

# ステップ3

スマートホームを  
プログラミングして動かしてみよう

# micro:bit用拡張ボード

とがっている部分があるので注意してね！

ブザースイッチ  
ブザー

P0~20、  
3V電源

5V電源  
(3個)

オーディオ  
端子(P0)

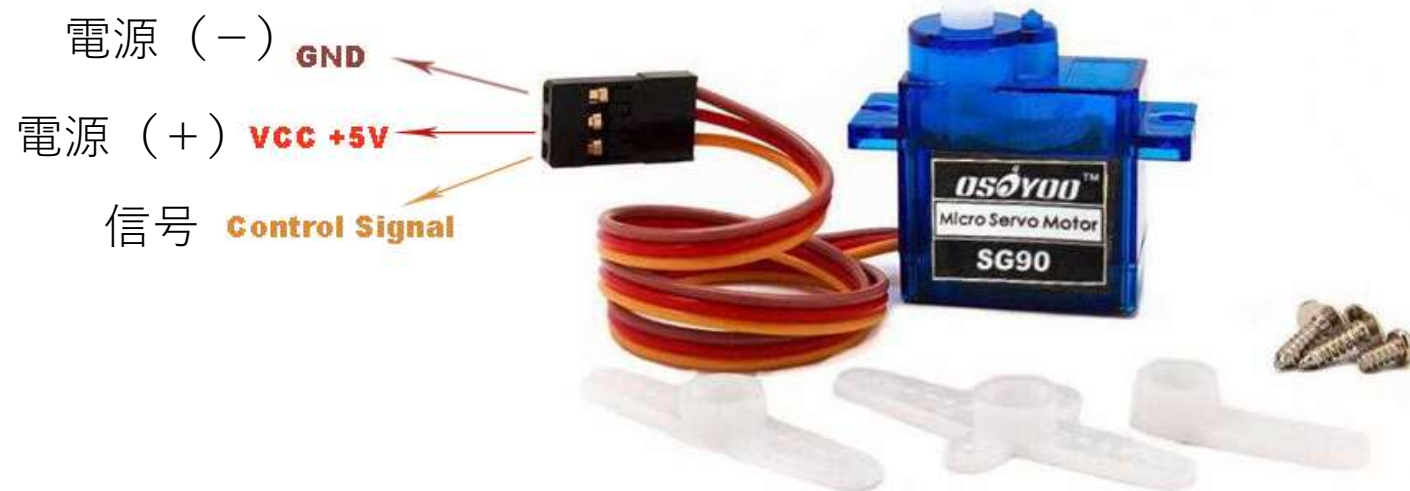
ワニ口クリップ用端子  
(P0~2、3V、GND)

5V電源用USB端子(入力)  
5V電源スイッチ



# サーボモーターについて

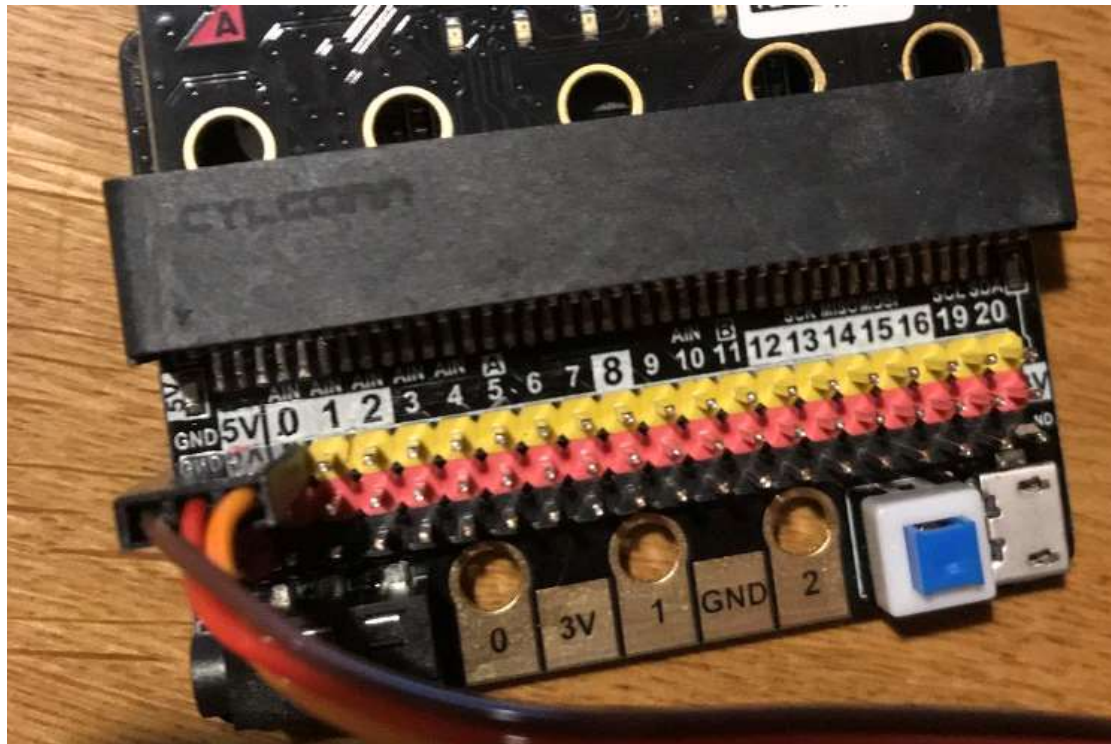
- モーターは電気を流すとそれを力に変えて物を動かす装置です。
- モーターにもいろいろな種類がありますが、サーボモーターは角度を指定すると、その角度まで回転して止まるモーターです。
- 回転できる角度には制限（ふつうは0度～180度まで）があり、普通のサーボモーターはずっと同じ方向に回すことはできません。
- サーボモーターにも360度回転して同じ方向にずっと回すことができる種類のものもあります。



# サーボモーターを接続

とがっている部分があるので注意してね！

- micro:bitの拡張ボードに、サーボモーターのケーブルを接続しましょう。
- サーボモーターのケーブルは赤が5V（+）、こげ茶がGND（-）、オレンジが0番端子につながるようにしてね。



# サーボモーターを動かそう

- サーボモーターを動かしてみましょう。



# 自動ドアを作ってみよう

- サーボモーターをレゴで歯車につなげて、プログラミングで自動ドアにしてみよう。

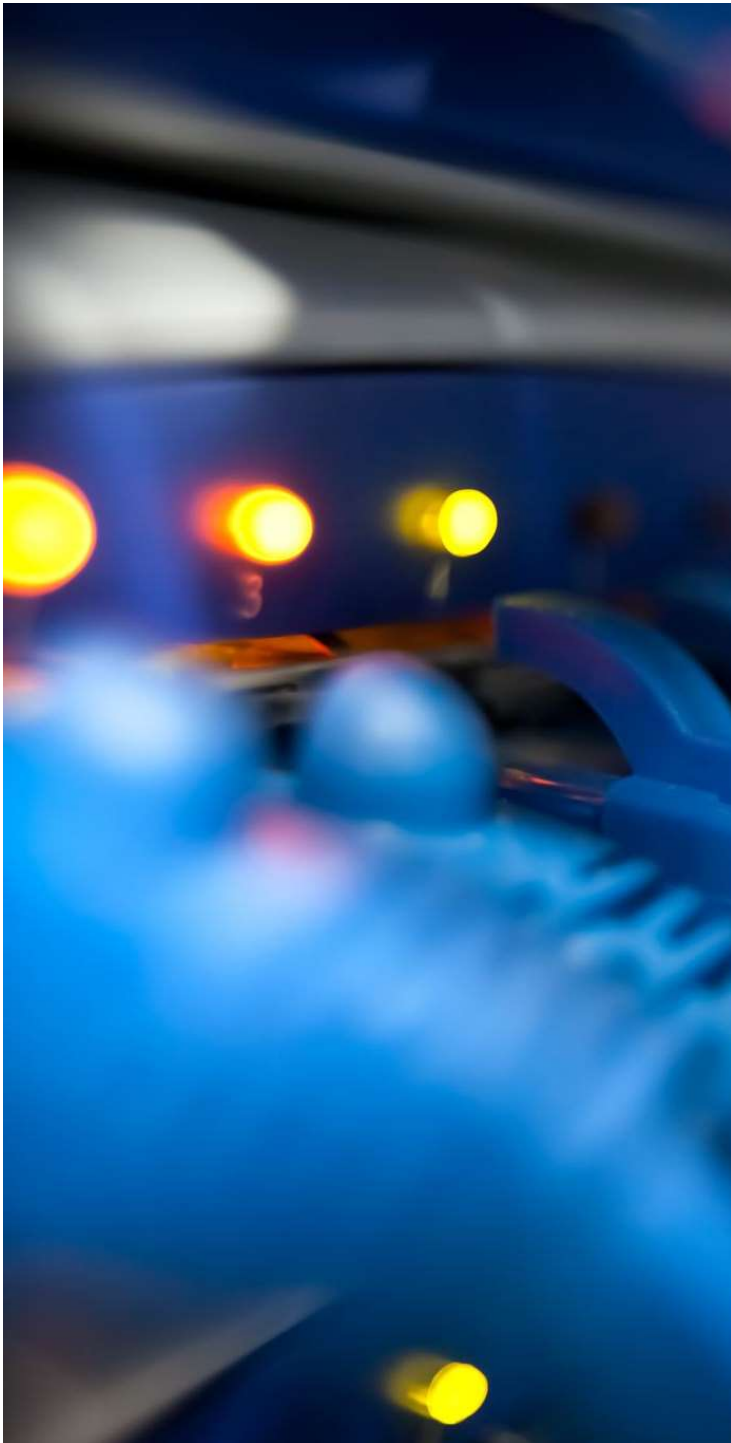




# 温度計とドアを連携

- 温度計と自動ドアを連携してみましょう。
- 気温がある温度より高くなったら開き、低くなったら閉じるようにプログラミングしてみましょう。





## LEDについて

---

LEDは発光ダイオードという電子部品ですが、以前は赤や緑色に光るものしかありませんでした。

青や白色に光るものが発明され、より明るく光るものも多くなり、今では照明器具として広く使われています。電球や蛍光灯と比べても省エネで寿命が長いのが特徴です。

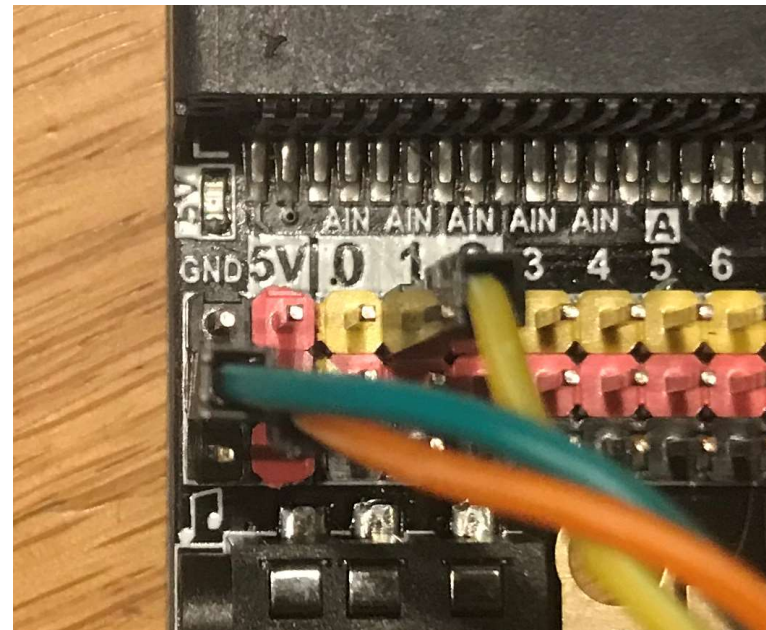
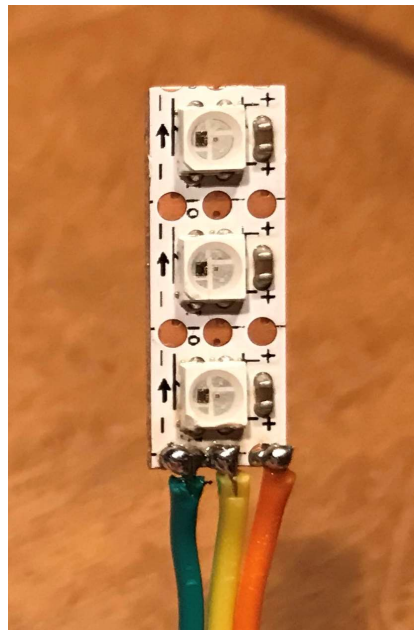
今日使うLEDはフルカラーLEDで、プログラミングによって色々な光らせ方ができます。



# フルカラーLEDをつなげましょう

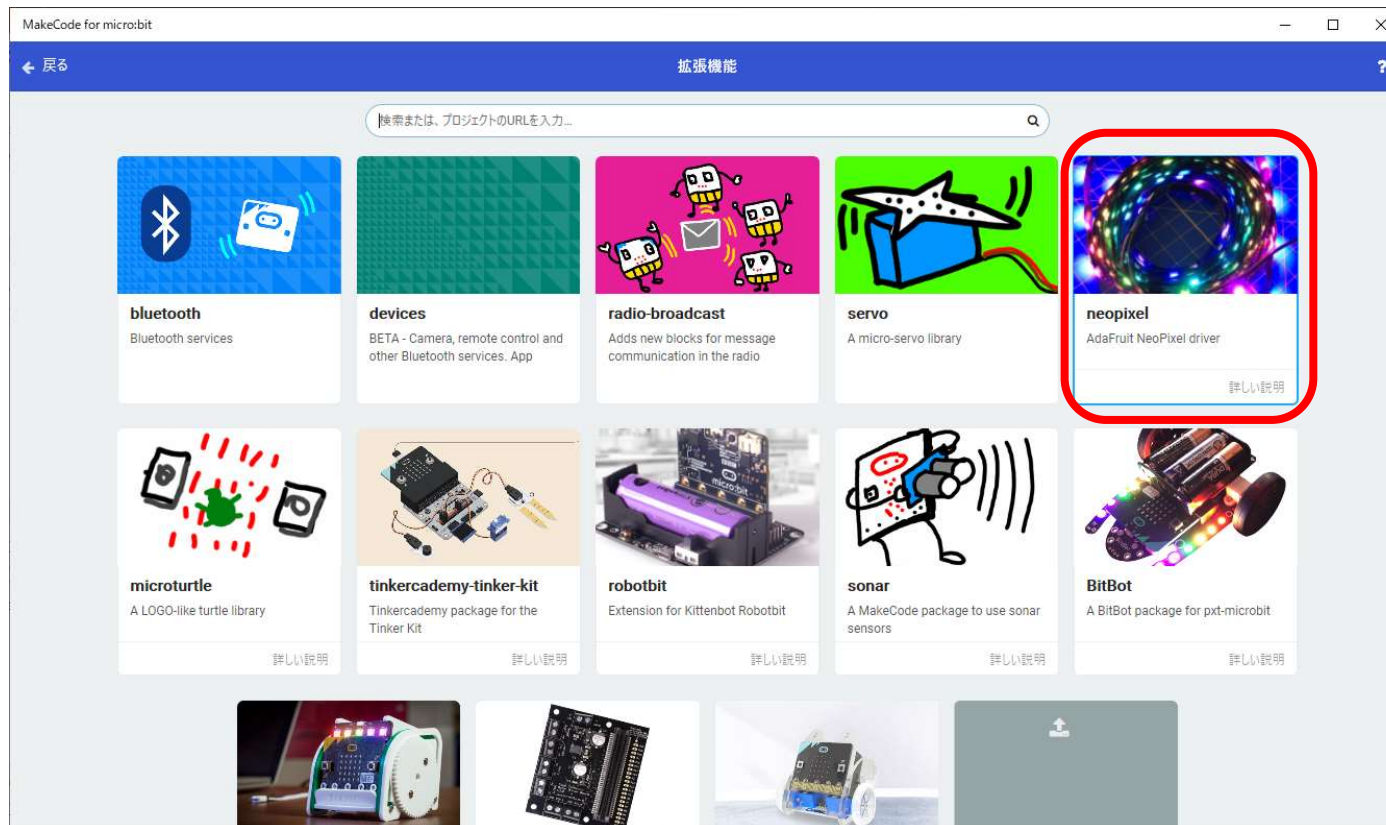
とがっている部分があるので注意してね！

- micro:bitにフルカラーLEDを接続しましょう。
- 拡張ボードのP1端子(1と書いてある列の黄色の端子)に、LEDの信号線(写真では真ん中の黄色の線)をつなげます。
- LEDの表面にプリントしてある+と-の記号をよく見て、そこにつながっている線と拡張ボードの5V端子とをつなぎます。  
(+(写真ではオレンジ)の線を5V端子、-(写真では緑)の線をGND端子につなぐ)



# LEDを光らせましょう

- 「拡張機能」から「neopixel」を選んでフルカラーLED用のブロックを追加します。



# LEDを光らせよう（テスト）

- 次のようにプログラミングしてみましょう。



# LEDを光らせよう（自動点灯）①

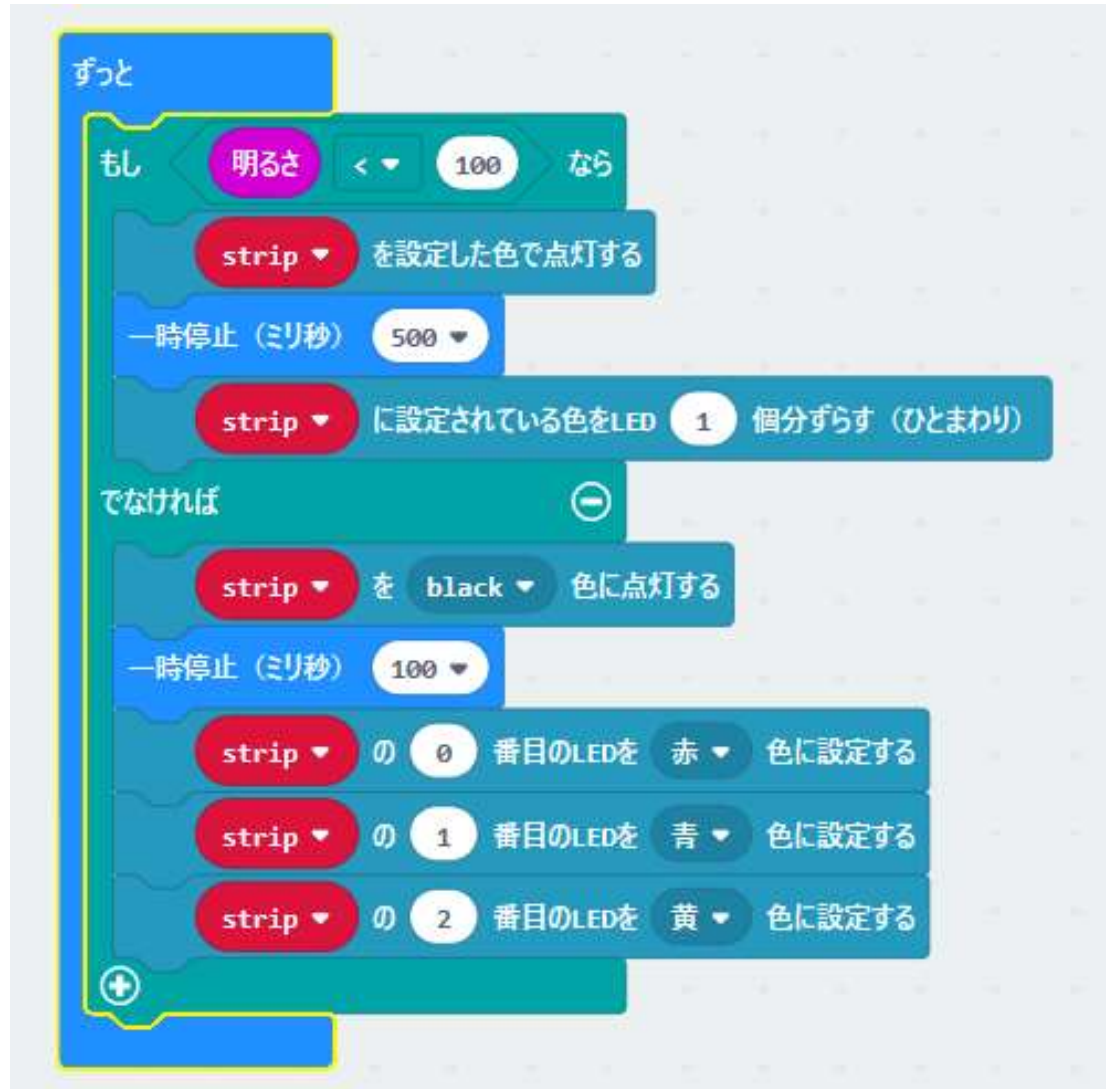
- 明るさで点いたり消えたりさせてみましょう。



次のページへつづく

# LEDを光らせよう（自動点灯）②

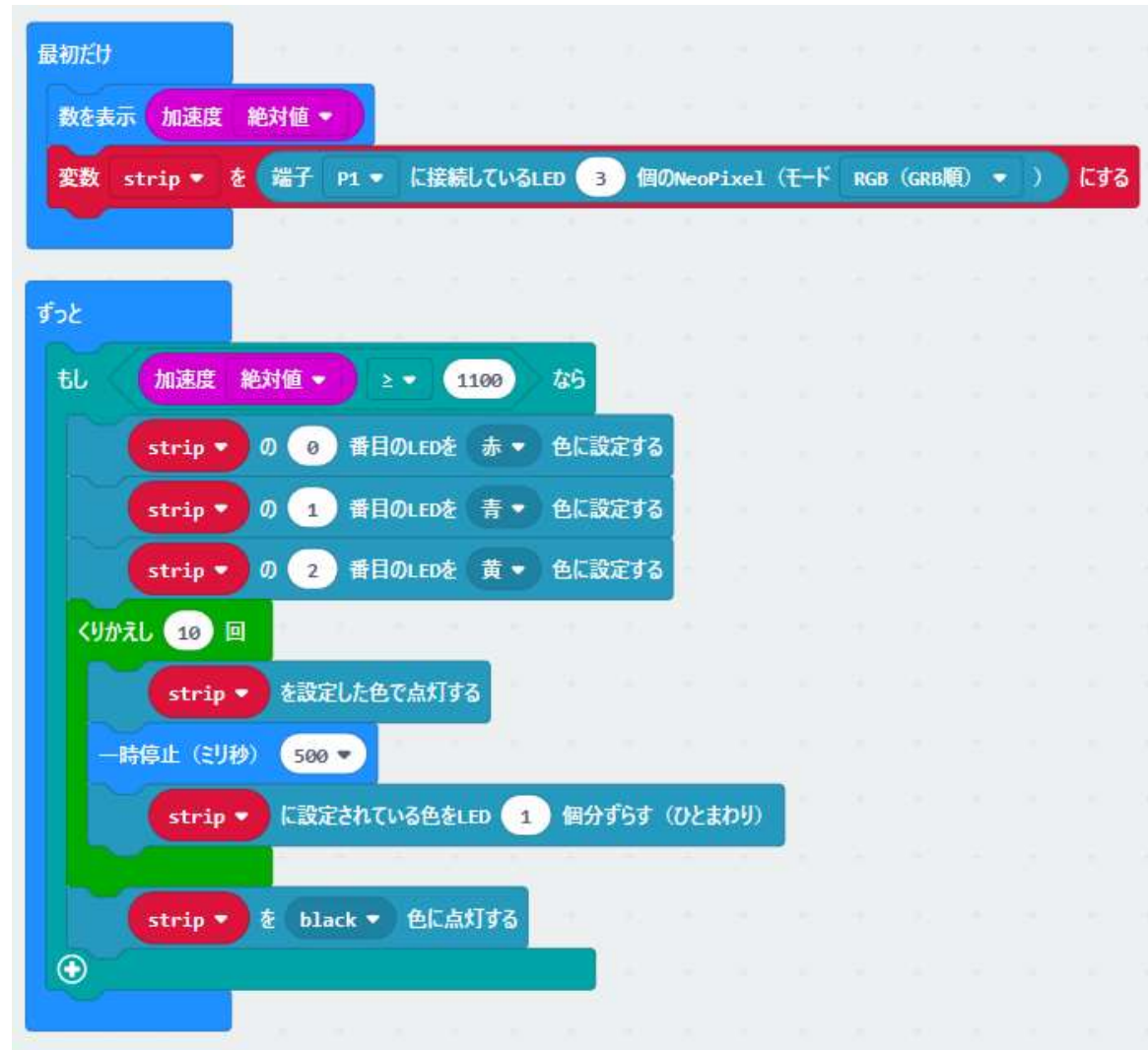
前のページのプログラムのつづき





# LEDを光らせよう

- 次のようにプログラミングしてみましょう。（振動で光る）



# LEDを光らせよう（ボタンで制御）

- 次のようにプログラミングしてみましょう。（A/BボタンでON/OFF）

The diagram shows a Scratch-style block diagram for controlling an LED strip. It consists of three main parts: an initialization block, and two event-driven loops for buttons A and B.

**Initialization Block (Blue):**

- 最初だけ (Only at the start)
- 変数 strip を 端子 P1 に接続しているLED 3 個のNeoPixel (モード RGB (GRB順)) にする (Set variable strip to the 3 NeoPixels connected to terminal P1 (mode RGB (GRB order)))

**Button A Event Loop (Pink):**

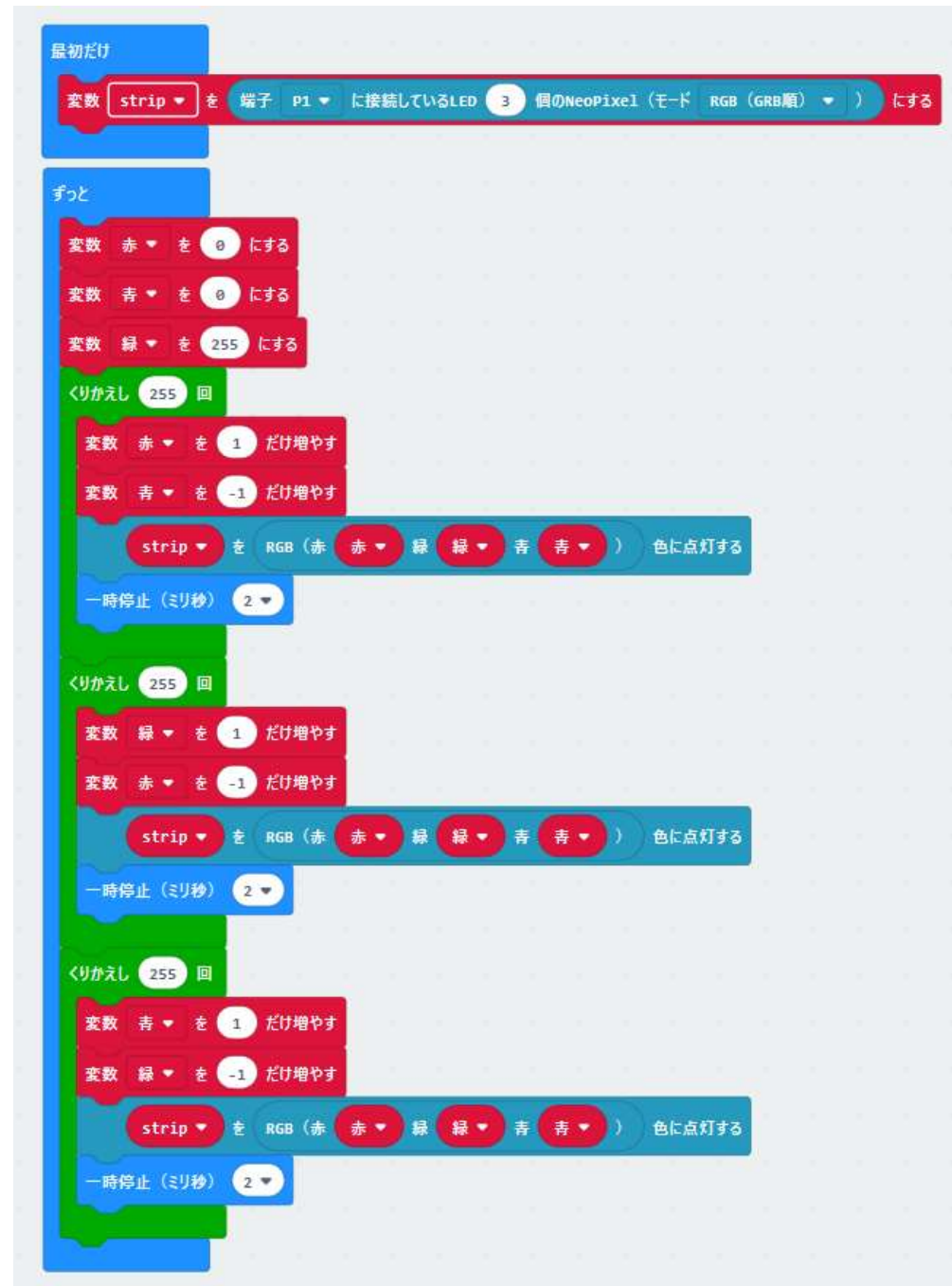
- ボタン A が押されたとき (When button A is pressed)
- 変数 switch を 1 にする (Set variable switch to 1)
- strip の 0 番目のLEDを 赤 色に設定する (Set the 0th LED of strip to red)
- strip の 1 番目のLEDを 青 色に設定する (Set the 1st LED of strip to blue)
- strip の 2 番目のLEDを 黄 色に設定する (Set the 2nd LED of strip to yellow)
- もし switch = 1 ならくりかえし (If switch = 1, loop)
- strip を設定した色で点灯する (Turn on strip with the set color)
- 一時停止 (ミリ秒) 500 (Pause 500 milliseconds)
- strip に設定されている色をLED 1 個分ずらす (ひとまわり) (Shift the color of the LED 1 position (one round))

**Button B Event Loop (Yellow):**

- ボタン B が押されたとき (When button B is pressed)
- 変数 switch を 0 にする (Set variable switch to 0)
- もし switch = 0 ならくりかえし (If switch = 0, loop)
- strip を black 色に点灯する (Turn on strip with black color)
- 一時停止 (ミリ秒) 100 (Pause 100 milliseconds)

# LEDを光らせよう (不思議な色)

- 色が不思議な  
  感じで変わります。



# メッセージを送信しよう（送信側）

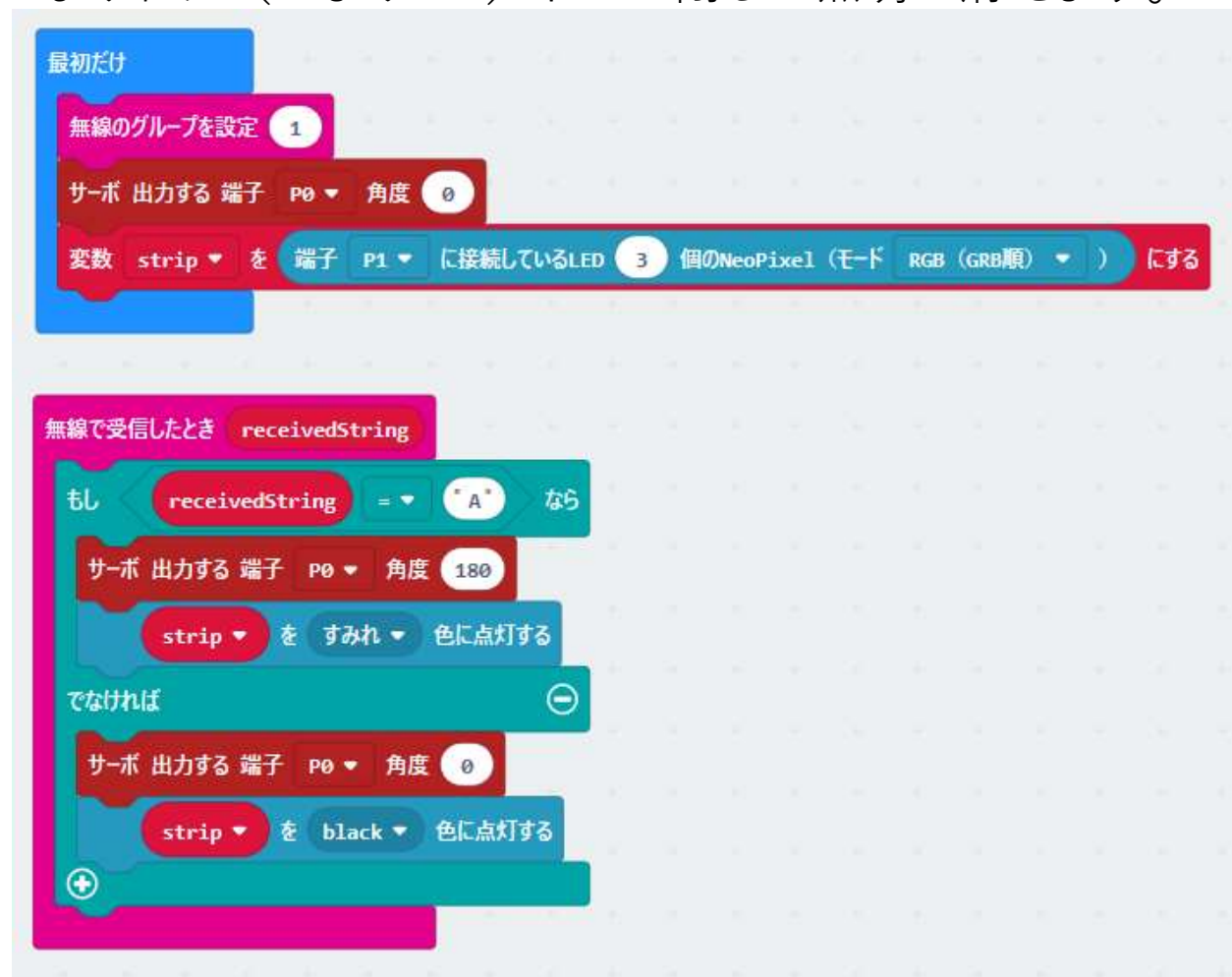
- micro:bitには無線でメッセージを送受信する機能があります。
- 送信側を次のようにしてみましょう。



- Aボタン、Bボタンを押すと信号（"A"または"B"という文字）が無線グループ1のみんなに送られます。

# メッセージを受信したら… (受信側)

- 受信側を次のようにしてみましょう。
- 受信したメッセージが”A”ならドアが開いて照明が点きます。
- そうでなければ (つまり”B”) ドアが閉じて照明が消えます。





# スマートシティに挑戦（家側）

- みんなの家の明るさを計って、まち全体の照明を点けるデモ（各家側のプログラム例）
- 無線でデータを送信（例："taro"の家の明るさを計ってセンターに送信する。）
- センターからスイッチ(switch)のコマンドを受信して動かす。

The diagram shows a Scratch-style block diagram for a program. It is organized into three main sections: '最初だけ' (Only at the beginning), 'ずっと' (Forever loop), and '無線で受信したとき' (When received wirelessly).

**最初だけ (Only at the beginning):**

- 変数 strip を 端子 P0 に接続しているLED 3 個のNeoPixel (モード RGB (GRB順)) にする
- 無線のグループを設定 1
- サーボ 出力する 端子 P0 角度 0

**ずっと (Forever loop):**

- 無線で送信 "taro" = 明るさ
- 棒グラフを表示する 値 明るさ
- 最大値 255
- 一時停止 (ミリ秒) 100

**無線で受信したとき (When received wirelessly):**

- もし name = "switch" なら
  - もし value = 1 なら
    - strip を 白 色に点灯する
    - サーボ 出力する 端子 P0 角度 0
  - でなければ
    - strip を black 色に点灯する
    - サーボ 出力する 端子 P0 角度 180

A yellow block labeled '名前を変える' (Change name) is shown, with a red arrow pointing from it to the '無線で送信' block in the 'ずっと' loop.

# スマートシティに挑戦（センター側）

- センターで各家の明るさが分かる。（例："taro"と"hanako"の家）
- 明るさの合計を出して全体が暗かったら、まち全体の照明を点ける。

The image displays two Scratch scripts for a smart city center system. The left script, titled '最初だけ' (Only at the start), sets a radio group to 1 and then enters a 'ずっと' (Forever) loop. In this loop, it adds the brightness values of 'taro' and 'hanako' to a variable named 'goukei'. It then sends serial messages with the names and values of 'taro', 'hanako', and 'goukei'. The right script, titled '無線で受信したとき' (When received via radio), has two 'もし' (If) blocks. The first checks if the received name is 'taro'; if so, it sets the 'taro' variable to the received value and toggles a switch (x=0, y=0). The second checks if the name is 'hanako'; if so, it sets the 'hanako' variable to the received value and toggles the switch (x=0, y=1). Below these is another 'ずっと' (Forever) loop that checks if 'goukei' is less than 50 and the switch is 0. If true, it sets the switch to 1, sends a radio message with 'switch' = 1, and plays a sound (upper D, 1/8 note). If 'goukei' is greater than or equal to 50 and the switch is 1, it sets the switch to 0, sends a radio message with 'switch' = 0, and plays a sound (middle D, 1/8 note).

```
最初だけ
無線のグループを設定 1

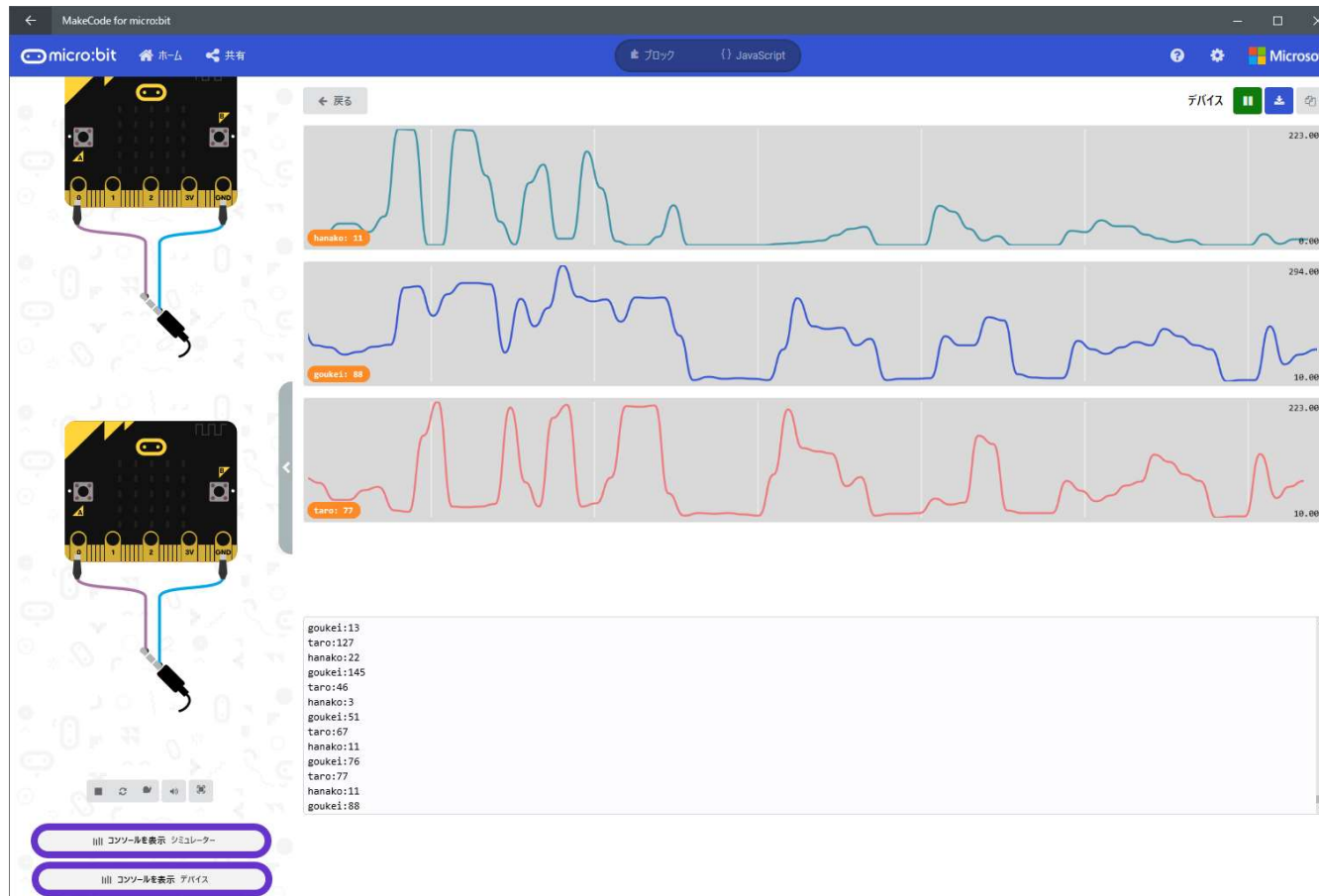
ずっと
変数 goukei を taro + hanako にする
シリアル通信 名前と数値を書き出す "taro" = taro
シリアル通信 名前と数値を書き出す "hanako" = hanako
シリアル通信 名前と数値を書き出す "goukei" = goukei

無線で受信したとき name value
もし name = "taro" なら
  変数 taro を value にする
  反転 x 0 y 0
でなければもし name = "hanako" なら
  変数 hanako を value にする
  反転 x 0 y 1

ずっと
もし goukei < 50 かつ switch = 0 なら
  変数 switch を 1 にする
  無線で送信 "switch" = 1
  音を鳴らす 高さ (Hz) 上のド 長さ 1/8 拍
でなければもし goukei ≥ 50 かつ switch = 1 なら
  変数 switch を 0 にする
  無線で送信 "switch" = 0
  音を鳴らす 高さ (Hz) 真ん中のド 長さ 1/8 拍
```

# スマートシティに挑戦（明るさのグラフ）

- センターで各家の明るさが分かる。
- 明るさの合計を出して全体が暗かったら、まち全体の照明を点ける。



グラフはデスクトップ版のMakeCode(無料)をインストールしてUSB接続のまま使うと表示できます。  
(ブラウザ版ではシミュレーションしかできません。)