

プログラム ～ アプリ編 ～

1 micro:bitとは

(1) BBC micro:bit

プログラムできるマイクロ
コンピュータ

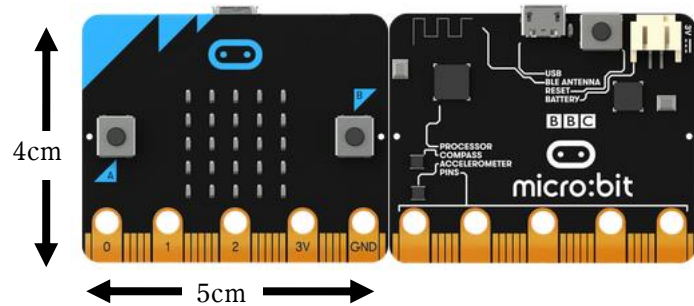
●5×5のLED

●2個のボタン

●センサー

(光、温度、加速度、磁力)

●無線通信機能



2 micro:bitによるプログラム作成


(1) サイトに接続してプログラムを作成する。

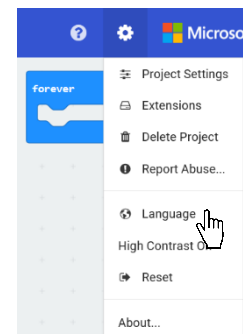
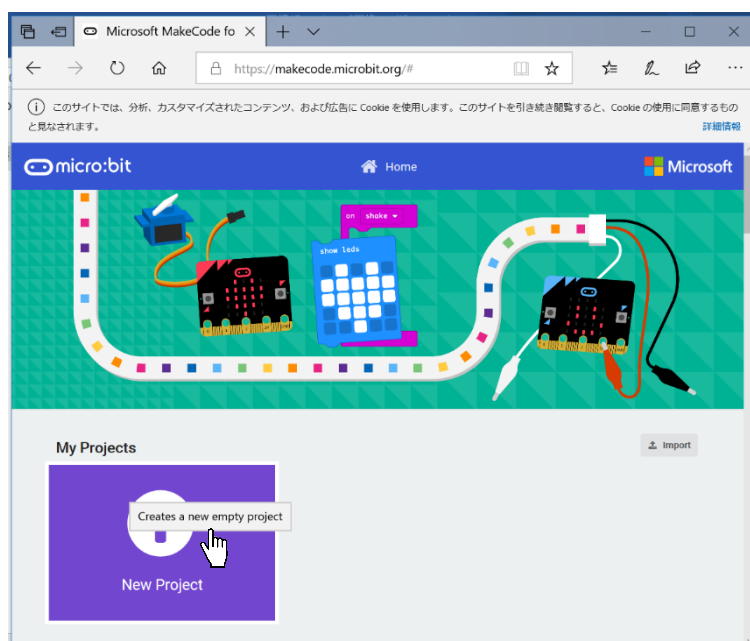
Microsoft Edgeを開き、下のURLをアドレス欄に入力する。

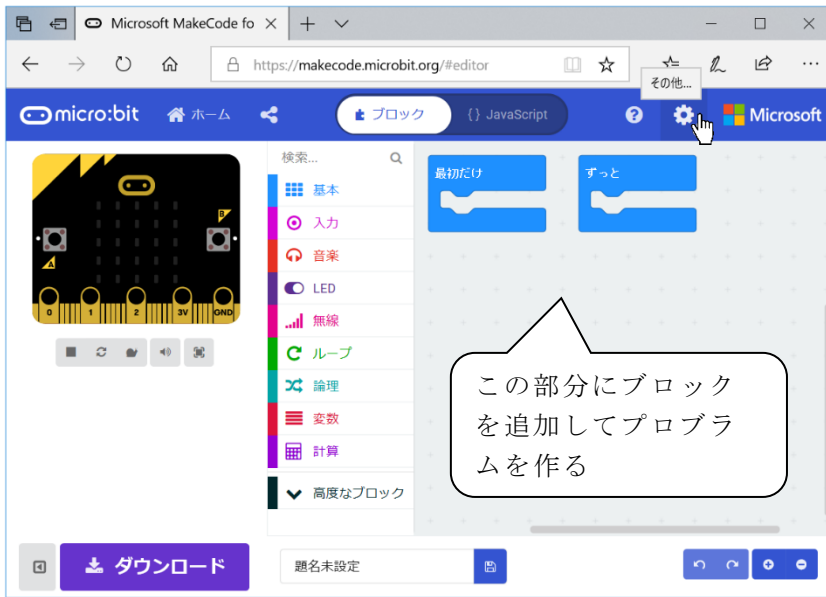
<https://makecode.microbit.org/#>

(2) 「新しいプロジェクト (New Project)」ボタンをクリック

英語表示になっている場合は、「New Project」ボタンをクリックして、次の画面で変更する。


補足 英語表示の場合は、「More...」ボタン  をクリックして、Language (言語) を日本語に変更する。(右下図)





(3) 表示させよう

- ① ハートを表示させる

「基本」→「アイコンを表示」 アイコンを表示  を選択し、ドラッグ&ドロップで、「最初だけ」にはめ込む。

- ② 表示をキリンに変えよう

「アイコン表示」をハートからキリンに変更する。

(4) ボタンで制御しよう

- ① ボタン A を押したらハート

を表示させて、ボタン B を押したら消す。

- ① 「入力」→「ボタンが押されたとき」

されたとき

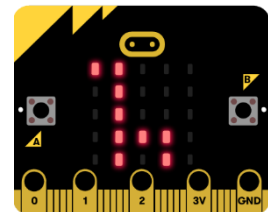
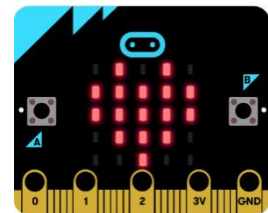
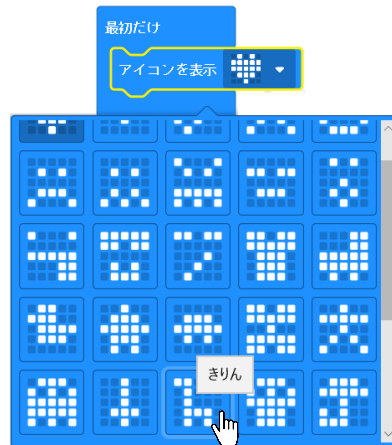


をドラッグする。

- ② 「基本」→「アイコンを表示」をドラッグしてはめ込む。

- ③ もう一度、「入力」→「ボタンが押されたとき」をドラッグして、ボタンを「B」に変更。「基本」→「その他」→「表示を消す」。(完成は右図)

- ④ ボタン A, B を押して試してみよう。



② ボタンを押してカウントダウン

ボタン A の方向を矢印 (←) で示し、ボタン A を押すと、5~0 のカウントダウンが始まるプロジェクトを作ろう。

(図 カウントダウン)

- ① 最初に左向きの矢印 (←) を表示させる。
- ② ボタン A が押されたとき、カウントが 0~5 に変えて繰り返され、それに従って、数 5~0 が表示される。

※ 使用するのは、右のブロック

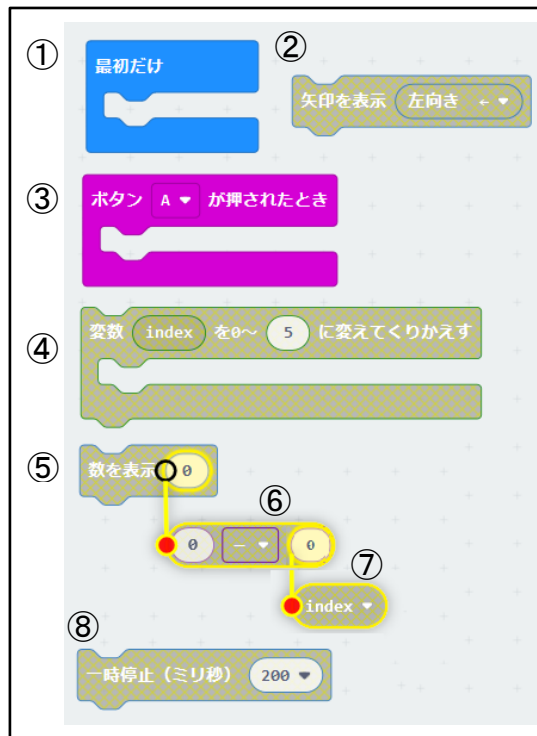
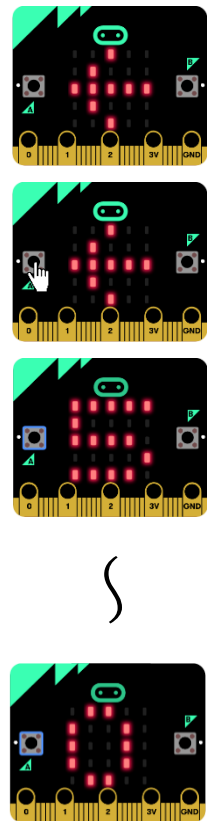


図 カウントダウン



作成手順

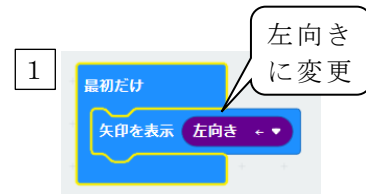
- 1 ①に②を差し込む。
- 2 ③に④を差し込む。(右図)
- 3 ⑤に⑥を、⑥に⑦を差し込む。(右図)
- 4 ③で作成したコマンドを④を差し込む。
- 5 ⑧を③で作成したコマンドの下に挿入する。(右図)

◎プロジェクトに名前を付けて保存しよう。

「このファイルをどのように処理するかを選んでください」で「ファイルを保存する」を選んで、「OK」をクリックする。

追加として、

- ③ カウントダウンの間隔を変えてみよう。
カウントダウンの間隔を長くしてみよう！
また、短くしてみよう！
- ④ もっと大きな数からカウントダウン
もっと大きな数からカウントダウンするように変更しよう。
- ⑤ 音を鳴らしてみよう。
0になったら、音を鳴らしてみよう。



- ③ ボタン A を押したらハートを点滅させて、
ボタン B を押したら消すというプロジェクトを作ろう。

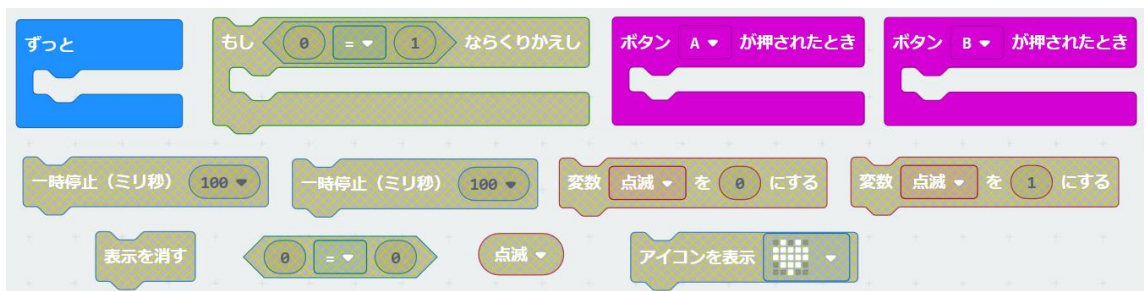
このプロジェクトには、「点滅」という命令は準備されていないので、繰り返し（ループ）、論理、変数（自分で変数を作る）などの命令を組み合わせて作る。
チャレンジしてみよう。

- ◎プロジェクトに名前を付けて保存しよう。

[参考]

- ① 「点滅」という名前の変数を追加する。
- ② ボタン A を押したら、変数「点滅」の値を 1 にする。
- ③ ボタン B を押したら、変数「点滅」の値を 0 にする。
- ④ 「ずっと」（コードをずっと繰り返す）の鍵の中に、
ループ「もし<真>なら繰り返す」を使用して、ハートを点滅させる。
- ⑤ ④の<真>には、『変数「点滅」=1』を設定する。
- ⑥ ④の鍵の中の繰り返しには、「ハートを表示」「一時停止（ミリ秒）」「表示を消す」「一時停止（ミリ秒）」を入れる。一時停止時間は各自の判断で。

上の①～⑥の手順でプロジェクトを作る場合、以下のコマンドを使用することになる。



3 micro:bitによる「ブロックキャッチ」ゲームを作成しよう

(1) ゲームの概要

乱数を利用して1行目（最上段）のどれかの位置にブロックを表示（LEDを点灯）させる。そのブロックが4段目までそのままの列を落下する。1段落下すると、次に新しくブロックが登場して落下を始める。

5行目（最下段）に、キャッチするためのブロックがあり、Aボタン、Bボタンを使って左右に移動させてキャッチする。キャッチできたら音を鳴らし、回数または時間で制御して終了する。A+Bボタンで終了することもできるようにする。最後にキャッチできたブロックの数を得点として表示させる。

(2) 準備 1 ブロック落とし I

まず、1行目（最上段）のどれかの位置にブロックを表示（LEDを点灯）させて、そのブロックが4段目までそのままの列を落下するプログラムを作ってみよう。

LEDの座標指定(x,y)

1行目	(0,0)	(1,0)	(2,0)	(3,0)	(4,0)
2行目	(0,1)	(1,1)	(2,1)	(3,1)	(4,1)
3行目	(0,2)	(1,2)	(2,2)	(3,2)	(4,2)
4行目	(0,3)	(1,3)	(2,3)	(3,3)	(4,3)
5行目	(0,4)	(1,4)	(2,4)	(3,4)	(4,4)

変数として、1行目～4行目のx座標を指定する変数として、B0, B1, B2, B3を使用することにする。

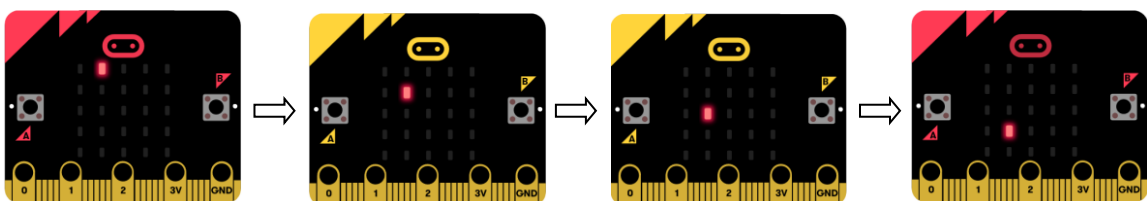
④ 「ずっと」に組み込む。

⑤ 変数「カウンター、B0」を追加して、変数B0を「0～4までの乱数」にする。

⑥ 変数カウンターを0～3に変えて繰り返す。

⑦ 繰り返して、②で決定した列に1行目から4行目までLEDを表示させ、間をあけて消灯する。

(下図参照)



(3) 準備2 ブロック落としⅡ

(2)準備1のブロック落としを連続的に繰り返すプログラム，すなわち

1行目のいずれかの位置にブロックを表示させて，そのブロックが1段落下すると次のブロックが1行目に現れる，これが繰り返されるプログラムを作ってみよう。

① 変数「B0, B1, B2, B3」を追加して，

「最初だけ」において，変数 B0～B3 を「5」にする。

② 4行目の表示を消して，3行目のブロック位置を4行目に表示させる。

次に3行目の表示を消して，2行目のブロック位置を3行目に表示させ，2行目の表示を消して，1行目のブロック位置を2行目に表示させる。

1行目の表示を消して，乱数を使って，1行目のブロック位置を決定して表示させる。

具体的には，「もし<真>ならくりかえし」を使って

❶ 消灯 $x(B3)y(3)$

〔補足〕 <真>の条件は後から入れる

❷ 点灯 $x(B2)y(3)$

ここはそのままにしておく

❸ 消灯 $x(B2)y(2)$

❹ 点灯 $x(B1)y(2)$

❺ 消灯 $x(B1)y(1)$

❻ 点灯 $x(B0)y(1)$

❼ 変数「値」を0～4までの乱数にする

❽ 消灯 $x(B0)y(0)$

❾ 点灯 $x(\text{値})y(0)$

❿ この後に，時間をおくために，

・「一時停止(ミリ秒)1000」を入れる。

さらに，変数の値を次のように代入する。

・ $B3 \leftarrow B2$ (変数 B3 を B2 にする)

・ $B2 \leftarrow B1$

・ $B1 \leftarrow B0$

・ $B0 \leftarrow \text{値}$

以上で，1行目に発生したブロックが連続的に落下するプログラムができる。

(4) ブロックキャッチゲーム

準備2で作成したプログラムに次の設定を追加しよう。

繰り返しの制御を設けた上で（今回は回数制限を設けた上で），ブロックをキャッチしたら得点を1ポイント増やし，終了したら得点を表示するように設定する。

また，そのために新たに3つの変数「count」（回数），「player」（ブロックを受け取る位置），「point」（得点）を追加する。

A ボタンが押されたら、最下段（5 行目）の player の位置（x 座標）が左に移動し、
B ボタンが押されたら、player の位置が右に移動するようにする。ただし、それ以上移動
できない場合の条件設定が必要である。

A+B ボタンが押されたときに、ゲームが終了できるように設定する。

- ① 変数 count, player, point を追加する。
- ② 「最初だけ」のブロックで、変数 count を 1 に、変数 player を 0 に、変数 point を 0
にする。
- ③ ループ「もし<真>ならくりかえし」の条件判断として、
<真>を `count > 0` かつ `count < 100` に設定しよう。

この条件判断の最後に、

- ① 「変数 count を 1 だけ増やす」を追加する。
- ② ①のあとに、条件判断「もし<真>なら」を使って、
「player = B3」（受け取るブロックの位置と 4 行目のブロックの位置が同じ）
のとき、「変数 point を 1 だけ増やす」を設定する。
得点が入ったことを知らせるために、「メロディーを開始する（ビコーン!）」を
〔1 度だけ〕を追加する。
- ④ ループの繰り返しが終了した後に、
「文字列を表示（"POINT"）」
「数を表示（point）」を入れる、
- ⑤ 入力「A ボタンが押されたとき」において、
条件判断「もし（player > 0）なら」に、次のように設定する。
この設定により、A ボタンが押されると player の位置が左へ移動する。
 - ① 「消灯 x(player) y(4) 」
 - ② 「変数 player を player - 1 にする」
 - ③ 「点灯 x(player) y(4) 」
- ⑥ 入力「B ボタンが押されたとき」において、
「A ボタンが押されたとき」と同様に、プレイヤーのブロックの位置が右に移動するよ
うに設定する。
※注 A ボタンの場合は左への移動であるが、B ボタンの場合は右への移動になるの
で、条件判断の「条件」や変数の値の「代入」などは変更する必要がある。
- ⑦ 入力「A+B ボタンが押されたとき」において、
「変数 count を 1000 にする」（繰り返しの回数制限以上の値にする）
とする。

以上で、プログラム完成。走らせてみよう。

(5) サンプルプログラムでいろいろなゲームにチャレンジ

次のサイトにサンプルプログラムがある。

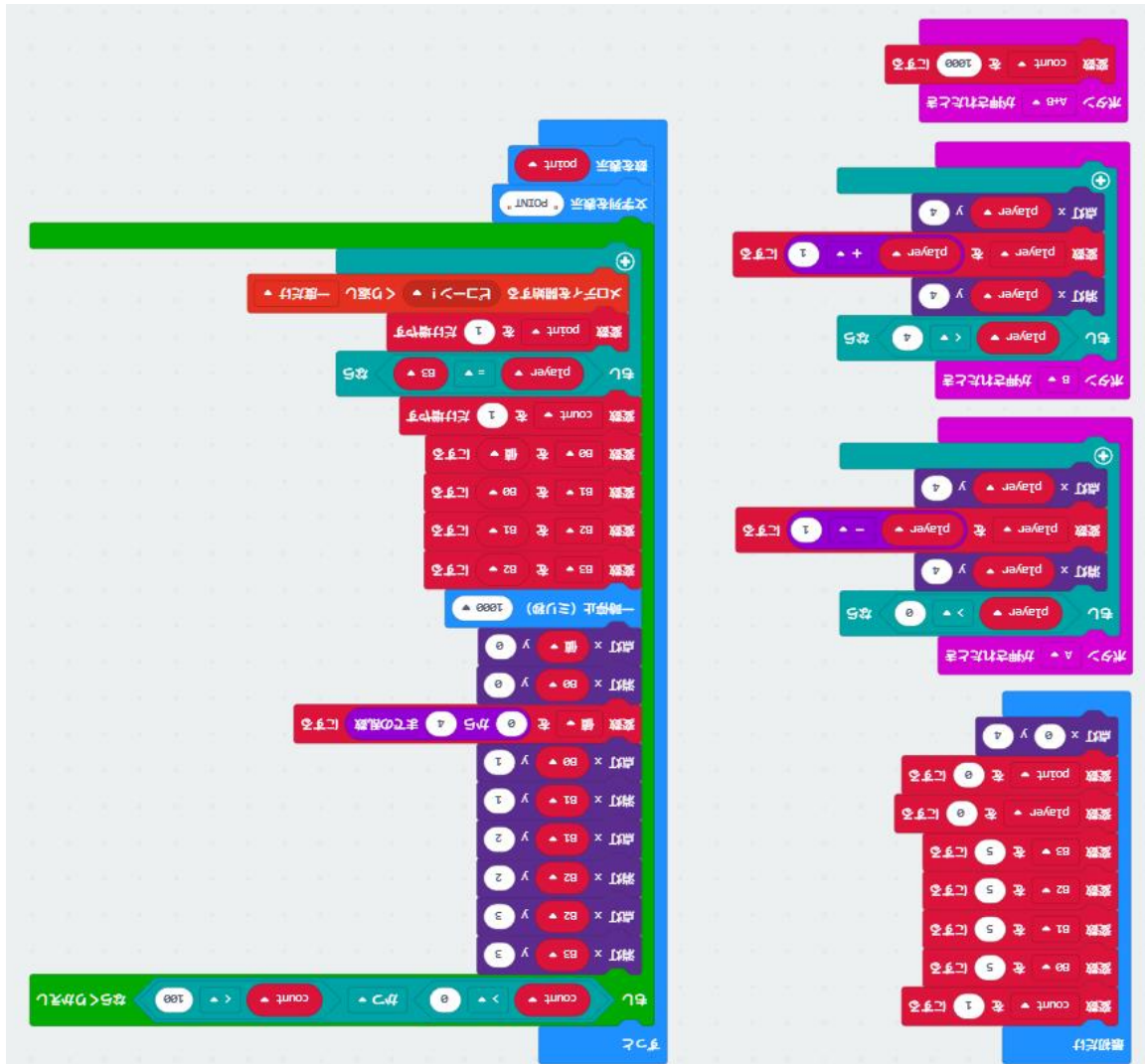
<https://sanuki-tech.net/micro-bit/appendix-sample-program/>

「microbit サンプル」で検索をして、

「micro:bit【マイクロビット】学んで遊べるサンプルプログラム | micro:bit Lab ...」

をクリックすれば、サンプルプログラムのページにアクセスできる。

プログラムを入力して試してみよう。自分なりにアレンジしてみるとよい。



解答例