

micro:bitによるプログラミング(3)

担当: 稲川 孝司(帝塚山学院大学)

実習内容

- micro:bitによるフルカラーLEDの制御

引用・参考文献

高橋参吉、喜家村奨、稲川孝司: micro:bitで学ぶプログラミング ブロック型からJavaScriptそしてPythonへ、コロナ社、(2019.9).

【NeoPixelについて】

- NeoPixelは、1つのセル毎に赤、緑、青の3つのLEDとその制御回路が入っており、シリアル通信でRGBの値を送ることで、簡単に複数個のLEDをフルカラーで光らせることができるLEDの集合体です。RGBそれぞれを256段階に調整でき、 $256*256*256=16777216$ 色で光ります。
- 今回は、8個のLEDがStick型に接続された製品を使って、micro:bitから様々な色で光らせるプログラミング実習を行います。
- 実物のmicro:bitとNeoPixelがなくてもシミュレータが動くので、**ブラウザのみでも実習は可能です。**

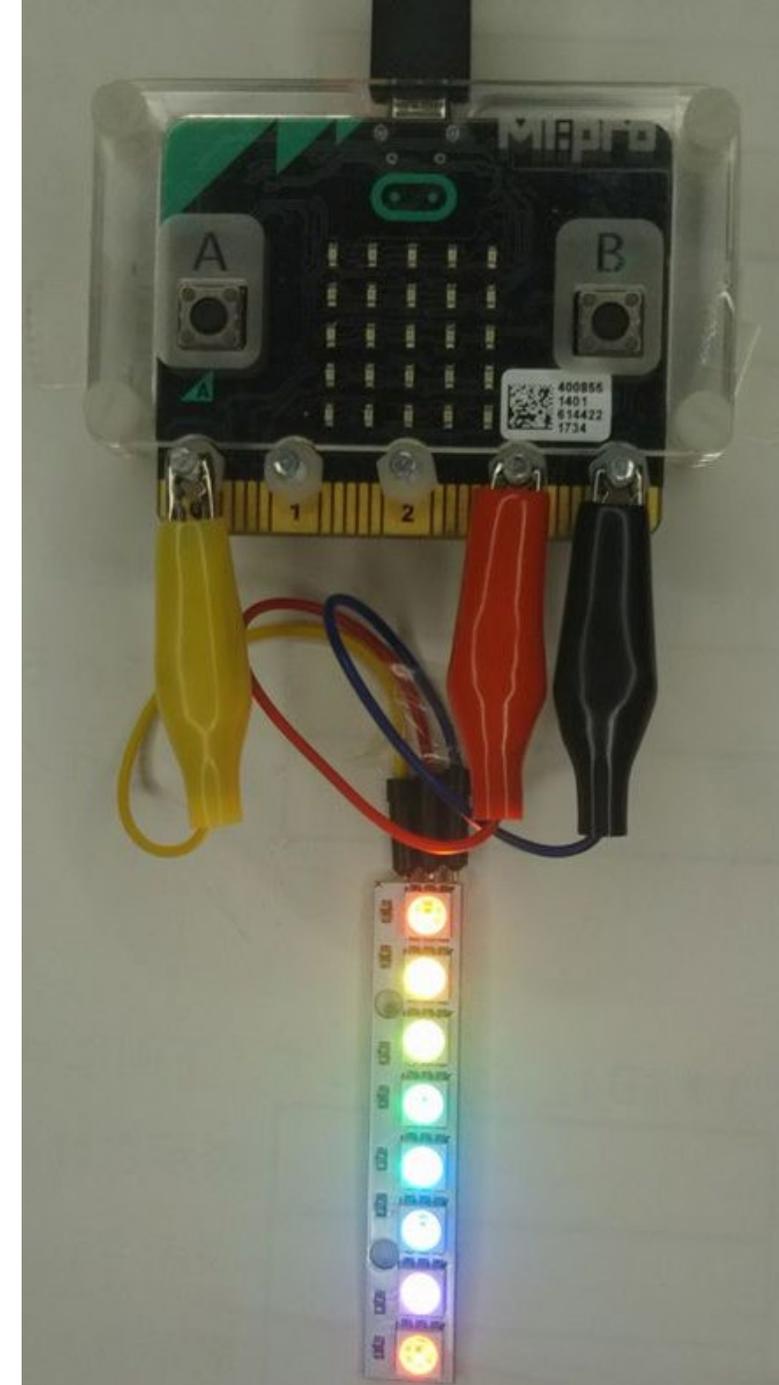
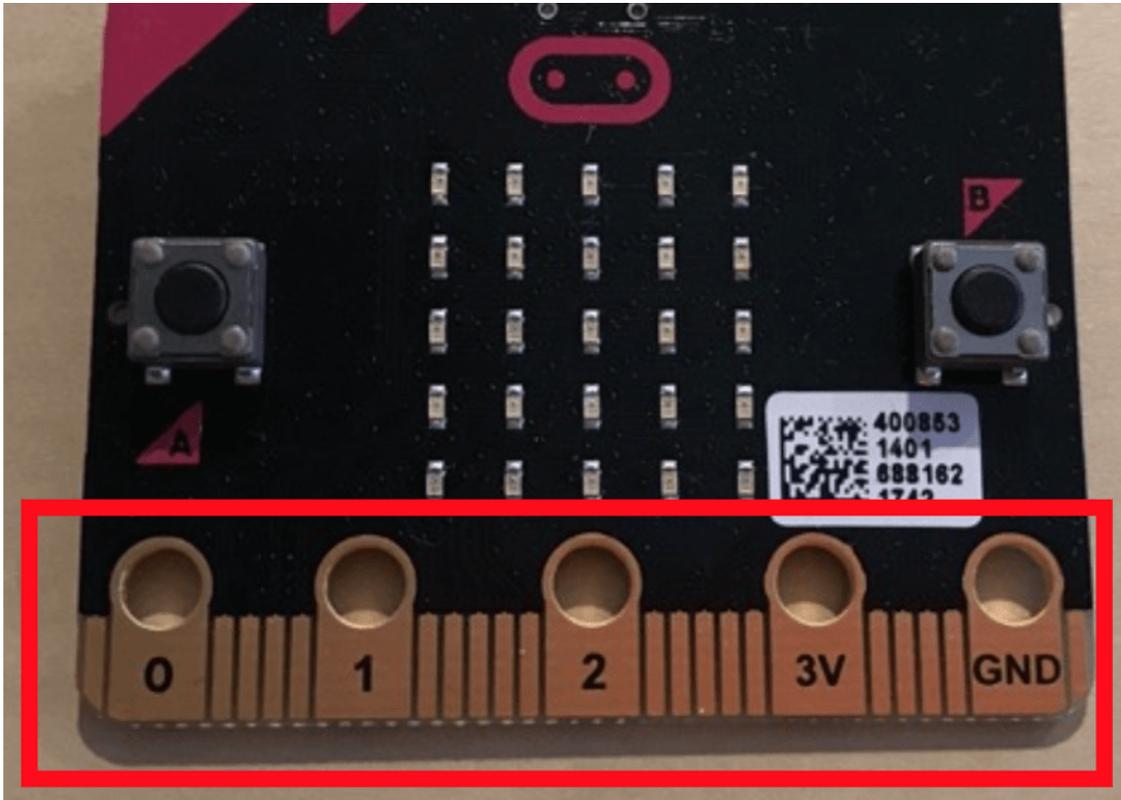


Micro:bitとNeoPixelの接続

Micro:bitとNeoPixelを右図のように接続します。

黄色はP0, 赤は3V, 黒はGND

3V端子から流せる電流は $V1 < 90\text{mA}$ $V2 < 200\text{mA}$



【準備】NeoPixelライブラリーの入手

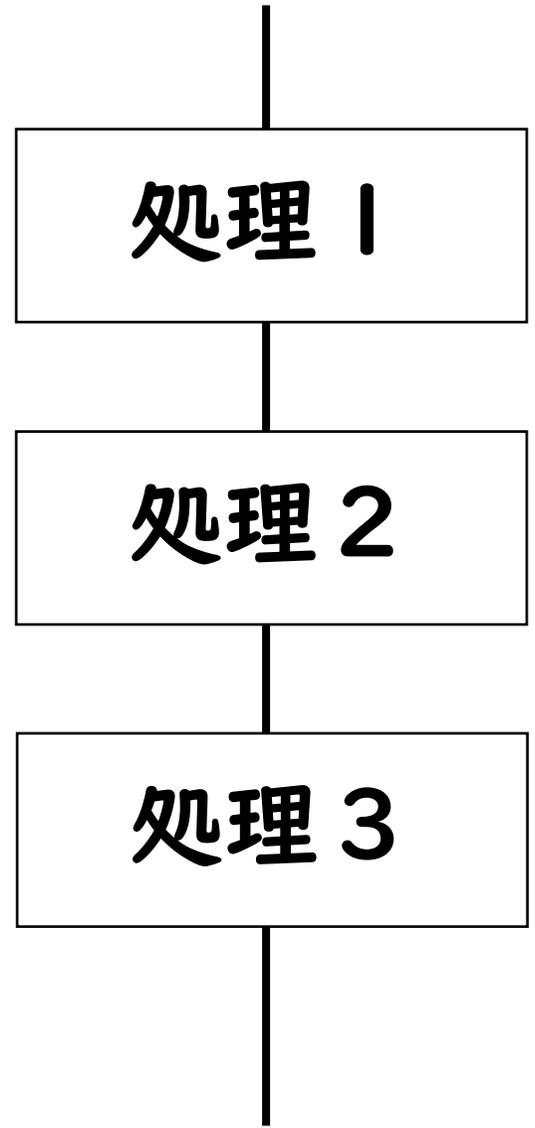
- 1) ブラウザで <https://makecode.microbit.org> サイトに行く。
- 2) 新しいプロジェクト⇒名前を「rei3-1」と入力し、作成ボタンをクリックする。
- 3) ツールボックスの下にある「⊕拡張機能」をクリックし、「neopixel」の**写真をクリック**する。
- 4) ツールボックスの「計算」の下に、水色のNeoPixelのブロック群が追加される。

The image shows a sequence of four screenshots illustrating the steps to add the NeoPixel library in MakeCode Microbit:

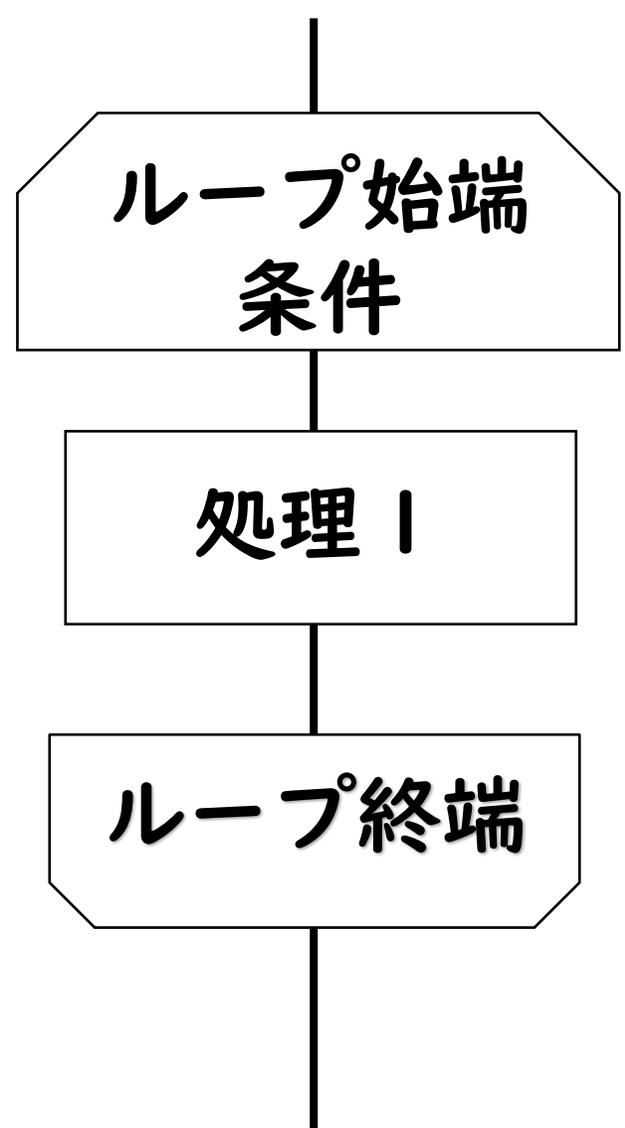
- Step 1:** A purple button with a white plus sign and the text "新しいプロジェクト" (New Project) is shown. A red arrow points to the next screenshot.
- Step 2:** A "プロジェクトを作成する" (Create Project) dialog box is shown. The text "プロジェクトに名前をつけてください。" (Please name your project.) is followed by a text input field containing "rei3-1". A red box highlights the input field. Below it is a green "作成" (Create) button with a checkmark, also highlighted with a red box. A red arrow points to the next screenshot.
- Step 3:** A "拡張機能" (Extensions) menu is shown. A red box highlights the "拡張機能" header. Below it is a grid of extension thumbnails. The "neopixel" thumbnail, which shows a colorful LED strip, is highlighted with a red box. A red arrow points to the next screenshot.
- Step 4:** A toolbar is shown with various categories: "変数" (Variables), "計算" (Math), "Neopixel", and "高度なブロック" (Advanced Blocks). A red box highlights the "Neopixel" category, which is represented by a gear icon. A red arrow points to the next screenshot.

プログラムの3つの基本構造

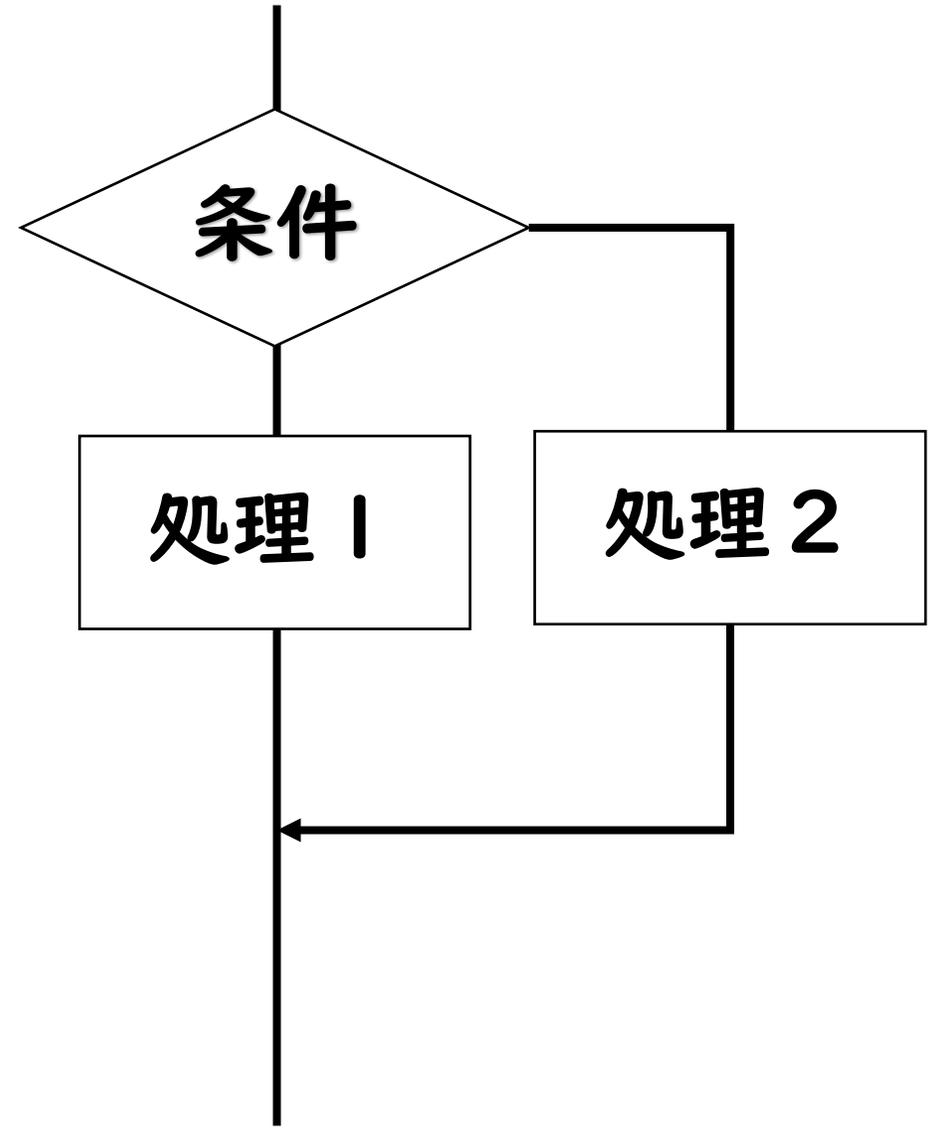
【順次構造】



【反復構造】



【選択構造】



【例題3-1】 NeoPixelを光らせてみよう。(ファイル名:rei3-1)

<手順>

- 1) 新しいプロジェクト⇒名前を「rei3-1」と入力し、作成ボタンをクリックする。
- 2) 「最初だけ」の中に「Neopixel」ブロックから「変数をNeoPixelモードにする」を追加する。
- 3) NeoPixelを接続している端子はP0なので、P0であることを**確認**する。
- 4) NeoPixelの数は8個なので、ブロック内の24を8に**変更**する。
- 5) 「Neopixel」ブロックから「赤色に点灯する」ブロックを「最初だけ」に追加する
- 6) **作成したプログラムをmicro:bitに書き込み**、赤色で点灯することを確認する。

【順次構造】

最初だけ

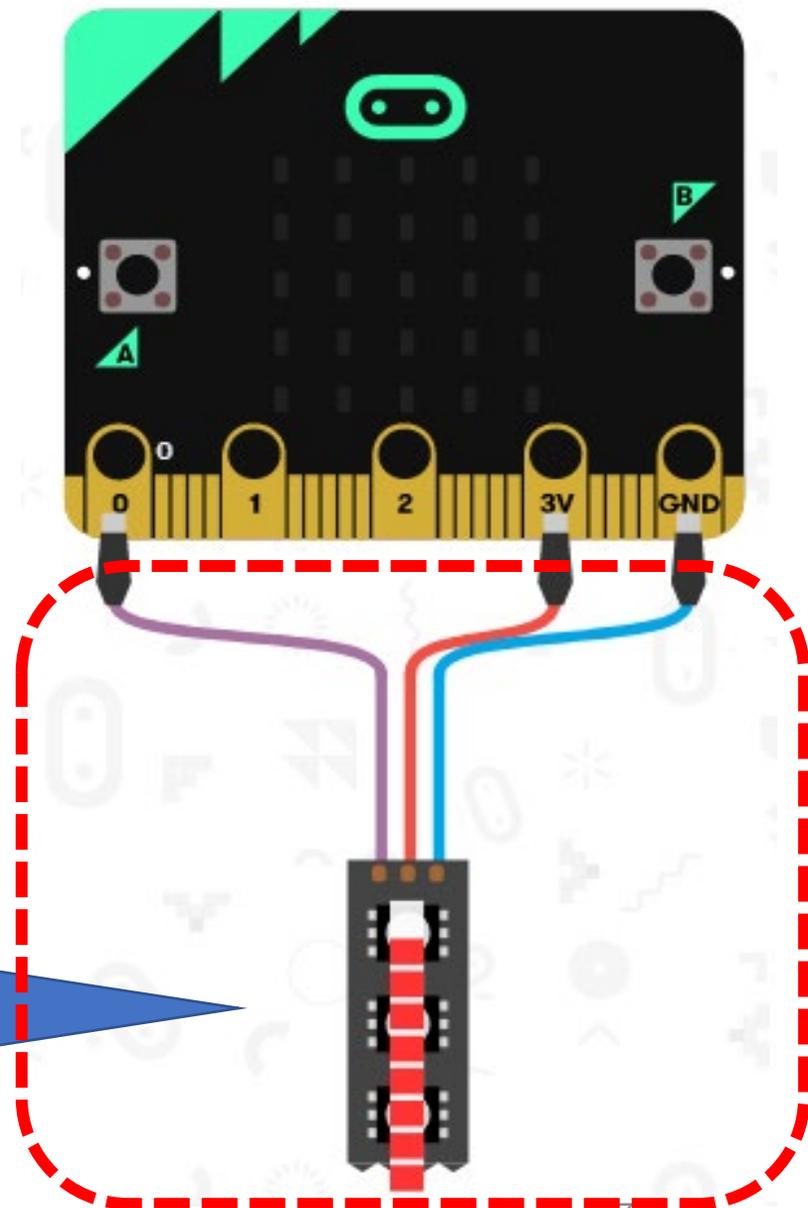
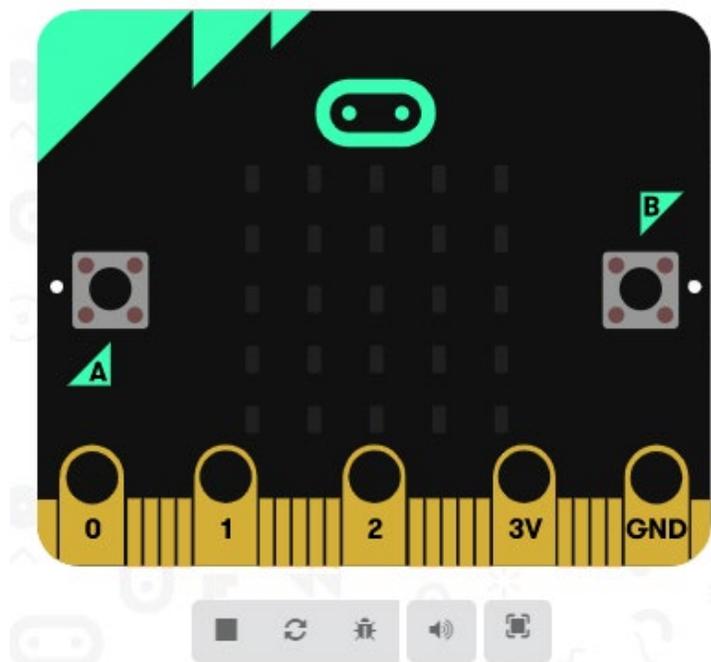
変数 strip を 端子 P0 に接続しているLED **8** 個のNeoPixel (モード RGB (GRB順)) にする

strip を **赤** 色に点灯する

24を8に変更

ここをクリックして、好きな色に変えることができる

シミュレータについて



NeoPixelライブラリを追加したプログラムを作成すると、シミュレータ画面上にNeoPixelが自動的に表示されます。

【例題3-2】 LEDを好きな色で100ミリ秒ごとに点滅させてみよう。 (ファイル名:rei3-2)

<手順>

- 1) ファイル名を「rei3-2」と入力する。
- 2) 「最初だけ」ブロックの中に「変数をNeoPixelモードにする」ブロックを追加する
- 3) NeoPixelを接続している端子はP0なので、P0であることを確認する。
- 4) NeoPixelの数は8個なので、ブロック内の24を8に**変更**する。

- 5) 「ずっと」ブロックの中に「赤色に点灯する」ブロックを追加する。
- 6) 基本から「一時停止（ミリ秒）100」ブロックを追加する
- 7) 「赤色に点灯する」ブロックを追加し、消灯するために**色をblackに変更**する。
- 8) 再度、基本から「一時停止（ミリ秒）100」ブロックを追加する
- 9) **作成したプログラムをmicro:bitに書き込みます。**

【例題3-2】 (ファイル名:rei3-2)

【順次構造】

最初だけ

変数 `strip` を 端子 `P0` に接続しているLED `8` 個のNeoPixel (モード `RGB (GRB順)`) にする

【反復構造】

ずっと

`strip` を `赤` 色に点灯する

一時停止 (ミリ秒) `100`

`strip` を `black` 色に点灯する

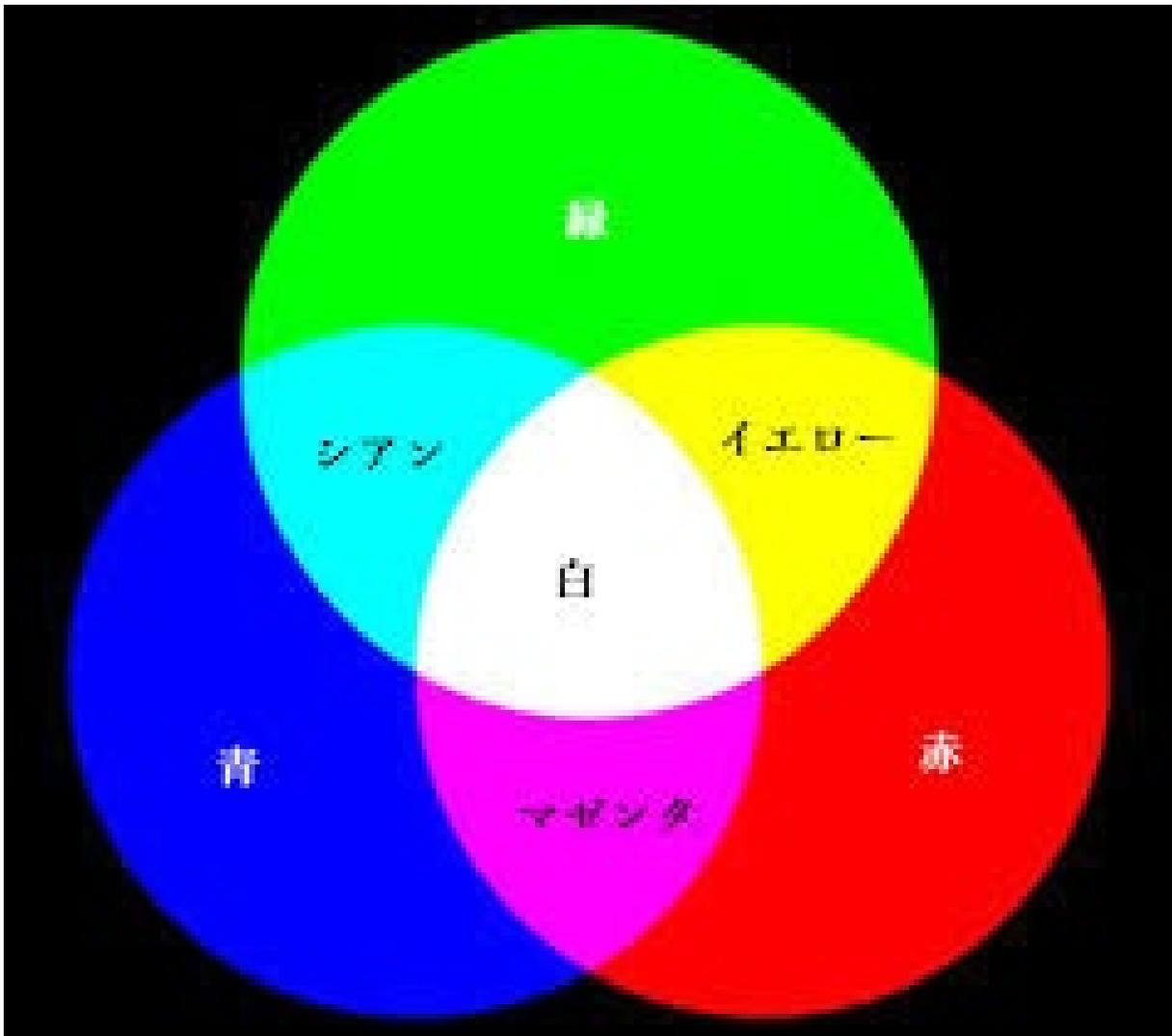
一時停止 (ミリ秒) `100`

Neopixelのその他ブロックから

`RGB (赤 255 緑 255 青 255)`

ブロックを選ぶと、RGBそれぞれを256段階に変えてフルカラーで表示させることができる。

光の3原色 R(赤) G(緑) B(青)



赤 緑 青

$256色 \times 256色 \times 256色 = 16777216色$

2進数	10進数	16進数
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

floralwhite #fffaf0	blue #0000ff	palegreen #98fb98	darkorange #ff8c00	thistle #d8bfd8
linen #faf0e6	dodgerblue #1e90ff	lightgreen #90ee90	goldenrod #daa520	magenta #ff00ff
antiquewhite #faebd7	cornflowerblue #6495ed	springgreen #00ff7f	peru #cd853f	fuchsia #ff00ff
papayawhip #ffe4d5	deepskyblue #00bfff	mediumspringgreen #00fa9a	darkgoldenrod #b8860b	violet #ee82ee
blanchedalmond #ffe4cd	lightskyblue #87cefa	lawngreen #7cfc00	chocolate #d2691e	plum #dda0dd
bisque #ffe4c4	skyblue #87ceeb	chartreuse #7fff00	sienna #a0522d	orchid #da70d6
moccasin #ffe4b5	lightblue #add8e6	greenyellow #adff2f	saddlebrown #8b4513	mediumorchid #ba55d3
navajowhite #ffdead	powderblue #b0e0e6	lime #00ff00	maroon #800000	darkorchid #9932cc
peachpuff #ffdab9	paleturquoise #afeeee	limegreen #32cd32	darkred #8b0000	darkviolet #9400d3
mistyrose #ffe4e1	lightcyan #e0ffff	yellowgreen #9acd32	brown #a52a2a	darkmagenta #8b008b
lavenderblush #fff0f5	cyan #00ffff	darkolivegreen #556b2f	firebrick #b22222	purple #800080

micro:bitでの16進数から10進数への自動変換】

(例)

skyblue
#87ceeb

16進数であることを示す
0xを先頭に追加する

RGB (赤 0x87 緑 0xce 青 0xeb)

自動的に10進数になる

RGB (赤 135 緑 206 青 235)

【コラム：LEDの番号の割り当てについて】

NeoPixelは、順番にLEDを点灯させるために、それぞれに番号が割り当てられています。8個のStick型の場合は、右図のように、信号線に近いほうから順に0から7が割り当てられます。

個別に色を設定して点灯する方法は、

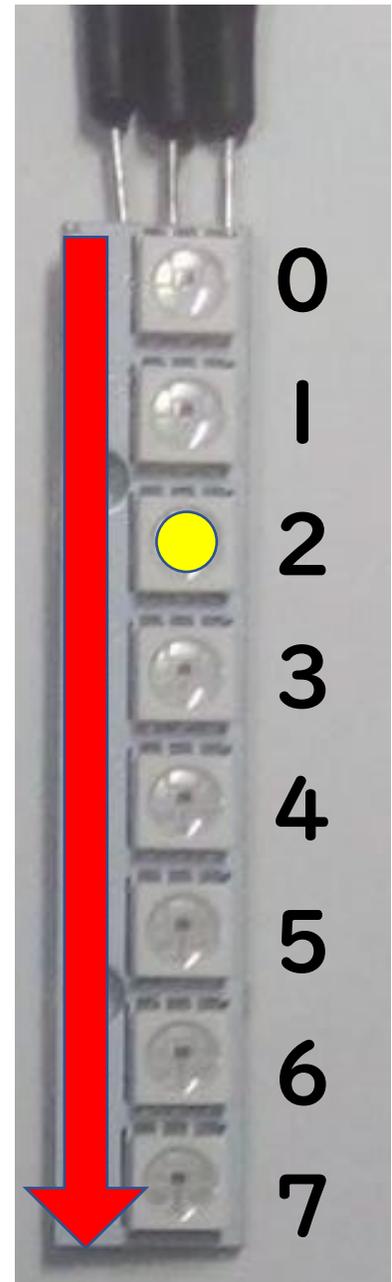
strip ▾ の 2 番目のLEDを 黄 ▾ 色に設定する

で番号を指定し、n番目のLEDを好きな色で点灯させることができます。

また、

strip ▾ に設定されている色をLED 1 個分ずらす

で、点灯しているLEDを下にずらすことができます。



【例題3-3】 8個のLEDを個別に好きな色に指定して、点灯させてみよう。 (ファイル名:rei3-3)

<手順>

- 1) ファイル名を「rei3-3」と入力する。
- 2) 「最初だけ」ブロックの中に「変数をNeoPixelモードにする」ブロックを追加する。
- 3) 今回のNeoPixelの数は8個なので、ブロック内の24を8に変更する。

- 4) 「ずっと」ブロックの中に、NeoPixelの「その他」ブロックから「stripの0番目のLEDを赤色に設定する」ブロックを追加する。
- 5) 「設定」ブロックを右クリックして複製し、計8個の「設定」ブロックを作成。
- 6) 設定する番号を0から7の順に変更し、色は好きな色に変更する。
- 7) 「stripを設定した色で点灯する」ブロックを追加する。
- 8) 作成したプログラムをmicro:bitに書き込みます。

【例3-3】 (rei3-3)

最初だけ

変数 strip を 端子 P8 に接続しているLED 8 個のNeoPixel (モード RGB (GRB順)) にする

【反復構造】

ずっと

strip の 0 番目のLEDを 赤 色に設定する

strip の 1 番目のLEDを だいだい 色に設定する

strip の 2 番目のLEDを 黄 色に設定する

strip の 3 番目のLEDを 緑 色に設定する

strip の 4 番目のLEDを 青 色に設定する

strip の 5 番目のLEDを あい 色に設定する

strip の 6 番目のLEDを すみれ 色に設定する

strip の 7 番目のLEDを 紫 色に設定する

strip を設定した色で点灯する

【行のコピペの方法】
ブロックの1行目を右
クリックして複製し、
貼り付け、数字と色を
変える

【例題3-4】 「1個分ずらす」ブロックを使って、点灯した緑のLEDを上から下へ移動させてみよう。 （ファイル名： rei3-4）

<手順>

- 1) ファイル名を「rei3-4」と入力します。
- 2) 「最初だけ」中に「変数をNeoPixelモードにする」ブロックを追加する。
- 3) 今回のNeoPixelの数は8個なので、ブロック内の24を8に変更します。

- 4) 「ずっと」の中に、NeoPixelの「その他」ブロックから「0番目のLEDの色を設定する」ブロックを追加する。
- 5) ループから「くりかえし」ブロックを入れ、回数を8回に設定します。
- 6) 「くりかえし」の中に、NeoPixelから「指定した色で点灯する」ブロックを追加する。
- 7) さらに、基本から「一時停止(ミリ秒)100」ブロックを追加する。
- 8) そのブロックの下に、NeoPixelから「色をLED1個分ずらす」ブロックを追加する。
- 9) 作成したプログラムをmicro:bitに書き込みます。

【例題3-4】 (rei3-4)

最初だけ

変数 `strip` を 端子 `P0` に接続しているLED **8** 個のNeoPixel (モード `RGB (GRB順)`) にする

【反復構造】

ずっと

`strip` の `0` 番目のLEDを **緑** 色に設定する

くりかえし **8** 回

`strip` を設定した色で点灯する

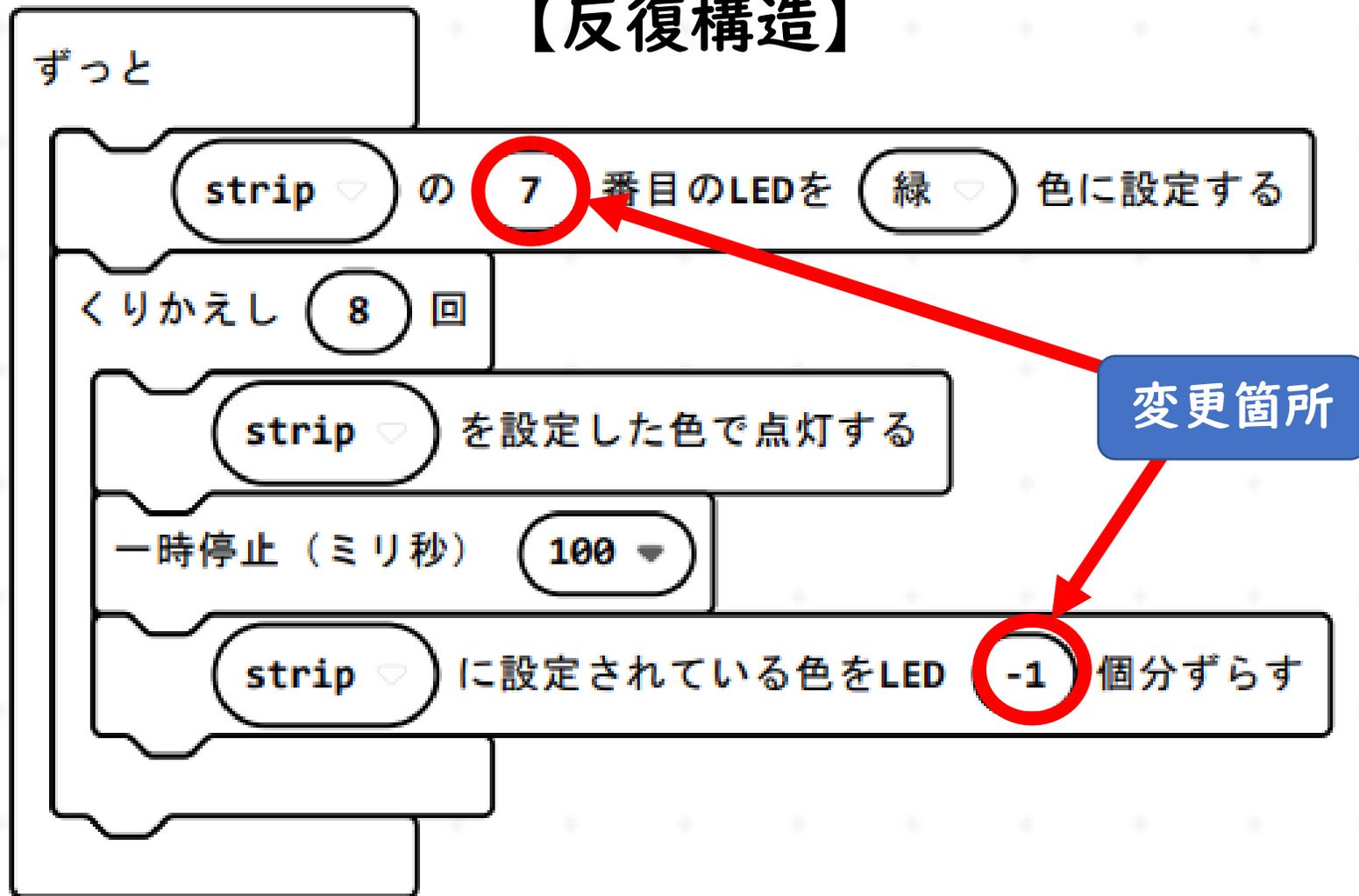
一時停止 (ミリ秒) `100`

`strip` に設定されている色をLED **1** 個分ずらす



【例題3-5】 1つだけLEDを点灯させて、下から上へ移動させてみよう。
(7番目から順に0番目までの点灯するために、例題3-4の次の2か所を修正します。
※最初だけブロックは例題3-4と同じなので省略している。) (rei3-5)

【反復構造】



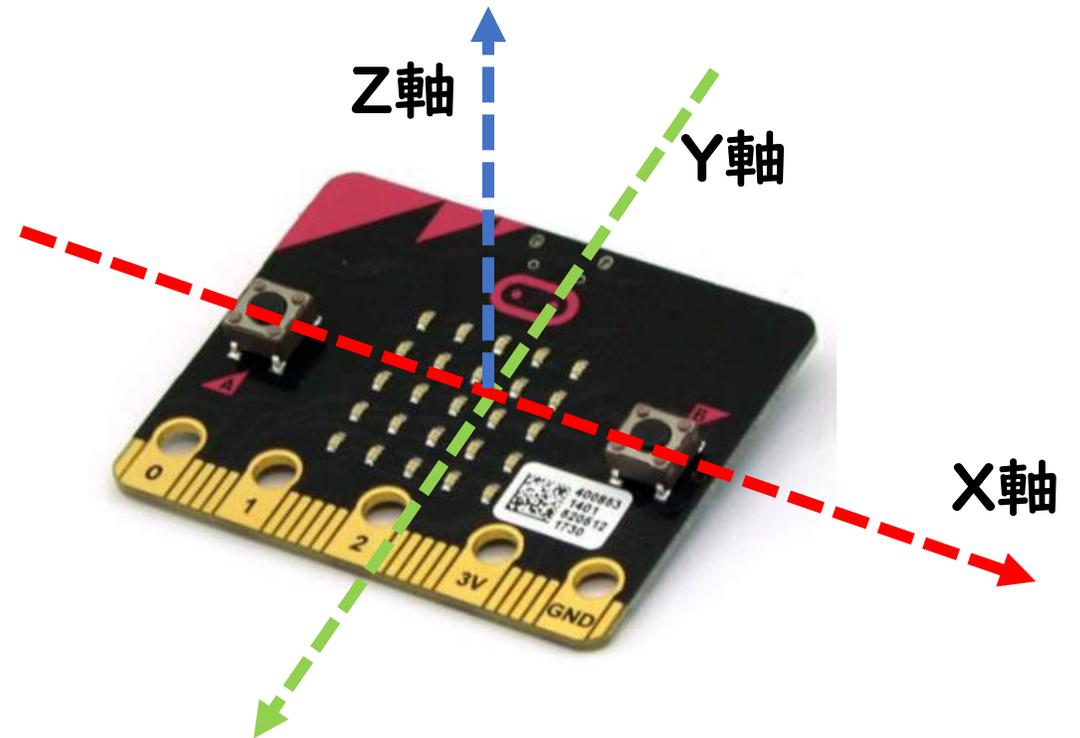
**【応用1】 例題3-4と例題3-5を組み合わせ、
LEDを上下に移動させるプログラムを作成してみよう。**

【コラム：加速度センサについて】

micro:bitには、3次元の加速度センサが搭載されており、図のようにx, y, zの3方向の加速度を計測することができます。

地上では下向きの重力加速度があるので、それを計測してmicro:bitを3次元的に動かすだけで値を変化させることができます。

また、動きや振動、衝撃なども検知することができます。



【例題3-6】 micro:bitを傾けて、搭載されている加速度センサから3つの値を読み込んでRGBの3つの色に対応したLEDを点灯させ、micro:bitを動かして色を変えてみよう。 (ファイル名:rei3-6)

<手順>

- 1) 「最初だけ」ブロックの中に「変数をNeoPixelモードにする」ブロックを追加する。

- 2) ずっとブロックの中に、NeoPixelのその他から「stripの0番目を赤色にする」ブロックを追加する。
- 3) 赤の部分を、NeoPixelのその他から「RGB(赤255、緑255、青255)」に置き換える。
- 4) 数値の255の部分を加速度X、加速度Y,加速度Zに置き換える。
- 5) Neopixelから「Stripを指定した色で点灯する」ブロックを追加する。
- 6) 基本から一時停止(ミリ秒) 100を追加する。
- 7) NeoPixelから「stripに設定されている色を1個分ずらす」ブロックを追加する。
- 8) **作成したプログラムをmicro:bitに書き込みます。**

【例題3-6】 (ファイル名:rei3-6)

最初だけ

変数 `strip` を 端子 `P0` に接続しているLED 8 個のNeoPixel (モード `RGB (GRB順)`) にする

【反復構造】

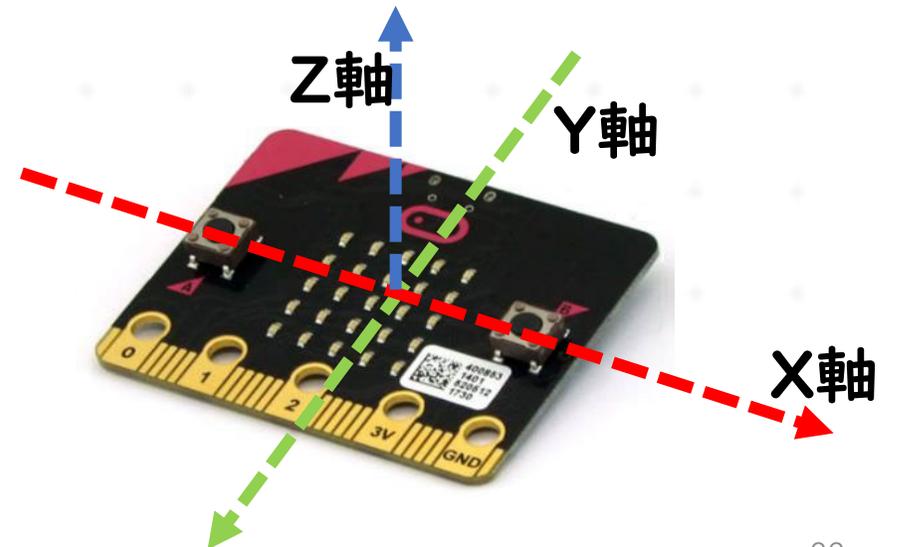
ずっと

`strip` の 0 番目のLEDを RGB (赤 `加速度 X` 緑 `加速度 Y` 青 `加速度 Z`) 色に設定する

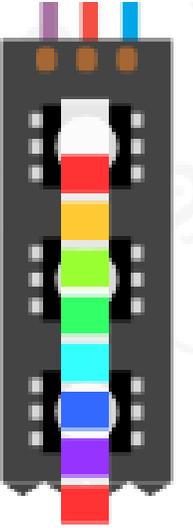
`strip` を設定した色で点灯する

一時停止 (ミリ秒) 100

`strip` に設定されている色をLED 1 個分ずらす



【例題3-7】 レインボーパターンで光らせてみよう。
NeoPixelのライブラリの中に「レインボーパターンに点灯する」という命令があり、それを使うと、虹色で光らせることができます。
(ファイル名:rei3-7)



<手順>

- 1) 「最初だけ」ブロックの中に、「変数をNeoPixelモードにする」ブロックを追加する。
- 2) 今回のNeoPixelの数は8個なので、ブロック内の24を8に変更する。
- 3) 次に「Stripをレインボーパターンに点灯する」ブロックを追加する。
- 4) **作成したプログラムをmicro:bitに書き込みます。**

最初だけ

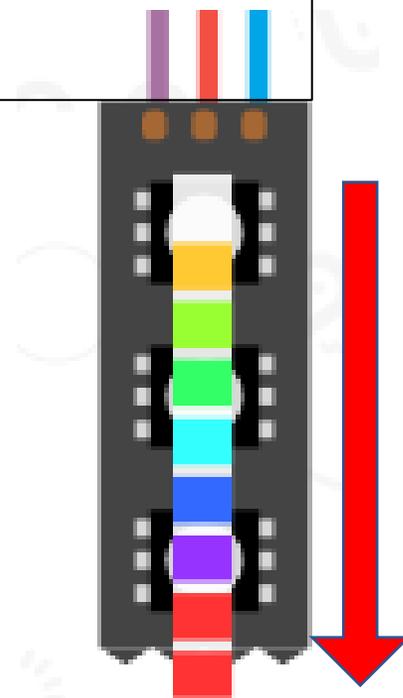
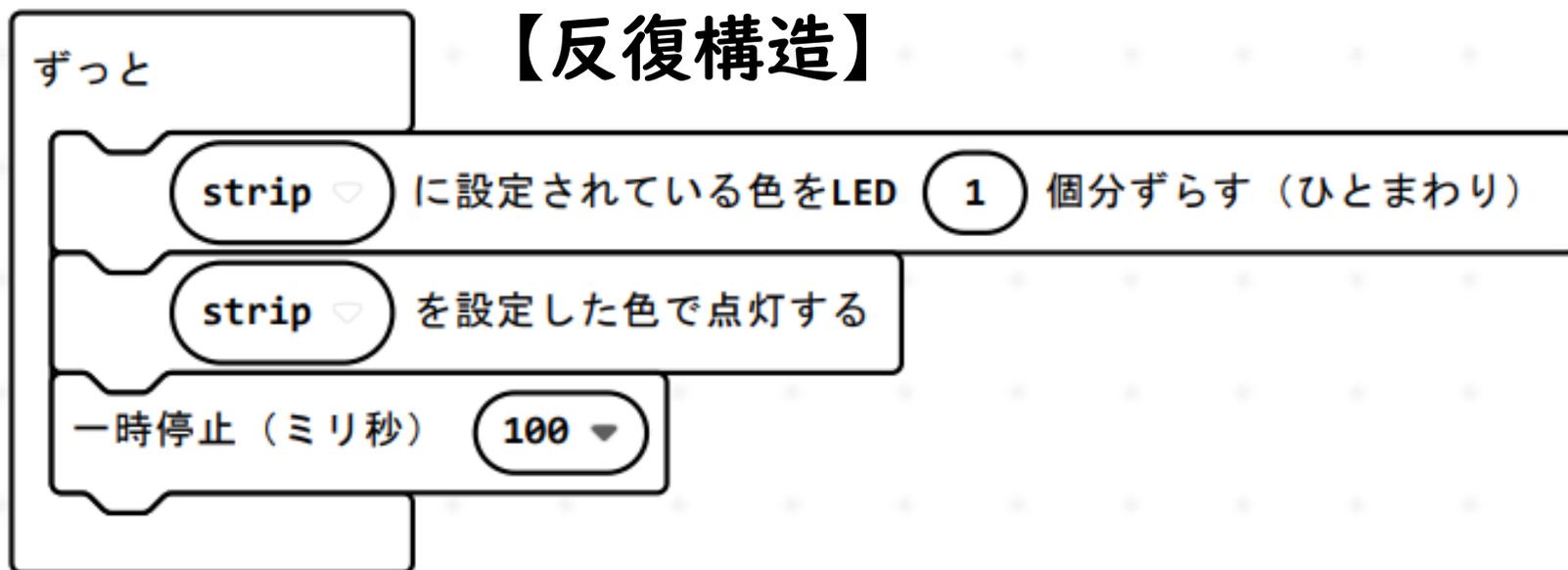
変数 **strip** を 端子 **P0** に接続しているLED **8** 個のNeoPixel (モード **RGB (GRB順)**) にする

strip をレインボーパターン (色相 **1** から **360**) に点灯する

【例題3-8】 レインボーパターンで下移動させよう。
例題3-7ではレインボーパターンの色が固定されていたので、「ずっと」ブロックの中に「1個分ずらす」ブロックを追加して、色が順番に下移動するようにしてみよう。（ファイル名:rei3-8）

<手順>

- 1) 「最初だけ」のブロックは、例題3-7をそのまま使います。
- 2) 「ずっと」ブロックの中に、Neopixelから「LED1個分ずらす（ひとまわり）」ブロックと「設定した色で点灯する」ブロックを追加する。
- 3) 基本から「一時停止（ミリ秒）100」ブロックを追加する。
- 4) 作成したプログラムをmicro:bitに書き込みます。



【例題3-9】 レインボーパターンで上移動させよう。
例題3-8ではレインボーパターンで下移動したので、ここでは上移動する
ようにしてみよう。 (ファイル名:rei3-9)

<手順>

- 1) ブロックは、例題3-8を**基本的にそのまま使います**。
- 2) 「LED 1個分ずらす (ひとまわり)」ブロックの**1**を「**-1**」に**変更**します。

最初だけ

変数 strip を 端子 P0 に接続しているLED 8 個のNeoPixel (モード RGB (GRB順)) にすず

strip をレインボーパターン (色相 1 から 360) に点灯する

ずっと

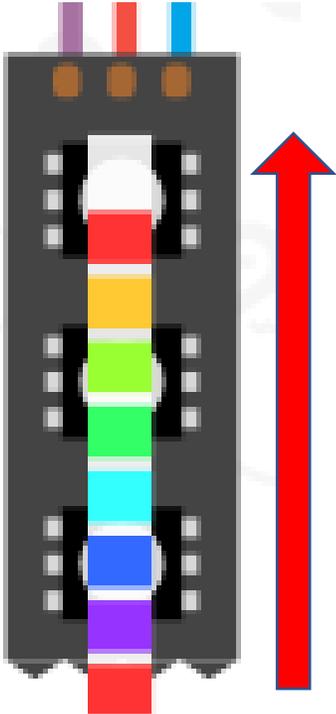
strip に設定されている色をLED **-1** 個分ずらす (ひとまわり)

strip を設定した色で点灯する

一時停止 (ミリ秒) 100

【反復構造】

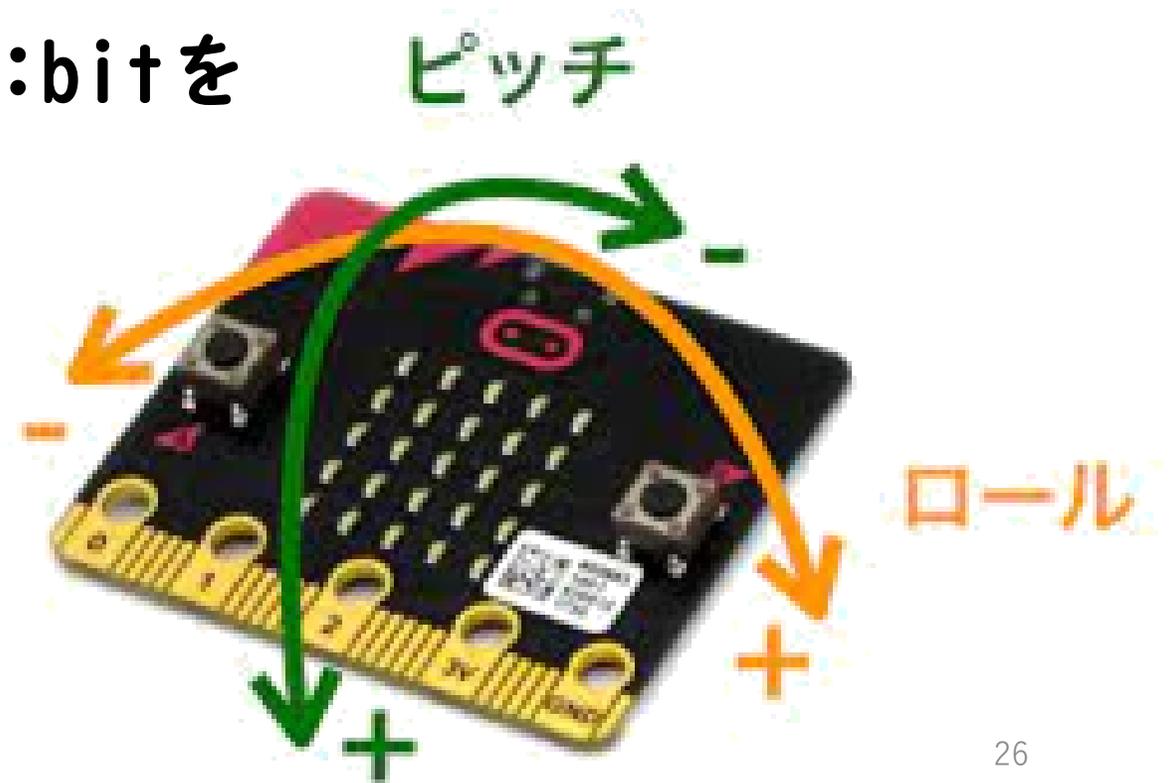
変更箇所



【コラム：ロールとピッチ】

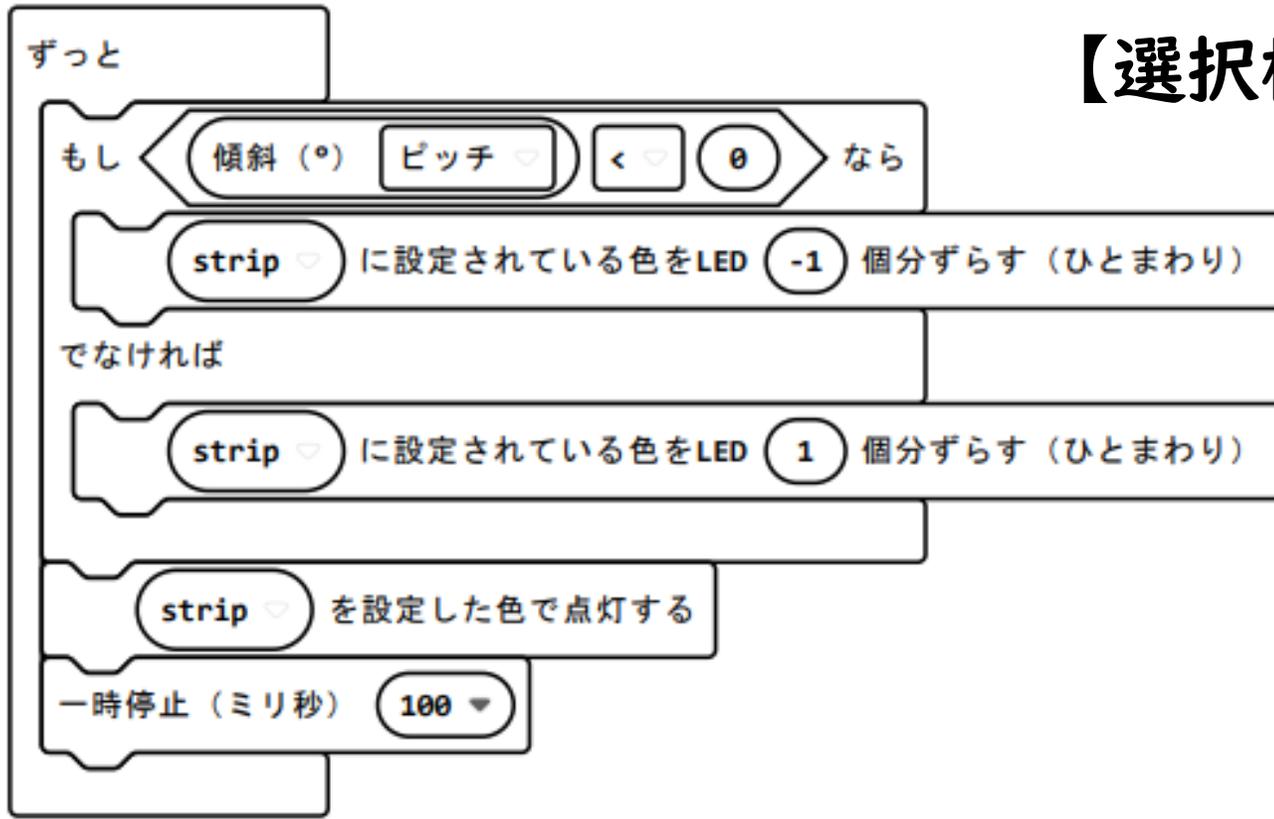
micro:bitには、3次元の加速度センサが搭載されており、右下のようにロールとピッチの角度を調べることができます。

ここでは、ピッチを使ってmicro:bitを前に傾けたときに、LEDが上移動
後ろに傾けたときにLEDが下移動
するプログラムを考えます。

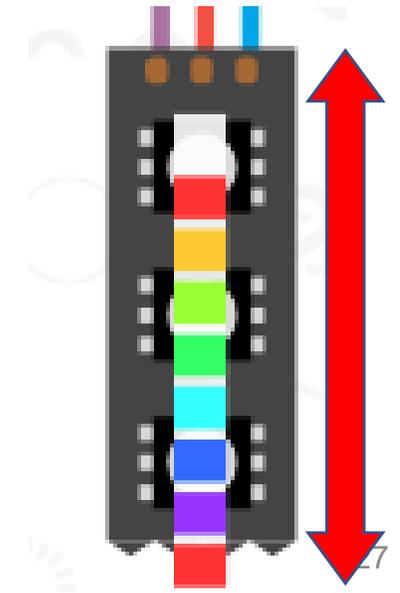


【例題3-10】 MICRO:BITを前後に傾けて、前に傾けた時はレインボーパターンが上移動し、後ろに傾けた時はレインボーパターンが下移動するプログラムを作成しよう。（ファイル名:rei3-10）

傾斜「ピッチ」の値は、MICRO:BITを前に倒した時は負の値になり、後ろに倒した時には正の値になるので、例題3-9のプログラムを次のように変更します。（最初だけ、の部分は例題3-9と同じなので省略）



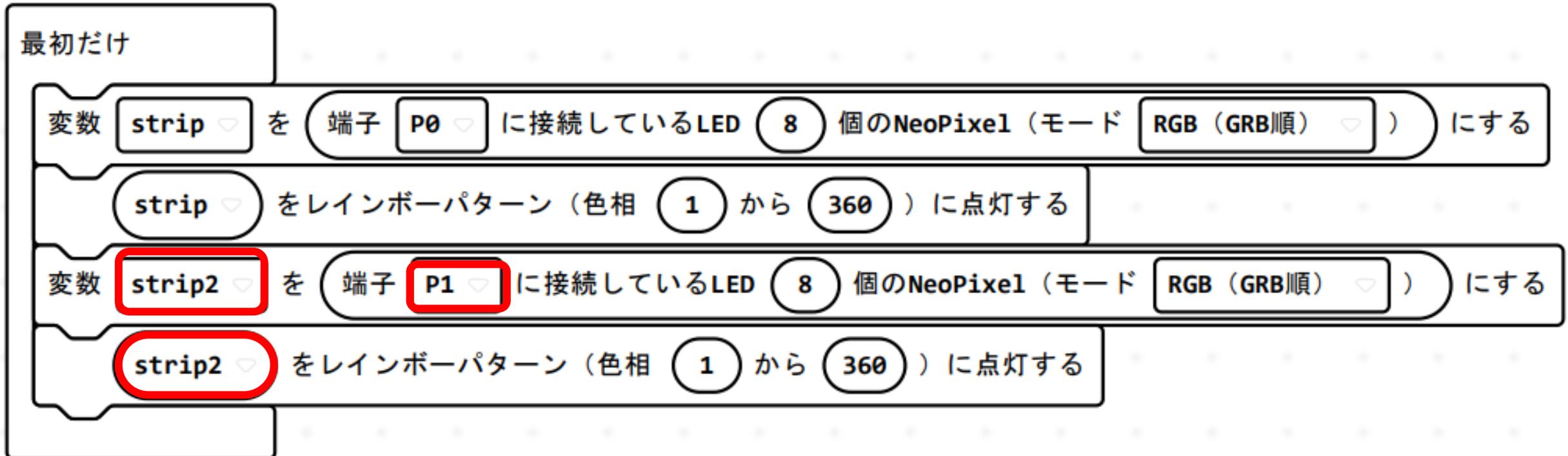
【選択構造】



【例題3-11】 2つのNeopixelを使って、レインボーパターンで、1つは下移動、もう1つは上移動で点灯させてみよう。（ファイル名:rei3-11）
（1つ目はP0端子に、2つ目はP1端子に接続するものとする）

<手順1>

「変数」から「変数を追加する」で「strip2」を追加し、最初だけのブロックのNeoPixelの2つの命令をコピーして、「strip2」で接続端子をP1にしたブロックを作成する。



The image shows a Scratch code editor with a '最初だけ' (Initially) loop block containing four code blocks. The first block sets up a variable 'strip' connected to terminal 'P0' with 8 NeoPixels in RGB (GRB) mode. The second block sets 'strip' to a rainbow pattern from 1 to 360 degrees. The third block sets up a variable 'strip2' connected to terminal 'P1' with 8 NeoPixels in RGB (GRB) mode. The fourth block sets 'strip2' to a rainbow pattern from 1 to 360 degrees. Red boxes highlight 'strip2' and 'P1' in the third and fourth blocks.

```
最初だけ
変数 strip を 端子 P0 に接続しているLED 8 個のNeoPixel (モード RGB (GRB順)) にする
strip をレインボーパターン (色相 1 から 360) に点灯する
変数 strip2 を 端子 P1 に接続しているLED 8 個のNeoPixel (モード RGB (GRB順)) にする
strip2 をレインボーパターン (色相 1 から 360) に点灯する
```

<手順2>

右図のように2つのNeopixelが表示されるので、ずっとブロックの中に、stripは下移動(1)を、strip2は上移動(-1)を設定する。

(ファイル名:rei3-11)

【反復構造】

ずっと

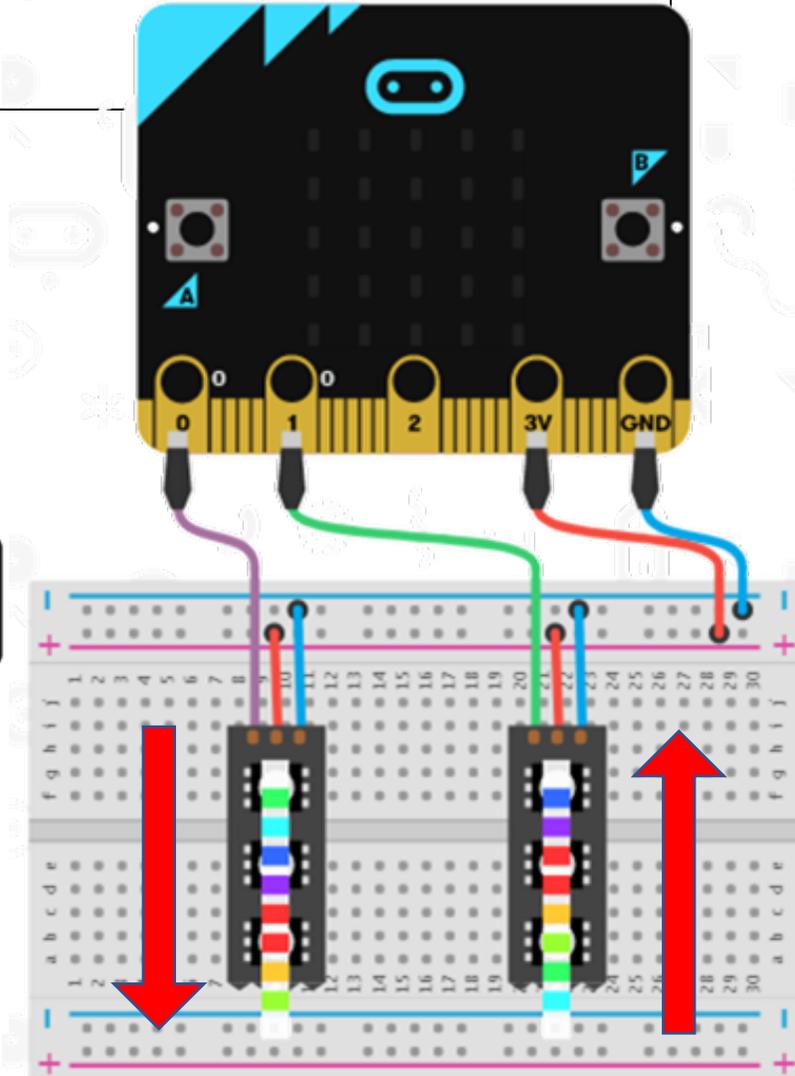
strip ▾ に設定されている色をLED 1 個分ずらす (ひとまわり)

strip ▾ を設定した色で点灯する

strip2 ▾ に設定されている色をLED -1 個分ずらす (ひとまわり)

strip2 ▾ を設定した色で点灯する

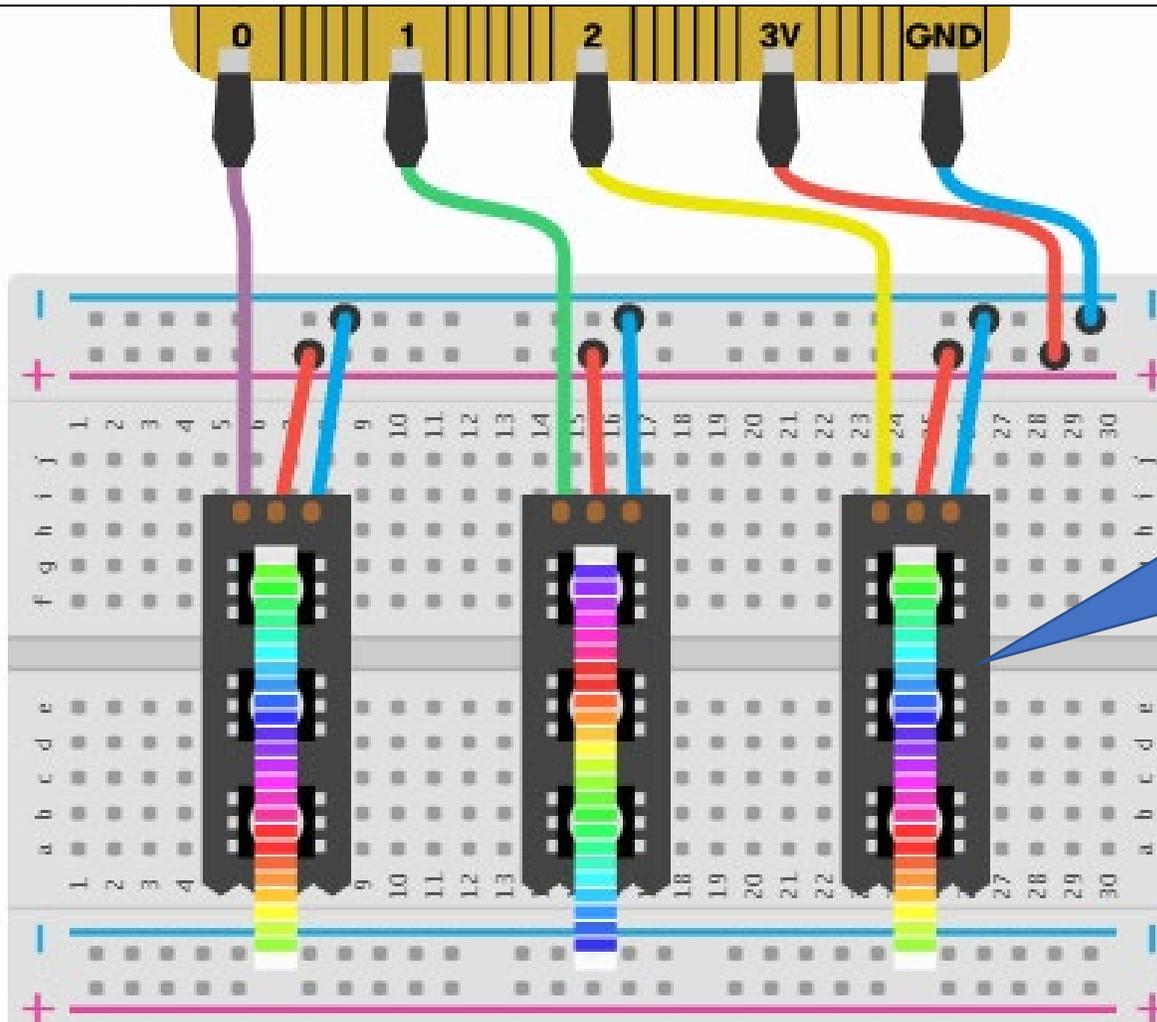
一時停止 (ミリ秒) 100 ▾



【シミュレータによるNeopixel動作】

【応用1】変数をもう1つ追加して、3つのNeopixelを動かしてみよう

【応用2】16個や24個のLEDを光らせてみよう



画面に、本物そっくりな
シミュレータが表示され、
動作する