

小中学生のための micro:bit を利用した

プログラミング教室（第 1 回）資料

氏 名 _____

主催 NPO 法人 学習開発研究所

後援 京田辺市教育委員会

講習(1)：カラーLED を点灯してみよう（テーマ 6）

日 時：2025 年 7 月 27 日(日) 10:00～12:30

場 所：京田辺市中央公民館 2F 第 3・4 研修室

問い合わせ先：ild-kemsyu@u-manabi.org

担当者：NPO 法人 学習開発研究所 三輪、高橋

テーマ6 カラーLED を点灯してみよう

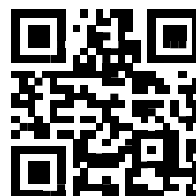
目 次

- | | |
|------------------------|--------------------------------|
| 1. micro:bit のプログラム | プログラム prei1-1,prei1-2 |
| 2. プログラムの基本(順次構造、反復構造) | プログラム prei2-1,prei2-2,prei2-3 |
| 3. プログラムの基本(分岐構造) | プログラム prei2-41,prei2-4,prei2-5 |
| 4. LED の点灯 | プログラム prei6-1,prei6-2,prei6-3 |
| 5. フルカラーLED の制御 | プログラム prei6-4,prei6-6,prei6-7 |

注) 例題番号は、「小中学生のためのプログラミング～学習ガイド」の例題番号に対応しています。
本文中の例題の右端の、「☆」は応用、「★」は発展、「無印」で基礎の問題を示しています。

<小中学生のためのプログラミング教室>

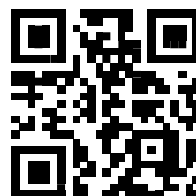
<https://u-manabi.net/ild-pkouza/>



<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)

<https://u-manabi.net/microbit/>



1. micro:bit のプログラム

【例題 1-1】 ハートの表示 (プログラム prei1-1)

「最初だけ」ブロックを利用して、ハートを表示してみよう。



「最初だけ」ブロック

*「基本」-「LED 画面に表示」LED をタップして、ハート形に見えるように LED を ON にする。

【例題 1-2】 ハートの点滅 (プログラム prei1-2)

「ずっと」ブロックに変更して、ハートを点滅してみよう。



「ずっと」ブロック

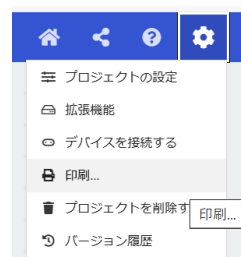
*「基本」-「一時停止」時間を 500 (ミリ秒) にする。

*「表示を消す」

*「基本」-「一時停止」時間を 500 (ミリ秒) にする。

<参考> プログラムの印刷

エディタの画面 (右上端) にある歯車 (⚙️) から「印刷」を選択する。少し経過して「プログラムを印刷する」の画面が表示され、ブロックのプログラムを印刷できる。



2. プログラムの基本(順次構造、反復構造)

【例題 2-1】 アイコンの表示 (プログラム prei2-1)

3つのアイコン(「ハート」「小さいダイヤモンド」「しかく」)を表示してみよう。

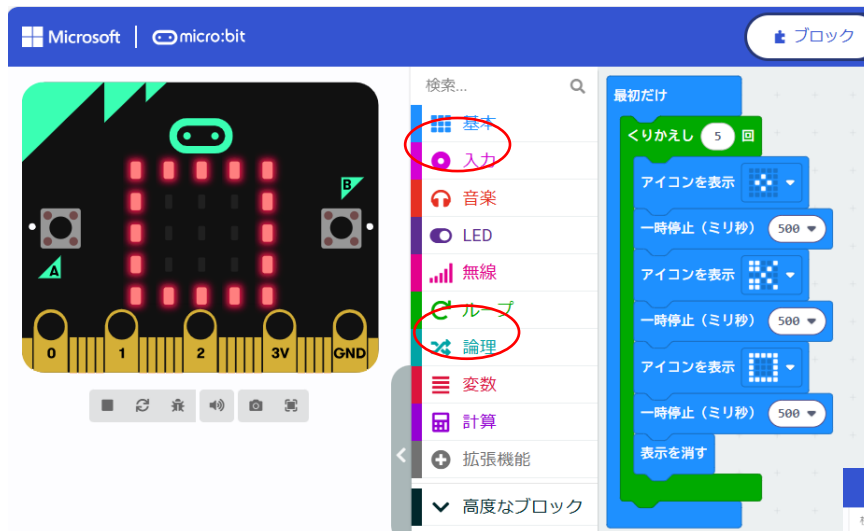


「最初だけ」ブロック

- *「基本」-「アイコン表示」(ハート)、「基本」-「一時停止」
- *「基本」-「アイコン表示」(小さいダイヤモンド)、「基本」-「一時停止」
- *「基本」-「アイコン表示」(しかく)、「基本」-「一時停止」
- *「基本」-「表示を消す」

【例題 2-2】 アイコンの繰り返し表示 (プログラム prei2-2)

アイコンを 3 つ(「小さいダイヤモンド」「はさみ」「しかく」)を繰り返し 5 回表示してみよう。



「最初だけ」ブロック

- *「基本」-「アイコン表示」の「はさみ」を選択する。
- *「ループ」-「くりかえし」数値 0 を 5 にする。

じゃんけんの「グー」「チョキ」「パー」をこの 3 つのアイコンで表示する。

反復構造は、繰り返し構造ともいう。



【例題 2-3】 カウンターの利用 (プログラム prei2-3)

「変数 カウンター」を利用して、表示してみよう。



「最初だけ」ブロック

*「ループ」の「変数 カウンター ~ くりかえす」数値 0 を 4 にする。

変数「カウンター」は、「0 から始まり、「0, 1, 2, 3, 4」の 5 回繰り返す。

注) 変数は数値や文字を入れるための領域

3. プログラムの基本 (分岐構造)

【例題 3-1】 「グー」「パー」の表示 (プログラム prei2-41)

変数 $c=0$ のときは、「グー」を表示し、 $c=1$ ときは、「パー」を表示してみよう。



「最初だけ」ブロック

*「変数」-「変数を追加する」変数 c にする。

*「変数」-「変数 c を 0 にする」

*「論理」-「条件判断」(もし なら ~ でなければ ~)

*「論理」-「くらべる」「 $0 = \nabla 0$ 」を「 $c = \nabla 0$ 」にする。

分岐構造は、選択構造ともいう。

順次構造、反復構造(繰り返し構造)、分岐構造(選択構造)をプログラムの基本構造という。

【例題 3-2】乱数の利用 (プログラム prei2-4)

乱数を利用して、変数 c に 0 から 1 の数値を入れるように変更してみよう。

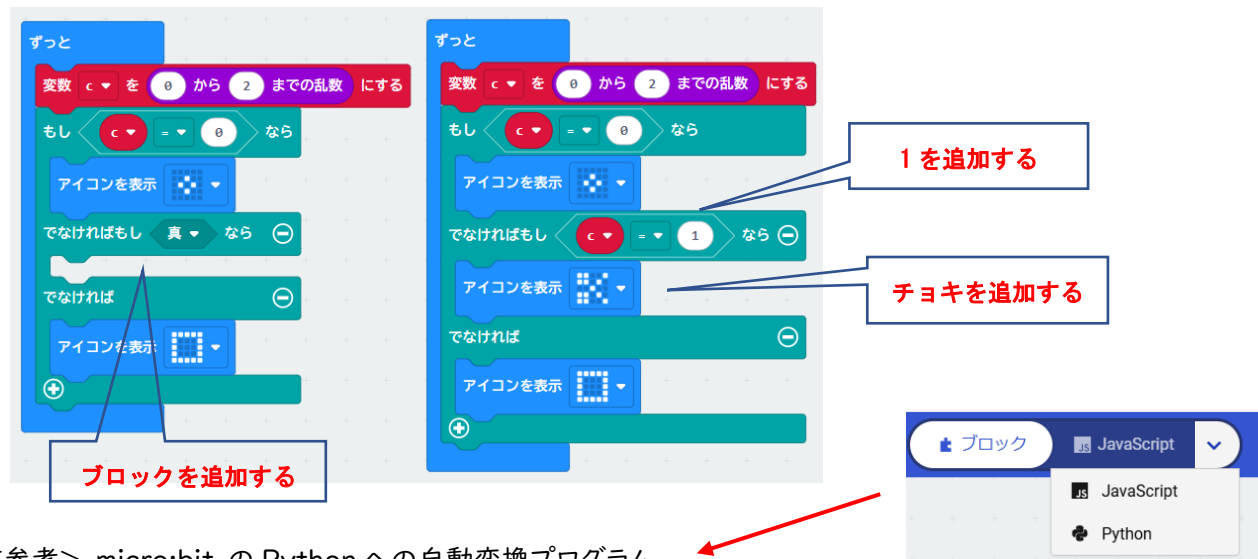


「ずっと」ブロック

- *「変数」-「変数を追加する」
- *変数の名前を c にする。
- *「計算」-「0~10 までの乱数」数値の 10 を 1 にする。
(乱数は 0、1 のいずれか)

【演習】「グー」「チョキ」「パー」の表示 (プログラム prei2-5)

例題 3-2 のプログラムを「グー」「チョキ」「パー」を表示するように変更してみよう。



1 を追加する

チョキを追加する

ブロックを追加する

<参考> micro:bit の Python への自動変換プログラム

(1) 反復構造 (prei2-3) <pre>for カウンター in range(5): basic.show_icon(IconNames.SMALL_DIAMOND) basic.pause(500) basic.show_icon(IconNames.SCISSORS) basic.pause(500) basic.show_icon(IconNames.SQUARE) basic.clear_screen()</pre>	(2) 分岐構造 (prei3-1) <pre>c = 0 if c == 0: basic.show_icon(IconNames.SMALL_DIAMOND) else: basic.show_icon(IconNames.SQUARE)</pre>
---	--

4. LED の点灯

【例題 4-1】 光センサによる「ハート」の点滅 (プログラム prei6-1)

光センサで明るさを調べて、暗ければ、アイコンの「ハート」を点滅してみよう。



「ずっと」ブロック

*「論理」-「条件判断」で、「もし なら～」を選択する。

*「論理」-「くらべる」で、「 $0 \leq 0$ 」を選択する。

*「入力」-「明るさ」を重ねる。数値 0 を 150 にする。シミュレータの明るさの数値は 128

【例題 4-2】 光センサによる LED の点灯 (プログラム prei6-2)

光センサで明るさを調べて、暗ければ、LED を点灯してみよう。



「ずっと」ブロック

*「論理」-「条件判断」で、「もし なら～」を選択する。

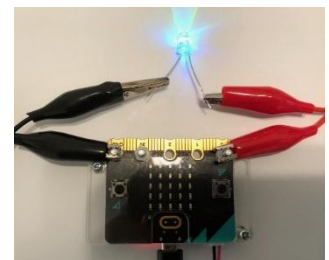
*「論理」-「くらべる」で、「 $0 \leq 0$ 」を選択する。

*「入力」-「明るさ」を重ねる。数値 0 を 125 にする。

*「高度なブロック」で「入出力端子」選択し、

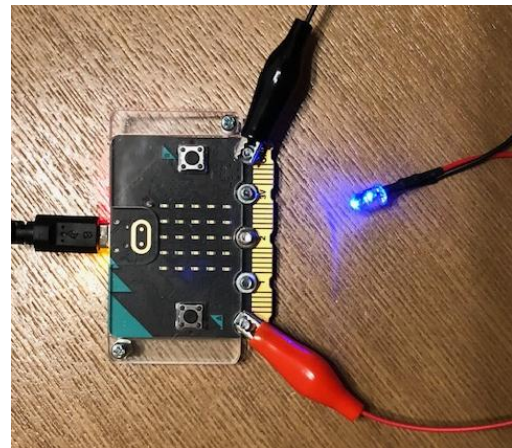
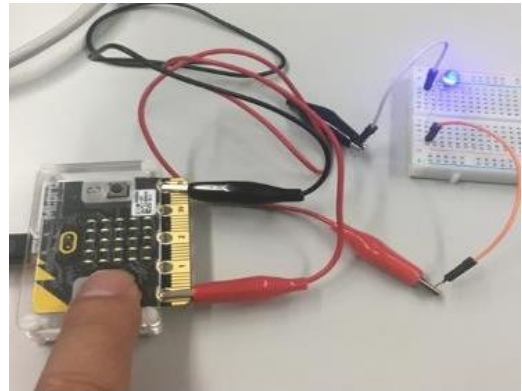
「デジタルで出力する 端子 P0 ▼ 値」を選択、数値は、1 と 0 にする。

注) 図のシミュレータでは、明るさの数値が 70 で、デジタル端子 P0 の出力は 1 になっている。



【例題 4-3】スイッチによる LED の点灯 (プログラム prei6-3)

A ボタンを押すと、LED が点灯し、もう 1 回押すと、消灯するプログラムを作成してみよう。



「最初だけ」ブロック

*「変数」-「変数を追加」で、変数は、s にする。

*「変数」-「変数 s を 0 にする」

「ボタン A」

*「論理」-「条件判断」で、「もし なら～」を選択し、「s = 0」を重ねる。

*「もし」ブロックでは、

「デジタルで出力する 端子 PO ▼ 値」の数値は 1

「変数」-「変数 s を 0 にする」で、数値は 1 にする。

*「でなければ～」ブロックでは、

「デジタルで出力する 端子 PO ▼ 値」の数値は 0

「変数」-「変数 s を 0 にする」で、数値は 0 にする。

<参考文献>

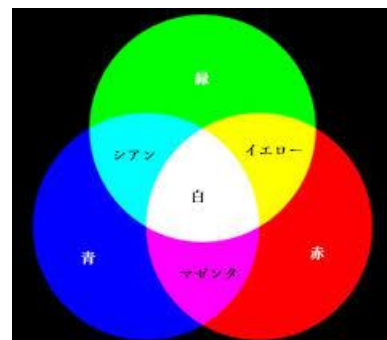
高橋、喜家村、稲川: micro:bit で学ぶプログラミング、pp.35-36、p.38、コロナ社(2019)

5. フルカラーLED の制御

【Neopixel】

Neopixel は、1つのセルごとに赤(R)、緑(G)、青(B)の3つのLEDと制御回路が入っており、フルカラーで光らせることができるLEDの集合体である。

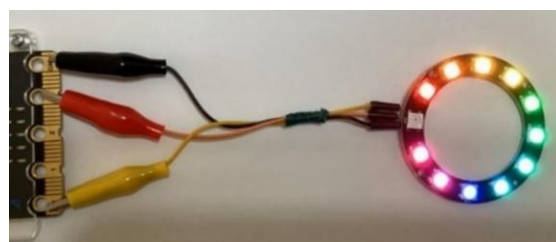
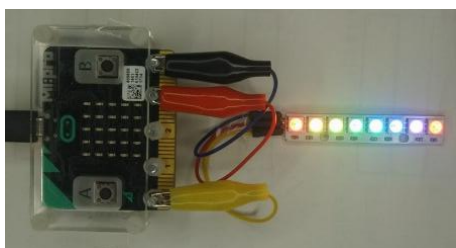
フルカラーは、色の明るさを、赤(R)、緑(G)、青(B)が、それぞれ0~255の256段階で選べるので、 $256 \times 256 \times 256 = 16,777,216$ 色の表現が可能となる。なお、R:255、G:255、B:255では白、R:0、G:0、B:0では黒になる。



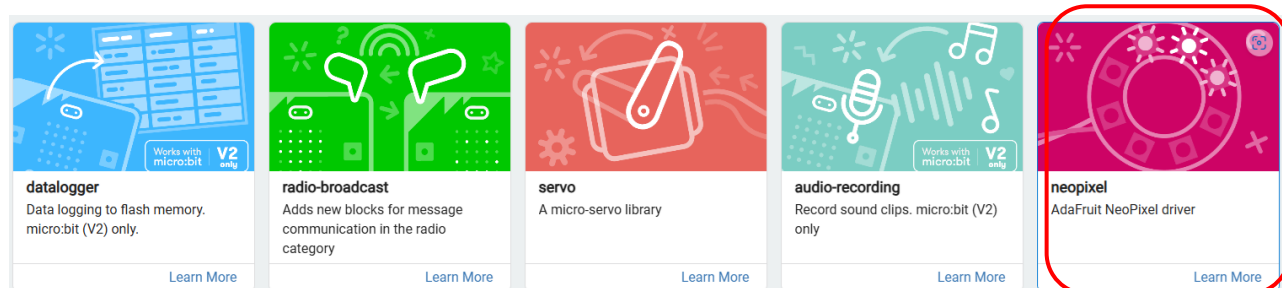
光の3原色 R(赤) G(緑) B(青)

【micro:bitとNeopixelの接続】

micro:bitとNeopixelを下図のように接続する(黄色はP0端子、赤は3V端子、黒はGND端子)。Neopixelは、8個のLEDが棒状(Stick型)(左)のに接続された製品で、その他にも、Neopixelには、リング状(右)の製品がある。

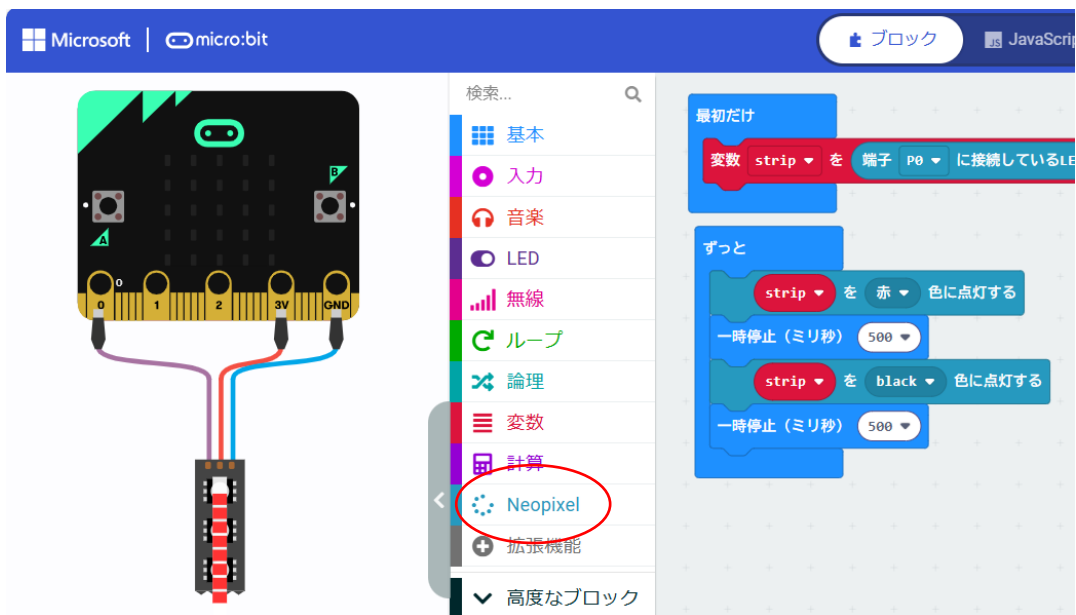


Neopixelを利用するには、ライブラリが必要である。ツールボックスの下にある「拡張機能」をクリックすると、拡張機能の一覧が表示されるので、「Neopixel」を選択する。ツールボックスの「計算」の下に、「Neopixel」のブロックが追加される。



【例題 5-1】 Neopixel の点滅 (プログラム prei6-4)

Neopixel(Stick 型)を赤色で点滅してみよう。



「最初だけ」ブロック

*「Neopixel」-「変数 strip を 端子 P0 に接続している LED 個… にする」

ここで、LED 24 個 → 8 個に変更しておく。

「ずっと」ブロック

*「Neopixel」-「strip 赤色に点灯する」色は、赤色のままにしておく。

*「Neopixel」-「strip black 色に点灯する」black では、消灯になる。

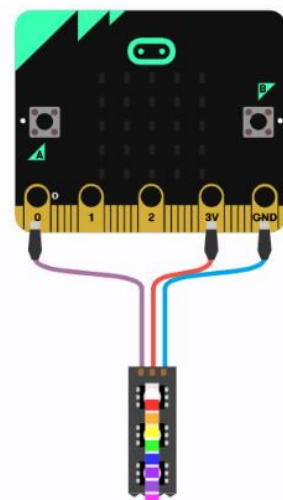
【例題 5-2】 Neopixel の点灯 (プログラム prei6-6)

Neopixel を 8 色で点灯させてみよう。

「最初だけ」ブロック

*「Neopixel」-「変数 strip を 端子 P0 に接続している LED 個・

ここで、LED 24 個 → 8 個に変更しておく。





「ずっと」ブロック

*「Neopixel」-「strip の 0 番目 色に設定する」 0~7 番目を図の色(赤…紫)にする。

*「Neopixel」-「strip で設定した色で点灯する」

【例題 5-3】 Neopixel の点灯(プログラム prei6-7)

Neopixel を 8 色で点灯させ、上下移動してみよう。

「最初だけ」ブロック 省略(前の例題に同じ)

「ずっと」ブロック

*「Neopixel」-「strip の 0 番目 赤 色に設定する」

*「ループ」-「くりかえし 8 回」にする。

*「Neopixel」-「strip で設定した色で点灯する」

*「Neopixel」-「strip 設定されている色を 1 個ずらす」

