

## 6年生 理科 電気を効率よく使う方法を考えよう

micro:bit 本体の内蔵 LED や市販の外部 LED を利用して電気の学習を行います。

### 【micro:bit】とは

「micro:bit」はイギリスの BBC (英国放送協会) が中心となって開発した教育向けのマイコンボードです。5 × 5 の赤色 LED やさまざまなセンサー (コンパス、加速度センサ等) が搭載されていて、任意の Web ブラウザを使用して、簡単にブロックプログラミングを行うことができます。また、外部の機器を接続することが可能です。

### 【準備】

授業で活用するためには、最低限下記のものが必要です。

- ・ micro:bit 本体
- ・ 電池ボックス (本体のみで動作させるために必要)
- ・ USB ケーブル microUSB TypeB (パソコンとの接続用)

## 1 micro:bit の接続

### (1) micro:bit 本体とパソコンを接続する

ロゴの上あたりに USB ケーブル Type-B を差し込み、パソコン本体と接続します (図 1 - 1)。

※ケーブルには向きがあるので、注意しましょう。

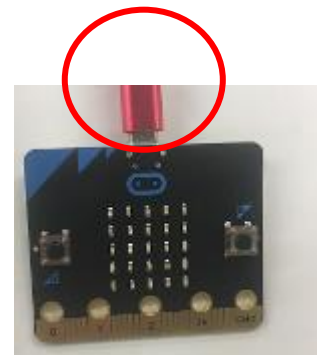


図 1 - 1

### (2) 「MICROBIT」ドライブ

micro:bit が接続されると、画面に「MICROBIT」ドライブが開きます (図 1 - 2)。

(注意 : USB メモリのように見えますが、USB メモリではありません)

パソコン上で作成したプログラムをこのドライブにコピーすることで、micro:bit 本体にデータを転送することができます (転送方法については、3 の (2) を参照のこと)。

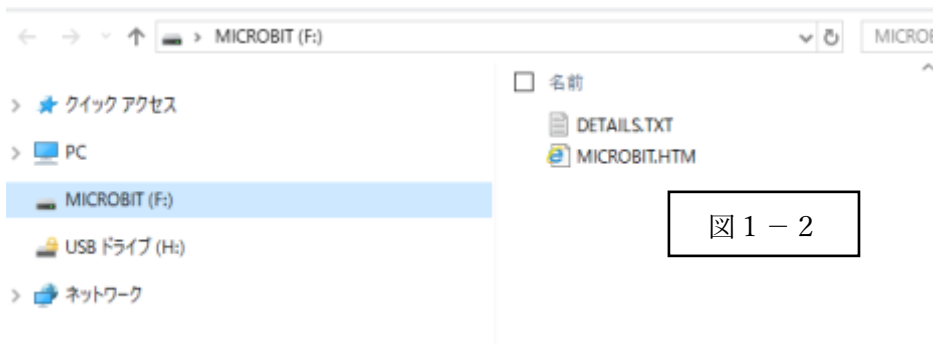


図 1 - 2

## 2 MakeCode for micro:bit について

### (1) MakeCode for micro:bit を利用する

MakeCode は Microsoft が提供する「自由に使えるプログラミング学習環境」です。  
micro:bit のプログラムもこれを利用します。

#### ① MakeCode にアクセスしましょう。

下記の URL アドレスで接続すると、MakeCode (図 2 - 1) の画面が表示されます。

URL : <https://www.microsoft.com/ja-jp/makecode>

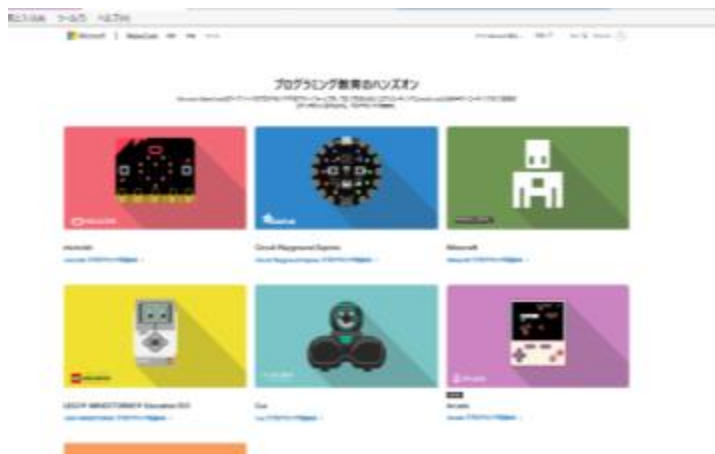


図 2 - 1

#### ② MakeCode for micro:bit にアクセスしましょう。

①の左上の「micro:bit でプログラミングを始める」をクリックします (図 2 - 2)。



図 2 - 2

#### ③ 「新しいプロジェクト」にアクセスしましょう。

新規のプログラムを作成するときは、「MakeCode for micro:bit」の画面で、「新しいプロジェクト」をクリックします (図 2 - 3)。



図 2 - 3

micro:bit 用の MakeCode エディタが起動しますので、その画面でプログラムの作成を行います。

## (2) MakeCode エディタについて

micro:bit 用のプログラムを作成するエディタについて



- ① プログラム作成画面 (②から必要なブロックをドラッグ&ドロップして作成します)。
- ② プログラミング用のブロックはここから選択します。用途により色分けされています。
- ③ シミュレータ画面 ①で作成したプログラムの動きを確認することができます。
- ④ 表示の拡大または縮小を行います。
- ⑤ 命令を、一つもとに戻したり、それを取り消したりできます。
- ⑥ 作成したプログラムをパソコン本体に保存します。
- ⑦ ブラウザにより作成したプログラムを直接 micro:bit にダウンロードすることができます。

### 3 プログラムを作成してmicro:bit本体に転送

#### (1) プログラムを作成する

- ① 画面に文字を表示しましょう。

中央のブロックから「基本」を選び「文字列を表示」を選びます（図3-1）。



図3-1

- ② 右にある「ずっと」の中にドラッグ&ドロップをしましょう（図3-2）。



図3-2

- ③ 左のシミュレーター上に文字が表示されていることを確認しましょう（図3-3）。

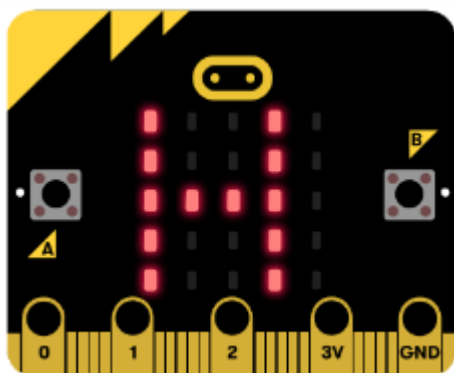


図3-3

(2) micro:bit 本体への転送

- ① プログラムを保存しましょう (図3-4)。

画面下の「題名未設定」となっている部分に保存名を入力します。

ここでは、「moji1」としておきます。

入力したら、「保存」アイコンをクリックします。



図3-4

- ② 名前を付けて保存しましょう。

「保存」アイコンをクリックすると、下記のメッセージが表示されます。

メッセージ部分の「保存」の右の矢印を選び、「名前を付けて保存」をクリックします。

(図3-5、図3-6)

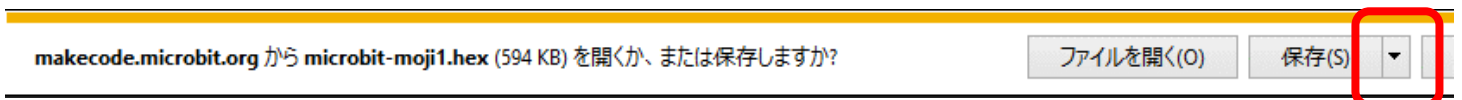


図3-5

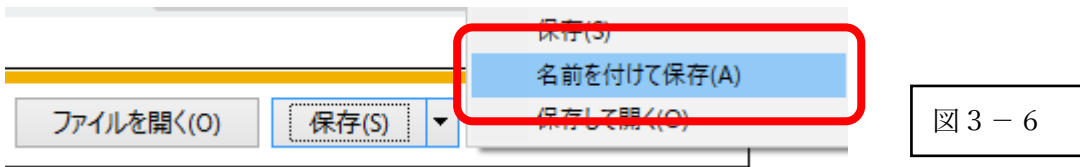


図3-6

※「保存」をクリックすると、保存フォルダを自分で指定できないので、後から探すのが難しい場合があります。

- ③ 保存場所を選択しましょう。

保存場所を指定するためのダイアログボックスが開きますので、場所を選び、保存をクリックします。

※ここでは、デスクトップ上の「マイクロビット」フォルダに保存をしています (図3-7)。

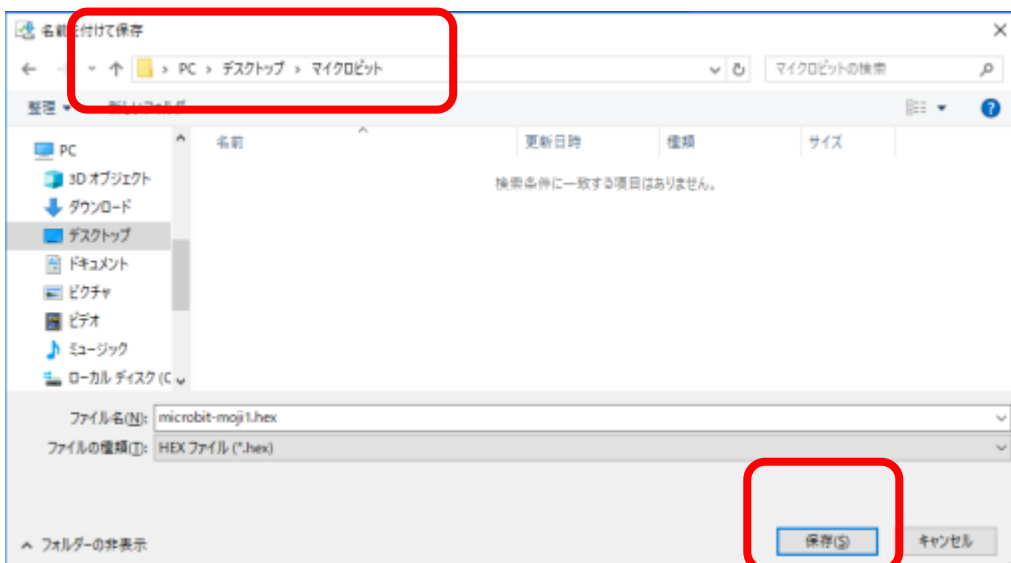


図3-7

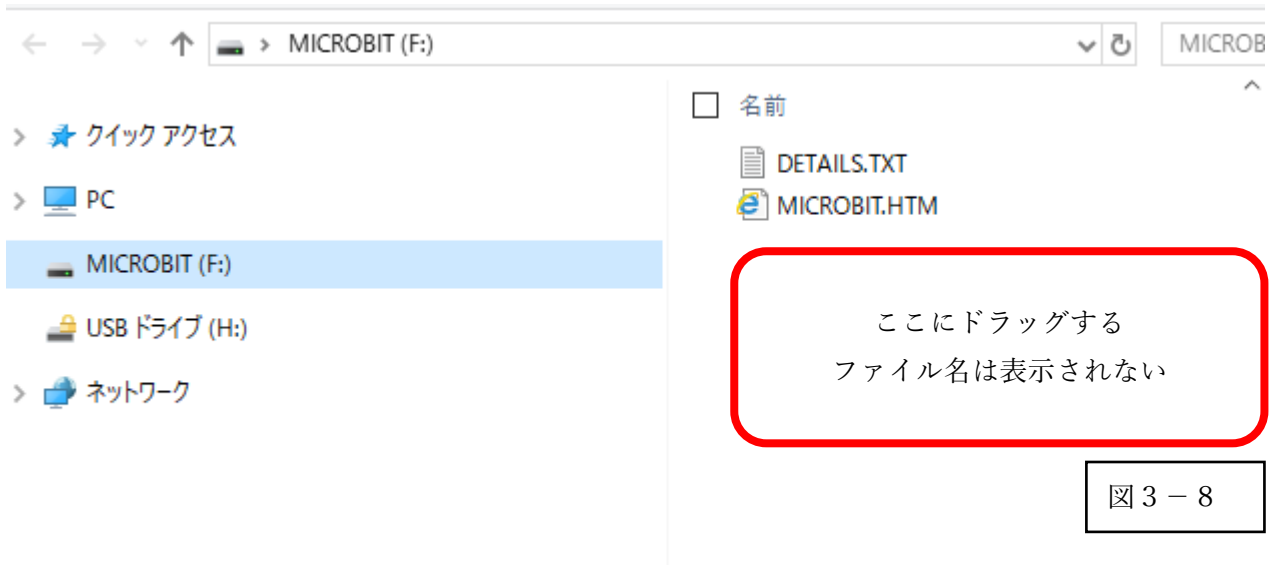
保存用のフォルダはあらかじめデスクトップなどに作成しておきましょう。

④ micro:bit 本体に転送しましょう。

保存されているフォルダから、「micorbit-moji1.hex」ファイルを「MICROBIT」ドライブに転送します。

「MICROBIT」ドライブへドラッグしてコピーしましょう（図3-8）。

MICROBIT ドライブには、転送したファイル名は表示されません。



※micro:bit の裏面のオレンジの点滅が始まると転送中となります。

転送中に USB ケーブルを抜かないようにしましょう。

転送が終了すると、本体の裏のオレンジの LED が点灯します。

※使用するブラウザにより、パソコン本体に保存せずに、直接 micro:bit に書き込むことも可能です。しかし、パソコン本体にプログラムを保存しておくことにより、何度も作成したプログラムを利用することができます。

micro:bit 本体では前のプログラムは、上書きされて消えてしまいます。

⑤ micro:bit で確認しましょう。

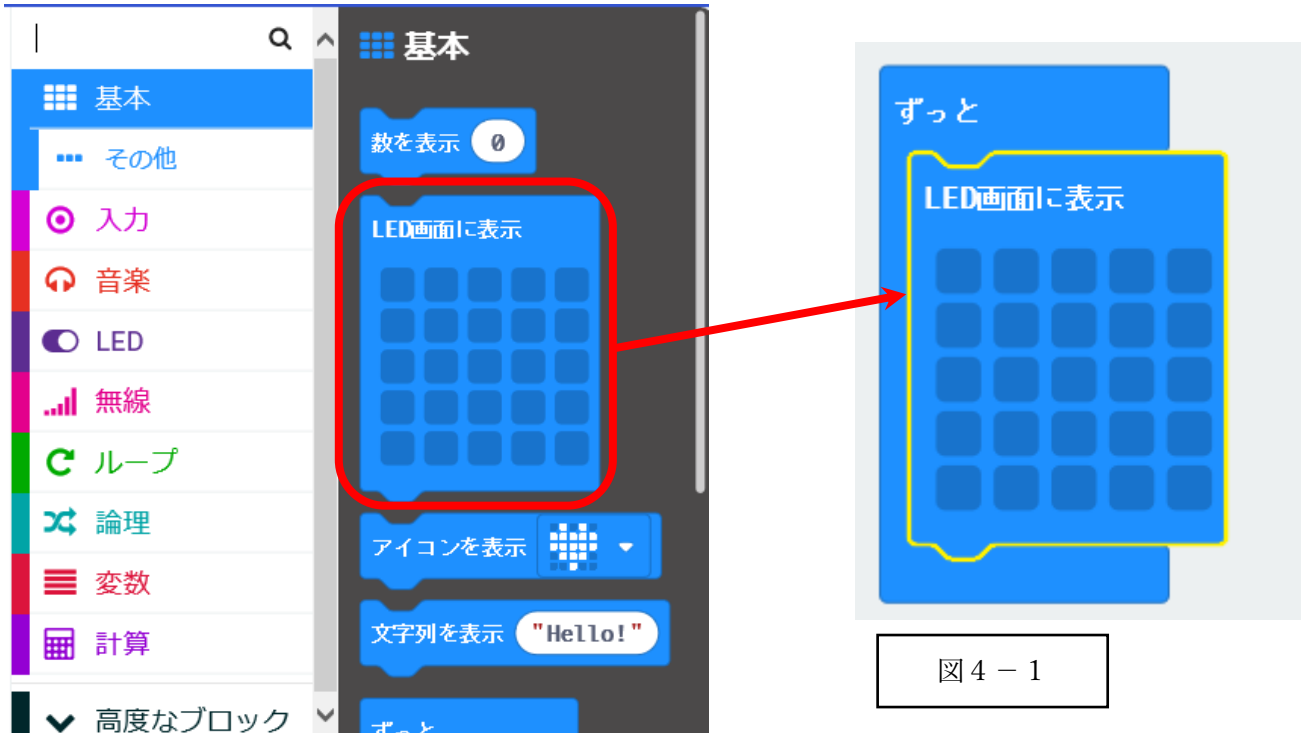
micro:bit 本体上で「Hello!」が表示されていることを確認します。

電池ボックスを micro:bit 本体に接続することで、パソコンから取り外した状態で動作させることができます。

#### 4 micro:bit 本体の内蔵 LED を利用

(1) 内蔵LED に○を点滅させます。

① 「ずっと」の中に基本から「LED 画面に表示」を選びドラッグしましょう (図4-1)。

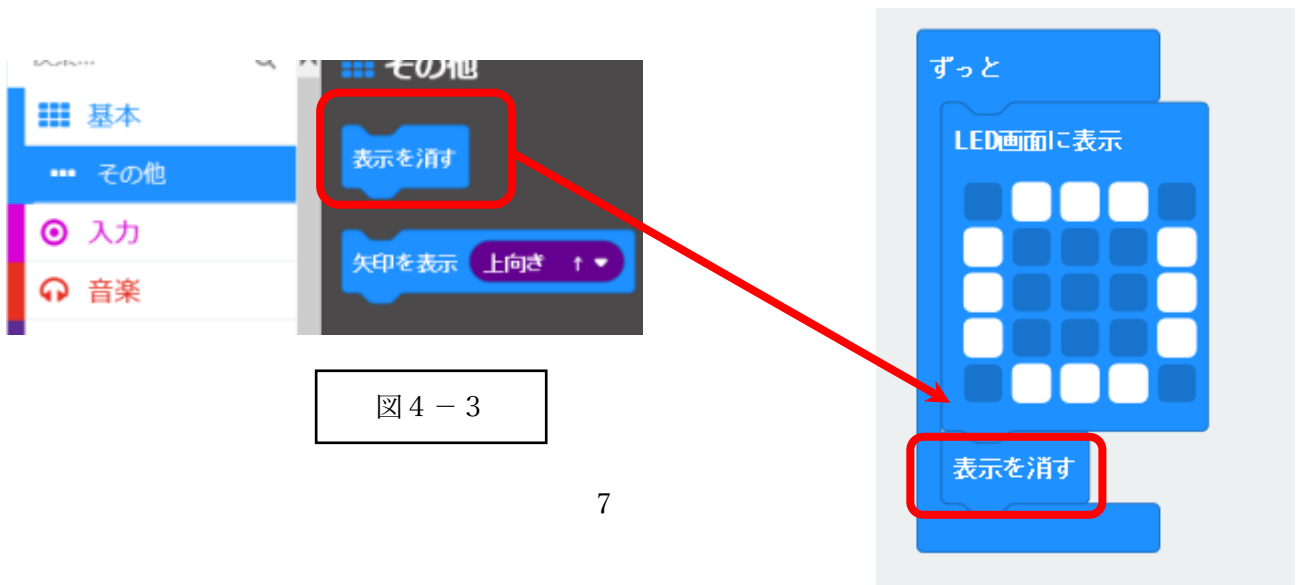


② 図4-2のようにLED画面に○を描きましょう。  
点灯させたいLEDに対応するマスでマウスをクリックします。

③ シミュレータで、画面に○が表示されていることを確認しまし  
ょう。



④ 「表示を消す」を追加しましょう (図4-3)。  
点滅させるために「表示を消す」を追加します。  
「表示を消す」は「基本」の「その他」にあります。



⑤ 「一時停止」を追加しましょう。

④の段階でプログラム上は点滅をしていますが、コンピュータの実行速度が速いため、人には点滅がわかりません。命令の間に一時停止を追加することで、人に確認できる速度に調節します。(図4-4)

⑥ シミュレータで動作を確認しましょう。

⑦ 問題がなければ保存後、micro:bit 本体に転送し、動作を確認しましょう。



図4-4

(2)暗くなると LED を自動的に光らせる

センサーを利用して、LED の表示をコントロールします。

① 最初に明るさを調べるプログラムを作成しましょう (図4-5)。  
「明るさ」は「入力」ブロックにあります。



図4-5

② シミュレータで動作を確認しましょう

左上の○の数字の部分「温度」を示しています (センサが得ている値)。

黄色い部分をドラッグして、値を変えて、正しく表示されていることを確認しましょう (図4-6)。

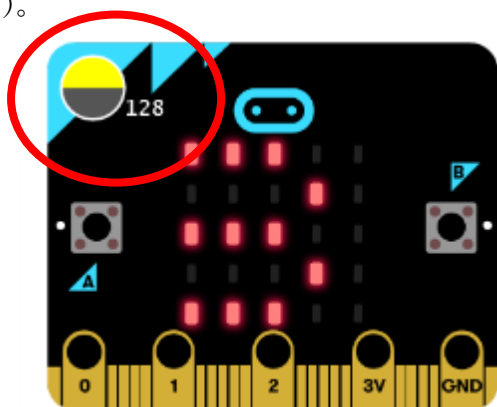


図4-6

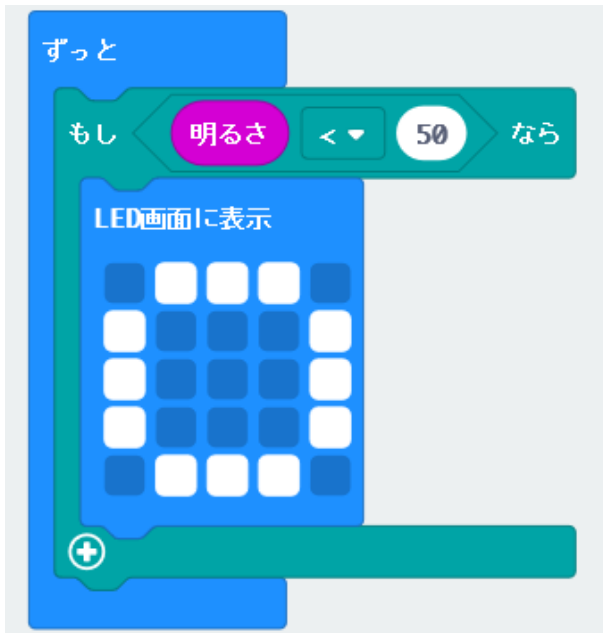
③ micro:bit 本体に転送し、周りの明るさの数値を確認しましょう。

教室の電気の下や、窓際などの明るさを確認しましょう。

また、光を遮るなどして、暗くしたときの明るさの数値を確認しましょう。



- ④ 暗くなると自動的に LED が光るように変更しましょう。  
 明るさが 50 より小さい場合、○を表示しましょう (図4-7)。



・「もし～なら」とくらべる内容は、「論理」に用意されています。  
 ※ ③で測った値にあわせて、境目の数値を考えましょう。

図4-7

- ⑤ シミュレータの左上の黄色い部分を動かして、表示を確認しましょう (図4-8)。  
 50 より小さい場合、○が表示されていることを確認します。

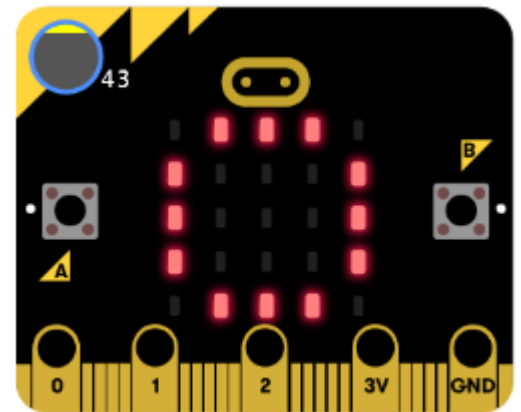


図4-8

- ⑥ 明るくなると消える機能を追加する  
 ④のプログラムでは、「表示を消す」がないため、いったん「○」が表示されたのちは、明るくなっても「○」が表示された状態になります。  
 電気の使い方として、「表示を消す」ことについても考えましょう。  
 「でなければ」を追加し、「表示を消す」の命令をいれます (図4-9)。

「もし」の下の+をクリックすると「でなければ」が追加されます。  
 「表示を消す」は基本のその他にあります。

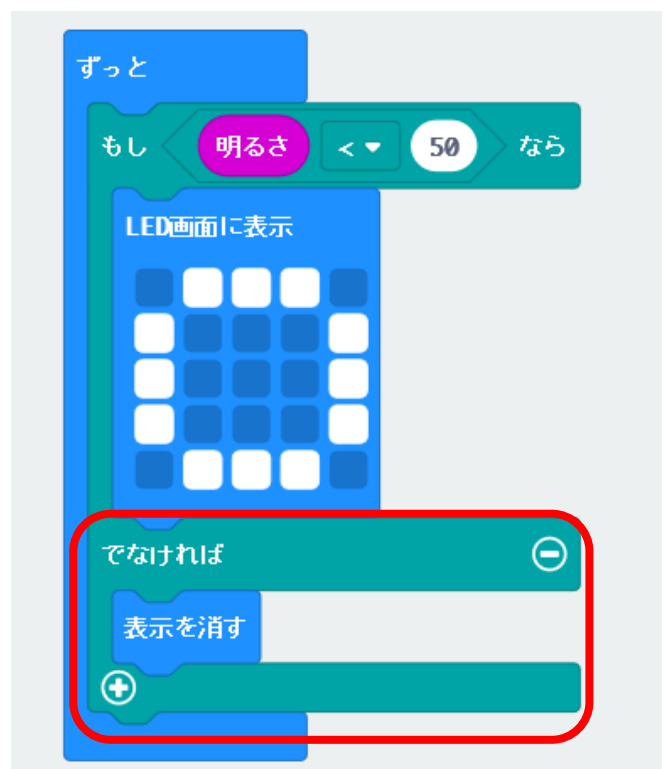
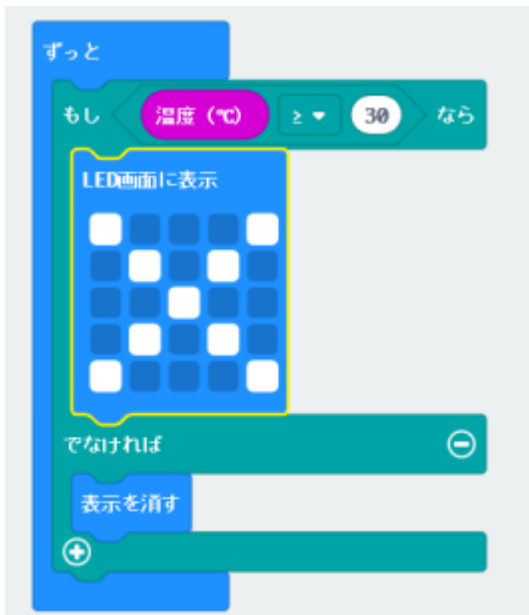


図4-9

- ⑦ シミュレータで確認しましょう。  
明るさを変えて、「○」の表示を確認しましょう。
- ⑧ ⑦で正しく表示されていれば、micro:bit 本体に接続し、明るさを変えて確認しましょう。  
例 懐中電灯で照らす  
光を遮る
- ⑨ シミュレータで確認できたら、micro:bit 本体に転送しましょう。

(3) 温度によって表示を変化させる (発展)

- ① 温度が上昇すると「×」を表示するプログラムを作成しましょう。  
micro:bit 本体のセンサーを利用して、温度の値を取得し、条件による分岐を行います。(2)の⑥のプログラムを修正してプログラムを作りましょう (図4-10)。



(2)の⑥からの変更

- ・センサーの種類  
「明るさ」→「温度」
- ・条件の向き  
「<」→「> =」
- ・条件の値  
「50」→「30」
- ・LEDの画面

図4-10

- ② シミュレータで確認しましょう。  
温度を利用したときは、左側に温度計が表示されます。  
ドラッグして値を動かして、確認しましょう (図4-11)。

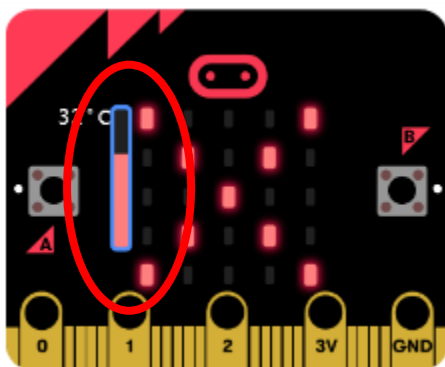
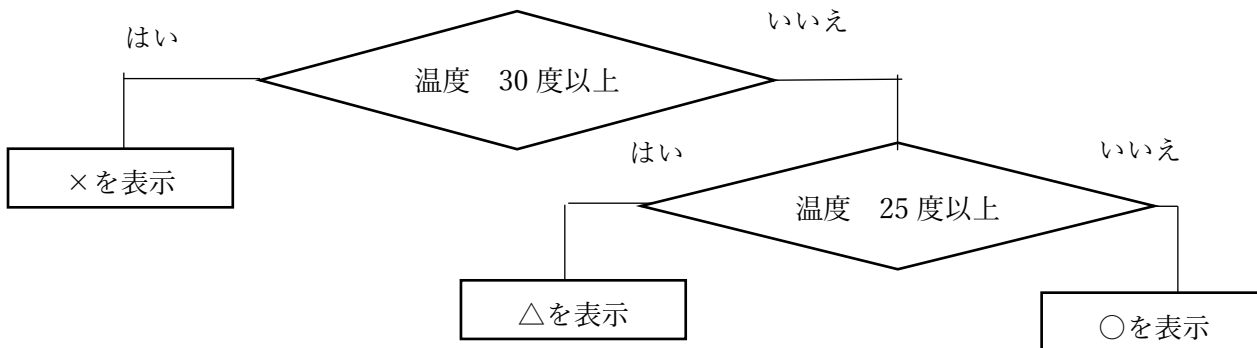


図4-11

- ③ シミュレータで問題がなければ、保存し、micro:bit 本体に転送して確認しましょう。

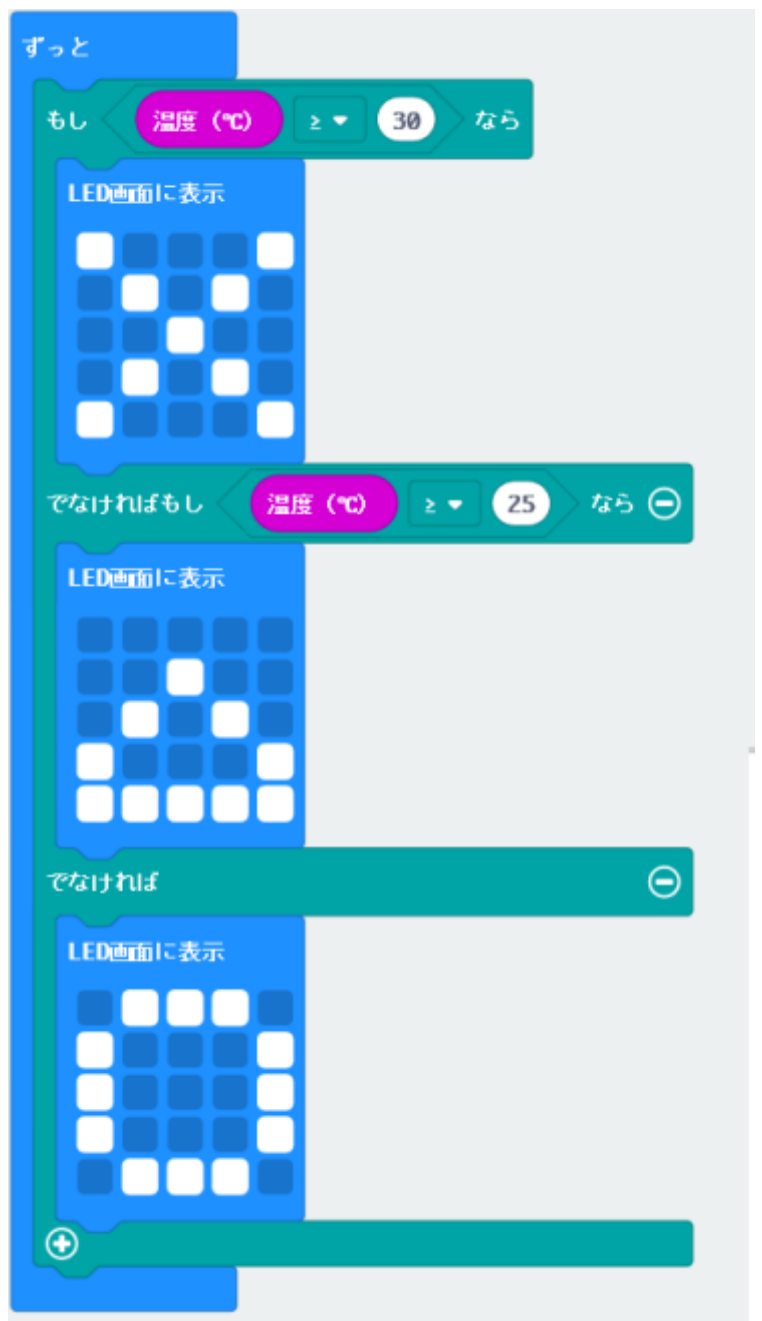
- ④ (発展) 温度により3種類のマークを表示しましょう。  
 30度以上の場合 ×  
 25度以上、30未満の場合 △  
 25度未満の場合 ○  
 3種類のマークを表示しますから、「もし」が2回になります。

<考え方>



<プログラム>

右のようなプログラムになります(図4-12)



- ⑤ シミュレータで確認できたら、micro:bit本体に転送しましょう。

図4-12

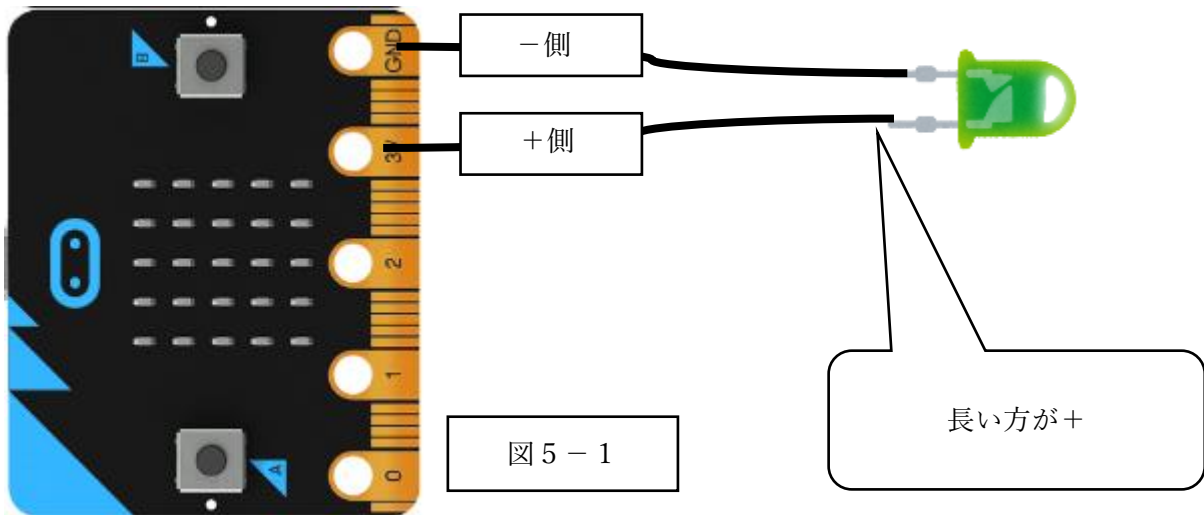
## 5 外部 LED を接続する

### (1) 外部 LED を接続する

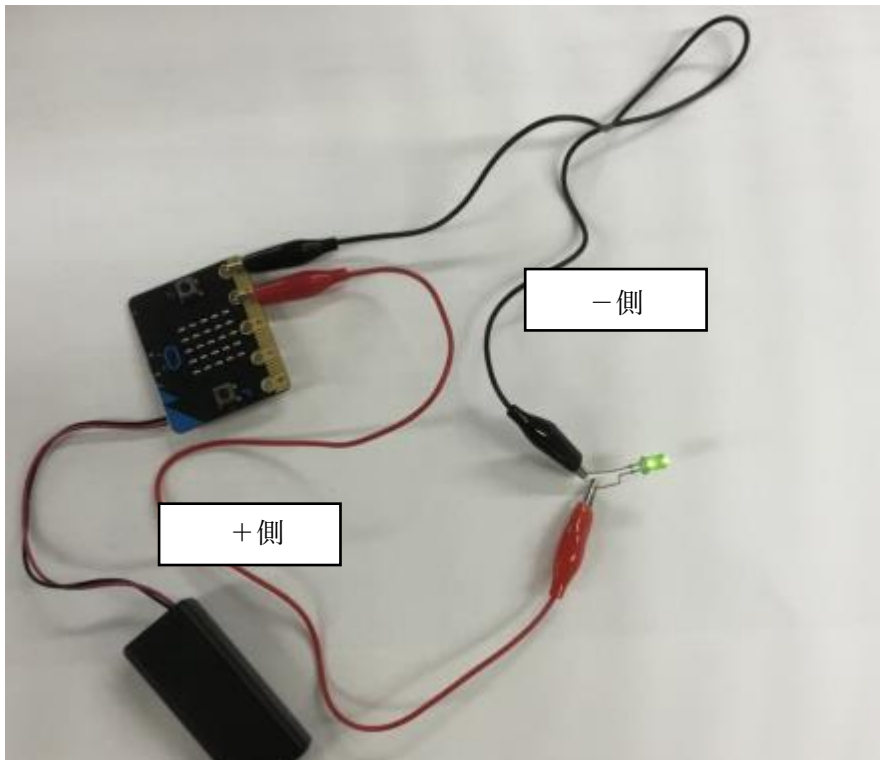
- ① LED を接続して電気が流れていることを確認しましょう (図5-1)。

micro:bit の端子「3 V」に LED のプラス側、「GND」に「マイナス側」を接続して電気が流れることを確認します (LED は長い方が+です)。

なお、個々のLEDには、適正な電圧があります。適正電圧が3V程度の場合は抵抗なしで接続しても大丈夫ですが、より低い電圧のLEDでは間に抵抗をはさまないと、過大な電流が流れてしまい、LED が破壊される恐れがあります。



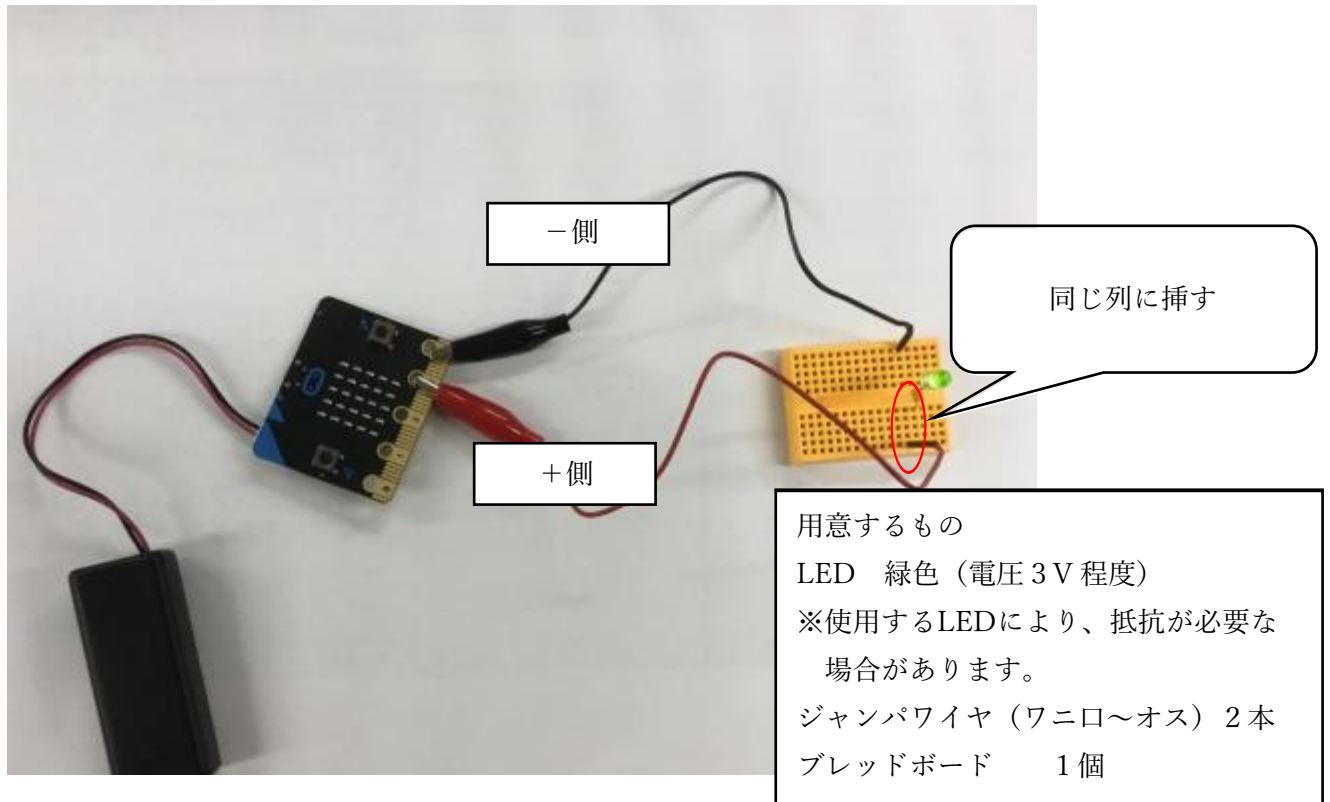
### ○ワニ口クリップによる接続 (写真)



用意するもの  
LED 緑色 (電圧3V程度)  
※使用するLEDにより、抵抗が必要な場合もあります。  
ワニ口クリップ 2本

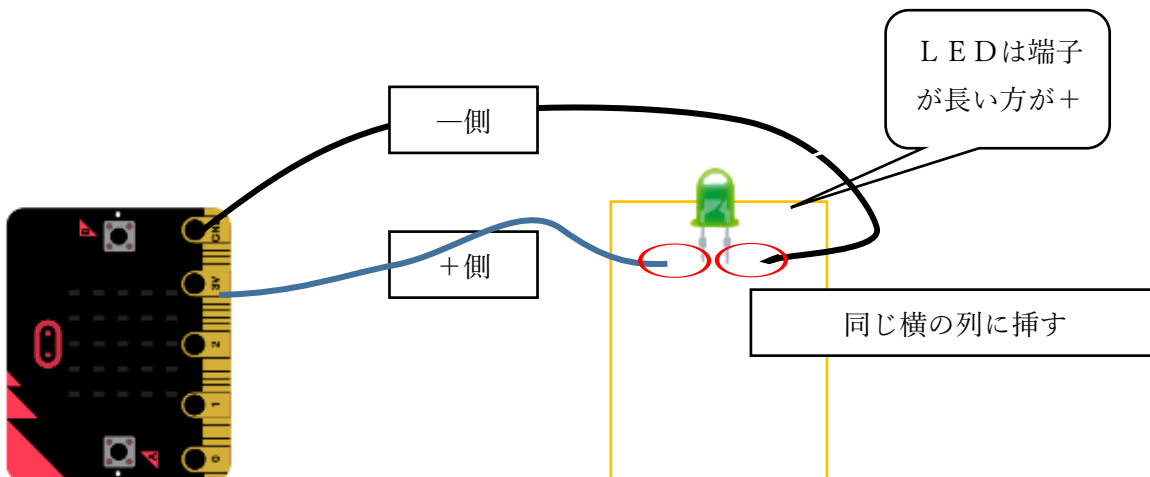
※LED の端子の間が狭いため、クリップが接触しないように気をつけましょう。

## ○ブレッドボードを利用した接続



ブレッドボードは、中で横の列が接続されていますので、同じ列にプラスからの端子、マイナスからの端子、LED を差すことで、電気を流すことが可能です。

## ブレッドボード簡易回路図



(2) micro:bit の入出力端子に接続する

プログラムから制御できる端子は、P0～P2（機械では「0」～「2」）の3つの端子です。

① 端子「P0」に接続して、通电してみましょう。

(1) の3Vに接続していた赤のクリップを「P0」（機械では「0」）につなぎます。

最初の状態では端子「P0」は電圧が0となっているため、次のプログラム（図5-2）を作成し転送して、LEDが点灯することを確認しましょう。

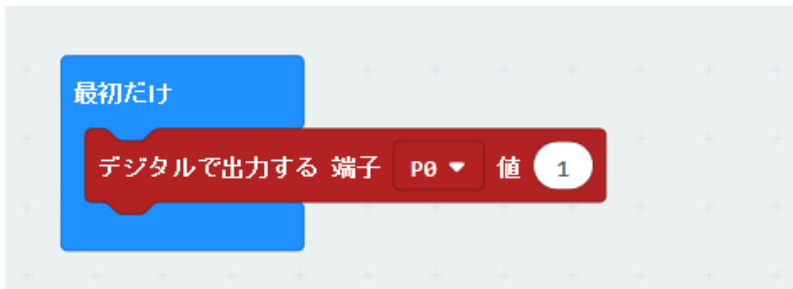


図5-2

これは、「P0」端子の電圧を1（ON）にする命令です。

micro:bit に転送することで、接続している外部LEDが点灯します。

（シミュレータでは、1と0についてのみ確認が可能です）

② 点滅をさせてみましょう。

4の(1)で内蔵LEDを点滅させました。ここでは、図4-4での「LEDの表示」と「表示を消す」の代わりに「P0」端子の状態を変更して、電気の流れを制御します（図5-3）。



図5-3

③ シミュレータで「P0」端子の「1」と「0」が変化していることを確認しましょう。

④ 表示に問題がなければ、micro:bit 本体に転送して、外部LEDが点滅することを確認しましょう。

### (3) 明るさセンサーで制御する

- ① プログラムを作成しましょう (図5-4)。

基本の考え方は4の「内蔵LED」と同様です。

「LED の表示」のかわりに、「P0」端子を制御します。

- ② シミュレータで問題がなければ、micro:bit 本体に転送して確認しましょう。

※内蔵 LED を制御せずに、外部 LEDの制御 から開始するときは、1の(2)の明るさの数値を表示させておきます。



図5-4

### (5) (発展) 温度でLEDの色を変更する

- ① 3色のLEDとmicro:bitと接続しましょう。

赤色LED プラス側 P0 端子

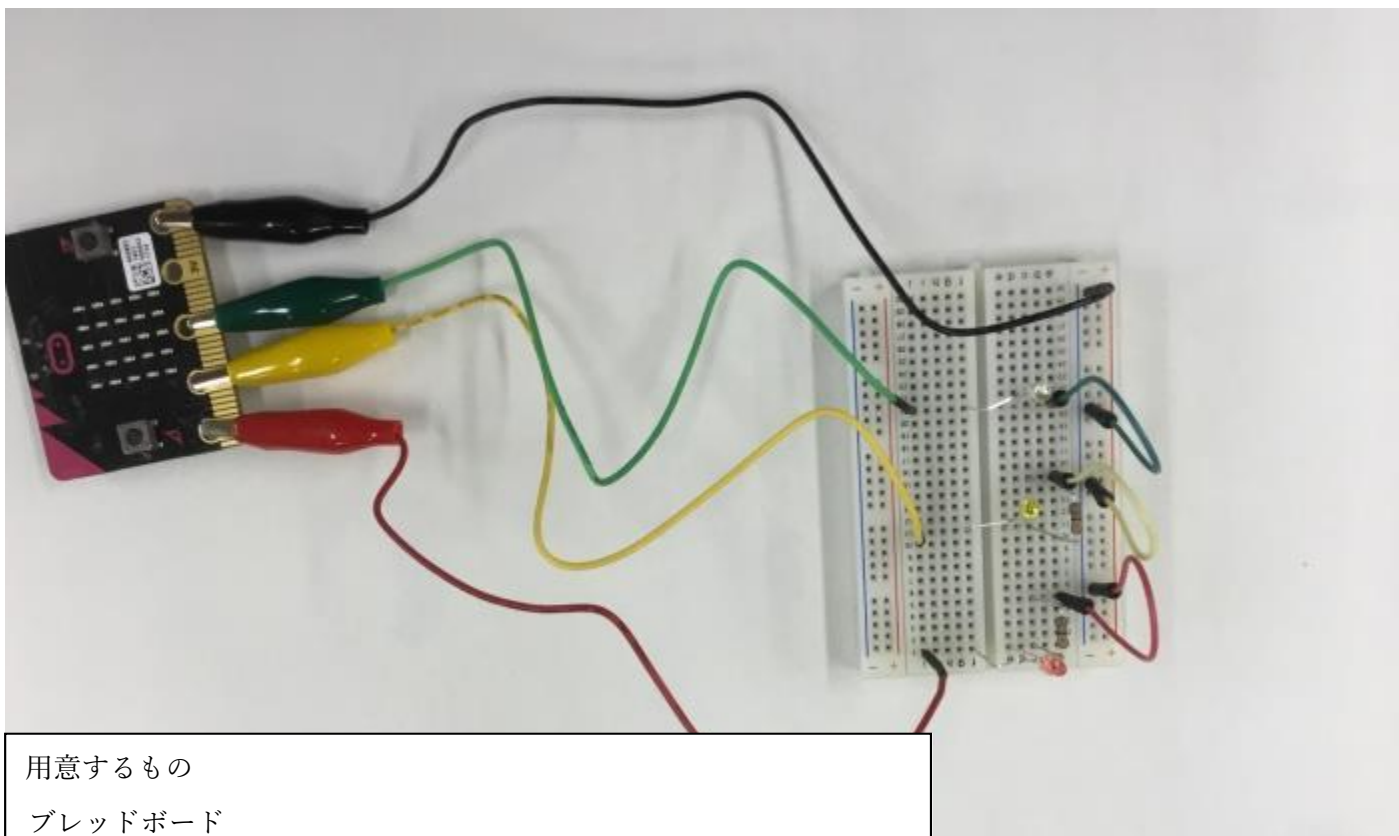
黄色LED プラス側 P1 端子

緑色LED プラス側 P2 端子

すべてのLED マイナス側 GND 端子

LEDの種類によっては、抵抗が必要となるため、ワニ口クリップのみではなく、ブレッドボードや専用の周辺機器を利用すると便利です。

○ジャンプワイヤー（ワニ口～オス）による接続（ブレッドボードによる接続例）



用意するもの

ブレッドボード

ジャンプワイヤ（ワニ口～オス） 4本

ジャンプワイヤ（オス～オス） 3本

LED（緑、赤、黄色）

抵抗 51 オーム 2個（適正な電圧の低いLEDに直列に入れる）

○ブレッドボード利用時の簡易回路図

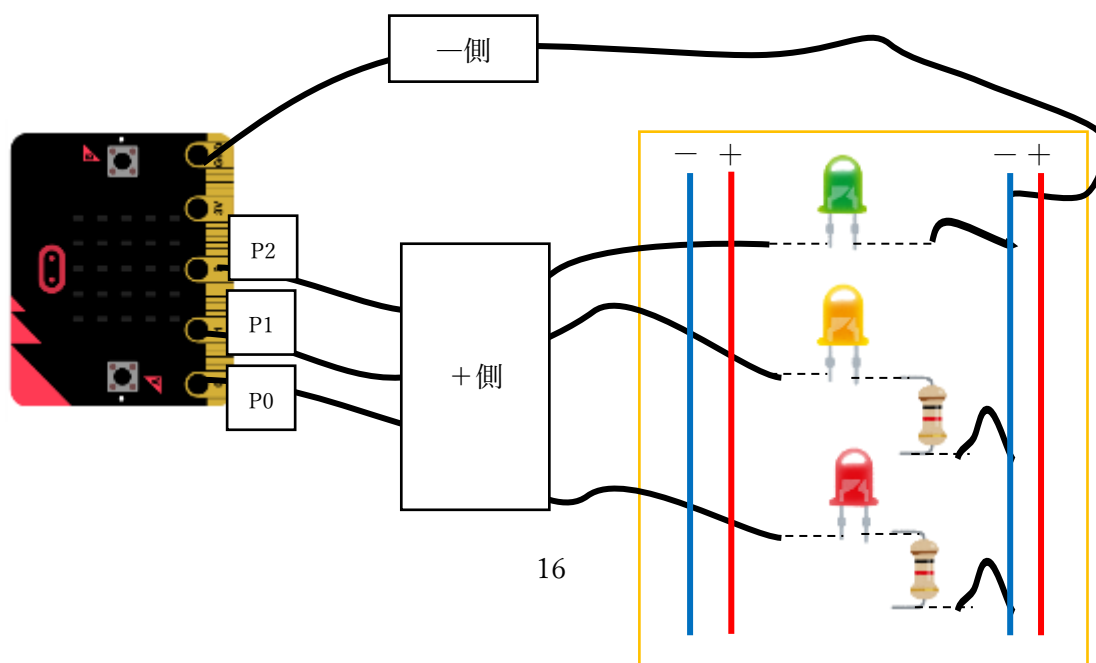
写真のブレッドボードは、内部的に下記の部分がつながっています。

同じ横の列 接続されている。

縦列 赤線部分（+） 接続されている。

縦列 青線部分（-） 接続されている。

したがって、マイナス側は、すべてを青色の線に差し込むことで、回路が接続されます。





② プログラムを作成しましょう。

考え方は内蔵LED使用時の(3)の④と同じです。

内蔵LEDを点灯させる代わりに、それぞれの入出力端子(P0~P2)に対する電圧を変更します。(図5-5)



図5-5

③ シミュレータで「1」「0」の表示を確認後、micro:bit本体に接続して確認しましょう。