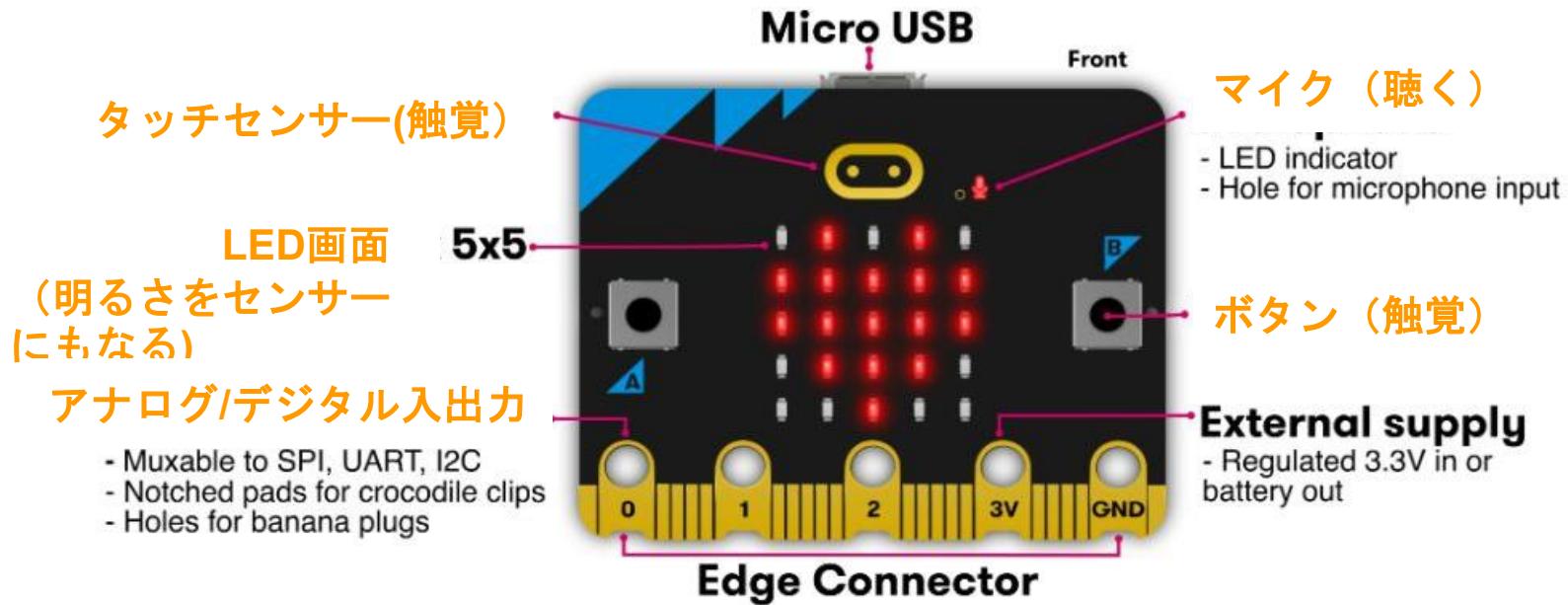
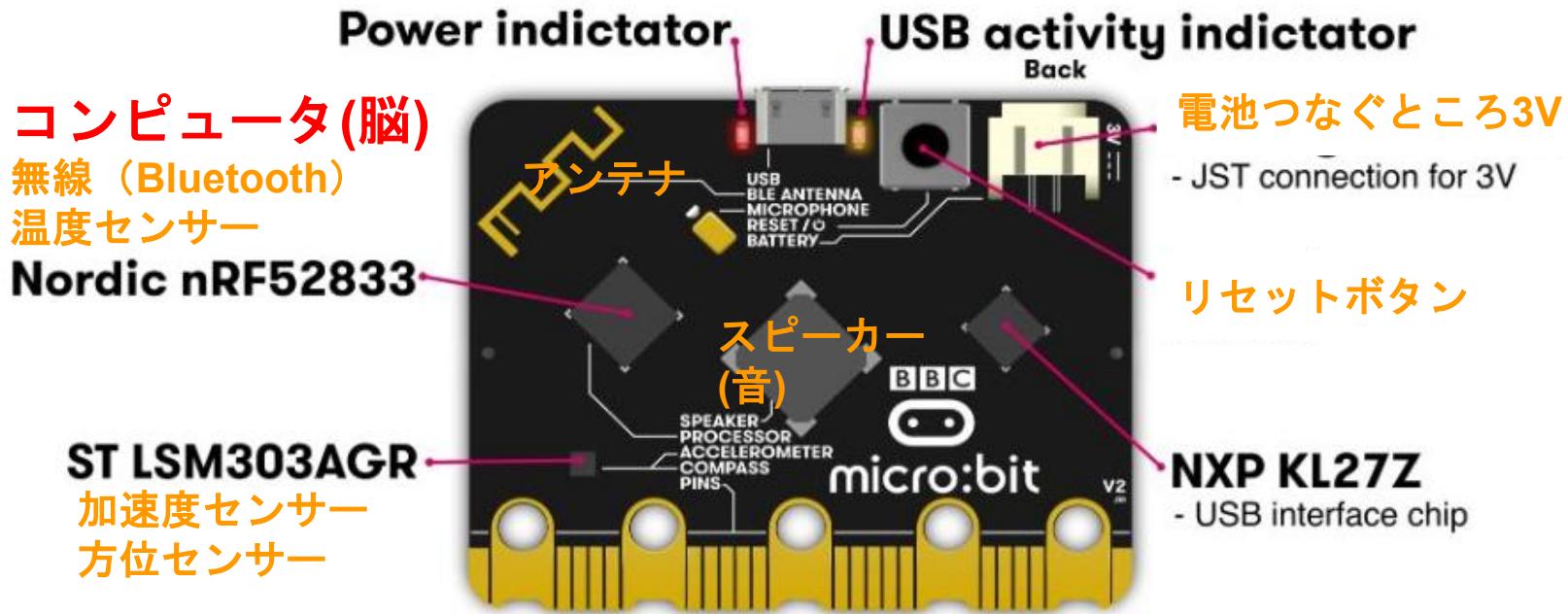


micro:bitでプログラミング

# Micro:bit って何



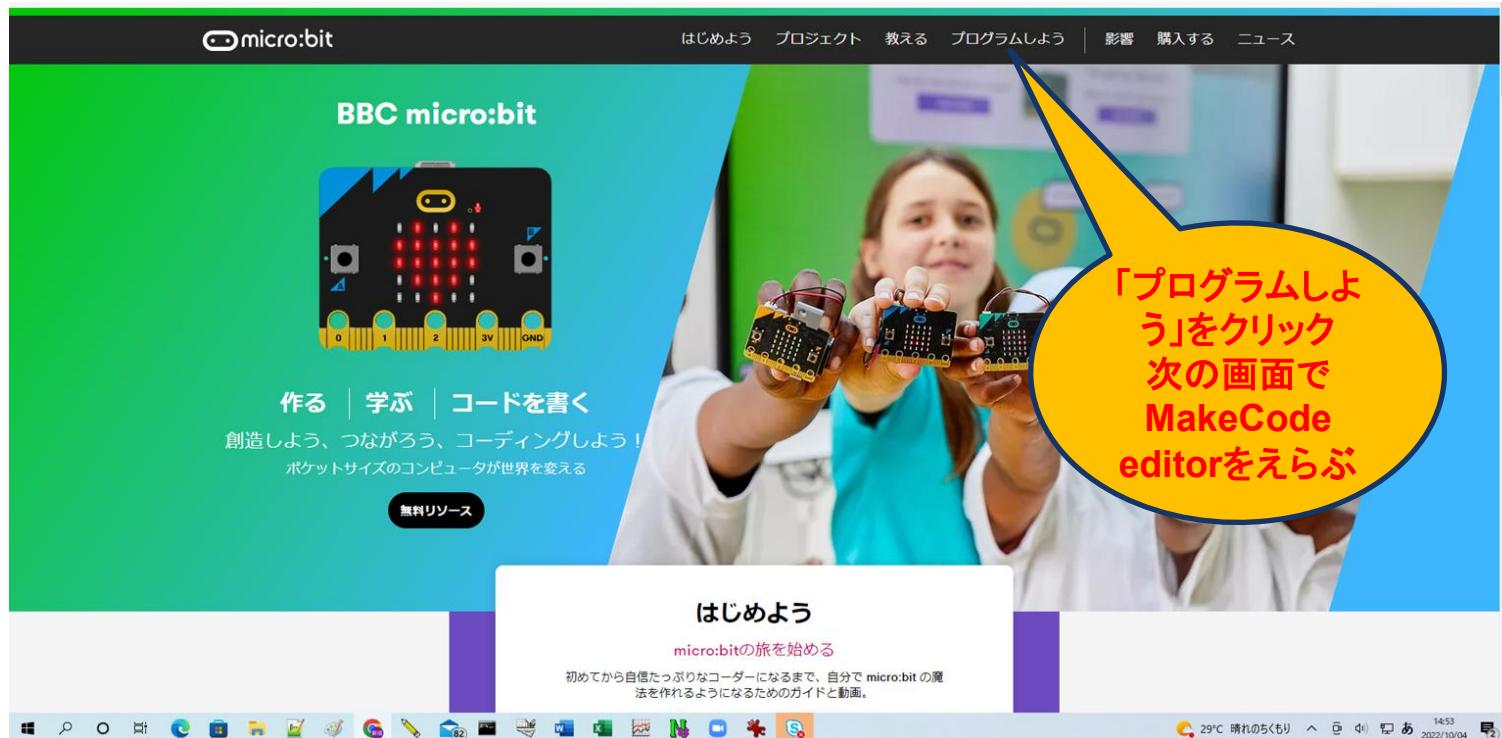
# Micro:bitの裏には何がある？



どんなことができる？

どんなセンサーが働いているか考えながら体験してみてね

# Micro:bitのホームページ (<https://microbit.org/ja/>)

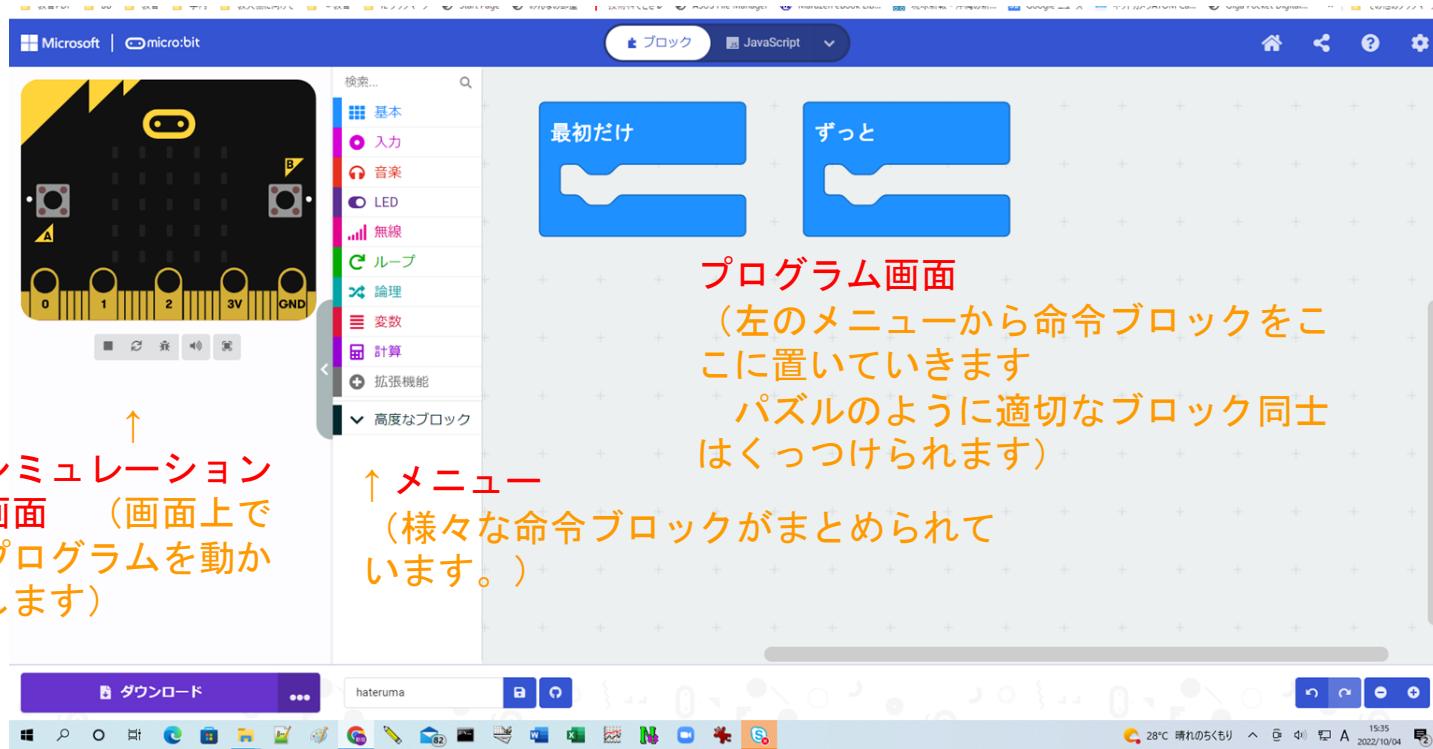


# プログラムしよう (Make Code editor)

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



# プログラム画面



# ずっとドキドキ

The image shows the Microsoft Scratch 3.0 interface for the micro:bit. The stage on the left displays a pattern of red squares on the micro:bit's LED matrix. The script on the right is as follows:

- Initial (when green flag is clicked):
  - Display icon (with a value of 100)
  - Wait (500 ms)
- Loop:
  - Display icon (with a value of 100)
  - Wait (500 ms)

The script is currently set to run in "Block" mode (JavaScript is selected in the dropdown). The "Advanced blocks" category is expanded in the block palette.

# micro:bitにプログラムをダウンロードしてみよう

1. micro:bitをパソコンにUSBケーブルで接続します。
2. プログラム画面左下のダウンロードボタンのすぐ右にある [...] ボタンにマウスを重ねたとき（押さないでね）「micro:bitに接続しています」と出れば、ダウンロードボタンを押すことで、プログラムが micro:bit に送られます
3. 「micro:bitに接続しています」がでなければ、 [...] を押して、まず接続を行ってください

# 基礎（順次、分岐、反復）



順次  
(じゅんじ)

分岐  
(ぶんき)

反復  
(はんぷく)

# 音を鳴らす

反復を行うためにループを使ってみよう

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

The image shows a Scratch script on a light blue background. On the left, a vertical toolbar lists categories: 基本 (Basic), 入力 (Input), 音楽 (Music), LED, 無線 (Radio), ループ (Loop), 論理 (Logic), 変数 (Variables), 計算 (Math), 拡張機能 (Extensions), and 高度なブロック (Advanced blocks). A red arrow points from the 'Loop' category in the toolbar to the 'Loop' control block in the script.

The script consists of the following blocks:

- Control: ポタン [A ボタン] が押されたとき
- Control: ポタン [B ボタン] が押されたとき
- Loop (highlighted in green): くりかえし [2 回]
  - Sound: 鳴らす [音を鳴らす 高さ (Hz) [真ん中のド] 長さ 1 拍 終わるまで]
  - Sound: 鳴らす [音を鳴らす 高さ (Hz) [真ん中のレ] 長さ 1 拍 終わるまで]
  - Sound: 鳴らす [音を鳴らす 高さ (Hz) [真ん中のミ] 長さ 2 拍 終わるまで]
  - Sound: 鳴らす [音を鳴らす 高さ (Hz) [真ん中のド] 長さ 1 拍 終わるまで]
  - Sound: 鳴らす [音を鳴らす 高さ (Hz) [真ん中のレ] 長さ 1 拍 終わるまで]
  - Sound: 鳴らす [音を鳴らす 高さ (Hz) [真ん中のミ] 長さ 2 拍 終わるまで]

暗くなったらLEDをつける

## 分岐を行うために論理を使ってみよう



# 変数を使ってみよう

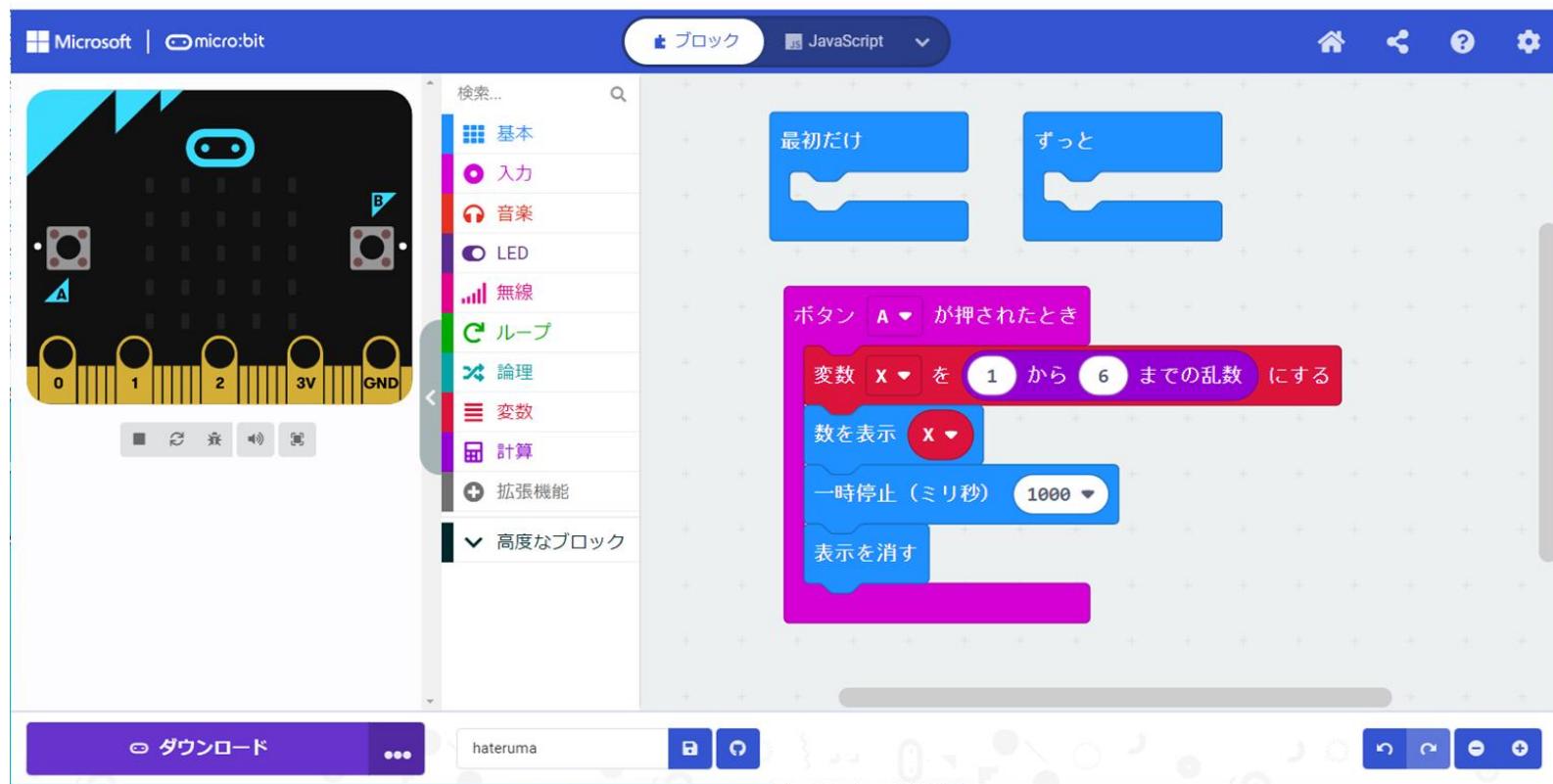
変数から「変数を追加する」を選び、変数  $x$  をつくります。

The image shows the Scratch script editor interface. On the left, a sidebar lists categories: 基本 (Basic), 入力 (Input), 音楽 (Music), LED, 無線 (Wireless), ループ (Loop), 論理 (Logic), 变数 (Variables), and 計算 (Math). The 'Variables' category is selected, indicated by a red background.

The main workspace displays a script using variable  $x$ :

- Step 1: **変数を追加する...** (Add variable...)
- Step 2: **変数  $x$  を 0 にする** (Set variable  $x$  to 0)
- Step 3: **変数  $x$  を 1 だけ増やす** (Change variable  $x$  by 1)
- Step 4: **最初だけ** (Only once):
  - 変数  $x$  を 140 にする** (Set variable  $x$  to 140)
  - 数を表示  $x$**  (Show number  $x$ )
- Step 5: **ずっと** (Forever):
  - アイコンを表示** (Show icon) (with a blue square icon selected)
  - LED画面の明るさを設定する  $x$**  (Set LED screen brightness to  $x$ )
- Step 6: **ボタン A が押されたとき** (When button A is pressed):
  - 変数  $x$  を -20 だけ増やす** (Change variable  $x$  by -20)
  - 数を表示  $x$**  (Show number  $x$ )
- Step 7: **ボタン B が押されたとき** (When button B is pressed):
  - 変数  $x$  を 20 だけ増やす** (Change variable  $x$  by 20)
  - 数を表示  $x$**  (Show number  $x$ )

# 変数 x と乱数（ランダムな数）を使ってみよう



# 無線通信

● 入力

● 音楽

● LED

● 無線

● ループ

● 論理

● 変数

● 計算

● 拡張機能

▼ 高度なブロック

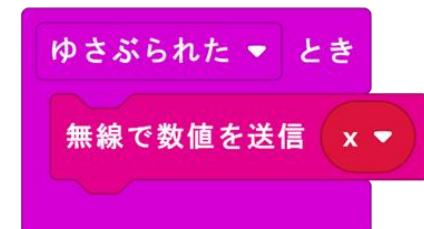
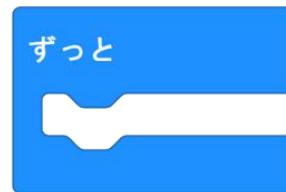
送信  
機側



受信  
機側



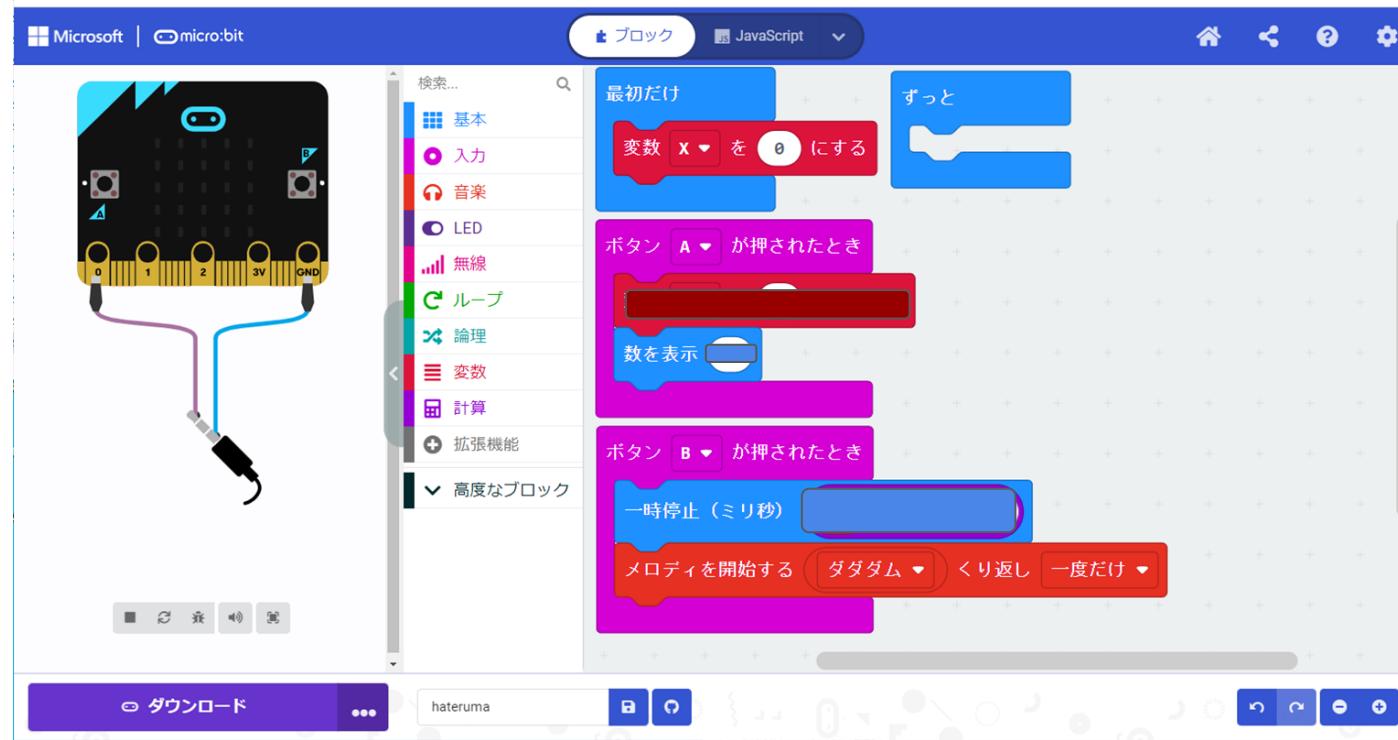
# 無線で変数値を送信、受信の例



考えてみよう

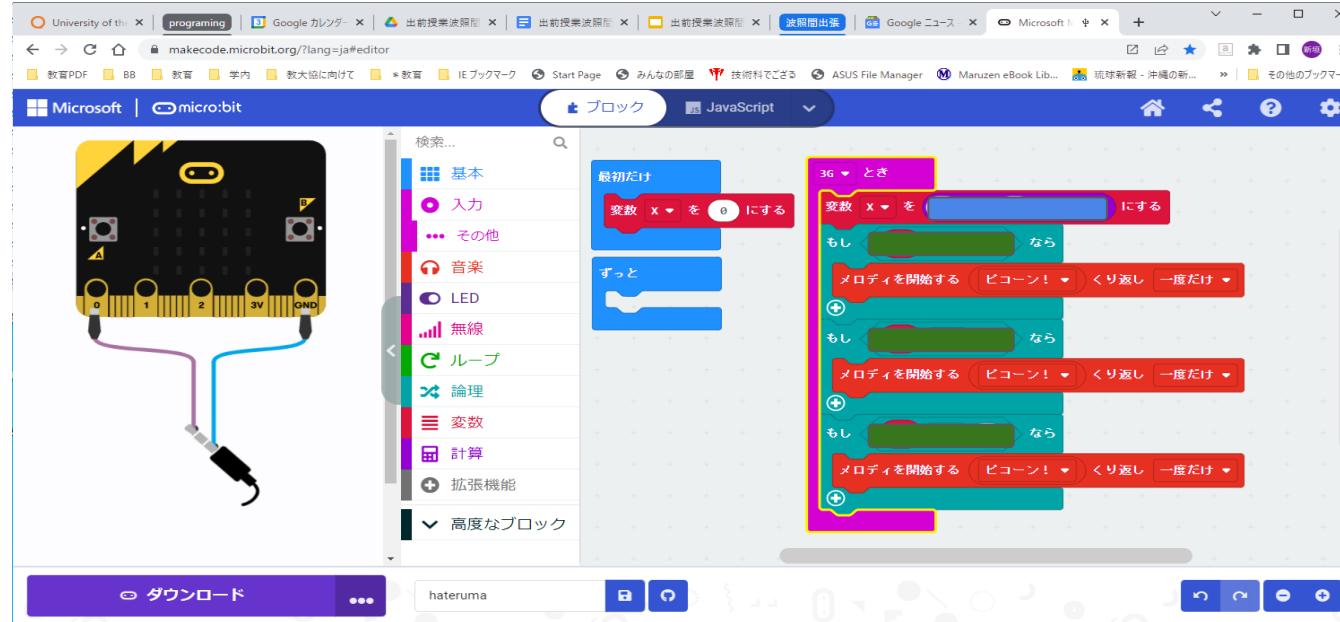
# タイマーのプログラムを考えてみてください

Aボタンで分を1ずつ増やし、Bボタンでスタート、時間になつたらメロディー

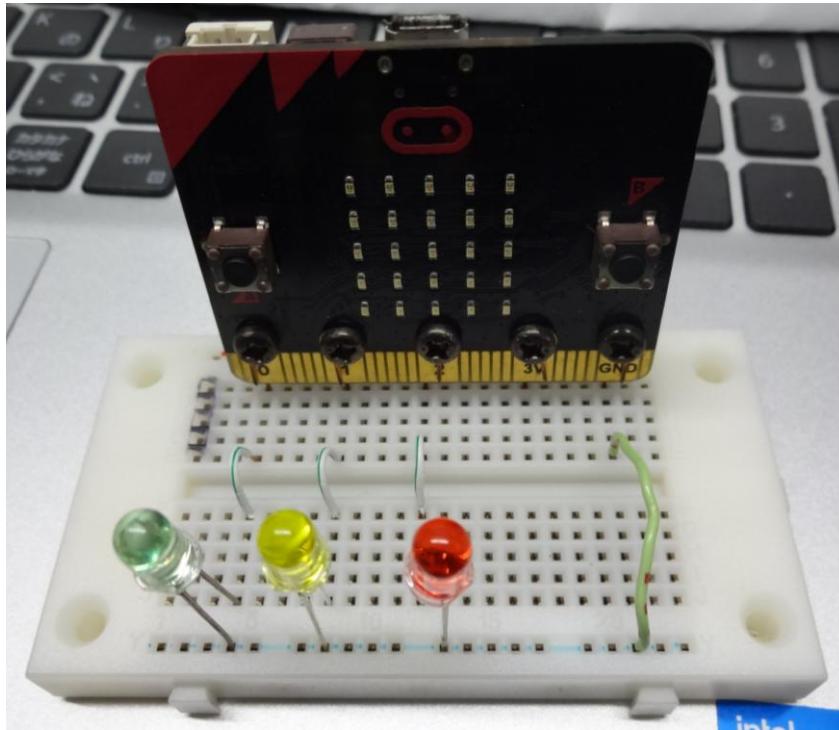


# じゃんけんプログラムを考えてみてください

ふった勢いで変数  $\times$  に1から3乱数を入れ、 $\times$  が何かによってグー、チョキ、パーを表示してみよう。メロディーブロックの下に[LED画面に表示]を入れて下さい



# 入出力端子の利用例



- マイクロビット下部にあるP0、P1、P2それぞれに、青、黄、赤のLEDを接続し、信号機を作つてみる

# 入出力端子のプログラム

Scratch script titled "◎ 入出力端子" (Input/Output Pins). It contains the following blocks:

- デジタルで読み取る 端子 P0
- デジタルで出力する 端子 P0 値 0 (highlighted with a blue oval)
- アナログ値を読み取る 端子 P0
- アナログで出力する 端子 P0 値 1023
- 数値をマップする 0 (red hat block)
  - 元の下限 0
  - 元の上限 1023
  - 結果の下限 0
  - 結果の上限 4
- アナログ出力 パルス周期を設定する 端子 P0 周期 (マイクロ秒) 20000
- 音を鳴らす端子を P0 にする

Scratch script titled "ずっと" (Loop). It contains the following blocks:

- デジタルで出力する 端子 P0 値 1
- 一時停止 (ミリ秒) 5000
- デジタルで出力する 端子 P0 値 0
- デジタルで出力する 端子 P1 値 1
- 一時停止 (ミリ秒) 1000
- デジタルで出力する 端子 P1 値 0
- デジタルで出力する 端子 P2 値 1
- 一時停止 (ミリ秒) 5000
- デジタルで出力する 端子 P2 値 0

デジタルで  
値1を出力す  
ると端子に  
は電圧3Vが  
である。  
値0の場合0V  
となる。

青が5秒、黄  
色が1秒、赤  
が5秒の点灯  
を繰り返す。

