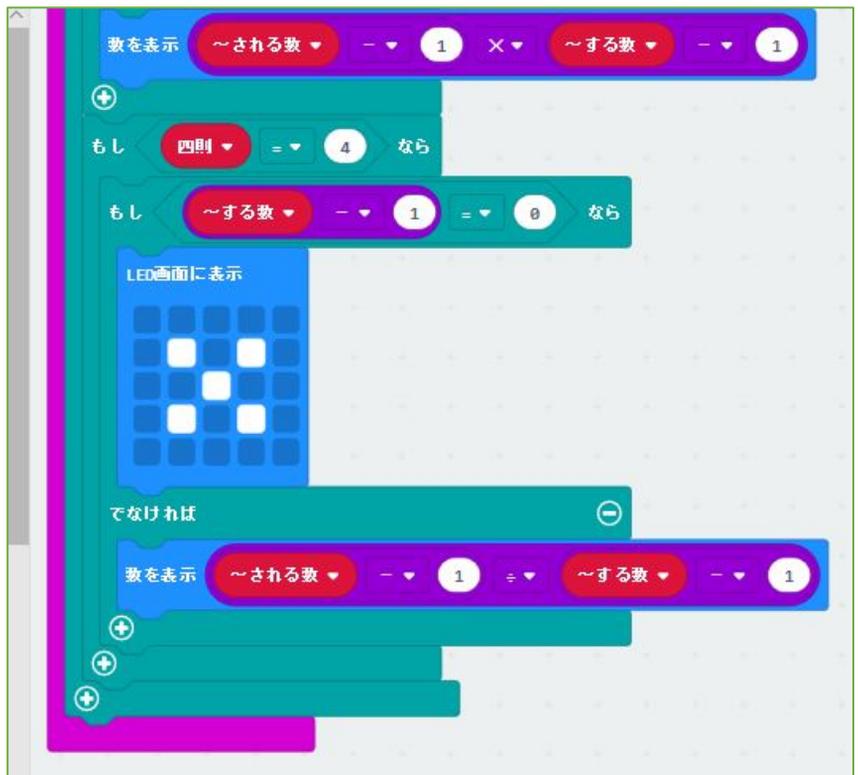
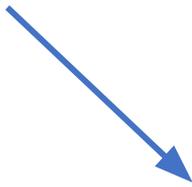
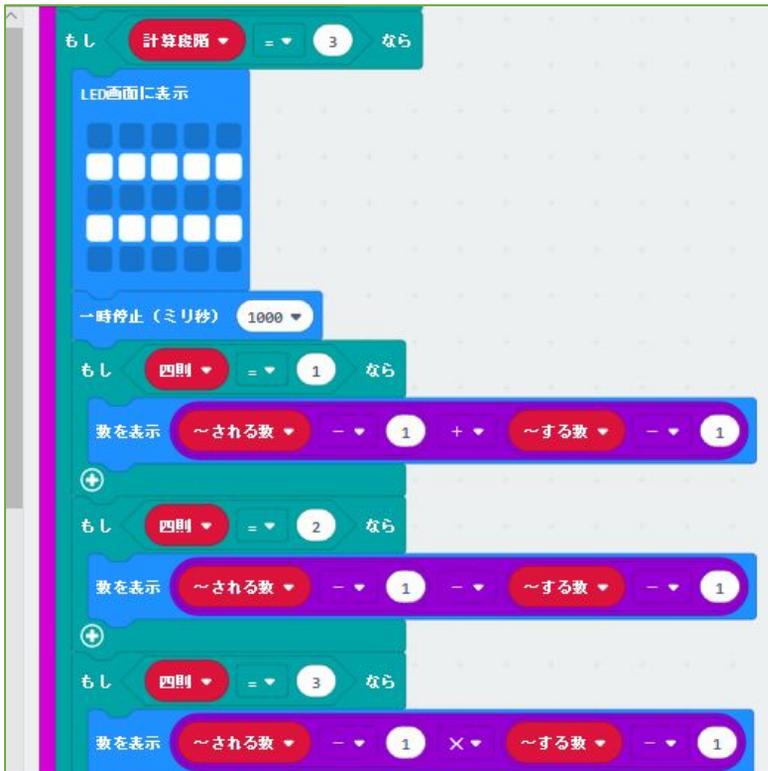
- ・ ~される数を決める。Aボタンを押すと数が増える。
- ・ Bボタンを押して、~される数を決定。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |

. . . ボタン・LED

※051 計算機-2 に続きます。

2019/3/4作成



## 【説明】

### 051 計算機-2

※051 計算機-1 の続きです。

- ・ Aボタンを押して、四則（ $+$   $-$   $\times$   $\div$ ）を変化させます。
- ・ Bボタンを押して、四則を決定。
  
- ・  $\sim$ する数を決める。Aボタンを押すと数が増える。
- ・ Bボタンを押して、 $\sim$ する数を決定。
  
- ・ Aボタンを押すと、 $=$ が1秒間表示され、計算結果が表示されます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |

・・・ボタン・LED

2019/3/4作成

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 | 入 | 才 |   |

## 052 計算機（関数）

※051 計算機と比較した変更部分(関数)のみ掲載しています。

The image shows the Scratch 'Edit Function' dialog box and several function blocks. The dialog box is titled 'Edit Function' and has a search bar and buttons for 'パラメーターを追加する', '文字列', '真偽値', 'Number', and 'LedSprite'. A blue arrow points to the '関数' (Function) category in the block palette. Below the dialog, there are four function blocks for arithmetic operations: '関数 たし算' (Addition), '関数 ひき算' (Subtraction), '関数 かけ算' (Multiplication), and '関数 わり算' (Division). Each block has a '数を表示' (Display Number) block with two input fields for numbers. To the right, a larger screenshot shows a sequence of 'もし' (If) blocks with conditions like '四則 = 1 なら' (If operation is 1) and corresponding function call blocks like '呼び出し たし算' (Call Addition).

## 【説明】

### 052 計算機（関数）

「051 計算機」の計算結果を表示する部分を、関数（処理をまとめたもの）に置き換えています。

関数は、「高度なブロック」→「f(x)関数」から作成します。

051の計算結果表示部分を、関数に入れて、表示する際は、関数を「呼び出し」て表示させるようにしました。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 | 入 | オ |   |

・・・ボタン・LED

2019/3/4作成

## 053 気温を小数表示

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | 才 |   |



## 【説明】

### 053 気温を小数表示

<https://tfabworks.com/product/tfw-en1/>

気温・湿度・気圧センサーを使用

以前拡張パーツ（Weather:bit）を使っていましたが、TFabWorksの新しいパーツに変えてプログラムしたものです。

<https://tfab.jp/stem>

専用ブロックを読み込んで編集しました。

（℃）で表示しますが、小数表示のほかに整数表示もできます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |

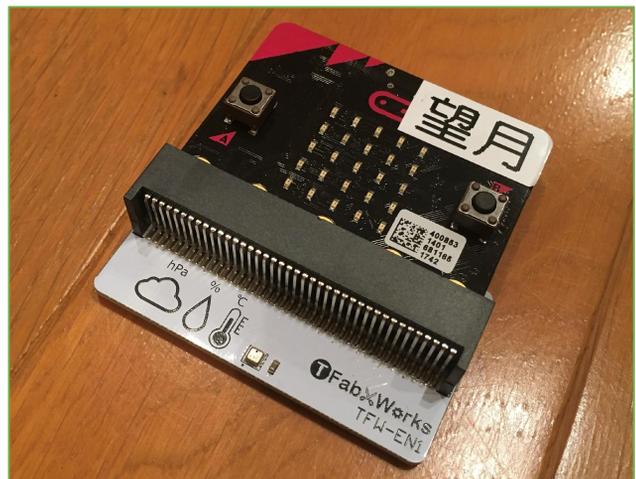
・・・ボタン・センサー・LED・オプション

2021/1/1作成

## 054 湿度を小数表示

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 | 入 | 才 |   |

The screenshot shows the micro:bit online editor interface. On the left is a virtual representation of the micro:bit board with pins labeled 0, 1, 2, 3V, and GND. The main workspace contains a block titled "ボタン A が押されたとき" (When button A is pressed). Inside this block, there are two sub-blocks: "数を表示" (Show number) with the text "湿度 [%] 小数" (Humidity [%] Decimal) and a dropdown menu set to "小数" (Decimal), and "文字列を表示" (Show text) with the text "%/". A callout box with a green border points to the "小数" dropdown and contains the text "小数・整数変更" (Change decimal/integer).



## 【説明】

### 054 湿度を小数表示

<https://tfabworks.com/product/tfw-en1/>

気温・湿度・気圧センサーを使用

以前拡張パーツ（Weather:bit）を使っていたが、TFabWorksの新しいパーツに変えてプログラムしたものです。

<https://tfab.jp/stem>

専用ブロックを読み込んで編集しました。

（%）で表示しますが、小数表示のほかに整数表示もできます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |

・・・ボタン・センサー・LED・オプション

2021/1/1作成

## 055 気圧を小数表示

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | 才 |   |



The screenshot shows the micro:bit JavaScript editor interface. On the left is a virtual representation of the micro:bit board with pins labeled 0, 1, 2, 3V, and GND. The main workspace contains a block titled "ボタン A が押されたとき" (When button A is pressed). Inside this block, there are two sub-blocks: "数を表示" (Show number) with the value "気圧[hPa] 小数" (Pressure[hPa] Decimal) selected, and "文字列を表示" (Show text) with the value "hPa/" (hPa/). A callout box with a green border and the text "小数・整数変更" (Change decimal/integer) points to the "小数" (Decimal) dropdown menu in the "数を表示" block. The left sidebar shows a search bar and a list of block categories: 基本 (Basic), 入力 (Input), 音楽 (Music), LED, 無線 (Wireless), STEM, ループ (Loops), 論理 (Logic), 変数 (Variables), 計算 (Math), TFabConnectBeta, and 高度なブロック (Advanced blocks).



## 【説明】

### 055 気圧を小数表示

<https://tfabworks.com/product/tfw-en1/>

気温・湿度・気圧センサーを使用

以前拡張パーツ（Weather:bit）を使っていましたが、TFabWorksの新しいパーツに変えてプログラムしたものです。

<https://tfab.jp/stem>

専用ブロックを読み込んで編集しました。

（hPa）で表示しますが、小数表示のほかに整数表示もできます。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |

・・・ボタン・センサー・LED・オプション

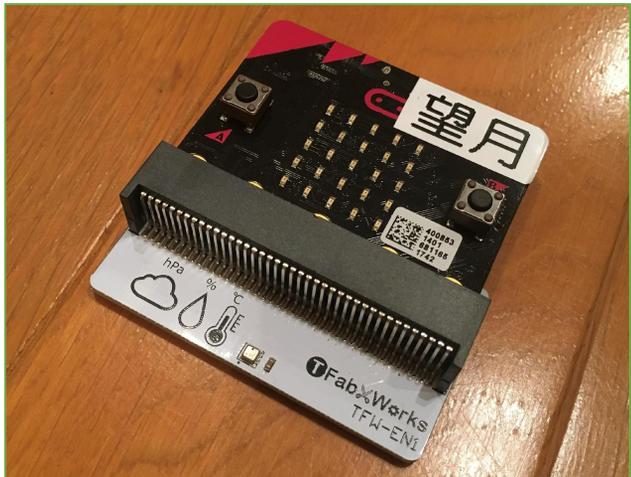
2021/1/1作成

# 056 気温・湿度・気圧を小数表示

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | 才 |   |



小数・整数変更



## 【説明】

### 056 気温・湿度・気圧を小数表示

<https://tfabworks.com/product/tfw-en1/>

気温・湿度・気圧センサーを使用

以前拡張パーツ（Weather:bit）を使っていましたが、TFabWorksの新しいパーツに変えてプログラムしたものです。

<https://tfab.jp/stem>

専用ブロックを読み込んで編集しました。

ボタンに割り当てて、それぞれの数値を表示させます。

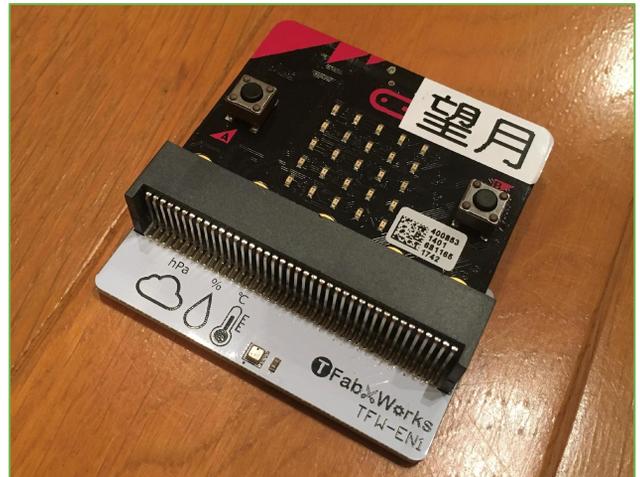
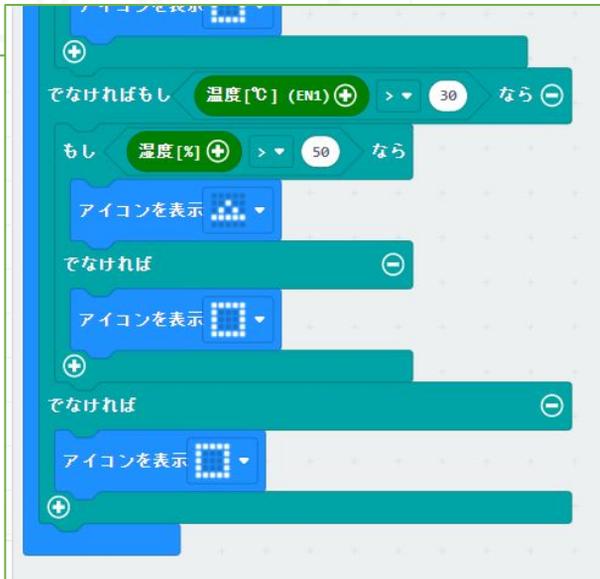
|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |

・・・ボタン・センサー・LED・オプション

2021/1/1作成

# 057 簡易熱中症計

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | 才 |   |



## 【説明】

### 057 簡易熱中症計

<https://tfabworks.com/product/tfw-en1/>

気温・湿度・気圧センサーを使用

以前拡張パーツ（Weather:bit）を使っていましたが、TFabWorksの新しいパーツに変えてプログラムしたものです。簡単な熱中症を予防する機械ができないかと考えたものです。気温が高くなる、湿度が高くなると、熱中症になりやすいと考え（あくまで簡易の考え方で）、表のように条件を考えました。

|      | 70% | 50% | 30% | 10% |
|------|-----|-----|-----|-----|
| 35°C | ×   | △   | ○   | —   |
| 30°C | △   | △   | ○   | —   |
| 25°C | ○   | ○   | ○   | —   |
| 20°C | —   | —   | —   | —   |

<https://tfab.jp/stem>

専用ブロックを読み込んで編集しました。

条件分岐を組み合わせて条件にあったら表示を出すようにしています。

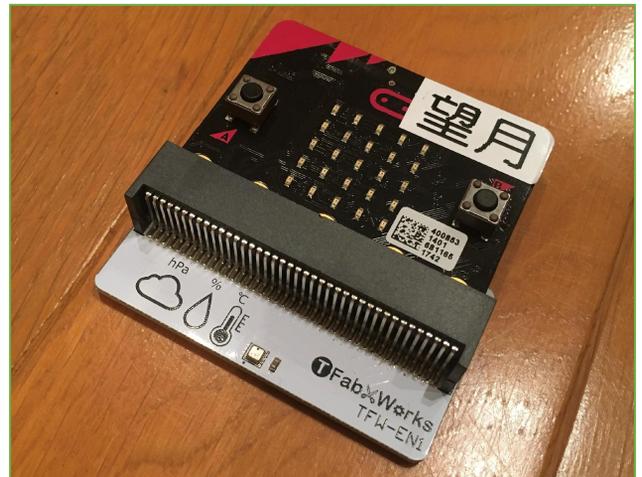
|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |

・・・ボタン・センサー・LED・オプション

2021/1/1作成

## 058 気温数値を送信

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | 才 |   |



※ 「059 気温数値を受信」とセットになります。

## 【説明】

### 058 温度数値を送信

<https://tfabworks.com/product/tfw-en1/>

オプションパーツ（拡張基板）である温度・湿度・気圧センサーをmicro:bit の端子部に差し込んで使います。

<https://tfab.jp/stem>

にアクセスすることで、専用の「STEM」ブロックから「温度」ブロックを使い、数値を送信します。

※「059 気温数値を受信」とセットになります。

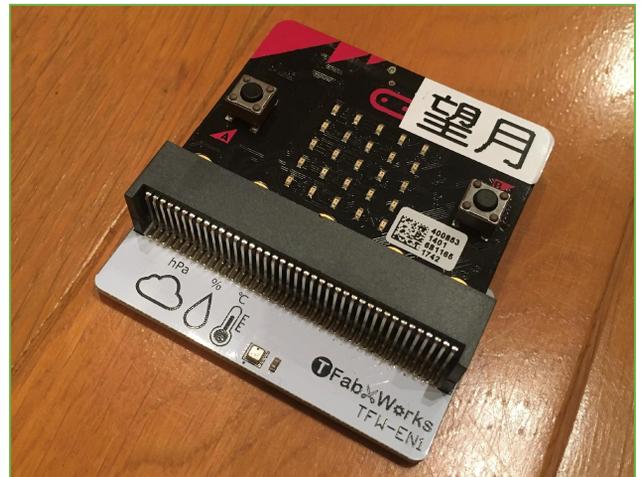
|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 | 入 | オ |   |

・・・センサー・LED・通信・オプション

2021/1/2作成

## 059 気温数値を受信

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |



※ 「058 気温数値を送信」とセットになります。

## 【説明】

### 059 温度数値を受信

<https://tfabworks.com/product/tfw-en1/>

オプションパーツ（拡張基板）である温度・湿度・気圧センサーをmicro:bit の端子部に差し込んで使います。

受け取った数値を温度として表示します。

※「058 気温数値を送信」とセットになります。

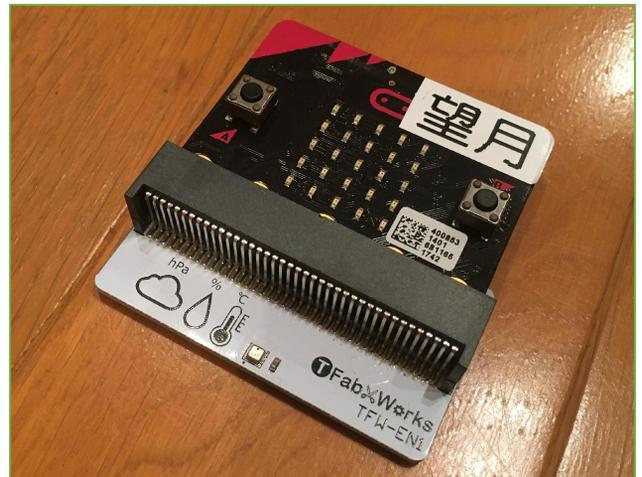
|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 | 入 | オ |   |

．．．LED・出力・通信

2021/1/2作成

## 060 温度・湿度を同時送信

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | 才 |   |



※「061 気温・湿度を同時送信」とセットになります。

## 【説明】

### 060 温度・湿度を同時送信

<https://tfabworks.com/product/tfw-en1/>

オプションパーツ（拡張基板）である温度・湿度・気圧センサーをmicro:bit の端子部に差し込んで使います。

<https://tfab.jp/stem>

にアクセスすることで、専用の「STEM」ブロックから「温度」「湿度」ブロックを使い、name（文字列）とvalue（数値）をセットで送信しています。

- ・ temp2=温度
- ・ humi2=湿度

※「061 気温・湿度を同時受信」とセットになります。

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |

- ・ ・ ・ センサー・LED・通信・オプション

2020/1/13作成

## 061 気温・湿度を同時受信

|   |   |   |   |
|---|---|---|---|
| ボ | セ | L | 出 |
| 通 | 入 | オ |   |

The screenshot shows the MIT App Inventor code editor for a project. The code is organized into three main sections:

- 最初だけ (Initially):** A blue block containing:
  - 無線のグループを設定 (1) (Set wireless group to 1)
  - 変数 temp3 を 0 にする (Set temp3 to 0)
  - 変数 humi3 を 0 にする (Set humi3 to 0)
- 無線で受信したとき (When receiving wirelessly):** A pink block containing:
  - 変数 temp3 を value にする (Set temp3 to value)
  - 変数 humi3 を value にする (Set humi3 to value)
  - もし name = "temp2" なら (If name is "temp2"):
    - 変数 temp3 を value にする (Set temp3 to value)
  - もし name = "humi2" なら (If name is "humi2"):
    - 変数 humi3 を value にする (Set humi3 to value)
  - もし temp3 ≠ 0 なら (If temp3 is not 0):
    - 数を表示 temp3 (Show number temp3)
    - 文字列を表示 "C/" (Show string "C/")
  - もし humi3 ≠ 0 なら (If humi3 is not 0):
    - 数を表示 humi3 (Show number humi3)
    - 文字列を表示 "%/" (Show string "%/")



※ 「060 気温・湿度を同時送信」とセットになります。

## 【説明】

### 061 温度・湿度を同時受信

<https://tfabworks.com/product/tfw-en1/>

オプションパーツ（拡張基板）である温度・湿度・気圧センサーをセットしたmicro:bit 送信側から送られている、文字列temp2（気温）とvalue（数値）と文字列humi2（湿度）とvalue（数値）のセットを受信しています。

変数temp3（気温用）とhumi3（湿度用）を追加して、セットで表示しています。

※論理→くらべる の、文字列比較で、

name=temp2 かどうか調べています。

※受信と表示のタイミングをはかるため、変数を一度0にして調整しています。

※「060 気温・湿度を同時送信」とセットになります。

|   |   |   |   |
|---|---|---|---|
| ポ | セ | L | 出 |
| 通 | 入 | オ |   |

・・・LED・通信

2020/1/13作成

# micro:bit 「サンプル プログラミング集」 V2 編

大分県立芸術文化短期大学  
情報コミュニケーション学科 非常勤講師  
望月陽一郎  
[mochizuki@oita-pjc.ac.jp](mailto:mochizuki@oita-pjc.ac.jp)

作品のクレジットを表示すること・営利目的での利用をしないこと・元の作品を改変しないこと  
(作者の許可を得ている場合は除きます)

# 目次

- 001 重ねて音を鳴らす
- 002 音がしたらビックリ
- 003 タッチしたらビックリ
- 004 音のしきい値
- 005 静かになったら
- 006 音量グラフ
- 007 音量を送信
- 008 音量を受信
- 009 タッチする楽器
- 010 メロディを打楽器に

# micro:bit プログラミングサイト

<https://makecode.microbit.org/#editor>

Python・JavaScript編集画面  
への移動

ホーム画面へ

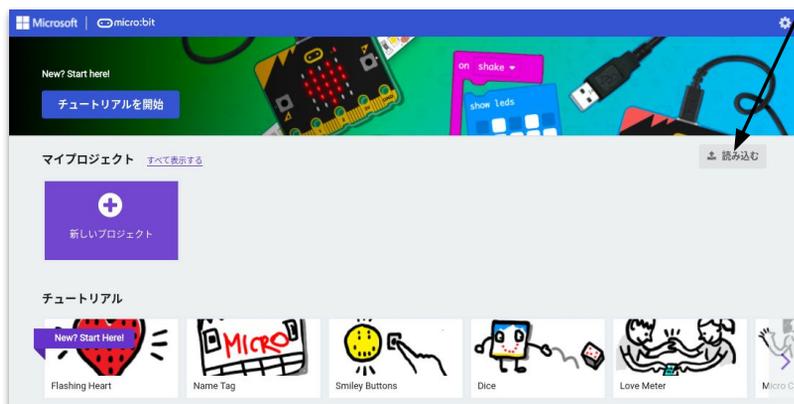


確認用シミュレータ

作成画面

作成したプログラムを  
USB接続した  
micro:bit にコピー

ローカル保存しておいた  
ファイル読込



## 【説明】

micro:bit プログラミングサイト

<https://makecode.microbit.org/#editor>

にアクセスしてプログラミングします。

- ①ブロックを組み合わせてプログラミング。
- ②確認用シミュレータで確認する。
- ③USB接続したmicro:bit にプログラムをコピー。
- ④micro:bit 本体で動作させる。

（プログラムをPCにダウンロードして保存することもできます。）

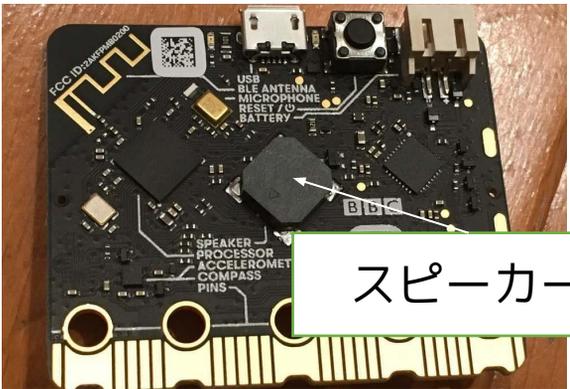
※ダウンロード保存しておいたプログラムを読み込んで修正することもできます。

# 001 重ねて音を出す

|   |   |   |    |
|---|---|---|----|
| ポ | セ | L | 出  |
| 通 | 入 | オ | v2 |



micro:bit v2  
のみで動作



## 【説明】

### 001 重ねて音を出す

micro:bit v2のみで動作するサンプルです。

・・・v1.5 ではエラーが出ます。

Aボタンには従来の音を鳴らすブロック

Bボタンには、v2用の音を鳴らすブロック

(音楽ブロックの下のほうにあります)を設定しました。

Aのメロディが流れながら、Bのメロディも流れます。

本体にスピーカーがついているのでオプションが不要になりました。

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

・・・使用するのは、ボタン・LED・スピーカー

2020/12/04作成

## 002 音がしたらビックリ

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | 才 | v2 |



micro:bit v2  
のみで動作



## 【説明】

### 002 音がしたらビックリ

micro:bit v2のみで動作するサンプルです。

・・・v1.5 ではエラーが出ます。

v2には、マイクがついているため、音センサーとして使うことができます。（入力ブロックの下のほうにあります）

大きな音をきっかけとして、LEDにビックリした顔を表示します。

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

・・・使用するのは、センサー（マイク）・LED

2020/12/04作成

## 003 タッチしたらビックリ

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

黄色のロゴが  
タッチセンサー



micro:bit v2  
のみで動作



## 【説明】

### 003 タッチしたらビックリ

micro:bit v2のみで動作するサンプルです。

・・・v1.5 ではエラーが出ます。

v2は、ロゴがタッチセンサーとなっています。

入力ブロックの下のほうにある「ロゴをタッチ」ブロックを使います。

ロゴタッチをきっかけとして、LEDにビックリした顔を表示します。

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

・・・使用するのは、センサー（タッチ）・LED

2020/12/04作成

## 004 音のしきい値

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |



micro:bit v2  
のみで動作



## 【説明】

### 004 音のしきい値

micro:bit v2のみで動作するサンプルです。

・・・v1.5 ではエラーが出ます。

v2には、マイクがついているため、音センサーとして使うことができます。

(入力ブロックの下のほうにあります)

大きな音をきっかけとして、LEDにハートマークを表示します。

最初に「大きな音のしきい値」を大きくしておくことで反応する音の大きさを変えることができます。

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

・・・使用するのは、センサー（マイク）・LED

2020/12/08作成

## 005 静かになったら

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |



micro:bit v2  
のみで動作



## 【説明】

### 005 静かになったら

micro:bit v2のみで動作するサンプルです。

・・・v1.5 ではエラーが出ます。

v2には、マイクがついているため、音センサーとして使うことができます。

(入力ブロックの下のほうにあります)

004「音のしきい値」の「小さな音」の使用例です。  
最初に「小さな音のしきい値」を設定し、まわりの音がそれより静かになったらニッコリした顔が表示されます。

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

・・・使用するのは、センサー（マイク）・LED

2020/12/11作成

## 006 音量グラフ

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

The screenshot shows the micro:bit V2 simulator interface. On the left, a virtual micro:bit board is displayed with a volume graph on its screen. The graph shows a blue bar representing volume, with a red arrow pointing to the value 29. Below the board, there are control buttons for play, stop, and volume. A purple button at the bottom says "コンソールを表示 シミュレーター".

On the right, the block-based code editor is visible. A search bar is at the top. A list of categories is shown on the left, including "基本", "入力", "音楽", "LED", "無線", "ループ", "論理", "変数", "計算", and "高度なブロック". A blue block labeled "ずっと" (Forever) is selected, containing a purple block "棒グラフを表示する 値 音量" (Show bar graph value volume), a "最大値 50" (Maximum value 50) block, and a "一時停止 (ミリ秒) 100" (Pause (milliseconds) 100) block.

micro:bit v2  
のみで動作



## 【説明】

### 006 音量グラフ

micro:bit v2のみで動作するサンプルです。

・・・v1.5 ではエラーが出ます。

v2には、マイクがついているため、音センサーとして使うことができます。

(入力ブロックの下のほうにあります)

入力された数値は「音量」として扱うことができるので、それをグラフで表示します。

|   |   |   |    |
|---|---|---|----|
| ポ | セ | L | 出  |
| 通 | 入 | オ | v2 |

・・・使用するのは、センサー（マイク）・LED

2020/12/11作成

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

## 007 音量を送信



micro:bit v2  
のみで動作



## 【説明】

### 007 音量を送信

micro:bit v2のみで動作するサンプルです。

・・・v1.5 ではエラーが出ます。

v2には、マイクがついているため、音センサーとして使うことができます。

(入力ブロックの下のほうにあります)

入力された数値は「音量」として扱うことができるので、それを無線で送信します。

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

・・・使用するのは、センサー（マイク）・通信

※008 音量を受信とセットです。

2020/12/18作成

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

## 008 音量を受信



micro:bit v2 「007 音量を送信」とセットです。

## 【説明】

### 008 音量を受信

micro:bit v2のみで動作するサンプルではありませんが、007 音量を送信するサンプルがv2のみで動作するので、このサンプルもv2対応です。

送られてきた数値（データとしては音量）をリアルタイムでグラフ化します。

|   |   |   |    |
|---|---|---|----|
| ポ | セ | L | 出  |
| 通 | 入 | オ | v2 |

・・・使用するのは、LED・通信

※007 音量を送信 とセットです。

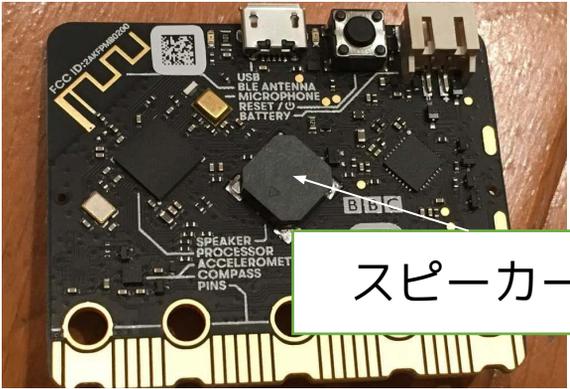
2020/12/18作成

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

## 009 タッチする楽器

黄色のロゴが  
タッチセンサー

micro:bit v2  
のみで動作



スピーカー

## 【説明】

### 009 タッチする楽器

micro:bit v2のみで動作するサンプルです。

・・・v1.5 ではエラーが出ます。

v2は、ロゴがタッチセンサーとなっています。

入力ブロックの下のほうにある「ロゴをタッチ」ブロックを使います。

ロゴタッチをきっかけとして、音を出します。

タッチを離すと音が切れるので、打楽器的に使うことができます。

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

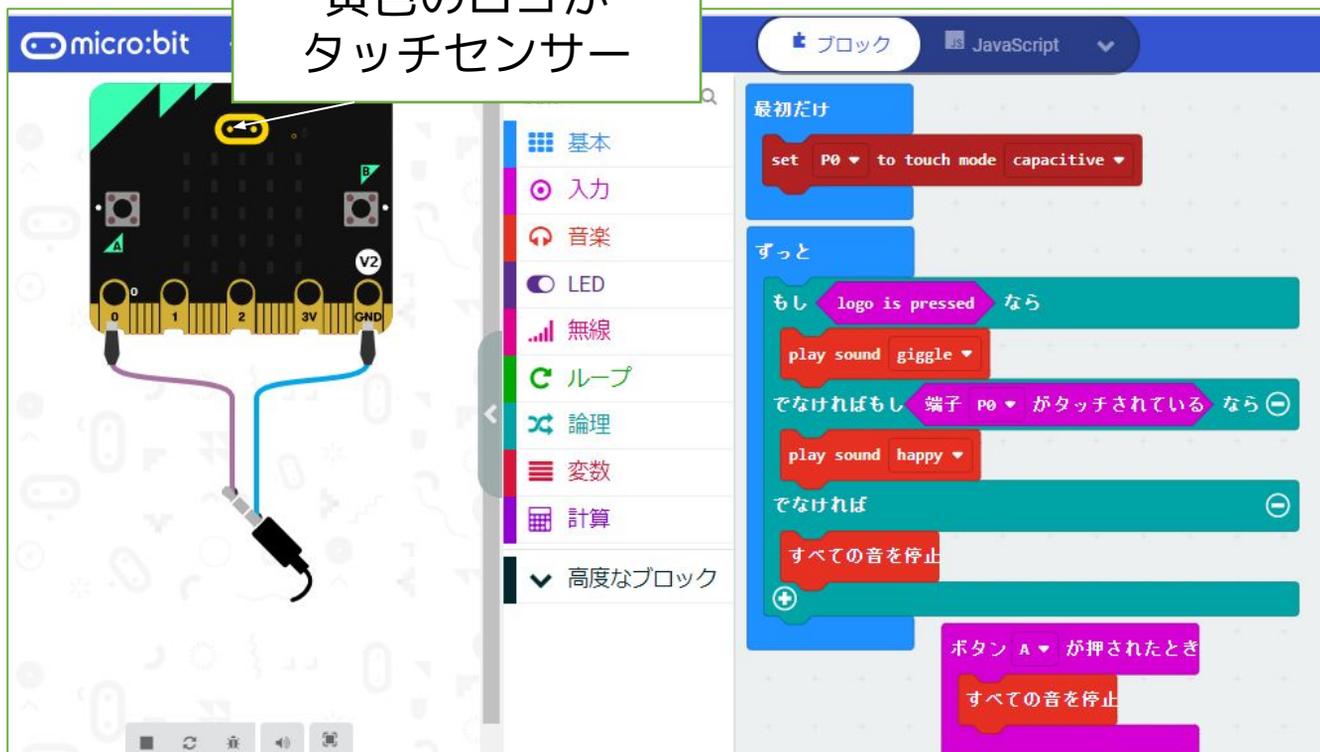
・・・使用するのは、センサー（タッチ）・スピーカー

2020/12/22作成

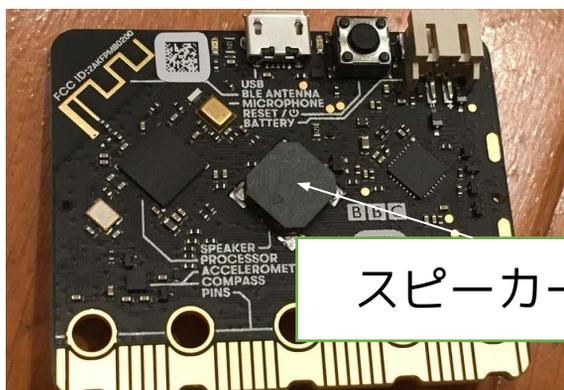
|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

## 010 メロディを打楽器に

黄色のロゴが  
タッチセンサー



micro:bit v2  
のみで動作



スピーカー

## 【説明】

### 010 メロディを打楽器に

micro:bit v2のみで動作するサンプルです。

・・・v1.5 ではエラーが出ます。

v2は、ロゴがタッチセンサーとなっています。

入力ブロックの下のほうにある「ロゴをタッチ」ブロックを使います。

また、P0端子をタッチセンサー（これまではP0とGNDを両方さわる必要があった）にするオプションが、

・高度なブロック→入出力端子→その他にある「set P0 to touch mode capacitive」にあります。

ロゴタッチとP0端子で、メロディを鳴らします。

Aボタンを押すと、音が切れるようにして打楽器的に使えるようにしました。

|   |   |   |    |
|---|---|---|----|
| ボ | セ | L | 出  |
| 通 | 入 | オ | v2 |

・・・使用するのは、センサー（タッチ）・スピーカー

2020/12/28作成



# micro:bit 「サンプル プログラミング集」 (第5.0版)

【ブロック+ V2 分冊版】

作品のクレジットを表示すること・営利目的での利用をしないこと・元の作品を改変しないこと  
(作者の許可を得ている場合は除きます)

2018/10/19 第1.0版  
2018/10/24 修正  
2018/11/02 第1.1版  
2018/11/03 修正  
2019/03/12 第1.98版  
2019/05/05 第2.0版  
2019/11/12 第2.1版  
2019/12/30 第2.11版  
2020/01/13 第3.0版  
2020/08/25 第3.5版  
2020/08/30 修正  
2021/01/04 第4.0版  
2021/03/04 修正  
2022/01/01 修正  
2022/02/01 第5.0版

