


micro:bitによるプログラミング(1)

担当: 高橋参吉(NPO法人 学習開発研究所)

実習内容

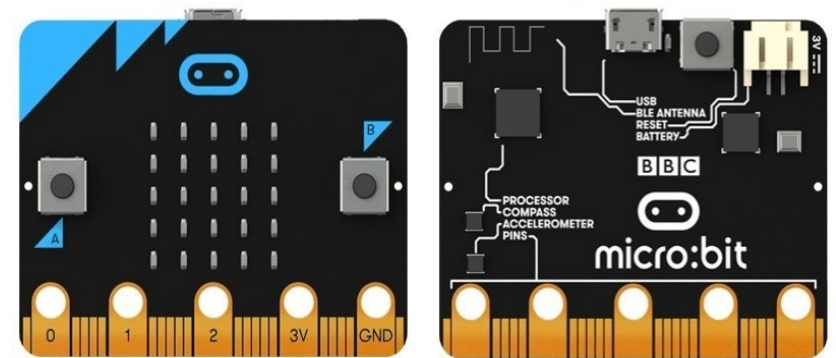
- micro:bitの特徴
- micro:bitの基本動作
- 「」マークの表示、点滅

引用・参考文献

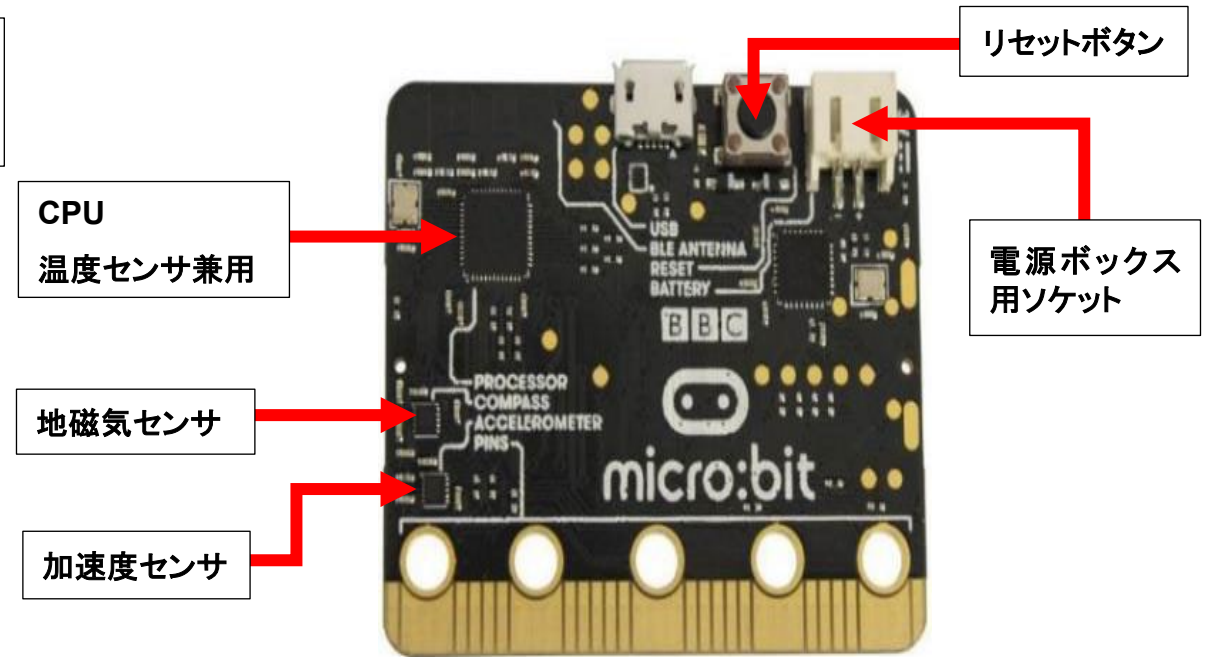
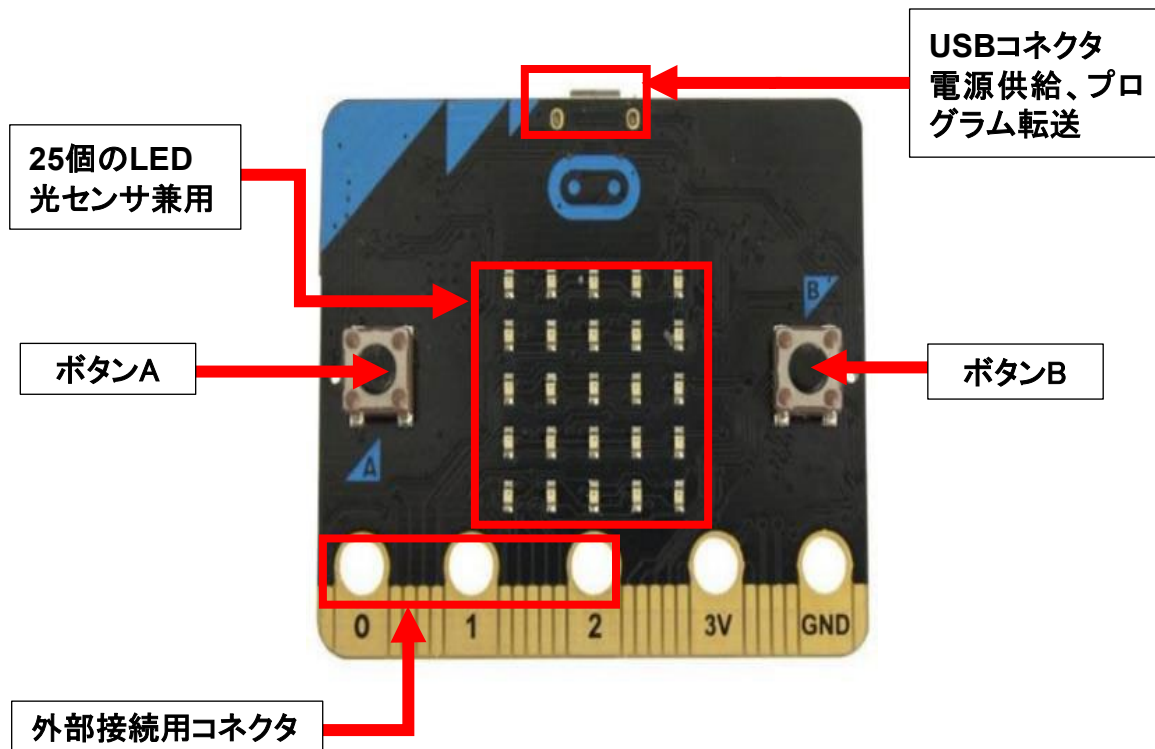
高橋参吉、喜家村奨、稲川孝司: micro:bitで学ぶプログラミング ブロック型からJavaScriptそしてPythonへ、コロナ社、(2019.9).

micro:bitの特徴

- イギリスBBC(英国放送協会)が開発
- micro:bit教育財団が7年生(11~12歳)を対象に無料配布
- 手のひらサイズの安価なコンピュータ
- 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
- 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

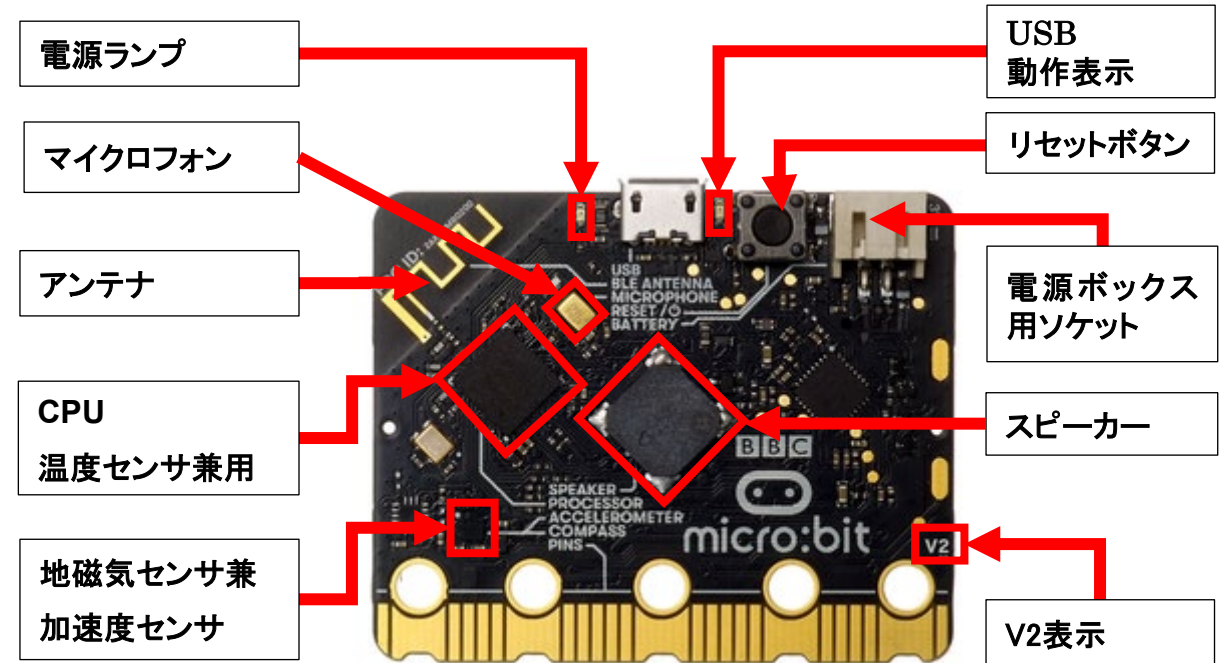
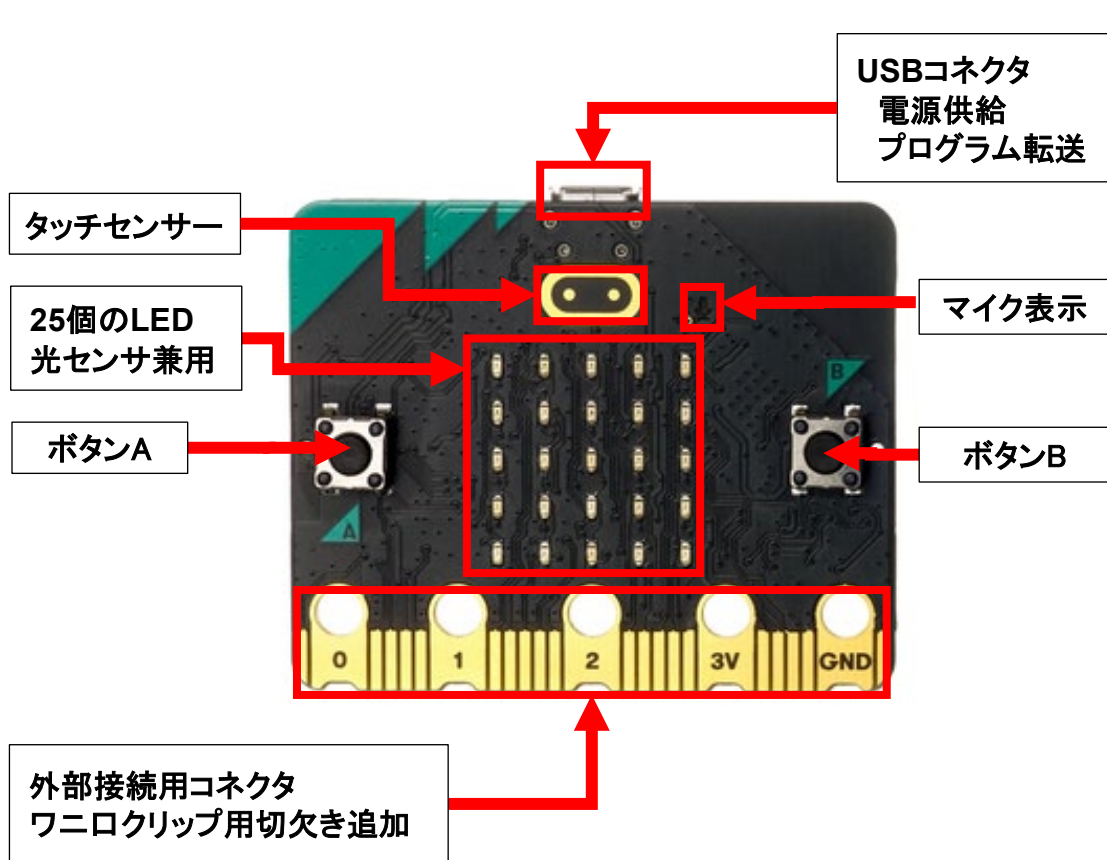


micro:bit(センサ、ボタン、コネクタ)



micro:bit V2 (センサ、ボタン、コネクタ、マイク、スピーカ)

(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/>

micro:bitの基本操作

- 下記のWebサイトへアクセスする

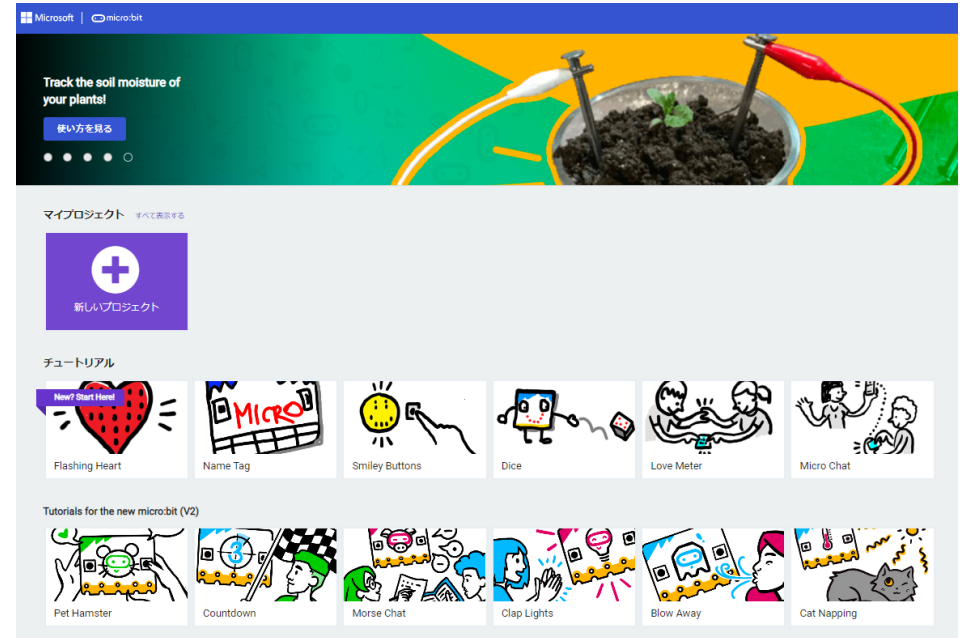
<https://makecode.microbit.org/>

<https://makecode.microbit.org/#editor>

*IEは、使えない

- 「マイプロジェクト」、「チュートリアル」、「ゲーム」などがある。
- 「マイプロジェクト」から、新しいプロジェクトを作成する。
- 「プロジェクトを作成する」ダイアログで、名前(rei1-1)をつける。

ここで、名前をつけていない場合は、「題名未設定」となるが、あとで、変更できる。



micro:bitのシミュレータ画面



シミュレータ画面の説明

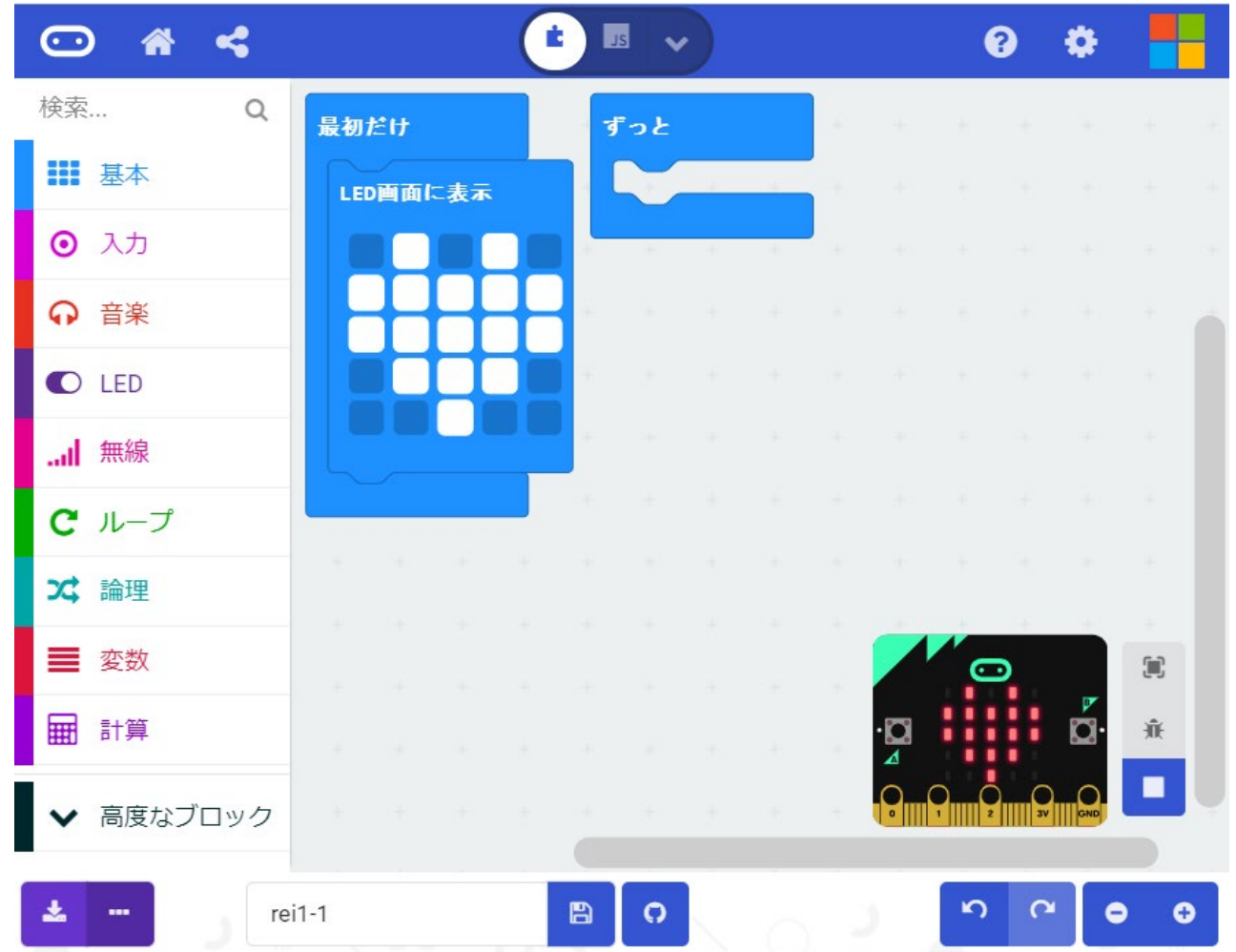
- **[ツールボックス]**には、基本、入力、音楽、LEDなどがあり、それぞれのツールをクリックすると、利用できるブロックが表示される。
- **[プログラミングエリア]**では、ツールボックスで選択したブロックをエリア内にドロップすることによって、プログラムが書ける。「最初だけ」「ずっと」のブロックが、置かれている。
- **[ホーム]**を選択すると、最初に名前をつけていれば、最初の画面に戻り、名前をつけていなければ、プロジェクトに名前をつけて保存できる。
- **[ブロック]**では、「JavaScript」や「Python」に切り替えることによって、「ブロック」プログラムをそれぞれの言語で表示することができる。
- **[ダウンロード]**では、パソコンやmicro:bitにプログラムをダウンロードしたりすることができる。
- **[シミュレータ]**には、表示画面のほか、プログラムを四角ボタン(■)で停止、三角ボタン(▶)で開始、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。

例題1 「♡」マークの表示

<手順>

- 「新しいプロジェクト」を選択する。
- ツールボックスの「基本」をクリックし、「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をクリック(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。

* 不要なブロックは、ツールボックスヘドラッグ&ドロップすると削除できる。



横幅を狭くすると、シミュレータ画面が右下に表示される。

例題1-1 「♡」マークの点滅

<手順> (ファイル名: rei1-1)

- 「最初だけ」ブロックを「ずっと」ブロックに変更する。
- 「基本」から「一時停止(ミリ秒)」ブロックをつなぎ、数値を100から500に変えておく。
- 「基本」から「表示を消す」ブロックをつなぐ。
- 「一時停止(ミリ秒)」ブロックをつなぎ、数値を100から500に変えておく。



横幅を狭くすると、シミュレータ画面が右下に表示される。

パソコンへのプログラムの保存 (プログラム名:rei1-1)

- ダウンロードの右のアイコンをクリックすると、ダウンロード(ファイル名は、「microbit-rei1-1.hex」)できる。

テキストファイルではなく、hexファイルであることに注意!!

- ダウンロードされたファイルをドラッグ&ドロップで保存したいフォルダに保存する。



*ファイル名は、ダウンロードの回数で異なる

micro:bitへのプログラムのダウンロード

- micro:bitをパソコンのUSBに接続した後、ダウンロードファイルを、micro:bitが接続されているドライブ（例えば、MICROBIT (D:)）に、ドラッグ&ドロップすると、プログラムをダウンロードできる。
- micro:bitをパソコンのUSBに接続した後、ギアをクリックした後「デバイスを接続する」(右図)。
- micro:bitのペアリングを実施すれば、「ダウンロード」をクリックすると、micro:bit へのプログラムのダウンロードが行える。



注意事項

- 2020年11月20日に、MakeCodeがmicro:bit v2対応版に変更になる。
- 2020年11月23日公開版に、ブロック「入力」や「音楽」に、micro:bit(v2)」が追加される。
- 2021年7月10日公開版(2021年版)では、拡張機能で「Datalogger」を検索すると、micro:bit(V2専用)のデータロギング機能を追加できる。

制作日:2020年9月11日

更新日:2020年11月22日

更新日:2021年8月25日

更新日:2022年8月22日

micro:bitによるプログラミング(2)

担当: 高橋参吉(NPO法人 学習開発研究所)

実習内容

- プログラミングの基礎(順次、繰り返し)
- スイッチボタンの利用～じゃんけんゲーム～
- プログラミングの基礎(分岐)
- 光センサによる学習指導

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).

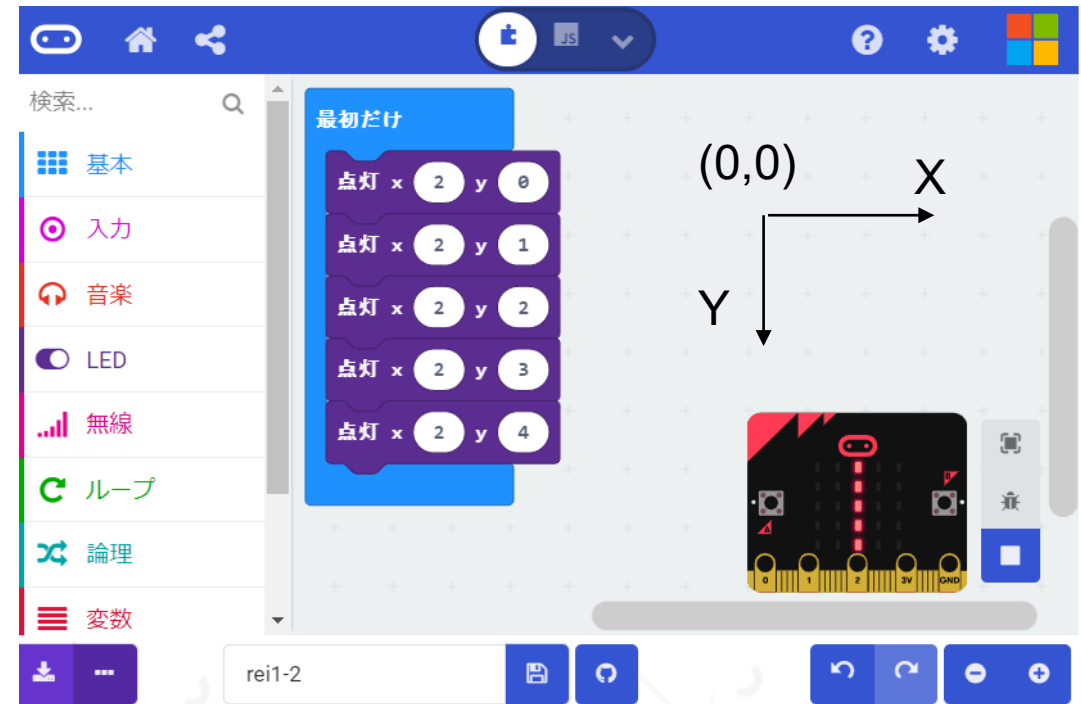
プログラムの基礎(順次、繰返し)

【例題1-2】 次のプログラムを作成して、図のように表示されることを確かめよう。
(ファイル名:rei1-2)

<手順>

- 1) 「基本」から「最初だけ」ブロックを選択する
- 2) 「LED」から「点灯」ブロックを選択し、xを「2」、yを「0」にする。左上のLEDの座標は(0、0)、右下のLEDの座標は(4、4)である。
- 3) 「点灯」ブロックにマウスをあて、マウスの右ボタンを押して複製を選択する。
- 4) 点灯ブロックを4回コピーし、xをすべて「2」、yを「1~4」にする。
- 5) 5つの点灯ブロックを「最初だけ」ブロックに接続し、動作を確認する。

このようなプログラムの構造を**順次構造**という。



横幅を狭くすると、シミュレータ画面が右下に表示される。

プログラムの基礎(順次、繰返し)

【例題1-3】 例題1-2のプログラムを、「ループ」から、繰返しのブロックを使って、プログラムを変更してみよう。(ファイル名:rei1-3)

<手順>

- 1) 点灯ブロックを「最初だけ」ブロックから削除する。
- 2) 「ループ」から「変数(index)を0~4に変えてくりかえす」ブロックを選択する。
- 3) 「変数(index)」の箇所を選択した後、「変数の名前を変更」を選択して、ダイアログが表示されるので「y」に変更する。
- 4) 「最初だけ」ブロックに接続する。
- 5) 「LED」から「点灯」ブロックを選択し、xの「0」を「2」に変更する。
- 6) 「変数」から「y」を選択し、「点灯」ブロックのyの「0」の上に置く。
- 7) 変更した「点灯」ブロックを「変数yを0~4に変えてくりかえす」ブロックに接続する。



このようなプログラムの構造を繰返し構造という。

プログラムの基礎(順次、繰り返し)

【例題1-4】 例題1-3で、点灯のx座標を変数「x」、y座標を変数「4-x」に変更して、図の形を確認してみよう。
(ファイル名:rei1-4)

<手順>

- 1) 「変数」から「変数を追加する…」を選択し、「x」を作成する。
- 2) 「y」の箇所を「x」に置き換える。
- 3) 「点灯」の「2」の箇所を「x」に置き換える。
- 4) 「計算」から「引き算」のブロックを選択する。
- 5) 計算式(4-x)を作成して、「点灯」のy座標に置く。



<デバッグモード>

スローモーションで、点灯の様子を確かめてみよう。

The screenshot shows the micro:bit IDE interface in Debug Mode. The top bar includes the 'micro:bit' logo, a 'ホーム' (Home) button, and the 'Debug Mode' indicator. The main workspace is divided into three sections:

- Left Panel:** A virtual representation of the micro:bit board with several red LEDs lit. Below the board are pins labeled 0, 1, 2, 3V, and GND. A 'デバッグモード' (Debug Mode) callout points to the orange bug icon in the bottom toolbar.
- Middle Panel:** A '変数' (Variables) table with a single entry: 'x: 5'. A 'スローモーション' (Slow Motion) callout points to the orange bug icon in the top toolbar.
- Right Panel:** A script editor showing a '最初だけ' (Start with) block containing two blocks: '変数 x を 0~4 に変えてくりかえす' (Set variable x to 0~4 and repeat) and '点灯 x x y 4 - x' (Turn on LED x at y=4). Both blocks are highlighted with a yellow border.

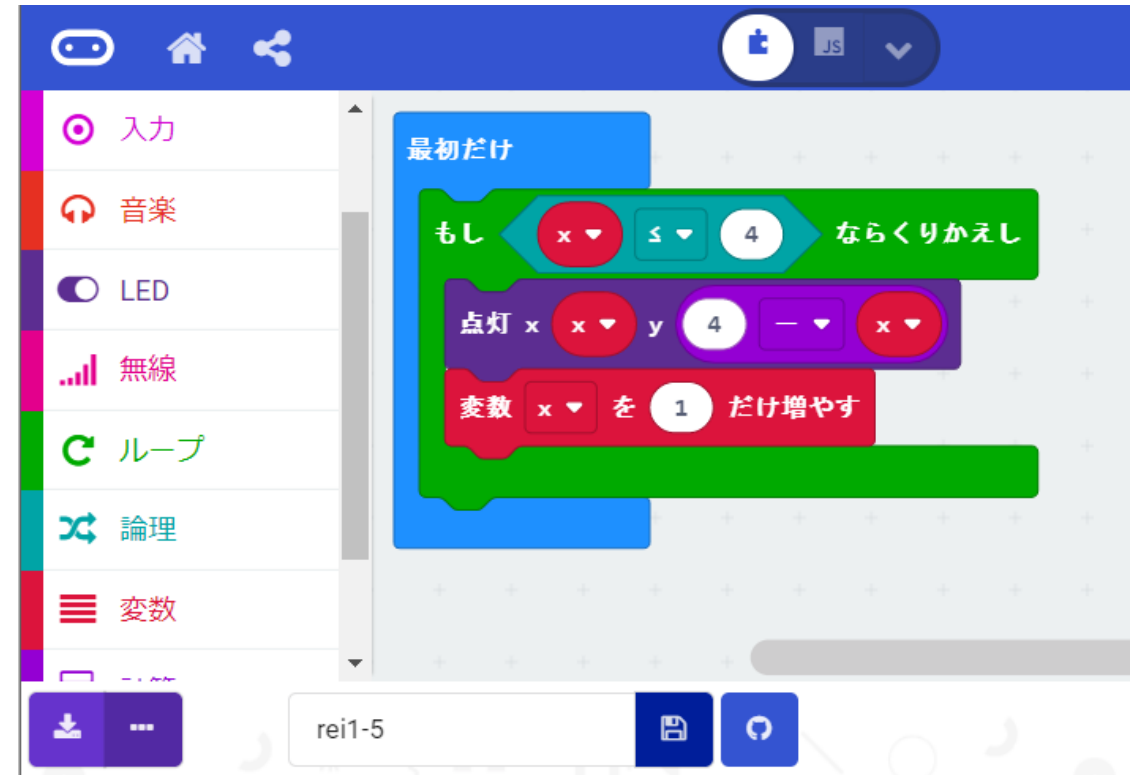
プログラムの基礎(順次、繰返し)(演習課題)

【例題1-5】 例題1-3のプログラムの繰返し「ループ」の箇所でブロックを変更したプログラムも同じように動くことを確認しよう。
(ファイル名:rei1-5)

<手順>

- 1) 「ループ」の「真ならくりかえす」を選択する。
- 2) 「論理」「くらべる」から「 $0 < 0$ 」を選択し、「 $x \leq 4$ 」を作成する。
- 3) 真の箇所を、「 $x \leq 4$ 」に置き換える。
- 4) 繰返し「ループ」の箇所を作成した「真ならくりかえす」に置き換える。
- 5) 「変数」から「変数xを1だけ増やす」を追加する。

繰返し回数がわからない時は、このループを使うとよい。



アイコンとスイッチボタンの利用

【例題1-6】 ボタンAを押すと「ゲー」を表示、ボタンBを押すと「パー」を表示するプログラムを作成しよう。なお、「最初だけ」ブロックで、「ゲー」を表示する。また、「ゲー」「パー」は、アイコン(小さいダイヤモンド、しかく)を利用する。 (ファイル名:rei1-6)

<手順>

- 1) 「基本」から「最初」ブロックを選択する。
- 2) 「基本」から「アイコンを表示」を選択し、「小さいダイヤモンド」にする。

<手順(ボタンA、ボタンB)>

- 1) 「入力」ブロックから、「ボタンを押されたとき」を選択する。
- 2) ボタンAでは、「アイコンを表示」で「小さいダイヤモンド」(ゲー)を表示する
- 3) ボタンBでは、「アイコンを表示」で「しかく」(パー)を表示する。

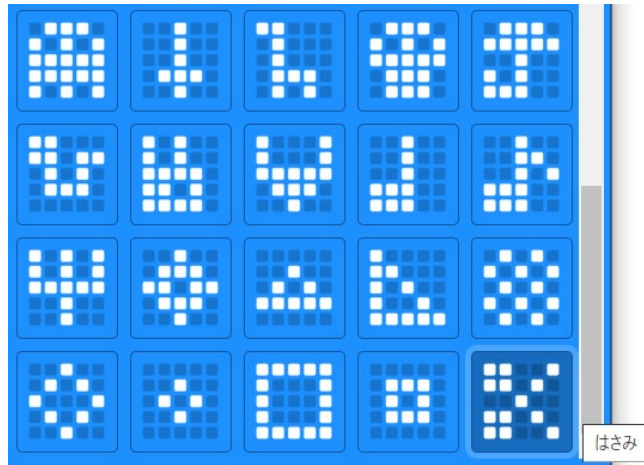


アイコンとスイッチボタンの利用(演習課題)

【例題1-7】 ボタン「A+B」が押されたとき、はさみ(チョキ)を表示するプログラムを追加してみよう。また、2台の micro:bit にプログラムをダウンロードして、2人でじゃんけんを行ってみよう。
(ファイル名:rei1-7)

<手順(ボタンA+B)>

ボタンA+Bでは、「アイコンを表示」で「はさみ」を表示する。

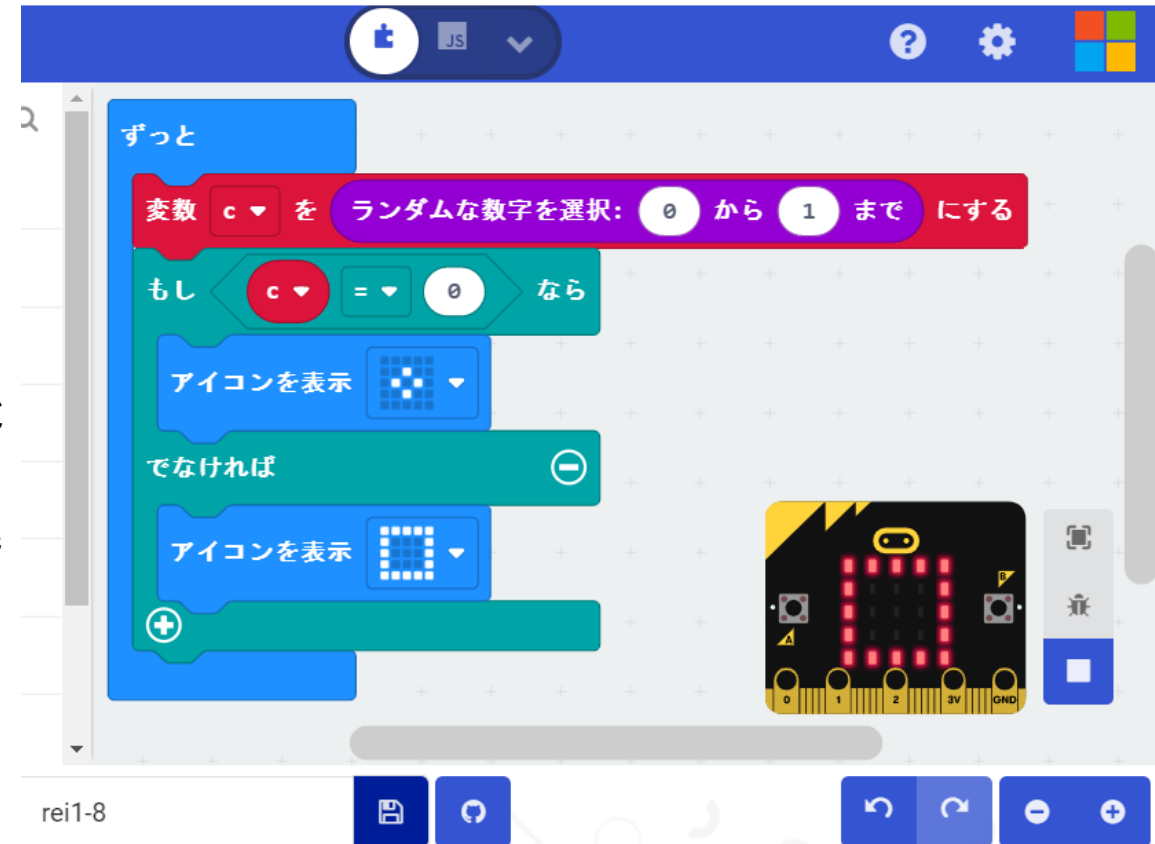


プログラムの基礎(分岐)

【例題1-8】 乱数(0、1)を発生させ変数「c」に代入し、cが0の時は(グー)、cが1の時は(パー)を「ずっと」くりかえし表示するようなプログラムを作成しよう。(ファイル名:rei1-8)

<手順>

- 1) 「基本」から「ずっと」ブロックを選択する。
- 2) 「論理」から「もし～なら～でなければ」ブロックを選択する。
- 3) 「基本」から「アイコン表示」ブロックを選択し、「小さいダイヤモンド」は「～なら」、「しかく」は「～でなければ」の後に接続しておく。
- 4) 「変数」から「変数を追加する」を選択し、変数の名前をcにする。また、「変数cを0にする」ブロックを選択する。
- 5) 「論理」「くらべる」から「 $0 = 0$ 」ブロックを選択し、「 $c = 0$ 」に変更し、「もし・・・」ブロックに重ねる。
- 6) 「計算」から「ランダムな数字を選択:0から10まで」を選択し、範囲を「0～1」にし、「変数c・・・」ブロックに重ねる。



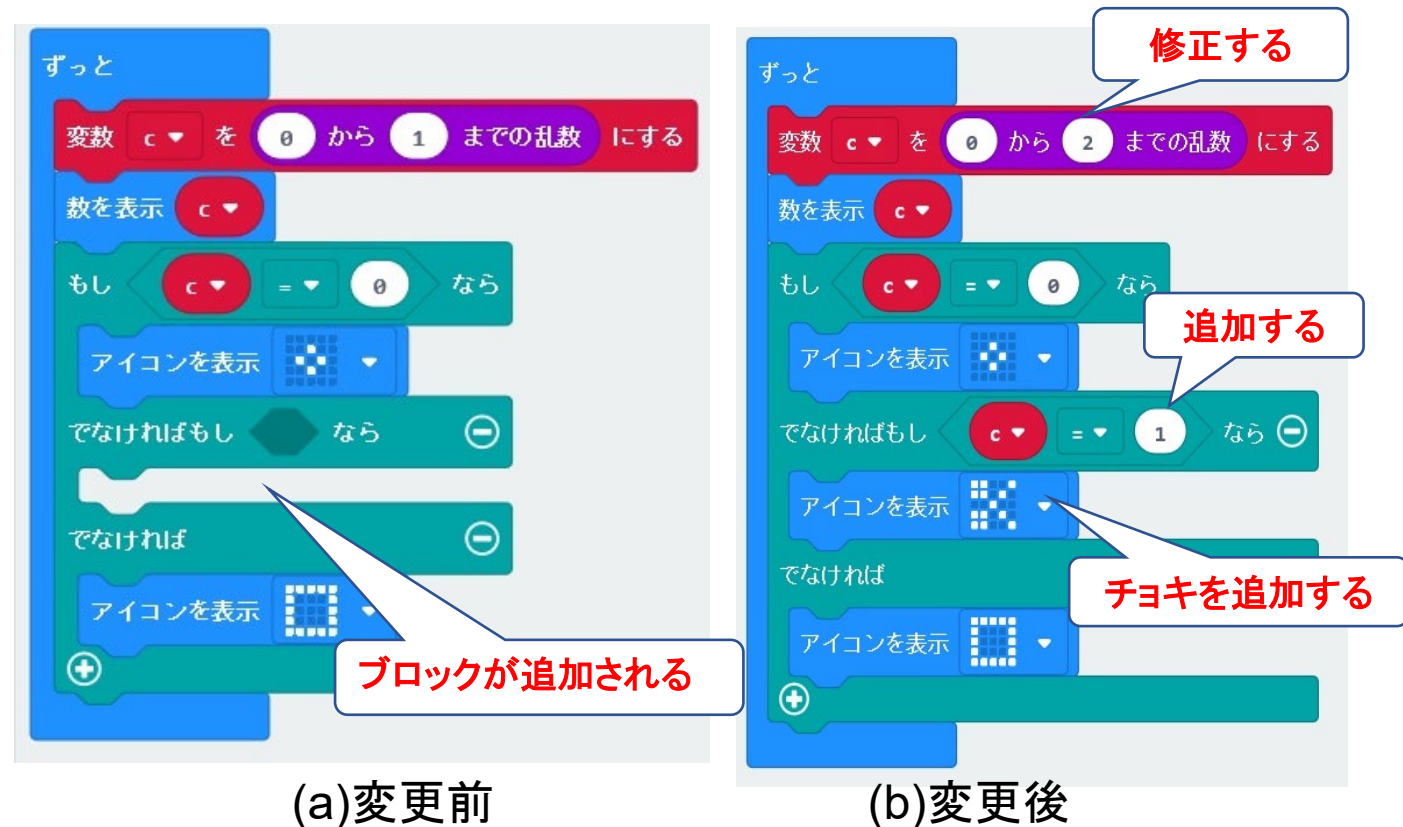
プログラムの基礎(分岐)(演習課題)

【例題1-9】 乱数(0、1、2)を発生させて、変数cに代入して、cが1の時は、「アイコン表示」の「はさみ」(チョキ)を表示するようなプログラムに変更しよう。(ファイル名:rei1-9)

<手順>

- 1) 「もし～なら～でなければ」のブロックの「+」マークをクリックすると、「でなければもし～なら」が追加される(図(a))。次に、ゲーを移動、チョキを追加する(図(b))。
- 2) 「0から1までの乱数」を「0から2までの乱数」にしておく。
- 3) 「でなければもし」の箇所、「 $c = 1$ 」にブロックを追加しておく。

例題1-8、例題1-9のようなプログラムの構造を分岐構造という。



じゃんけんゲーム(応用課題)

- 2台の micro:bit にプログラムをダウンロードして、2人でじゃんけんを行ってみよう。また、勝敗を判定する表から、判定式を考えてみよう。

種類	数値	A	B	判定	(A-B)の値
グー	0	0	0	引分け	0
		0	1	A	-1
		0	2	B	-2
チョキ	1	1	0	B	1
		1	1	引分け	0
		1	2	A	-1
パー	2	2	0	A	2
		2	1	B	1
		2	2	引分け	0



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

光センサによる学習指導～光センサ

【例題1－10】 光センサ(LED)を使って明るさの値をLED に表示するプログラムを作り、LED の部分を覆って値が変化することを確認してみよう。 (rei1-10)

<手順>

- 1) 「基本」から「ずっと」ブロックを選択する
- 2) 「基本」から「数を表示」ブロックを選択する。
- 3) 「入力」から「明るさ」ブロックを選択する。
- 4) 「基本」から「表示を消す」ブロックを選択する。
- 5) 「基本」から「一時停止(ミリ秒)」ブロックを選択し、「1000」(1 秒)を選択する。4) で一度表示を消して、5) で1 秒待つことで、表示を見やすくしている。

光センサは、暗いときは0 で、最大の明るさのときは255の値になる。



光センサによる制御

【例題1-11】 LEDセンサを利用して、「♡」マークを暗い時に点灯させてみよう。明るさを変化させて考えてみよう。(rei1-11)

手順>

- 1) 「基本」から「ずっと」ブロックを選択する
- 2) 「論理」から(条件判断)「もし～なら～でなければ」ブロックを選択する。
- 3) 「論理」から(くらべる)「 $0 < 0$ 」ブロックを選択し、右の数値「0」を「150」にする。
- 4) 「入力」から「明るさ」ブロックを選択し、「明るさ」を「 $0 < 150$ 」の左の数値「0」に重ねる。
- 5) 「明るさ < 150 」を「もし～なら～でなければ」ブロックの「真」の箇所を重ねる。
- 6) 作成した「もし～なら～でなければ」ブロックに、「♡」マークの点灯のプログラムを入れる。

シミュレータ画面の光センサの表示箇所をドラッグすると、数値が変わる。



光センサによるLEDの点灯(応用課題)

理科第6学年:

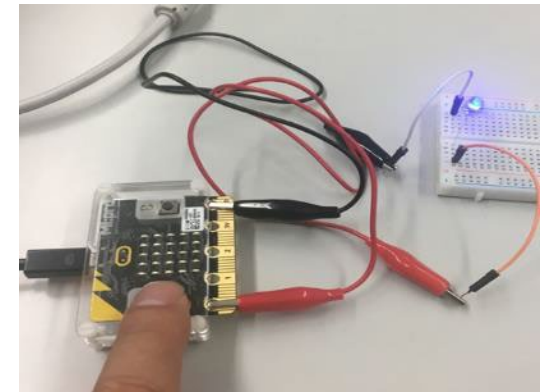
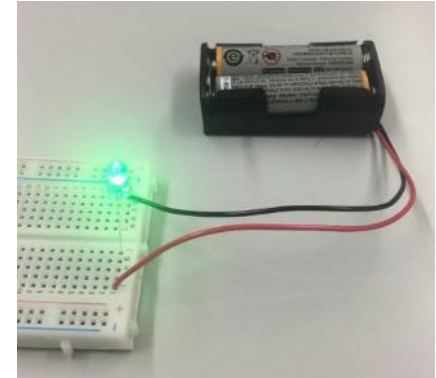
身の回りには、電気の性質や働きを利用した道具があること等をプログラミングを通して学習する場面

- プログラムを変更して、LED回路に接続する。
- 暗くなれば、LEDが点灯する。
(ファイル名:kadai1)
- スイッチを押すと、LEDが点灯する。
(ファイル名:kadai2)

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



プログラムの変更



スイッチによるLEDの点灯

(参考)「プログラミング的思考を育成する」ための教材と学習指導

■ プログラミング思考を育成するためには、発達段階に応じた教材が必要である。

学習者の主体的な学びを促すには、

- 1) 教材を観て、実行して、自分で確認する。
- 2) 教材プログラムを観て、自らが納得する。
- 3) 教材プログラムの変更も試みて、思考する。

の3つのステップが重要である。

■ 「自分で確認して、自らが納得する」のステップにおいて、プログラミングの過程と結果の可視化は必要である。

■ micro:bit を利用することにより、1)～3)のステップの学習は、実現しやすい。

「光センサによる制御」の学習指導

■ステップ1(例題1-11)

- シミュレータ画面から、マイクロビットの光センサの明るさ(数値)は128である。
- また、プログラムからは、光センサの明るさ(数値)は150である。
- 128<150 なので、プログラムでは、シミュレータ画面では点灯する。

■ステップ2(例題1-11)

- シミュレータ画面の黄色の箇所をドラッグして、数値(128)を変更する。
- 数値を大きくすると消える。小さくすると点灯する。
- 次に、プログラム内の数値の値を変えて、プログラムを変更してみる。
- 数値を小さくすると消える。大きくすると点灯する。

■ステップ1)、2)を通じて、値を変更すると、消えたり、点灯したりすることを**確認**し、体験(納得)する。

■このステップ1)、2)は、まだ、ステップ3)の**思考**には至っていない。

「光センサによるLEDの点灯」の学習指導

■ステップ3(例題1-11)

➤ 例題1-11のプログラムをマイクロビットにダウンロードする。

- プログラムのダウンロードは、プログラミングの指導である。

➤ 発問:「どうすれば、シミュレータと同じように、点滅させることができるでしょうか？」

- 教室や部屋の環境が異なる(条件が異なる)ので、子どもたちのマイクロビットは、同じように点滅しないことも起こる。

➤ これは、子どもに「気づき」を促す、「発問」です。

- LEDのセンサを手で隠したりして、あるいは、明るいところに持っていったりして、確かめてみる子どもがでてくる。

■ステップ3(応用課題)

➤ LED回路が用意できれば、実際のLEDの点灯、スイッチによる点灯を行う。

- 教師の机上実験でもよい。

■ステップ3の「思考」で、「学習者の主体的な学び」を促す、学習のねらいは達成です。

プログラムのダウンロードについて

<プログラムに対する注意事項>

利用している例題プログラムなどは、NPO法人学習開発研究所の下記のWebサイトからダウンロードしてください。

本書の中で記載しているファイル名、例えば、rei〇〇は、保存ファイル名では、microbit-rei〇〇.hex、 になっています。

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

