プログラミング資料（テーマ１）

コンピュータとじゃんけんをしてみよう

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

日時：2023 年 6 月 25 日(日)

場所：京田辺市立中央公民館 2F　研修室(第 3、第 4)

＜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/

主催　NPO 法人 学習開発研究所

後援　京田辺市教育委員会

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．コンピュータとじゃんけん

4-1　アイコン（グー、チョキ、パー）表示（分岐３つ）（プログラム prei4-1）

「ずっと」ブロック

＊「変数」－「変数を追加する」　→　「c」にする

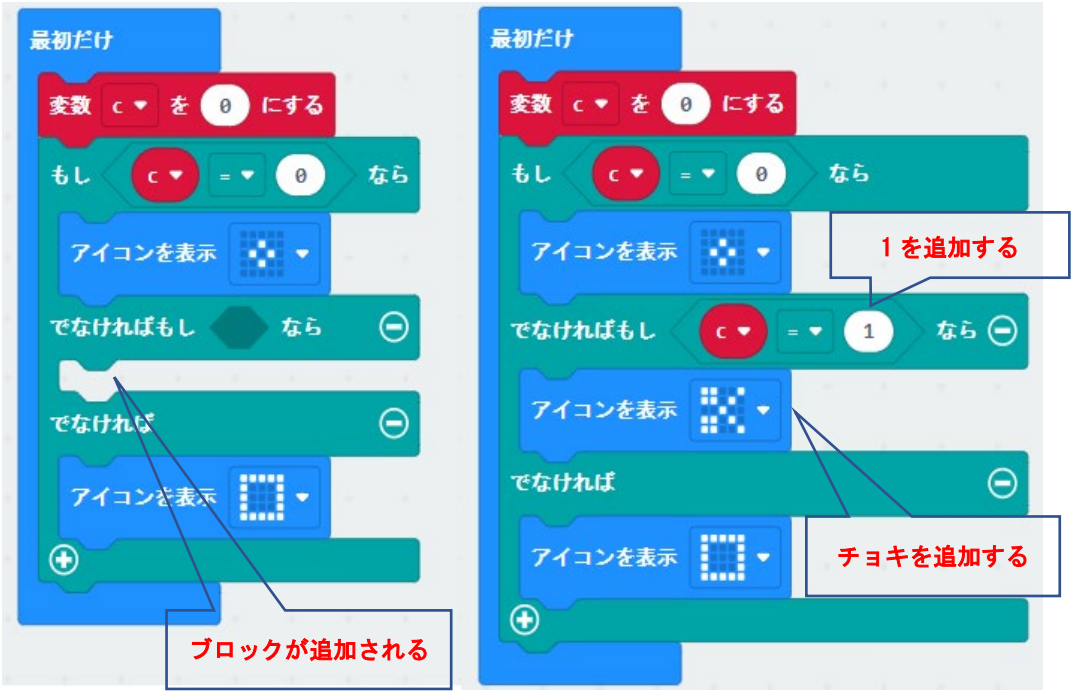
＊「変数」－「変数ｃを０にする」

＊「論理」－「条件判断」（もし　なら～　でなければ～）で、「⊕」をクリックする

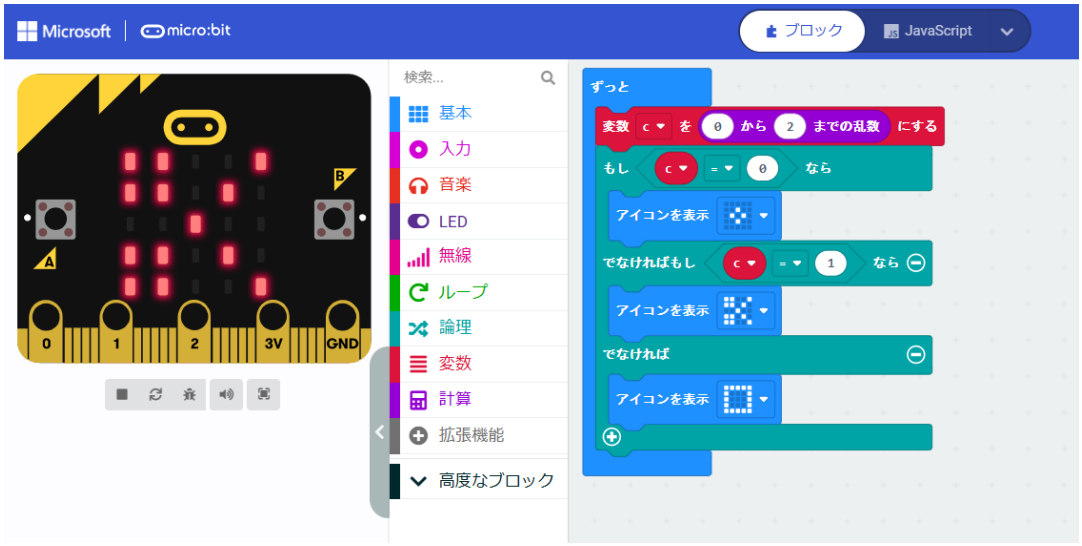
（もし　なら～　でなければもし　なら～　でなければ～）のブロックになる

＊「でなければもし　なら」の箇所、「ｃ　＝▼　１」　を追加する

＊「基本」－「アイコン表示」で、「チョキ」を追加する



4-2　アイコン（グー、チョキ、パー）表示（乱数の利用）（プログラム prei4-2）



「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→　2（乱数は 0、1、2 のいずれか）

4-3　コンピュータの手（繰り返しは 5 回）（プログラム prei4-3）

「最初だけ」ブロック

＊「基本」－「アイコン表示」で、「グー」を表示する

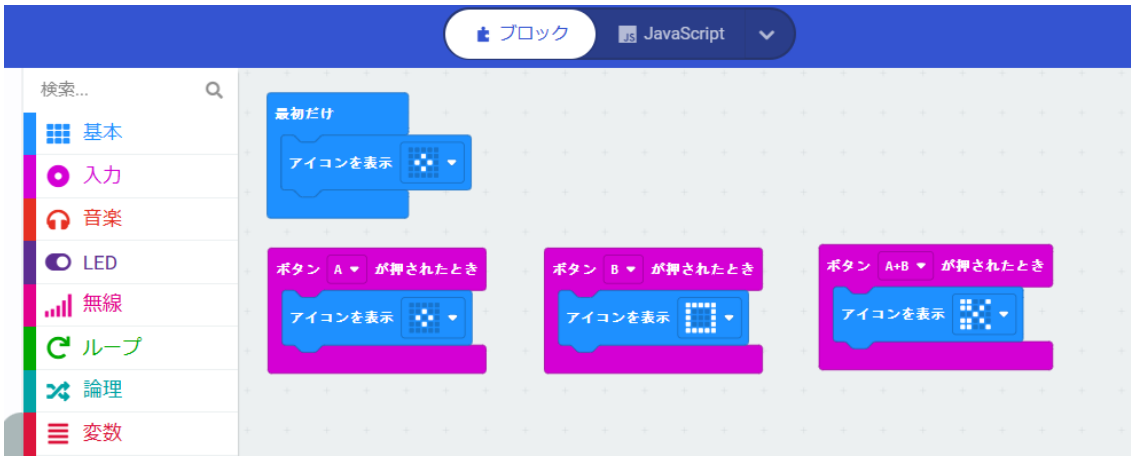
＊「カウンター」（0～4）で、5 回繰り返す

＊「基本」－「数を表示」で、カウンターの数値（回数）を表示する

＊自分の番の時は、「矢印を表示」で「右向き」（→）を表示して、2 秒待つ

＊コンピュータの時は、「文字列を表示」で「Com」と表示する

4-4　自分の手（スイッチボタンを利用）（プログラム prei4-4）



「最初だけ」ブロック

＊「基本」－「アイコンを表示」で「グー」を表示する

＊「入力」－「ボタン A が押されたとき」を選択する

＊A ボタンでは「グー」、B ボタンでは「パー」、A+B ボタンでは「チョキ」とする。

＜参考＞　勝敗の判定表

じゃんけんの勝敗を判定するプログラムを考えてみよう。

A さん、B さんの 2 人の勝敗を判定する表は、下の通りです。まずは、この表の「判定」欄から右列の欄の意味を考えてみよう。

種類	数値	A	B	判定	（A－B）の値	（A-B+3）/3 の余り
グー	0	0	0	引分け	0	0
		0	1	A	-1	2
		0	2	B	-2	1
チョキ	1	1	0	B	1	1
		1	1	引分け	0	0
		1	2	A	-1	2
パー	2	2	0	A	2	2
		2	1	B	1	1
		2	2	引分け	0	0

＜参考文献＞

高橋、喜家村、稲川：micro:bit で学ぶプログラミング、pp.14-15、pp.41-43、コロナ社(2019)

＜参考＞「グー」「チョキ」「パー」の出す回数（プログラム prei4-5）

コンピュータの手を 100 回実施したとき、「グー」「チョキ」「パー」の出す回数(実際は、乱数の出す数値(0,1,2)の回数)を調べるプログラムを考えてみよう。下のプログラム例は、「ブロック」から「JavaScript」欄の「Python」を選択し、表示したものです。


```
1  c = 0
2  k0 = 0
3  k1 = 0
4  k2 = 0
5  for カウンター in range(100):
6      c = randint(0, 2)
7      if c == 0:
8          k0 += 1
9      elif c == 1:
10         k1 += 1
11     else:
12         k2 += 1
13 basic.show_string("G=")
14 basic.show_number(k0)
15 basic.clear_screen()
16 basic.show_string("C=")
17 basic.show_number(k1)
18 basic.clear_screen()
19 basic.show_string("P=")
20 basic.show_number(k2)
21 basic.clear_screen()
```

このプログラムを作成してみよう。

新しいプロジェクト(例えば、「sankou」とする)を作成し、「JavaScript」欄から「Python」を選択し、まずは、このプログラムの 12 行目までを入力し、エラーがでれば修正し、エラーがでなくれば、ブロックで表示してみよう。

つぎに、見やすい結果の表示方法を考えてみよう。右は、先のプログラムの「ブロック」での「回数」を表示した例です。

プログラミング資料（テーマ 2）

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

日時：2023 年 6 月 25 日(日)

場所：京田辺市立中央公民館 2F　研修室(第 3、第 4)

＜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/

主催　NPO 法人 学習開発研究所

後援　京田辺市教育委員会

Ⅰ. micro:bit のプログラム

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

「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）

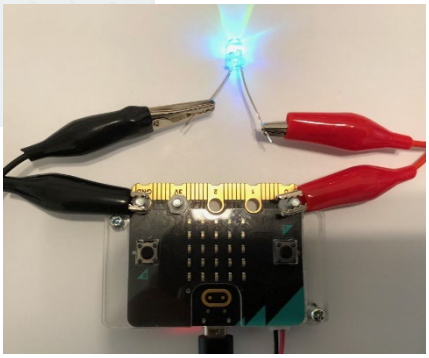
「ずっと」ブロック

＊「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

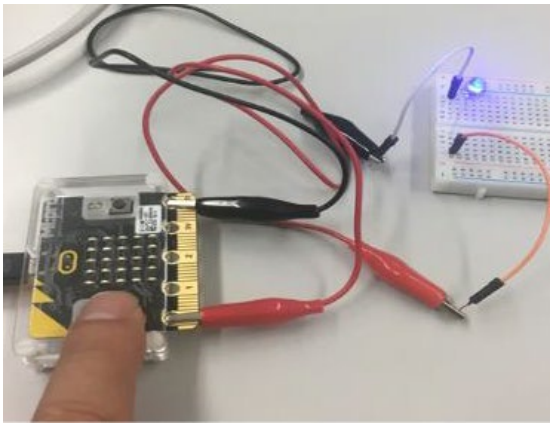
4-3　スイッチによる LED の点灯（プログラム p-rei4-3）

「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)

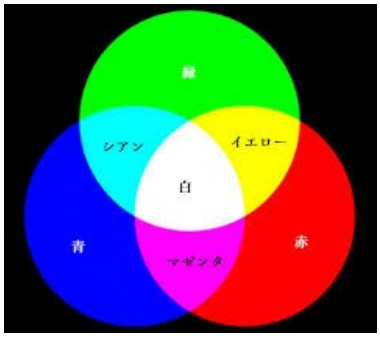
５．フルカラーLED の制御

【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。

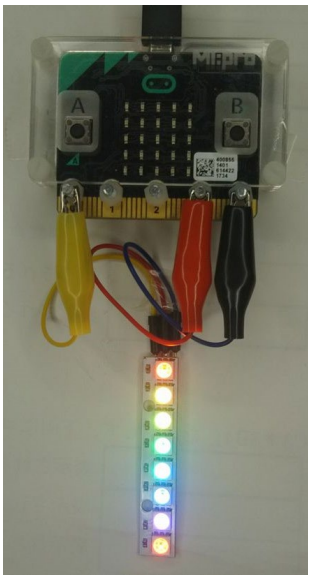
なお、R:255、G:255、B:255 では白、R:0、G:0、B:0 では黒になります。

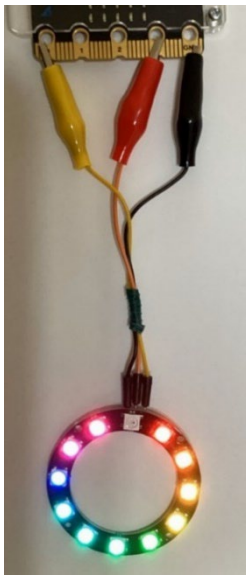


光の３原色　R(赤) G(緑) B(青)

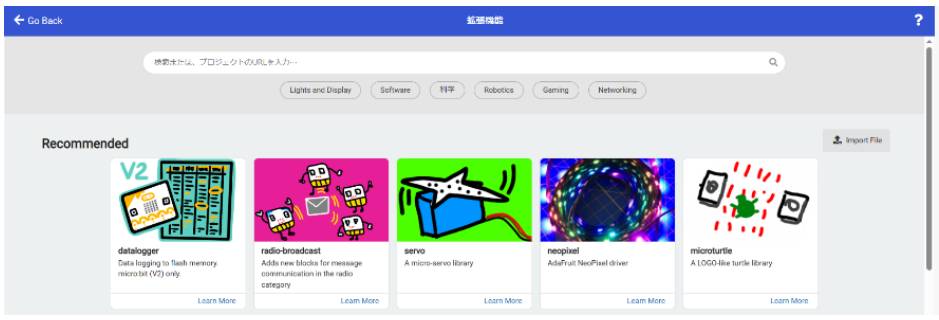
【micro:bit と Neopixel の接続】

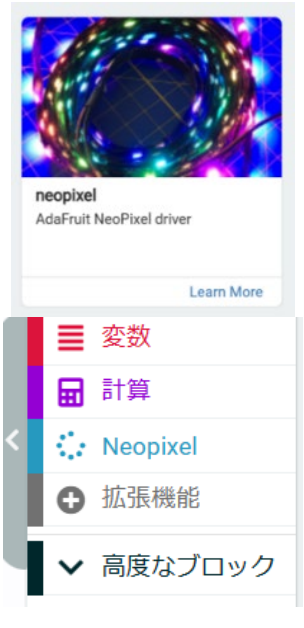
micro:bit と Neopixel を右図のように接続します（黄色は P0 端子,赤は 3V 端子,黒は GND 端子)。右図の Neopixel は、８個の LED が棒状（Stick 型）に接続された製品です。その他にも、Neopixel には、リング状の製品があります。



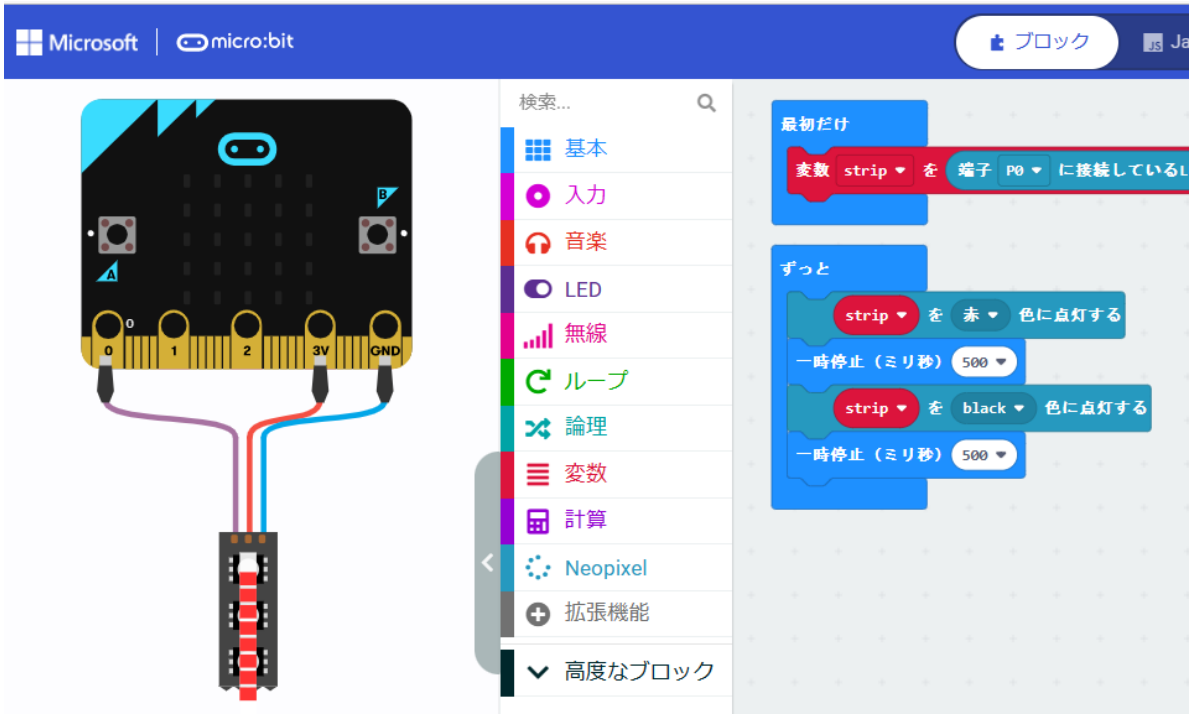


Neopixel を利用するには、ライブラリが必要です。ライブラリは、ツールボックスの下にある「⊕拡張機能」をクリックすると、拡張機能の一覧が表示されるので、「Neopixel」を選択する。ツールボックスの「計算」の下に、「Neopixel」のブロックが追加されます。





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」

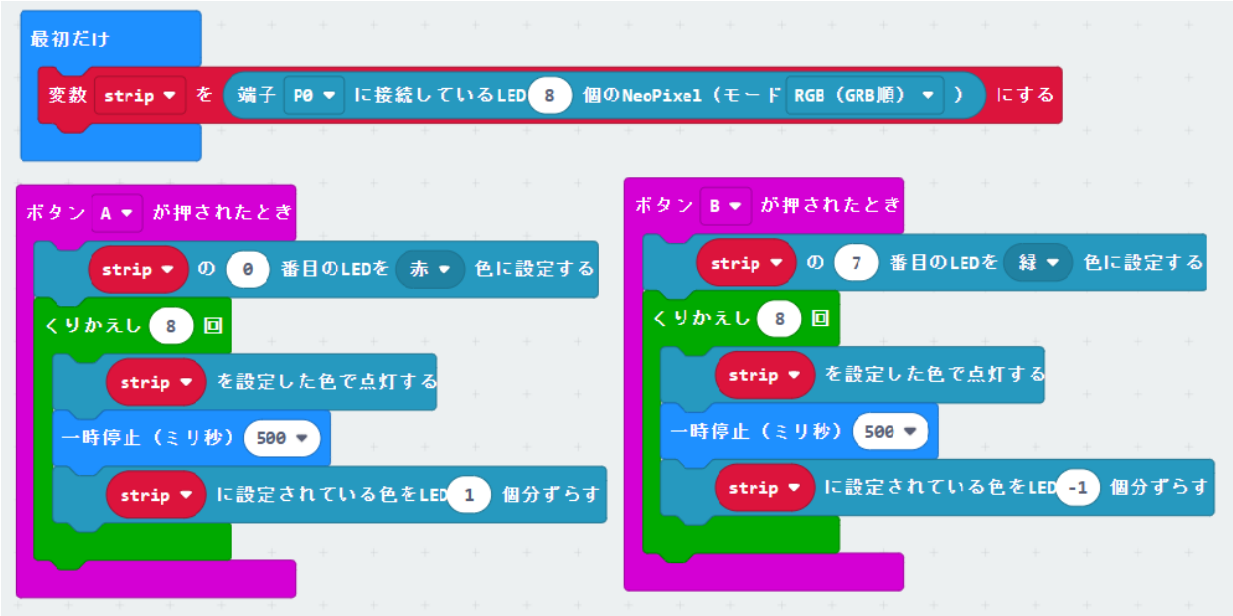
＊前の例題の「ずっと」ブロックを変更する。

「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）

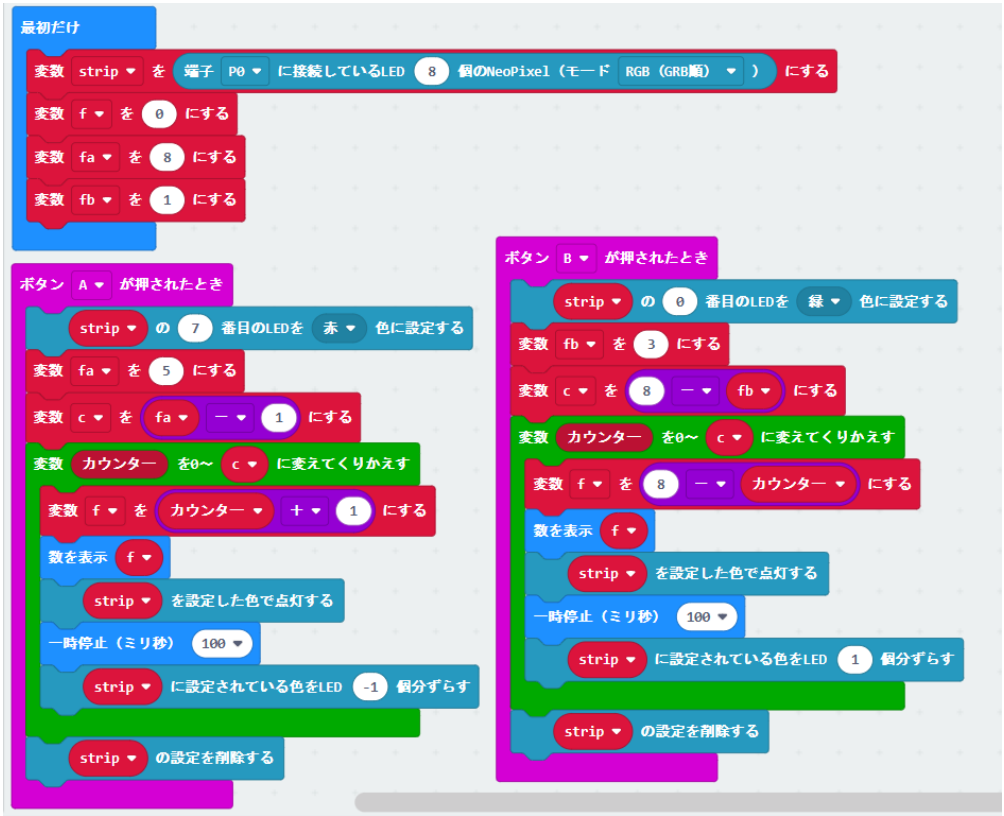
１階から８階までの建物で、エレベータで移動できるものする。次のような場合、Neopixel で表示するプログラムを考えてみよう。

・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。



プログラミング資料

micro:bit による iPad での操作

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

日時：2023 年 6 月 25 日(日)

場所：京田辺市立中央公民館 2F　研修室(第 3、第 4)

主催　NPO 法人 学習開発研究所
後援　京田辺市教育委員会

1. micro:bit 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。

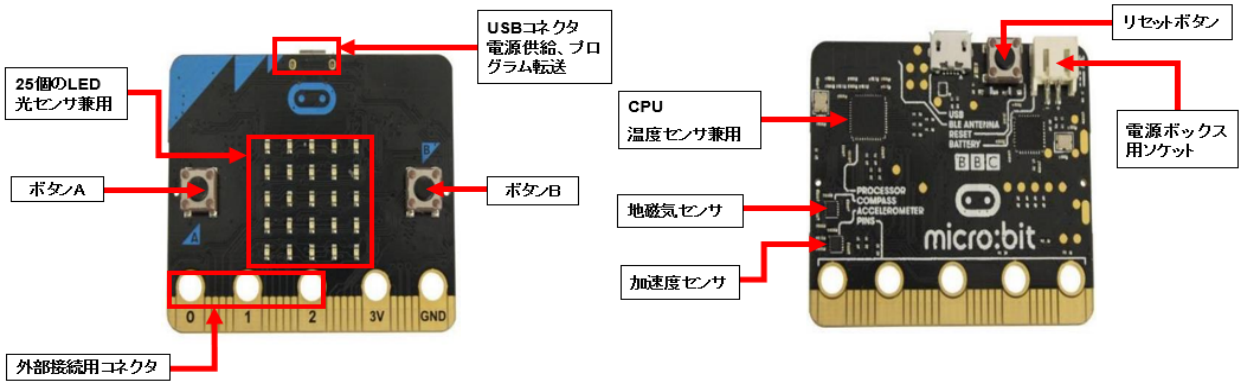


図 1-1　micro:bit(実習で利用する micro:bit)

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

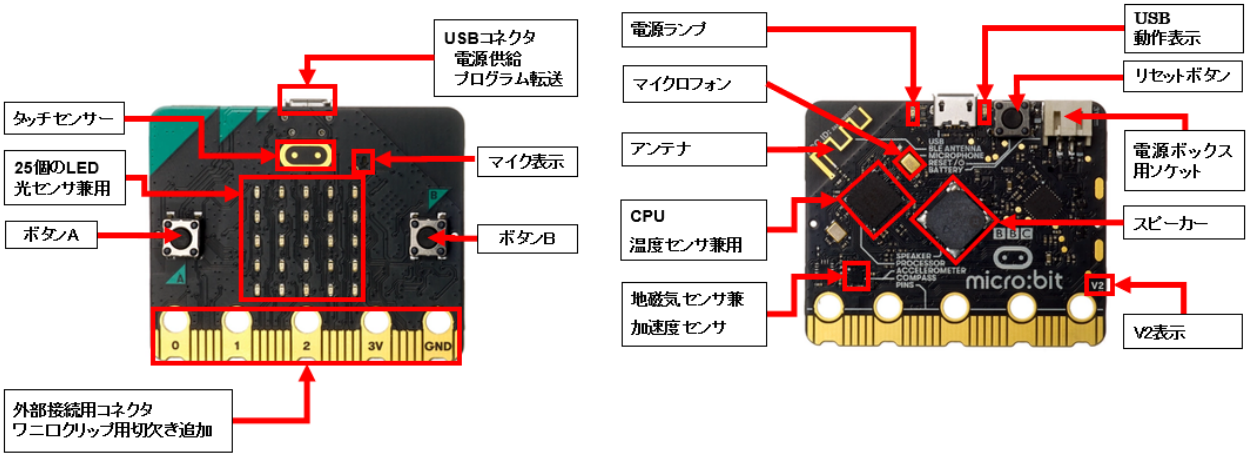


図 1-2　　micro:bit V2（2021 年 8 月以降、販売されているバージョン）

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

2. iPad での micro:bit の利用

iPad の「App Store」から、micro:bit を検索して(図 2-1)、インストールします。一度、インストールされていれば、「開く」をクリックすると、micro:bit のアプリのメニューが表示されます（図 2-2)。micro:bit のアプリのメニューには、5 つの項目があり、ここでは、上の 3 つの項目を利用します。右上の「ヘルプ」をクリックすると、このメニューの説明や注意事項が書かれています。

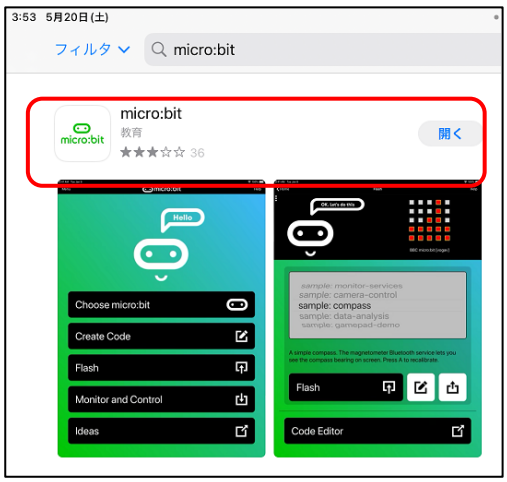


図 2-1　micro:bit のアプリ

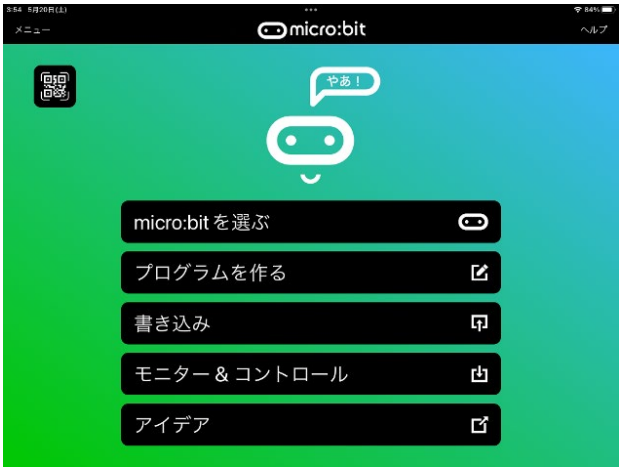


図 2-2　micro:bit のアプリのメニュー

次に、micro:bit を、iPad の Bluetooth 機能を利用して接続するために「ペアリング」を行います。

注）最初に、iPad の「設定」-「Bluetooth」で、オンになっていることを確認しておく。

図 2-2 の micro:bit のアプリのメニューから「micro:bit を選ぶ」をタップすると、現在、iPad に接続されている micro:bit があれば、名前（5 文字、パターン）が表示されます(図 2-3)。ここでは、新しい micro:bit を接続するので、図 2-3 の一番下のメニュー「新しい micro:bit をペアリング」をタップします。すると、図 2-4 のペアリングモードになります。

【ステップ 1：ペアリングモード】

micro:bit の A と B ボタン(表側)を押したまま、リセットボタン(裏側)を押します。一旦、micro:bit の LED(25 個)がすべて点灯し、そのあと「パターン」が表示されます。micro:bit に「パターン」が表示されたら、「次」をタップします。

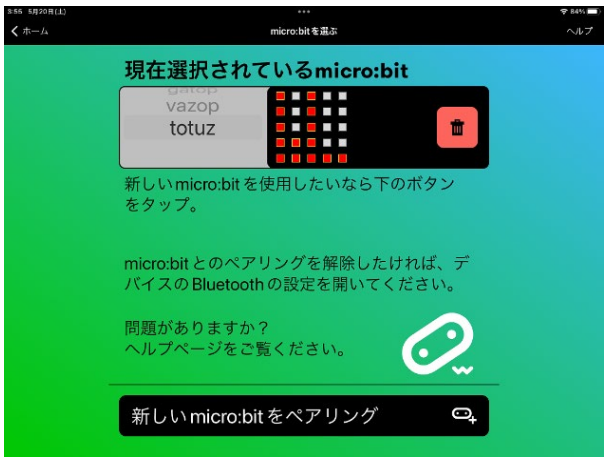


図 2-3　micro:bit のペアリング

図 2-4　ペアリングモード

【ステップ 2：パターンの入力】

次に、図 2-5 のようなパターン入力の画面が表示されれば、micro:bit の LED に表示されているパターンと同じように入力し(図 2-6 は入力済み)、 ペアリングの準備が完了します。A ボタンを押して、「次へ」でタップします。

注）LED の各列のパターンの一番上の□をクリックすれば、その下のパターンは、すべて入力されます。

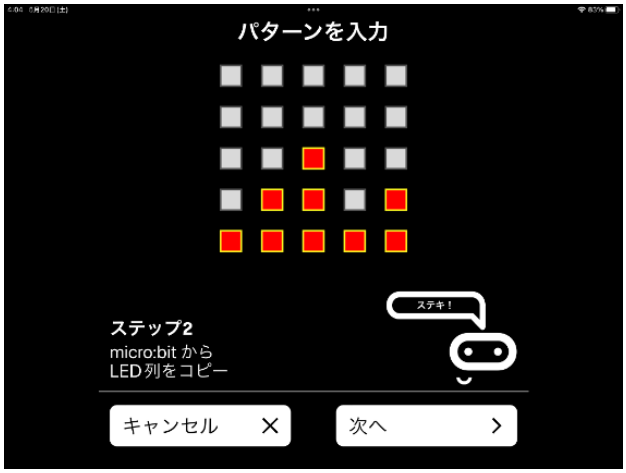


図 2-5　パターンの入力

図 2-6　ペアリングの準備完了

図 2-7 に示すようにペアリングに成功すれば、micro:bit のリセットボタンを押してから「OK」をタップします。新しくペアリングされた micro:bit の名称（ここでは、「zoguv」）とパターンが、現在選択されている micro:bit として表示されます（図 2-8)。

図 2-7　ペアリングの完了

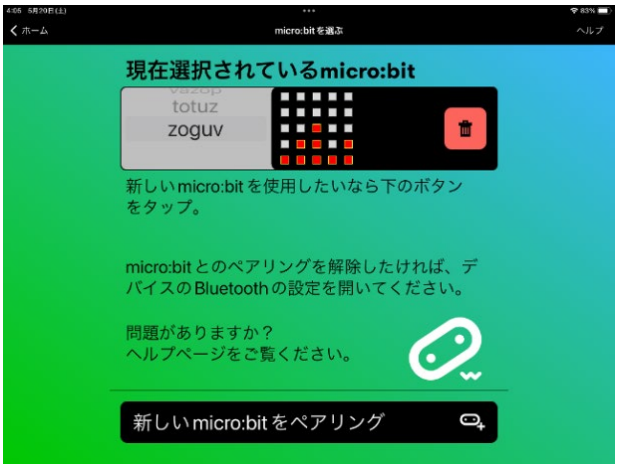


図 2-8　選択されたされた micro:bit

3. エディタによるプログラムの作成

図 2-2 の micro:bit のアプリのメニューから、「プログラムを作る」をタップすると、図 3-1 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」をタップすると、「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」をタップします。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 3-2)。

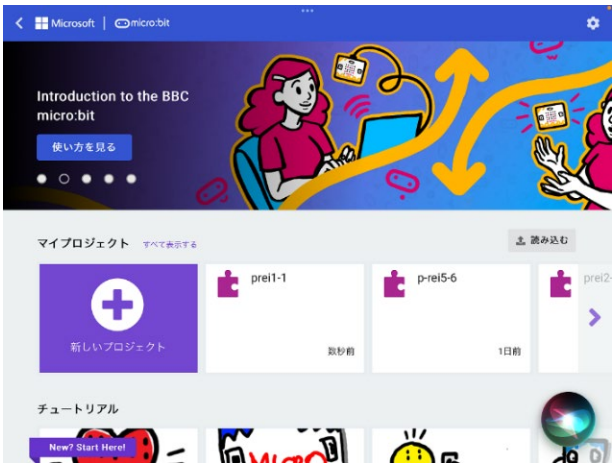


図 3-1　新しいプロジェクト

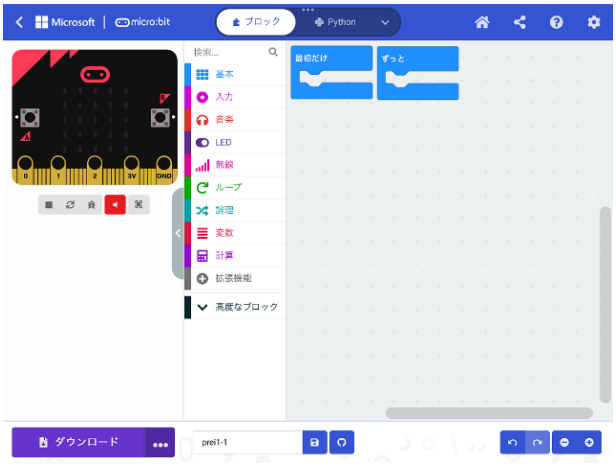


図 3-2　シミュレータ画面

図 3-3　シミュレータ画面の名称

注）iPad の画面の向きを、横から縦にすると、シミュレータ画面の表示位置が変わる。

シミュレータ画面の名称は、図 3-3 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 3-3 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 3-1 に同じ）画面が表示されます。
https://makecode.microbit.org/

４．プログラムの micro:bit へダウンロード

図 2-2 の micro:bit のアプリのメニューから「書き込み」をタップすると、図 4-1 の画面が表示されます。なお、図 3-3 の画面で、「保存」(FD のアイコン）をタップしても、図 4-1 の画面が表示されます。

注）書き込み先の micro:bit は、あらかじめペアリングモード（A+B を押した状態で、micro:bit のリセットボタンを押す）にしておきます。

次に、プログラム名（ここでは、prei1-1）を選択して、「書き込み」をタップします(図 4-1)。書き込みを行う micro:bit（ここでは、「zoguv」）を検索し(図 4-2)、見つからない場合は、ペアリングモードのリセットを行います(図 4-3)。

「続行」を選択すると、「通信中・・・」と表示され、少ししてから書き込みが終われば、図 4-4 の終了画面が表示されます。

図 4-1　書き込みの画面

図 4-2　micr:bit の検索

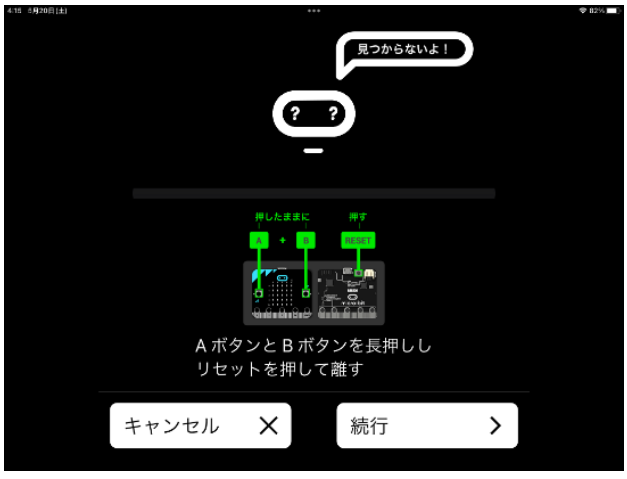


図 4-3　ペアリングモード

図 4-4　書き込みの終了

５．　外部ファイルの読み込み

ホーム画面の右にある「読み込む」(図 5-1 の赤での囲み)を選択すると、図 5-1 に示すようなダイアログが表示されるので、左の「ファイルを読み込む・・・」の箇所をタップすると、図 5-2(a)に示すように、ファイルを選択するダイアログが表示されます。図 5-2(a)の「ファイルを選択」の箇所をタップすると、図 5-2(b)が表示されるので、上から 3 つ目の「ファイルを選択」の箇所をアップします。

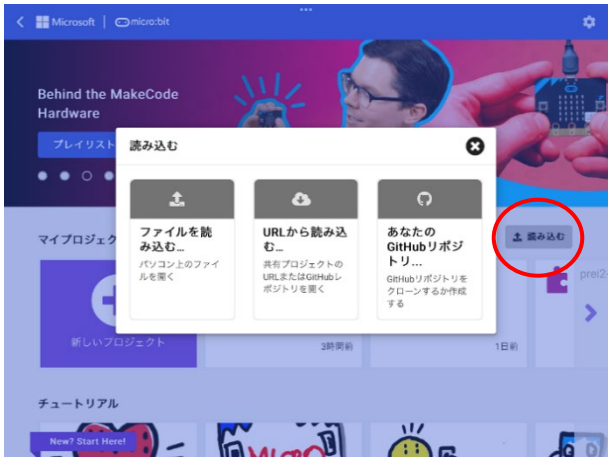
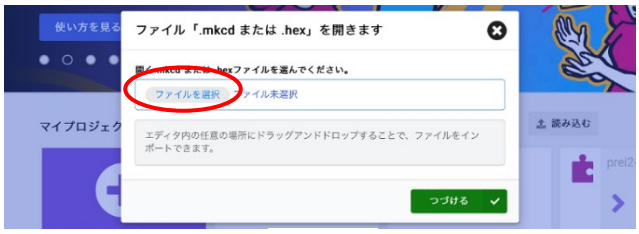
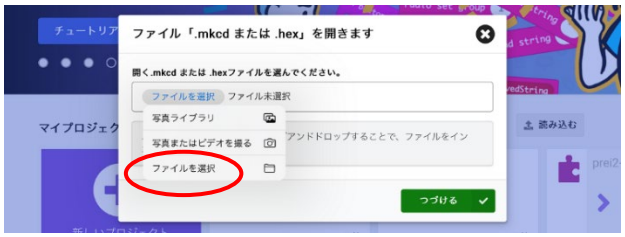


図 5-1　ファイルの読み込み



(a)



(b)

図 5-2　ファイルの選択

次に、ファイルが書き込まれている場所、図 5-3 では、左のメニューにある「iCloud Drive」「この iPad 内」「Google ドライブ」「OneDrive」のなかから選択します（図 5-3)。図 5-3 では、「OneDrive」をタップし、その「ファイル」から保存されている場所をタップすると、ファイル名が表示されます(図 5-4)。この中から、読み込みたいファイルをタップすると、図 5-2（a）に、ファイル名が表示されます。

注）ファイルが書き込まれている場所（ここでは、OneDrive）の読み込める状態になっているものとしています。

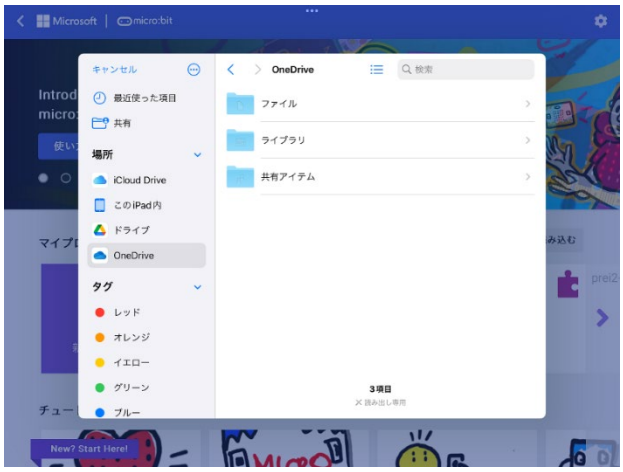


図 5-3　ファイルの格納場所

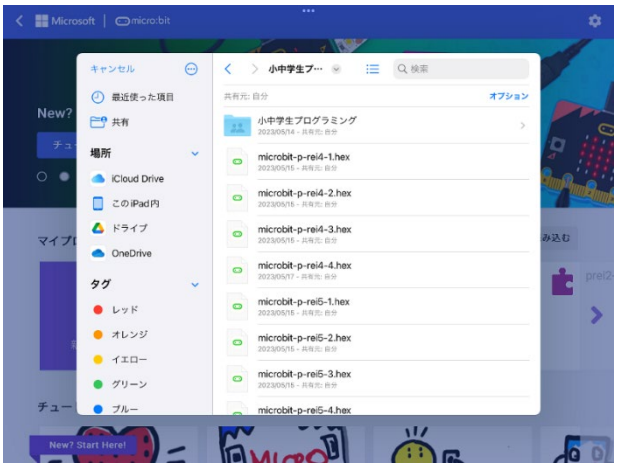


図 5-4　OneDrive を選択