マイクロビット活用
（プログラミング教育）講座

京田辺市教員研修資料

日時：2019 年 8 月 8 日(木)　13:30～16:00
場所：京田辺市中央公民館
担当：高橋　参吉（NPO 法人　学習開発研究所）

目　次

１．micro:bit の基本操作 1
２．プログラムの基礎（順次，繰返し） 4
３．プログラムの基礎（分岐） 6
４．演習　～スイッチボタンの利用～ 9
５．課題　～miicro:bit による LED の制御～ 10

＜プログラムに対する注意事項＞

本テキストで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 JavaScript のファイル名，例えば，rei○○は，保存ファイル名では，microbit-rei○○.hex，　になっています。

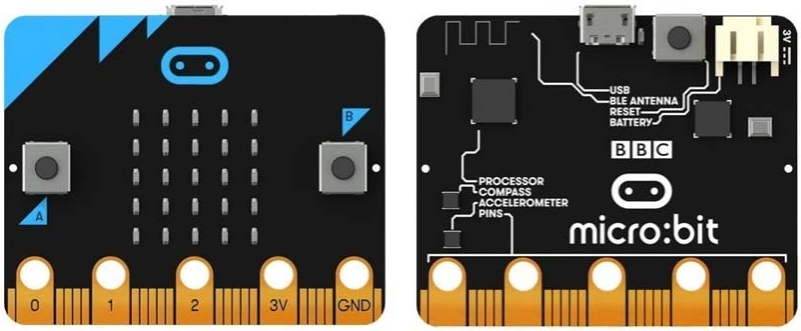
http://www.u-manabi.org/microbit/

＜参考文献＞

1)　micro:bit の公式 Web サイト（日本語）：https://microbit.org/ja/
micro:bit の冒険を始めよう　https://microbit.org/ja/guide/
2)　ガレス・ハルファクリー著，金井哲夫訳：BBC マイクロビット公式ガイドブック，日経 BP 社(2018.10).
3)　高橋参吉，喜家村奨，稲川孝司：micro:bit で学ぶプログラミングーブロック型，JavaScript そして Python へー，コロナ社(2019.9).

１．micro:bit の基本操作

micro:bit は，下記の図のような手のひらサイズのコンピュータです[1),2)]。



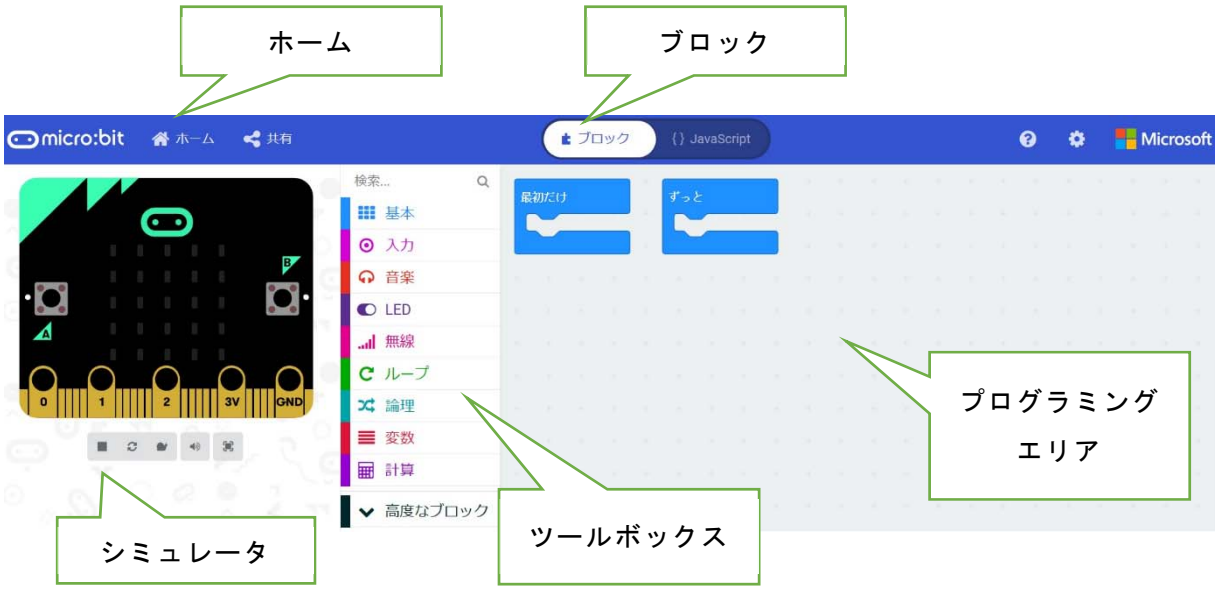
下記の micro:bit の Web サイトへアクセスします。

https://makecode.microbit.org/

その中に，「マイプロジェクト」，「チュートリアル」，「ゲーム」などがあり，「マイプロジェクト」では，新しいプロジェクトを作成したり，保存したプロジェクト（プログラム）を読み込むことができます。

新しいプロジェクトを選択すると，次の図のような micro:bit のシミュレータ画面が表示されます。

＊日本語で表示されていない場合は，Web ページの一番下で日本語を選択しておく。



画面の左側から，micro:bit での実行が確認できるシミュレータ，ツールボックス，そして一番右側が，プログラミングエリアです。

［ツールボックス］

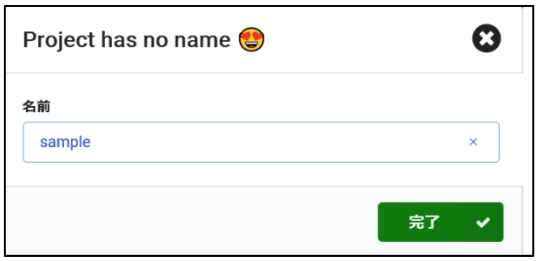
ツールボックスには，基本，入力，音楽，LED，無線，ループ，論理，変数，計算，そして，高度なブロックがあり，それぞれのツールボックスのツールをクリックすると，利用できるブロックが表示されます。

［プログラミングエリア］

プログラミングエリアは，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ブロックエディタになっています。最初に，「最初だけ」，「ずーと」のブロックが置かれています。

［ホーム］

画面の上部左側には「ホーム」があり，「ホーム」を選択すると，新しいプロジェクトに名前を付けて保存できます。



［ブロック］［JavaScript］

画面の上部の真ん中の「ブロック」を「JavaScript」に切り替えることによって，「ブロック」で描かれたプログラムを「JavaScript」で表示することができます。

［ダウンロード］

画面の下に，［ダウンロード］「題名未設定…」「アイコン（FD)」というボックスがあり，プログラム名を入れてパソコンにプログラムを保存したり，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ます。名前を入れて，右にある「アイコン（FD)」をクリックすると，パソコンのフォルダにファイルを保存することができます。

また，パソコンの USB に micro:bit をつないで，次の表示に従って micro:bit にプログラムを転送することができます。

転送が完了すれば，micro:bit でプログラムを動かすことができます。

もし，micro:bit にプログラムが格納されていれば，上書きされるので注意しょう。

［シミュレータ］

micro:bit のシミュレータは，micro:bit の画面の下のボタンが，四角ボタン（■）であれば，クリックすると停止，三角ボタン（▶）であれば，クリックすると開始できます。また，スローモーション（左から三つ目）で，ゆっくり実行することもできます。

それでは，次の例題で，micro:bit の基本操作を確かめてみましょう。

【例題 1－1】　図のようなハートマークを LED 画面に表示させよう。次に，ハートマークを点滅させてみよう。作成したプログラムはパソコンに保存する。

＜手順 1＞（ファイル名：rei1-1-1）

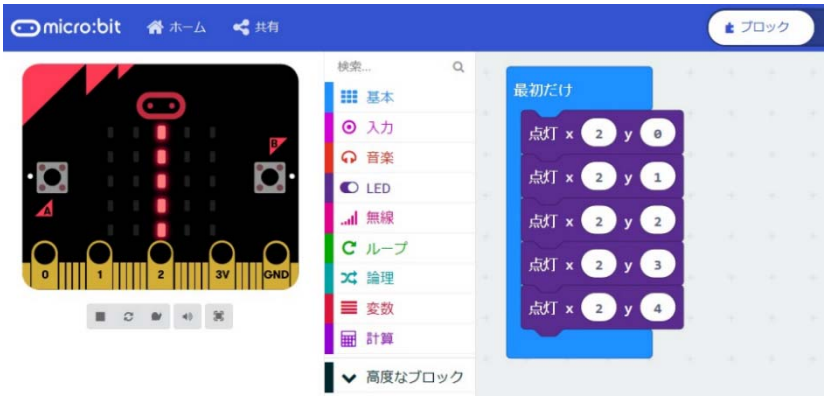
1)　「ホーム」をクリックし，「新しいプロジェクト」を選択する。
2)　ツールボックスの中の「基本」をクリックし，「LED 画面に表示」ブロックをドラッグ＆ドロップでプログラミングエリアに移動する。
3)　「最初だけ」ブロックに「LED 画面に表示」ブロックをつなげる。
4)　LED をクリックすると光の ON/OFF が切り替わるので，ハート形に見えるように LED を ON にし，動作を確認する。
　　なお，不要なブロックは，ツールボックスへドラッグ＆ドロップすると削除できる。

＜手順 2＞（ファイル名：rei1-1-2）

1)　「最初だけ」ブロックを「ずっと」ブロックに変更する。
2)　「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。
3)　「基本」から「表示を消す」ブロックをつなぐ。
4)　「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。
5)　ダウンロードの右のアイコンをクリックして，適切なフォルダにプログラム名（rei とすると，実際には microbit-rei.hex となる）をつけて，保存する。
6)　パソコンの USB に micro:bit をつなぎ，保存したプログラムを選んで，右クリック⇒送る⇒MICROBIT　でファイルを転送できる。
7)　転送している間，mocro:bit の裏側の LED がオレンジ色に点滅し，点滅が終われば，リセットボタンを押して，プログラムを起動させる。

２．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。なお，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。

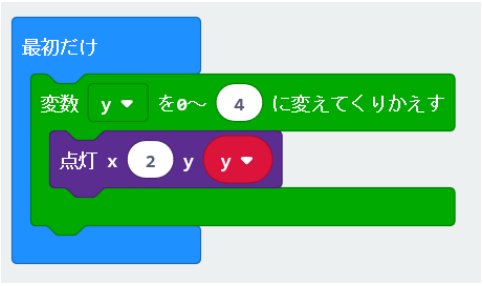
4)　点灯ブロックを 4 回コピーし，x をすべて「2」，y を「1～4」にする。

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。

【例題 1－3】　例題 1－2 のプログラムを，「ループ」から，繰り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（5 つ）を「最初だけ」ブロックから外す。

2)　「ループ」から「変数（カウンター）を 0～4 に変えてくりかえす」ブロックを選択する。

3)　「変数（カウンター)」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。

変数は，数値や文字などのデータを入れる容器に当たるものであり，数値や文字などのデータを定数という。このようなプログラムの基本構造を繰返し構造という。

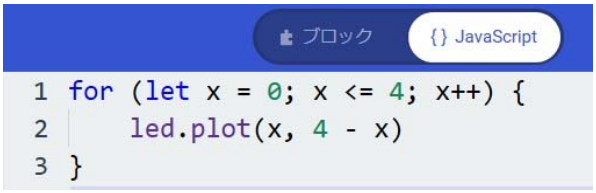
【例題 1－4】　例題 1－3 で，点灯の x 座標を変数「x」，y 座標を変数「4-x」に変更して，図の形を確認してみよう。次に，作成したプログラムが「JavaScript」では，どのように書かれているか確認してみよう。（ファイル名：rei1-4）

＜手順（概略）＞

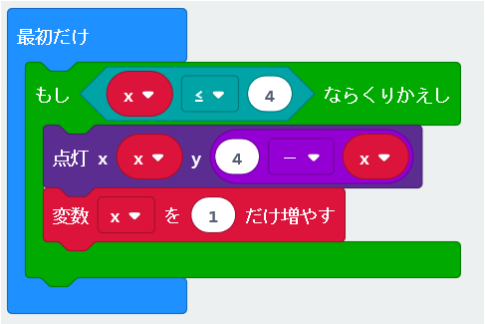
1)　「計算」から，「引き算」のブロックを選択する。

2)　計算式（4-x）を作成して，「点灯」の y 座標に置く。

JavaScript では，繰返しは for を使う。使用する変数は，「let = 0」のように定義される。また，x++は，x を 1 ずつ増やすことである。



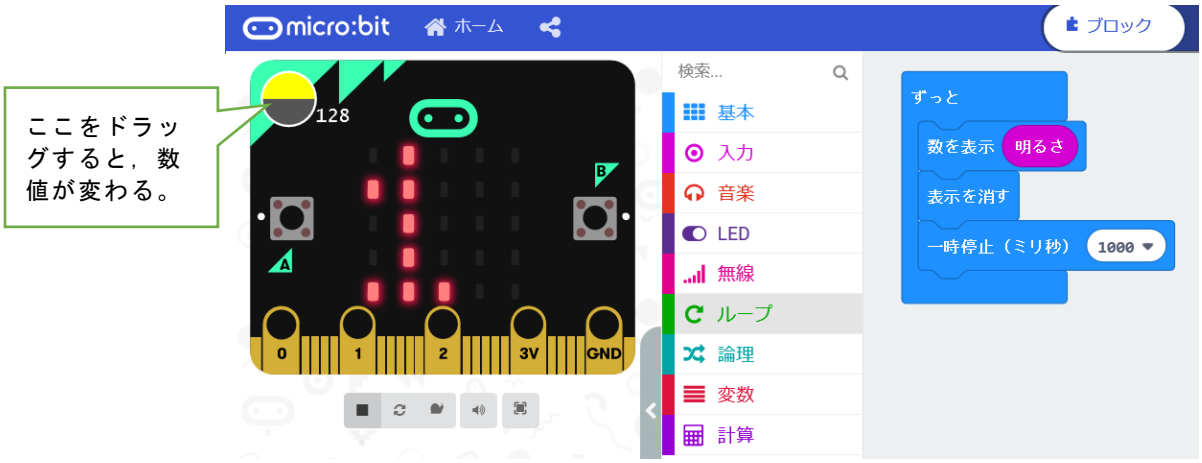
【例題 1－5】　例題 1－3 のプログラムの繰り返し「for～」が「while～」になるように，「ループ」の箇所でブロックを変更して，プログラムを作成しよう。また，JavaScript のプログラムを確認してみよう。（ファイル名：rei1-5）



「x += 1」は，「 x = x + 1 」と同じ内容で，最初の「let x = 0」，ループ内にある「x += 1」で，x を 1 ずつ増やし，繰り返していくことになる。このような繰り返しを「カウンター」という。なお，繰り返し回数がわからない時は，for は使えないので While を使うとよい。

３．プログラムの基礎（分岐）

【例題 1－6】　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かめてみよう。（rei1-6）



＜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 0 で，最大の明るさのとき 255 の値になる。

【例題 1－7】　LED センサを利用して，例題 1-1 のハートマークを暗い時に点滅させてみよう。明るさは，150 未満(rei1-7-1)，50 未満（rei1-7-2)の場合について考えてみよう。

＜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」の箇所に重ねる。
6) 作成した「もし～なら～でなければ」ブロックに，例題 1-1(rei1-1-2)のプログラムを入れる。

このようなプログラムの構造を分岐構造という。

＜注意＞

シミュレータの明るさは，「128」（シュミュレータの左上の数値）になっているので，このプログラム(150 未満)(rei7-1-1)では，LED のハートマークが点灯し，50 未満のプログラム（rei7-1-2）では点灯しない。したがって，50 未満のプログラムで，micro:bit の LED の箇所を手で覆って暗くして，点灯するかどうか確かめてみよう。

【例題 1－8】　乱数（0，1）を発生させて，変数「c」に代入して，c が 0 の時は「小さいダイアモンド」（グー），c が 1 の時は「しかく」（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。なお，グーとなる図形は，「基本」の「アイコン表示」から選択する。（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，「小さいダイアモンド」「しかく」

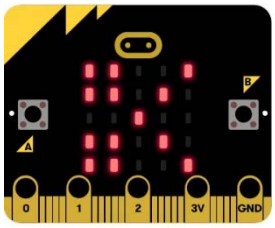
を選択する。「小さいダイアモンド」は「～なら」，「しかく」は「～でなければ」の後に接続しておく。

4) 「変数」から「変数」ブロックを選択し，変数の名前を c にする。また，「変数を 0 にする」ブロックを選択して，変数の箇所を c に変更する。

5) 「論理」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「0 から 4 までの乱数」を選択し，範囲を「0～1」にし，「変数・・・」ブロックに重ねる。

【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 2 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：rei1-10）



＜手順＞

1) 「もし～なら～でなければ」ブロックを「もし～なら～でなければ」とするには，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，グーを移動，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」の箇所に，「 c ＝ 1 」にブロックを追加しておく。

(a)変更前

(b)変更後

４．演習　～スイッチボタンの利用～

【演習 1】　ボタン A を押すと「グー」を表示，ボタン B を押すと「パー」を表示するプログラムに作成しよう。なお，「最初だけ」ブロックで，変数 c を「0」（グー）にして，表示するようにする。（ファイル名：ens1）

＜手順＞

1)　「基本」から「最初」ブロックを選択する。

2)　「変数」から「変数」ブロックを選択し，「c」を作成する。

3)　「変数」から「変数を0にする」ブロックを選択し，変数cを「0」にする。

4)　「基本」から「数を表示」ブロックを選択し，変数cを表示する。

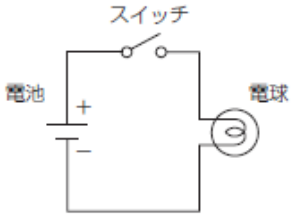
＜手順（ボタンA，ボタンB）＞

1) ボタンAでは，「アイコンを表示」で「小さいダイアモンド」（グー）を表示する。

2) ボタンBでは，「アイコンを表示」で「しかく」（パー）を表示する。

【演習2】　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。

５．課題　～miicro:bit による LED の制御～



（a） 電球の回路

3Vの電池を使って電球をON/OFF するには図（a）のように接続する[3)]。ここでは，豆電球の代わりにLED（発光ダイオード）を接続して，光センサを使ったLEDの制御を行う。なお，使用するLEDは，3Vで点灯し，豆電球とは違い「＋」「－」がある。

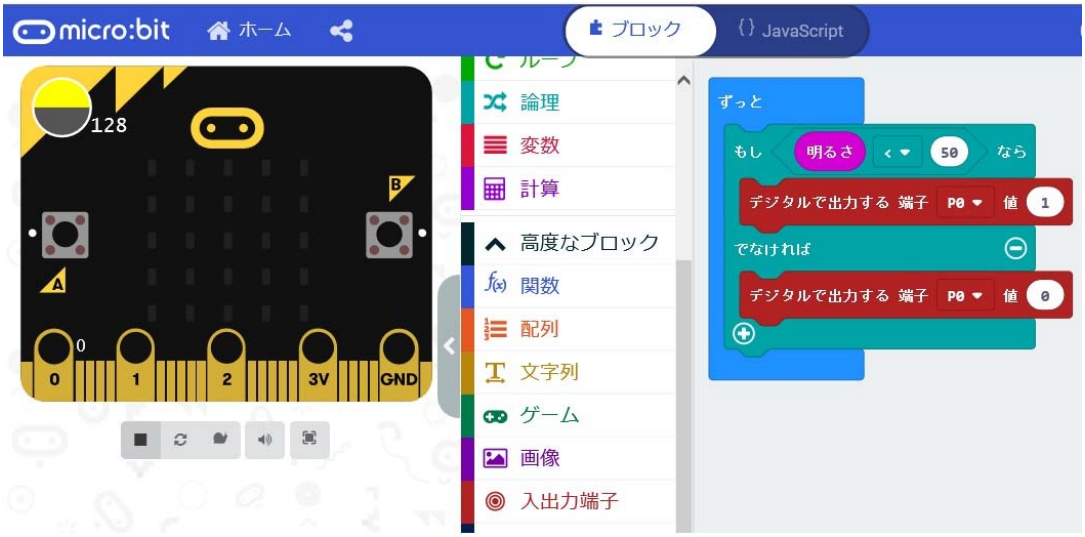
micro:bit の下端の三つの端子（P0，P1，P2）はプログラムで制御できる。「高度なブロック」の「入出力端子」の「デジタルで出力する」を選んで
デジタル出力（ON）

とすると，端子P0 に＋3 V の電圧が出る。そのP0 端子をLED の＋側に，GND をLED の－側に接続するとLED に電流が流れ，出力側がON になる。

デジタル出力（OFF）

とすると，端子P0 が0 V になり，接続しているLED に電流が流れなくなり，出力側がOFFになる。P1，P2 も同様に利用できる。

【課題1】　micro:bit の光センサとLEDを使って，明るいときは消灯し，暗くなったときに点灯する常夜灯を作ってみよう。＜kadai1＞



＜手順＞

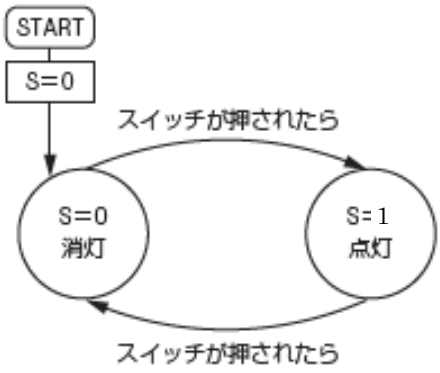
1)　「論理」から「もし～なら～でなければ」ブロックを選択する。
2)　「論理」から「くらべる」ブロックで不等式を選択する。
3)　「入力」から「明るさ」ブロックを選択する。
4)　現在の場所での明るさの数値を入力する（図の数値は「50」）。
5)　「高度なブロック」の「入出力端子」ブロックから「デジタルで出力する」を選択する。
6)　暗くなったら，端子P0 を「1」に，そうでなければ，端子P0 を「0」に設定する。

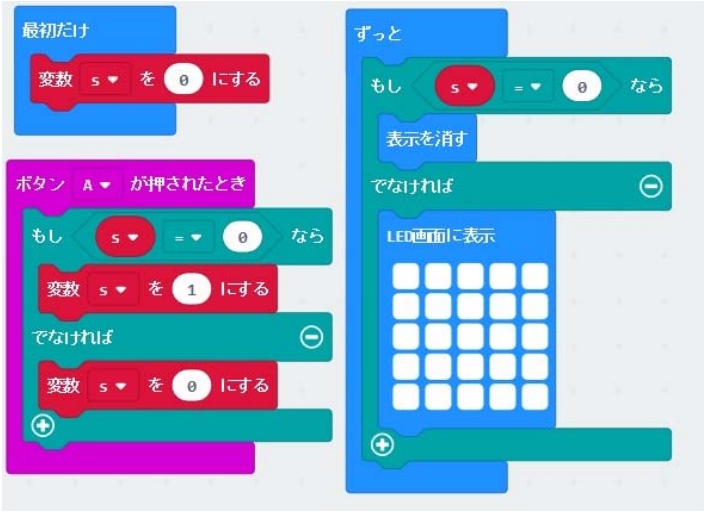
実際に，電池，LEDをブレッドボードに接続して，確かめてみよう。
現在の場所の明るさの数値は，環境によって大きく変わる。実験場所での明るさを調べ（例題1－6），明るさの値がわかったら，その値を使ってプログラムを作成する。

【課題2】　Aスイッチを一度押すとLEDがすべて点灯し，再度押すとLEDがすべて消灯するプログラムを作成してみよう。＜kadai2＞

＜プログラムの考え方＞

下記の状態遷移図[3]で，初期設定をs＝0とする。s＝0なら消灯，s＝1なら点灯とする。s＝0のときにボタンAを押すとs＝1になる。また，s＝1のときにボタンAを押すとs＝0にする。そして，sの状態を調べ，s が0 なら「表示を消し」，sが1なら「すべてのLED を表示」する。





【課題3】

「表示を消す」「LED画面に表示」の箇所を変更して，実際に，電池，LEDをブレッドボードに接続して，確かめてみよう。＜kadai3＞