

# micro:bit 「サンプル プログラミング集」 (第3.0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

- 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 weather bitで気温表示
- 012 weather bitで湿度表示
- 013 weather bitで気圧表示
- 014 weather bitで気温湿度気圧表示
- 015 ストップウォッチ
- 016 明るさで秒数計測
- 017 コンパス (Compass)
- 018 2台で送受信
- 019 水準器
- 020 傾きメーター
- 021 方位磁針
- 022 方位磁針16方位
- 023 方位磁針16方位program分割
- 024 暗くなったことを検知
- 025 相互にストップウォッチ
- 026 通信で合奏
- 027 明るさ変化を送信
- 028 揺れたら送信
- 029 明るさ数値送信
- 030 簡易熱中症計 (weather:bit)
- 031 ひらがな表示
- 032 傾き (ピッチ) 角度を表示
- 033 明るさにより発信音の高さを変える
- 034 太陽高度計
- 035 ふりこ送信
- 036 ふりこ送信 段階1
- 037 ふりこ送信 段階2
- 038 スイッチ機能付き扇風機を回す
- 039 文字列を送信
- 040 接近したら点灯 (距離センサー)
- 041 スイッチを制御 (制御スイッチ)
- 042 リモコン・スイッチで点灯 (制御スイッチ)
- 043 人感センサー (距離センサー・制御スイッチ)
- 044 距離を表示 (距離センサー)
- 045 明るさグラフ
- 046 明るさグラフリアルタイム
- 047 距離を測定 (距離センサー)
- 048 小数点計算 (新バージョンテスト)
- 049 秒数表示 (新バージョンテスト)
- 050 micro:bitファームウェアアップデート

# 目次

- 051 人感センサー
  - 052 ランダム数値表示
  - 053 じゃんけん
  - 054 じゃんけん無線判定
  - 055 磁気表示
  - 056 磁力計
  - 057 検流(ペルチェ素子)
  - 058 検流グラフ(ペルチェ素子)
  - 059 計算機
  - 060 計算機(関数)
  - 061 気温数値を送信
  - 062 気温数値を受信
  - 063 weather:bitの数値を同時送信
  - 064 weather:bitの数値を同時受信
  - 065 センサーの気温・湿度数値を同時送信
  - 066 センサーの気温・湿度数値を同時受信
- microPython編 1.0版
- 001 「HelloWorld」と「ハートマーク」を表示する
  - 002 LED上に「O」を表示する
  - 003 傾きを表示する
  - 004 0から9まで表示する



# micro:bit プログラミングサイト

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

ホーム画面へ

JavaScript編集画面  
面への移動

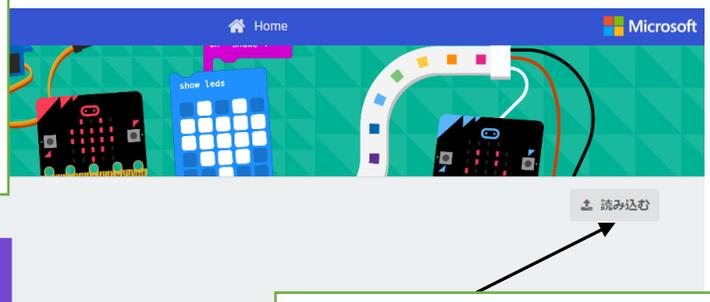


確認用シミュレータ

言語設定

作成画面

作成したプログラムのローカル  
保存  
→micro:bit にコピー



ローカル保存しておいた  
ファイル読込

micro:bit プログラミングは、上記サイトにアクセスするか、専用アプリ (Win10のみ使用可) で作成したプログラムを、USB接続したmicro:bit にデータ保存することで動作させます。事前にシミュレータで確認が可能です。

## 【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。

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

※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」してみてもよいですね。

ボ	セ	シ	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にグラフ表示するやり方を調べて作りました。センサーの数値→グラフ。

ボ	セ	シ	出
通		オ	

- ・・・使用するのは、センサー・LED

2018/4/18作成

ボ	セ	シ	出
通		オ	

## 003 ボタンで温度を表示する

The screenshot shows the Scratch simulator interface. On the left, a black Scratch character is displayed with a temperature gauge showing 21°C. A red button labeled 'A' is visible. On the right, the 'Events' block palette is open, showing a 'When button A is clicked' block with a 'Show number' block set to 'Temperature (°C)'. Below the simulator, a white box contains the text 'ボタンで動作'.

ボ	セ	シ	出
通		オ	

## 004 ボタンで明るさを表示する

The screenshot shows the Scratch simulator interface. On the left, a black Scratch character is displayed with a brightness gauge showing 128. A blue button labeled 'A' is visible. On the right, the 'Events' block palette is open, showing a 'When button A is clicked' block with a 'Show bar graph' block set to 'Brightness', a maximum value of 255, and a stop time of 1000 milliseconds. Below the simulator, a purple box contains the text '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」したものです。

「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	シ	出
通		オ	

The screenshot shows a project workspace with a black background. On the left, a temperature gauge displays "21°C". A red bar graph is visible. Two buttons, labeled "A" and "B", are connected to the board. The bottom of the workspace shows a breadboard with components labeled "0", "1", "2", "3V", and "GND".

In the center, a search bar "検索..." is above a block palette with categories: 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.

On the right, the script area contains two event blocks: "ボタン A が押されたとき" and "ボタン B が押されたとき".

- Under "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
- Under "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	シ	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		オ	



## 【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。

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

ボ	セ	シ	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	



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	



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（計算より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。

ボ	セ	L	出
通		オ	

2018/4/23作成

2019/1/3修正

・・・出力（この場合は音声として）

### 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「電流がながれたら」音を鳴らすように置き換えたものです。

別に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端子のタッチが動作するか確認したものです。

ボ	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。PI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」短音が鳴ります。

ボ	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。初期の電子楽器である「テルミン」のように手を明るさセンサーにかざすことで、音の高さを変えるようにしています。

明るさの数値を変えることで、変化の度合いを変えることができます。

ボ	セ	L	出
通		オ	

・・・センサー・出力

2018/4/23作成

# 011 weather bitで気温表示

ボ	セ	シ	出
通		オ	



パッケージの追加



Sparkfun Weather bit で検索



ブロックが追加される

## 【説明】

### 011 weather:bit で気温表示

<https://www.switch-science.com/catalog/3383/>

オプションパーツ（拡張基板）である weather:bit を micro:bit の端子部に差し込んで使います。Weather:bit は、気象観測センサー（気温・湿度・気圧）が組み込まれています。

専用のブロックを追加するには、「パッケージの追加」から検索していきます。

本体の温度センサーではなく、weather:bit のセンサーを使っている形になります。

※小数点以下の数値を四捨五入して表示させています。

・・・センサー・LED・オプション

ボ	セ	L	出
通		オ	

2018/5/6作成

2019/1/3修正

## 012 weather bitで湿度表示

ボ	セ	シ	出
通		オ	

The image shows a Scratch project interface for displaying humidity. On the left is a category menu with options like '基本', '入力', '音楽', 'LED', '無線', 'ループ', '論理', '変数', '計算', and 'Weatherbit'. The main workspace contains two event blocks: '最初だけ' (When green flag clicked) and 'ずっと' (When green flag clicked and looped). The '最初だけ' block contains a 'start weather monitoring' block. The 'ずっと' block contains a '数を表示' (Show number) block with 'humidity' selected and a value of '1024', and a '文字列を表示' (Show text) block with the format string '%/'.

専用ブロックで  
プログラム

## 013 weather bitで気圧表示

ボ	セ	シ	出
通		オ	

The image shows a Scratch project interface for displaying pressure. On the left is a category menu with options like '基本', '入力', '音楽', 'LED', '無線', 'ループ', '論理', '変数', '計算', and 'Weatherbit'. The main workspace contains two event blocks: '最初だけ' (When green flag clicked) and 'ずっと' (When green flag clicked and looped). The '最初だけ' block contains a 'start weather monitoring' block. The 'ずっと' block contains a '数を表示' (Show number) block with 'pressure' selected and a value of '25000', and a '文字列を表示' (Show text) block with the format string 'hPa/'.

## 【説明】

### 012 weather:bit で湿度表示

<https://www.switch-science.com/catalog/3383/>

オプションパーツ（拡張基板）である weather:bit を micro:bit の端子部に差し込んで使います。Weather:bit 専用のブロックでプログラムし、センサーで湿度を測ります。

・・・センサー・LED・オプション

ボ	セ	L	出
通		オ	

2018/5/6作成  
2019/1/3修正

### 013 weather:bit で気圧表示 Weather:bit 専用のブロックでプログラムし、センサーで気圧を測ります。

・・・センサー・LED・オプション

ボ	セ	L	出
通		オ	

2018/5/6作成  
2019/1/3修正

※小数点以下の数値を四捨五入して表示させています。

# 014 weather bitで気温湿度気圧表示

ボ	セ	L	出
通		オ	

Scratch code for weather bit monitoring:

- 最初だけ (When green flag clicked):
  - start weather monitoring
- ずっと (Forever loop):
  - 数を表示 (Show number): 小数点以下四捨五入 (Round to nearest integer), temperature(C), ÷, 100
  - 文字列を表示 (Show text): "C/"
  - 数を表示 (Show number): 小数点以下四捨五入 (Round to nearest integer), humidity, ÷, 1024
  - 文字列を表示 (Show text): "%/"
  - 数を表示 (Show number): 小数点以下四捨五入 (Round to nearest integer), pressure, ÷, 25000
  - 文字列を表示 (Show text): "hPa/"

# 015 ストップウォッチ

ボタンにより動作

ボ	セ	L	出
通		オ	

Scratch code for a stopwatch:

- 最初だけ (When green flag clicked):
  - 変数 秒数 を 0 にする (Set seconds to 0)
  - 変数 停止 を 1 にする (Set stop to 1)
  - 文字列を表示 "Hello!"
  - 表示を消す (Hide)
  - 文字列を表示 "Ready!"
  - 表示を消す (Hide)
  - 数を表示 秒数
- ボタン A が押されたとき (When button A is pressed):
  - 変数 秒数 を 0 にする (Set seconds to 0)
  - 変数 停止 を 1 にする (Set stop to 1)
  - 文字列を表示 "Reset!"
  - 表示を消す (Hide)
  - 数を表示 秒数
- ずっと (Forever loop):
  - もし 停止 = 1 なら (If stop = 1 then)
  - 数を表示 秒数 (Show seconds)
  - でなければ (Otherwise)
  - 一時停止 (ミリ秒) 1000 (Pause 1000 ms)
  - 変数 秒数 を 1 だけ増やす (Increment seconds by 1)
  - 数を表示 秒数 (Show seconds)

## 【説明】

### 014 weather:bit で気温湿度気圧表示

<https://www.switch-science.com/catalog/3383/>

オプションパーツ（拡張基板）である weather:bit を micro:bit の端子部に差し込んで使います。Weather:bit 専用のブロックでプログラムし、センサーで同時に気温湿度気圧を測ります。※小数点以下の数値を四捨五入して表示させています。

・・・センサー・LED・オプション

2018/5/6作成  
2020/1/3修正

ボ	セ	L	出
通		オ	

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

・・・ボタン・LED

2018/5/6作成

ボ	セ	L	出
通		オ	

# 016 明るさで秒数計測

ボ	セ	シ	出
通		オ	

明るさ変化で動作

## 【説明】

### 016 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、何か明るさをさえぎる（前を横切る）までを計測できないか考えたプログラムです。

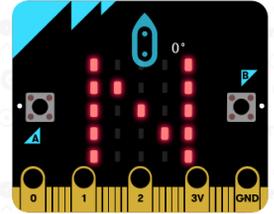
ボ	セ	シ	出
通		オ	

・・・ボタン・センサー・LED

2018/5/6作成

# 017 コンパス (Compass)

ボ	セ	シ	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

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

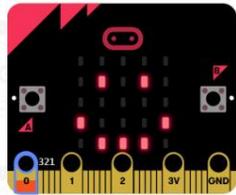
## 【説明】

### 017 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。  
(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→下記のマークが出るように)

ボ	セ	シ	出
通		才	

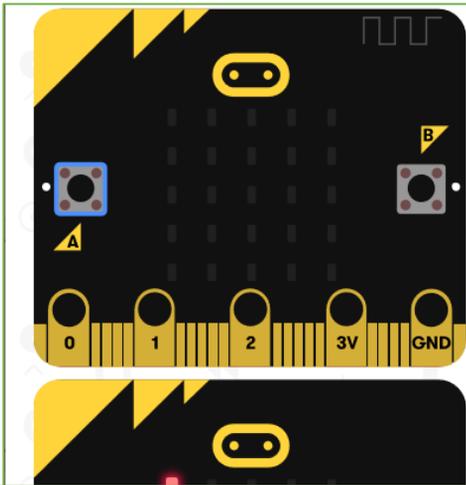


・・・センサー・LED

2018/5/14作成

# 018 2台で送受信

ボ	セ	シ	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

- 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 receivedNumber

もし receivedNumber = 0 なら

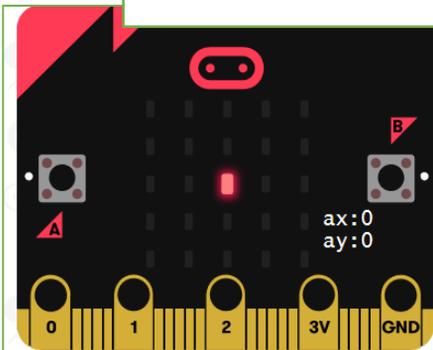
- 文字列を表示 "Receive!"
- 一時停止 (ミリ秒) 1000
- 表示を消す

Bluetoothによる通信

# 019 水準器

ボ	セ	シ	出
通		オ	

傾きセンサーで計測



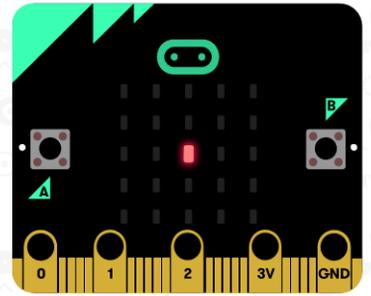
- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なブロック

最初だけ

- 変数 x を 2 にする
- 変数 y を 2 にする

ずっと

- 消灯 x 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 y



## 【説明】

### 018 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、無線を受け取ったブロックからとり出します)

ボ	セ	L	出
通		オ	

・・・ボタン・LED・通信

2018/5/17作成

### 019 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。

ボ	セ	L	出
通		オ	

・・・センサー・LED

2018/5/21作成

## 020 傾きメーター

ボ	セ	シ	出
通		才	

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算
- 高度なプロット

ずっと

変数 Y を 0 にする

変数 X を  $-1 \times \text{加速度 X} \div 400$  にする

くりかえし 5 回

点灯 x X y Y

変数 Y を 1 だけ増やす

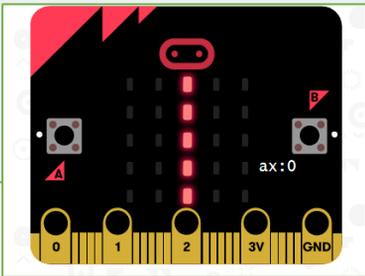
一時停止 (ミリ秒) 100

表示を消す

最初だけ

変数 X を 2 にする

傾きをLED表示



## 【説明】

### 020 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。

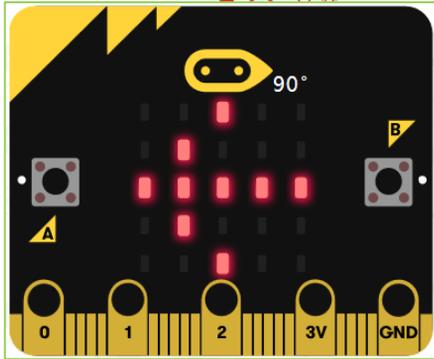
ボ	セ	シ	出
通		オ	

・・・ボタン・LED・通信

2018/5/21作成

# 021 方位磁針

ボ	セ	シ	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】

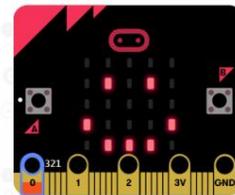
### 021 方位磁針

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。

(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 → 下記のマークが出るように)

ボ	セ	シ	出
通		オ	



・・・センサー・LED

2018/5/23作成

ボ	セ	シ	出
通		才	

## 022 方位磁針 | 6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

ずっと

もし 337 < 方向 (\*) または 方向 (\*) < 22 なら

LED画面に表示

でなければもし 22 < 方向 (\*) かつ 方向 (\*) < 67 なら

LED画面に表示

でなければもし 67 < 方向 (\*) かつ 方向 (\*) < 112 なら

LED画面に表示

でなければもし 67 < 方向 (\*) かつ 方向 (\*) < 112 なら

LED画面に表示

でなければもし 112 < 方向 (\*) かつ 方向 (\*) < 157 なら

LED画面に表示

でなければもし 157 < 方向 (\*) かつ 方向 (\*) < 202 なら

LED画面に表示

### 16方位版

## 【説明】

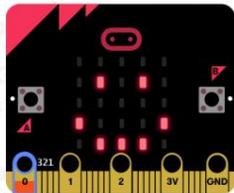
### 022 方位磁針16方位-1

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。

(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 →下記のマークが出るように)

ポ	セ	L	出
通		オ	



・・・センサー・LED

(プログラムの続きがあります 022-2)

2018/5/24作成

## 022 方位磁針16方位-2

ボ	セ	シ	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】

### 022 方位磁針16方位-2

022-1の続きになります。

2018/5/24作成

# 023 方位磁針 I 6方位program分割-I

ボ	セ	シ	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】

### 023 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。

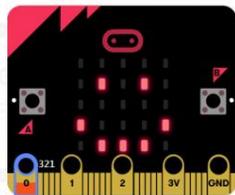
5×5のLEDで、図形を表示する方法を模索したプログラムですが、8方位版と違い、より矢印の図形を表すのが難しくなっています。

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。

(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 → 下記のマークが出るように)

ポ	セ	L	出
通		オ	



・・・センサー・LED 2018/5/24作成  
(プログラムの続きがあります 023-2)

## 023 方位磁針 I 6方位program分割-2

ボ	セ	シ	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

でなければもし 157 < 方角 (°) かつ 方角 (°) < 202 なら

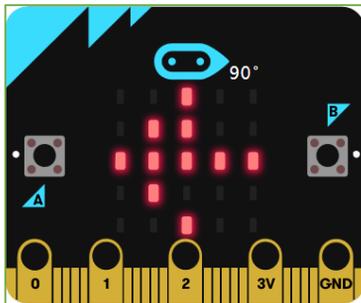
LED画面に表示

ずっと

もし 202 < 方角 (°) かつ 方角 (°) < 247 なら

LED画面に表示

でなければもし 247 < 方角 (°) かつ 方角 (°) < 292 なら



でなければもし 247 < 方角 (°) かつ 方角 (°) < 292 なら

LED画面に表示

でなければもし 292 < 方角 (°) かつ 方角 (°) < 337 なら

LED画面に表示

## **【説明】**

### **023 方位磁針16方位program分割-2**

**023-1の続きになります。**

**2018/5/24作成**

# 024 暗くなったことを検知

ボ	セ	L	出
通		オ	

The image shows a Scratch-like programming environment. On the left is a 5x5 LED matrix with a few red LEDs lit. The code editor on the right shows a 'when green flag clicked' event followed by a 'set brightness to 15' block, an 'LED matrix display' block, a 'when brightness is less than 30' block, another 'LED matrix display' block, and a 'when brightness is less than 30' block. The LED matrix is mostly dark, with a few red LEDs lit.

## 【説明】

### 024 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、変化の度合いが変わります。

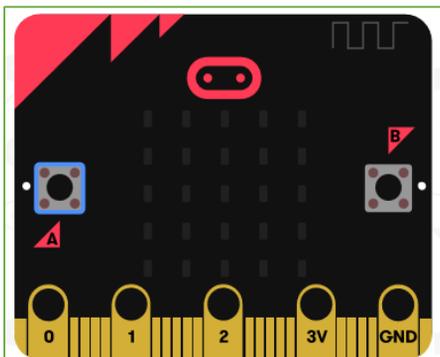
ボ	セ	シ	出
通		オ	

・・・センサー・LED

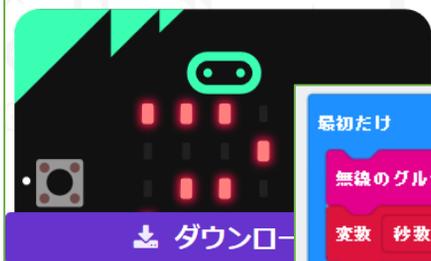
2018/5/28作成  
2019/1/6修正

## 025 相互にストップウォッチ

ボ	セ	シ	出
通		オ	



受信したらスタート



スクリーンショットのArduino IDEのブロックコード:

```
最初だけ
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
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

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】

### 025 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。

ボ	セ	シ	出
通		オ	

・・・ボタン・LED・通信

2018/6/14作成

## 026 通信で合奏

ボ	セ	L	出
通		オ	

The image shows a Scratch code editor with a search bar and a category menu on the left. The main workspace contains two scripts:

- Script 1 (Blue):**
  - 最初だけ (When green flag clicked)
  - 無線のグループを設定 (1) (Set radio group to 1)
- Script 2 (Purple):**
  - 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) (Broadcast message)
  - 一時停止 (ミリ秒) (100) (Wait 100 ms)
  - 音を鳴らす 高さ (Hz) (真ん中のド) 長さ (1) 拍 (Play sound 'Middle C' for 1 beat)
- Script 3 (Pink):**
  - 無線で受信したとき (receivedNumber) (When received number is received)
  - もし receivedNumber = 0 なら (If receivedNumber = 0)
  - 音を鳴らす 高さ (Hz) (真ん中のシ) 長さ (1) 拍 (Play sound 'Middle D' for 1 beat)

A callout box on the right contains the text: **一对多送信可能** (One-to-many transmission possible).

## 027 明るさ変化を送信

ボ	セ	L	出
通		オ	

The image shows a Scratch code editor with a search bar and a category menu on the left. The main workspace contains two scripts:

- Script 1 (Blue):**
  - 最初だけ (When green flag clicked)
  - 無線のグループを設定 (1) (Set radio group to 1)
  - LED画面に表示 (LED display)
- Script 2 (Pink):**
  - ずっと (Forever loop)
  - もし 明るさ < 20 なら (If brightness < 20)
  - 無線で数値を送信 (0) (Broadcast message)
- Script 3 (Purple):**
  - 無線で受信したとき (receivedNumber) (When received number is received)
  - もし receivedNumber = 0 なら (If receivedNumber = 0)
  - LED画面に表示 (LED display)
  - 一時停止 (ミリ秒) (100) (Wait 100 ms)
  - LED画面に表示 (LED display)

## 【説明】

### 026 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

### 027 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	

## 028 揺れたら送信

The Scratch code for '揺れたら送信' (Send when shaken) 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 (LED display grid)
- ずっと (Forever loop):**
  - もし 加速度 X > 200 なら (If acceleration X > 200)
  - 無線で数値を送信 (0) (Send value wirelessly)
- 無線で受信したとき (When received wirelessly):**
  - 変数 receivedNumber を 0 にする (Set variable receivedNumber to 0)
  - もし receivedNumber = 0 なら (If receivedNumber = 0)
  - LED画面に表示 (LED display grid)
  - 一時停止 (ミリ秒) 100 (Pause 100 ms)
  - LED画面に表示 (LED display grid)

ボ	セ	L	出
通		オ	

## 029 明るさ数値送信

The Scratch code for '明るさ数値送信' (Send brightness value) 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 (Set variable 送信数値 to 0)
  - 変数 受信数値 を 0 にする (Set variable 受信数値 to 0)
- ボタン A が押されたとき (When button A is pressed):**
  - 変数 送信数値 を 明るさ にする (Set variable 送信数値 to 明るさ)
  - 無線で数値を送信 送信数値 (Send value wirelessly 送信数値)
- 無線で受信したとき (When received wirelessly):**
  - 変数 受信数値 を receivedNumber にする (Set variable 受信数値 to receivedNumber)
  - 数を表示 受信数値 (Show number 受信数値)

Two photographs of the micro:bit are shown on the right. The top one shows the LED display with the number '128' and a yellow light indicator. The bottom one shows the LED display with several red lights lit up.

## 【説明】

### 028 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

### 029 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

# 030 簡易熱中症計 (weather:bit) - I



気温・湿度条件  
による

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

## 【説明】

### 030 簡易熱中症計 (weather:bit) - I

拡張基板であるweather:bitの気温・気圧センサーの値を使って、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

	70%	50%	30%	10%
35℃	×	△	○	-
30℃	△	△	○	-
25℃	○	○	○	-
20℃	-	-	-	-

易の考え方です)、表のように条件を考えました。一つずつ条件を設定すると16種類つくらないといけないので、気温の4種類の条件をつくり、その中に湿度の4種類の条件を入れることでプログラムをやや短くしています。

ボ	セ	シ	出
通		オ	

・・・ボタン・センサー・LED・オプション  
(プログラムの続きがあります 030-2)

2018/6/18作成  
2020/1/10修正

# 030 簡易熱中症計 (weather:bit) -2

ボ	セ	シ	出
通		才	

プログラムの続き

プログラムの続き

## 【説明】

### 030 簡易熱中症計 (weather:bit) -2

030-1の続きになります。

# 031 ひらがな表示

ボ	セ	L	出
通		才	

ひらがなに見えるよう

検索...

基本

ボタン A が押されたとき

LED画面に表示

LED画面に表示

LED画面に表示

検索...

基本

変数

計算

高度なブロック

関数



## 【説明】

### 031 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。

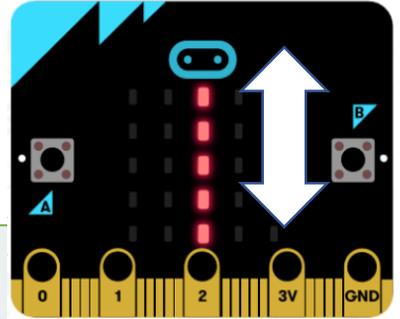
ボ	セ	シ	出
通		才	

・・・LED

2018/7/29作成

## 032 傾き (ピッチ) 角度を表示

ボ	セ	シ	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

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

## 【説明】

### 032 傾き (ピッチ) 角度を表示

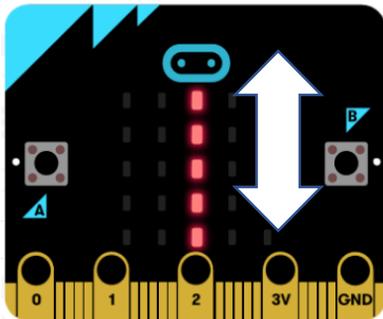
ボタンを押したときのmicro:bit 基板の角度 (ピッチのほう。もう一つの角度がロール) を使って表示するプログラムです。

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。

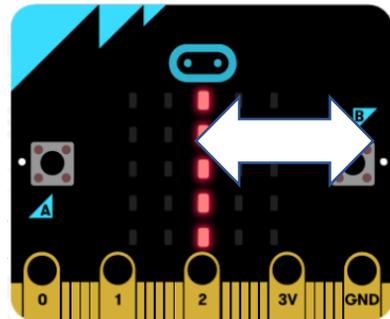
ボ	セ	シ	出
通		オ	

2018/7/29作成

・・・ボタン・センサー・LED

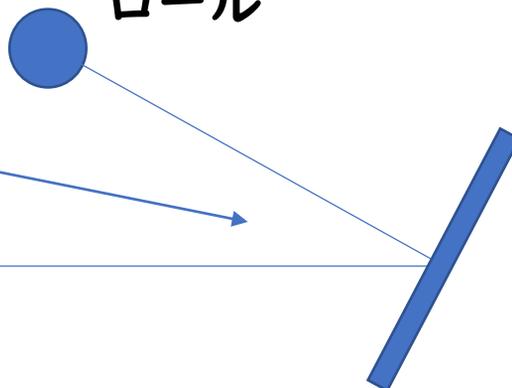


ピッチ



ロール

この角度



# 033 明るさにより発信音の高さを変える

ボ	セ	L	出
通		オ	

The image shows a Scratch project interface. On the left, a micro:bit is connected to a MIDI controller. The Scratch script is as follows:

- When button A is pressed:
  - When brightness is 50 or less, play sound '真ん中のド'.
  - When brightness is 75 or less, play sound '真ん中のレ'.
  - When brightness is 100 or less, play sound '真ん中のミ'.
  - When brightness is 125 or less, play sound '真ん中のファ'.
  - When brightness is 150 or less, play sound '真ん中のソ'.
- When button A+B is pressed:
  - Play sound '真ん中のラ'.

A callout box on the right contains the text: **明るさで変化**

## 【説明】

### 033 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	シ	出
通		オ	

・・・ボタン・センサー・出力

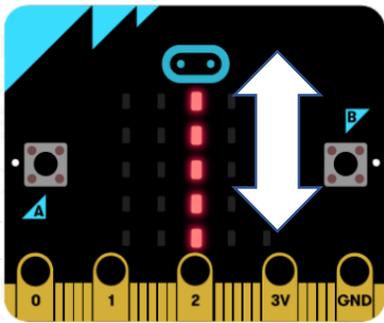
2018/7/29作成

# 034 太陽高度計

ボ	セ	シ	出
通		才	

The screenshot shows the Scratch environment with a hardware setup on the left and code blocks on the right. The hardware setup includes a microcontroller board with a potentiometer and a speaker. The code blocks are as follows:

- 最初だけ** (When first clicked):
  - 変数 高度 を 0 にする (Set variable '高度' to 0)
- ボタン A+B が押されたとき** (When button A+B is pressed):
  - 休符 (ミリ秒) 1 拍 (Pause 1 beat)
- ボタン B が押されたとき** (When button B is pressed):
  - 変数 高度 を 90 - 傾斜 (°) ピッチ にする (Set variable '高度' to 90 minus slope)
  - くりかえし 3 回 (Repeat 3 times):
    - 数を表示 高度 (Show number '高度')
    - 文字列を表示 /\* (Show text '/\*')
- ボタン A が押されたとき** (When button A is pressed):
  - もし 明るさ < 50 なら (If brightness < 50):
    - 音を鳴らす 高さ (Hz) 真ん中のド (Play sound '真ん中のド')
  - でなければもし 明るさ < 75 なら (If not, if brightness < 75):
    - 音を鳴らす 高さ (Hz) 真ん中のレ (Play sound '真ん中のレ')



太陽に向けて使用

The screenshot shows the Scratch environment with a loop of code blocks for playing notes based on brightness. The code blocks are as follows:

- でなければもし 明るさ < 100 なら (If not, if brightness < 100):
  - 音を鳴らす 高さ (Hz) 真ん中のミ (Play sound '真ん中のミ')
- でなければもし 明るさ < 125 なら (If not, if brightness < 125):
  - 音を鳴らす 高さ (Hz) 真ん中のファ (Play sound '真ん中のファ')
- でなければもし 明るさ < 150 なら (If not, if brightness < 150):
  - 音を鳴らす 高さ (Hz) 真ん中のソ (Play sound '真ん中のソ')
- でなければもし 明るさ < 200 なら (If not, if brightness < 200):
  - 音を鳴らす 高さ (Hz) 真ん中のラ (Play sound '真ん中のラ')
- でなければ (If not):
  - 音を鳴らす 高さ (Hz) 真ん中のシ (Play sound '真ん中のシ')

## 【説明】

### 034 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。

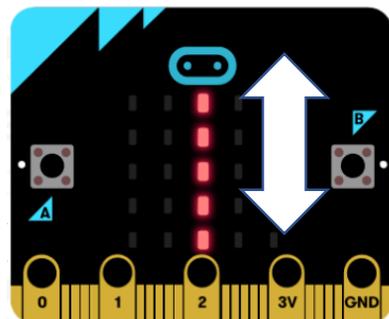
リアルタイムで明るさセンサーの値から音の高さを変えて出力するプログラムと、ボタンを押したときの micro:bit 基板の角度（ピッチのほう。もう一つの角度がロール）から太陽高度を表示するプログラムを組み合わせています。

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



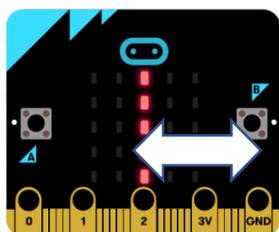
2018/8/6作成

ボ	セ	シ	出
通		オ	

## 035 ふりこ送信



最下点に来た時をカウント



## 【説明】

### 035 ふりこ送信

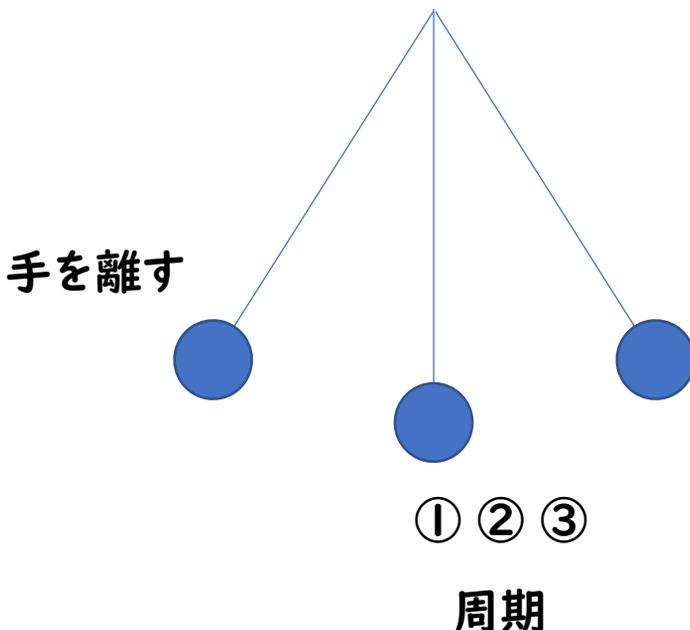
ふりこの周期を測るためのプログラムの一部です。「ふりこの最下点に来た時、ロールが0(水平)になる」ことを使っています。

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

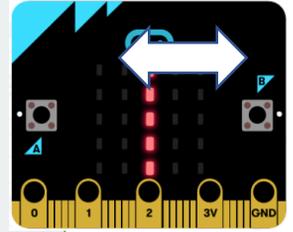
2018/8/15作成



ボ	セ	L	出
通		オ	

# 036 ふりこ受信 段階1

Scratch code editor showing the first stage of a pendulum reception program. The code includes initialization blocks for variables (time, start, count) and a loop that checks for received data and updates the count.



Scratch code editor showing the second stage of the pendulum reception program. The code adds a button press event that resets the variables and starts a new cycle.

## 【説明】

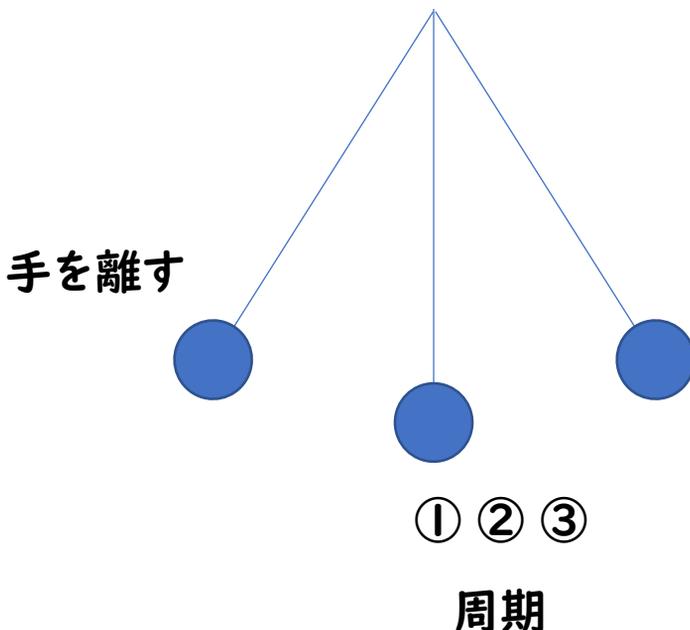
### 036 ふりこ受信 段階1

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。

ボ	セ	シ	出
通		オ	

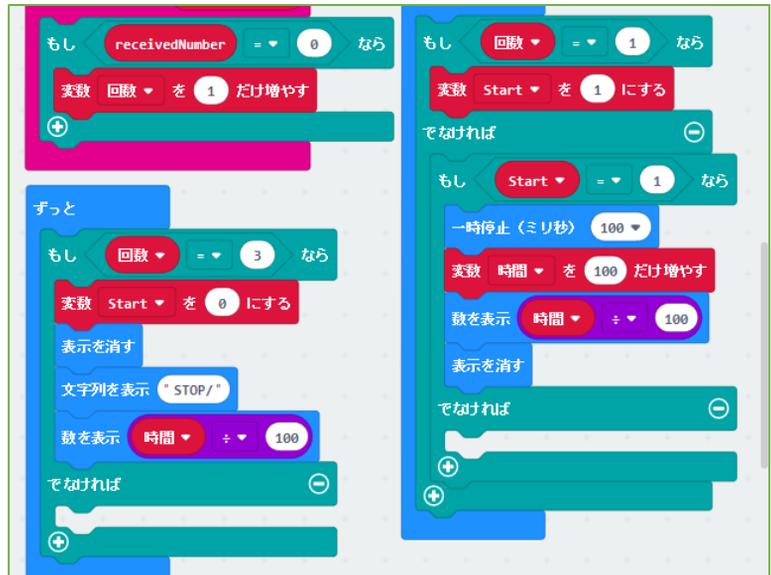
・・・ボタン・センサー・LED  
(段階2があります)

2018/8/15作成



# 037 ふりこ受信 段階2

ボ	セ	シ	出
通		オ	



## 【説明】

### 037 ふりこ受信 段階2

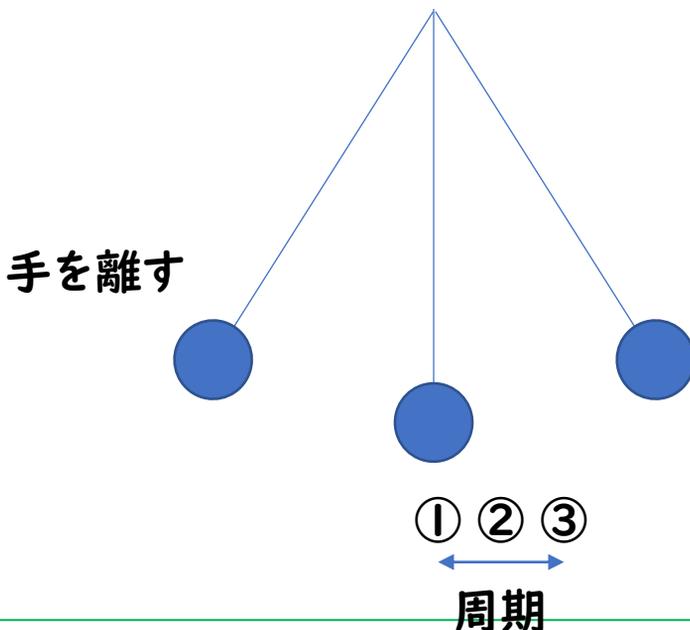
ふりこの周期を測るためのプログラムの一部です。  
035「ふりこ送信」と組み合わせます。

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	シ	出
通		オ	

・・・ボタン・センサー・LED・通信



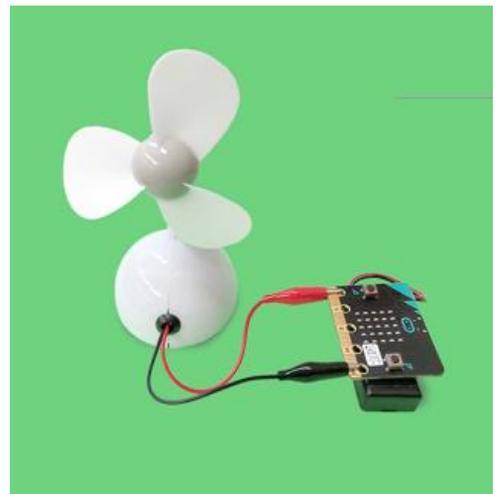
2018/8/15作成

## 038 スイッチ機能付き扇風機を回す

ポ	セ	シ	出
通		オ	



PWM制御の扇風機  
(TFabWorks)



## 【説明】

### 038 スイッチ機能付き扇風機を回す

<https://tfabworks.com/product/mb-st-set-b3/>

(キットの中の扇風機を使用)

扇風機自体に信号を受け取ってオンオフする機能があるので、それを動かすために作ってみたプログラムです。

P0端子とGND端子にそれぞれつないで、そこに「1」と「0」の信号を出力しています。

Aボタンで「オン」Bボタンで「オフ」を送信します。

ボ	セ	L	出
通		オ	

・・・ボタン・出力・オプション

2018/8/27作成

## 039 文字列を送信

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

無線のグループを設定 1

ボタン A が押されたとき

無線で文字列を送信 "Konichiwa!"

無線で受信したとき receivedString

文字列を表示 receivedString

## 【説明】

### 039 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。

これは送受信両方入っていますが、分割して2台に入れても動作します。

ボ	セ	シ	出
通		オ	

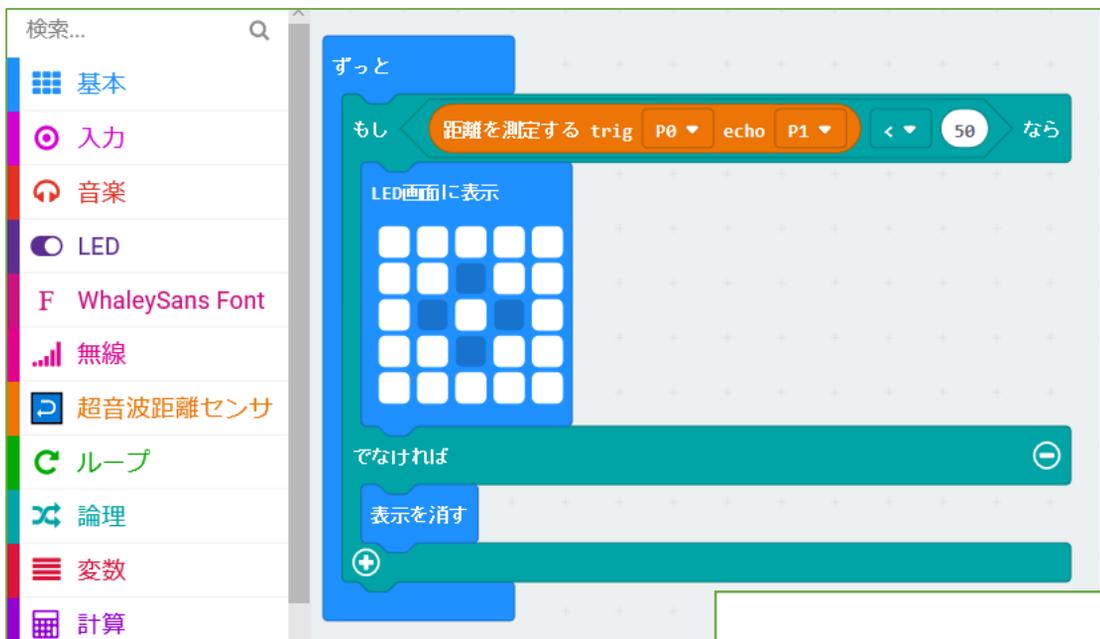
・・・ボタン・LED・通信

2018/8/27作成

## 040 接近したら点灯 (距離センサー)

ボ	セ	L	出
通		オ	

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



近づいたらLED表示

ワンタッチ距離センサー (TFabWorks)



## 【説明】

### 040 接近したら点灯 (距離センサー)

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

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。

ボ	セ	シ	出
通		オ	

・・・センサー・LED・オプション

2018/9/24作成

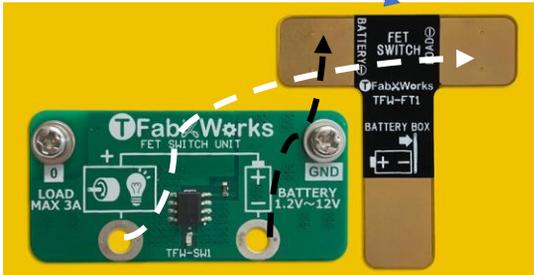
## 04 | スイッチを制御 (制御スイッチ)

ボ	セ	L	出
通		オ	



乾電池の接触部分に

micro:bit用プログラム  
制御スイッチセット  
(TFabWorks)



## 【説明】

### 04 | スイッチを制御 (制御スイッチ)

<https://tfabworks.com/product/mb-set-sw1/>

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、「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

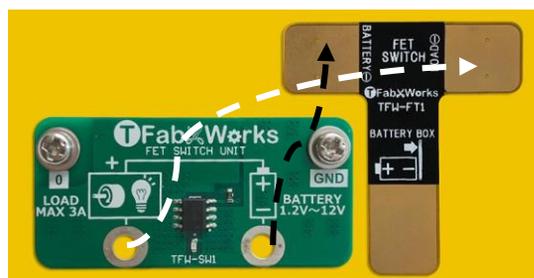
2018/9/27作成

## 042 リモコン・スイッチで点灯 (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  
(TFabWorks)



## 【説明】

### 042 リモコン・スイッチで点灯(制御スイッチ)

<https://tfabworks.com/product/mb-set-sw1/>

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。

041「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	

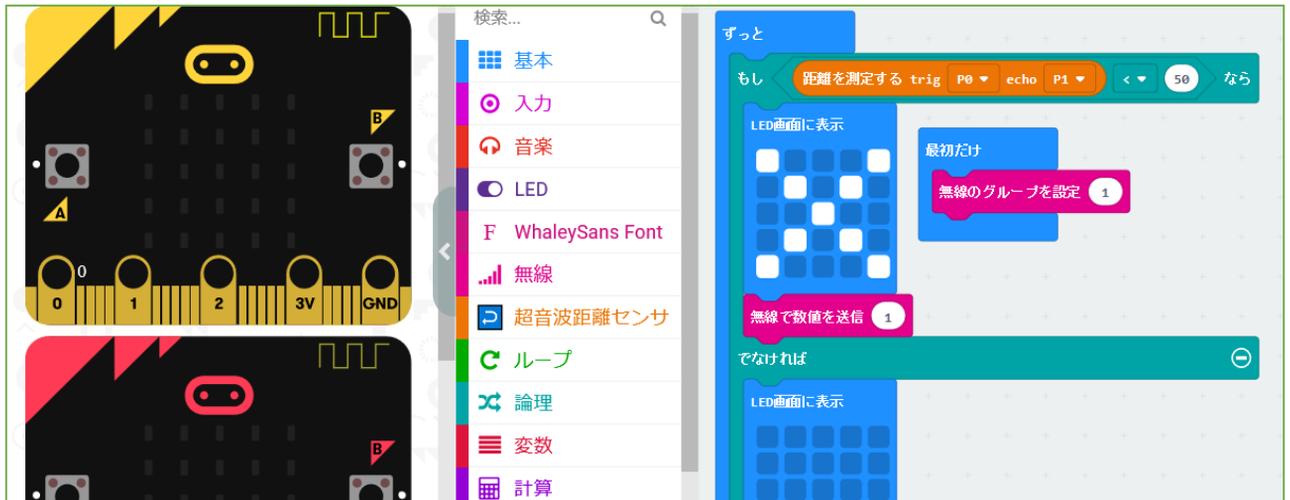
・・・ボタン・出力・通信・オプション

2018/9/27作成

## 043 人感センサー (距離センサー・制御スイッチ) - I

ボ	セ	シ	出
通		才	

### 送信部



### ワンタッチ距離センサー (TFabWorks)

## 【説明】

### 043 人感センサー（距離センサー・制御スイッチ） -1

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

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)

<https://tfabworks.com/product/mb-set-sw1/>

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。

042「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	シ	出
通		オ	

2018/10/1作成

- ・・・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)

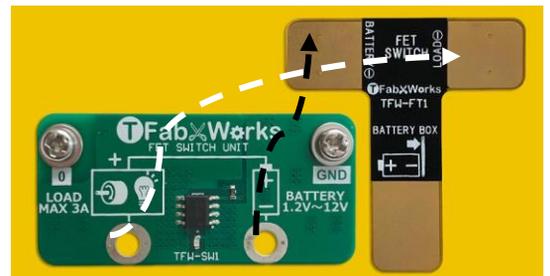
## 043 人感センサー（距離センサー・制御スイッチ）-2

ボ	セ	シ	出
通		オ	

受信部



micro:bit用プログラム  
制御スイッチセット  
(TFabWorks)



## 【説明】

### 043 人感センサー（距離センサー・制御スイッチ） -2

043-1の続きになります。

2018/10/1作成

## 044 距離を表示 (距離センサー)

ボ	セ	シ	出
通		オ	



ワンタッチ距離セン  
サー (TFabWorks)



## 【説明】

### 044 距離を表示 (距離センサー)

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

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

## 045 明るさグラフ

ボ	セ	シ	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】

### 045 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	シ	出
通		オ	

・・・センサー・LED

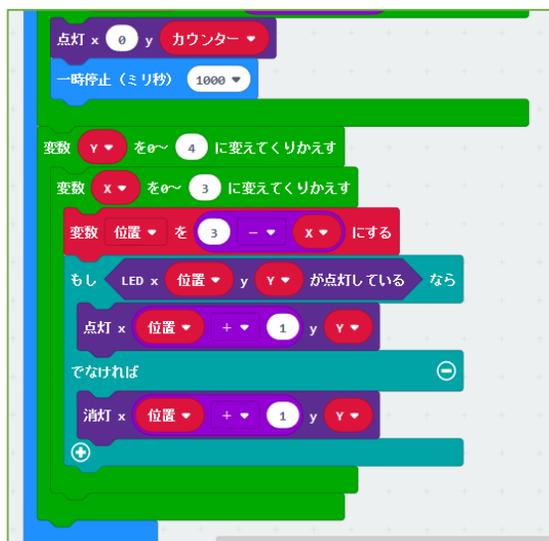
2018/10/3作成  
2020/1/10修正

## 046 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】

### 046 明るさグラフリアルタイム

045「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	シ	出
通		オ	

・・・センサー・LED

2018/10/3作成  
2020/1/10修正

## 047 距離を測定 (距離センサー)

ボ	セ	シ	出
通		才	



ワンタッチ距離セン  
サー (TFabWorks)



## 【説明】

### 047 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しています。

(<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

## 048 小数点計算 (新バージョンテスト)

ボ	セ	シ	出
通		オ	へ



最初だけ 0.330000000000000002  
Aボタン 0.330000000000000002  
Bボタン 0.3

## 【説明】

### 048 小数点表示(新バージョンテスト)

作成時点でのベータサイトを使って作ったサンプルです。2018/10/26以降ベータ版が正式版になりました。

micro:bit では小数点が扱えないので、表示できるか試してみました。小数点以下2位まで表示されたので、 $1/3$ を表示させてみたところ、「0.330000000000000002」という値が出たので、パターンを変えてどうなるか確認しました。

$$1/3 = 0.330000000000000002$$

$((1*100)/3)$ を四捨五入

$$/100 = 0.330000000000000002$$

$((1*10)/3)$ を四捨五入

$$/10 = 0.3$$

式により表示される値が違うのはバグなのかもしれません。

ボ	セ	L	出
通		オ	ベ

・・・ボタン・LED・ベータ

2018/10/9作成  
2020/1/11修正

## 049 秒数表示 (新バージョンテスト)

ボ	セ	L	出
通		オ	へ

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 秒数 を 0 にする

数を表示 秒数

ボタン A+B が押されたとき

表示を消す

変数 秒数 を 0 にする

数を表示 秒数

ボタン A が押されたとき

変数 秒数 を 0 にする

変数 動作 を 1 にする

表示を消す

もし 動作 = 1 ならくりかえし

一時停止 (ミリ秒) 100

変数 秒数 を 0.1 だけ増やす

変数 動作 を 1 にする

表示を消す

もし 動作 = 1 ならくりかえし

一時停止 (ミリ秒) 100

変数 秒数 を 0.1 だけ増やす

ボタン B が押されたとき

変数 動作 を 0 にする

表示を消す

数を表示 小数点以下四捨五入 秒数 × 100 ÷ 100

## 【説明】

### 049 秒数表示 (新バージョンテスト)

作成時点でのベータサイトを使って作ったサンプルです。2018/10/26以降ベータ版が正式版になりました。

Aボタンを押してスタート、Bボタンを押すまでの時間を、小数表示するようにしてみました。

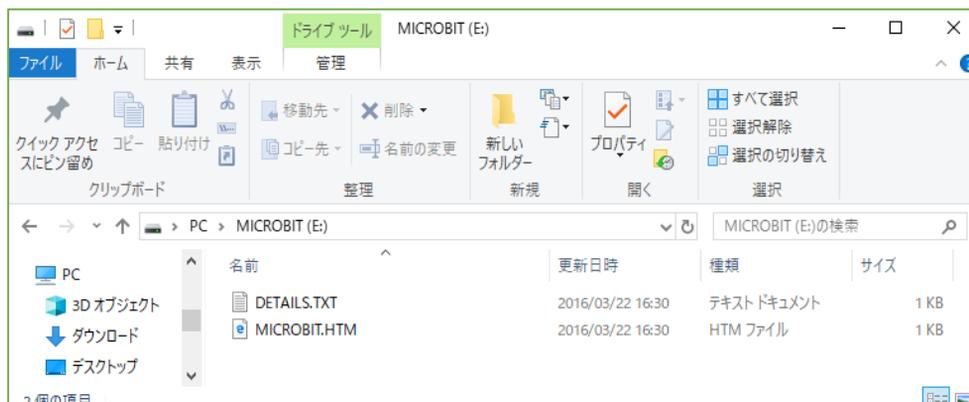
ボ	セ	シ	出
通		オ	ベ

・・・ボタン・LED・ベータ

2018/10/8作成  
2020/1/11修正

# 050 micro:bit ファームウェアアップデート - I

## USB接続して、DETAILS.TXTを確認



## この場合 version 0241

```
DETAILS.TXT - メモ帳
ファイル(E) 編集(E) 書式(O) 表示(V) ヘルプ(H)
# 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】

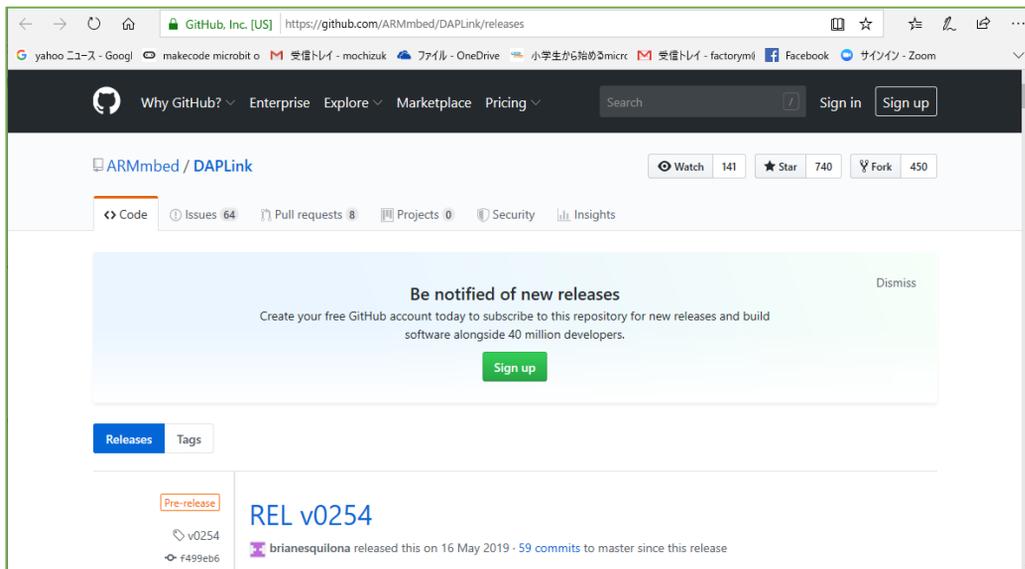
### 050 micro:bit ファームウェアアップデート

作成当時のベータサイトで作ったサンプル048「小数点計算（makecodeベータ）」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

# 050 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



- New convention for adding a family id with vendor id.
- Firmware version and unique ID can be access via DAP\_Info command.

## Notes

- Win7 Mbed Composite driver crash occurs in some boards just because of the added interface in USB. Current solution is to unistall this driver and just keep the Mbed Serial driver on Win 7. These drivers should not be installed on Windows 8 to 10.

## Assets 3

0254_release_package_f499eb6e.zip	10.3 MB
Source code (zip)	
Source code (tar.gz)	

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

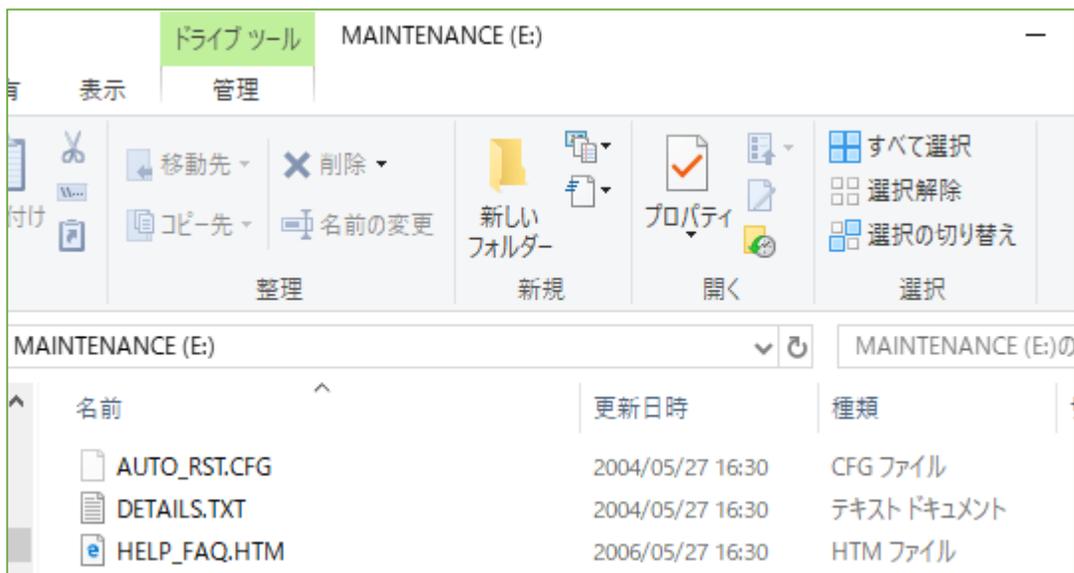
**【メモ】**

## 050 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

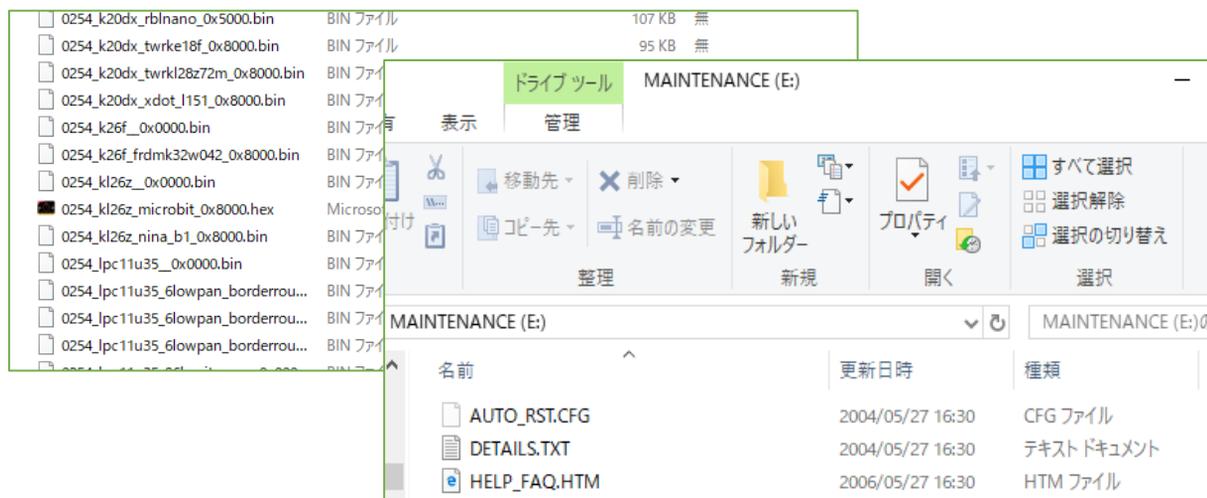
### リセットボタンを押しながら基板をUSB接続すると メンテナンスモードが開く



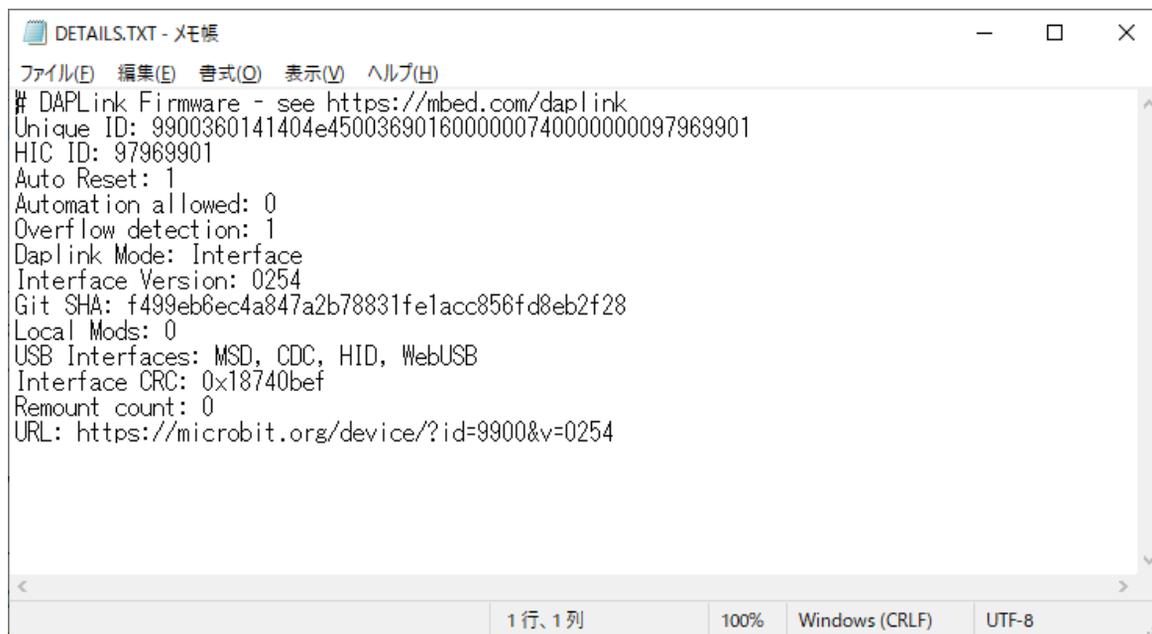
**【メモ】**

# 050 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】**

ボ	セ	シ	出
通		オ	

## 051 人感センサー



## 【説明】

### 051 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	シ	出
通		オ	

2019/1/16作成

・・・ボタン・センサー・LED・オプション

## 052 ランダム数値表示

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

ボタン A が押されたとき

数を表示 0 から 9 までの乱数

## 【説明】

### 052 ランダム数値表示

ここまで「乱数」を使ってこなかったのが、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	

2019/1/17作成

・・・ボタン・LED

# 053 じゃんけん

ボ	セ	し	出
通		才	

The image shows a Scratch-like block editor interface for programming a microcontroller board (likely an Arduino Uno) to play Rock-Paper-Scissors. The board is shown on the left with its pins labeled 0, 1, 2, 3V, and GND. The editor has a search bar and a category menu on the left with options: 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). The main workspace contains the following code blocks:

- ボタン A が押されたとき** (When button A is pressed):
  - 変数 乱数 を 0 から 2 までの乱数 にする** (Set random number variable to a random number between 0 and 2).
  - 数を表示 乱数** (Show number: random number).
  - もし 乱数 = 0 なら** (If random number = 0):
    - LED画面に表示** (Show on LED screen).
  - でなければもし 乱数 = 1 なら** (If not, if random number = 1):
    - LED画面に表示** (Show on LED screen).

The LED screen displays a 4x4 grid of lights, with some lights illuminated in red and others in blue, representing the current state of the game.

## 【説明】

### 053 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	シ	出
通		オ	

・・・ボタン・LED

2019/1/17作成  
2020/1/11修正



パー



チョキ



グー

# 054 じゃんけん無線判定

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 0 から 2 までの乱数 にする
- 数を表示 乱数
- 無線で数値を送信 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
- でなければ

最初だけ

- 無線のグループを設定 1

無線で受信したとき receivedNumber

- もし 乱数 = 0 なら
- もし receivedNumber = 0 なら
- アイコンを表示
- でなければもし receivedNumber = 1 なら
- アイコンを表示
- でなければもし receivedNumber = 2 なら
- アイコンを表示
- でなければもし 乱数 = 1 なら
- もし receivedNumber = 0 なら
- アイコンを表示
- でなければもし receivedNumber = 1 なら

もし receivedNumber = 0 なら

- アイコンを表示

でなければもし receivedNumber = 1 なら

- アイコンを表示

でなければもし receivedNumber = 2 なら

- アイコンを表示

でなければ

もし receivedNumber = 0 なら

- アイコンを表示

でなければもし receivedNumber = 1 なら

- アイコンを表示

でなければもし receivedNumber = 2 なら

- アイコンを表示

## 【説明】

### 054 じゃんけん無線判定

「053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。

・・・ボタン・LED

2019/1/18作成  
2020/1/12修正

ボ	セ	シ	出
通		オ	



パー



チョキ



グー

## 055 磁気表示

ボ	セ	シ	出
通		才	

The screenshot displays the Scratch IDE interface. On the left, the stage shows a breadboard with a sensor and several red LEDs. The script area on the right contains a 'ずっと' (Forever) loop block. Inside the loop, there is a '数を表示' (Display number) block with '磁力 (μT)' (Magnetic force) selected and a dropdown arrow. Below it is a '一時停止 (ミリ秒)' (Pause in milliseconds) block with the value '100' and a dropdown arrow. A sidebar on the left lists various Scratch categories: 基本 (Basic), 入力 (Input), 音楽 (Music), LED, 無線 (Wireless), ループ (Loops), 論理 (Logic), 変数 (Variables), 計算 (Calculations), and 高度なブロック (Advanced blocks).

## 【説明】

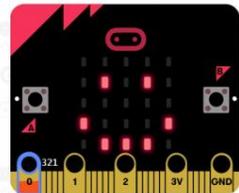
### 055 磁気表示

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。

(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 → 下記のマークが出るように)

ボ	セ	シ	出
通		オ	



2019/1/23作成

・・・ボタン・センサー・LED

## 056 磁力計

ボ	セ	シ	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

ずっと

棒グラフを表示する 値 磁力 ( $\mu\text{T}$ ) x

最大値 1000

一時停止 (ミリ秒) 100

🔊 コンソールを表示 シミュレーター

## 【説明】

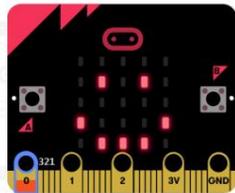
### 056 磁力計

「055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、磁石を見つかったり、磁力の強さを測ることができるとかんがえました。

(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 → 下記のマークが出るように)

ボ	セ	L	出
通		オ	

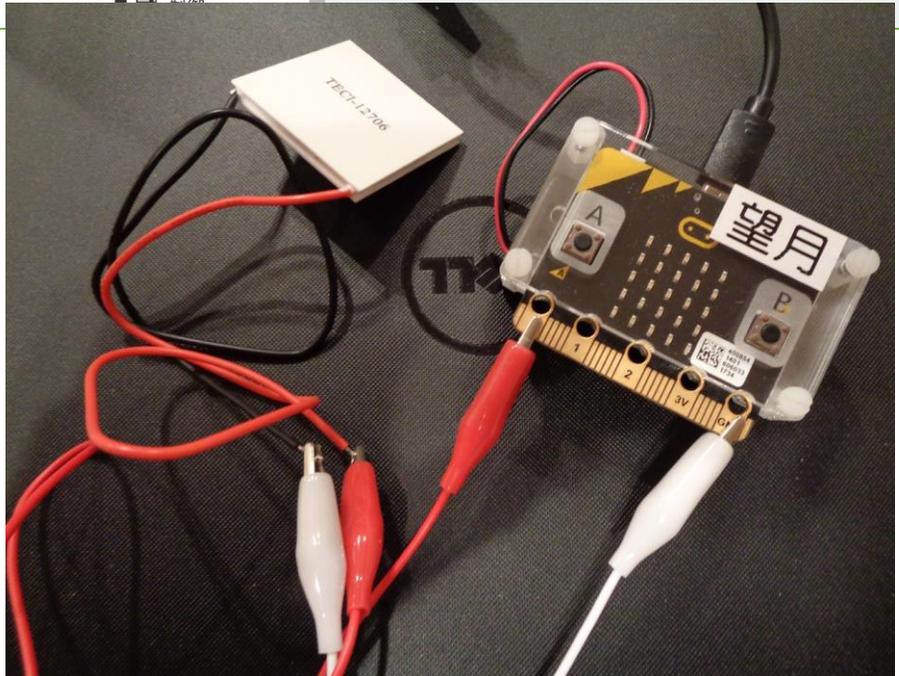


2019/2/19作成

・・・ボタン・センサー・LED

# 057 検流 (ペルチェ素子)

ボ	セ	シ	出
通	入	才	



## 【説明】

### 057 検流 (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を作ってみました。

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。

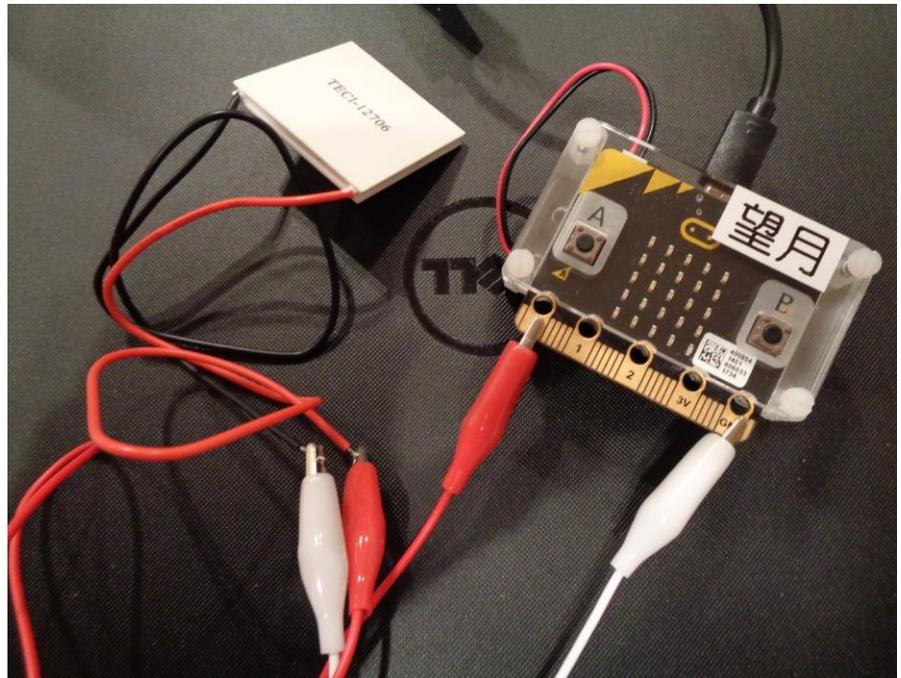
ポ	セ	L	出
通	入	才	

2019/3/2作成

・・・LED・入力

# 058 検流グラフ(ペルチェ素子)

ボ	セ	L	出
通	入	才	



## 【説明】

### 058 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057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

# 059 計算機-1

ボタン A が押されたとき

もし 計算段階 = 0 なら

数を表示 ~される数

変数 ~される数 を 1 だけ増やす

もし 計算段階 = 1 なら

もし 四則 = 0 なら

LED画面に表示

もし 四則 = 1 なら

最初だけ

変数 ~する数 を 0 にする

変数 ~される数 を 0 にする

変数 計算段階 を 0 にする

変数 四則 を 0 にする

ボタン B が押されたとき

変数 計算段階 を 1 だけ増やす

ボタン A+B が押されたとき

変数 ~する数 を 0 にする

変数 ~される数 を 0 にする

変数 計算段階 を 0 にする

変数 四則 を 0 にする

表示を消す

変数 四則 を 0 にする

表示を消す

もし 四則 = 1 なら

LED画面に表示

もし 四則 = 2 なら

LED画面に表示

もし 四則 = 3 なら

LED画面に表示

もし 四則 = 3 なら

LED画面に表示

変数 四則 を 1 だけ増やす

もし 計算段階 = 2 なら

数を表示 ~する数

変数 ~する数 を 1 だけ増やす

もし 計算段階 = 3 なら

LED画面に表示

※計算機-2に続きます。

## 【説明】

### 059 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。

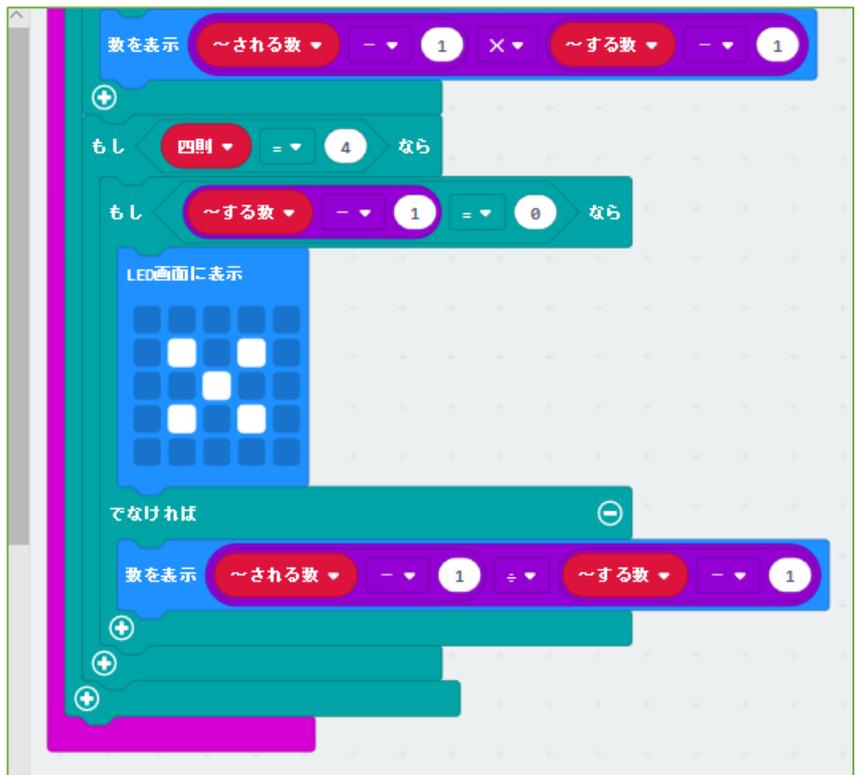
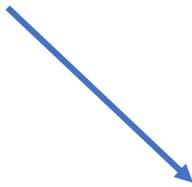
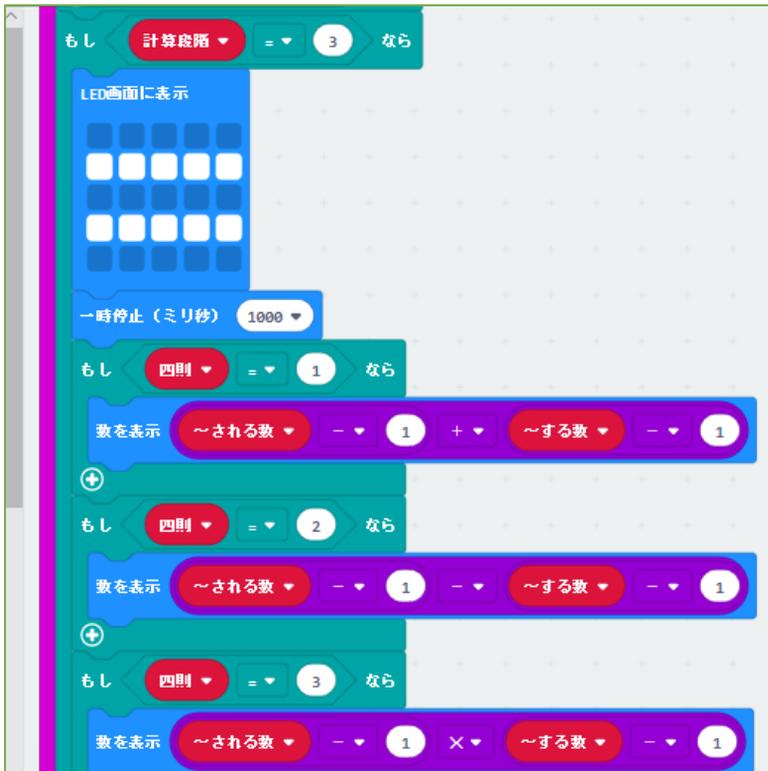
ボ	セ	L	出
通	入	オ	

2019/3/4作成

・・・ボタン・LED

※059 計算機-2 に続きます。

# 059 計算機-2



## 【説明】

### 059 計算機-2

※059 計算機-1 の続きです。

- ・Aボタンを押して、四則（+-×÷）を変化させます。
- ・Bボタンを押して、四則を決定。
  
- ・～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、～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	才	

・・・ボタン・LED

2019/3/4作成

# 060 計算機(関数)

※059 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 Scratch block palette. Below the dialog, there are four function blocks: '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' (Show Number) block with two input fields for '~される数' (Number to be operated on) and '~する数' (Number to operate with), and a dropdown menu for the operation (+, -, ×, ÷). To the right, a vertical sequence of blocks shows a series of 'もし' (If) blocks with conditions like '四則 = 1 なら' (If operation is 1), followed by '呼び出し' (Call) blocks for the corresponding function, and an 'LED画面に表示' (Show on LED screen) block.

## 【説明】

### 060 計算機(関数)

「059 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。

059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。

ボ	セ	L	出
通	入	才	

・・・ボタン・LED

2019/3/4作成

## 06 | 気温数値を送信

ボ	セ	シ	出
通	入	オ	



The image shows a Scratch-like block-based programming environment. On the left, a microcontroller board (likely a Raspberry Pi Pico) is connected to a temperature sensor. The board's display shows a temperature of 26°C. The programming interface includes a search bar at the top, a category menu on the left with options like '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). The script area on the right contains the following blocks:

- 最初だけ (Initially)
- 無線のグループを設定 (1) (Set wireless group)
- ずっと (Forever loop)
- 数を表示 (温度 (°C)) (Show number)
- 無線で数値を送信 (温度 (°C)) (Send numerical value wirelessly)
- もし (温度 (°C)) > 30 なら (If temperature is greater than 30)
- 音を鳴らす (高さ (Hz) 上のド 長さ 1 拍) (Play sound)

※「気温数値を受信」とセットになります。

## 【説明】

### 061 気温数値を送信

micro:bitで熱中症予防ができないか試したものです。

温度センサーの値を随時受信側のmicro:bitに送信するとともに、一定の温度以上になったら警告音を出すようにしています。

小さなスピーカーをつけるとウェアラブルな機器になりますね。

※「062気温数値を受信」とセットになります。

ボ	セ	L	出
通	入	オ	

・・・センサー・LED・出力・通信

2019/11/7作成

## 062 気温数値を受信

ボ	セ	シ	出
通	入	才	

The image shows a Scratch-like programming environment. On the left, a microcontroller board (likely an Arduino Uno) is connected to a USB cable. The board's pins are labeled 0, 1, 2, 3V, and GND. The code editor on the right contains the following blocks:

- 最初だけ (Initially)
- 無線のグループを設定 (1) (Set wireless group)
- 無線で受信したとき (receivedNumber) (When received wirelessly)
- 数を表示 (receivedNumber) (Show number)
- もし (receivedNumber) > 30 なら (If receivedNumber > 30 then)
- 音を鳴らす 高さ (Hz) 上のド 長さ 1 拍 (Play sound: pitch: upper D, duration: 1 beat)

※「気温数値を送信」とセットになります。

## 【説明】

### 062 気温数値を受信

micro:bitで熱中症予防ができないか試したものです。

温度センサーの値が送信側のmicro:bitから送信されるので、離れたところでそれを受信してLEDに表示。

一定の温度以上になったら警告音を出すようにしています。

※「06 | 気温数値を送信」とセットになります。

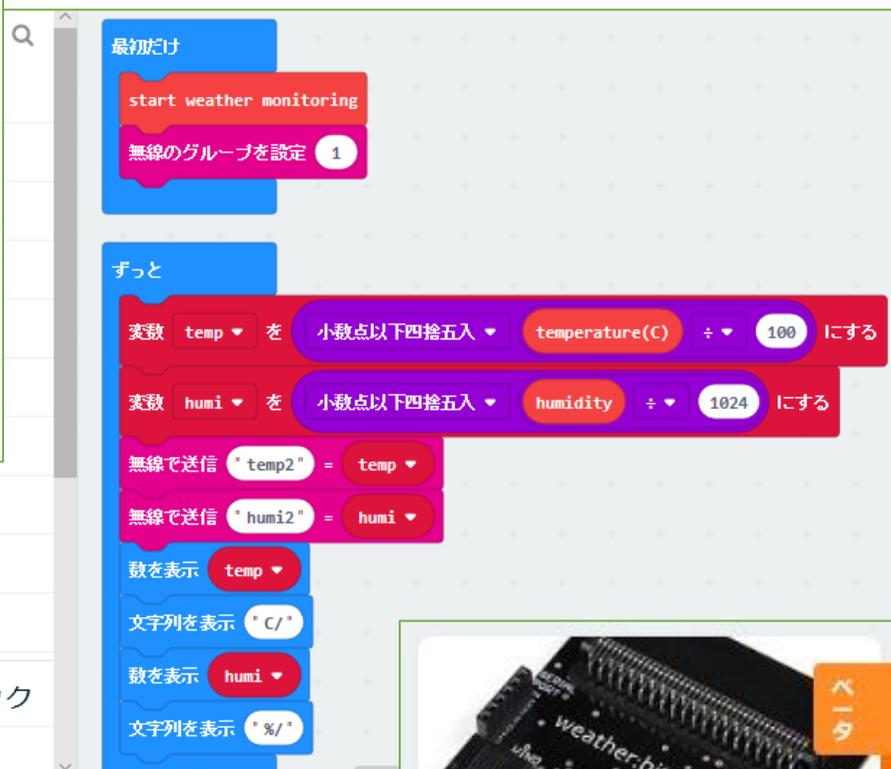
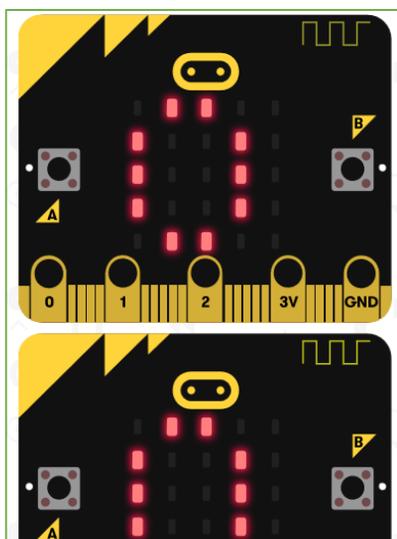
ボ	セ	L	出
通	入	オ	

・・・LED・出力・通信

2019/11/7作成

## 063 weather:bitの数値を同時送信

ボ	セ	シ	出
通	入	才	



※「064 weather:bitの数値を同時受信」とセットになります。

## 【説明】

### 063 weatherbitの数値を同時送信

<https://www.switch-science.com/catalog/3383/>

オプションパーツ（拡張基板）であるweather:bitをmicro:bitの端子部に差し込んで使います。Weather:bitは、気象観測センサー（気温・湿度・気圧）が組み込まれています。

専用のブロックを追加するには、「パッケージの追加」からweatherbitを検索します。

変数temp（気温用）とhumi（湿度用）を追加して、name（文字列）とvalue（数値）をセットで送信しています。

※「064 weather:bitの数値を同時受信」とセットになります。

ボ	セ	L	出
通	入	オ	

・・・ボタン・LED・通信・オプション

2019/11/11作成

# 064 weather:bitの数値を同時受信

ボ	セ	L	出
通	入	才	

The screenshot shows the MakeCode IDE interface. On the left is a sidebar with a search bar and a category list: 基本, 入力, 音楽, LED, 無線, ループ, 論理, 変数. The main workspace contains a script starting with a '最初だけ' (Run once) block containing:

- '無線のグループを設定' (1)
- '変数 temp3 を 0 にする'
- '変数 humi3 を 0 にする'

Below this is a '無線で受信したとき' (When received wirelessly) block with a 'name' and 'value' input. The logic is as follows:

- When 'name' is 'temp2', set 'temp3' to 'value'.
- When 'name' is 'humi2', set 'humi3' to 'value'.
- When 'temp3' is not 0, display 'temp3' as a number and 'C/' as a string.
- When 'humi3' is not 0, display 'humi3' as a number and '%/' as a string.

**weather-bit**  
MakeCode package for the SparkFun weather:bit board - beta

※「063 weather:bitの数値を同時送信」とセットになります。

## 【説明】

### 064 weatherbitの数値を同時受信

<https://www.switch-science.com/catalog/3383/>

オプションパーツ（拡張基板）であるweather:bit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。

※「063 weather:bitの数値を同時送信」とセットになります。

ボ	セ	L	出
通	入	オ	

・・・LED・通信

2019/11/11作成

## 065 センサーの温度・湿度数値を同時送信

ボ	セ	シ	出
通	入	才	



※「066 センサーの気温・湿度数値を同時送信」とセットになります。

## 【説明】

### 065 センサーの温度・湿度数値を同時送信

<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://makecode.microbit.org/58591-53241-75197-35498>

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

※「066 センサーの気温・湿度数値を同時受信」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

## 066 センサーの気温・湿度数値を同時受信

ボ	セ	L	出
通	入	才	

The image displays a Scratch code editor with a project titled "066 センサーの気温・湿度数値を同時受信". The code is organized into three main sections:

- 最初だけ (Initially):** A blue "最初だけ" block containing three red blocks: "無線のグループを設定" (Wireless group set) set to 1, "変数 temp3 を 0 にする" (Set variable temp3 to 0), and "変数 humi3 を 0 にする" (Set variable humi3 to 0).
- 無線で受信したとき (When received wirelessly):** A pink "無線で受信したとき" block with a dropdown menu set to "name" and a "value" input field. It contains two red blocks: "変数 temp3 を 0 にする" and "変数 humi3 を 0 にする".
- データ処理 (Data Processing):** A large teal block containing several conditional blocks:
  - "もし name = 'temp2' なら" (If name is 'temp2') followed by "変数 temp3 を value にする" (Set variable temp3 to value).
  - "もし name = 'humi2' なら" (If name is 'humi2') followed by "変数 humi3 を value にする" (Set variable humi3 to value).
  - "もし temp3 ≠ 0 なら" (If temp3 is not 0) followed by "数を表示 temp3" (Show number temp3) and "文字列を表示 'C/'" (Show string 'C/').
  - "もし humi3 ≠ 0 なら" (If humi3 is not 0) followed by "数を表示 humi3" (Show number humi3) and "文字列を表示 '%/'" (Show string '%/').

At the bottom right, there is an image of a micro:bit (TFW-MB) with a FabWorks TFW-EN1 sensor module attached. The sensor is labeled "温度・湿度・気圧センサー (TFW-EN1)".

※「065 センサーの気温・湿度数値を同時送信」とセットになります。

## 【説明】

### 066 センサーの温度・湿度数値を同時受信

<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。

※「065 センサーの気温・湿度数値を同時送信」とセットになります。

ボ	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+」 (第1版) microPython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 「HelloWorld」と「ハートマーク」を表示する

002 LED上に「O」を表示する

003 傾きを表示する

004 0から9まで表示する

# micro:bit microPythonプログラミングサイト

<https://microbit.org/ja/code/>

micro:bit    プログラムしよう    アイデア    はじめよう    教える    購入    日本語

## プログラムで想像力を発揮する

BBC micro:bit は、ブロック、JavaScript、Pythonを使ってプログラムできるって知ってましたか？

もしBBC micro:bitを使ったことがなければ、試してみてください**クイックスタートガイド**（簡単にはじめる方法）。

### MakeCodeエディター

Microsoft提供のMakeCodeエディターを使い

エディターへのアクセスに問題があつたり、エディターをオフラインで使いたい場合は、FAQをご確認ください。

[プログラムしよう](#)

[リファレンス](#)

[Lessons](#)

### Pythonエディター

Pythonエディターは自分のプログラミングのスキルをさらに向上させたい人に最適です。選ばれたプログラム例とあらかじめ用意された様々な画像や音楽は、プログラミングの手助けになります。グローバルなPythonコミュニティの成果物によって動いています。Find out more about the [latest updates in version 2 of the Python Editor](#).

[プログラムしよう](#)

[リファレンス](#)

## オンラインエディター

ダウンロード on the App Store

micro:bitアプリでは、Bluetoothを使用してワイヤレスであなたのmicro:bitにプログラムを送り込みます。ケーブルは必要ありません！使用方法についての詳細については、このページを見てください。 [アプリケーション \(ソフトウェア\)](#)。

## 【説明】

micro:bit プログラミングサイト

<https://microbit.org/ja/code/>

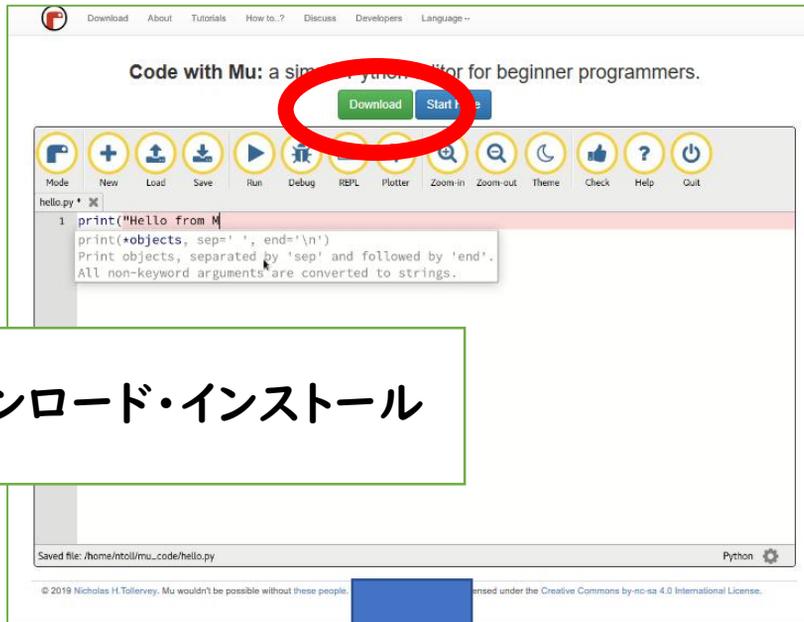
のPythonエディターでオンラインプログラミングします。

・Download→プログラムをダウンロードして、micro:bitへ送る

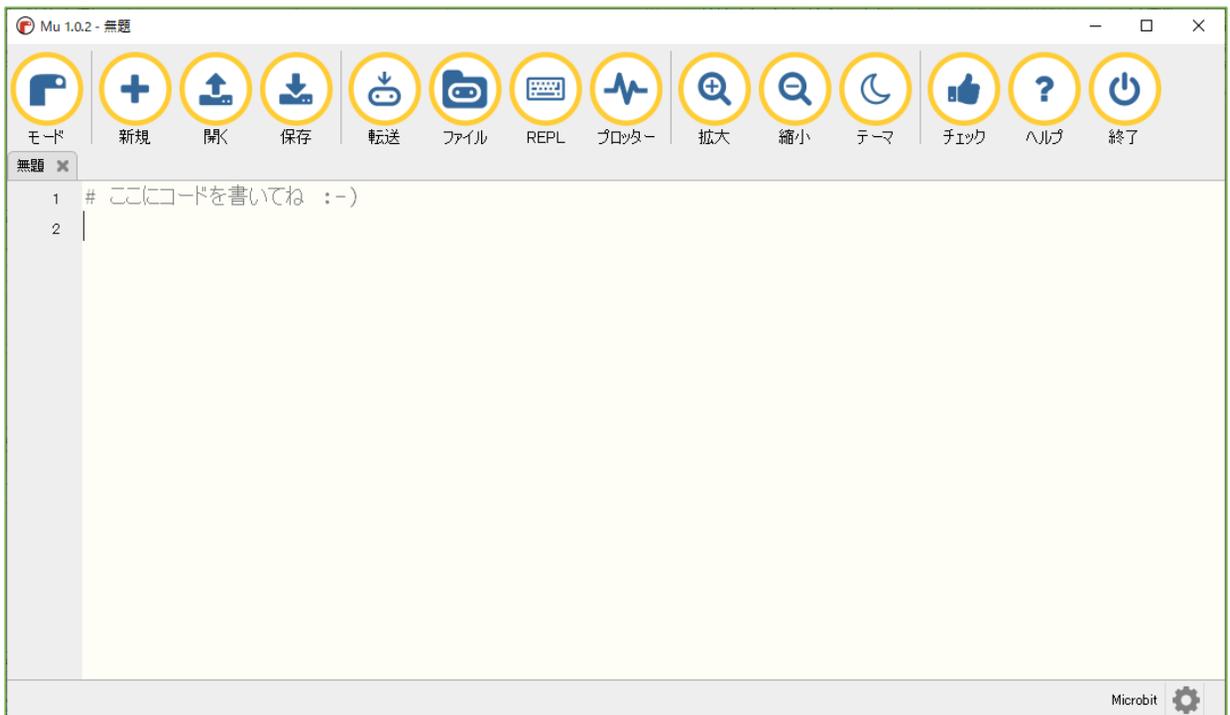
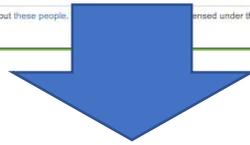
・Load/Save→\*.py (Mu編集用ファイル)やプログラムを読み込み・保存できる

# micro:bit microPythonプログラミングツール Mu

<https://codewith.mu/>



ダウンロード・インストール



## 【説明】

micro:bit プログラミングツール Mu サイト

<https://codewith.mu/>

からダウンロード・インストールします。

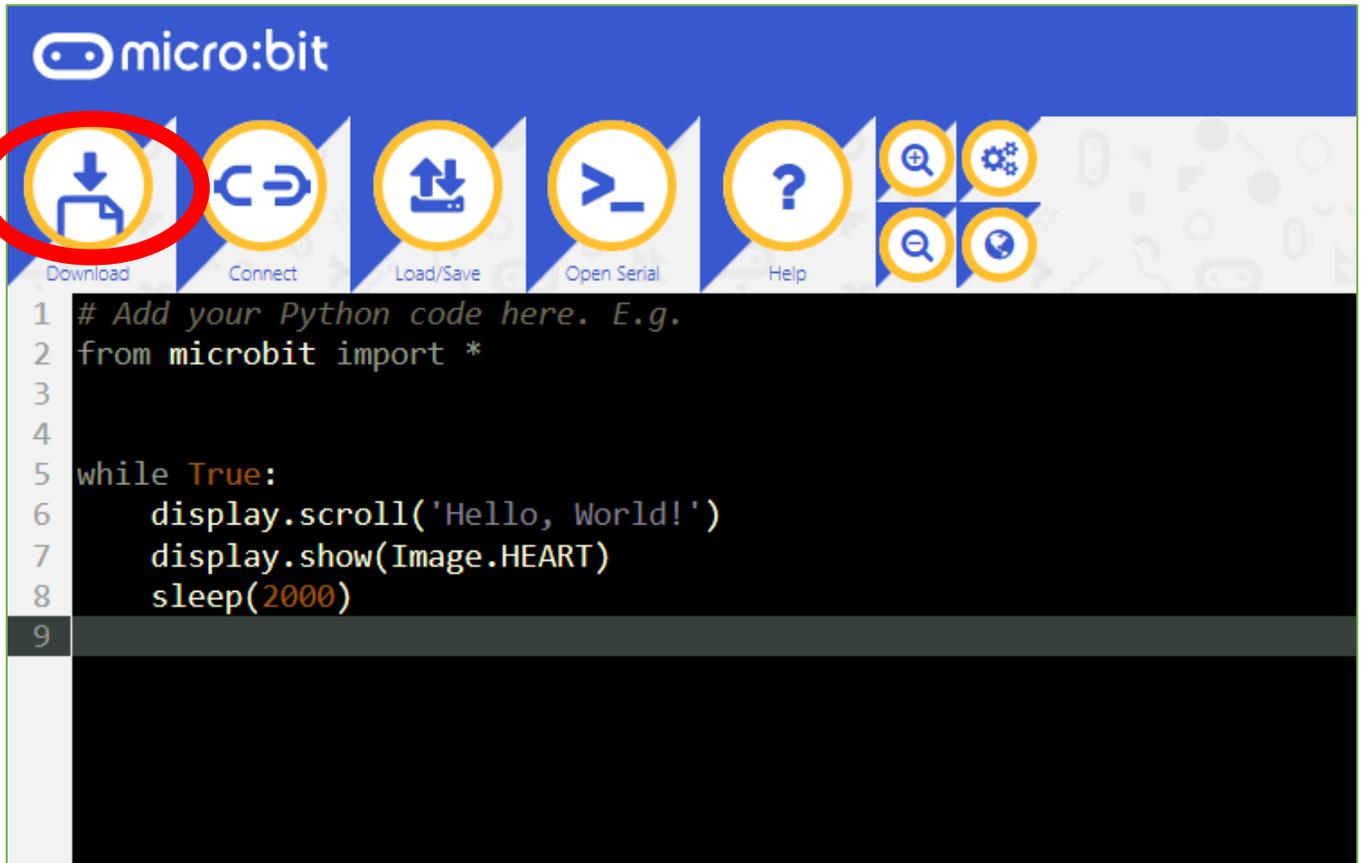
※オンラインエディターとの違い

- ・転送→プログラムを接続されているmicro:bitへ
- ・保存→\*.py (Mu編集用ファイル)

## 001 「HelloWorld」と「ハートマーク」を表示する

ボ	セ	L	出
通	入	才	

<https://python.microbit.org/v/2.0>



The screenshot shows the MicroPython IDE interface. The 'Download' button is circled in red. The code editor contains the following Python code:

```
1 # Add your Python code here. E.g.
2 from microbit import *
3
4
5 while True:
6     display.scroll('Hello, World!')
7     display.show(Image.HEART)
8     sleep(2000)
9
```

# …プログラムに関係しないメモ  
from microbit import \* …micro:bit用モジュールの読み込み  
while True …ずっと繰り返す  
display.scroll(' ')…LEDに文字列を表示  
display.show(Image.HEART)…LEDに「ハート」を表示  
sleep(2000)…2000ミリ秒=2秒停止

ダウンロード→micro:bitに転送

## 【説明】

ボ	セ	L	出
通	入	オ	

# IMAGEの種類

Image.HEART ハート  
Image.HEART\_SMALL 小ハート  
Image.HAPPY 笑顔  
Image.SMILE 笑った口  
Image.SAD 悲しい顔  
Image.CONFUSED 困り顔  
Image.ANGRY 怒った顔  
Image.ASLEEP 眠っている顔  
Image.SURPRISED びっくり顔  
Image.SILLY 変顔  
Image.FABULOUS 素晴らしい顔  
Image.YES はい(チェック)  
Image.NO いいえ(×)  
Image.CLOCK12, Image.CLOCK11,  
Image.CLOCK10, Image.CLOCK9,  
Image.CLOCK8, Image.CLOCK7,  
Image.CLOCK6, Image.CLOCK5,  
Image.CLOCK4, Image.CLOCK3,  
Image.CLOCK2, Image.CLOCK1  
0時(短針を表示)  
Image.ARROW\_N,  
Image.ARROW\_NE,  
Image.ARROW\_E,  
Image.ARROW\_SE,  
Image.ARROW\_S,  
Image.ARROW\_SW,  
Image.ARROW\_W,  
Image.ARROW\_NW  
矢印表示(上を北として)  
Image.TRIANGLE 三角  
Image.TRIANGLE\_LEFT 左三角  
(RIGHTはない)

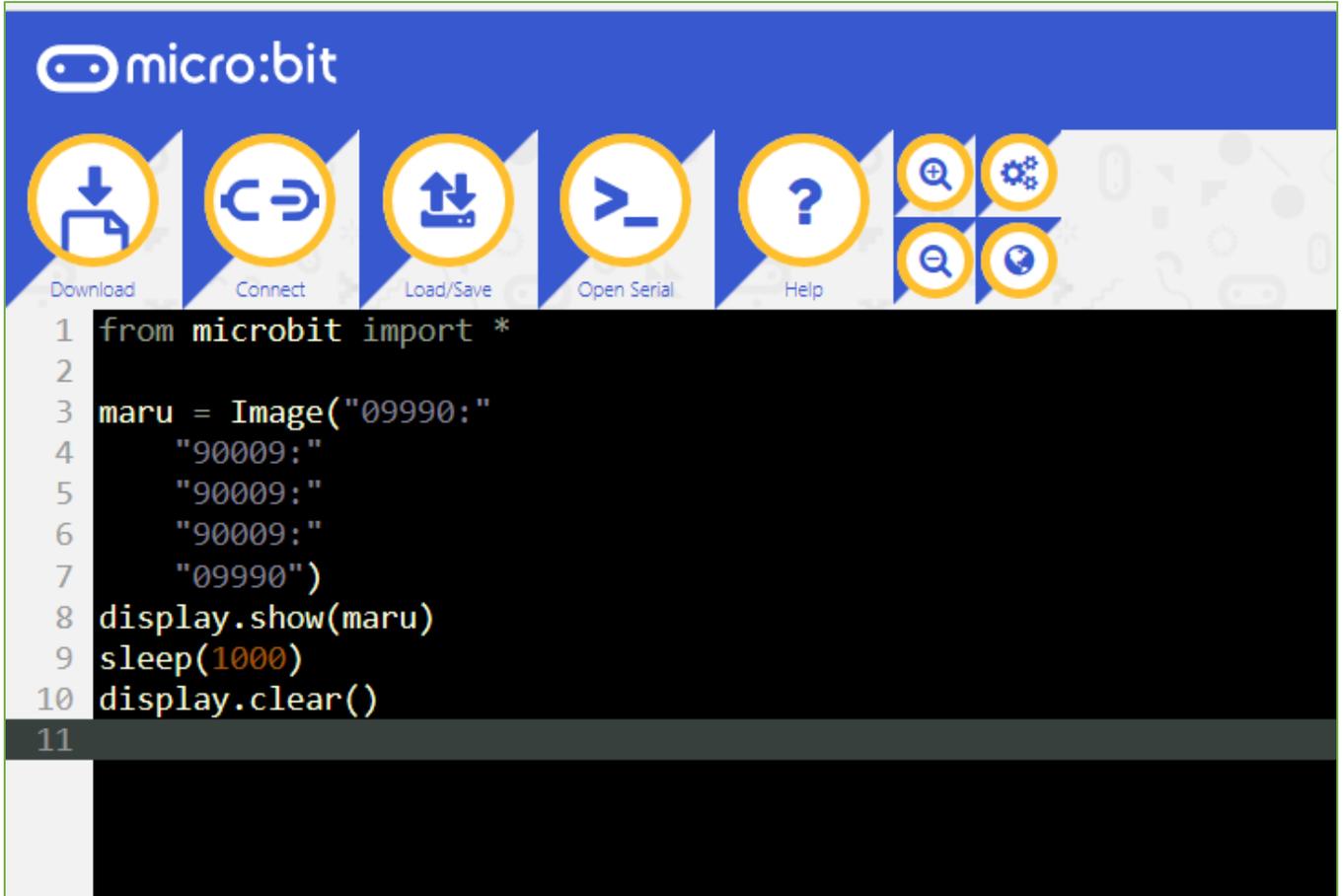
Image.CHESSBOARD チェス盤  
Image.DIAMOND ダイヤ  
Image.DIAMOND\_SMALL 小ダイヤ  
Image.SQUARE 四角  
Image.SQUARE\_SMALL 小四角  
Image.RABBIT うさぎ  
Image.COW 牛  
Image.MUSIC\_CROTCHET 四分音符  
Image.MUSIC\_QUAVER 八分音符  
Image.MUSIC\_QUAVERS つないだ八分音符  
Image.PITCHFORK フォーク  
Image.XMAS クリスマス  
Image.PACMAN パックマン  
Image.TARGET ターゲット  
Image.TSHIRT Tシャツ  
Image.ROLLERSKATE ローラースケート  
Image.DUCK アヒル  
Image.HOUSE 家  
Image.TORTOISE カメ  
Image.BUTTERFLY チョウ  
Image.GHOST おばけ  
Image.SWORD 剣  
Image.GIRAFFE キリン  
Image.SKULL どくろ  
Image.UMBRELLA 傘  
Image.SNAKE ヘビ

文部科学省  
高等学校情報科「情報Ⅰ」教員研修用教材より  
※オプション例は2019/9 別途調べました。

## 002 LED上に「O」を表示する

ボ	セ	L	出
通	入	才	

<https://python.microbit.org/v/2.0>



The screenshot shows the micro:bit Python editor interface. At the top, there is a blue header with the "micro:bit" logo and several icons for "Download", "Connect", "Load/Save", "Open Serial", "Help", and a search/settings menu. Below the header is a code editor with the following Python code:

```
1 from microbit import *
2
3 maru = Image("09990:"
4             "90009:"
5             "90009:"
6             "90009:"
7             "09990")
8 display.show(maru)
9 sleep(1000)
10 display.clear()
11
```

OO=Image("09990:" ...0がLED消灯、9が最も明るい  
...

"09990:")...変数として設定  
display.show(OO)...変数を表示  
display.clear()...LED表示を消す

## 【説明】

### 002 LED上に「O」を表示する

・・・順次の例として、LEDで○を1秒間点灯したあと消灯するプログラムです。

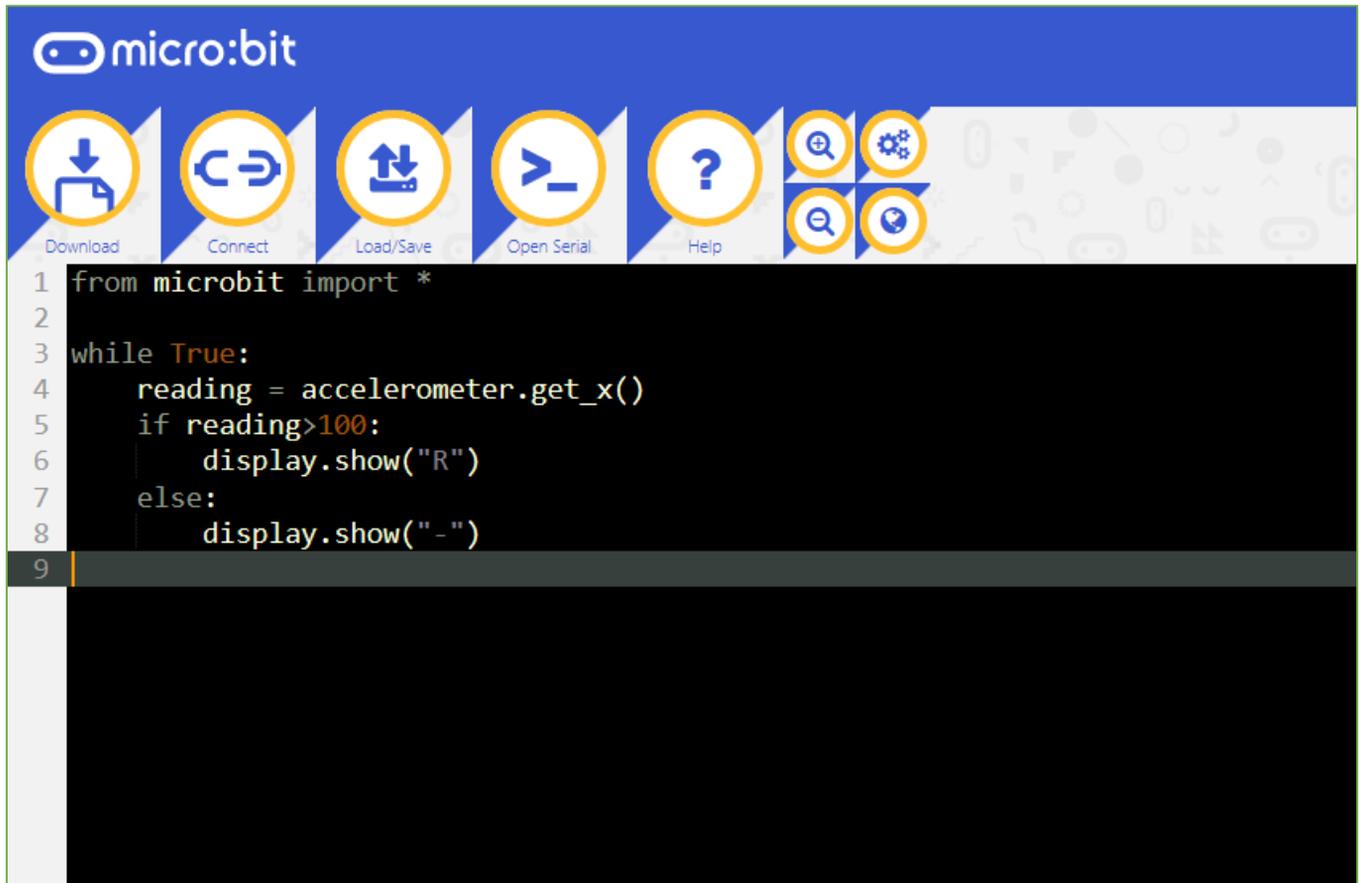
ボ	セ	L	出
通	入	才	

文部科学省  
高等学校情報科「情報Ⅰ」教員研修用教材より

## 003 傾きを表示する

ボ	セ	L	出
通	入	才	

<https://python.microbit.org/v/2.0>



The screenshot shows the MicroPython IDE interface. At the top, there is a blue header with the "micro:bit" logo and several icons: Download, Connect, Load/Save, Open Serial, Help, and a search/settings menu. Below the header is a code editor with the following Python code:

```
1 from microbit import *
2
3 while True:
4     reading = accelerometer.get_x()
5     if reading > 100:
6         display.show("R")
7     else:
8         display.show("-")
9
```

`reading=accelerometer.get_x()`

- ・・・変数に加速度センサーの値を入力
- ・・・>100 右に傾き  
( <100 左に傾き =100 左右水平)

※`accerometer.get_y()`

- ・・・>100 手前に傾き <100 向こうに傾き

## 【説明】

### 003 傾きを表示する

・・・分岐の例として、micro:bitの加速度センサが右に傾いているデータを検出すれば、LEDに「R」が点灯し、そうでなければLEDに「-」が点灯するプログラムです。

文部科学省  
高等学校情報科「情報Ⅰ」教員研修用教材より

### ※温度センサーの例

```
1 from microbit import *
2
3 while True:
4     reading = temperature()
5     display.scroll(str(reading)+'C/')
6     sleep(1000)
7
```

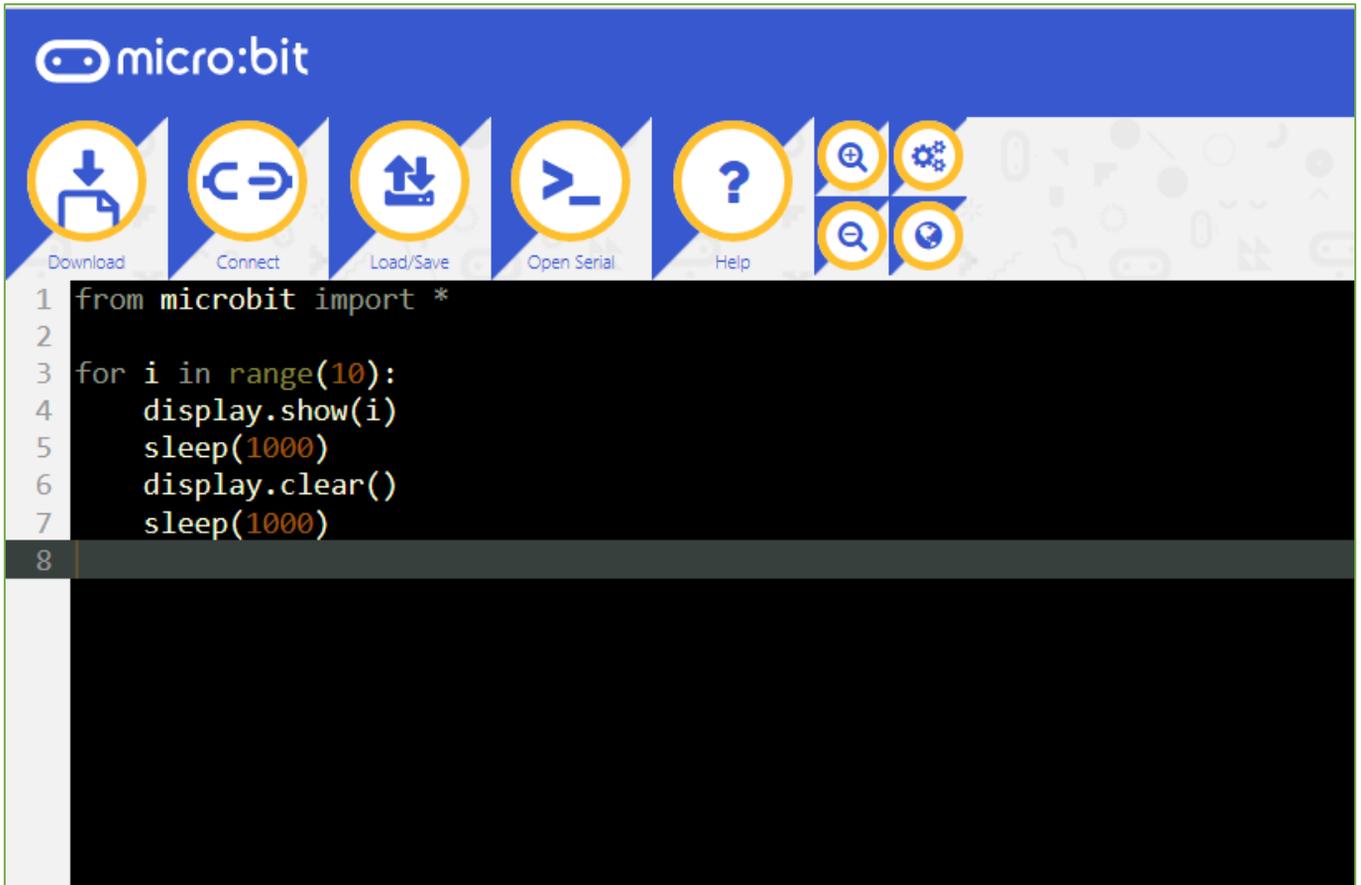
ボ	セ	L	出
通	入	才	

2020/1/13作成

## 004 0から9までを表示する

ボ	セ	リ	出
通	入	才	

<https://python.microbit.org/v/2.0>



The screenshot shows the MicroPython IDE interface. At the top, there is a blue header with the "micro:bit" logo and several icons: Download, Connect, Load/Save, Open Serial, Help, and a 2x2 grid of icons (Magnifying Glass, Gear, Search, Refresh). Below the header is a code editor with a black background and white text. The code is as follows:

```
1 from microbit import *
2
3 for i in range(10):
4     display.show(i)
5     sleep(1000)
6     display.clear()
7     sleep(1000)
8
```

for i in range(10)・・・i を0～9までカウントしながら繰り返し  
display.show(i)・・・i に入っている値を表示

## 【説明】

### 004 0から9までを表示する

・・・反復の例として、0から9までの数字を順に表示するプログラムです。

ボ	セ	ル	出
通	入	才	

文部科学省  
高等学校情報科「情報Ⅰ」教員研修用教材より



**micro:bit 「サンプル  
プログラミング集」  
(第3.0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