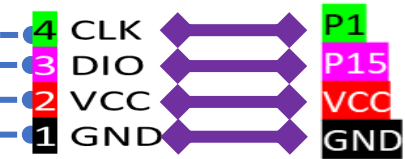


7セグメント 4桁LEDを使おう TM1637

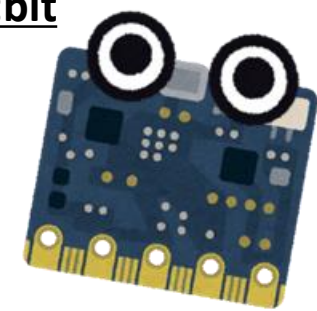


4桁の数字 (0~9999) を表示できます。
センサーの測定値 (整数) や変数などの数値を表示できます。
コロン (:) も点灯でき、時計としての活用もできます。

TM1637

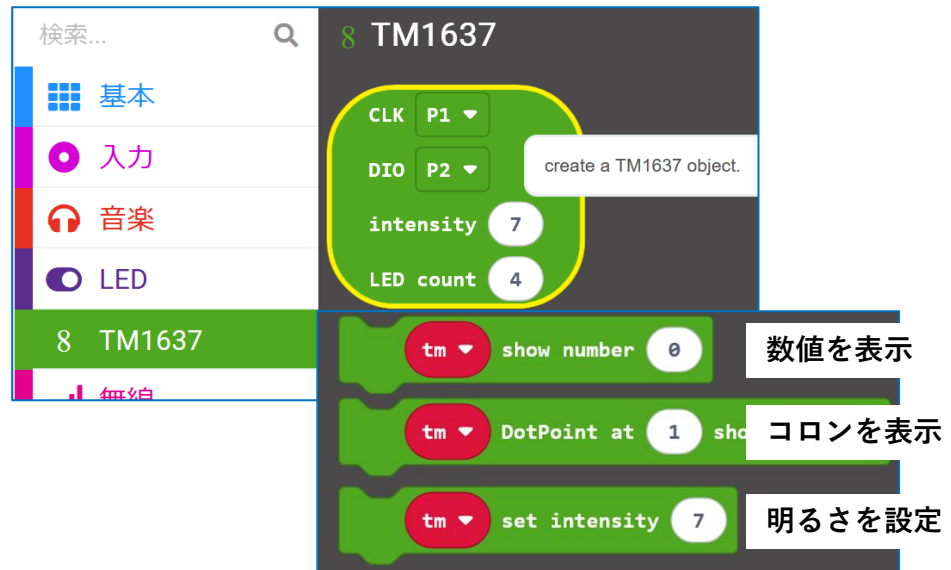


micro:bit



1

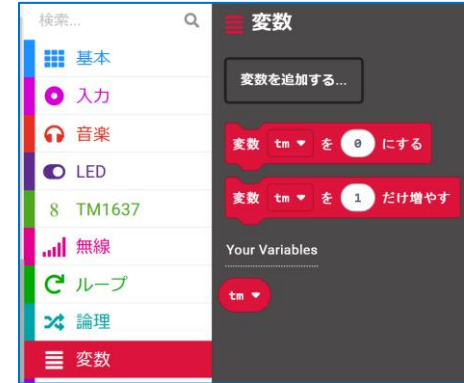
TM1637の拡張機能を追加します。
検索窓に、 TM1637 を入力します。



2

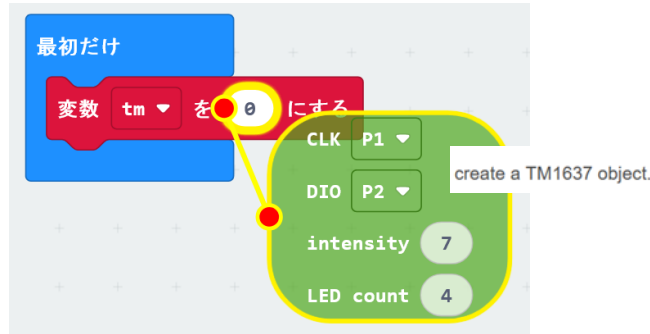
ずっとブロックに、**show number**ブロックを入れる。すると変数ブロックに、変数tmが表示される。

TM1637 > show number



3

最初だけブロックに、変数 **tm** ブロックを入れ、さらに**TM1637 object**ブロックを入れる。



4

TM1637 objectブロックの **CLK** と **DIO** のI/O端子番号を変更する。

- **intensity** : 明るさ 0~7
- **intensity** : 桁数



- 好きな数字を入れてみよう
- 入力 > **明るさ** ブロックを入れてみよう



プログラムをダウンロードして動作を確認しよう



いろいろと試して、どのような活用ができるか考えてみよう!



3種類のサンプルプログラムをそれぞれ作って、
どのように動作するか観察してみよう。
また、自由に改造してみよう！！

ルーレット

4桁の数字 (0~9999) を
ランダムに表示

秒数計

稼働時間を秒数で表示

1秒 = 1000ミリ秒 ÷

コロンの点滅

コロンを 1 秒ごとに点滅

便利に使えるようにしよう！

ボタンを押して、3種類のサンプルプログラムを切り替えて操作できるようにします。

The main Scratch code is organized into three sections, each highlighted with a yellow border:

- ルーレット (Roulette):** Starts with a loop 'もし bt = 1 なら' (If button 1 is pressed). It contains a 'くりかえし 10 回' (Repeat 10 times) block with 'tm show number 0 から 9999 までの乱数' (Random number 0-9999), a '一時停止 (ミリ秒) 100' (Pause 100ms) block, and another '一時停止 (ミリ秒) 2000' (Pause 2000ms) block.
- 秒数計 (Stopwatch):** Starts with a loop 'でなければもし bt = 2 なら' (If button 2 is pressed). It contains a 'tm show number 小数点以下切り捨て (truncate) 稼働時間 (ミリ秒) / 1000' (Show number: truncate working time in ms / 1000) block and a '一時停止 (ミリ秒) 10' (Pause 10ms) block.
- コロンの点減 (Dot Point Reduction):** Starts with a loop 'でなければもし bt = 3 なら' (If button 3 is pressed). It contains two 'tm DotPoint at 1 show' blocks: one with '真' (True) and one with '偽' (False), each followed by a '一時停止 (ミリ秒) 500' (Pause 500ms) block.

Three callout boxes on the right show detailed views of the button triggers:

- ルーレット:** 'ボタン A が押されたとき' (When button A is pressed) - '変数 bt を 1 にする' (Set variable bt to 1) - '文字列を表示 "A"' (Show text "A") - 'clear tm' (Clear timer).
- 秒数計:** 'ボタン B が押されたとき' (When button B is pressed) - '変数 bt を 2 にする' (Set variable bt to 2) - '文字列を表示 "B"' (Show text "B") - 'clear tm' (Clear timer).
- コロンの点減:** 'ボタン A+B が押されたとき' (When button A+B is pressed) - '変数 bt を 3 にする' (Set variable bt to 3) - '文字列を表示 "C"' (Show text "C") - 'clear tm' (Clear timer).

Labels with arrows point to the corresponding code sections: 'ルーレット', '秒数計', and 'コロンの点減'.

使いやすいですか？
・Aボタンを押したら、数値が増える
・Bボタンを押したら、数値が減る
ようにするにはどのように改造すればいいのかな考えてみよう。

お天気パレット サンプルプログラム-1

モニターの設定
OLED

環境センサーの設定
BME280

7セグ4桁LEDの設定

7セグ4桁LEDのをルーレット
風に数値を表示
(5秒間: 50*100)

7セグ4桁LEDに起動時に
表示する測定値
1: 温度 2: 湿度 3: 気圧
4: 防水温度 5: 暑さ指数

測定間隔の設定
1秒: 1000 5秒間隔: 1000*5*1
1分間隔: 1000*60*1 10分間隔: 1000*60*10

1000 × 5 × 1 ミリ秒ごとに

変数 count を 1 だけ増やす

点灯 x 0 y 0

呼び出し keisoku

呼び出し atusa

呼び出し moni-1

呼び出し moni-2

呼び出し LED

呼び出し 7segLED

消灯 x 0 y 0

測定値をボタン (A/B) で選択して
7セグ4桁LEDに表示する呼出関数

関数 keisoku

センサーの測定値を変数に記録

function keisoku

変数 ondo-1 を temperature c にする

変数 sitsudo を humidity にする

変数 kiatsu を 小数点以下四捨五入 pressure Pa / 10 / 10 にする

変数 ondo-2 を 小数点以下四捨五入 temperature (°C) on P2 にする

関数 atusa

暑さ指数を計算

function atusa

変数 WBGT を -3.246 にする

変数 WBGT を 0.725 × ondo-1 だけ増やす

変数 WBGT を 0.0368 × sitsudo だけ増やす

変数 WBGT を 0.00364 × ondo-1 × sitsudo だけ増やす

変数 WBGT を 小数点以下四捨五入 WBGT にする

お天気パレット サンプルプログラム-2

関数 LED

暑さ指数をLEDアイコンで表示

function LED

```
もし WBGT ≥ 31 なら
  アイコンを表示
でなければもし WBGT ≥ 28 なら
  アイコンを表示
でなければもし WBGT ≥ 25 なら
  アイコンを表示
でなければもし WBGT ≤ 24.9 なら
  アイコンを表示
でなければ
  アイコンを表示
```

モニターに単位等を表示

関数 moni-2

function moni-2

```
show string at x 2
y 0
text "c"
color 1
show string at x 7
y 0
text "%"
color 1
show string at x 7
y 1
text "hPa"
color 1
show string at x 3
y 2
text "c"
color 1
show string at x 0
y 3
text "WBGT"
color 1
```

関数 moni-

モニターに測定値を表示

function moni-1

```
show a Number at x 0
y 0
number ondo-1
color 1
show a Number at x 5
y 0
number sitsudo
color 1
show a Number at x 0
y 1
number kiatsu
color 1
show a Number at x 1
y 2
number ondo-2
color 1
show a Number at x 5
y 3
number WBGT
color 1
show a Number at x 9
y 3
number count
color 1
```

お天気パレット サンプルプログラムー3

7セグ4桁LEDに表示する測定値を選択
(数値を増やす)

ボタン A が押されたとき

もし 7seg < 5 なら

変数 7seg を 1 だけ増やす

数を表示 7seg

7セグ4桁LEDに表示する測定値を選択
(数値を減らす)

ボタン B が押されたとき

もし 7seg > 1 かつ 7seg ≤ 5 なら

変数 7seg を -1 だけ増やす

数を表示 7seg

7セグLEDに表示する測定値を選択
切り替えは、A/Bボタン

1. 温度
2. 湿度
3. 気圧
4. 防水温度
5. 暑さ指数

function 7segLED

関数 7segLED

もし 7seg = 1 なら

tm show number ondo-1

でなければもし 7seg = 2 なら -

tm show number sitsudo

でなければもし 7seg = 3 なら -

tm show number kiatsu

でなければもし 7seg = 4 なら -

tm show number ondo-2

でなければもし 7seg = 5 なら -

tm show number WBGT

計測回数をリセット

ボタン A+B が押されたとき

アイコンを表示

変数 count を 0 にする

clear

表示を消す

・配列を使った表示の切り替え

これまでの**変数**は、**1つの変数で1つの値**だけを記録しました。
配列は、**1つの箱に仕切り**があり**複数の値**を同時に記録できます。

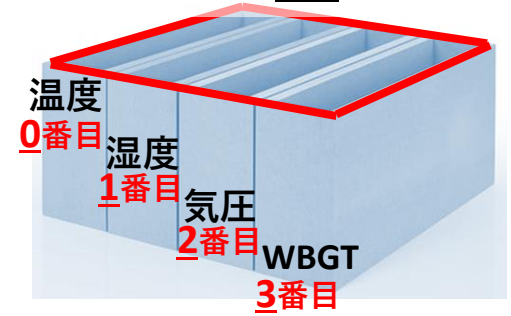
数値の配列

変数 配列 を 配列 0 1 - + にする

文字の配列

変数 文字列の配列 を 配列 "a" "b" "c" - + にする

一つ一つの値を**要素**と呼びます



最初の値は**0番目の値**と呼び、
0（ゼロ）から数えます。

```
ボタン A が押されたとき
もし index < 配列の長さ 配列 なら
  変数 index を 1 だけ増やす
  数を表示 index
+
ボタン B が押されたとき
もし index > 1 なら
  変数 index を -1 だけ増やす
  数を表示 index
```

変数 **配列** は配列ブロックを使うと自動的に作られます
変数 **index** を作ります

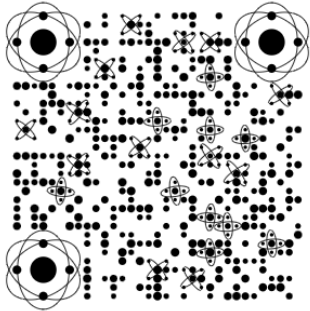
```
function 7segLED表示
  変数 配列 を
  配列
  ondo ← 温度 0番目
  situdo ← 湿度 1番目
  ondo2 ← 温度2 2番目
  WBGT ← 暑さ指数 3番目
  kiatsu ← 気圧 4番目
  akarusa ← 明るさ 5番目
  7segLED show number 配列 の index - 1 番目の値
```

例) 変数 **index** をが1の時、
 $index-1=0$ となり、0番目の温度を表示

出展：© micro:bit Educational

タッチストップウォッチ

<https://microbit.org/ja/projects/make-it-code-it/touch-stopwatch/>



```

ボタン A が押されたとき
  変数 running を 真 にする
  変数 start を 稼働時間 (ミリ秒) にする

```

```

ロゴが 短くタップされた とき
  もし running ではない なら
    数を表示 time integer / 1000

```

```

ボタン B が押されたとき
  もし running なら
    変数 time を 稼働時間 (ミリ秒) - start だけ増やす
  変数 running を 偽 にする

```

```

ずっと
  もし running なら
    アイコンを表示 [stopwatch icon]
    アイコンを表示 [stopwatch icon]
  でなければ
    アイコンを表示 [stopwatch icon]

```

数字をシャッフルする

付録2-36 【実用】ビンゴマシン

📅 2019-10-21 🔄 2019-10-21

```

関数 ビンゴの初期化 最大値
  変数 カウンター を 0 ~ 最大値 に変えてくりかえす
    ビンゴの数字 の カウンター 番目の値を カウンター + 1 にする
  変数 入替元 を 0 ~ 最大値 - 1 に変えてくりかえす
    変数 入替先 を 入替元 から 最大値 - 1 までの乱数 にする
    変数 入替元の数字 を ビンゴの数字 の 入替元 番目の値 にする
    ビンゴの数字 の 入替元 番目の値を ビンゴの数字 の 入替先 番目の値 にする
    ビンゴの数字 の 入替先 番目の値を 入替元の数字 にする

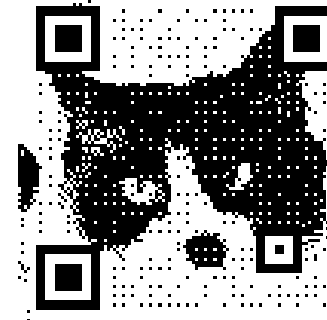
```

数字をシャッフルする
付録2-36.【実用】ビンゴマシン

出展：サヌキテックネット

【実用】ビンゴマシン

<https://sanuki-tech.net/micro-bit/appendix-sample-program/bingo-machine/>



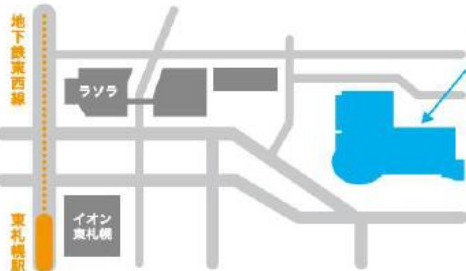
国内最大級のプログラミングイベントが今年も開催!!



2024.12.14 [SAT] ▶ 15 [SUN]
AM10:00~PM16:00

**参加
無料**

体験内容 & お申し込み方法は
裏面をチェック!!



札幌コンベンションセンター

札幌市白石区東札幌 3条 2丁目 1-20

※地下鉄東西線「東札幌駅」の1番出口から徒歩10分

- 【共 催】 札幌市 / 一般財団法人さっぽろ産業振興財団 / 札幌コンベンションセンター / ジュニア・プログラミング・ワールド実行委員会
- 【全 面】 株式会社インフィニットループ / クリプトン・フューチャー・メディア株式会社
- 【運 営】 株式会社北海道新聞社営業局
- 【特別協力】 NoMaps 実行委員会
- 【後 援】 札幌市教育委員会 / 北海道教育委員会 / 北海道私立中学高等学校協会 / 一般社団法人北海道IT推進協会 / 一般社団法人北海道モバイルコンテンツ・ビジネス協議会 / 一般社団法人北海道情報システム産業協会

▼特設ホームページ開設期間

2024年11月5日 (火) 公開

▼参加申込受付期間

2024年11月16日 (土) 10:00

~12月9日 (月) 23:59