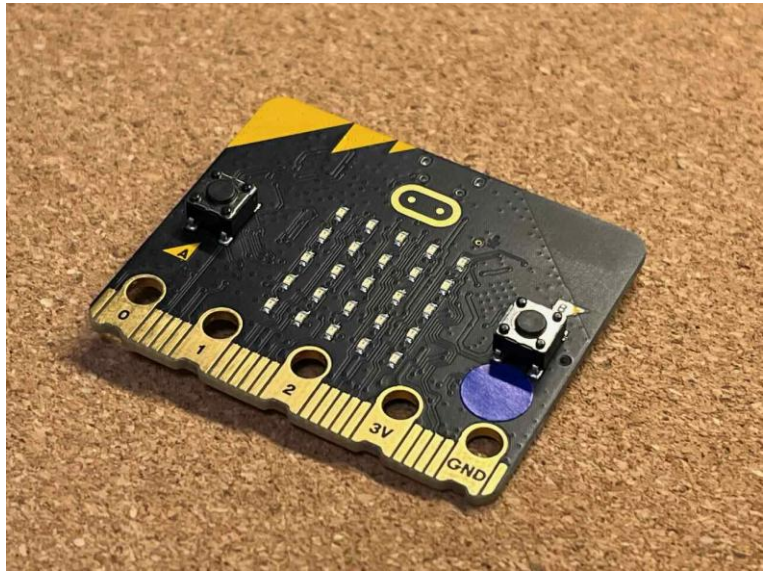
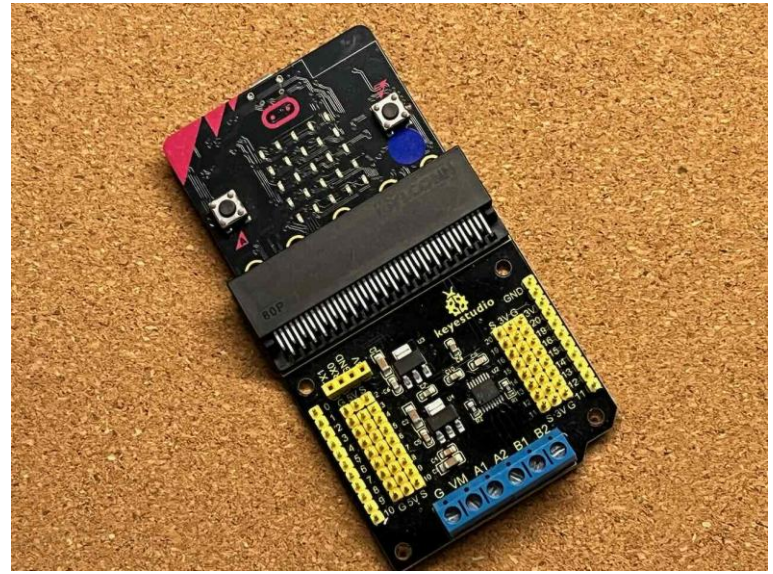
さとやまプログラミングクラブ2026

マイクロビットプログラミング

マイクロビットで「イルミネーショングラス」をつくろう！





今日やること

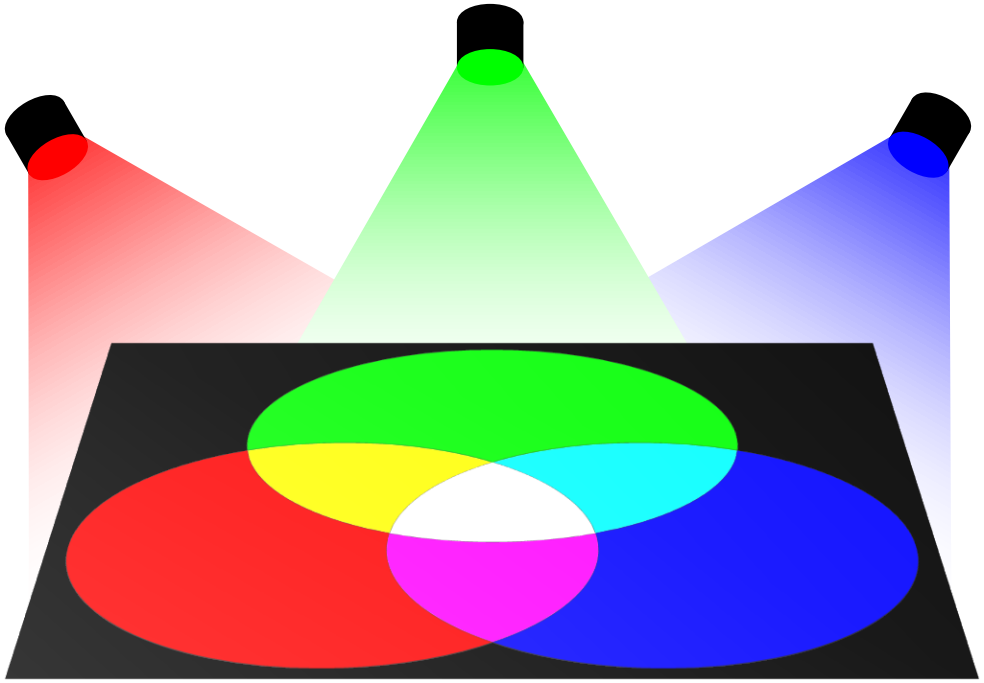
- 「光の3原色」と「フルカラーLED」についてまなびます。
- マイクロビットのプログラムで、フルカラーLEDを光らせてみます。
- イルミネーショングラスをつくります。

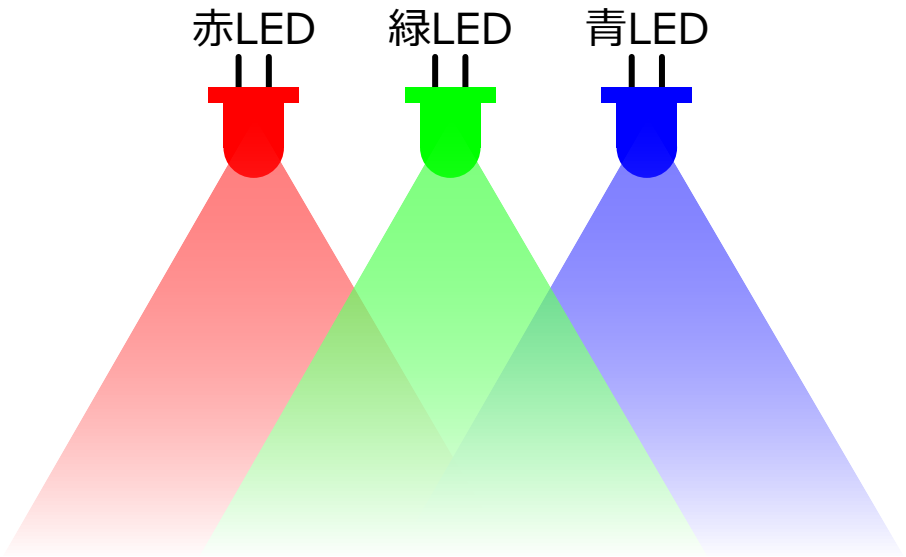
部品の確認

部品	個数
フルカラーLED	1
ガラスコップ	1
インテリアストーン	2
ボタン基板	1
ジャンパーワイヤ（メス-メス）	10

光の3原色について

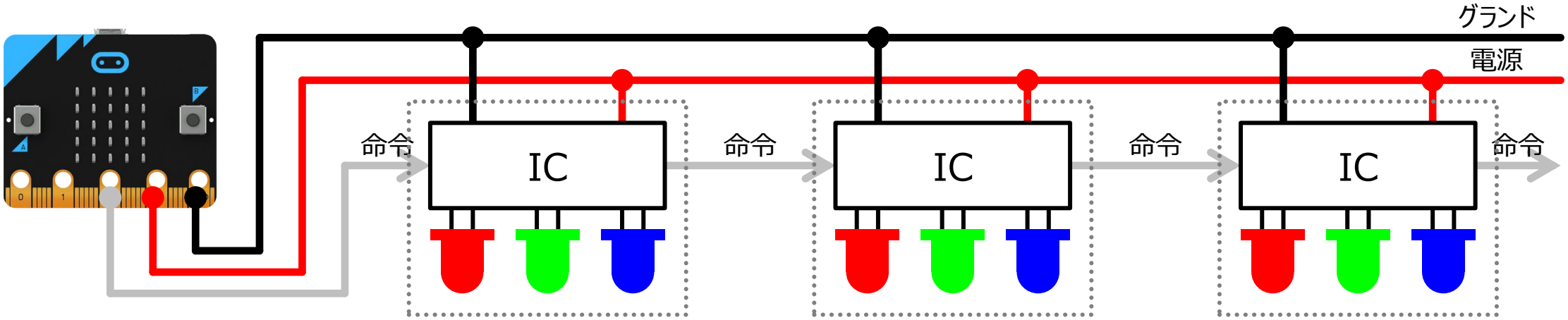
- 光の3原色は「赤（Red）」「緑（Green）」「青（Blue）」です。
- この3つの光をまぜると、さまざまな色がつくれます。
- この3色のLEDのあかるさを調整することで、さまざまな色で光るライトになります。





フルカラーLEDについて

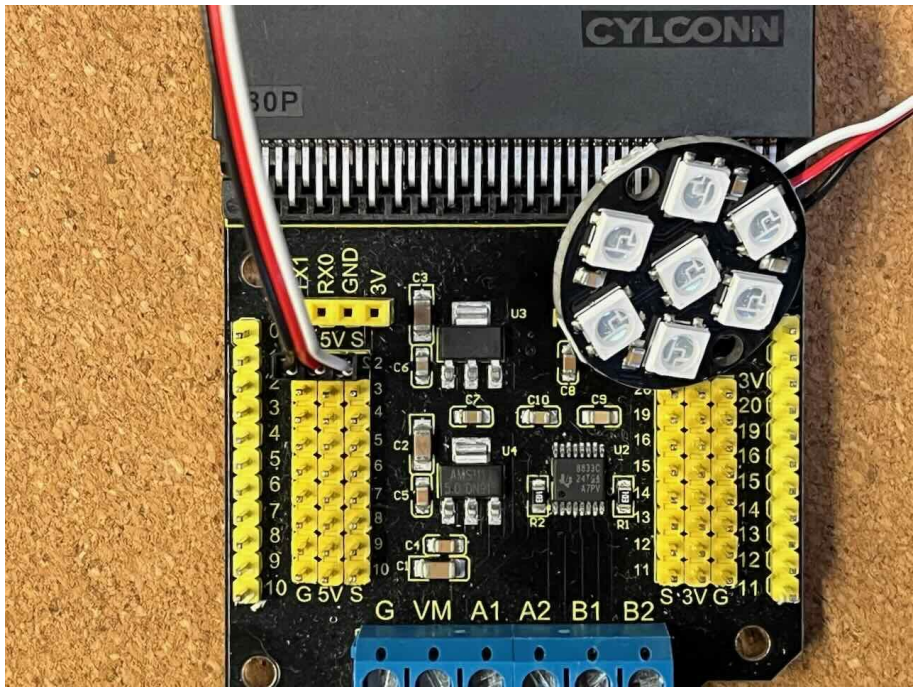
- ひとつのフルカラーLEDの中には、「赤」「緑」「青」の3つのLEDが入っています。
- 外からの命令で、それぞれのLEDのあかるさを調整する部品（IC）も入っています。
  - 命令によって、さまざまな色で光ることができます。
- 1本の信号で命令することができます。
- たくさんのフルカラーLEDをつなげることができます。
  - うけとった命令を、となりのフルカラーLEDにわたすことができます。



拡張ボードと部品の接続

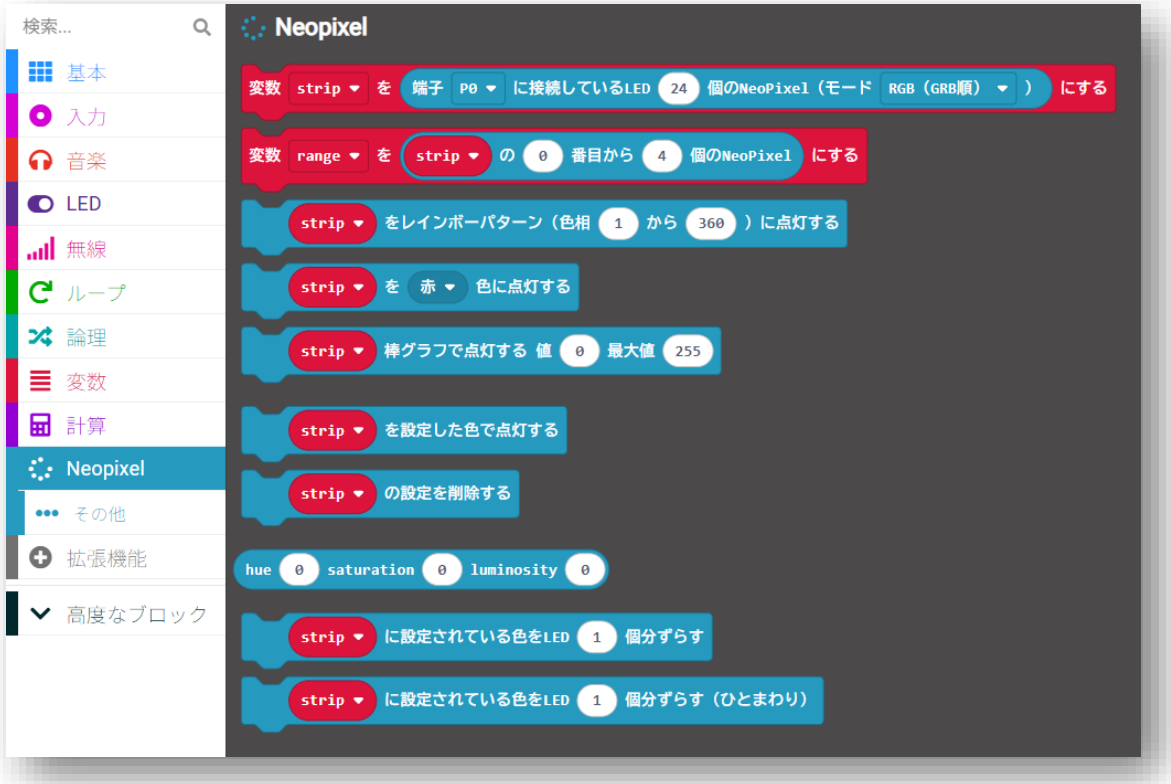
- 「拡張ボード」と「電池ボックス」がつながっていることを確認してください。
- 「拡張ボード」と「フルカラーLED」を右のようにつないでください。
- 今回つかう「LEDリング」には、7個のフルカラーLEDがついています。マイクロビットのプログラムでそれぞれのLEDをちがう色でひからせることができます。
  - プログラムでは「フルカラーLEDの数」や「何番目のLEDをどのように光らせるか？」を書きます。
  - 白のコード1本だけで、7個のLEDに命令を送ることができます。

フルカラーLED	拡張ボード
黒のコード	G
赤のコード	5V
白のコード	2



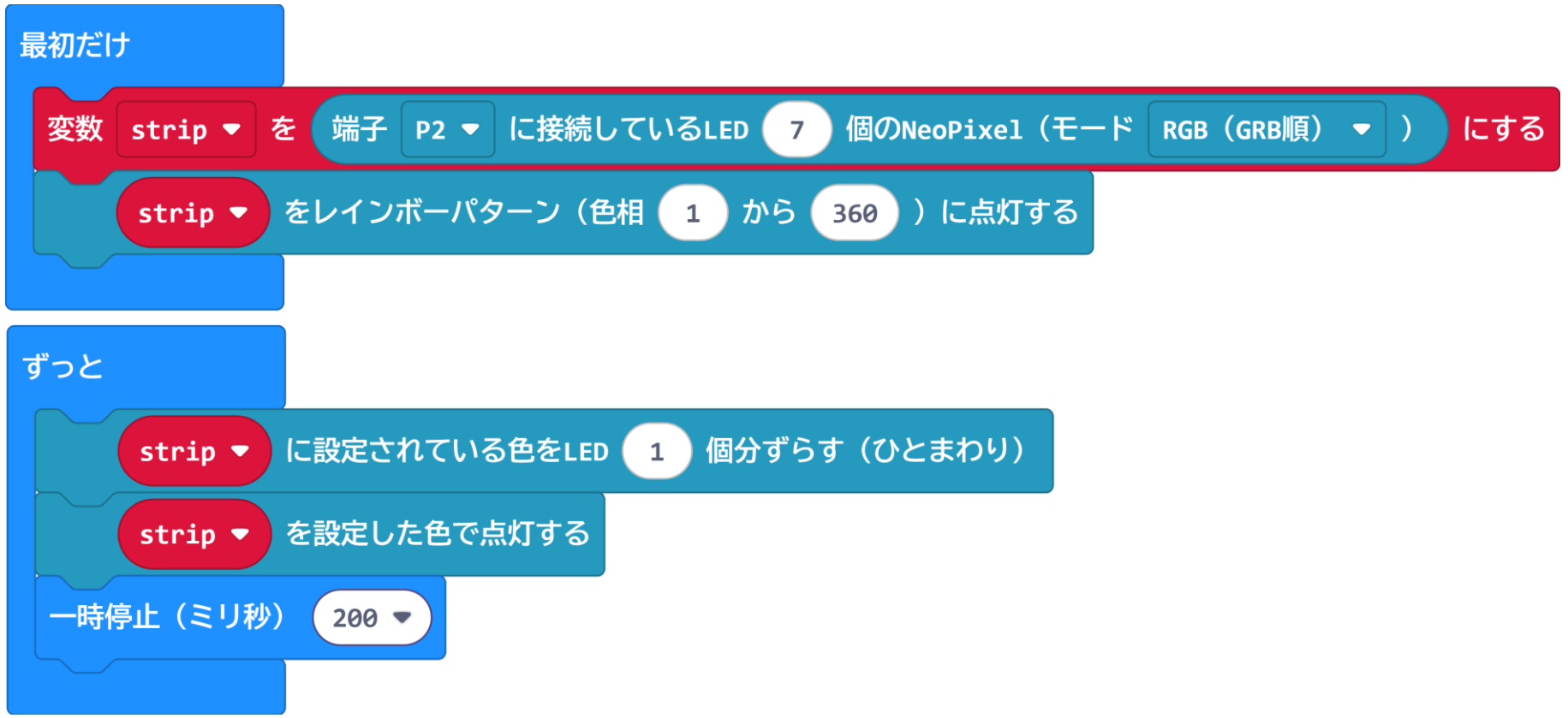
拡張機能について

- MakeCodeエディタには「拡張機能」というものがあります。
- 「拡張機能」をつかうと、特別な機能のブロックを追加することができます。
- 今回は「フルカラーLED」をつかうための拡張機能をつかいます。
  - ツールボックスで「拡張機能」をクリックします。
  - 検索ウィンドウに「neopixel」と入力します。
  - 表示された中から「neopixel」をクリックします。
  - ツールボックスに「Neopixel」があらわれます。



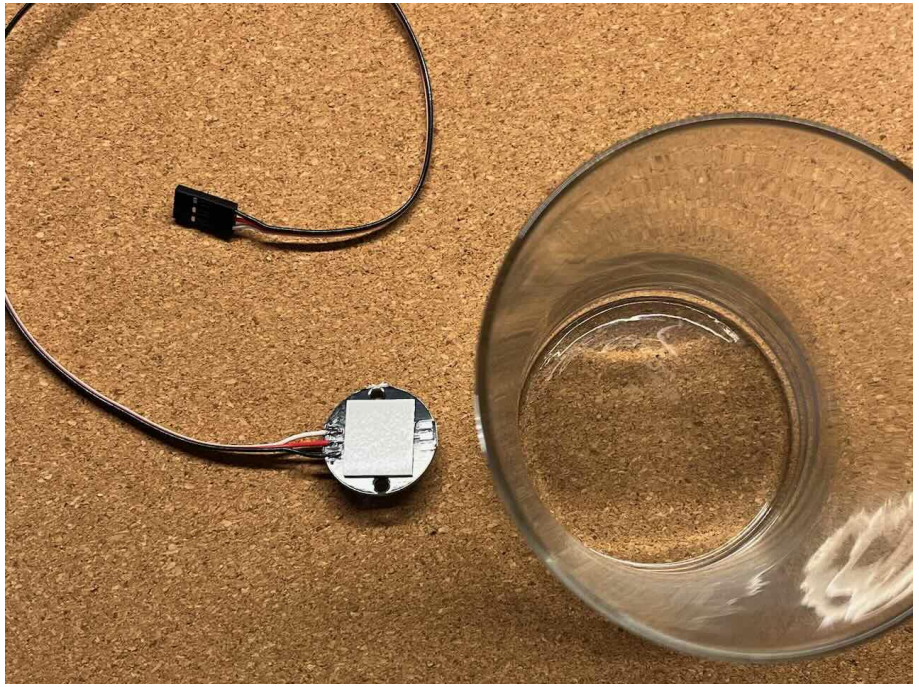
動作確認（フルカラーLEDをひからせてみる）

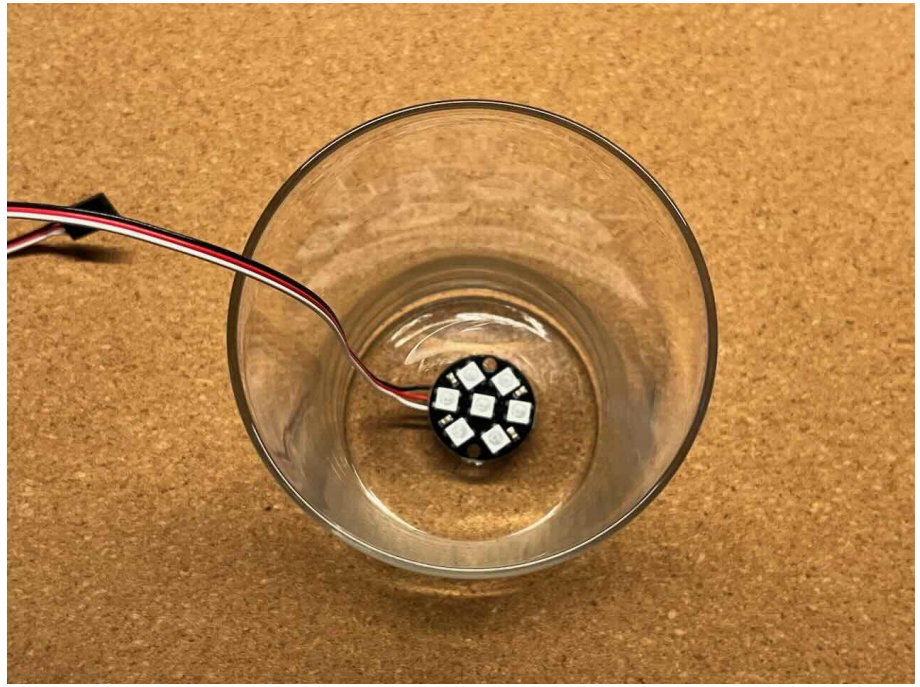
- MakeCodeエディタで以下のプログラムをつくり、マイクロビットにかきこんでください。
- フルカラーLEDがレインボーカラーにひかり、0.2秒ごとに色がかわります。



イルミネーショングラスのくみたて

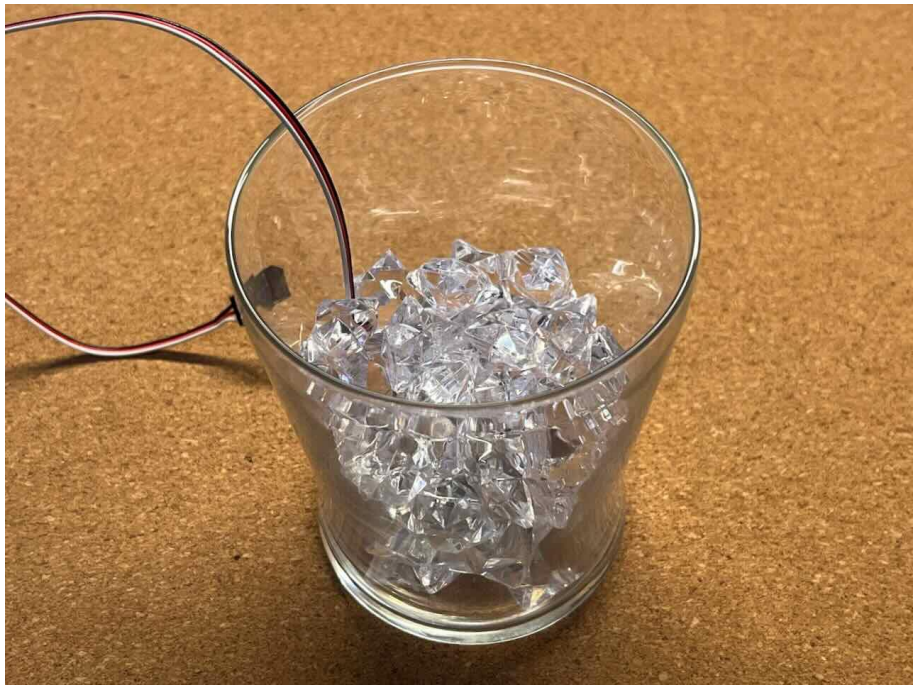
- 両面テープで、フルカラーLEDをガラスコップのそこにはりつけます。
- コードをコップの外にひきだします。





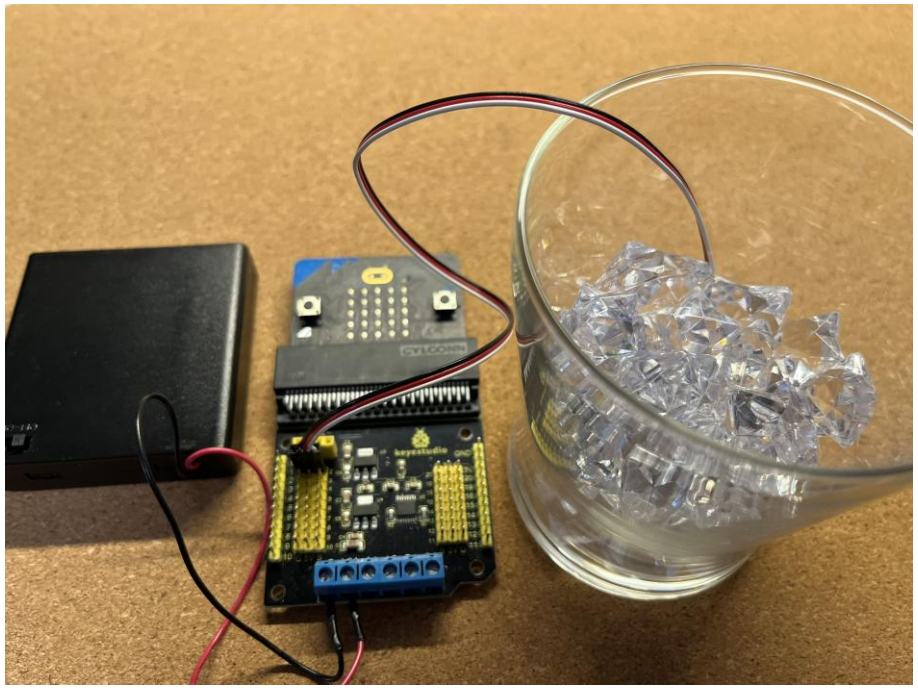
イルミネーショングラスのくみたて

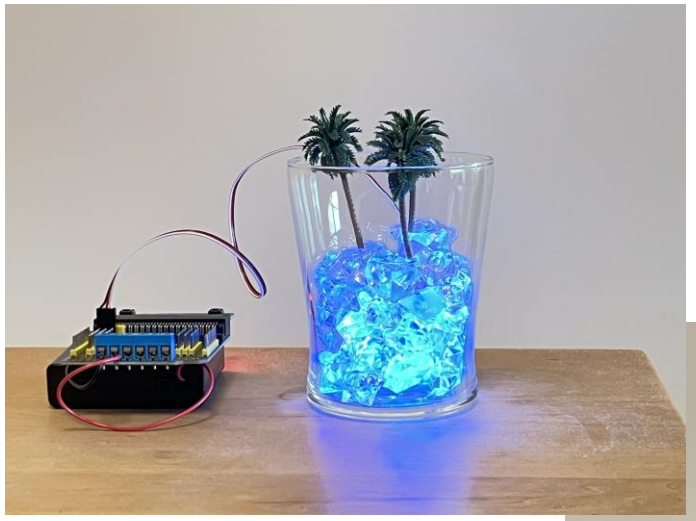
- ガラスコップの中にインテリアストーンをいれます。

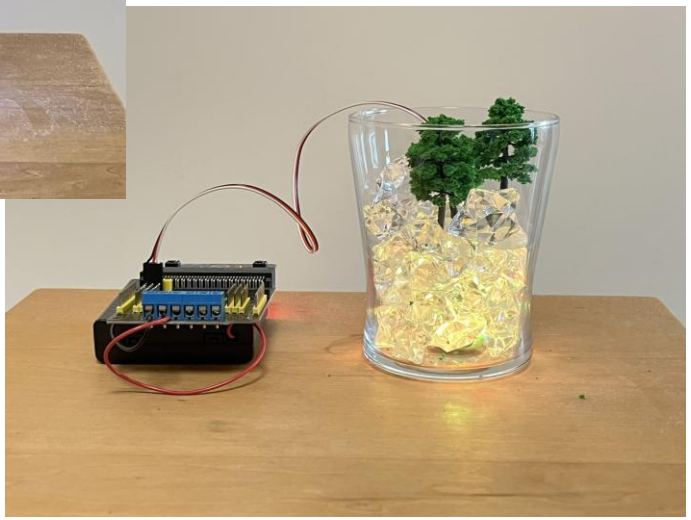


イルミネーショングラスのくみたて

- これで完成です。いろいろくふうして、もっときれいにかざりつけてください。



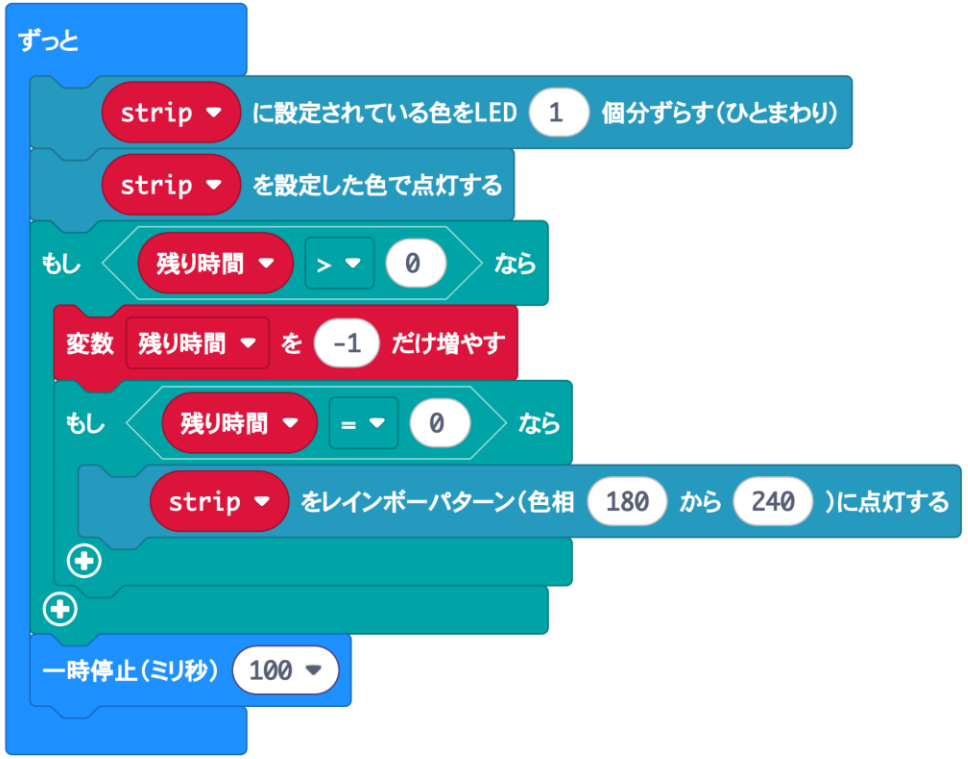




イルミネーショングラスのプログラム

- ふだんは青系のグラデーションでLEDが光ります。
- まわりの音がうるさくなったら、しばらくの間だけ、LEDの色がオレンジ系のグラデーションにかわります。
- LEDの色は0.1秒ごとにとなりのLEDにずれていき、色がクルクル回ります。

イルミネーショングラスのプログラム ～作成例

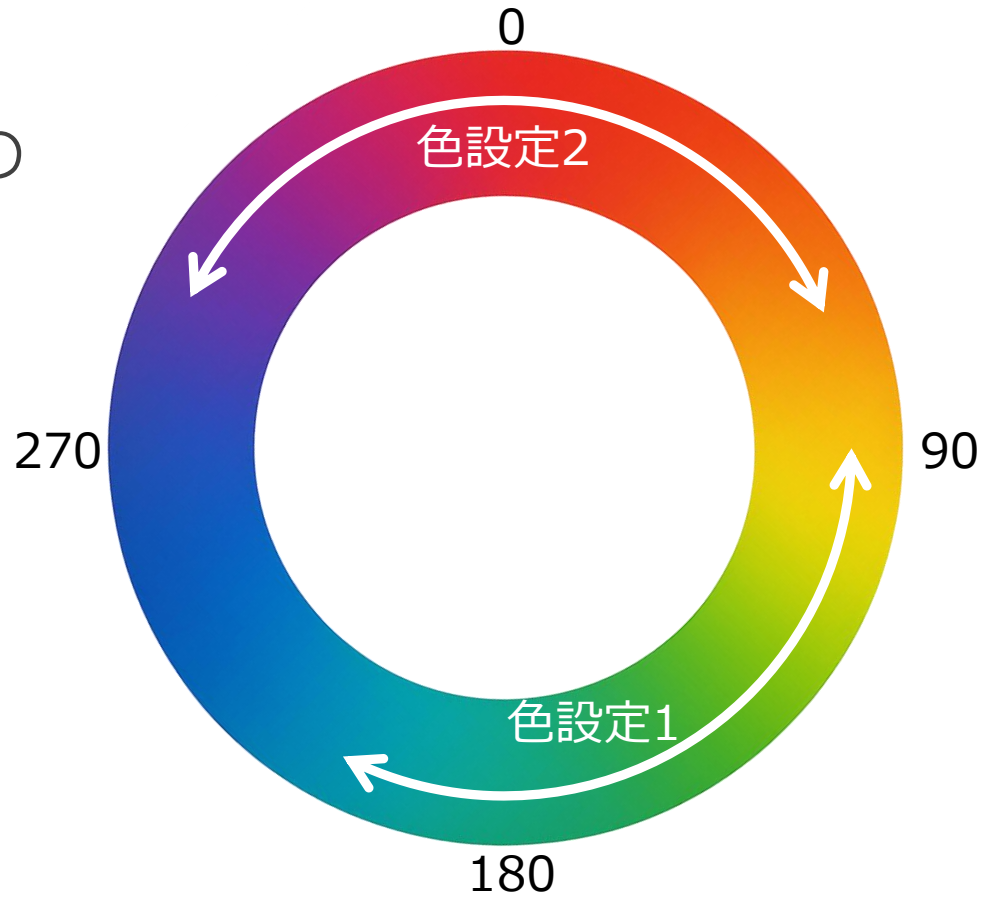


プログラムの説明

- 7個のLEDにそれぞれ色を設定して光らせます。フルカラーLEDは端子「2」につなぎます。
- 0.1秒ごとに、設定した色をとなりにずらして光らせます。
- 最初に、LEDに青系のグラデーションを設定します。
- まわりの音がうるさくなったら、LEDにオレンジ系のグラデーションを設定します。また、オレンジ系グラデーションで光らせる時間を「残り時間」という変数であらわします。「残り時間」を「10」にします。
- 0.1秒ごとに、「残り時間」をマイナス1 します。つまり、1秒たったら「残り時間」がゼロになります。
- 「残り時間」がゼロになったら、LEDに青系のグラデーションを設定します。これで最初の状態にもどります。

プログラムの説明

- このプログラムでは、「色相」という数値をつかってLEDの色を設定しています。
- 「色相」というのは「色あいのちがい」のことで、右の図のように「0～360」の数値であらわします。
- このプログラムでは、以下のように色相を設定しています（数値を変えるとLEDの色あいが変わります）。
  - 青系のグラデーション：180～240
  - オレンジ系のグラデーション：0～60



hue（色相）

プログラムの改良

- 自分のすきなように、プログラムを改良してみましょう。
- 「マイク」のかわりに「光センサ」や「加速度センサ」をつかって、「くらくなったときに色をかえる」「かたむいたら色をかえる」ことなどもできます。
- フルカラーLEDの光らせ方をかえることもできます。

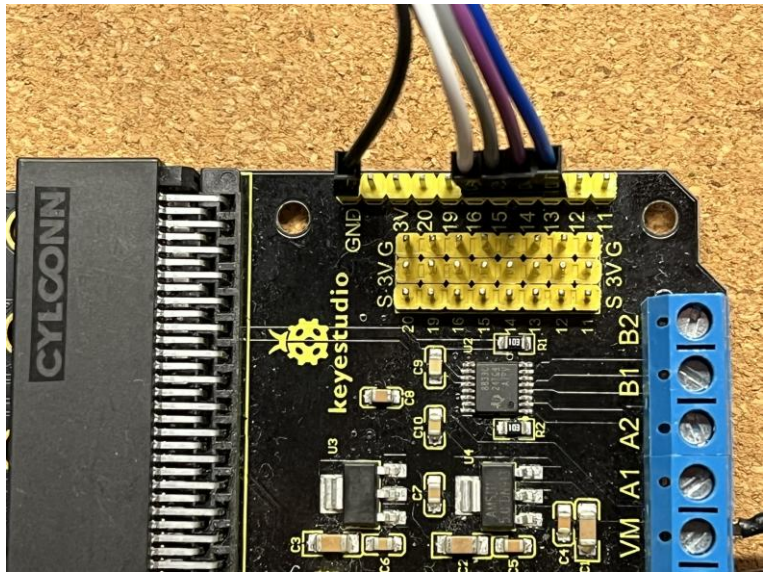
早おしゲームをつくろう！

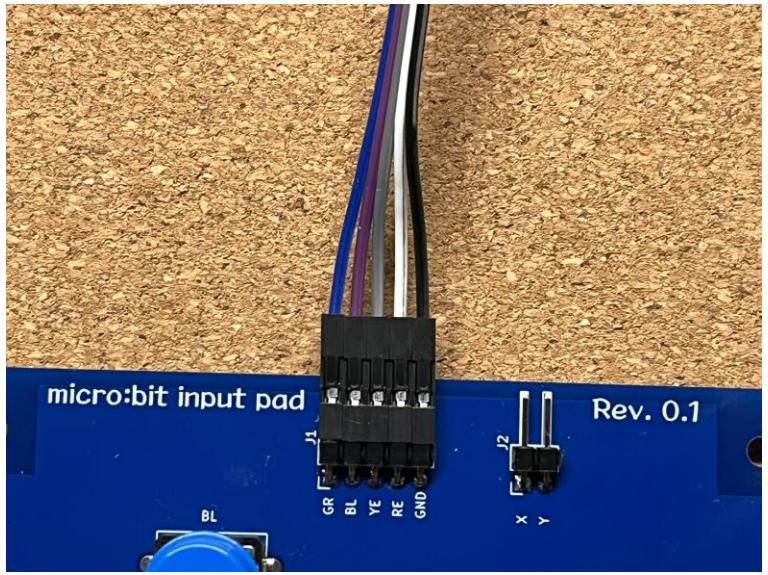
- 「ボタン基板」もくみあわせて、「早おしゲーム」をつくってみましょう。
  - Aボタンをおしたらゲーム開始です。
  - 少しまつと、フルカラーLEDが「緑」「青」「黄」「赤」のどれかの色でひかります。
  - LEDがひかったら、できるだけ早く、同じ色のボタンをおします。
  - LEDがひかってからボタンをおすまでの時間をはかります。
  - これを5回くりかえし、ボタンをおすまでの時間の合計から成績を計算します。
  - 成績によって、フルカラーLEDをちがう色でひからせます。

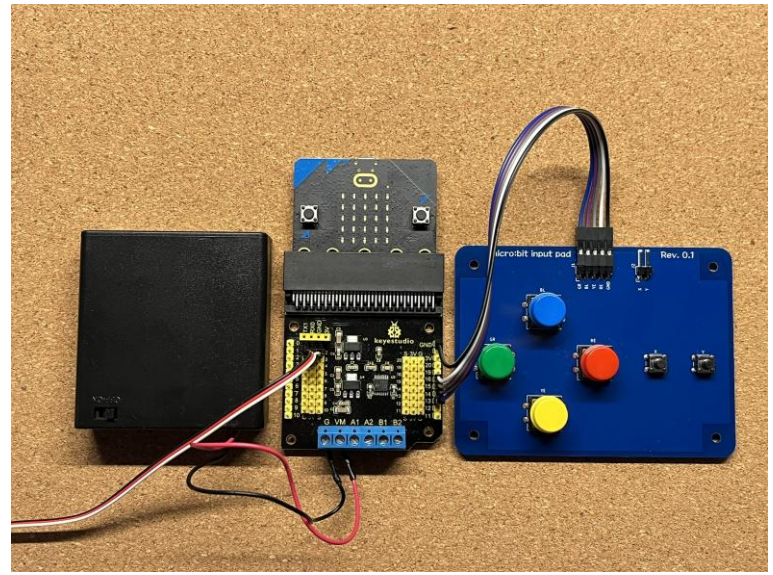
ボタン基板の接続

- 「拡張ボード」と「ボタン基板」を以下のとおりにつなぎます。

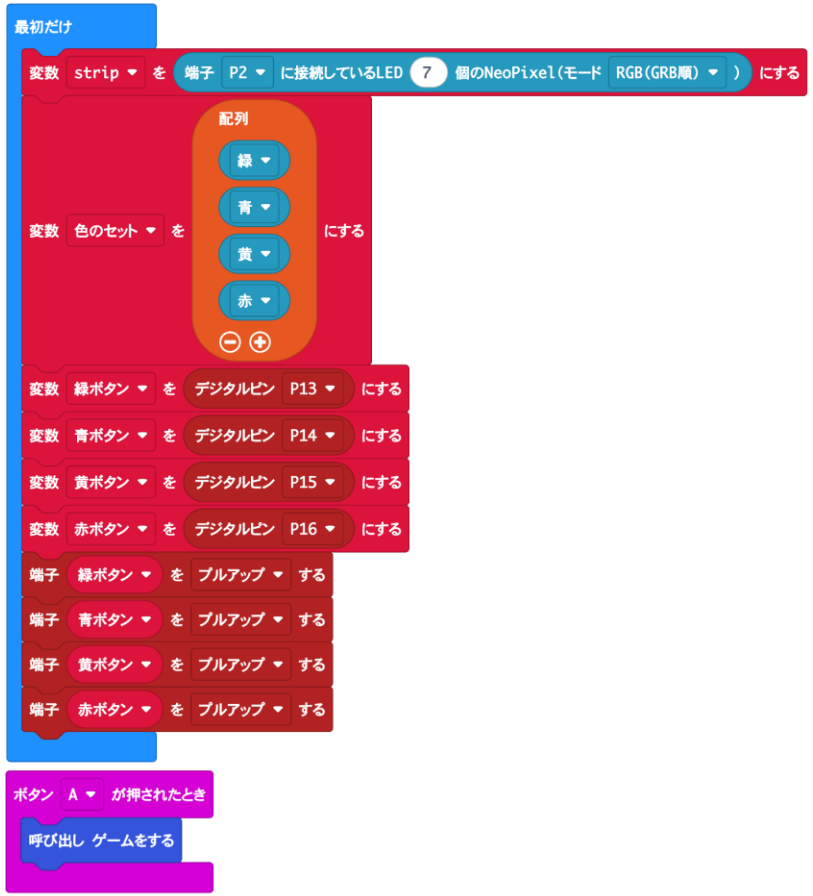
拡張ボード	ボタン基板
13	GR
14	BL
15	YE
16	RE
GND	GND

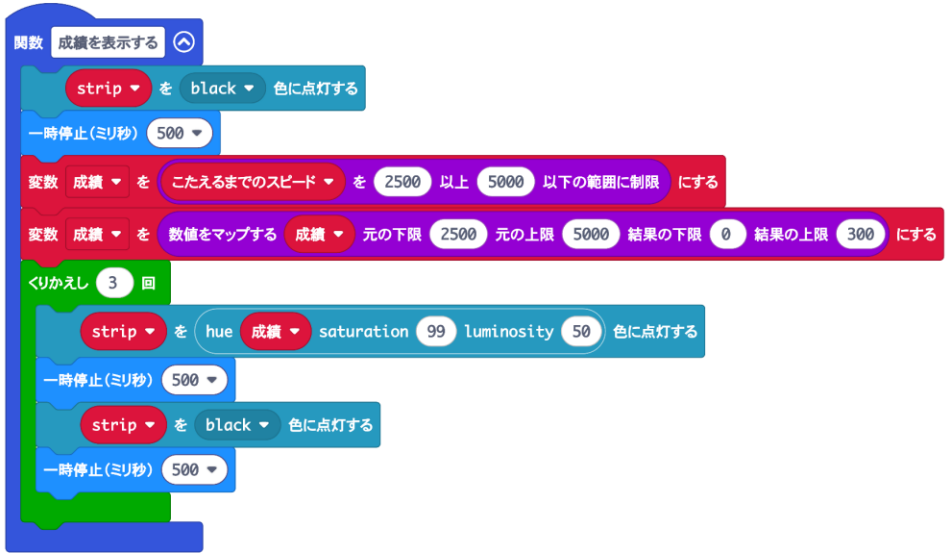






早おしゲームのプログラム





プログラムの説明

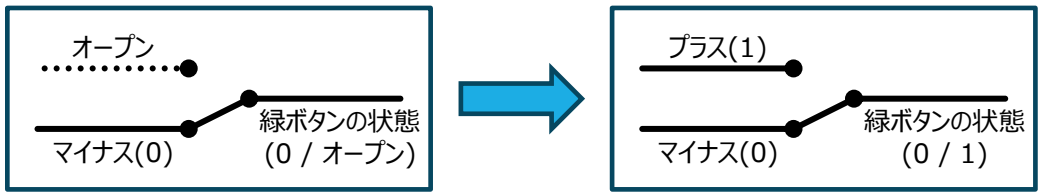
```
最初だけ
  変数 strip を 端子 P2 に接続しているLED 7 個のNeoPixel(モード RGB(GRB順)) にする
  変数 色のセット を 配列 [緑 青 黄 赤] にする
  変数 緑ボタン を デジタルピン P13 にする
  変数 青ボタン を デジタルピン P14 にする
  変数 黄ボタン を デジタルピン P15 にする
  変数 赤ボタン を デジタルピン P16 にする
  端子 緑ボタン を プルアップ する
  端子 青ボタン を プルアップ する
  端子 黄ボタン を プルアップ する
  端子 赤ボタン を プルアップ する
```

入出力端子の2番につながっているフルカラーLED（LEDは7つ）を、「strip」という名前の変数に設定します。

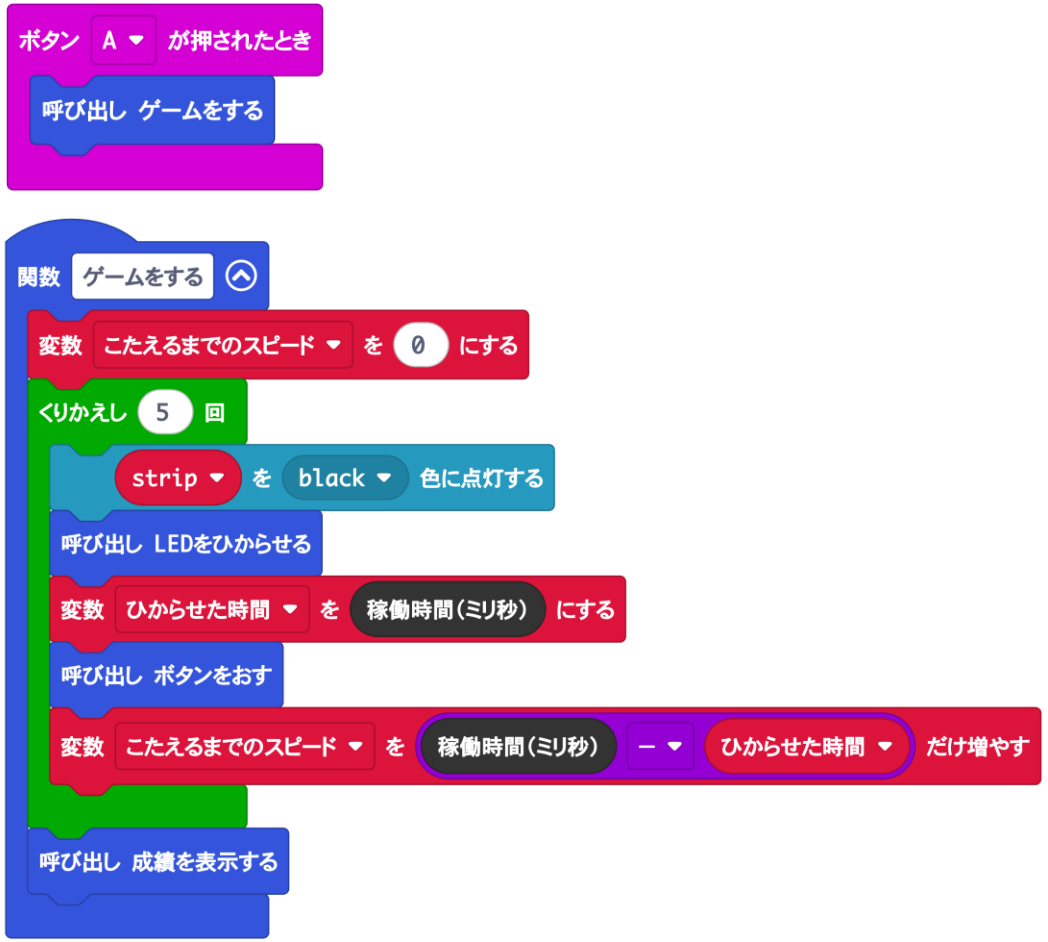
「配列」をつかうと、ひとつの変数にたくさんの値を設定することができます。
ここでは「色のセット」という変数の0番目に、Neopixelの「緑」を、1番目に「青」を、2番目に「黄」を、3番目に「赤」を設定しています。

入出力端子の13番（デジタルピン）を「緑ボタン」という名前の変数に設定します（ほかの端子も同じ）。
最初にこのように設定しておくことで、このあとのプログラムで、入出力端子への処理をするときに、変数名で処理することができるようになり、プログラムがわかりやすくなります。

ボタンは、そのままだと「マイナス（0）につながっているか？」それとも「はなれているか？」をよみとりますが、最初にこの処理をしておくことで、「マイナス（0）につながっているか？」それとも「プラス（1）につながっているか？」としてあつかうことができるようになります。



プログラムの説明



Aボタンがおされたとき「ゲームをする」関数を実行します。

さいしょに「こたえるまでのスピード」という変数の値をゼロにしておきます。

このあとの内容を5回くりかえします。

フルカラーLEDをけします。

「LEDをひからせる」関数を実行して、そのときの時間を覚えておきます。

「ボタンをおす」関数を実行して、LEDをひからせてからボタンをおすまでの時間を計算します。
5回分の計算結果を「こたえるまでのスピード」にたしていきます。

5回のくりかえしがおわったら、「成績を表示する」関数を実行します。

プログラムの説明

0～5秒のあいだのランダムな時間だけまちます。

ひからせる色の番号（0～3）をランダムにきめます。

きめた色でフルカラーLEDをひからせます。
「最初だけ」で設定しておいた「色のセット」という変数の配列をつかっています。

プログラムの説明

最初に「おした色の番号」という変数の値を「99」にしておきます。

先ほどLEDをひからせたときの色の番号と、「おした色の番号」がおなじになるまでくりかえします。

緑ボタンをよみとった結果が「0」だったら（緑ボタンがおされていたら）、
「おした色の番号」という変数の値を「0」にします。

青ボタンをよみとった結果が「0」だったら（青ボタンがおされていたら）、
「おした色の番号」という変数の値を「1」にします。

黄ボタンをよみとった結果が「0」だったら（黄ボタンがおされていたら）、
「おした色の番号」という変数の値を「2」にします。

赤ボタンをよみとった結果が「0」だったら（赤ボタンがおされていたら）、
「おした色の番号」という変数の値を「3」にします。

それ以外のときは（どのボタンもおされていなかったら）、
「おした色の番号」という変数の値を「99」にします。

プログラムの説明

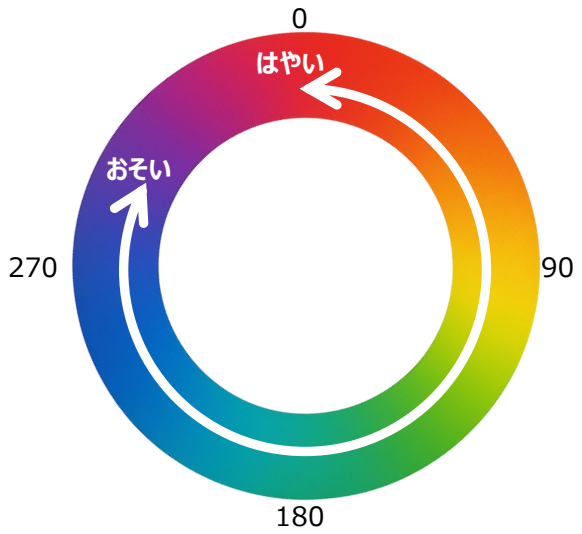
```
関数 成績を表示する
    strip を black 色に点灯する
    一時停止(ミリ秒) 500
    変数 成績 を こたえるまでのスピード を 2500 以上 5000 以下の範囲に制限 にする
    変数 成績 を 数値をマップする 成績 元の下限 2500 元の上限 5000 結果の下限 0 結果の上限 300 にする
    くりかえし 3 回
        strip を hue 成績 saturation 99 luminosity 50 色に点灯する
        一時停止(ミリ秒) 500
        strip を black 色に点灯する
        一時停止(ミリ秒) 500
```

フルカラーLEDをけして、0.5秒まちます。

「こたえるまでのスピード」が2.5秒よりはやかったら2.5秒に、5秒よりおそかったら5秒にして、「成績」という名前の変数に設定します。これにより「成績」は2.5秒～5秒の間になります。

「成績」の値（2.5秒～5秒）が「0～300」の範囲になるように変換します。

フルカラーLEDを、「成績」の色で3回点滅させます。



【ノウハウ】NeoPixel拡張機能のブロック

- **【strip をレインボーパターン (色相 1 から 360) に点灯する】**ストリップを虹色のグラデーションで点灯させます。
- **【strip を 赤 色に点灯する】**ストリップを指定した色で点灯させます（すべてのLEDが同じ色になります）。
- **【strip を設定した色で点灯する】**ストリップをあらかじめ設定した色で点灯させます。
- **【strip の設定を解除する】**ストリップに対し、すべてのLEDをオフ（黒色）に設定します。
- **【strip に設定されている色をLED 1 個分ずらす】**ストリップに設定されている色情報をとなりのLEDにずらします。あいたLEDはオフ（黒色）になります。
- **【strip に設定されている色をLED 1 個分ずらす (ひとまわり)】**ストリップに設定されている色情報をとなりのLEDにずらします。あいたLEDには先頭のLEDの色情報が入ります。
- **【strip の 0 番目のLEDを 赤 色に設定する】**ストリップの各LEDに、個別に色を設定します。
- **【strip の明るさを 255 に設定する】**ストリップの明るさを 0 ～ 255 の数値で設定します。
- **【RGB (赤 255 緑 255 青 255)】**任意のRGB値でつくった色情報をつくります。

まとめ

- 「光の3原色」と「フルカラーLED」についてまなびました。
- マイクロビットのプログラムで、フルカラーLEDを光らせました。
- イルミネーショングラスをつくりました。
- プログラムによって、フルカラーLEDの光らせ方をいろいろ変えられることがわかりました。