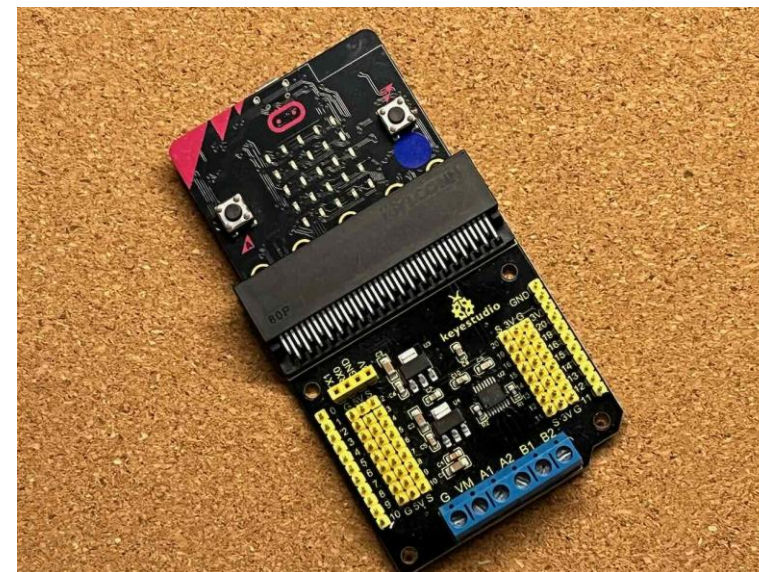
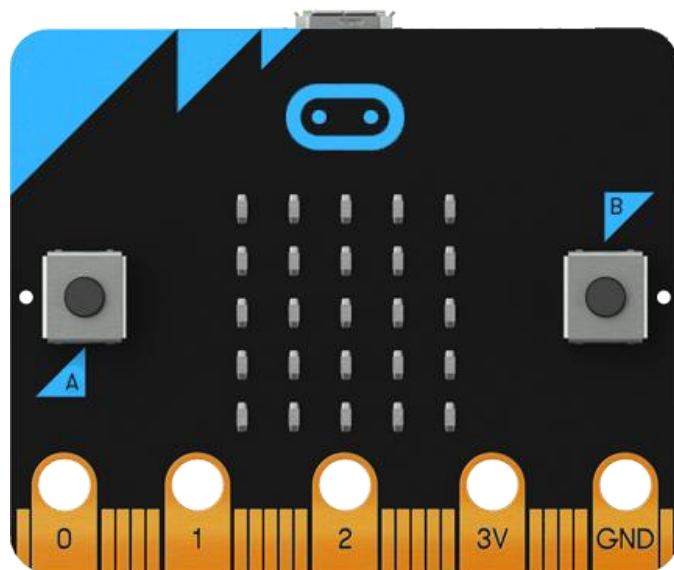


さとやまプログラミングクラブ2026

# マイクロビットプログラミング ～基礎～

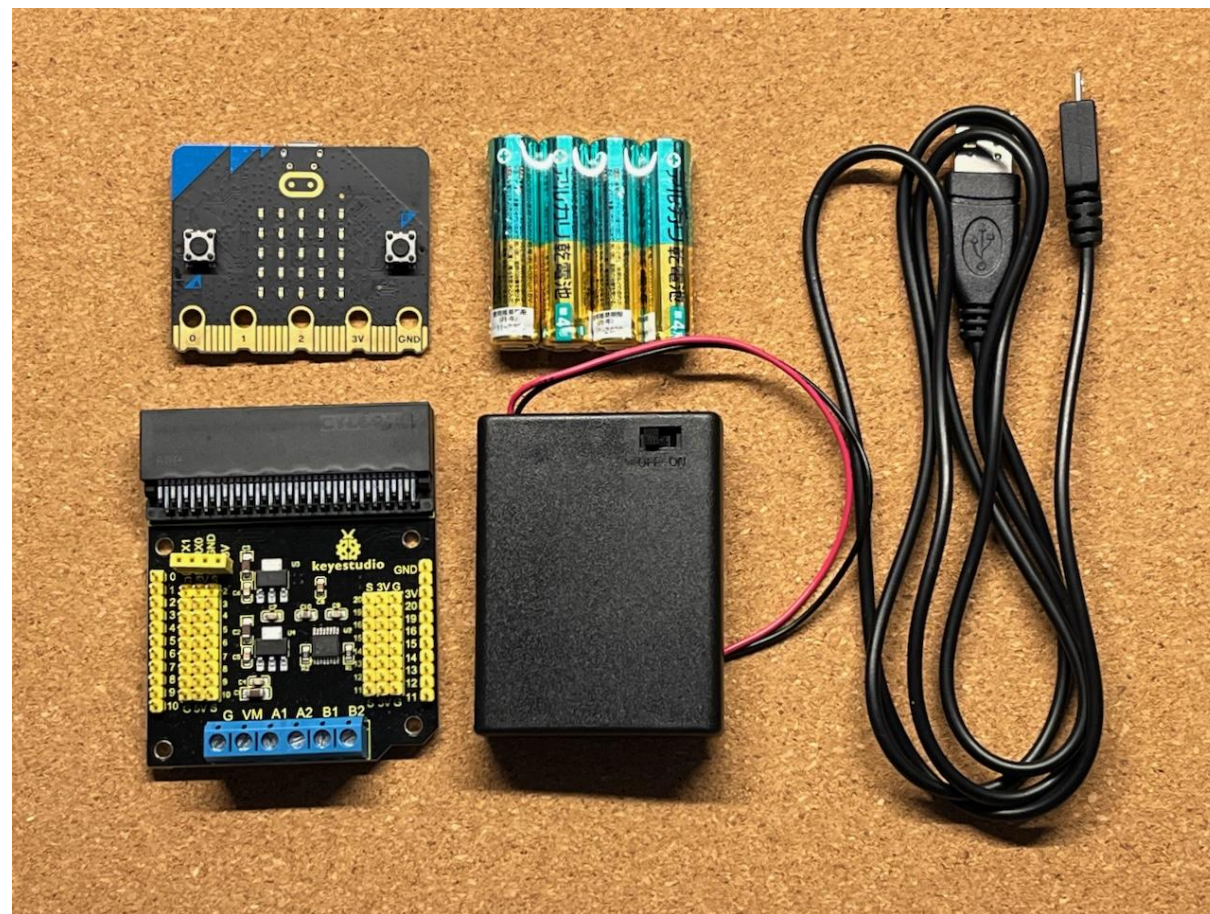


# 今日やること

- 「マイクロビット」についてまなびます。
- パソコンでマイクロビットのプログラムをつくってみます。
- マイクロビットにプログラムをかきこんでみます。
- マイクロビットで「キッチンタイマー」をつくり、プログラミングの基礎についてまなびます。

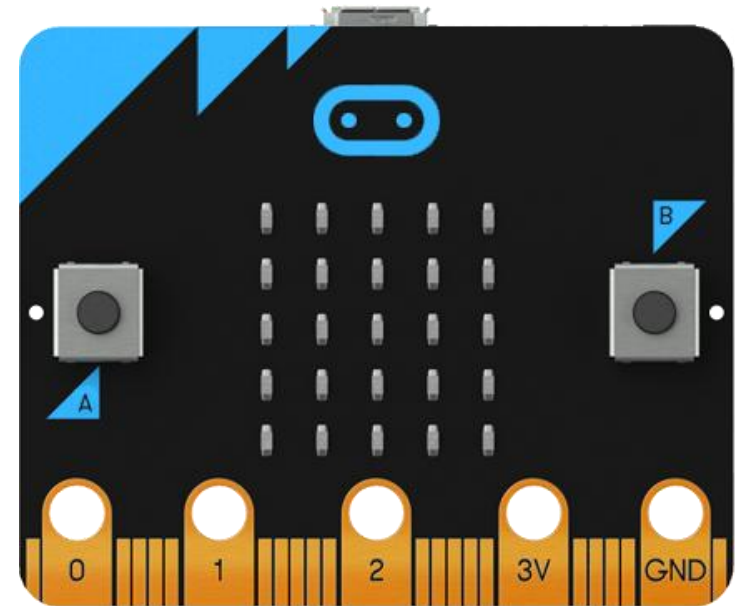
# 部品の確認

部品	個数
マイクロビット	1
拡張ボード (KEYE STUDIO KS4033)	1
USBケーブル	1
単三x4電池ボックス	1
単三乾電池	4



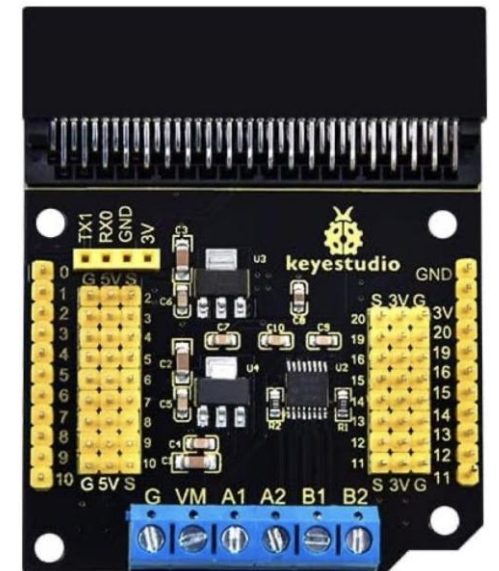
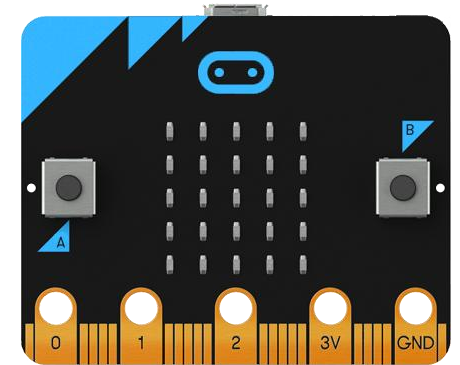
# マイクロビットについて

- マイクロビットは、ちいさなコンピュータです。
- プログラムはパソコンで作り、それをマイクロビットにかきこんでうごかします。
- マイクロビットにプログラムをかきこんだあとは、パソコンからとりはずしても、マイクロビットだけでうごきます。
- LEDやセンサなどの部品がついていて、いろいろなプログラムをつくってうごかすことができます。



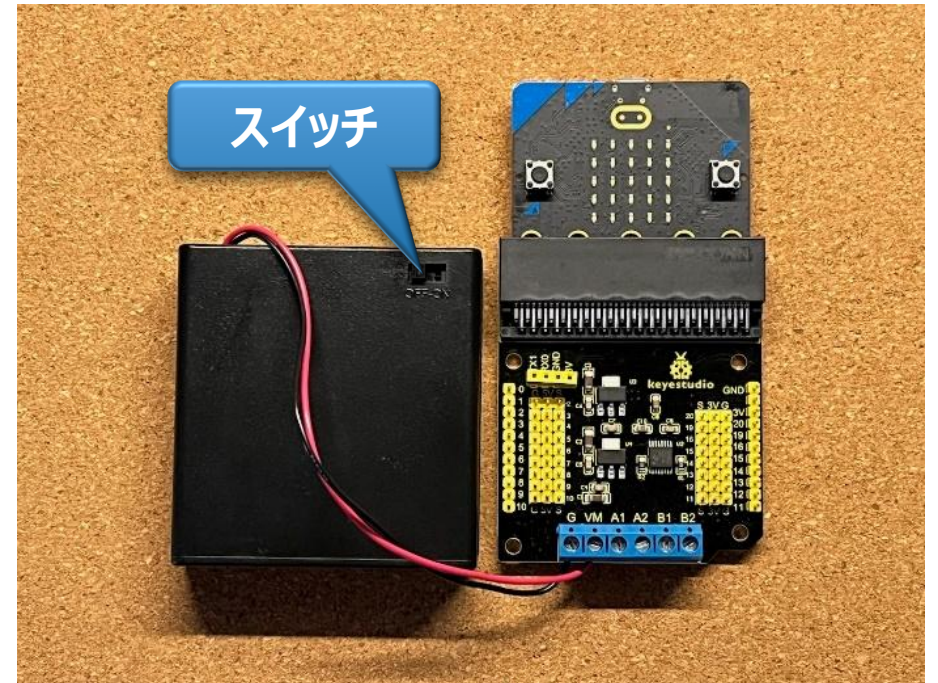
# 拡張ボードについて

- 今回は、マイクロビットを「拡張ボード」にとりつけてつかいます。
- 拡張ボードに電池ボックスをとりつけ、電池でマイクロビットをうごかします。
- 拡張ボードには端子がたくさんついており、それをつかってマイクロビットとほかの部品をつなぐことができます。



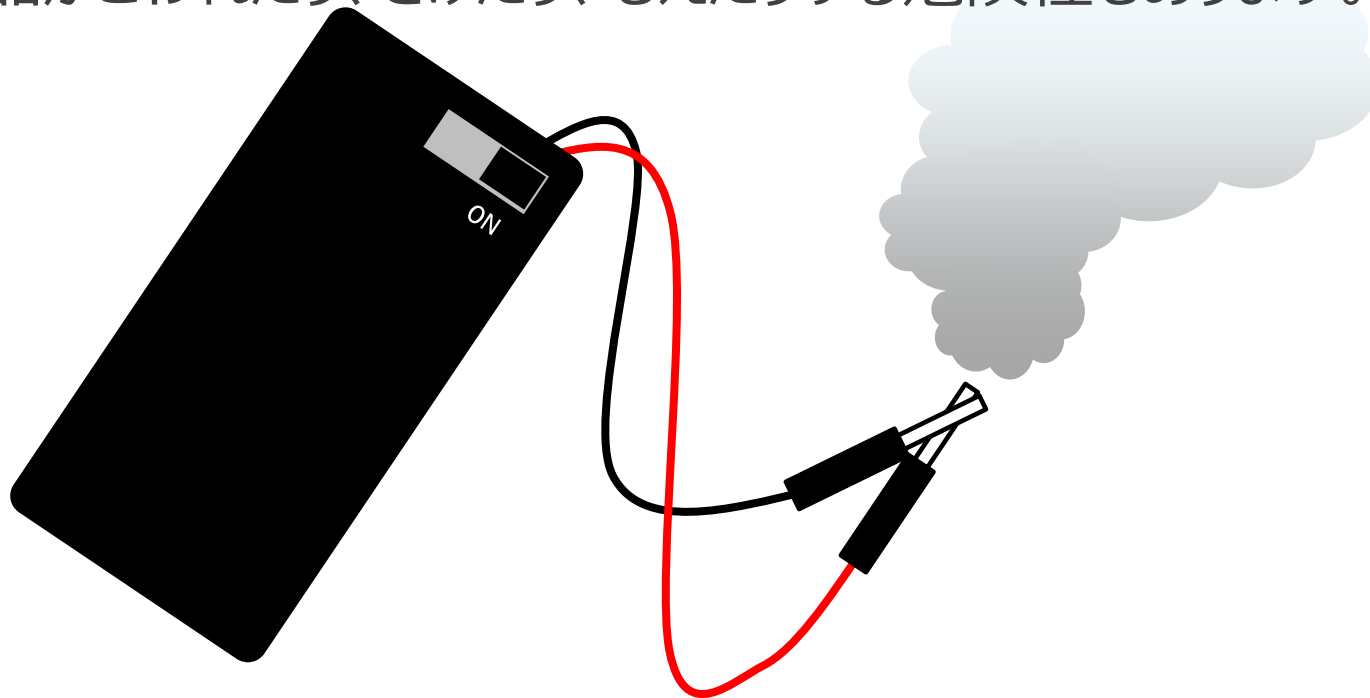
# 準備作業

- マイクロビットを拡張ボードにさしこんでください。
- 電池ボックスの黒いコードを、拡張ボードの青い端子のうち「G」とかいてあるところにとりつけてください。
- 電池ボックスの赤いコードを、拡張ボードの青い端子のうち「VM」とかいてあるところにとりつけてください。
- 電池ボックスのフタをあけて電池をいれてください。プラスとマイナスをまちがえないようにしてください。



## 注意事項

- 使わないときは、電池ボックスのスイッチをオフにしてください。
- オンのままにしていると、電池がすぐになくなるだけでなく、もしもプラスとマイナスがショートすると、部品がこわれたり、とけたり、もえたりする危険性もあります。

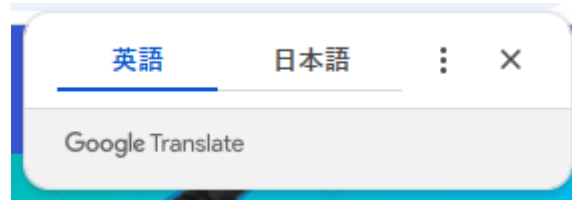


# MakeCodeエディタのたちあげかた

- マイクロビットのプログラムは「MakeCodeエディタ」で作ります。
- Webブラウザ（Edge, Chrome, Safariなど）で、以下のページをひらきます。

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

- もしもウィンドウの右上にこのようなフォームが表示されたら、「×」をクリックしてとじるか、「⋮」をクリックして「このサイトは翻訳しない」をえらびます。



# MakeCodeエディタのたちあげかた

- MakeCodeエディタのホーム画面です。
- 「新しいプロジェクト」をクリックすると、プロジェクトに名前をつけるウィンドウが表示されます。
  - ❖ 「プロジェクト」というのがプログラムのことです。
- すきな名前を入力して「作成」をクリックすると、あたらしいプログラムをつくる画面がひらきます。

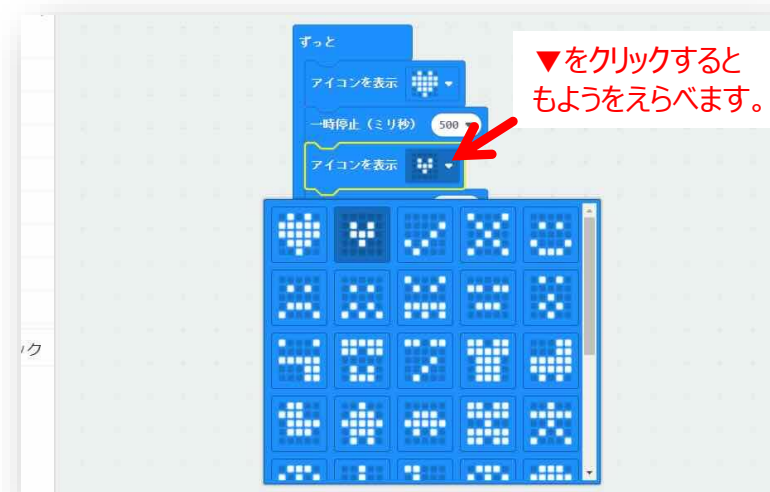
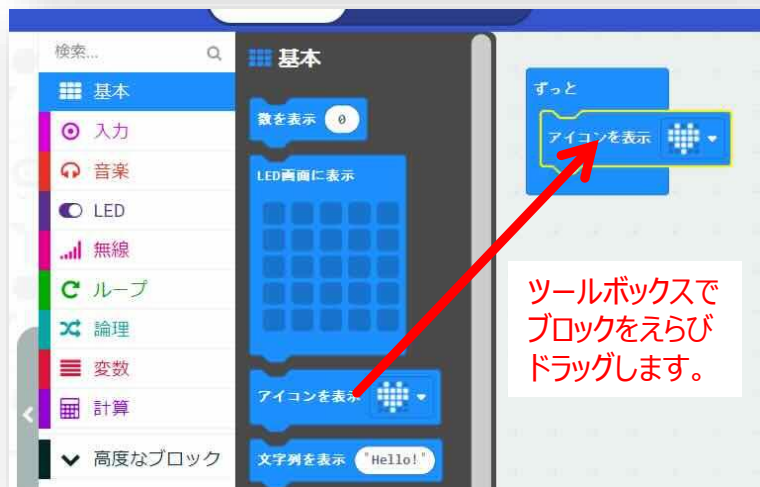


# プログラム作成画面

The screenshot shows the Microsoft MakeCode editor for Micro:bit. The browser address bar displays `https://makecode.microbit.org/#editor`. The interface features a blue header with the 'micro:bit' logo, navigation links for 'ホーム' (Home) and '共有' (Share), and tabs for 'ブロック' (Blocks) and 'JavaScript'. A search bar is located in the top center. On the left, a simulator shows a Micro:bit board with pins labeled 0, 1, 2, 3V, and GND. Below the board are icons for running, refreshing, and saving. The center panel contains a '検索...' (Search) bar and a list of block categories: 基本 (Basic), 入力 (Input), 音楽 (Music), LED, 無線 (Wireless), ループ (Loops), 論理 (Logic), 変数 (Variables), 計算 (Math), and 高度なブロック (Advanced Blocks). The right panel shows a workspace with a grid background and two buttons: '最初だけ' (Only at the start) and 'ずっと' (Forever). Three blue callout boxes are overlaid on the interface:

- シミュレータ**  
マイクロビットに書きこむ前にプログラムの動作を確認できます。
- ツールボックス**  
いろいろなブロックが  
おいてあります。
- ワークスペース**  
ここでプログラムをつ  
くれます。

# プログラムのつくりかた

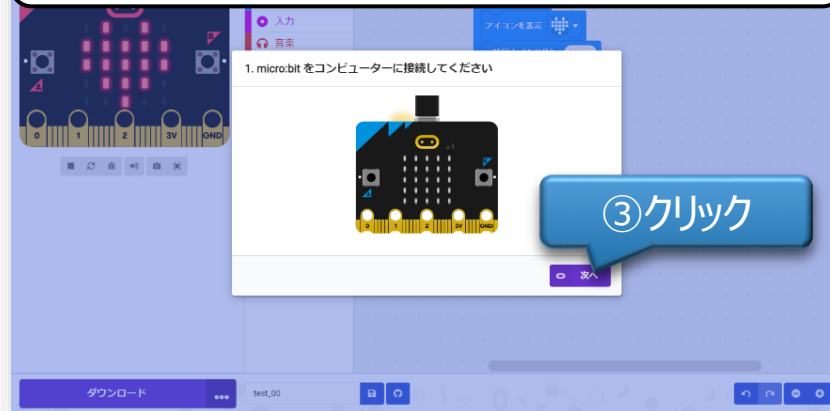


# マイクロビットへのプログラムのかきこみかた

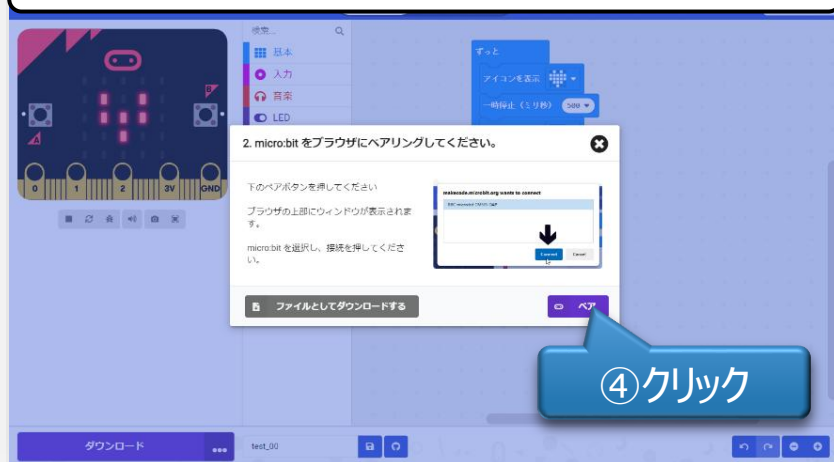
①「ダウンロード」をクリックします



② PCとマイクロビットをUSBケーブルでつなぎます  
③「次へ」をクリックします



④「ペア」をクリックします



⑤「BBC micro:bit」をえらびます  
⑥「接続」をクリックします



# マイクロビットへのプログラムの書きこみかた

⑦「ダウンロード」をクリックします



⑧ 2回目からは画面左下の「ダウンロード」のクリックだけでダウンロードできます



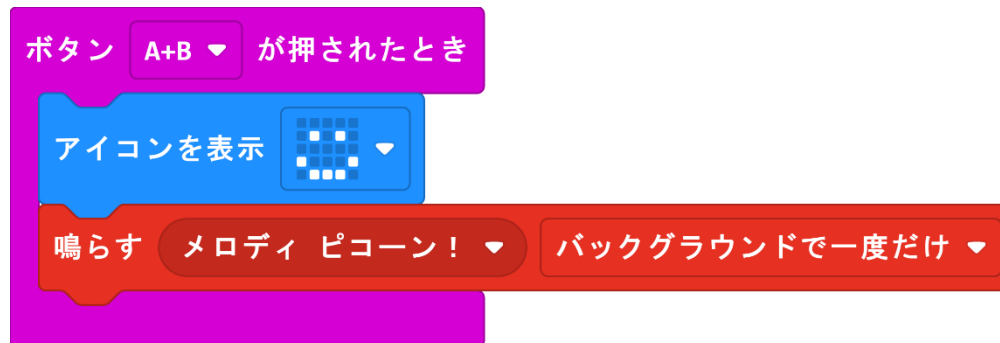
# キッチンタイマーをつくろう！

- 今回は、みなさん自身に考えてもらって、プログラムをつくってもらいます。
- つくるのは「**キッチンタイマー**」です。
  - AボタンとBボタンをいっしょにおすと、タイマーがうごきはじめます。
  - 時間がくると音がなり、アイコンがひかります。
  - AボタンとBボタンで、はかる時間をかえられるようにします。
  - タイマーがうごいている間はアイコンがひかるようにします。
- パスタのゆで時間やカップラーメンをつくる時間をはかるのに役立ちます。

# さいしょのプログラム

- AボタンとBボタンをいっしょにおしたら、「えがお」アイコンがひかり、「ピコーン！」と音になるプログラムをつくってください。
  - 「入力」に「ボタン A が押されたとき」ブロックがあります。「A」のところをクリックすると「A」「B」「A+B」をえらぶことができます。
  - 「基本」に「アイコンを表示」ブロックがあります。アイコンをクリックすると、好きなアイコンをえらぶことができます。
  - 「音楽」に「鳴らす メロディダダダム バックグラウンドで一度だけ」ブロックがあります。「ダダダム」のところをクリックすると、好きなメロディをえらぶことができます。
- **プログラムができたなら、ただしくごくかシミュレータでたしかめてください。**

# さいしよのプログラム ～作成例



# 改良プログラム 1

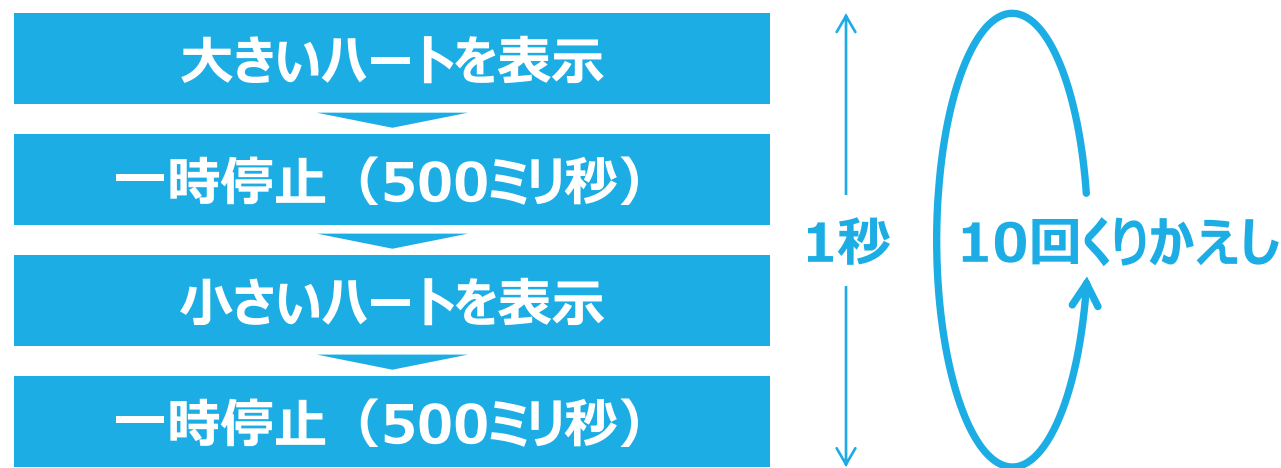
- AボタンとBボタンをいっしょにおしてから、アイコンがひかるまでに、10秒まつように改良してください。
  - 「基本」に「一時停止（ミリ秒）」ブロックがあります。
  - 「ミリ秒」とは、1/1000秒のことです。なので、1秒まちたいときには「1000」と入力します。
- **プログラムができたら、ただしくごくかシミュレータでたしかめてください。**

# 改良プログラム 1 ～作成例



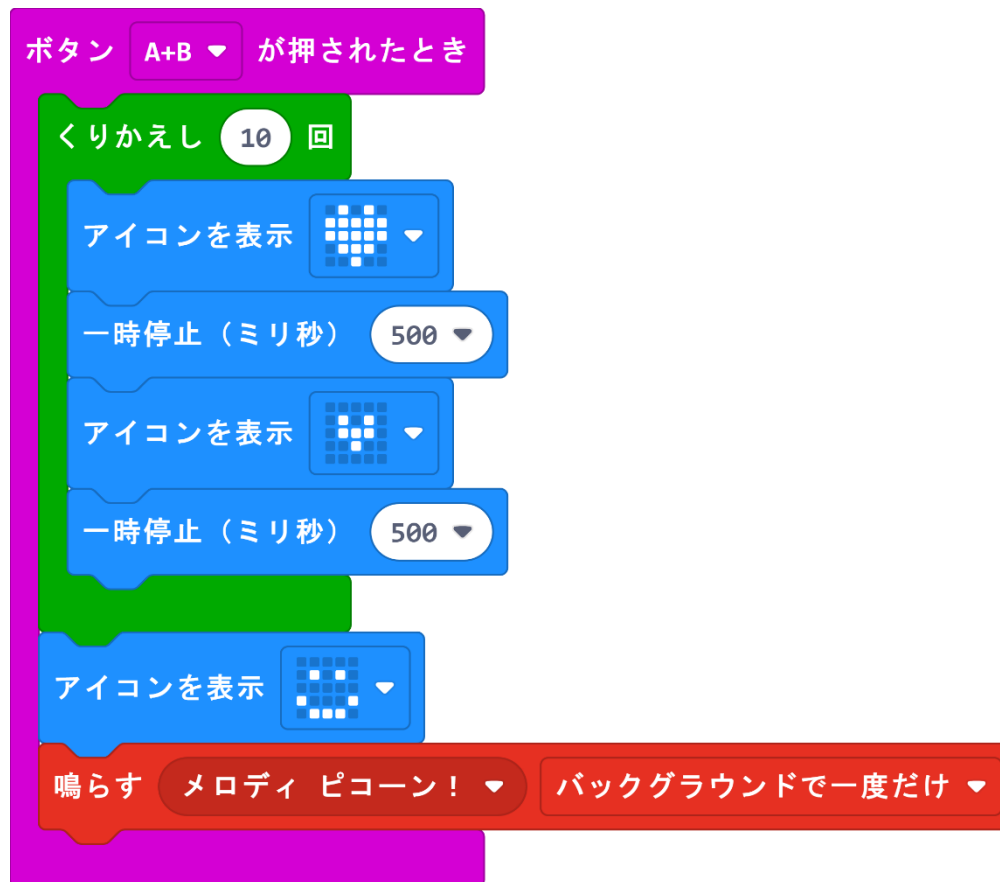
## 改良プログラム 2

- 10秒まっているあいだ、なにもおこらないのはつまらないので、そのあいだ、0.5秒おきに「ハート」アイコンが大きくなったり小さくなったりするように改良してください。
  - 「ループ」に「くりかえし 4 回」ブロックがあるので、これをつかってください。
  - 以下のうごきを、「くりかえし」ブロックで10回くりかえします。



➤ プログラムができたなら、ただしくごくかシミュレータでたしかめてください。

# 改良プログラム 2 ～作成例



ボタン A+B が押されたとき

- くりかえし 10 回
  - アイコンを表示
  - 一時停止 (ミリ秒) 500
  - アイコンを表示
  - 一時停止 (ミリ秒) 500
- アイコンを表示
- 鳴らす メロディ ピコーン! バックグラウンドで一度だけ

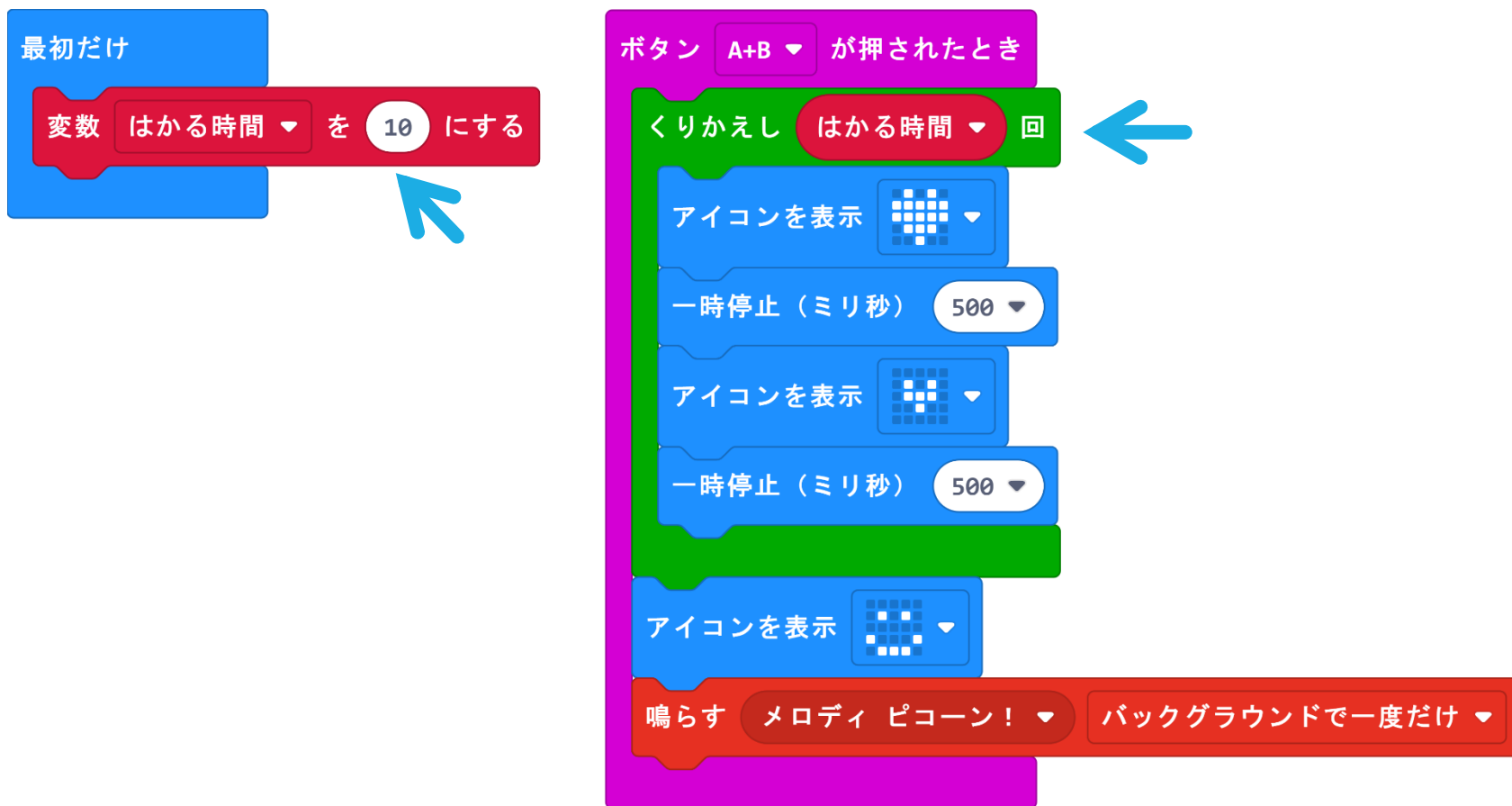
## ここまでの確認

- AボタンとBボタンをいっしょにおしたら「ハート」アイコンが10秒間ピカピカし、そのあとで「えがお」アイコンがひかって「ピコーン！」と音になるプログラムができました。
- これで「**キッチンタイマー**」のできあがりです！
- ただし、このキッチンタイマーではかえる時間は「10秒」だけです。
- はかる時間をかえられるようにするために「**変数**」をつかいます。「変数」というのは、プログラムの中でかえることができる値のことです。

## 改良プログラム 3

- さっきのプログラムで、くりかえしの回数で「10」と書いていたところを、変数をつかうように改良してください。
  - 「変数」の「変数を追加する」をクリックします。「作成する変数の名前」に「はかる時間」と入力して「OK」をクリックしてください。これで「はかる時間」という名前の変数ができました。
  - 「基本」に「最初だけ」ブロックがあります。また「変数」に「変数 はかる時間 を 0 にする」ブロックがあります。これらを使って、「最初」に変数「はかる時間」の値を「10」にします。
  - くりかえしの回数は、変数「はかる時間」で指定します。
- **プログラムができたなら、ただしくごくかシミュレータでたしかめてください。**

## 改良プログラム3 ～作成例



- この改良では、プログラムのかきかたがかわるだけで、プログラムのうごきはなにもかわりません。

## 改良プログラム 4

- Bボタンをおすたびに、変数「はかる時間」が10ずつふえるように、またAボタンをおすと、変数「はかる時間」が10にもどるように改良してください。
- 変数の値をかえたときには、その値が表示されるようにしてください。
  - 「入力」に「ボタン A が押されたとき」ブロックがあります。「A」のところをクリックすると「A」「B」「A+B」をえらぶことができます。
  - 「変数」に「変数 はかる時間 を 1 だけ増やす」「変数 はかる時間 を 0 にする」ブロックがあります。
  - 「基本」に「数を表示」ブロックがあります。
- **プログラムができたなら、マイクロビットにかきこんでください。**

# 改良プログラム4 ～作成例

The image displays a Scratch script with the following blocks:

- 最初だけ** (Initially):
  - 変数 はかる時間 を 10 にする (Set variable 'はかる時間' to 10)
- ボタン B が押されたとき** (When button B is clicked):
  - 変数 はかる時間 を 10 だけ増やす (Increase variable 'はかる時間' by 10)
  - 数を表示 はかる時間 (Show number 'はかる時間')
- ボタン A が押されたとき** (When button A is clicked):
  - 変数 はかる時間 を 10 にする (Set variable 'はかる時間' to 10)
  - 数を表示 はかる時間 (Show number 'はかる時間')
- ボタン A+B が押されたとき** (When button A+B is clicked):
  - くりかえし はかる時間 回 (Repeat 'はかる時間' times)
  - アイコンを表示 (Show icon)
  - 一時停止 (ミリ秒) 500 (Pause 500 milliseconds)
  - アイコンを表示 (Show icon)
  - 一時停止 (ミリ秒) 500 (Pause 500 milliseconds)
  - アイコンを表示 (Show icon)
  - 鳴らす メロディ ピコーン! バックグラウンドで一度だけ (Play melody 'ピコーン!' once in the background)

# まとめ

- マイクロビットのつかいかたを確認しました。
- マイクロビットで「キッチンタイマー」をつくりました。
- プログラミングの基礎についてまなびました。
  - じゅんじ**順次**：かいた順番に作業すること。
  - ぶんき**分岐**：条件にあてはまるかどうかで作業の内容をかえること（もし～なら）。
  - はんぷく**反復**：ある作業をなんどもくりかえすこと。
  - **アルゴリズム**：「順次」「分岐」「反復」をくみあわせてつくった作業の手順。
  - へんすう**変数**：数や文字などを入れておく「ハコ」のようなもの。