

micro:bit 「サンプル プログラミング集」 (第4.0版) 【ブロック・microPython・Pythonプログラミング】

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

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 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

- 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

【microPython編】

- 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までを表示する

【Pythonプログラミング編】

関数など

- 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
- ### 条件(if文)

- 006 暗くなるとLEDが点く
- ### 音

- 007 Happy Birthdayを鳴らす
- ### 複数の条件

- 013 コンパス(Compass)

無線

- 014 2台で送受信

傾き

- 015 水準器

- 016 傾きメーター

乱数と繰り返し

- 044 ランダム数値表示

- 045 じゃんけん

micro:bit プログラミングサイト

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

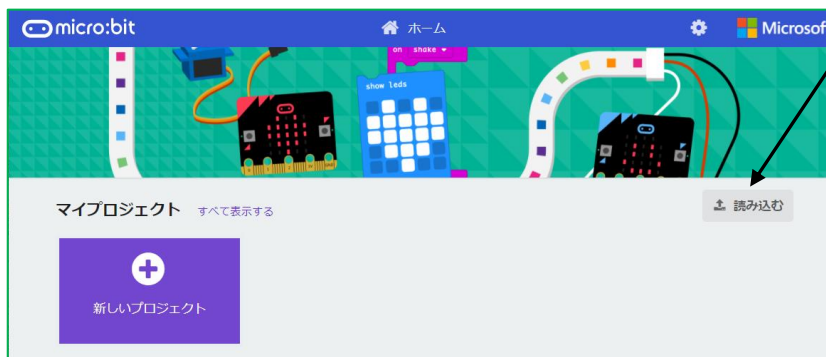
ホーム画面へ

Python・JavaScript
編集画面への移動



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。

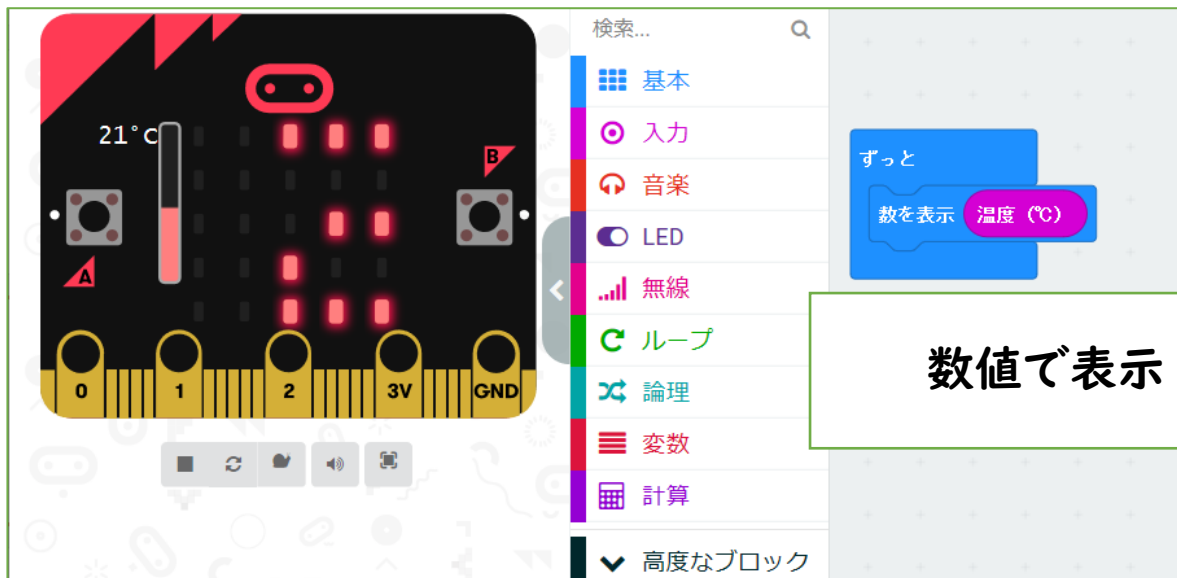
- ①ブロックを組み合わせてプログラミング。
- ②確認用シミュレータで確認する。
- ③USB接続したmicro:bit にプログラムをコピー。
- ④micro:bit 本体で動作させる。

(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

【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。

ボ	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。

ボ	セ	L	出
通		オ	

- ・・・使用するのは、センサー・LED

2018/4/18作成

ボ	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

ボ	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	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

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	リ	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	



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

ボ	セ	L	出
通		オ	

007 Happy Birthday を鳴らす



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

スピーカーやイヤホンをつなぐ

ボタン A 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

ボ	セ	L	出
通		オ	

008 電子オルゴールを鳴らす



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

P1端子とGND端子の間に電流が流れると動作

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

【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。

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。

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



検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算

端子 P1 がタッチされたとき

音を鳴らす 高さ (Hz) 真ん中のド 長さ 1 拍

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



検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算

ずっと

もし 明るさ < 20 なら

音を鳴らす 高さ (Hz) 真ん中のド

でなければもし 明るさ < 30 なら

音を鳴らす 高さ (Hz) 真ん中のレ

でなければ

音を鳴らす 高さ (Hz) 真ん中のミ

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」音が鳴ります。

ボ	セ	L	出
通		オ	

・・・センサー・出力

010 明るさで音が変わるテルミン

2018/4/23作成

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。

ボ	セ	L	出
通		オ	

・・・センサー・出力

2018/4/23作成

ボ	セ	L	出
通		才	

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。

・・・ボタン・LED

ボ	セ	L	出
通		オ	

2018/5/6作成

012 明るさで秒数計測

ボ	セ	シ	出
通		オ	

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。

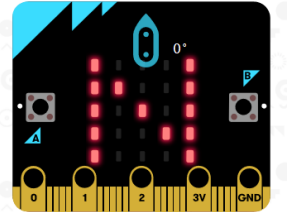
ボ	セ	Ｌ	出
通		オ	

・・・ボタン・センサー・LED

2018/5/6作成

013 コンパス (Compass)

ボ	セ	し	出
通		才	



検索...

Q

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】

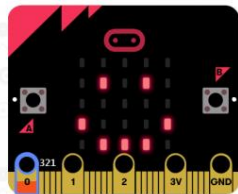
013 コンパス (Compass)

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	し	出
通		才	

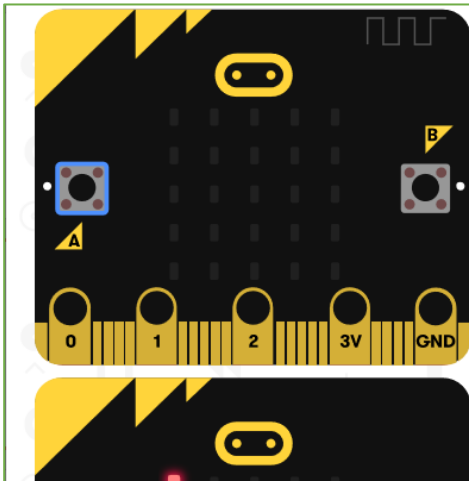


・・・センサー・LED

2018/5/14作成

ボ	セ	L	出
通		オ	

014 2台で送受信



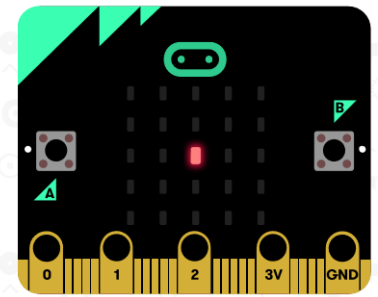
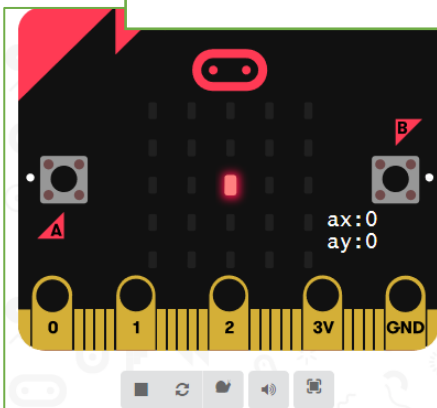
Bluetoothによる通信



ボ	セ	L	出
通		オ	

015 水準器

傾きセンサーで計測



【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)

ボ	セ	シ	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

ボ	セ	シ	出
通		オ	

2018/5/21作成

016 傾きメーター

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 $-1 \times \text{加速度} \times 1000 \div 400$ 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

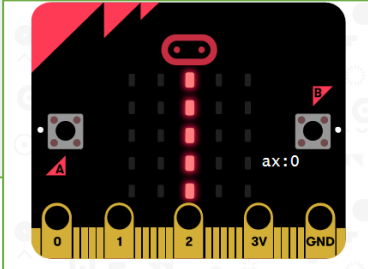
表示を消す

最初だけ

変数 X を 2 にする

傾きをLED表示

を $-1 \times \text{加速度} \times 1000 \div 400$ にする



【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。

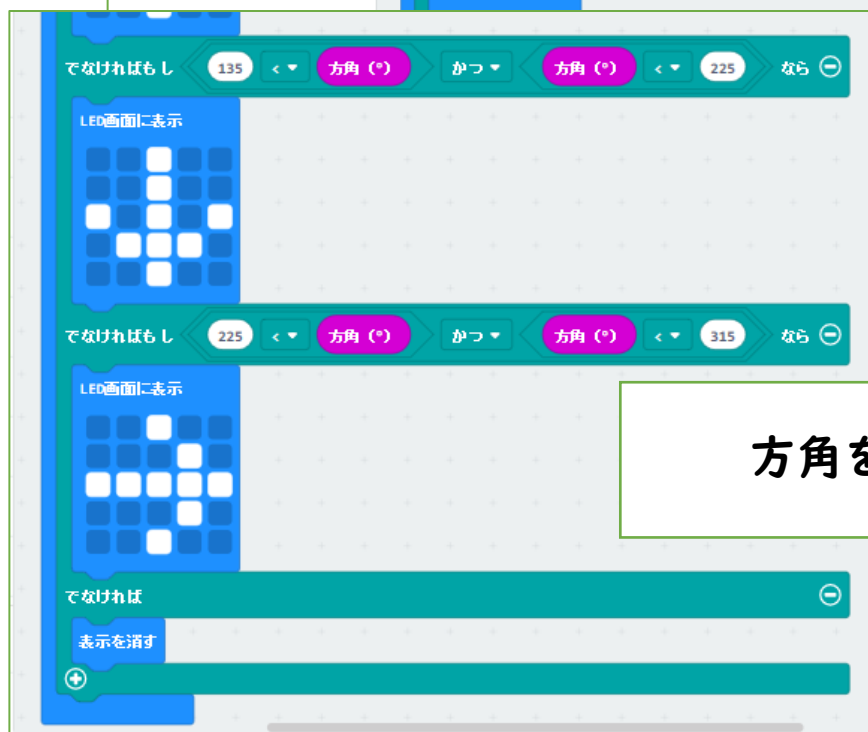
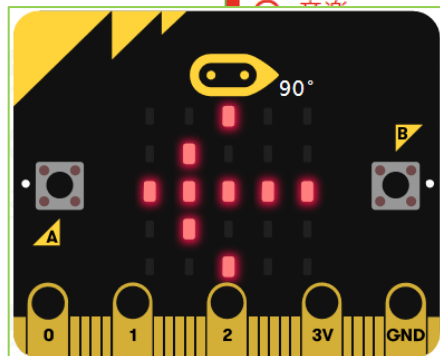
ボ	セ	L	出
通		オ	

・・・ボタン・LED・通信

2018/5/21作成

017 方位磁針

ボ	セ	シ	出
通		才	



方向を→表示

【説明】

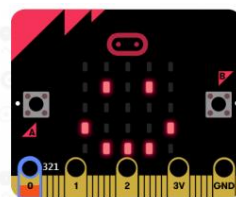
017 方位磁針

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	シ	出
通		才	



・・・センサー・LED

2018/5/23作成

ボ	セ	し	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

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 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】

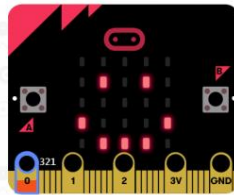
018 方位磁針16方位-1

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。

ボ	セ	L	出
通		才	



・・・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	シ	出
通		才	

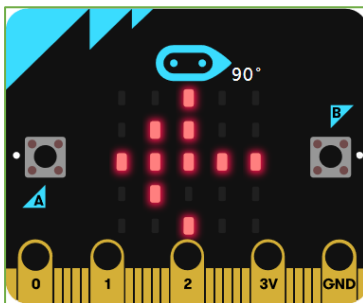
検索...

- 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

```

        if (direction < 157) then
            LED matrix display
        if (direction < 202) then
            LED matrix display
        if (direction < 247) then
            LED matrix display
        clear display
    
```



【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 I 6方位program分割- I

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

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。

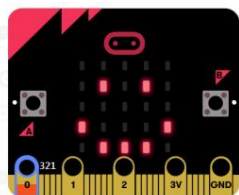
5×5のLEDで、図形を表示する方法を模索したプログラムですが、8方位版と違い、より矢印の図形を表すのが難しくなっています。

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。

ボ	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 019-2)

019 方位磁針I6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

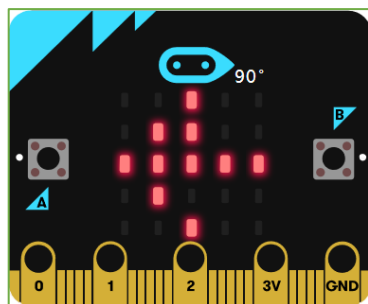
ずっと

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

長いプログラムを分割



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】

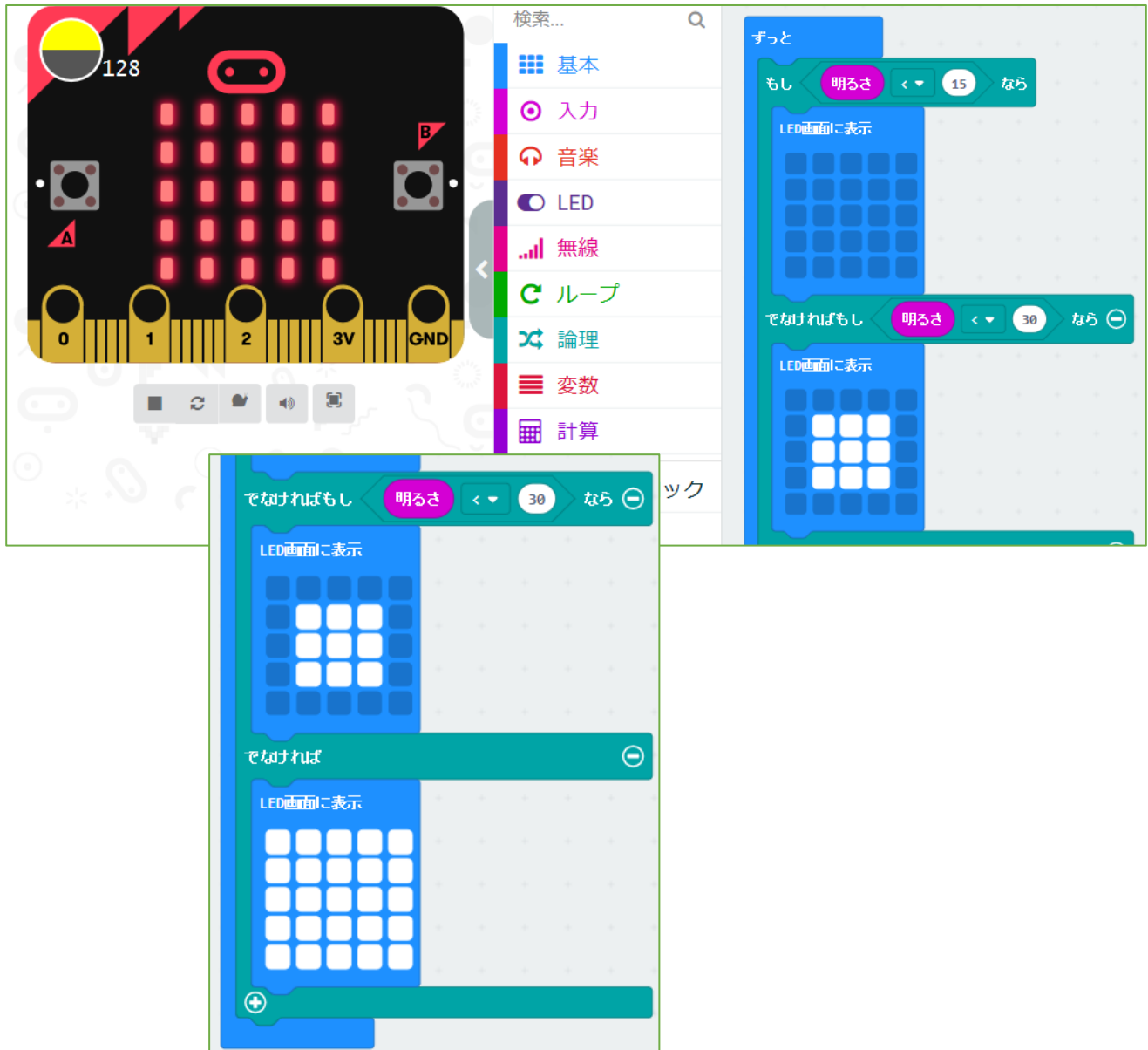
019 方位磁針16方位program分割-2

019-1の続きになります。

2018/5/24作成

ボ	セ	し	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。

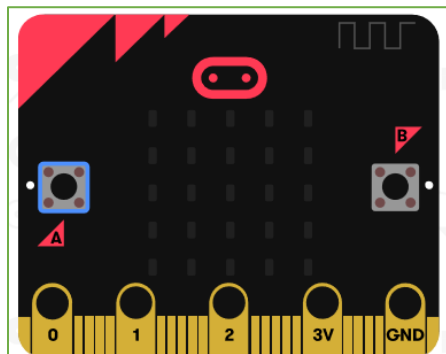
ボ	セ	シ	出
通		オ	

・・・センサー・LED

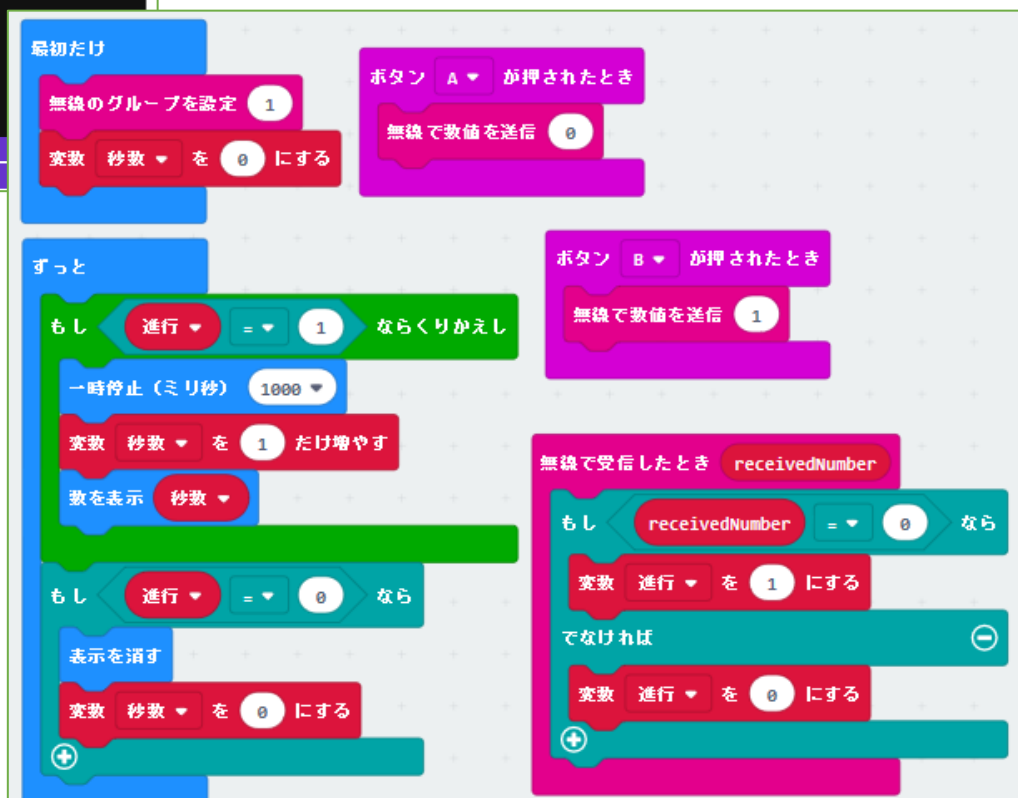
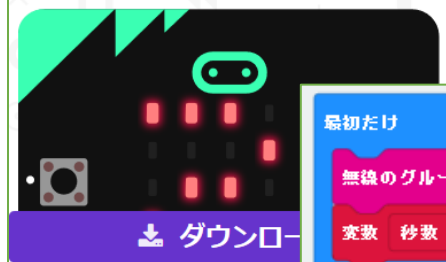
2018/5/28作成
2019/1/6修正

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



【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。

ボ	セ	し	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	

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

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

もし receivedNumber = 0 なら

音を鳴らす 高さ (Hz) 真ん中のレ 長さ 1 拍

一対多送信可能

023 明るさ変化を送信

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

最初だけ

無線のグループを設定 1

LED画面に表示

ずっと

もし 明るさ < 20 なら

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

LED画面に表示

一時停止 (ミリ秒) 100

LED画面に表示

【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

検索...

- 基本
- 入力
- 音楽
- LED

最初だけ

- 無線のグループを設定 1
- LED画面に表示
- ずっと
 - もし 加速度 X > 200 なら
 - 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
 - LED画面に表示
 - 一時停止 (ミリ秒) 100
 - LED画面に表示

ボ	セ	L	出
通		オ	

025 明るさ数値送信

検索...

- 基本
- 入力
- 音楽
- LED
- 無線
- ループ
- 論理
- 変数
- 計算

最初だけ

- 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
- 無線で受信したとき receivedNumber
 - 変数 受信数値 を receivedNumber にする
 - 数を表示 受信数値

ボタン A が押されたとき

- 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

【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	

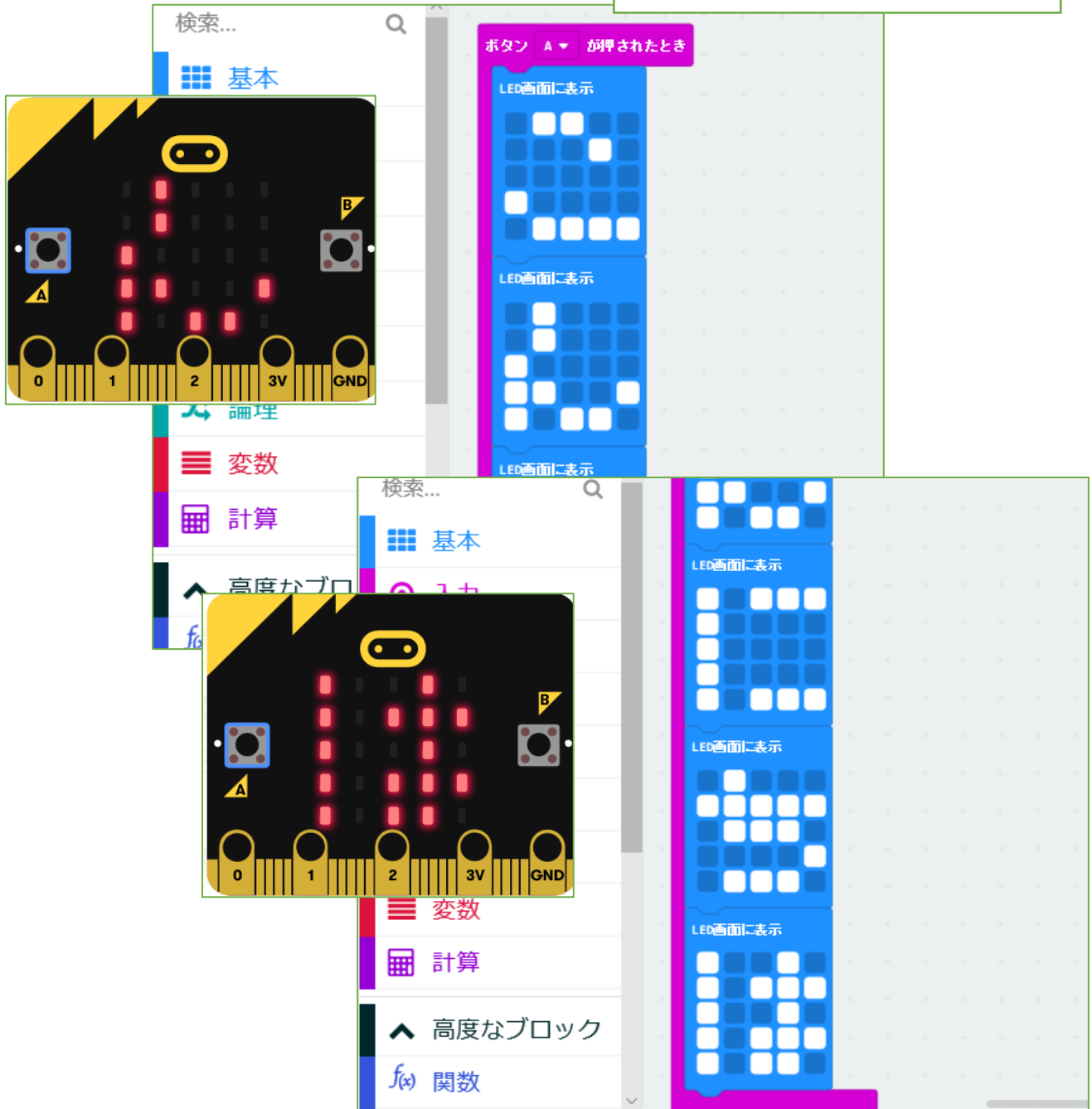
・・・ボタン・センサー・LED・通信

2018/6/18作成

026 ひらがな表示

ボ	セ	L	出
通		才	

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。

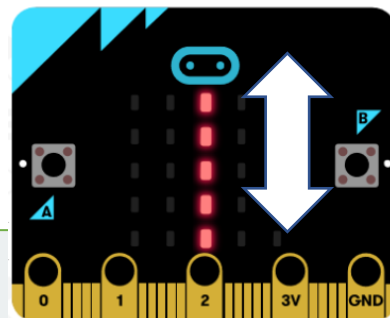
ボ	セ	し	出
通		才	

・・・LED

2018/7/29作成

027 傾き (ピッチ) 角度を表示

ボ	セ	し	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

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

【説明】

027 傾き (ピッチ) 角度を表示

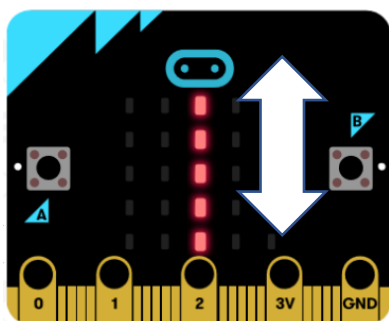
ボタンを押したときのmicro:bit 基板の角度 (ピッチのほう。もう一つの角度がロール) を使って表示するプログラムです。

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。

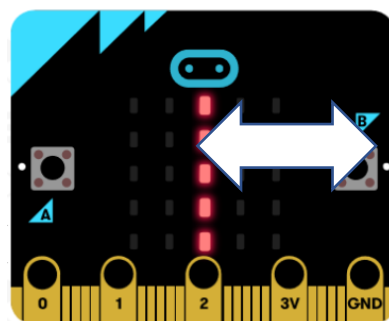
ボ	セ	し	出
通		オ	

2018/7/29作成

・・・ボタン・センサー・LED

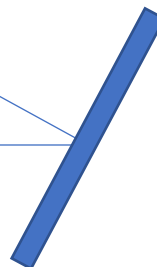


ピッチ



ロール

この角度



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 Scratch-like block-based programming environment. On the left, a circuit diagram of an Arduino Uno is shown with a potentiometer connected to the 5V, GND, and A0 pins. The potentiometer's wiper is connected to the A0 pin. The potentiometer is labeled with '128' and 'A+B'. The potentiometer's wiper is connected to the A0 pin. The potentiometer is labeled with '128' and 'A+B'.

The main workspace shows a script for a music block. The script is as follows:

```

ボタン A が押されたとき
もし 明るさ < 50 なら
 音を鳴らす 高さ (Hz) 真ん中のド
でなければもし 明るさ < 75 なら
 音を鳴らす 高さ (Hz) 真ん中のレ
でなければもし 明るさ < 100 なら
 音を鳴らす 高さ (Hz) 真ん中のミ
でなければもし 明るさ < 125 なら
 音を鳴らす 高さ (Hz) 真ん中のファ
でなければもし 明るさ < 150 なら
 真ん中のソ
 真ん中のラ
 真ん中のシ
  
```

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

【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。

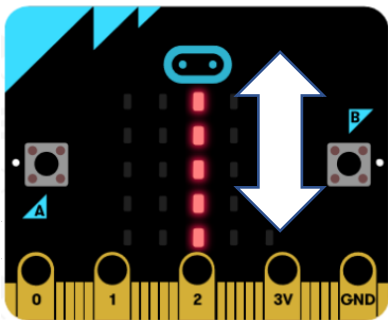
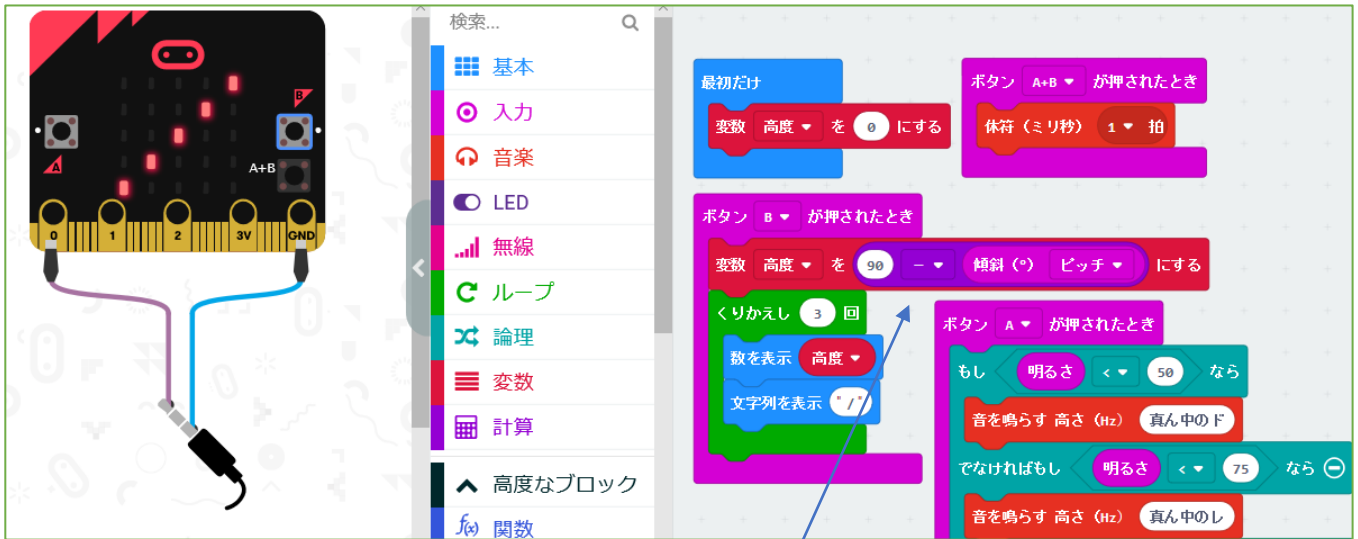
ボ	セ	ン	サ
ー		出	

・・・ボタン・センサー・出力

2018/7/29作成

029 太陽高度計

ボ	セ	シ	出
通		才	



太陽に向けて使用



【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。

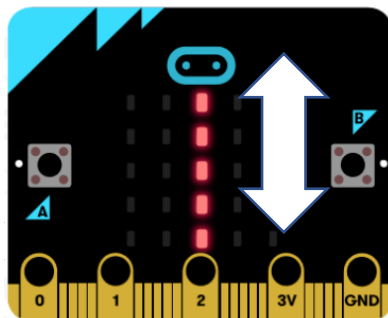
リアルタイムで明るさセンサーの値から音の高さを変えて出力するプログラムと、ボタンを押したときの micro:bit 基板の角度（ピッチのほう。もう一つの角度がロール）から太陽高度を表示するプログラムを組み合わせています。

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



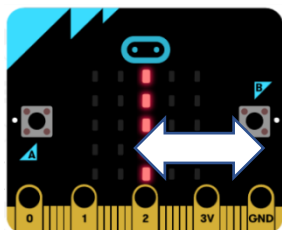
2018/8/6作成

030 ふりこ送信

ボ	セ	シ	出
通		オ	



最下点に来た時をカウント



【説明】

030 ふりこ送信

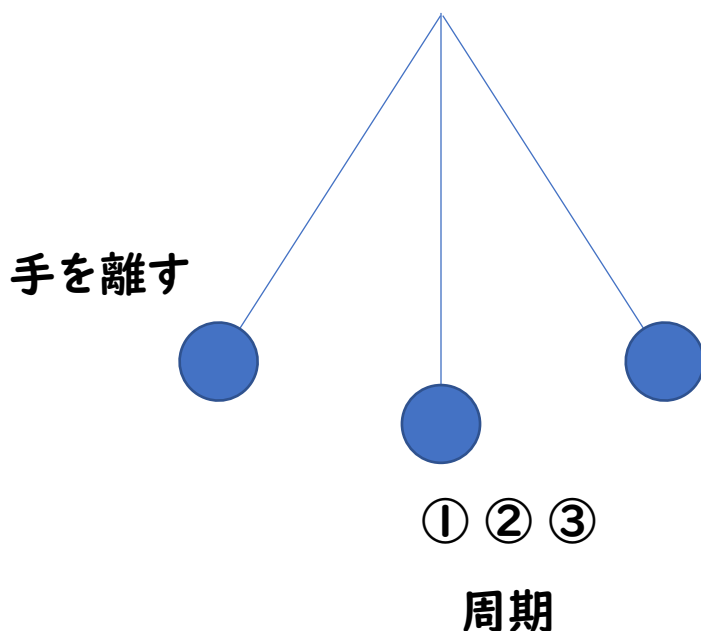
ふりこの周期を測るためのプログラムの一部です。「ふりこの最下点に来た時、ロールが0（水平）になる」ことを使っています。

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



ボ	セ	L	出
通		オ	

03 | ふりこ受信 段階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
- 関数

最初だけ

- 表示を消す
- 無線のグループを設定 1
- 変数 時間 を 0 にする
- 変数 Start を 0 にする
- 変数 回数 を 0 にする

無線で受信したとき receivedNumber

- もし receivedNumber = 0 なら
- 変数 回数 を 1 だけ増やす

ボタン A が押されたとき

- 表示を消す
- 無線のグループを設定 1
- 変数 時間 を 0 にする
- 変数 Start を 0 にする
- 変数 回数 を 0 にする

ずっと

- もし 回数 = 1 なら
- 変数 Start を 1 にする
- でなければ
- もし Start = 1 なら
- 一時停止 (ミリ秒) 100
- 変数 時間 を 100 だけ増やす
- 数を表示 時間 ÷ 100

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
- 関数

ボタン A が押されたとき

- 表示を消す
- 無線のグループを設定 1
- 変数 時間 を 0 にする
- 変数 Start を 0 にする
- 変数 回数 を 0 にする

ずっと

- もし 回数 = 1 なら
- 変数 Start を 1 にする
- でなければ
- もし Start = 1 なら
- 一時停止 (ミリ秒) 100
- 変数 時間 を 100 だけ増やす
- 数を表示 時間 ÷ 100
- 表示を消す
- でなければ

【説明】

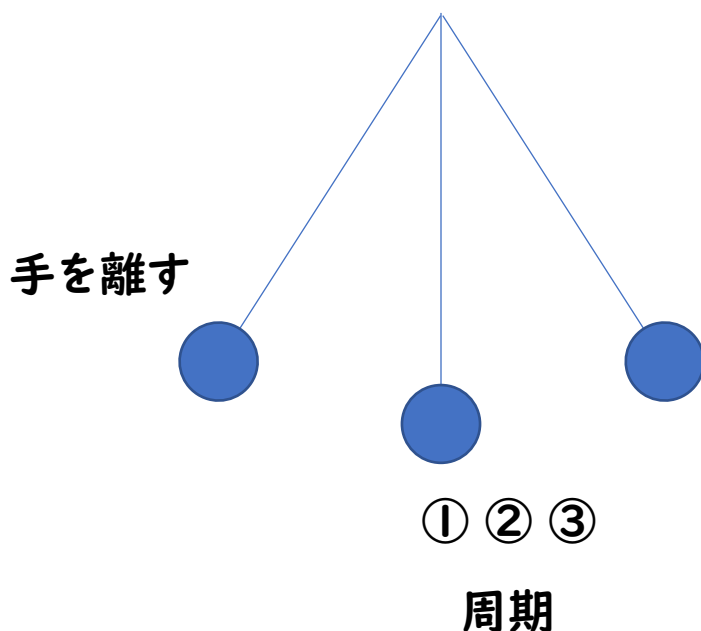
031 ふりこ受信 段階1

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。

ボ	セ	L	出
通		オ	

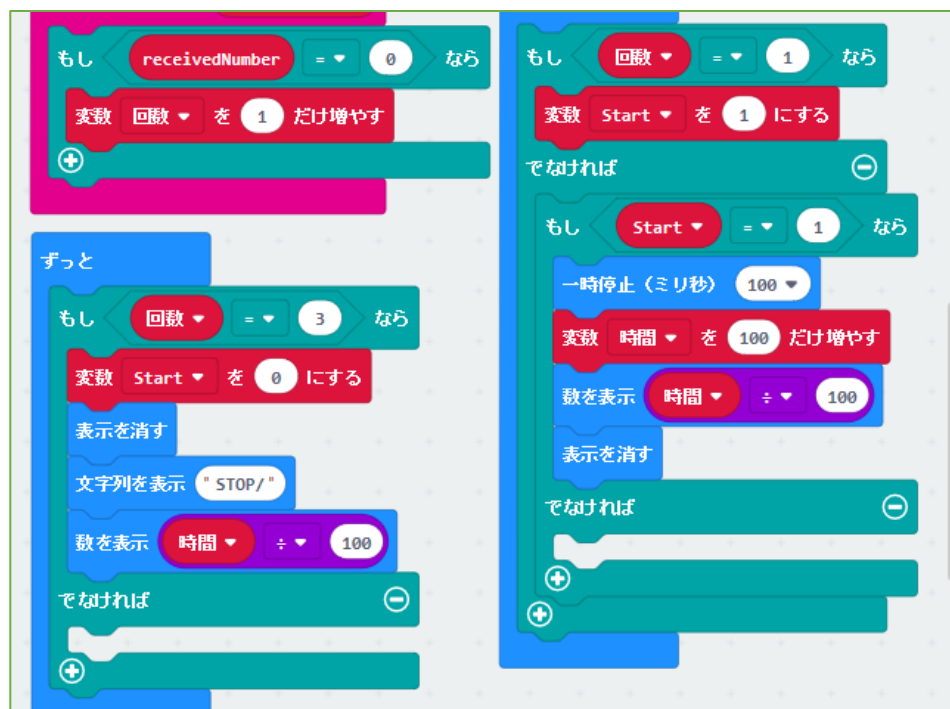
・・・ボタン・センサー・LED
(段階2があります)

2018/8/15作成



032 ふりこ受信 段階2

ボ	セ	シ	出
通		オ	



【説明】

032 ふりこ受信 段階2

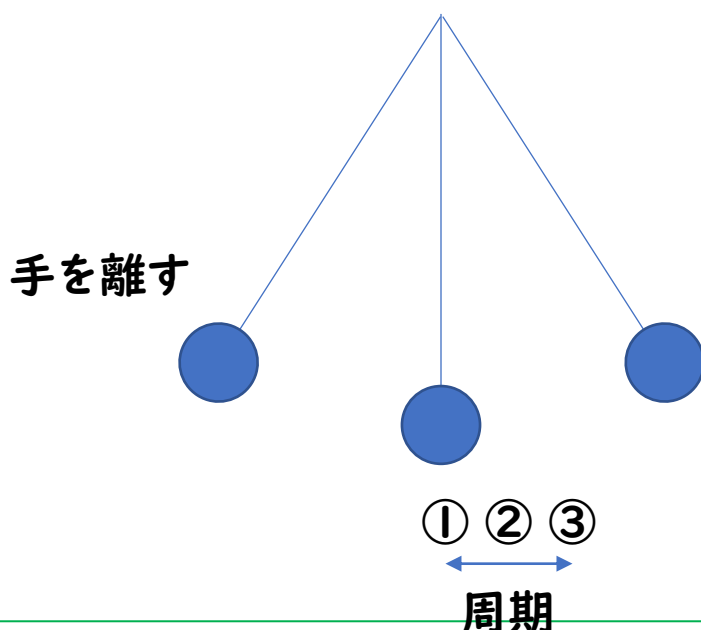
ふりこの周期を測るためのプログラムの一部です。
030「ふりこ送信」と組み合わせます。

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	し	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

ボ	セ	入	出
通		才	



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。

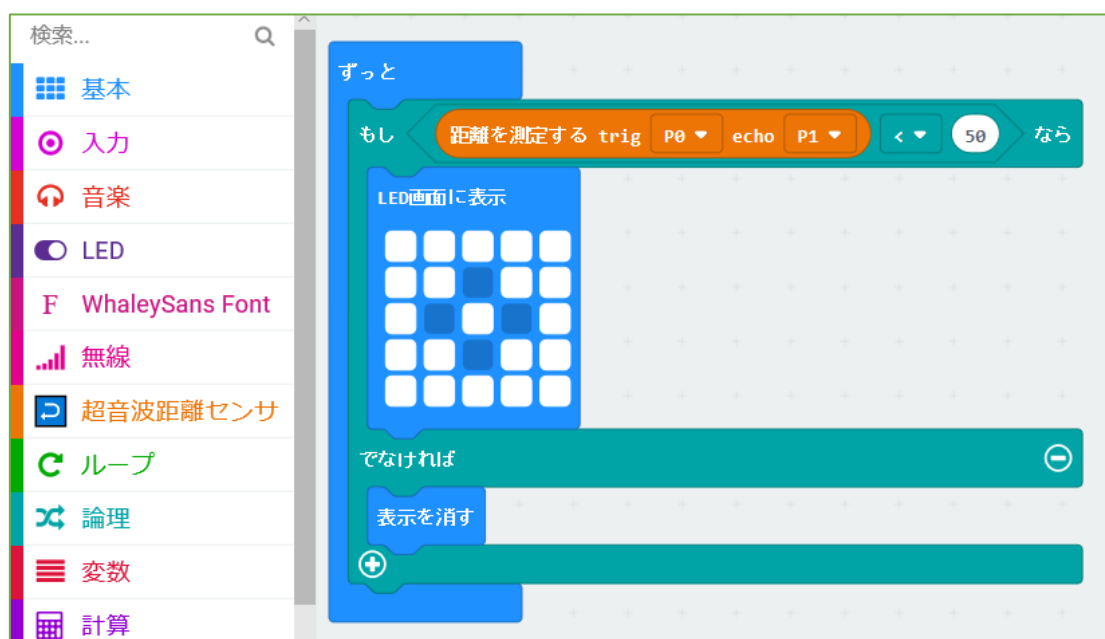
ボ	セ	シ	出
通		オ	

・・・ボタン・LED・通信

2018/8/27作成

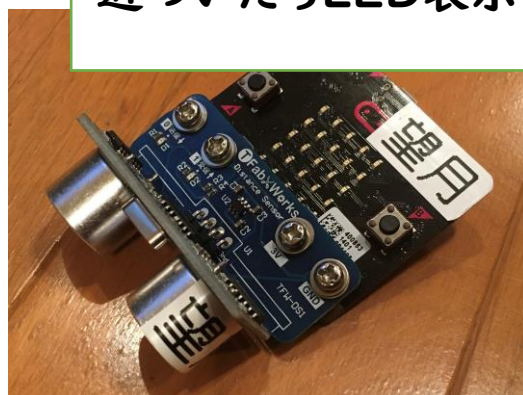
034 接近したら点灯（距離センサー）

ボ	セ	L	出
通		オ	



ワンタッチ距離セン
サー (TFabWorks)

近づいたらLED表示



【説明】

034 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	Ｌ	出
通		オ	

・・・センサー・LED・オプション

2018/9/24作成

ボ	セ	L	出
通		才	

035 スイッチを制御 (制御スイッチ)



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、PI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）

ボ	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

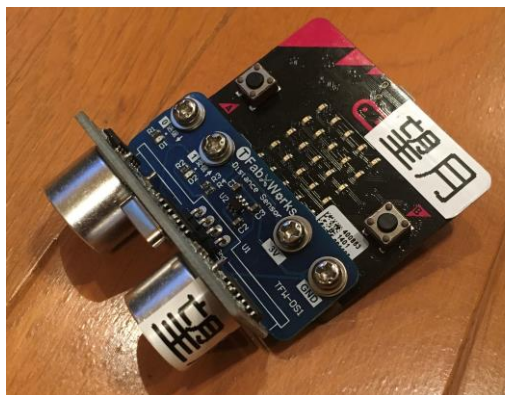
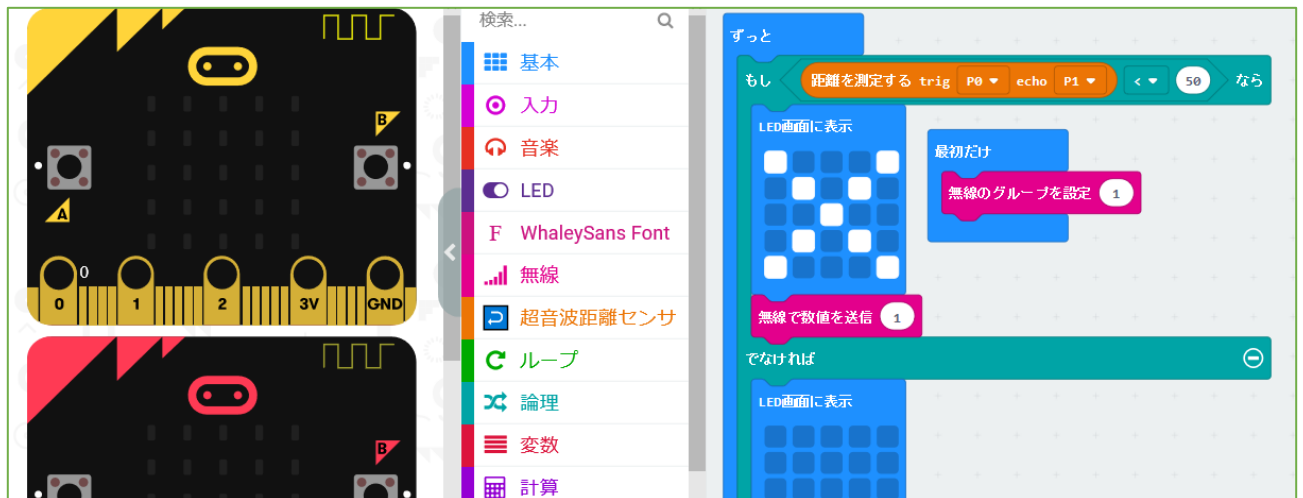
2018/9/27作成

2021/1/ 4修正

037 人感センサー (距離センサー・制御スイッチ) - I

ボ	セ	し	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て近づくたびに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	

2018/10/1作成

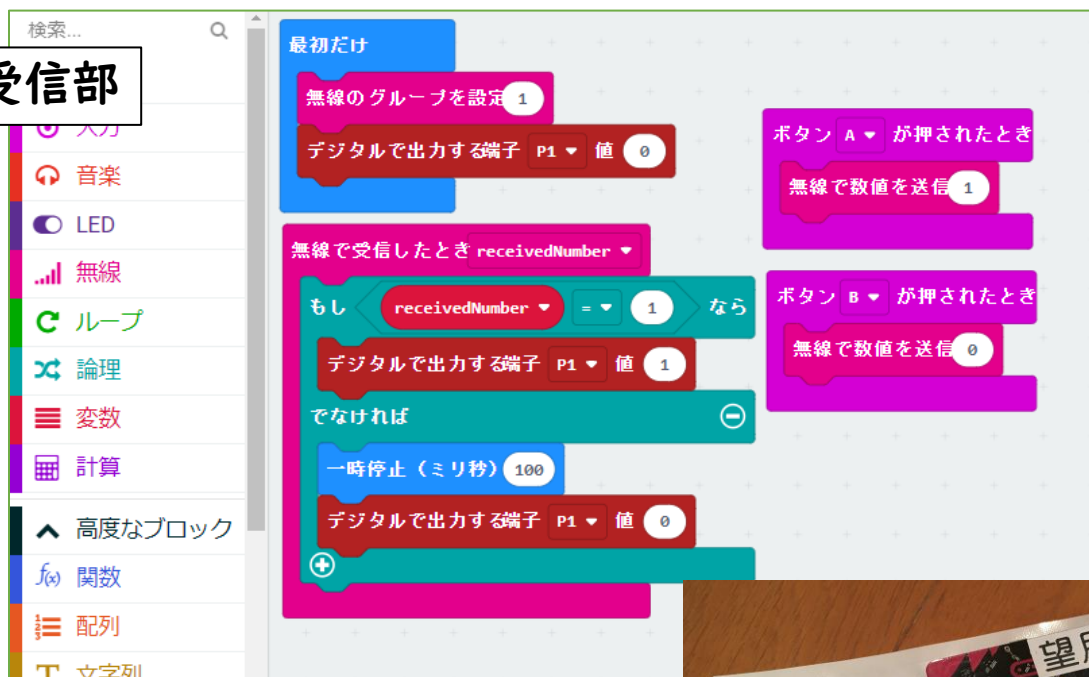
・・・ボタン・センサー・LED・出力
・通信・オプション

(プログラムの続きがあります 043-2)

037 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
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	し	出
通		才	



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-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)

ボ	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	シ	出
通		オ	



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はしてみたものです。

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。

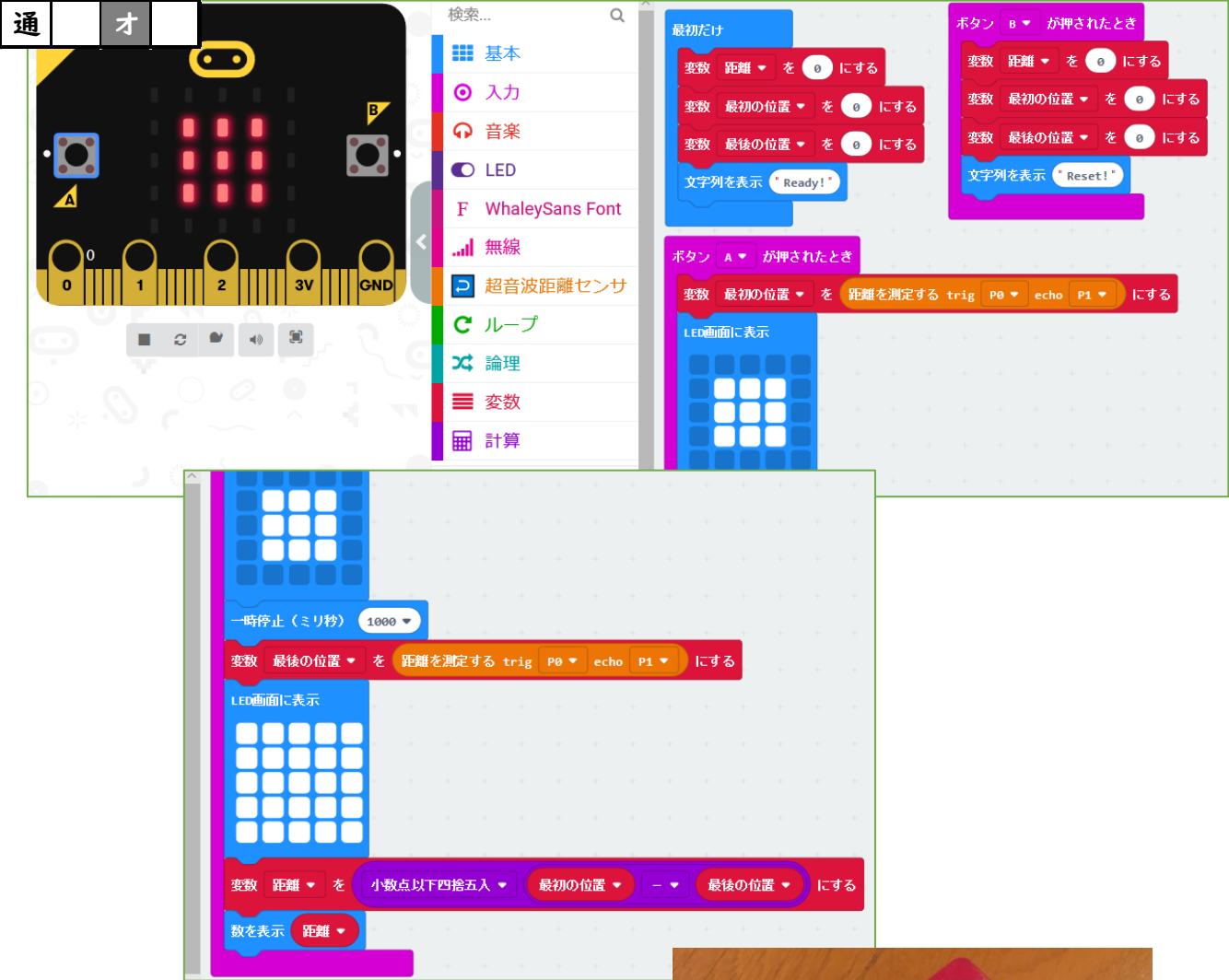
ボ	セ	L	出
通		オ	

・・・センサー・LED

2018/10/3作成
2020/1/10修正

04 | 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】

041 距離を測定(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。

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)

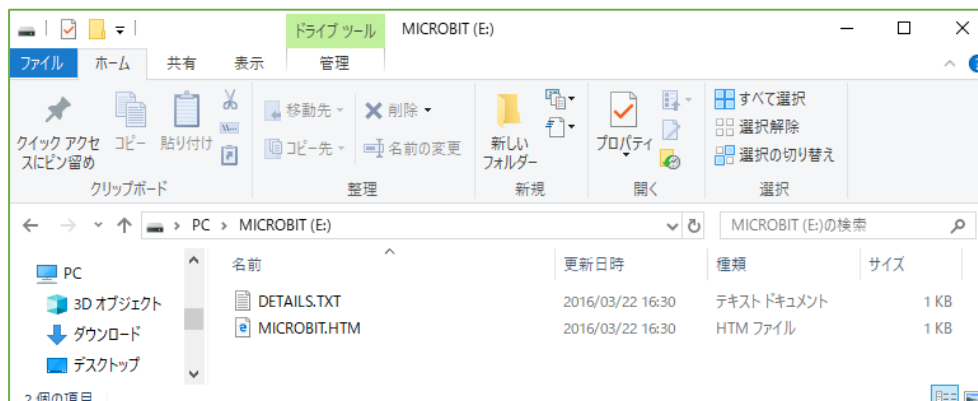
ボ	セ	L	出
通		オ	

・・・センサー・LED・オプション

2018/10/4作成
2020/1/11修正

042 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



【説明】

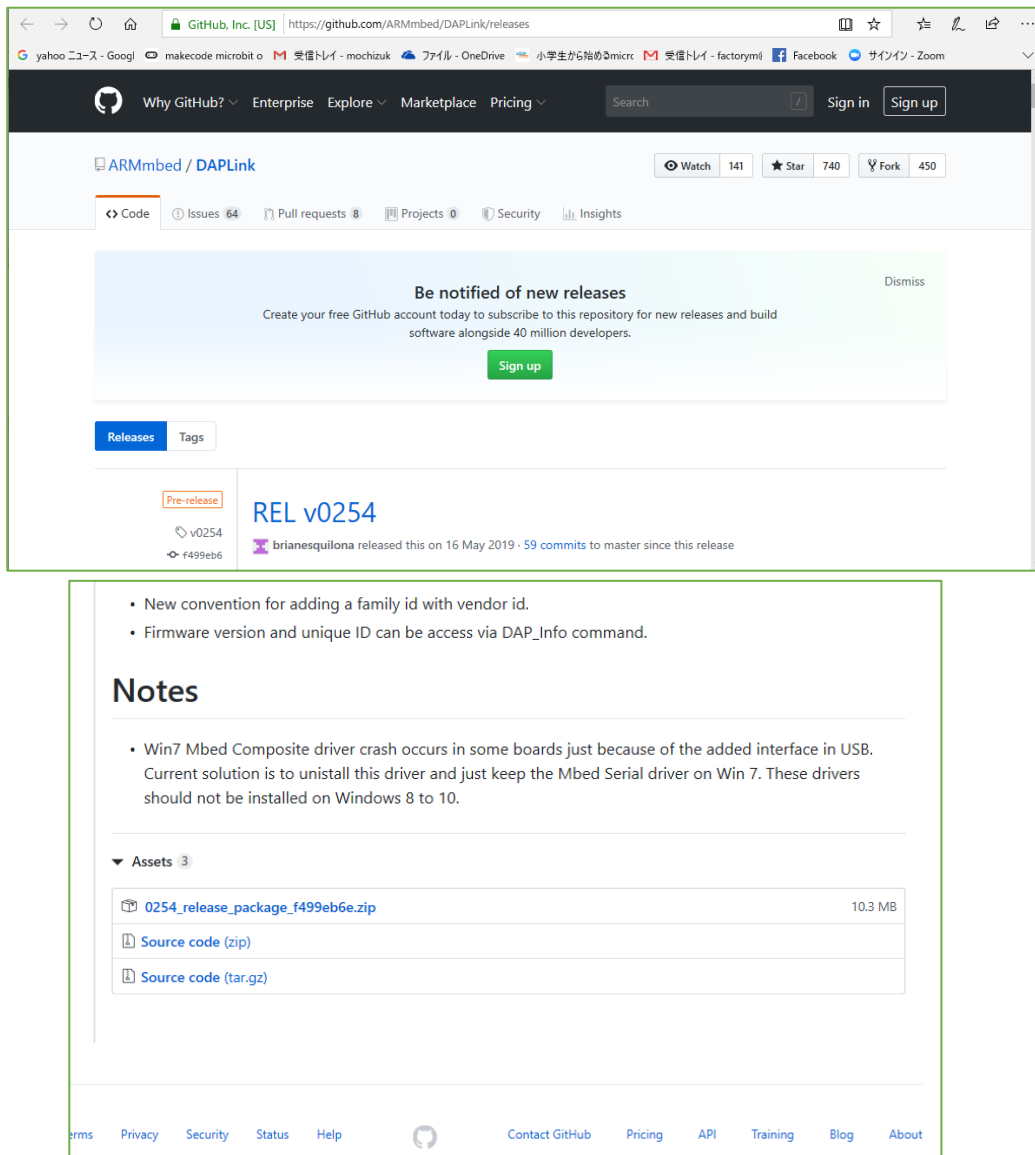
042 micro:bit ファームウェアアップデート

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
と考えると、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

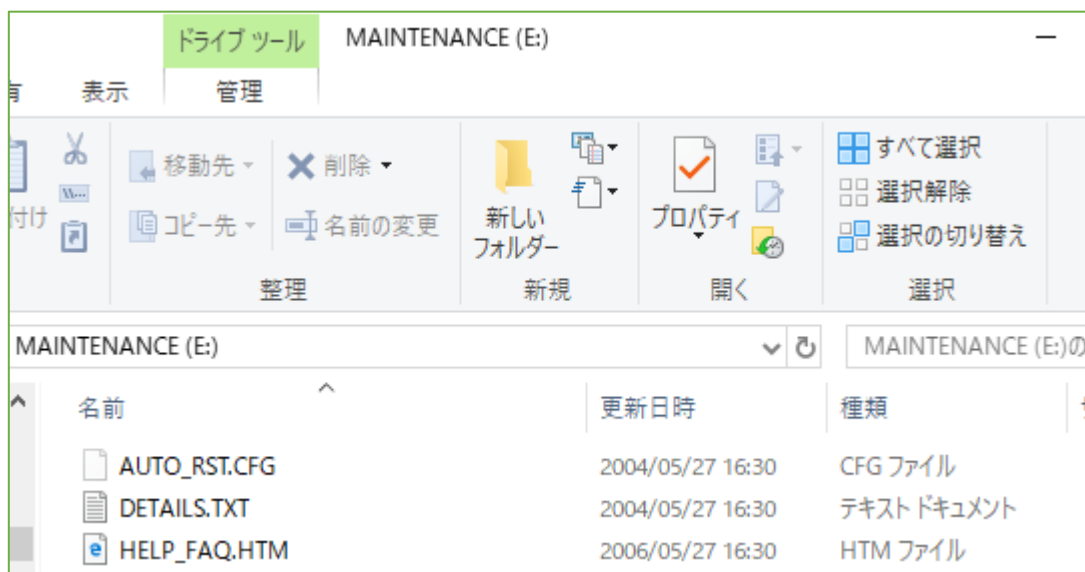
【メモ】

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

	0254_k20dx_rblnano_0x5000.bin	BIN ファイル	107 KB	無
	0254_k20dx_twrke18f_0x8000.bin	BIN ファイル	95 KB	無
	0254_k20dx_twrkl28z72m_0x8000.bin	BIN ファイル	95 KB	無
	0254_k20dx_xdot_l151_0x8000.bin	BIN ファイル	95 KB	無
	0254_k26f_0x0000.bin	BIN ファイル	188 KB	無
	0254_k26f_frdmk32w042_0x8000.bin	BIN ファイル	188 KB	無
	0254_kl26z_0x0000.bin	BIN ファイル	95 KB	無
	0254_kl26z_microbit_0x8000.hex	Microsoft MakeCode bina...	268 KB	無
	0254_kl26z_nina_b1_0x8000.bin	BIN ファイル	95 KB	無
	0254_lpc11u35_0x0000.bin	BIN ファイル	60 KB	無
	0254_lpc11u35_6lowpan_borderrou...	BIN ファイル	60 KB	無
	0254_lpc11u35_6lowpan_borderrou...	BIN ファイル	60 KB	無
	0254_lpc11u35_6lowpan_borderrou...	BIN ファイル	60 KB	無
	0254_lpc11u35_6lowpan_borderrou...	BIN ファイル	60 KB	無

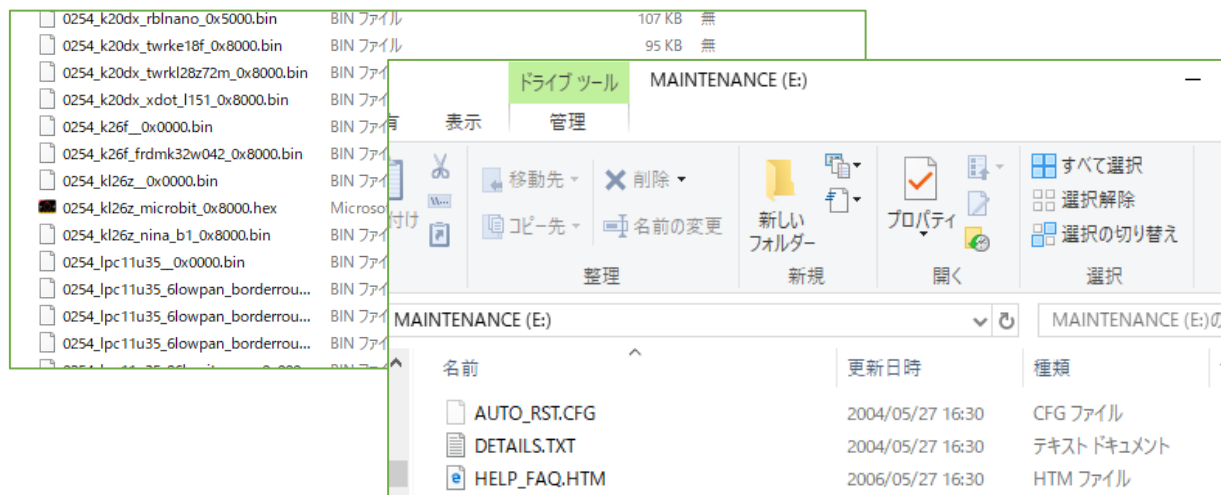
リセットボタンを押しながら基板をUSB接続すると メンテナンスモードが開く



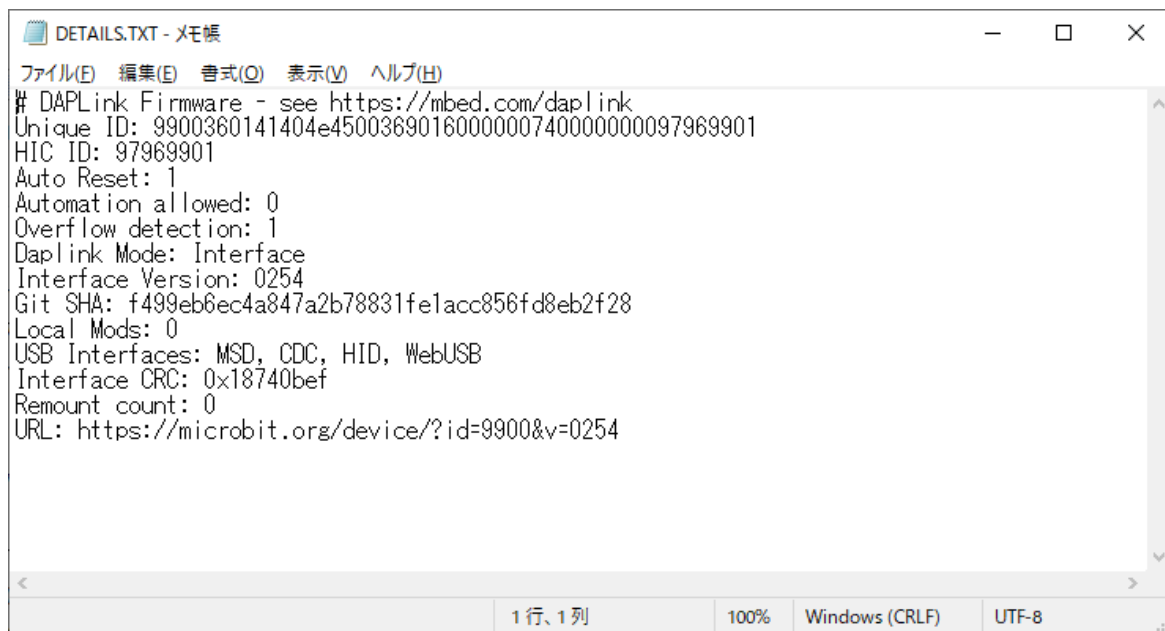
【メモ】

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】

ボ	セ	ㄥ	出
通		才	

043 人感センサー



【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	

2019/1/16作成

・・・ボタン・センサー・LED・オプション

044 ランダム数値表示

ボ	セ	L	出
通		才	



【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	

2019/1/17作成
2020/8/25修正

・・・ボタン・LED

045 じゃんけん

ボ	セ	し	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

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

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

ボ	セ	し	出
通		才	

046 じゃんけん無線判定

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

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

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 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

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

アイコンを表示

でなければ

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

アイコンを表示

相手 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。

・・・ボタン・LED

ボ	セ	し	出
通		オ	

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		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

数を表示 磁力 (μT) X ▼

一時停止 (ミリ秒) 100 ▼

【説明】

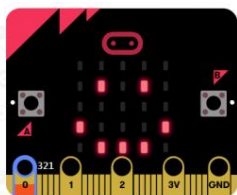
047 磁気表示

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	シ	出
通		オ	



2019/1/23作成

・・・ボタン・センサー・LED

048 磁力計

ボ	セ	し	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 (μT) 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】

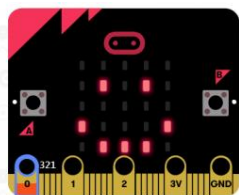
048 磁力計

「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。

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		才	

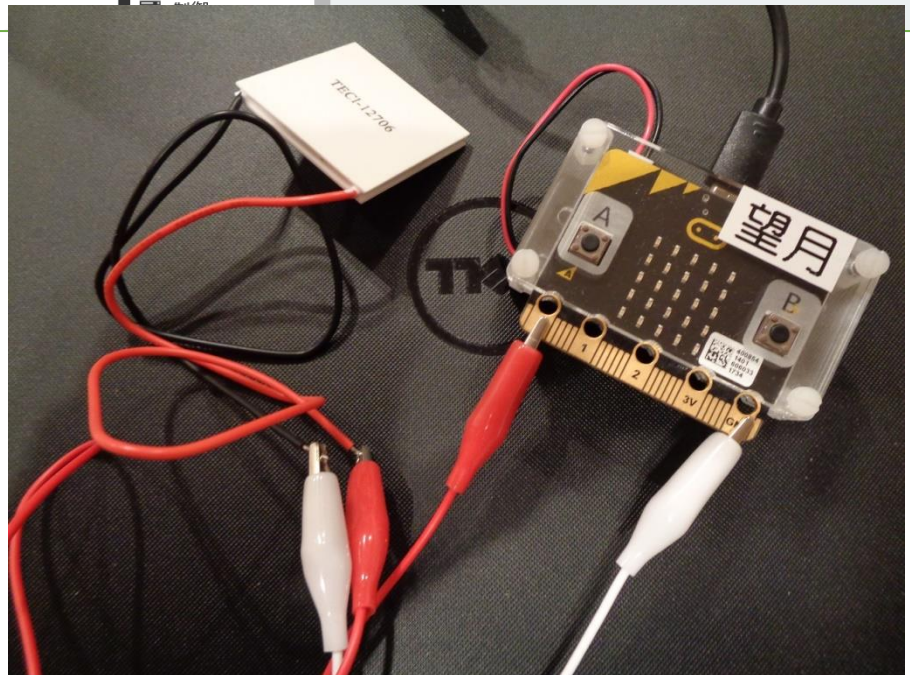


2019/2/19作成

・・・ボタン・センサー・LED

049 検流 (ペルチェ素子)

ボ	セ	し	出
通	入	才	



【説明】

049 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）」を使いました。

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が発生させます。

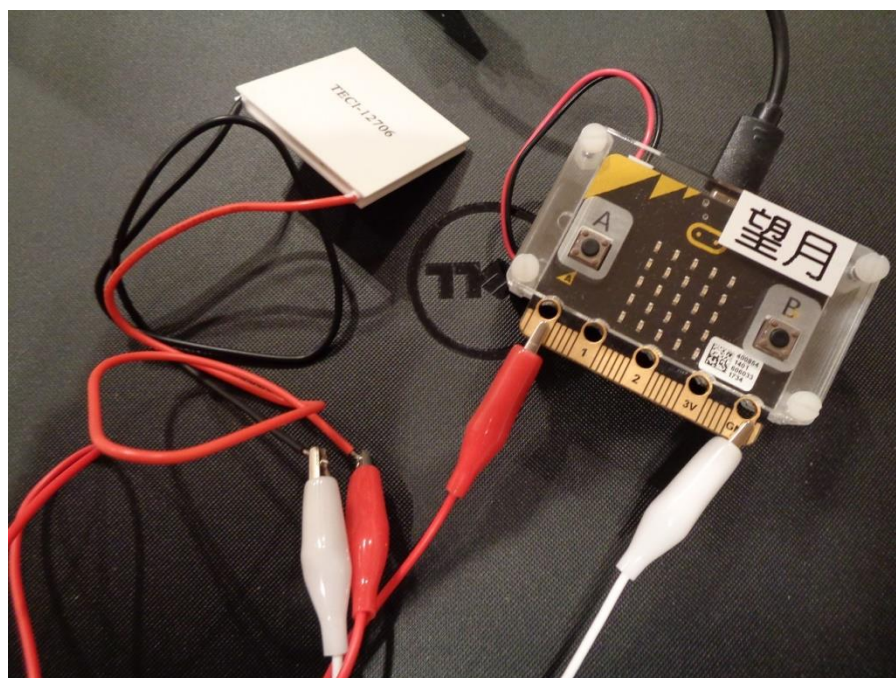
ボ	セ	L	出
通	入	才	

2019/3/2作成

・・・LED・入力

050 検流グラフ(ペルチェ素子)

ボ	セ	L	出
通	入	オ	



【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が発生させます。

ボ	セ	L	出
通	入	オ	

2019/3/4作成
2020/1/12修正

・・・LED・入力

05 | 計算機-1

ボタン A が押されたとき

- もし **計算結果** = 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 **計算結果** = 1 なら
 - もし **四則** = 0 なら
 - LED画面に表示
 - もし **四則** = 1 なら
 - LED画面に表示

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 **計算結果** 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 **計算結果** 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 **計算結果** 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 **計算結果** = 2 なら
 - 数を表示 **～する数**
 - 変数 **～する数** を 1 だけ増やす
- もし **計算結果** = 3 なら
 - LED画面に表示

※計算機-2に続きます。

【説明】

05 | 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。

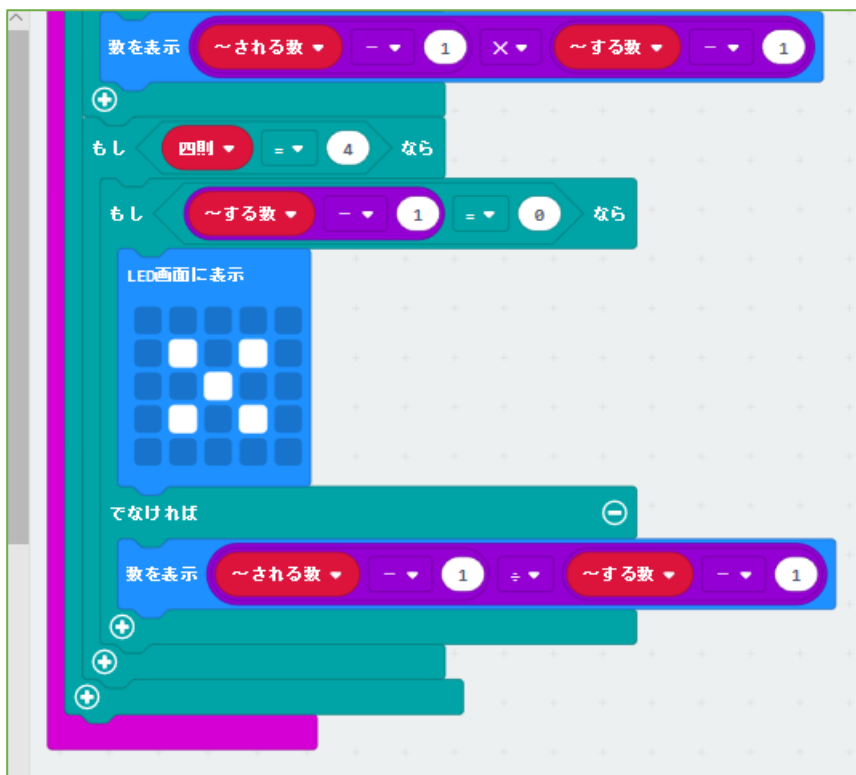
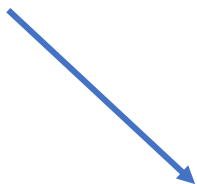
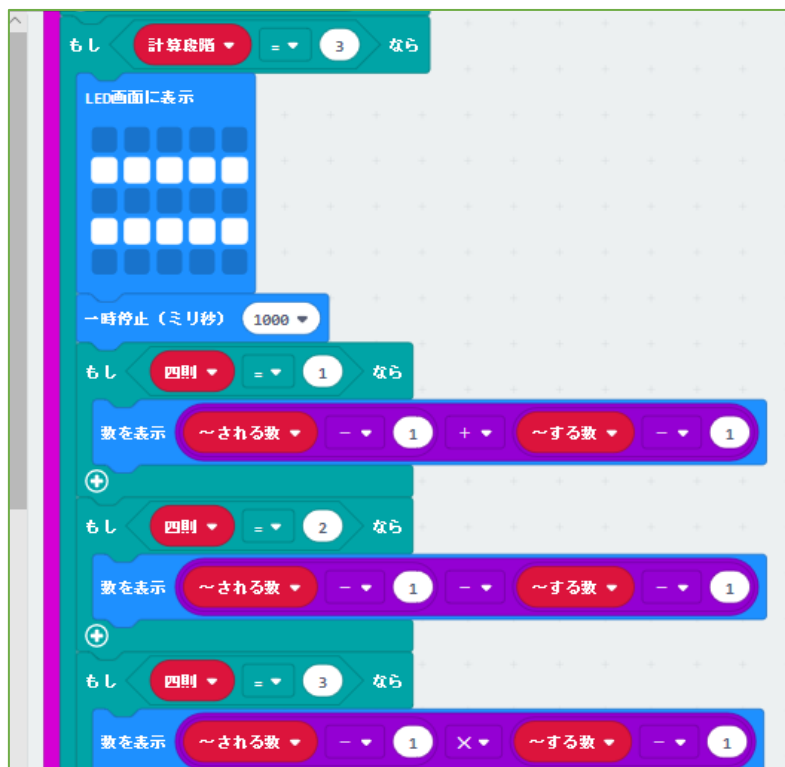
ボ	セ	L	出
通	入	オ	

2019/3/4作成

・・・ボタン・LED

※05 | 計算機-2 に続きます。

051 計算機-2



【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（+-×÷）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	オ	

・・・ボタン・LED

2019/3/4作成

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 and several function blocks for a calculator. The dialog is titled 'Edit Function' and has tabs for 'パラメーターを追加する' (Add Parameter), '文字列' (Text), '真偽値' (Boolean), 'Number', and 'LedSprite'. The 'Number' tab is selected, and a '関数 たし算' (Function Addition) block is shown. Below the dialog, four function blocks are displayed: '関数 たし算' (Function Addition), '関数 ひき算' (Function Subtraction), '関数 かけ算' (Function Multiplication), and '関数 わり算' (Function Division). Each block has a '数を表示' (Show Number) block and two input fields for numbers. The 'たし算' block has a '+' operator, 'ひき算' has a '-' operator, 'かけ算' has a '×' operator, and 'わり算' has a '÷' operator. To the right, a sequence of blocks is shown, including 'もし' (If) blocks with conditions like '四則' (Arithmetic) and 'なら' (Then), followed by '呼び出し' (Call) blocks for the functions, and a 'LED画面に表示' (Display on LED Screen) block.

【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。

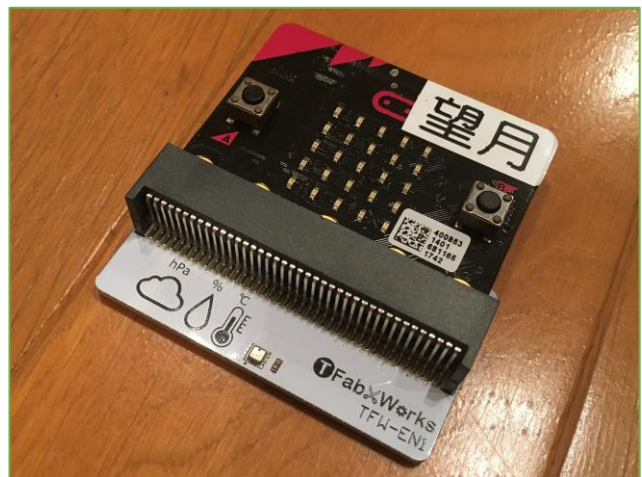
ボ	セ	L	出
通	入	才	

・・・ボタン・LED

2019/3/4作成

053 気温を小数表示

ボ	セ	L	出
通		オ	



【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。

(°C) で表示しますが、小数表示のほかに整数表示もできます。

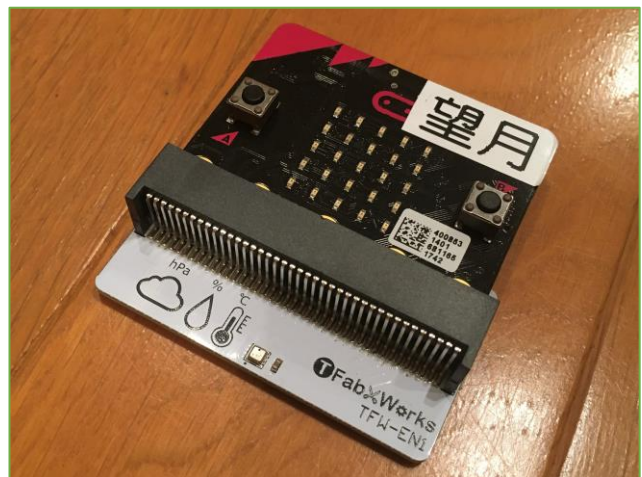
ボ	セ	L	出
通		オ	

・・・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

ボ	セ	L	出
通		才	



【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。

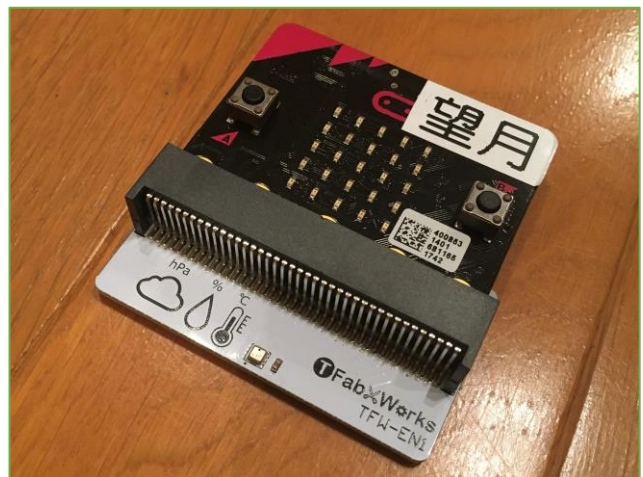
ボ	セ	L	出
通		オ	

・・・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		オ	



【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	シ	出
通		オ	

・・・ボタン・センサー・LED・オプション

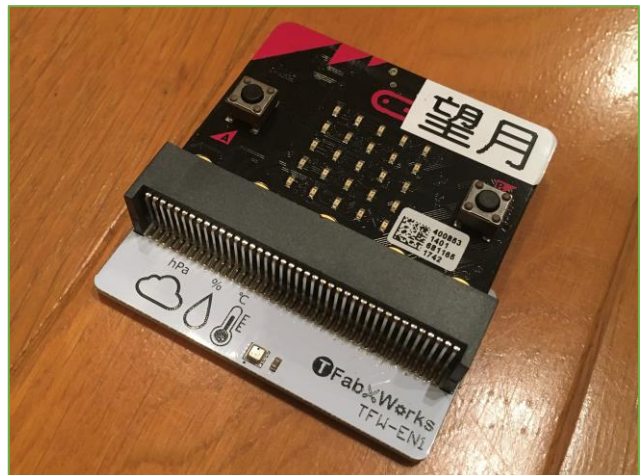
2021/1/1作成

056 気温・湿度・気圧を小数表示

ボ	セ	L	出
通		オ	



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。

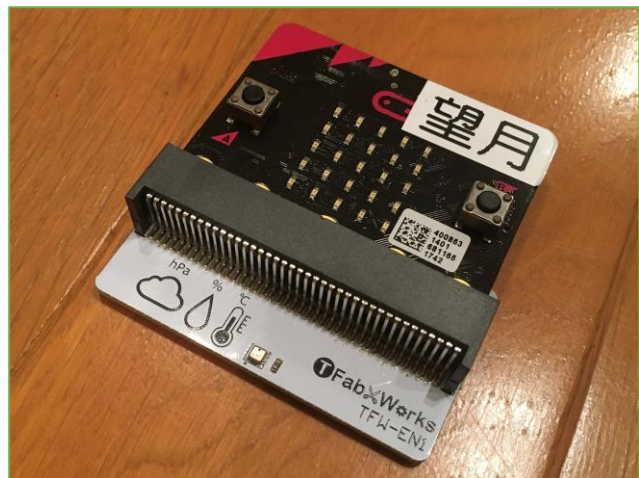
ボ	セ	ン	サ
ー		出	

・・・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	し	出
通		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す）、表のように条件を考えました。

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

<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。

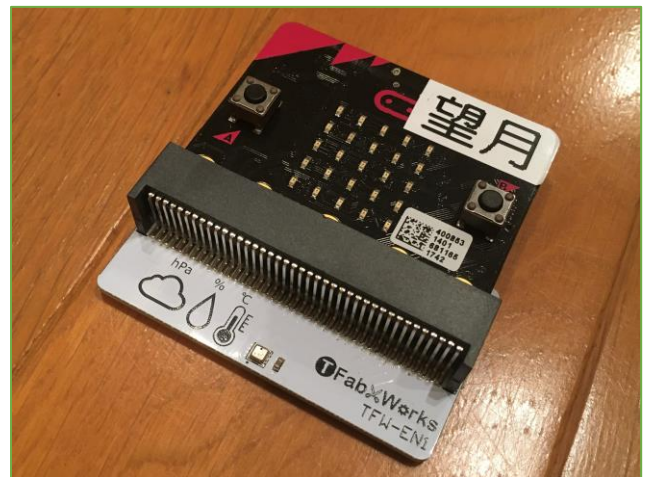
ポ	セ	L	出
通		オ	

・・・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。

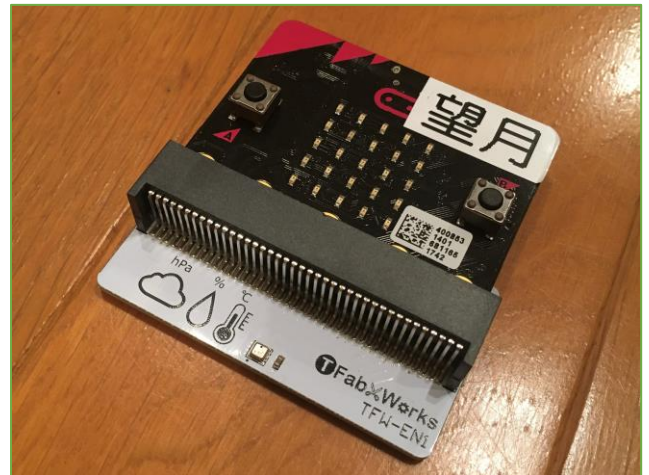
ボ	セ	L	出
通	入	オ	

・・・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。

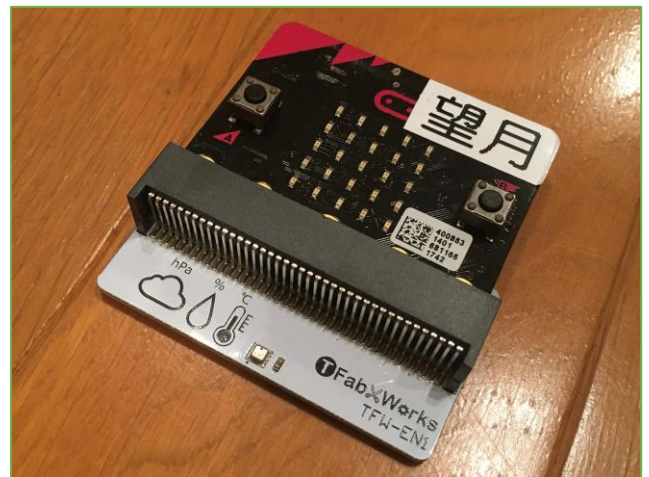
ボ	セ	L	出
通	入	オ	

・・・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	才	

The image shows a Scratch script for receiving temperature and humidity data via radio. The script is organized into three main sections: initialization, reception of temperature data, and reception of humidity data.

Initialization (最初だけ):

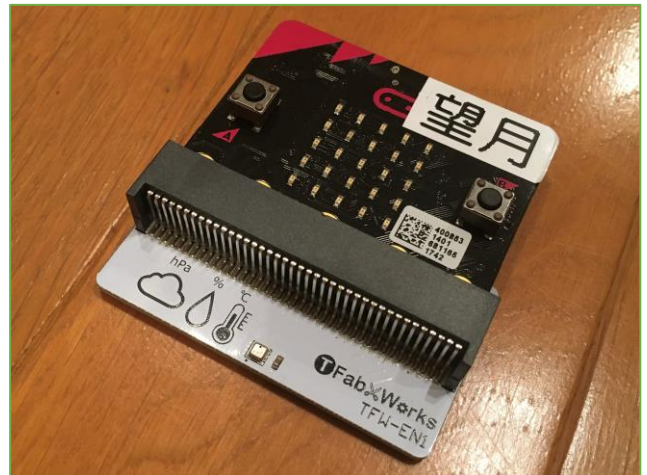
- 無線のグループを設定 (1) (Set radio group to 1)
- 変数 temp3 を 0 にする (Set variable temp3 to 0)
- 変数 humi3 を 0 にする (Set variable humi3 to 0)

Reception of Temperature Data (無線で受信したとき):

- 変数 temp3 を 0 にする (Set variable temp3 to 0)
- 変数 humi3 を 0 にする (Set variable humi3 to 0)
- もし name = "temp2" なら (If name is "temp2")
 - 変数 temp3 を value にする (Set variable temp3 to value)
- もし name = "humi2" なら (If name is "humi2")
 - 変数 humi3 を value にする (Set variable humi3 to value)
- もし temp3 ≠ 0 なら (If temp3 is not equal to 0)
 - 数を表示 temp3 (Show number temp3)
 - 文字列を表示 "C/" (Show text "C/")

Reception of Humidity Data (無線で受信したとき):

- 変数 temp3 を value にする (Set variable temp3 to value)
- 変数 humi3 を value にする (Set variable humi3 to value)
- もし temp3 ≠ 0 なら (If temp3 is not equal to 0)
 - 数を表示 temp3 (Show number temp3)
 - 文字列を表示 "C/" (Show text "C/")
- もし humi3 ≠ 0 なら (If humi3 is not equal to 0)
 - 数を表示 humi3 (Show number humi3)
 - 文字列を表示 "%/" (Show text "%/")



※「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。

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」 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

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

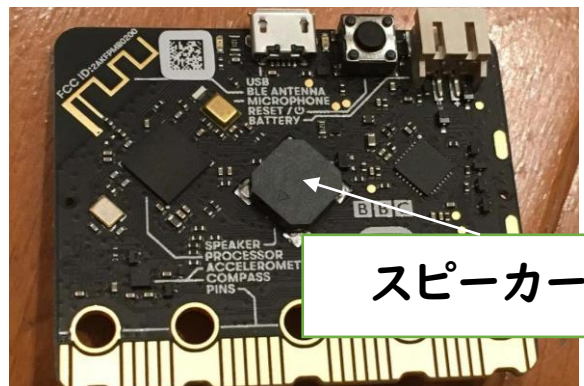
- 001 重ねて音を鳴らす
- 002 音がしたらビックリ
- 003 タッチしたらビックリ
- 004 音のしきい値
- 005 静かになったら
- 006 音量グラフ
- 007 音量を送信
- 008 音量を受信
- 009 タッチする楽器
- 010 メロディを打楽器に

001 重ねて音を出す

ボ	セ	L	出
通		オ	v2



micro:bit v2
のみで動作



スピーカー

【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		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		オ	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		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		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		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		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		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		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		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		オ	v2

006 音量グラフ



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方にあります)

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。

ボ	セ	L	出
通		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		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方にあります)

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		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		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。

ボ	セ	L	出
通		オ	v2

・・・使用するのは、LED・通信

※007 音量を送信 とセットです。

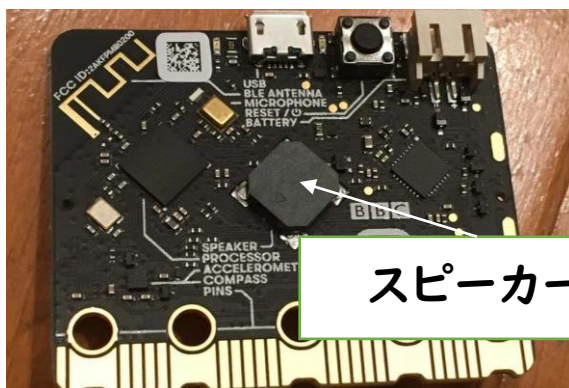
2020/12/18作成

ボ	セ	L	出
通		オ	v2

009 タッチする楽器



micro:bit v2
のみで動作



【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		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

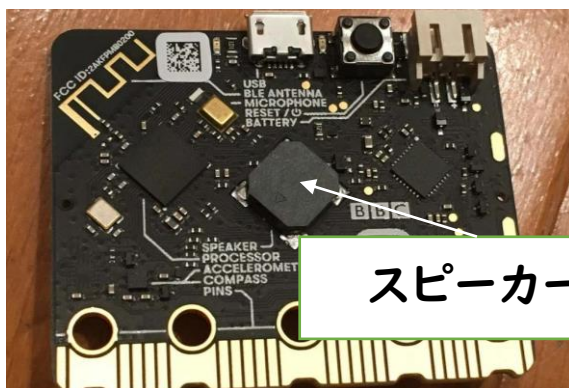
ボ	セ	L	出
通		オ	v2

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



【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		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」 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

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「○」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/>

【説明】

micro:bit プログラミングサイト

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

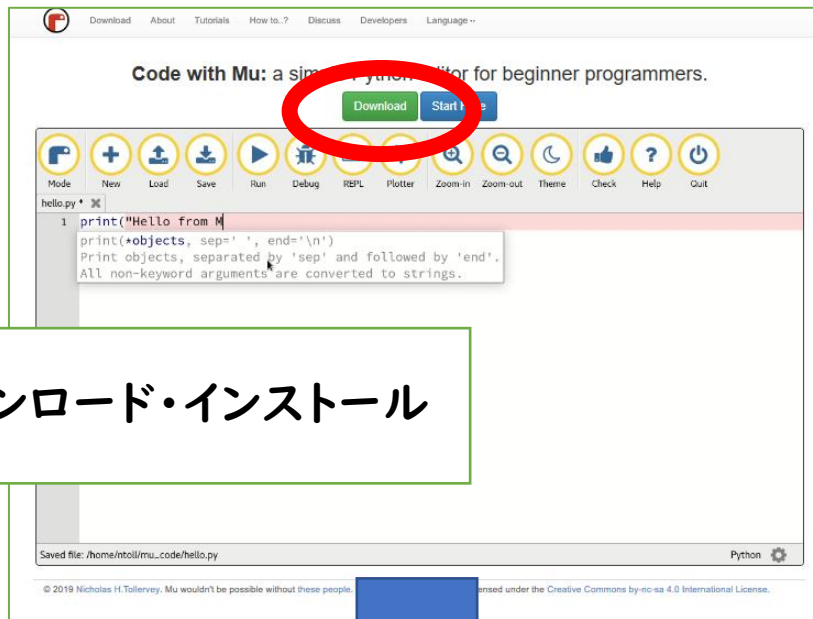
のPythonエディターでオンラインプログラミングします。

- ・Download→プログラムをダウンロードして、micro:bitへ送る

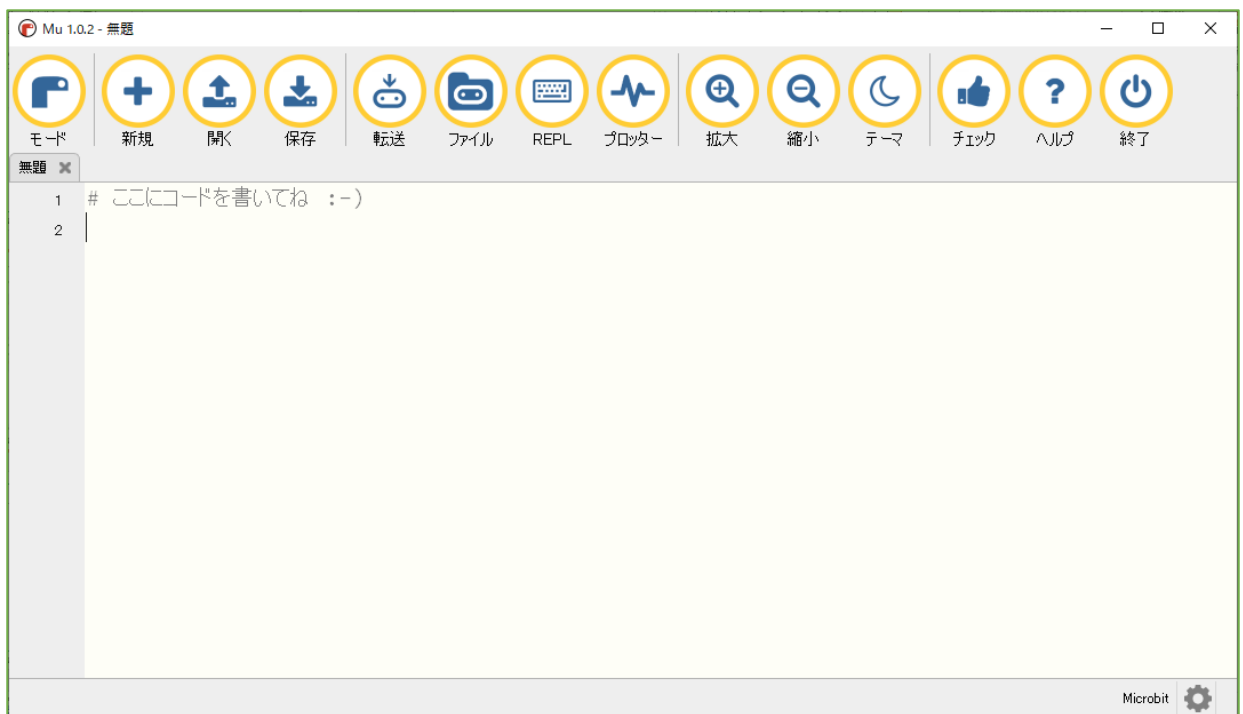
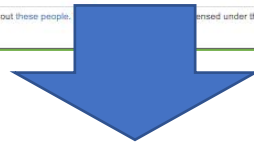
- ・Load/Save→*.py (Mu編集用ファイル)やプログラムを読み込み・保存できる

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。

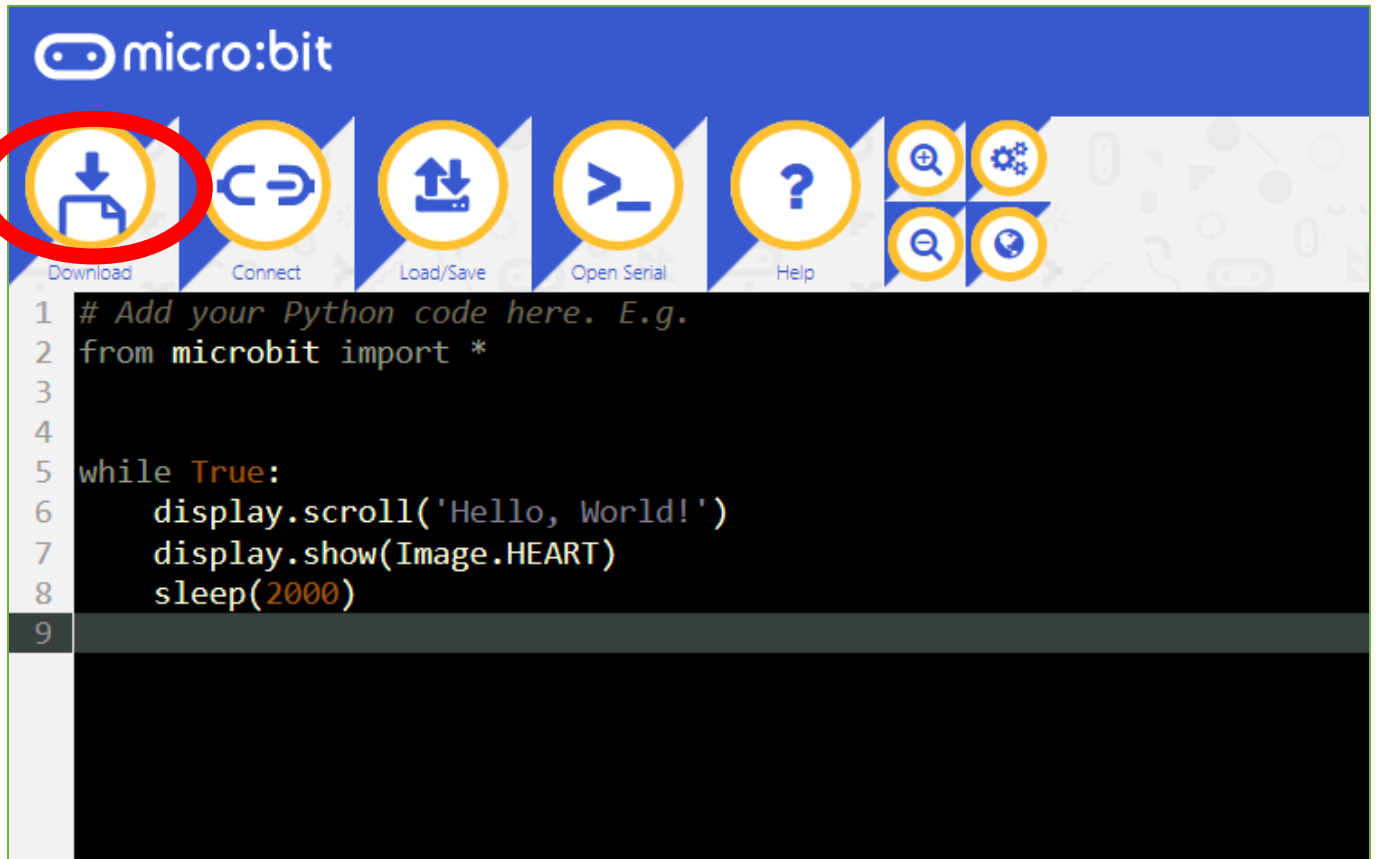
※オンラインエディターとの違い

- ・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>



…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
〇時(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)

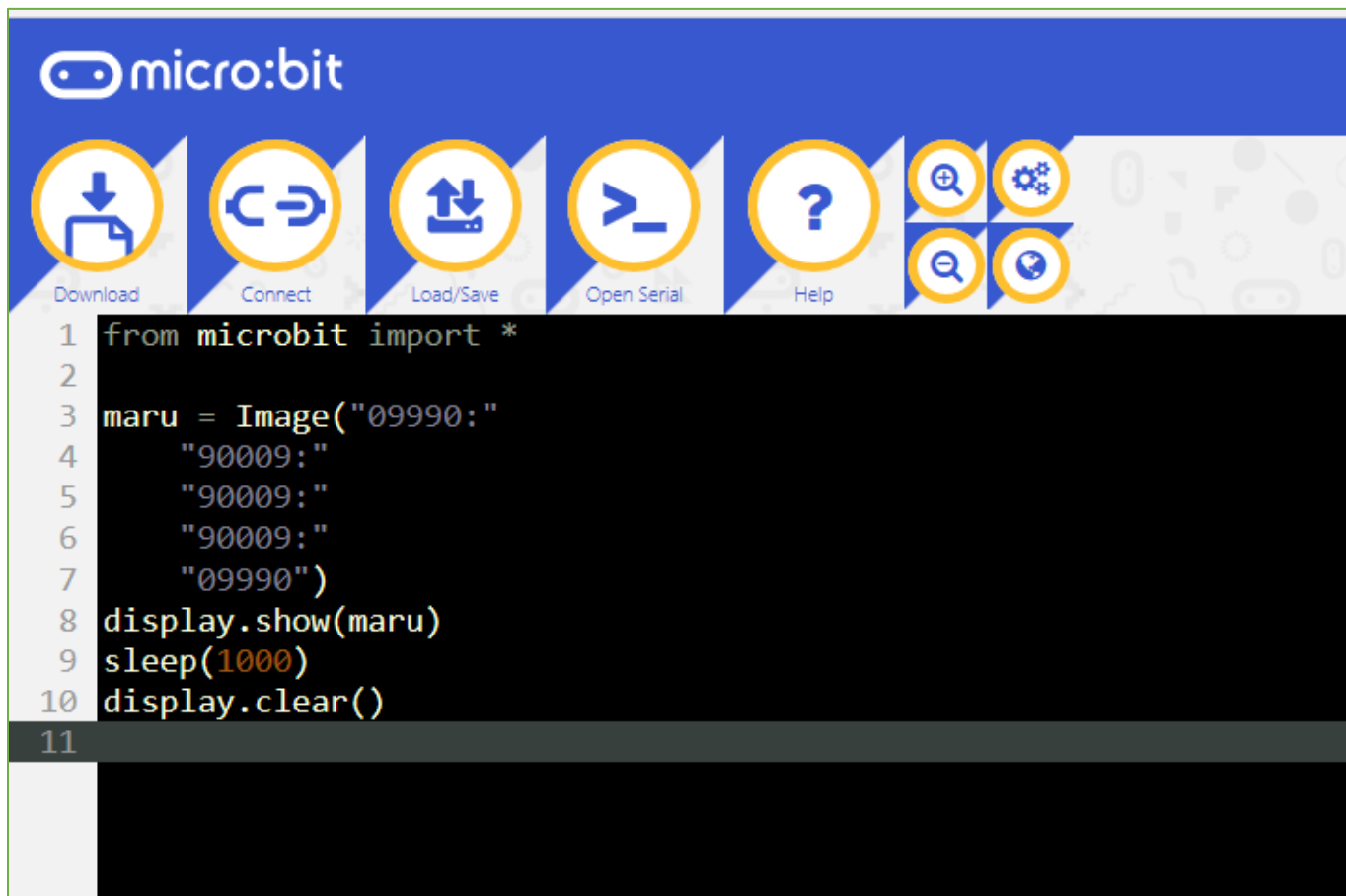
Image.CHESSBOARD チェス盤
Image.DIAMOND ダイヤ
Image.DIAMOND_SMALL 小ダイヤ
Image.SQUARE 四角
Image.SQUARE_SMALL 小四角
Image.RABBIT うさぎ
Image.COW 牛
Image.MUSIC_CROTCHET 四分音符
Image.MUSIC_QUAVER 八分音符
Image.MUSIC_QUAVERS つないだ八分音符
Image.PITCHFORK フォーク
Image.XMAS クリスマス
Image.PACMAN パックマン
Image.TARGET ターゲット
Image.TSHIRT 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>



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「○」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。

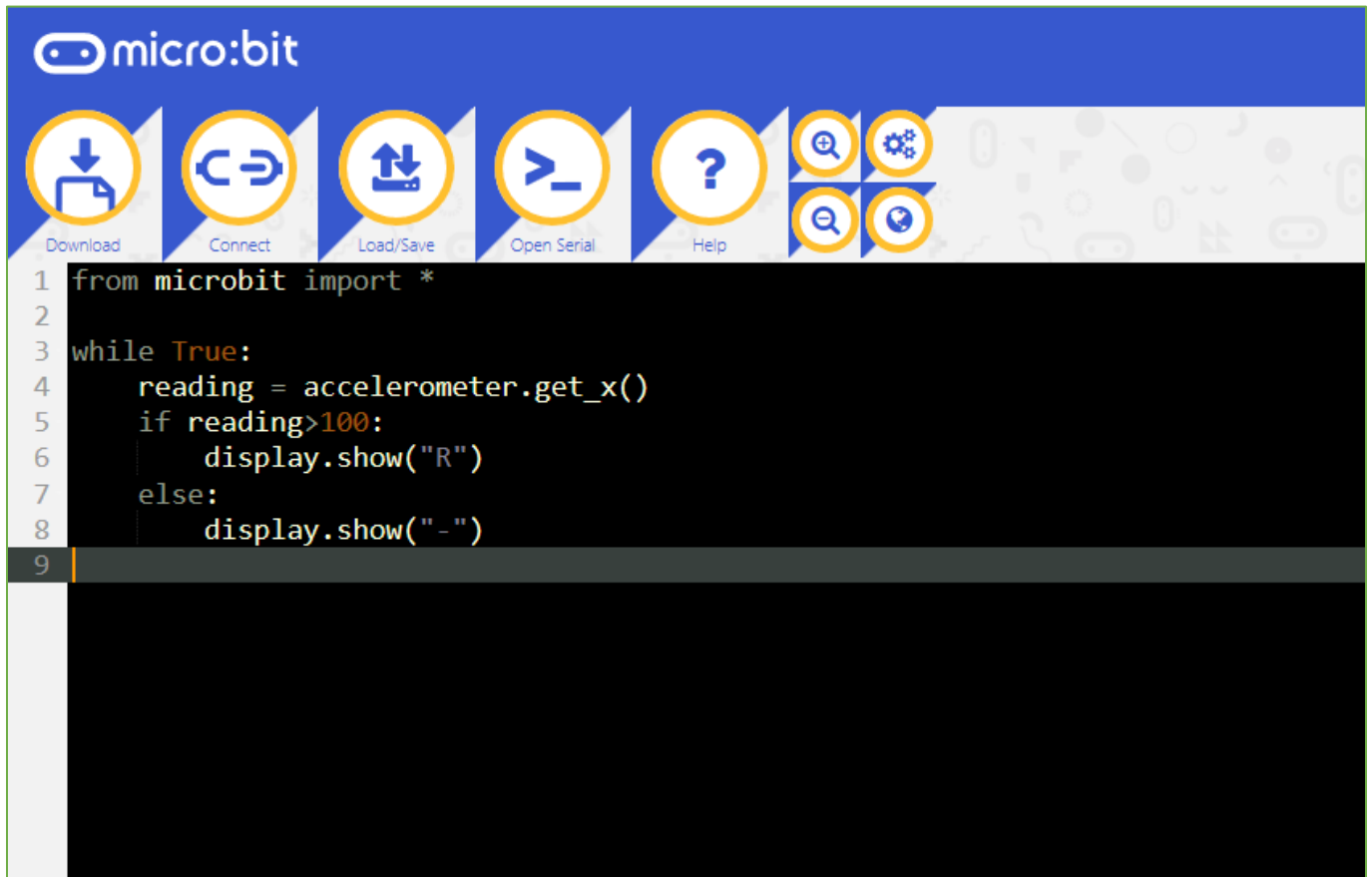
ボ	セ	L	出
通	入	才	

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>



`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
```

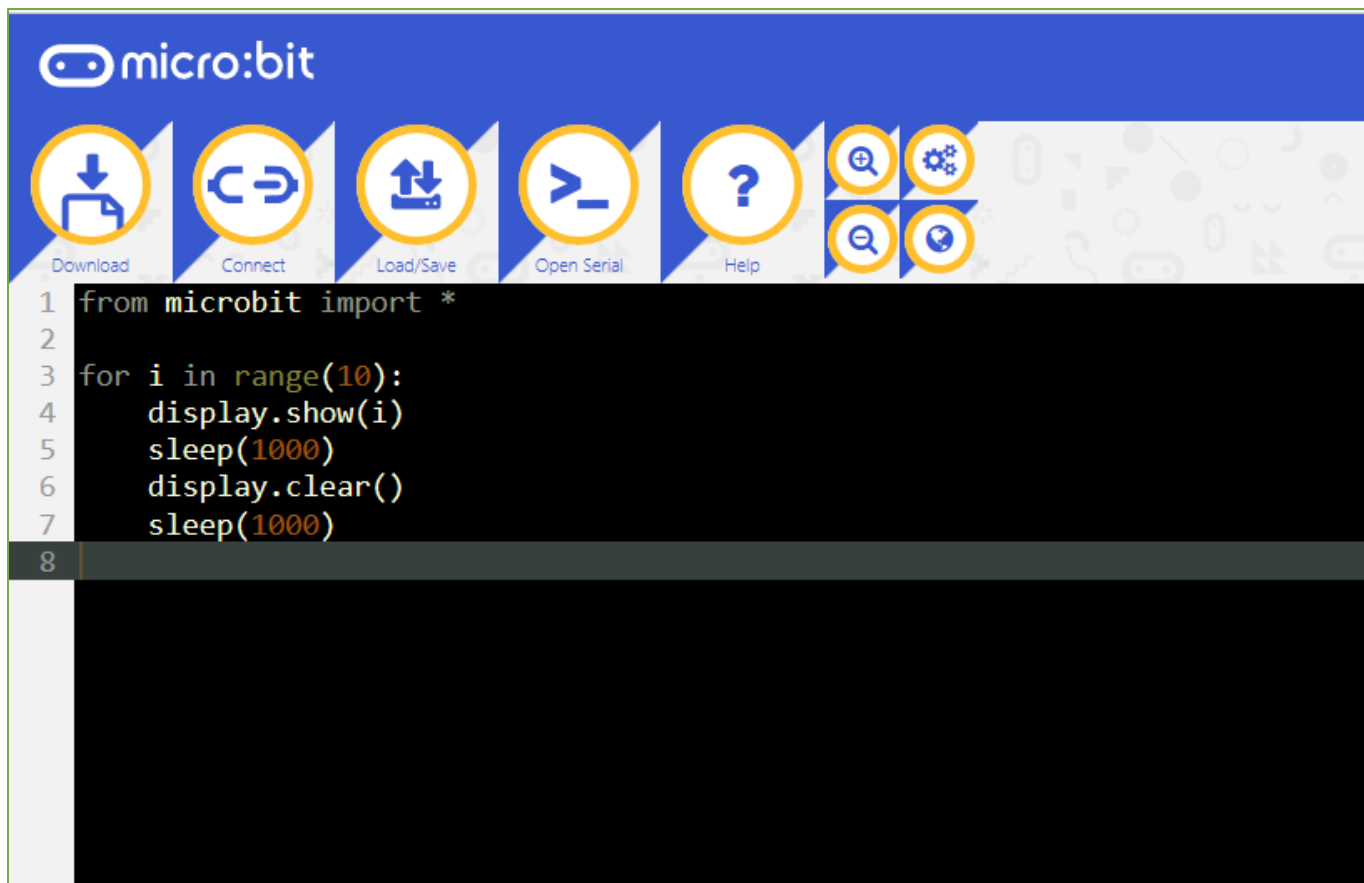
2020/1/13作成

ボ	セ	L	出
通	入	才	

004 0から9までを表示する

ボ	セ	L	出
通	入	才	

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



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」 Pythonプログラミング編

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

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

関数など

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

条件 (if文)

006 暗くなるとLEDが点く

音

007 Happy Birthdayを鳴らす

複数の条件

017 コンパス (Compass)

無線

018 2台で送受信

傾き

019 水準器

020 傾きメーター

乱数と繰り返し

052 ランダム数値表示

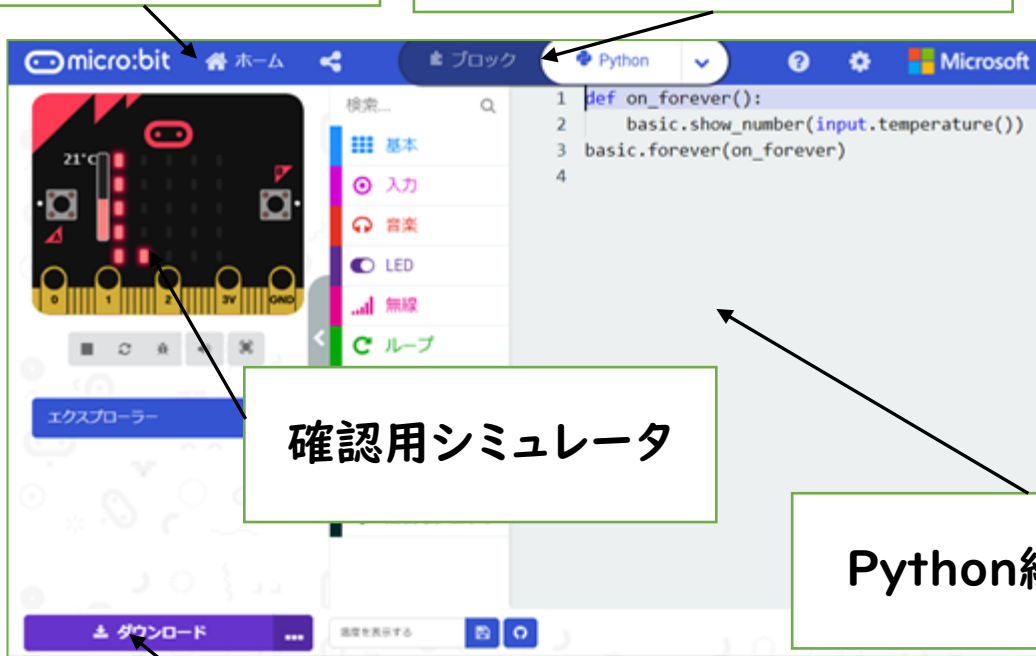
053 じゃんけん

micro:bit プログラミングサイト

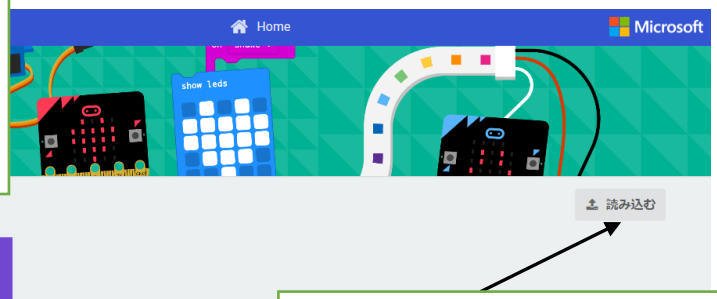
<https://makecode.microbit.org/#editor>

ホーム画面へ

ブロック編集/Python画面
の切り替え



作成したプログラムを
USB接続した
micro:bit にコピー



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。

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。

- ・Python画面でプログラミング。
 - ・確認用シミュレータで確認する。
 - ・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。
- (・micro:bitを接続せずにプログラムをダウンロード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 温度を表示する

【コード】

```
def on_forever():  
    basic.show_number(input.temperature())  
basic.forever(on_forever)
```

【解説】

- ・・・def on_forever(): ・・・関数「ずっと実行」
 実行内容
- ・・・basic.show_number ・・・数値を表示
- ・・・input.temperature() ・・・温度の値
 - ・明るさの値 input.light_level()
 - ・磁力の値 input.magnetic_force(Dimension.X)
- ・・・basic.forever(on_forever) ・・・関数on_forever を呼び出し

002 明るさを表示する

【コード】

```
def on_forever():  
    led.plot_bar_graph(input.light_level(), 255)  
    basic.pause(1000)  
basic.forever(on_forever)
```

【解説】

- ・・・led.plot_bar_graph ・・・LEDにグラフを表示 255は最大値(変更可)
- ・・・basic.pause() ・・・一時停止 1000は停止時間(ミリ秒)

003 ボタンで温度を表示する

【コード】

```
def on_button_pressed_b():  
    basic.show_number(input.temperature())  
input.on_button_pressed(Button.B, on_button_pressed_b)
```

【解説】

・・・def on_button_pressed_b(): ・・・関数「Bボタンを押すと実行」
・・・input.on_button_pressed(Button.B, on_button_pressed_b)
・・・呼び出し
・・・A+Bボタンは、Button.AB on_button_pressed_ab

004 ボタンで明るさを表示する

【コード】

```
def on_button_pressed_a():  
    led.plot_bar_graph(input.light_level(), 255)  
    basic.pause(1000)  
    basic.clear_screen()  
input.on_button_pressed(Button.A, on_button_pressed_a)
```

【解説】

・・・basic.clear_screen() ・・・画面を消去

006 暗くなるとLEDが点く

【コード】

```
def on_forever():
    if input.light_level() < 15:
        basic.show_leds("""
            # # # # #
            # # # # #
            # # # # #
            # # # # #
            # # # # #
            """)
    elif input.light_level() < 30:
        basic.show_leds("""
            . . . . .
            . # # # .
            . # # # .
            . # # # .
            . . . . .
            """)
    else:
        basic.show_leds("""
            . . . . .
            . . . . .
            . . . . .
            . . . . .
            . . . . .
            """)
basic.forever(on_forever)
```

【解説】

```
...if input.light_level() < 15:
    ...if 文 もし~ならば、
    elif input.light_level() < 30:
        ...追加条件
    else:
        ...さもなくば
...basic.show_leds("""
    . . . . .
    . # # # .
    . # # # .
    . # # # .
    . . . . .
    """)
...LEDのオンオフ
... 消灯 # 点灯
```

007 Happy Birthdayを鳴らす

【コード】

```
def on_button_pressed_a():
    music.play_tone(392,750)
    music.play_tone(392,music.beat(BeatFraction.QUARTER))
    music.play_tone(440,music.beat(BeatFraction.WHOLE))
    music.play_tone(392,music.beat(BeatFraction.WHOLE))
    music.play_tone(523,music.beat(BeatFraction.WHOLE))
    music.play_tone(494,music.beat(BeatFraction.DOUBLE))
    music.play_tone(392,750)
    music.play_tone(392,music.beat(BeatFraction.QUARTER))
    music.play_tone(440,music.beat(BeatFraction.WHOLE))
    music.play_tone(392,music.beat(BeatFraction.WHOLE))
    music.play_tone(587,music.beat(BeatFraction.WHOLE))
    music.play_tone(523,music.beat(BeatFraction.DOUBLE))
input.on_button_pressed(Button.A, on_button_pressed_a)
```

【解説】

- ・・・music.play_tone(440,music.beat(BeatFraction.WHOLE))
 - ・・・440 音の高さ(ラ)
392 ソ 440 ラ 494 シ 523 上のド 587 上のレ など
 - ・・・WHOLE 音の長さ一拍(4分音符)
DOUBLE 二拍(2分音符)
QUARTER 1/4拍(16分音符)
HALF 1/2拍(8分音符)
750 数値(付点4分音符)

013 コンパス (Compass)

【コード】

```
def on_forever():
    if 315 < input.compass_heading() or input.compass_heading() < 45:
        basic.show_string("N")
    elif 45 < input.compass_heading() and input.compass_heading() < 135:
        basic.show_string("E")
    elif 135 < input.compass_heading() and input.compass_heading() < 225:
        basic.show_string("S")
    elif 225 < input.compass_heading() and input.compass_heading() < 315:
        basic.show_string("W")
    else:
        basic.clear_screen()
basic.forever(on_forever)
```

【解説】

・・・or・・・または
・・・and・・・かつ
・・・basic.show_string()・・・文字列の表示

014 2台で送受信

【コード】

```
def on_received_number_deprecated(receivedNumber):  
    if receivedNumber == 0:  
        basic.show_string("Receive!")  
        basic.pause(1000)  
        basic.clear_screen()  
radio.on_received_number_deprecated(on_received_number_deprecated)  
  
def on_button_pressed_a():  
    radio.send_number(0)  
input.on_button_pressed(Button.A, on_button_pressed_a)  
  
radio.set_group(1)
```

【解説】

- ・・・radio.set_group() ・・・無線グループの設定
- ・・・radio.send_number() ・・・数値を送信
- ・・・radio.on_received_number_deprecated
 (on_received_number_deprecated) ・・・数値を受信したら
- ・・・on_received_number_deprecated(receivedNumber):
 ・・・受信した数値の処理関数

015 水準器

【コード】

```
x = 2
y = 2
def on_forever():
    global x, y
    led.unplot(x, y)
    x = (-1 - input.acceleration(Dimension.X) + 1000) / 400
    y = (-1 - input.acceleration(Dimension.Y) + 1000) / 400
    led.plot(x, y)
basic.forever(on_forever)
```

【解説】

- …global グローバル変数の利用
- …led.unplot(x, y)
 - …LED 位置x,yの消灯 x 0~4 y 0~4
- …led.plot(x, y)
 - …LED 位置x,yの点灯 x 0~4 y 0~4
- …x = (-1 - input.acceleration(Dimension.X) + 1000) / 400
 - …X方向の位置変化 → x の座標 (0~4) に

016 傾きメーター

【コード】

```
y = 0
x = 2
def on_forever():
    global y, x
    y = 0
    led.unplot(x, y)
    x = (-1 - input.acceleration(Dimension.X) + 1000) / 400
    for index in range(5):
        led.plot(x, y)
        y += 1
    basic.pause(100)
    basic.clear_screen()
basic.forever(on_forever)
```

【解説】

- …for index in range(5):
 - …ループ (繰り返し)
 - …range(5) …0~4 (5回)
 - …y += 1 …yに1を足す(1増える)

044 ランダム数値表示

【コード】

```
def on_button_pressed_a():
    basic.show_number(randint(0, 9))
input.on_button_pressed(Button.A, on_button_pressed_a)
```

【解説】

・・・randint(0, 9) ・・・0~9の間で乱数(ランダムに)

045 じゃんけん

【コード】

乱数 = 0

```
def on_button_pressed_a():
```

```
    global 乱数
```

```
    乱数 = randint(0, 2)
```

```
    basic.show_number(乱数)
```

```
    if 乱数 == 0:
```

```
        basic.show_leds("""
```

```
            # # # # .
```

```
            # # # # .
```

```
            # # # # #
```

```
            # # # # .
```

```
            . # # # .
```

```
            """)
```

```
    elif 乱数 == 1:
```

```
        basic.show_leds("""
```

```
            . # . # .
```

```
            . # # . .
```

```
            # # # # .
```

```
            # # # # .
```

```
            . # # # .
```

```
            """)
```

```
    else:
```

```
        basic.show_leds("""
```

```
            . . . . .
```

```
            # # # # .
```

```
            # # # # #
```

```
            # # # # .
```

```
            . # # # .
```

```
            """)
```

```
input.on_button_pressed(Button.A, on_button_pressed_a)
```

【解説】

・・・乱数 = 0 ・・・変数に日本語が使える。

・・・if 乱数 == 0: ・・・== 等しい (= は右から左への代入)

コードを打つ時に気をつけること

- 縦の列をそろえます。

```
def OO():
```

```
    basic....
```

↑

このタブは、「半角スペース4つ」。Tabキーは使いません。

- 「,」の次は、見やすくするため半角スペースを入れます。

- 「=」「+」などの両側には、半角スペースを入れます。

- 問題があるときは、画面の下に

「Problems 数字」のように表示されます。

micro:bit「サンプル プログラミング集」 (第4.0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
2020/08/25 第3.5版
2020/08/30 修正
2021/01/04 第4.0版
2021/03/04 修正