プログラミング資料

カラーLEDを点灯してみよう

氏　名＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

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２回）

日時：2024 年 3 月 24 日(日)　10:00～12:00

場所：大阪芸術大学 短期大学部 伊丹学舎　E202教室

＜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/

問い合わせ先：ild-kensyu@u-manabi.org
担当：NPO法人　学習開発研究所　佐藤、西野

主催　NPO 法人 学習開発研究所
共催　大阪技術大学 短期大学部

Ⅰ. micro:bit のプログラム

Ⅰ-1　「最初だけ」ブロックの利用（ハートの表示）（プログラム prei1-1）

「最初だけ」ブロック

＊「基本」-「LED 画面に表示」

Ⅰ-2　「ずっと」ブロックの利用（ハートの点滅）（プログラム prei1-2）

「ずっと」ブロック

＊「基本」-「一時停止」「表示を消す」

2．プログラムの基本（順次構造、反復構造）

2-1　順次構造（プログラム prei2-1）

「ずっと」ブロック

＊「基本」-「アイコン表示」（ハート）

＊「基本」-「一時停止」

＊「基本」-「表示を消す」

2-2　反復構造（繰り返し 5 回）（プログラム prei2-2）

「最初だけ」ブロック

＊「基本」-「アイコン表示」の「はさみ」を選択

　じゃんけんの「グー」「チョキ」「パー」表示とする。

＊「ループ」－「くりかえし」　数値の 0→ 5

反復構造は、繰り返し構造ともいう。

別解（カウンター利用）（繰り返し５回）（プログラム prei2-3）

「最初だけ」ブロック

＊「ループ」の「変数 カウンター ～ くりかえす」　数値の 0→ 4

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

３．プログラムの基本（分岐構造）

3-1　分岐構造（分岐２つ）（プログラム prei3-1）

「最初だけ」ブロック

＊「変数」-「変数を追加する」　→　「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　＝▽　0」　→　「c　＝▽　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。

3-2　分岐構造（分岐２つ、乱数の利用）（プログラム prei3-2）

「ずっと」ブロック

＊「変数」-「変数を追加する」　→　「c」にする

＊「計算」-「0～10 までの乱数」　数値の 10→　1（乱数は 0、1 のいずれか）

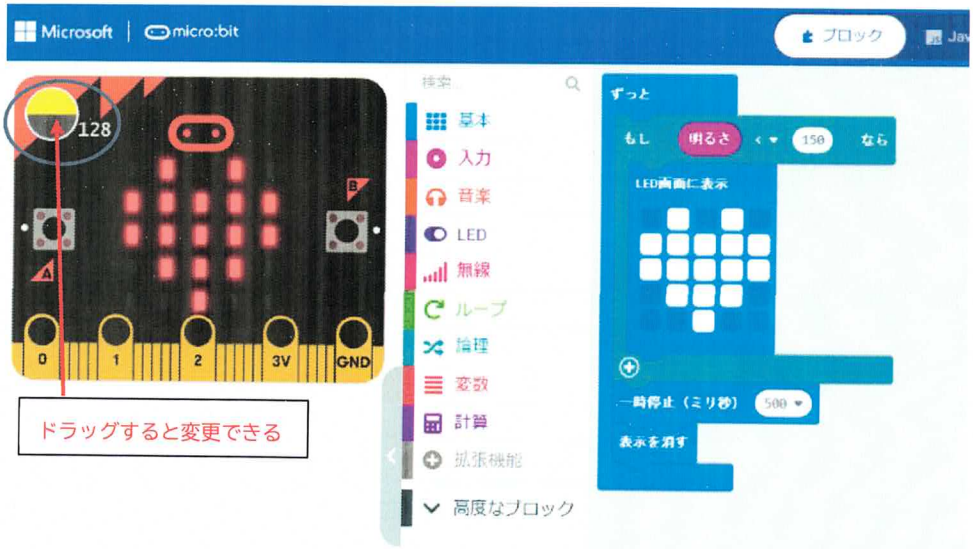
＜参考＞　micro:bit の Python への自動変換プログラム

```
(1) 反復構造 (prei2-3)
  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()

(2) 分岐構造 (prei3-1)
  c = 0
  if c == 0:
      basic.show_icon(IconNames.SMALL_DIAMOND)
  else:
      basic.show_icon(IconNames.SQUARE)
```

4．LED の点灯

4-1　光センサによる「ハート」の点滅（プログラム p-rei4-1）



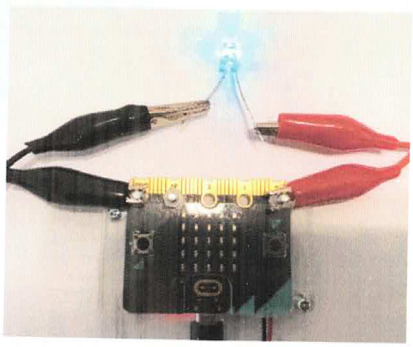
「ずっと」ブロック

＊「論理」－「条件判断」（もし　なら～）

＊「論理」－「くらべる」「0 < = 0」

＊「入力」－「明るさ」を重ねる。あとの数値 0→　150、シミュレータの明るさの数値は 128

4-2　光センサによる LED の点灯（プログラム p-rei4-2）



「ずっと」ブロック

＊「論理」－「条件判断」（もし　なら～）

＊「論理」－「くらべる」「0 < = 0」

＊「入力」－「明るさ」を重ねる。あとの数値 0→　125

＊「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

＊シミュレータの明るさの数値は 75、デジタル端子 P0 の出力は 1

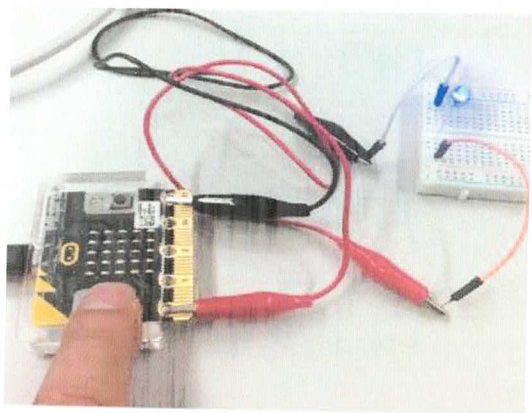
4-3　スイッチによるLEDの点灯（プログラム p-rei4-3）

「最初だけ」ブロック
＊「変数」－「変数を追加」　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」（もし　なら～）
＊「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 P0 ▼ 値」の数値は1
＊「変数」－「変数sを0にする」数値は1にする

「なら～」ブロックでは、
＊「デジタルで出力する 端子 P0 ▼ 値」の数値は0
＊「変数」－「変数sを0にする」　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)

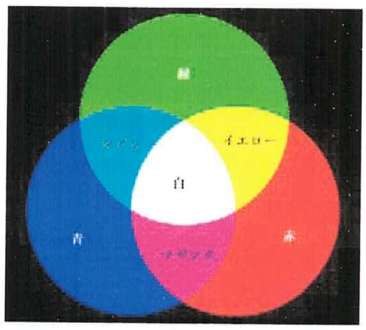
５．フルカラーLED の制御

【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 の集合体です。

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*256*256=16,777,216 色の表現が可能となります。

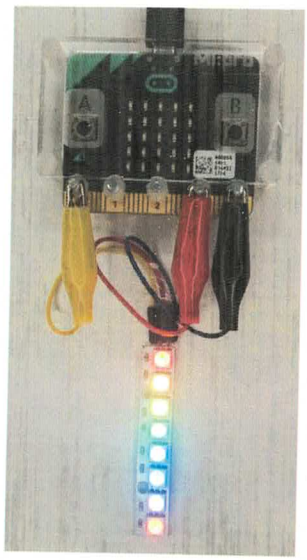
なお、R:255、G:255、B:255 では白、R:0、G:0、B:0 では黒になります。



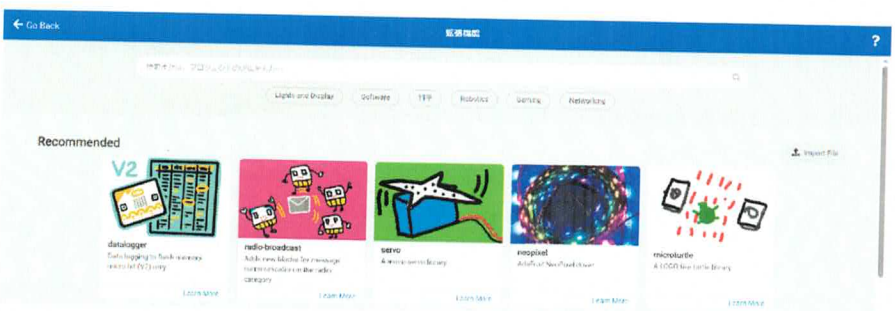
光の３原色　R(赤) G(緑) B(青)

【micro:bit と Neopixel の接続】

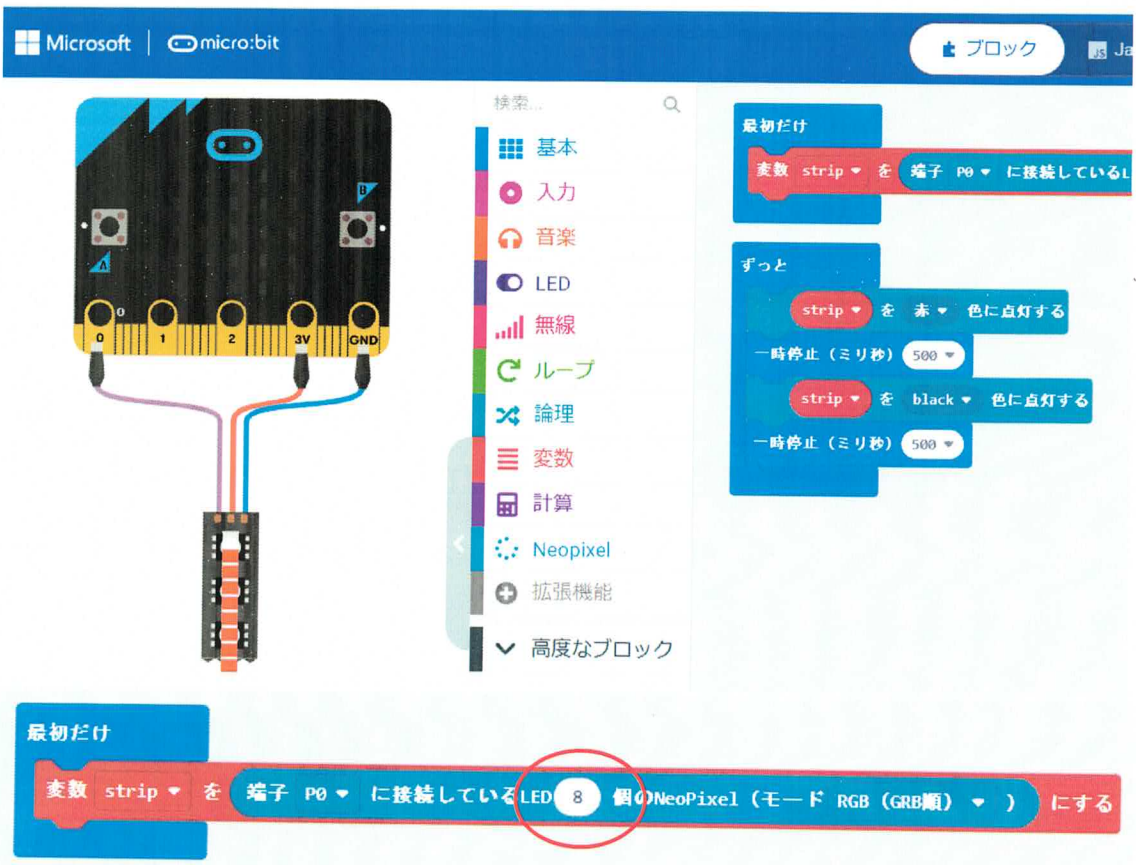
micro:bit と Neopixel を右図のように接続します（黄色は P0 端子,赤は 3V 端子,黒は GND 端子)。右図の Neopixel は、８個の LED が棒状（Stick 型）に接続された製品です。その他にも、Neopixel には、リング状の製品があります。



Neopixel を利用するには、ライブラリが必要です。ライブラリは、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ます。



5-1　Neopixel の点滅(赤色の点滅)（プログラム p-rei5-1)



「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

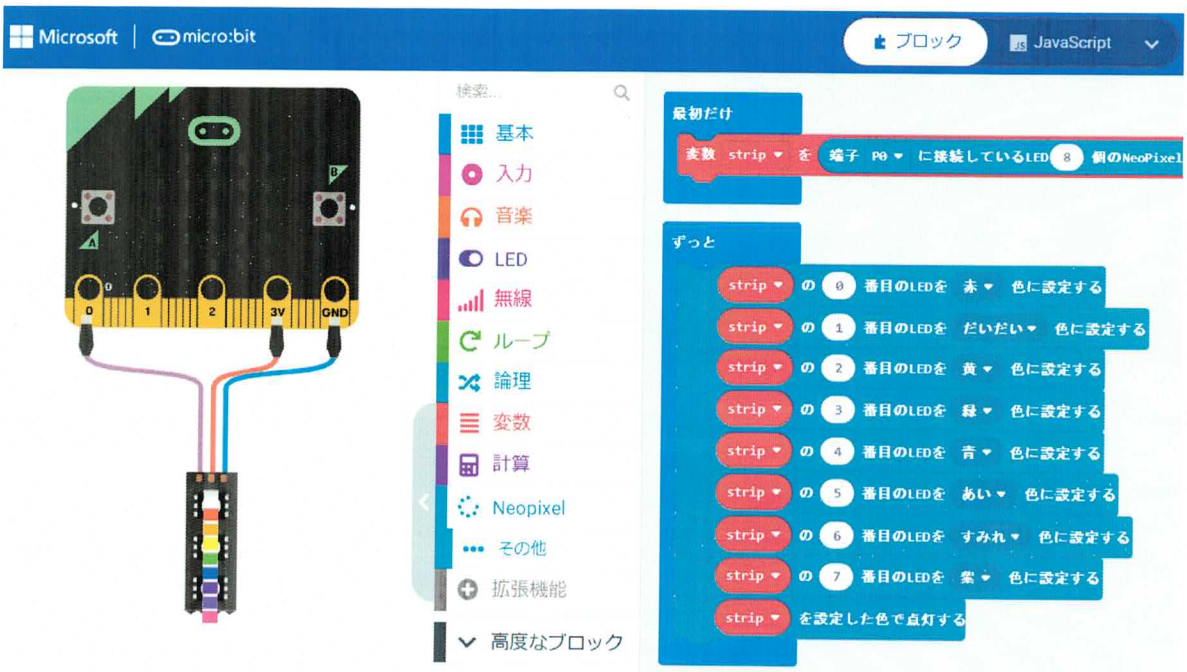
＊「Neopixel」－「strip　赤色に点灯する」　色は、赤色のままにしておく

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

＜参考＞（プログラム p-rei5-2)

　「Neopixel」で、「RGB（赤　緑　青)」色に点灯する」に変更すると、色はフルカラーで表現できる。下図では、シアン（水色に近い青緑色）になる。

5-2　Neopixel の点滅（8 色の点灯）（プログラム p-rei5-3）



「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」

5-3　Neopixel の点滅（色の上下移動）（プログラム p-rei5-4）

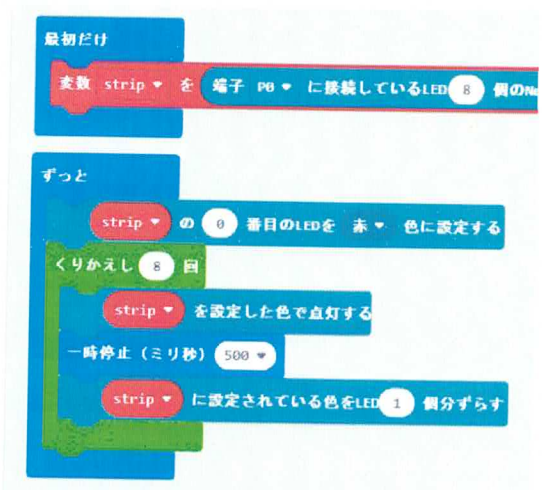
「ずっと」ブロック

＊「Neopixel」－「strip の 0 番目　赤 色に設定する」

＊「ループ」－「くりかえし　回」で、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」



＜2 色の上下移動）＞（プログラム p-rei5-5）

A ボタンで、赤色が上から下へ、B ボタンで、緑色が下から上へ移動するように変更する。

「A ボタン」

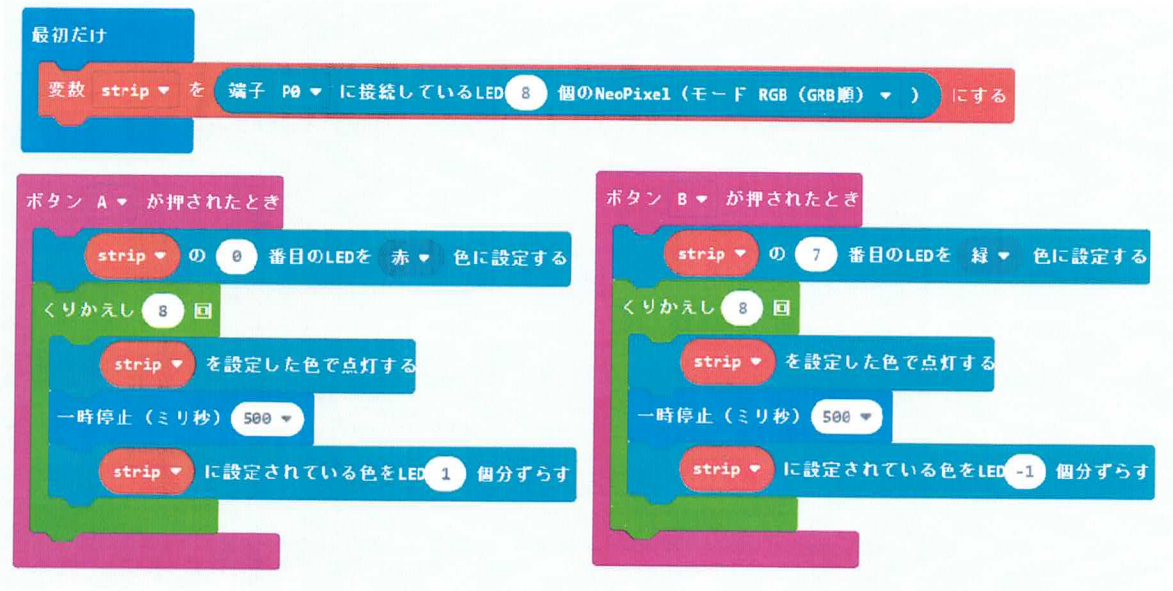
＊前の例題の「ずっと」ブロックを変更する。

「B ボタン」

＊「strip の 0 番目　緑 色に設定する」

＊「strip で設定した色で点灯する」

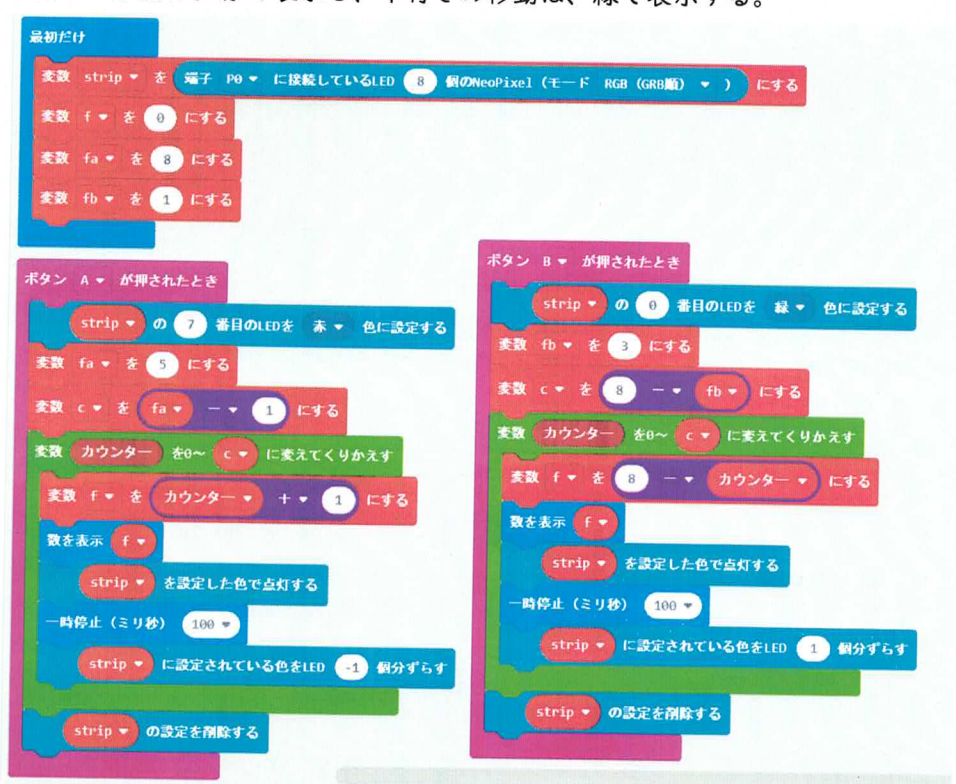
＊「strip 設定されている色を LED -1 個ずらす」とする



＜参考＞ エレベータのシミュレーション（プログラム p-rei5-6)

１階から８階までの建物で、エレベータで移動できるものする。次のような場合、Neopixel で表示するプログラムを考えてみよう。

・エレベータは、最初は１階（もしくは、８階）にある。
・１階でＡボタンを押すと、１階から上へ移動して、５階で止まる。
・８階でＢボタンを押すと、８階から下へ移動して、３階で止まる。
・上行きの移動は、赤で表示し、下行きの移動は、緑で表示する。



（参考）１．micro:bit（マイクロビット）の特徴

micro:bit は、イギリス BBC（英国放送協会）が開発し、Micro:bit 教育財団が７年生（11～12 歳）の生徒を対象に無料配布した手のひらサイズの安価なコンピュータです。

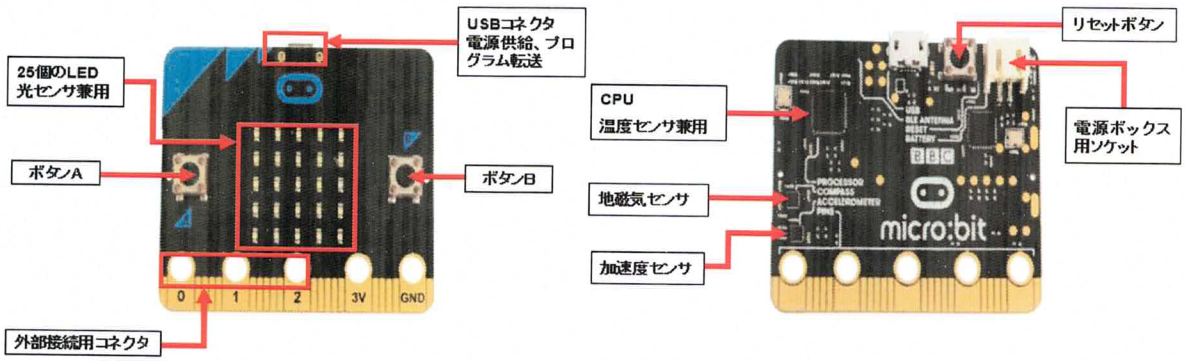


図 1-1　micro:bit(実習で利用する micro:bit)

micro:bit のハードウエア機能としては、

・25 個の LED（表示、センサ）、光、温度、加速度計などのセンサ
・プログラムができるスイッチボタン（2 個）
・Bluetooth による無線通信
・物理的に接続するための端子

などがあります。さらに、以下のような特徴があります。

・ビジュアル言語で、簡単な操作で利用できる
・シミュレータがついている
・JavaScript、Python に自動変換できる

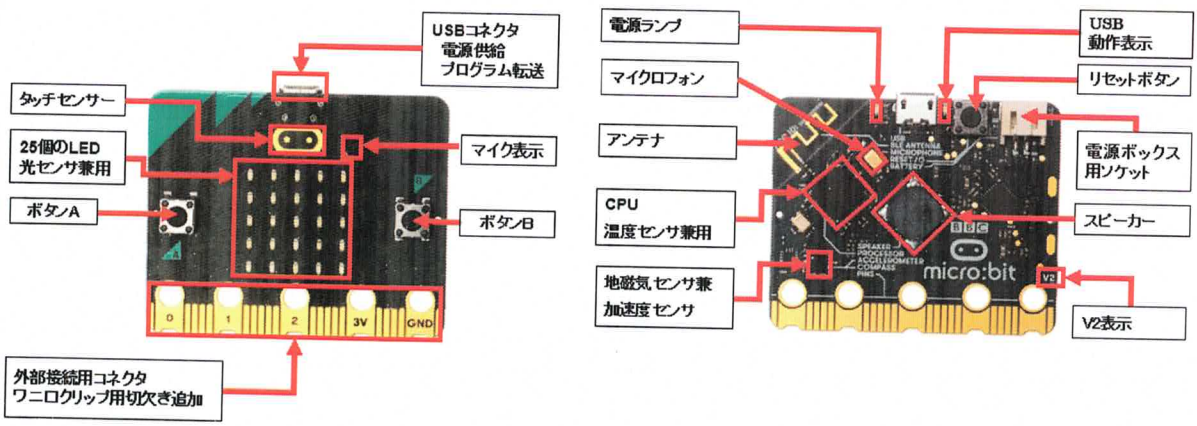


図 1-2　　micro:bit V2（2021 年８月以降、販売されているバージョン）

【参考資料】
https://microbit.org/ja/new-microbit/
https://tech.microbit.org/hardware/

2．エディタによるプログラムの作成

インターネットに接続し，Webブラウザで、Microsoft MakeCode for micro:bitのWebサイト（https://makecode.microbit.org/）に入ると、図2-1のような画面（ホーム）が表示されます。ここで、「新しいプロジェクト」をクリックすると、「プロジェクトを作成する」ダイアログで、プロジェクト（プログラム）の名前（ここでは、prei1-1）をつけて、「作成」をクリックします。すると、micro:bit用のエディタ（MakeCode）やシミュレータの機能があるシミュレータ画面が表示されます(図2-2)。

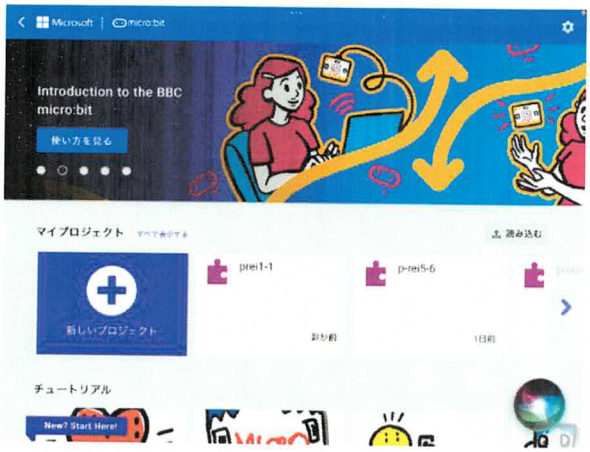


図2-1　新しいプロジェクト

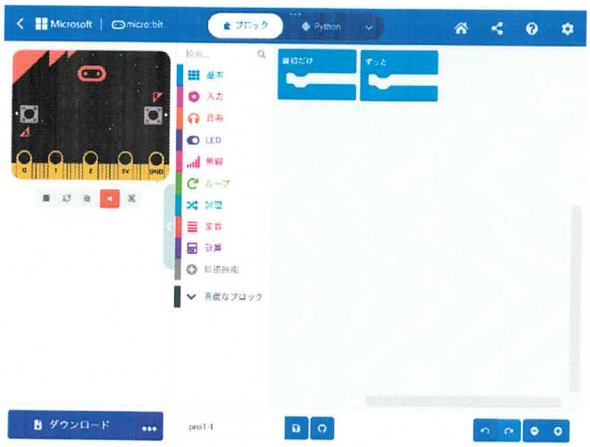


図2-2　シミュレータ画面

図2-3　シミュレータ画面の名称

シミュレータ画面の名称は、図 2-3 に示す通りで、それぞれの概要は、以下の通りです。

［ツールボックス］
基本、入力、音楽、LED、無線、ループ、論理、変数、計算、そして、高度なブロックがあり、それぞれのツールをクリックすると、利用できるブロックが表示される。
［プログラミングエリア］
ツールボックスで選択したブロックをエリア内にドロップすることによって、プログラムが作成できる。「最初だけ」「ずっと」のブロックが、最初に置かれている。
［ホーム］
「ホーム」を選択すると、新しいプロジェクトの場合には、名前を付けることができる。最初に名前をつけていれば、最初の画面に戻る。
注）最初に、名前をつけていない場合は、「題名未設定」となる。
［ブロック］
「ブロック」を「JavaScript」や「Python」に切り替えることによって、「ブロック」で書かれたプログラムをそれぞれの言語で表示することができる。
［ダウンロード］
「ダウンロード」では、プログラムを micro:bit に書き込み(ダウンロード)することができる。また、「…」を開くと、ダウンロードのオプションを指定できる。「FD のアイコン」では、プロジェクト名のついたプログラムをファイルとして、指定した場所に保存することができる。
［シミュレータ］
micro:bit の画面の下には、プログラムを四角ボタン（■）で停止、三角ボタン（▶）で開始できる。そのほか、再起動、デバッグモードの切り替えなどができる。

新しいプロジェクトを作成すると、プログラミングエリアには、「最初だけ」ブロックと「ずっと」ブロックが、最初に置かれています。図 2-3 では、不要なブロックである「ずっと」ブロックは、ツールボックスへ、ドラッグ＆ドロップして削除しています。
そして、
・ツールボックスの「基本」をタップし、「LED 画面に表示」ブロックを、ドラッグ＆ドロップで、プログラミングエリアに移動し、「最初だけ」ブロックにつなげる。
・LED をタップ（光の ON/OFF が切り替わる）して、ハート形に見えるように LED を ON にする。
を行っています。

＜MakeCode エディタ＞
Windows からは、下記の Web サイトにアクセスすれば、ホーム（図 2-1 に同じ）画面が表示されます。
https://makecode.microbit.org/