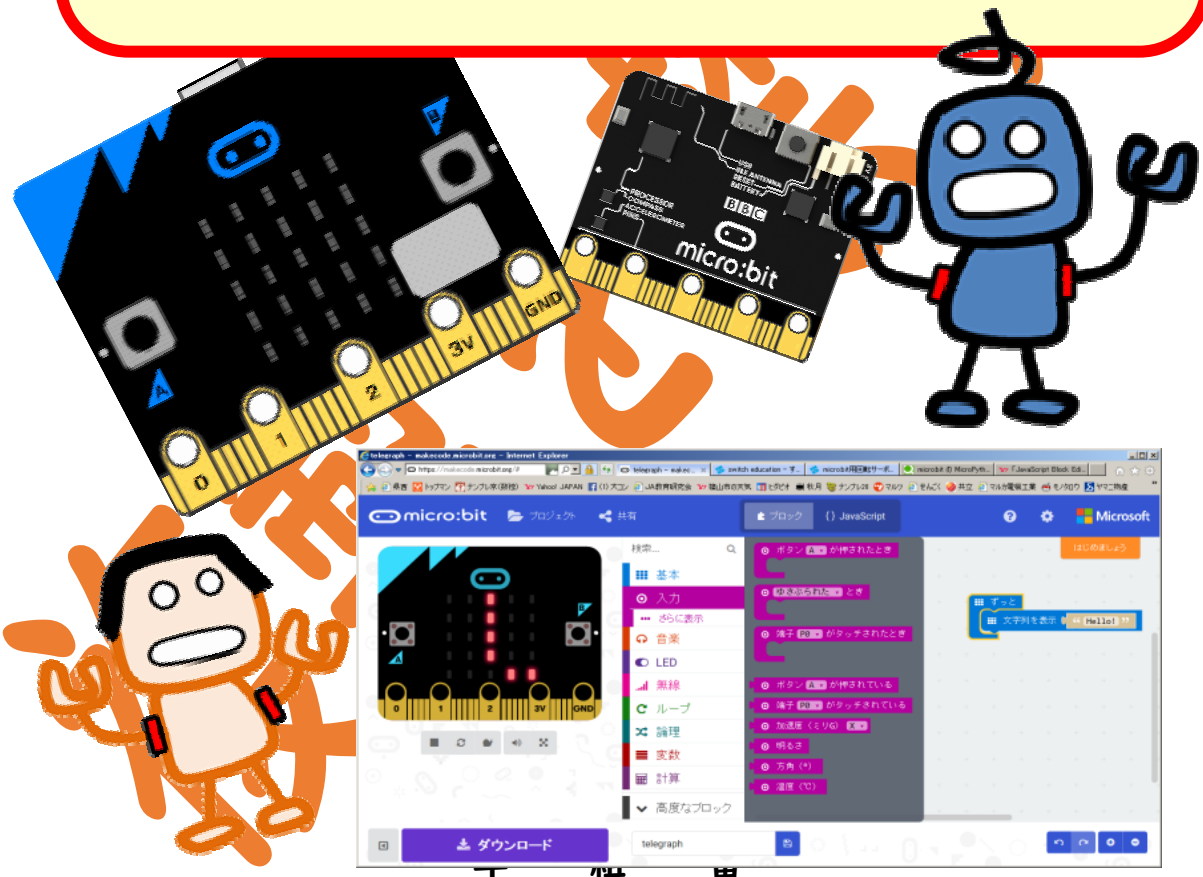


【教師用指導書】

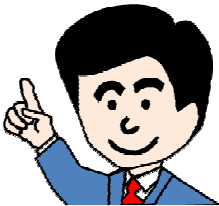
～プログラミングにチャレンジ～

ポンコツロボットの
まいくろびと あそ
micro:bitで遊ぼう



なまえ
名前【 】

チャレンジ は時間があれば挑戦してください！



はじめに

この学習ノートは、小学校でプログラミングを学ぶ児童の役に立つことを願って制作いたしました。内容等の改善や指導法の検討など、これからすべきところが多々あるとは思っておりますので、ご指摘頂きながら充実を図りたいと思っておりますので、どうぞよろしくお願いいたします。

目次

はじめに	2
学習ノート開発の意図	4
プログラミングの基礎知識	4
プログラム	4
プログラミング	4
基本的な3つの処理	4
ビジュアルプログラミング	5
プログラムの構造化	5
MICRO:BIT	6
micro:bit とは	6
主な名称	6
micro:bit のプログラミング環境	7
各ページの使い方	8
ポンコツロボット登場	9
小さなコンピューター	9
micro:bit の電源	9
ここで使用するプログラム	9
プログラムとプログラミング	11
プログラムなければただの箱	11
(参考)micro:bit の OS	11
micro:bit の色	11
micro:bit とコンピューター	13
ブロックエディター	13
プログラミングの基本的な手順	15
ソフトを使う	15
(参考)アプリのアイコンと起動画面	15
ブロックの貼りつけと削除	15

シミュレーター	17
プログラムを micro:bit に書き込む	19
(参考)機械語と micro:bit	19
『最初だけ』と『ずっと』ブロックの違い	19
LED 画面の表示ブロック	21
LED 画面の表示ブロック	21
ハートがドキドキ	23
もっとドキドキ	23
一時停止の時間	25
一時停止のプログラム	25
ロボットダンス	27
もっとプログラミング	29
ボタンを使って	29
割り込み処理	29
くりかえしてみよう	31
じゃんけんをしよう	33
ばらばらに出てくる数字	33
プログラムの中のかれ道	35
条件分岐処理のブロック	35
ゆさぶってみよう	37
変数という箱	39
もっとプログラミング	45
さいころプログラム	45
おみくじプログラム	47
無線でやりとり	49
数値の送受信	49
日本語は送れるか?	49
数値を送って、文字を表示	51
いろんな警報器をつくろう	53
音を鳴らしてみよう	53
音階と周波数	55
救急車のピーポー音	55
(参考)ピヨピヨ音	55
警報器を作ってみよう	57
地磁気センサーを使う	59
拡張機能 (出力装置)	59
拡張機能 (入力装置)	60
《参考》 関連製品価格表	62
《参考》 姉妹冊子	63

学習ノート開発の意図

●このノートは、主に小学校の中・高学年を対象に開発したものです。プログラミングの学習の中で、コンピューターの画面上だけでなく、実際のもの（micro:bit 等）を動かしながら、プログラミング教育ができることを図っております。

●micro:bit は小さなコンピューターです。micro:bit を動かしながらプログラミングに必要な基礎的な知識を楽しく学習し、また考えたアイデアを実現できます。また扱いやすく拡張性の高い micro:bit を使うことで、さらに IoT についても学習することが可能です。

プログラミングの基礎知識

プログラム

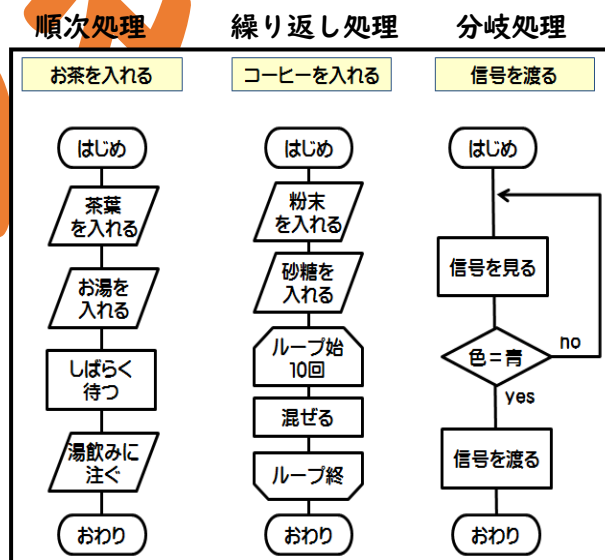
コンピューターを動かすための手順を、「命令」の組み合わせで表した「手順書」のことです。

プログラミング

さまざまな「命令」を組み合わせたプログラムを作成し、コンピューターに実行させることです。コンピューターに命令する順序を計画するといってもいいでしょう。

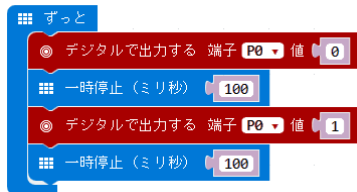
基本的な3つの処理

プログラムは、どんなに複雑なものであっても、3つの基本的な処理の組み合わせで構成されています。本書では、全てを学習できます。



ビジュアルプログラミング

パズルを組み立てるように、ブロックとして用意されている「命令」を組み合わせることで。一般的なプログラミングでは文字によって命令を記述していきますが、ビジュアルプログラミングではマウス操作だけでほぼプログラミングができるため、子どものプログラミング教育の現場でもよく使われています。



ブロック形式 (ビジュアルプログラミング)

```
basic.forever(() => {  
  pins.digitalWritePin(DigitalPin.P0, 0)  
  basic.pause(100)  
  pins.digitalWritePin(DigitalPin.P0, 1)  
  basic.pause(100)  
})
```

文字形式

プログラムの構造化

プログラムの構造化を促す内容となっています。これにより、プログラムの深化や改造にスムーズに取り組めると考えています。



構造化したプログラム



羅列したプログラム


micro:bit

micro:bit とは

●英国放送協会（BBC）が主体となって開発した STEM 教育用マイコンボードです。イギリスでは school year 7（11～12 歳）の生徒に無償で配布されています。

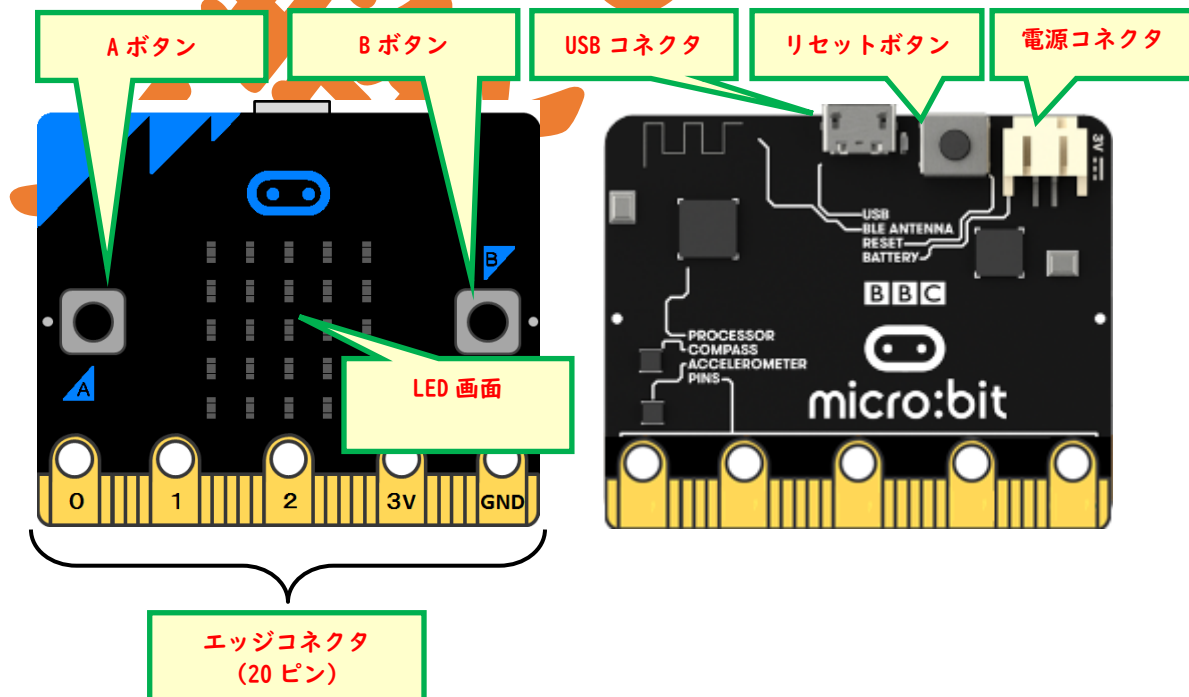
日本では 2017 年の 8 月から販売され、標準価格 2000 円(税抜)でています。

●Micro:bit 教育財団 … イギリスの NPO 団体。世界中の子どもたちがテクノロジーを使った創造性を育み、学校教育、家庭などにおいてデジタルスキルを習得することを目的としています。

●日本市場向けの micro:bit … Micro:bit は Bluetooth Low Energy (BLE) による通信機能が搭載されているため、通電時は電波が発射されています。日本では電波法で規制されるため、「micro:bit Telec version」とされたもの（本体では区別できない）で、同時に工事設計認証（技適）を受けた証書（白黒印刷物または箱に技適マーク  認証番号「018-170172」）を持った人のみで使用可能となります。

●搭載機器 … 25 個（5×5）の赤色 LED、2 つのボタンスイッチ、センサー（地磁気、加速度）、無線通信機能（BLE: Bluetooth Low Energy）などが搭載されています。

主な名称



micro:bit のプログラミング環境

- JavaScript (ジャバスクリプト) ブロックエディター … インターネット上でブロックを組み合わせるだけでプログラミングが簡単にできます。また JavaScript と切り替えることもでき、文字ベースのプログラミングもできます。
- micro:bit モバイルアプリ … JavaScript ブロックエディターのスマートフォン、タブレット版。アプリを読み込んで使用します。
- Python (パイソン) エディター … 文字ベースのプログラミングができます。
- MbedCompiler (エンベッドコンパイラ) … 文字ベースのプログラミングができます。(要登録)
- Scratch (スクラッチ) … ブロックベースの Scratch では、micro:bit の操作で、中のスプライトを動かすプログラムをつくることもできます。スクラッチは micro:bit 用オフライン版が提供されています。
- Arduino-IDE (アルドゥイーノ アイディーイー) … Arduino 開発のためのフリーの、文字ベースのプログラミングソフトウェアで、Arduino ソフトウェアとも呼びます。オフラインでプログラミングできます。

各ページの使い方

プログラミングをやってみよう

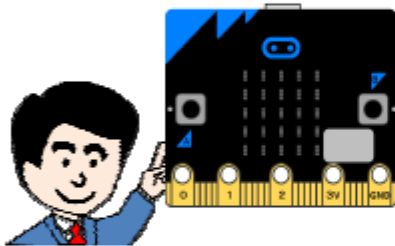
1. ポンコツロボット登場？

ぼくはポンコツロボット。名前はまだない。
友だちのノブ君は「ポンロボ君」と呼んでくれる
のだけど、ほんとうの名前は決まっていな



今日、ノブ君と学校へ行くと先生がちっちゃな板
みたいなものを持ってきていた。

「何だろう？」

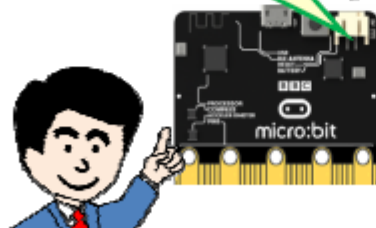


「これはmicro:bitという
小さなコンピューターだよ」

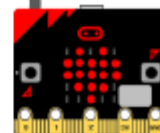
「コンピューター？ これが？」

電源コネクタ

「裏側に電源コネクタがあるか
ら電池をつないでみよう！」



「わーっ
ハートが出てきたよ！」



ポンコツロボット登場



●ポンコツロボット…名前はないがポンロボ君と呼ばれることがあります。学習する児童の立場にいます。



●ノブ君…ポンコツロボットの友達。コンピューターについて詳しい児童の立場です。



●先生

小さなコンピューター

micro:bitのようにマイコン（正しくはマイクロ コントローラ、マイクロ コンピューターということもあります）とその周辺の回路を載せた基板をマイコンボードといいます。これ自体が小さなコンピューターです。

micro:bit の電源

●外部電源としてパソコンの USB、または乾電池 2 本（ $1.5\text{ V} \times 2\text{ 本} = 3\text{ V}$ ）を使用します。

- ・ USB コネクタを介して
- ・ PH コネクタを介して
- ・ エッジコネクタ（端子）を介して
- ・ 裏面端子を介して

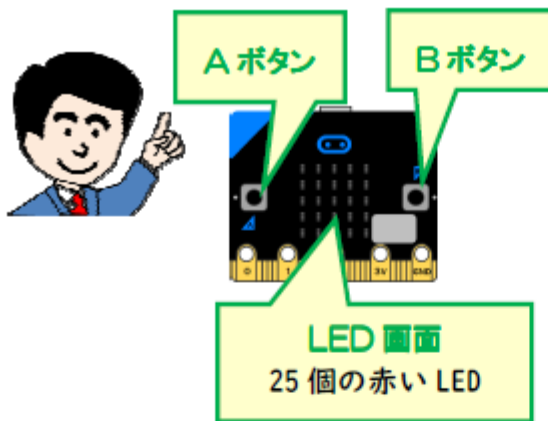
の 4 方式が可能です。



ここで使用するプログラム

ここで使用する micro:bit には、ハートを表示する下のプログラムがあらかじめ書かれたものを準備します。（提示用の micro:bit1 を 1 台だけ準備したら良いでしょう）

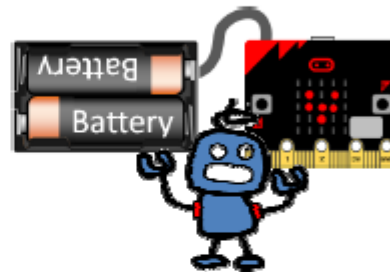




「micro:bit にはボタンが
2 つついているでしょう
A ボタンと B ボタンだよ」

「今度は
A ボタンを押してごらん」

「あれっ？
ハートが小さくなったよ！」



もんだい 何でこうなるかを、私に代わって、みんなに
教えてあげてください。

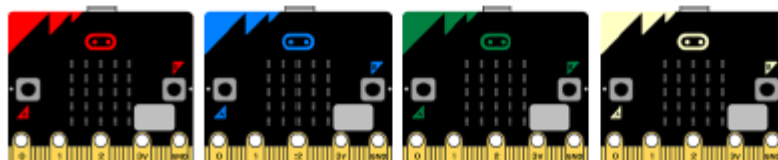


(何でハートがでたり、小さくなったりするのですか?)

micro:bit にプログラムと呼ばれるものが書かれていて、それでハートが
出たり小さくなったりしています。

【豆知識】

micro:bit は 4 色 ありますが、どれも同じです！



プログラムとプログラミング

コンピューターを動かすため、「命令」の組み合わせで表した「手順書」のことをプログラムと
いいます。

プログラムを作成することをプログラミングといえます。

プログラムを作成する人をプログラマーといえます。初めてプログラムを作成し、実行できた
とき、「君たちはプログラマーになったのだよ」と声掛けしてみてください。

プログラムなければただの箱

コンピューターには基礎となるプログラム（ソフトウェア）である OS（オペレーティングシス
テム）があらかじめ書き込まれています。さらにプログラム（応用ソフトウェア）を書き込むこ
とで、期待する動作をします。

本書でプログラムとは応用ソフトウェアのことを指し、OS のこととは区別します。

(参考)micro:bit の OS

micro:bit の OS は「ファームウェア(DAPLink)」といわれます。

最新バージョンは

<https://microbit.org/get-started/user-guide/firmware/>

からリリースされています。(2020/3/31 現在 V253) |

また、以下からもリリースされています。

<https://github.com/ARMmbed/DAPLink/releases/> (2020/3/31 現在 V254)

これを、メンテナンスモードにした micro:bit に書き込むと最新のものになります。

もんだい

コンピューターはプログラムが書かれていて、初めて動くこと理解させましょう。

micro:bit の色

- ・4色のモデルが販売されていますが、機能は全く同じです。
- ・ブロックエディターのシミュレーターにも4色がランダムに表示されますが、これも気にする
ことはありません。

P.2

P.2

2. ♥をだそう！

(1) ブロックエディター



「micro:bitに【 **プログラム** 】をつくるには、コンピューターと専用のソフトウェアが必要です。今日は、**ジャバスクリプトブロックエディター**という名前のソフトウェアを使います」



「先生！長すぎて覚えられないよ～」

「えーっ！ソフト…？
ジャバ…？」



「そうだな～。ソフトの**ブロックエディター**を使います。これなら覚えられるかな？まずは、名前より使い方を覚えてくださいね」



【 **ブロックメニュー** 】
用途別のメニュー
いろいろなブロックを表示

【 **プログラミングエリア** 】
プログラムを作るところ

micro:bit とコンピューター

micro:bit の専用のソフトウェアを制作できるコンピューターとは

パソコン (PC)

iPad などのタブレット

携帯電話 (スマートフォン)

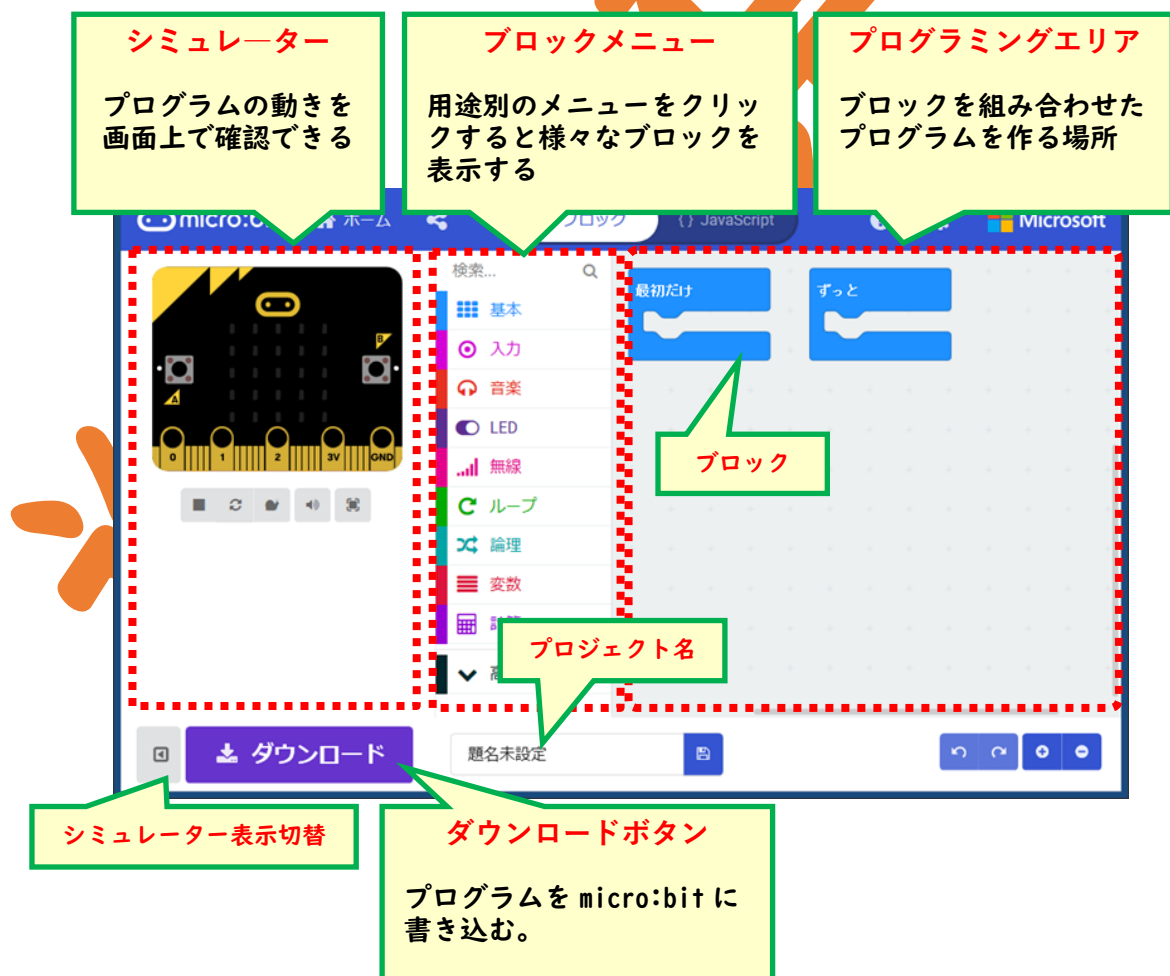
などが考えられます。

ブロックエディター

本書で使用するソフトウェアは Microsoft 社の JavaScript (ジャバスクリプト) ブロックエディターである「MakeCode エディター」の「ブロックベース(モード)」です。

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

本書では「製品名」ではなく、また「テキストベース」では使用しませんので、「ブロックエディター」という名前で進めていきます。



(2) ソフトを使う

「とにかく、やってみながらブロックエディターの使い方を覚えましょう。はじめにコンピューターにブロックエディターを表示しよう



まとめよう 使うコンピューターは何ですか？表示の方法は？



(例) iPadで右のアイコンを選んで「Create Code」でスタートする



「ソフトが出たらはじめに、ブロックを出したり消したりすることからやってみましょう」



(ブロックを貼りつける)



(ブロックを削除する)



「ブロックを出すときはブロックメニューからプログラミングエリアに動かします。消すときは、その逆です」

【豆知識】

ブロックを出すことを『貼りつける』
消すことを『削除する』ともいいます。



プログラミングの基本的な手順

- ブロックエディターを起動する
- プログラムする。…ブロックを貼り付けたり、削除したりする。
(●シミュレーターで確認する)
- micro:bitにダウンロード(書き込み)する。
- 動作を確認する。

ソフトを使う

- コンピュータ(PC、モバイル)の種類や機材に応じて、表示や使い方が微妙に異なりますので、ここでは詳細な使い方は省略しています。
- ※各校の状況に応じて、プリントや掲示物で補ってください。
- ブロックの貼り付け、削除の方法をここでしっかりなれさせてください。

(参考)アプリのアイコンと起動画面

「MakeCode エディター」のアイコンと基本画面



(PCの場合)

(モバイルコンピューターの場合)

ブロックの貼りつけと削除

- 貼りつけ…プログラミングエリアにドラッグ&ドロップします。



プログラミングエリア

- 削除…ブロックメニューにドラッグ&ドロップします。



ブロックメニュー



「先生！ブロックを出したり、
消したりができるようになったよ
これでプログラムが書けるのかなあ？」

「じゃあ、『ずっと』ブロックの中に
ブロックを入れてください」



「できました！」



「これでプログラムは完成です！
何かに、気づきませんでしたか？」



「えっ、ほんとうに？
あっ、micro:bitの絵に♥がでているよ。」



でも、ボクの micro:bit には何もでてないなあ。
これでプログラムができたのかな？」

【豆知識】

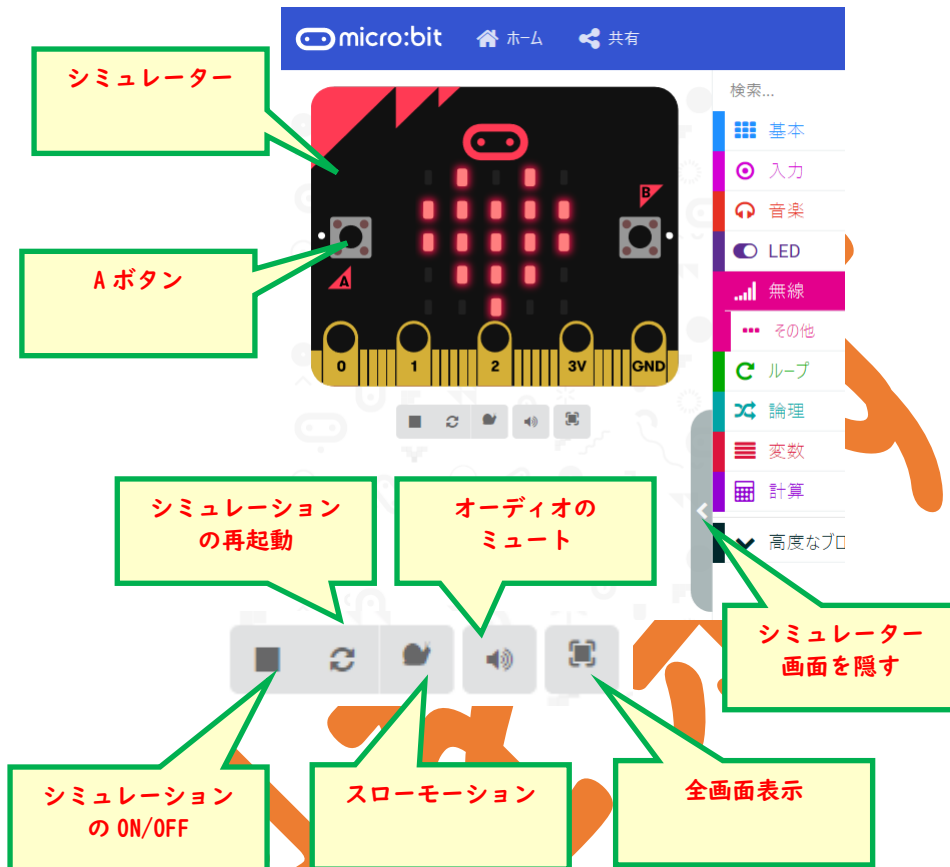
画面の micro:bit の絵は『シミュレーター』といい、
プログラムの動きをコンピューターだけで確かめる
ことができます。



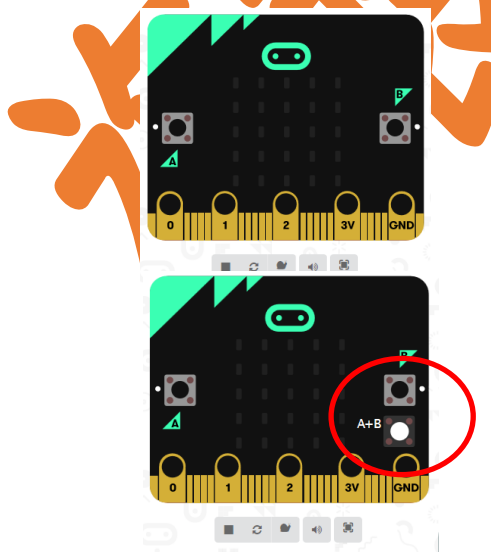
シミュレーター

●模擬ソフトウェアのこと。

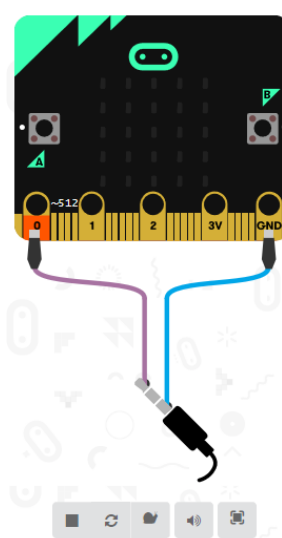
●micro:bitのプログラムの結果はシミュレーターのみでも確認できますが、実際のmicro:bitで確認することが重要です。



●プログラムの内容により、表示が少しずつ異なります。



(A+B ボタンの入力があるとき)



(音を鳴らすとき)

(3) プログラムを micro:bit に書き込む

「micro:bitがプログラムどおりに動くためには、

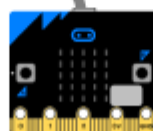
- micro:bit をコンピューターにつないでから
- プログラムを micro:bit に【 **ダウンロード** 】する

作業が必要です。



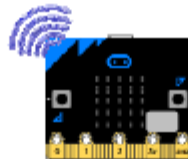
【豆知識】 micro:bit とコンピューターのつなぎかた

- パソコンと



USB ケーブル

- タブレットと



つなぎ方は先生に教えていただく！



「micro:bit とコンピューターが繋がれたら

ダウンロード

ボタンを押してみよう

micro:bit にハートがでたら

プログラムの書き込み成功！」



「できたよ！できた！
ハートが、現れたよ！
やったー！やったー！」

プログラムを micro:bit に書き込む

※PCの種類や環境によって、手順が若干異なります。各学校の状況に応じて、プリントや掲示物で補ってください。

- ① micro:bitを接続する ※micro:bitドライブができる。
- ② プログラムをダウンロード（micro:bitに直接保存）

(1) ダウンロードボタンをクリックします。

※ これだけで完了する場合があります。

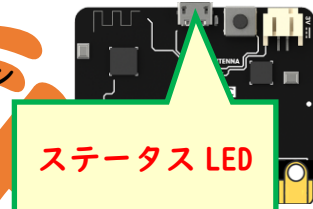
(2) 名前を付けて保存をクリックします。

(3) 保存先を「MICROBIT」ドライブにして、保存ボタンをクリックします。

(micro:bitのステータスLEDが点滅します)

- ③ プログラムを保存する。

※ プログラム名をつけておく



(参考)機械語と micro:bit

- コンピュータが直接解釈し実行できるプログラミング言語「マシン語」ともいいます。0と1（2進数）の数字で書かれ、人間に理解しにくいいため低級言語といわれます。
- JavaScriptブロックエディターで作られた機械語は拡張子「hex」で表示されるファイルになり、これを micro:bit に書き込むと動作します。

『最初だけ』と『ずっと』ブロックの違い

『最初だけ』ブロックは、プログラムに必要な初期設定などを入れます。プログラムの一番最初にだけ実行されます。

『ずっと』ブロックは『最初だけ』ブロックに続いて実行され、その後繰り返し実行されます。電源が切られるまで、プログラムの終了はありません。



まとめ プログラムを作ってから書き込むまで、難しかったこと、覚えておかななくてはならないことなどをまとめておきましょう。



チャレンジ 他の模様も使ってみよう。



さてみんなの micro:bit は
どうなったかな？

いろいろなもようができて
楽しかった。



まとめ

内容は自由に記述させましょう。

チャレンジ

時間があれば取り組む課題です。

※遅い児童は、あとで時間があるときにするように指導します。

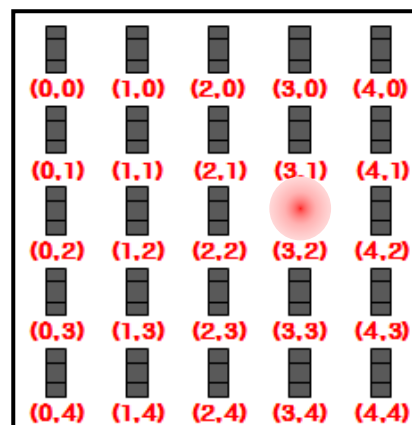
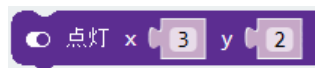
●その他のLED画面の表示も確かめさせましょう。

LED画面の表示ブロック

LED画面の表示用のブロックは、下記のものがあります。



LED画面の表示ブロック



3. ロボットダンスを作ろう

(1) ハートがドキドキ



「右のプログラムを作って書き込んでごらん」



「プログラムができたら、『ダウンロード』ボタンを押してっど…」

「うわーっ！ボクの心臓しんぞうみたいになったよ」



さてみんなの micro:bit はどうなったかな？

ハートが大きくなったり、小さくなったりした。

(2) もっとドキドキ



「『基本』メニューを選えらび、下に『その他』メニューが出てくるよ。その中の『表示を消す』プロ



ックつかを使ってみよう」

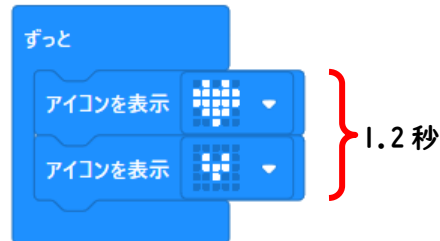
さてみんなの micro:bit はどうなったかな？

ハートが点めつした。



ハートがドキドキ

- micro:bit の LED 表示プログラムの実行には時間が約 0.6 秒程度かかります。
- 表示時間がかかるため。下のプログラムでは高速での切り替えはできません。2 回で約 1.2 秒かかります。



もっとドキドキ

- 『表示を消す』の実行時間はほとんどかかりません。
- 表示する時間に比べ、消える時間は非常に短くなります。



(約 0.6 秒)

(約 1.2 秒)

(約 1.2 秒)

- ハートを高速で点滅するだけなら、LED画面の表示の ON/OFF で切り替えれば可能です。



(3) 一時停止のプログラム



『基本』メニューの中に、『一時停止』ブロックがあります。これを使ってみよう」



【豆知識】

100ms = 100ミリ秒 = 0.1秒

1second = 1秒 = 1000ミリ秒

※複数の時はsを最後につけます。2seconds



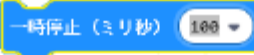
さてみんなの micro:bit はどうなったかな？

ハートが点めつした。前のプログラムと変わらない。

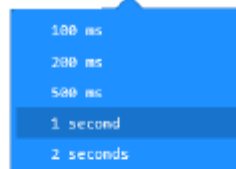


やってみよう

一時停止の時間を長く変えてみよう。



さてみんなの micro:bit はどうなったかな？



1秒や2秒のときはハートがゆっくり点めつした。100m秒や200m秒では変わらないように思った。



【豆知識】

25個のLED（発光ダイオード）でハートは表示されています。表示するのに時間がかかるので、速く点めつさせることはできません。



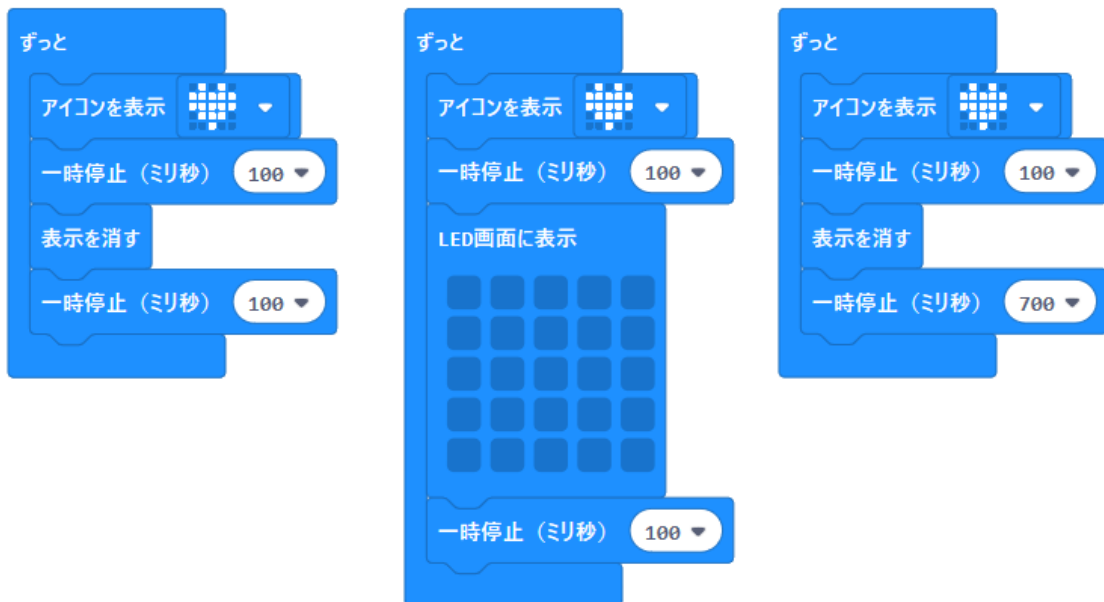
一時停止の時間

一時停止の時間は、プルダウンメニューでは5種類だけですが、直接時間を打ち込むこともできます。



一時停止のプログラム

- micro:bit の LED 表示プログラムの実行には時間がかかります。
- 1 回の点滅にかかる大体の実行時間を測定してみました。



(約 0.8 秒)

(約 1.4 秒)

(約 1.4 秒)

やってみよう

- 感覚的に処理時間の違いをつかませるだけでよいでしょう。

追加チャレンジ

※余裕のある児童用の課題

- 「アイコンを表示」ブロックの表示時間を調べてみよう」
(方法)ストップウォッチで、20 回とか 50 回の点滅にかかる時間を計測させて、1 回にかかる時間を計算で求めましょう。

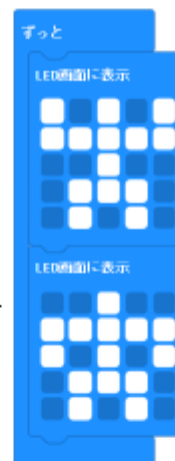
チャレンジ ハートがだんだんちい小さくなるプログラム
をかんがえよう。



(プログラム)
「大きなハート」「小さなハート」「中央に1個点灯」「点灯なし」の順に
プログラムした。

(4) ロボットダンス

チャレンジ 右のプログラムを参考さんこうにして、ロボットが
ダンスしているように見えるプログラム
をかんがえよう。

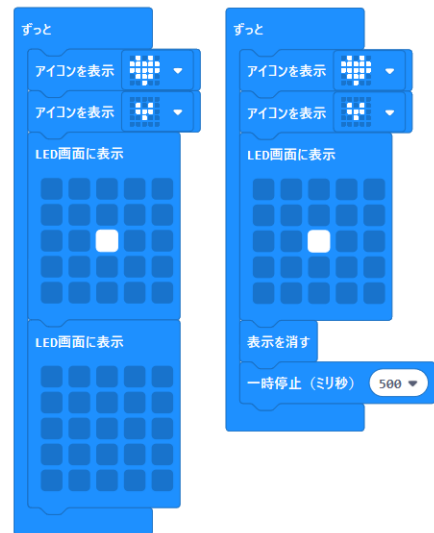


(プログラム)
全部で5つのロボットを作り、順番を考えて
プログラムしたら、面白いダンスができた。



チャレンジ 時間があれば取り組む課題です。

※遅い児童は、あとで時間があるときにするように指導します。

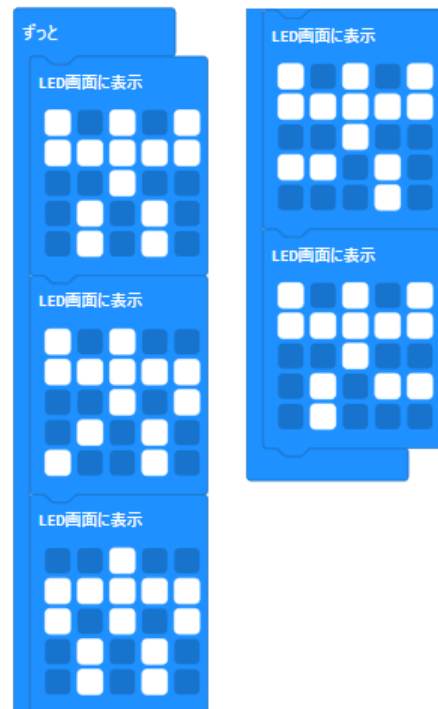


ロボットダンス

チャレンジ 時間があれば取り組む課題です。

※遅い児童は、あとで時間があるときにするように指導します。

※プログラムの記述は、略図で描かせるか、画面をハードコピー（キャプチャー）して印刷したものを張らしてもよいでしょう。



追加チャレンジ ※余裕のある児童用の課題

- 自ら課題を設定して取り組ませましょう。
- （例）次々と模様が変わるプログラムを考えてみよう。

もっとプログラミング

1. 今の気持ちを表そう

(1) ボタンを使って



『入力』メニューの中に『ボタンが A が押されたとき』ブロックがあります。



▼を押すと、『B』と『A+B』を選ぶことができます。それでは、これを使ってみよう



やってみよう 次のプログラムをためしてみよう。



最初だけ



さてみんなの micro:bit はどうなったかな？

A ボタンを押して離したときハートが小さくなった。
B ボタンを押すと今度は大きくなった。
ボタンを長押ししたら、変わらなかった。



【豆知識】

micro:bit のボタン A とボタン B で別の動作をさせるようなことを『割り込み』といいます。



もっとプログラミング

これまでは簡単なプログラムの制作を通してプログラミングの手順を学習しました。ここからは、もう少し利用できるものを作っていくプログラミングの応用編です。

ボタンを使って

●ボタンを押した時に割り込むプログラム

- ・ボタンを押して、すぐに放した時に実行します。
- ・ボタンを長押しして、放しても実行しません。
- ・ボタンを連続して押すと、押した分だけ蓄積し、くり返し実行します。

- ボタンを押して A ボタンを押したとき、アイコンが表示されます。

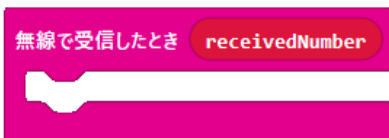
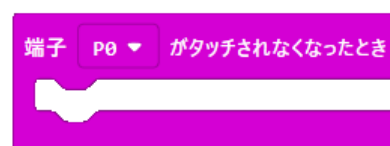
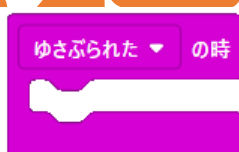
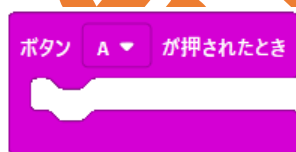


『きなハート』が

割り込み処理

micro:bit の割り込み処理は下のようなものがあります。

- A ボタン、B ボタン、A+B ボタンでの割り込み処理
- P0~2 と GND に同時に触れたタッチセンサでの割り込み処理
- 加速度センサーの検知による割り込み処理



- 『ゆさぶられた時』は学習ノートの p.13 の「じゃんけんプログラム」で使用します。

(2) くりかえしてみよう



AボタンとBボタンを同時に押したときのプログラムを作ってみた。左と右の違いを確認しよう。また、くりかえしの数を増やしてみよう。

さてみんなの micro:bit はどうなったかな？

左は1回だけ小さいハートが出たけど、右は4回も出た。



【豆知識】

同じことをくりかえすことを『くりかえし』または『ループ』といいます。

『ずっと』は『無限くりかえし』

『くりかえし ~回』は『有限くりかえし』

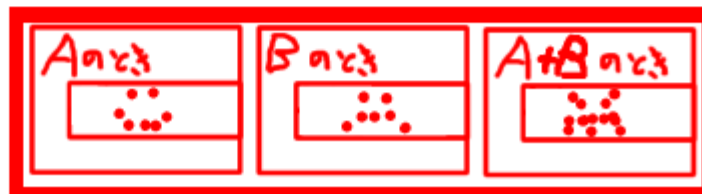
と区別できます。



やってみよう ボタンによって、感情（うれしい、悲しいなど）を表示できるプログラムを考えよう。



(プログラム)



くりかえしてみよう

●くりかえしのブロックの種類



(無限くりかえし)

(有限くりかえし)

やってみよう

※プログラムの記述は、略図で描かせるか、画面をハードコピー（キャプチャー）して印刷したものを張らしてもよいでしょう。



2.じゃんけんをしよう

(1) ばらばらに出てくる数字

右のプログラムをためしてみよう。



さてみんなの micro:bit はどうなったかな？



グー、チョキ。パーが順番にでた。



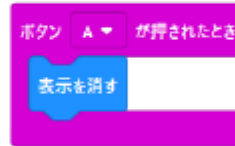
『計算』メニューの中の

0 から 10 までの乱数

ブロックを使って、右のプログラムをためしてみよう。



さてみんなの micro:bit はどうなったかな？



0、1、2の数が、順番ではなく、ばらばらの順で表示される。



【豆知識】

次に何が出てくるかわからない数を『乱数』といいます。



じゃんけんをしよう

●じゃんけんのプログラムの学習展開

グー・チョキ・パーの表し方 → 乱数の発生 → 条件分岐 → 変数の利用の順に学習していきます。

●グー・チョキ・パーの表し方はいろいろです。各自工夫させましょう。



ばらばらに出てくる数字

●この後、じゃんけんプログラムやサイコロで乱数を使います。

追加チャレンジ ※余裕のある児童用の課題

●乱数が均等な割合で発生するかを確かめるのも面白いでしょう。

●乱数の発生の様子を調べるプログラム例



1	2	3	4	5	6
正正	正下	正一	正正	正下	正下
正正	正正	正正	正正	正正	正正
正正	正正	正正	正正	正正	正正

(2) プログラムの中の分かれ道

『論理』メニューの中の



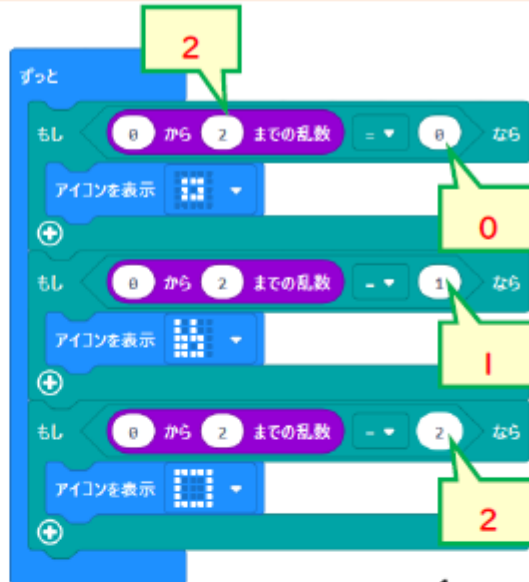
ブロックを使

って、右のプログラムをためし
てみよう。



ブロックも

『論理』メニューの中にあリま
す。



さてみんなの micro:bit はどうなったかな？

グー、チョキ。パーがばらばらにでた。



【豆知識】

条件によって行われることをかえることを、条件分岐処理とい
います。



プログラムの中の分かれ道

●このプログラムはP.18でのじゃんけんプログラムの完成を目指すための導入プログラムです。このプログラムを実行したLED画面では不都合は感じられませんが、問題を含んだプログラムです。どんな問題があることを、これから理解していきます。

●ここでは『もし』という条件分岐のブロックの使い方に慣れるようにします。

「0」…表示される
「1」…表示されない
「2」…表示されない

「0」…表示されない
「1」…表示される
「2」…表示されない

「0」…表示されない
「1」…表示されない
「2」…表示される

条件分岐処理のブロック

●「もし」ブロックの分岐数は、 をクリックすることで増減できます。

+で増える

-で減る

-で減る

(3) ゆさぶってみよう



右のプログラム
をためしてみよ
う。

```

ゆさぶられた ↓ の時
もし < 0 から 2 までの乱数 - - 0 > なら
  アイコンを表示 [ 0x00000000 ]
+
もし < 0 から 2 までの乱数 - - 1 > なら
  アイコンを表示 [ 0x00000001 ]
+
もし < 0 から 2 までの乱数 - - 2 > なら
  アイコンを表示 [ 0x00000002 ]
+
  
```

さてみんなの micro:bit はど
うなったかな？

micro:bit を振るとゲー、チョキ。パーが順番にでた。
時々、2、3回表示したりすることがある。



どうして2、3回表示することがあるのだろう？

1回振ると、乱数の発生を3回もするから。

【豆知識】

micro:bit には、**加速度センサ**
-がついているので、micro:bit
の動きや向きがわかります。



ゆさぶってみよう

- このプログラムはP.18でのじゃんけんプログラムの完成を目指すためのプログラムです。
- ここでは、加速度センサーを使った割り込み処理プログラムを経験します。
- このプログラムでは、1回振ると、3回乱数が発生します。3回のうち、条件が合えば1~3回表示されますが、全く表示されないときもある不完全なプログラムです。

ゆさぶられた の時

もし 0 から 2 までの乱数 = 0 なら

アイコンを表示

もし 0 から 2 までの乱数 = 1 なら

アイコンを表示

もし 0 から 2 までの乱数 = 2 なら

アイコンを表示

「0」…表示される
「1」…表示されない
「2」…表示されない

「0」…表示されない
「1」…表示される
「2」…表示されない

「0」…表示されない
「1」…表示されない
「2」…表示される

追加チャレンジ ※余裕のある児童用の課題

- 他の状態での割り込み処理を試させてみましょう。
(プログラム例)

ずっと

一時停止 (ミリ秒) 1000

表示を消す

ロケットが上になった の時

矢印を表示 上向き ↑

ロケットが下になった の時

矢印を表示 下向き ↓

左に傾けた の時

矢印を表示 左向き ←

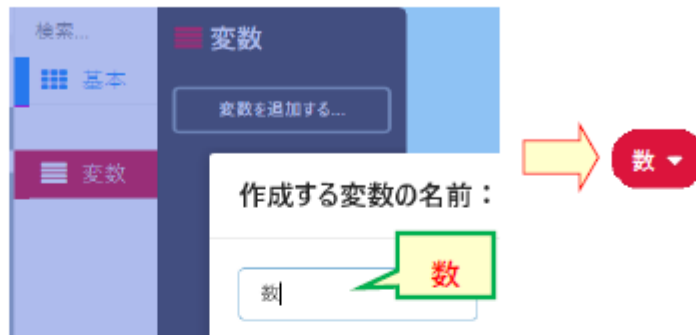
右に傾けた の時

矢印を表示 右向き →

(4) 変数という箱



『変数』メニューで『変数を追加する』ボタンで、『数』
という名前の変数を作ってみよう



次のプログラムをためしてみよう。



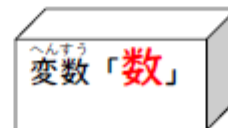
さてみんなの micro:bit はどうなったかな？

A ボタンを押すと「0」が、
B ボタンを押すと「1」が表示された。



【豆知識】

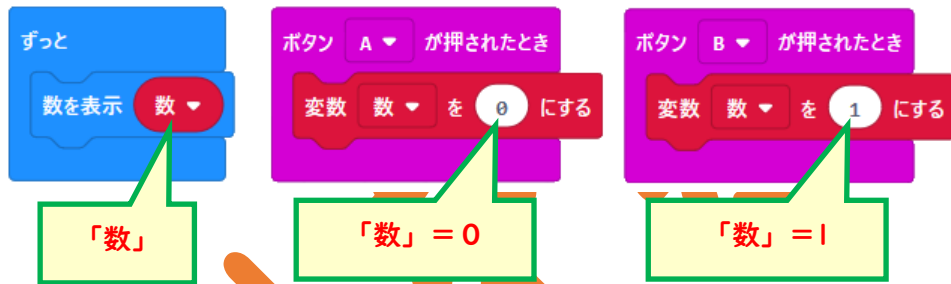
文字や数値をかりに入れておく箱
みたいなものを『変数』といいます。
変数の名前は、わかりやすい名前におきましょう。



変数という箱

- プログラム中で変数が使えると、プログラミング技術が大きく上達します。
 - 変数の追加で、変数の名前はわかりやすい名前にするのが良いでしょう。
- ここでは『数』にしていますが、『変数』とか『へんすう』や『X』などでも、何でも使えます。

● (プログラムの解説)





つぎのプログラムをためしてみよう。

```
ずっと
  数を表示 数
```

```
ボタン A が押されたとき
  変数 数 を 0 にする
```

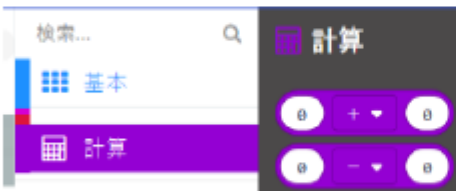
```
ボタン B が押されたとき
  変数 数 を 1 だけ増やす
```

さてみんなの micro:bit はどうなったかな？

B ボタンを押すと1ずつ数字が増えたが、A ボタンを押すと「0」になった。



チャレンジ 『計算』メニューのボタンを使って、つぎのプログラムをためしてみよう。



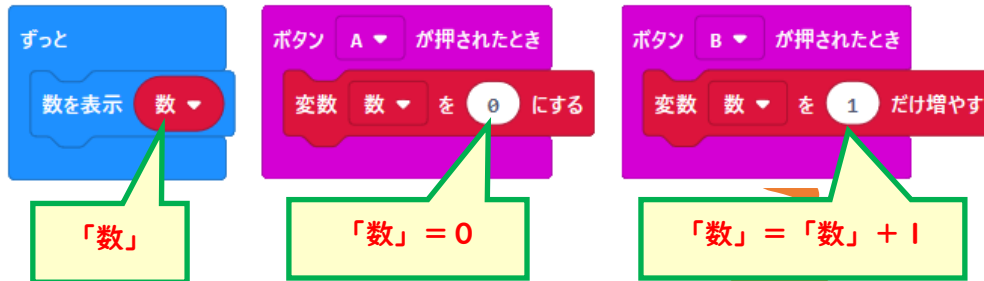
```
ボタン B が押されたとき
  変数 数 を 10 にする
  くりかえし 10 回
    一時停止 (ミリ秒) 1000
  変数 数 を 数 - 1 にする
  数を表示 数
```

(感想)

カウントダウンのプログラムができた。
B ボタンを押すと数字が9から0まで変わった。

●プログラムの解説

プログラムでは 「数」 = 「数」 + 1 という式がよく使われます。数学的には等号の左右は違っているのが誤った式ですが、「数」に1を加えたものを、新たな数にするという意味で使われます。

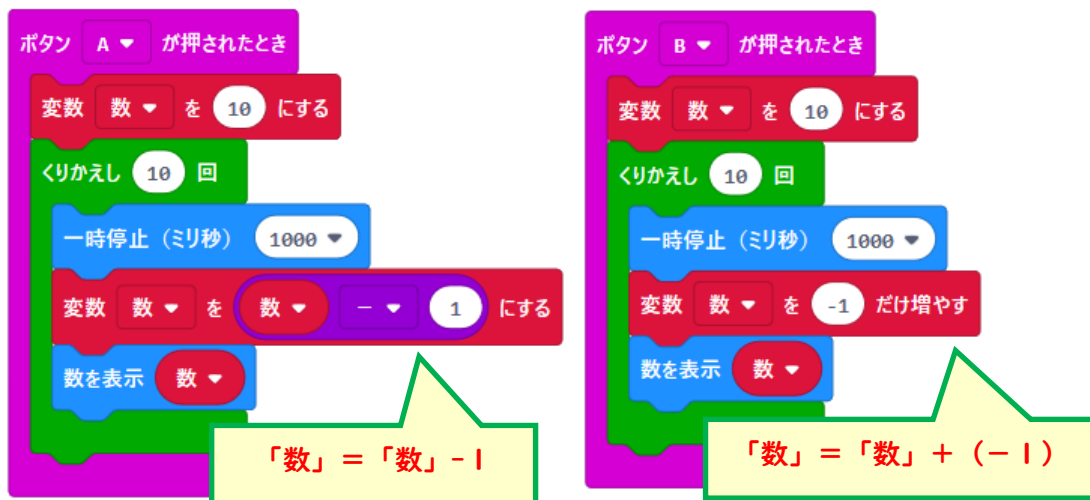


チャレンジ 時間があれば取り組む課題です。

※遅い児童は、あとで時間があるときにするように指導します。

●カウントダウンのプログラムです。

●下の二つのプログラムは同等の動きをします。



追加チャレンジ ※余裕のある児童用の課題

●ラーメンタイマー…3分経つと、何かの表示をする。



右のじゃんけんのプログラムをためしてみよう。

```

    とき
    変数 数 を 0 から 2 までの乱数 にする
    もし 数 が 0 なら
        アイコンを表示
    もし 数 が 1 なら
        アイコンを表示
    もし 数 が 2 なら
        アイコンを表示
    
```

さて、P16 のプログラムとくらべて、みんなの micro:bit はどうなったかな？

2, 3回表示するのがなくなった。



チャレンジ

上のプログラムと右のプログラムは同じ動きをするよ。

ほんとうかどうか、確かめよう。

(感想)

```

    とき
    変数 数 を 0 から 2 までの乱数 にする
    もし 数 が 0 なら
        アイコンを表示
    でなければ
    もし 数 が 1 なら
        アイコンを表示
    でなければ
        アイコンを表示
    
```

● (プログラムの解説)

ゆさぶられた の時

変数 数 を 0 から 2 までの乱数 にする

もし 数 = 0 なら

アイコンを表示

もし 数 = 1 なら

アイコンを表示

もし 数 = 2 なら

アイコンを表示

乱数「0」～「2」を発生

「0」…表示する

「1」…表示する

「2」…表示する

チャレンジ 時間

があれば取り組む課題です。

※遅い児童は、あとで時間があるときにするように指導します。

遅い児童

ゆさぶられた の時

変数 数 を 0 から 2 までの乱数 にする

もし 数 = 0 なら

アイコンを表示

でなければ

もし 数 = 1 なら

アイコンを表示

でなければ

アイコンを表示

「0」…表示する

「0」でない「1」…表示する

「0」でも「1」でもない…表示する

追加チャレンジ

※余裕のある児童用の課題

●(例)ゆさぶられたときは2秒表示し、それ以外は表示しない。

もし 数 = 2 なら

アイコンを表示

一時停止 (ミリ秒) 2000

表示を消す

もっともっとプログラミング

1. サイコロプログラム



サイコロのプログラムを作
ってみました。

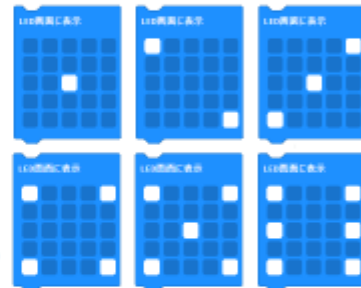
```

ゆきぶられた の時
  数を表示 1 から 6 までの乱数
  
```

振ったときに1~6の数字がバラバラな順でた。



やってみよう じゃんけ
んのプログラム (P.18) を
参考に、サイコロの目
が出るプログラムを作ろう。



(プログラムと感想)

```

ゆきぶられた の時
  変数 数 を 1 から 6 までの乱数 にする
  もし 数 - 1 なら
    LED画面に表示
  +
  もし 数 - 2 なら
    LED画面に表示
  
```



もっとプログラミング

ここからは、もう少し難しいものを作っていくプログラミングの発展編です。

さいころプログラム

●micro:bitのLED画面に、「1」～「6」の数字を出すプログラムから始め、目を出すプログラムに発展していきます。

●右は「1」～「6」の数字を表示するプログラム

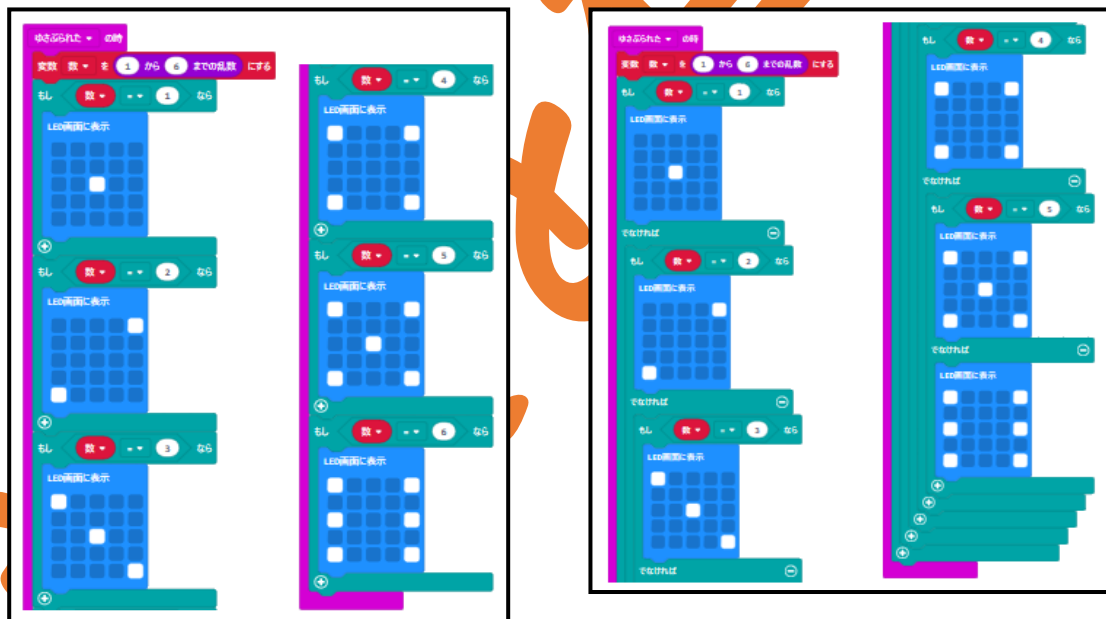


やってみよう

●さいころの目が出るプログラム

※プログラムの記述は、略図で一部を描かせるか、画面をハードコピー（キャプチャー）して印刷したものを張らしてもよいでしょう。

●下の二つのプログラムは同等の動きをします。

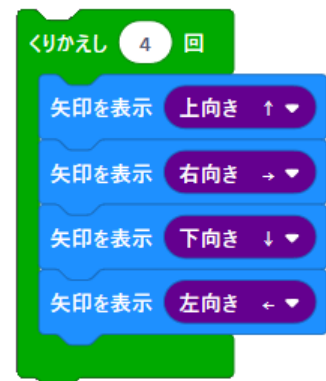


追加チャレンジ

※余裕のある児童用の課題

●さいころのプログラムの改造

(例)いきなり目が出ないように、右のようなプログラムをいれてみましょう。



2. おみくじプログラム



チャレンジ おみくじのプログラムを作ろう。

(プログラムのしくみ)

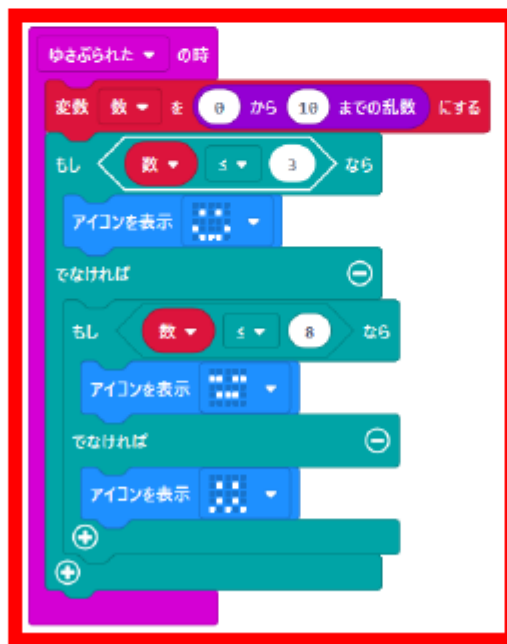
ゆさぶったときに

乱数 (0~10) を発生して

- ・ 0~3のときは「うれしい顔(大吉)」
 - ・ 4~8は「普通の顔(中吉)」
 - ・ 9~10の時は「悲しい顔(小吉)」
- を表示します。



(プログラムと感想)



おみくじプログラム

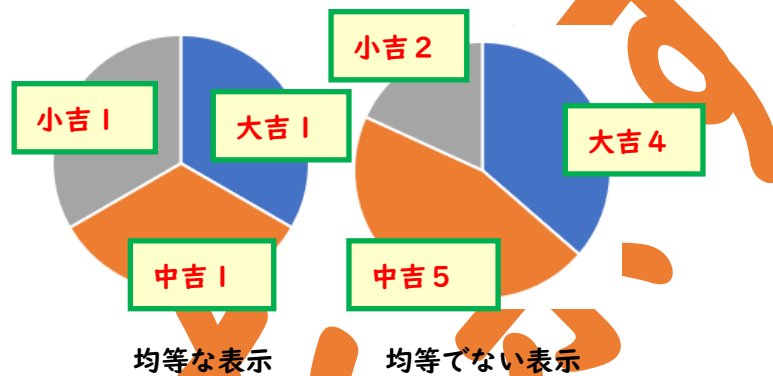
チャレンジ 時間があれば取り組む課題です。

※遅い児童は、あとで時間があるときにするように指導します。

※プログラムの記述は、略図で描かせるか、画面をハードコピー（キャプチャー）して印刷したものを張らしてもよいでしょう。

●乱数は、多くすれば均等な割合で発生しますが、ここでは意図して発生割合に差をつけたプログラムにします。

●指導上『凶』は外しています。



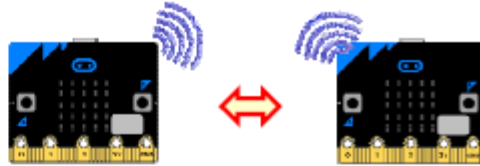
追加チャレンジ ※余裕のある児童用の課題

●「ゆさぶられた」以外の割り込み処理を試させてみましょう。

3. 無線でやり取り



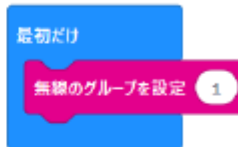
micro:bitは、無線を使って違う micro:bit に、文字や数値を送ることができます。



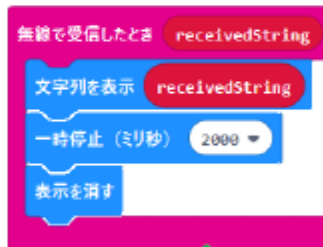
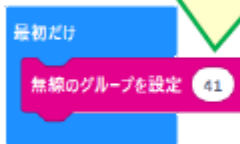
【豆知識】



無線のグループ（0～255）が同じなら、複数の micro:bit 間で通信することが可能です。



通信する人は同じ数にします



数字のときは…Number

文字のときは…String



やってみよう

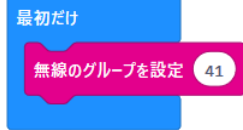
上のプログラムをためしてみよう。

ローマ字や数字を送ることができたけど、表示するスピードが遅かった。



無線でやりとり

- 無線のグループは 0-255 の数値から指定できます。
- 授業ではグループ内の出席番号の一番小さい数値を利用するのが簡便でしょう。



数値の送受信

- 数値は半角数字だけが可能です。
- 変数「receivedNumber」のコピーは、ドラッグ&ドロップで行います。



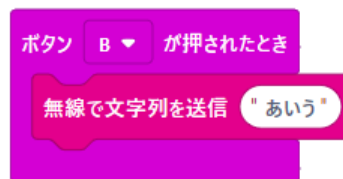
- 送り主を分かるようにするプログラム



この数値を変えると、だれが送ってきたか区別できます。

やってみよう

- 数値は半角数字を入力します。
- 文字列は日本語でも送受信できますが、表示することはできません。



日本語は送れるか？

※右のプログラムでは日本語文字は送受信できますが、micro:bitには「BS」と表示されます。





数字を送ると、文字が出るようにしたプログラムです。ためしてみよう。

```

最初だけ
無線のグループを設定 41

ボタン A が押されたとき
無線で数値を送信 0

ボタン B が押されたとき
無線で数値を送信 1

無線で受信したとき receivedNumber
もし receivedNumber が 0 なら
  文字列を表示 "Hello!"
もし receivedNumber が 1 なら
  文字列を表示 "Yahho!"
  
```

グループの人の micro:bit に A ボタンを押すと、「Hello!」が、B ボタンを押すと、「Yahho!」が表示された。



チャレンジ 次のプログラムを使って秘密の暗号送信機を完成してみよう。

```

最初だけ
無線のグループを設定 41
数を表示 数

ボタン A が押されたとき
無線で数値を送信 数

ボタン B が押されたとき
変数 数 を 1 だけ増やす
もし 数 が 5 なら
  変数 数 を 0 にする
数を表示 数
  
```

(感想)

数値を送って、文字を表示

- 送られてきた数値を判断して。それによって、文字が表示されます。

チャレンジ 時間があれば取り組む課題です。

※遅い児童は、あとで時間があるときにするように指導します。

●暗号送信機のプログラム

B ボタンで変数「数」の値を順に変え、A ボタンで送信する

●暗号受信機のプログラム

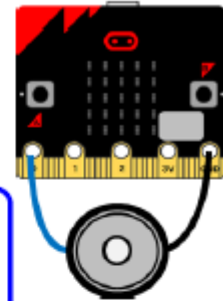
受信数値によって表示する文字を決める。



4. いろんな^{けいほうき}警報器をつくろう



micro:bit にスピーカーをつけると音^{おと}を鳴らすことができます。



^{まめらしき}【豆知識】 スピーカーはP0とGNDに^{せつぞく}接続します。
(P0と3Vでもかまいません)



(1) 音^{おと}を鳴らしてみよう。



^{みぎ}右のプログラムをためしてみよう。



ド、ド、ドと鳴った



^{みぎ}右のプログラムをためしてみよう。



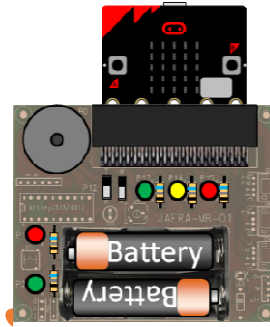
^{おと}音程の違うダダダーンが2回鳴った。



※一時停止^{いちじていし}がないとうまく鳴りません。

いろんな警報器をつくろう

ここからは、スピーカーを使った学習になります。
micro:bit にスピーカーを直接つなぐか、
スピーカーのつながったボードなどを利用してください。



mbAT (キングエース/JA 教育研究会より販売)

音を鳴らしてみよう

●音楽を鳴らすプログラム



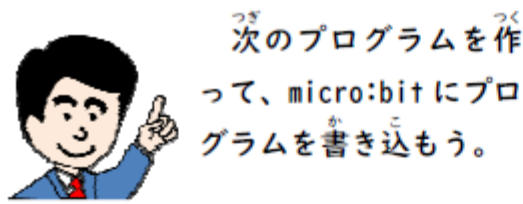
●音階は鍵盤を使って入力します。



●音の鳴らないプログラム

このプログラムを実行しても音はならない。





```

ずっと
音を鳴らす (Hz) 770 長さ (ミリ秒) 500
音を鳴らす (Hz) 060 長さ (ミリ秒) 500
    
```

(ヒント) 『入出力端子』の『その他』の中にある「音を鳴らす」ブロックを使います。

```

⊕ 入出力端子
... その他
音を鳴らす (Hz) 0 長さ (ミリ秒) 0
    
```

救急車のピーポー音がした。



チャレンジ カッコーのプログラムを作ろう。

(プログラムのしくみ)

1200Hz、150 ミリ秒 → 無音200 ミリ秒
 → 950Hz、250 ミリ秒 → 無音1秒のくり返しです。

(プログラムと感想)

```

ずっと
音を鳴らす (Hz) 1200 長さ (ミリ秒) 150
一時停止 (ミリ秒) 200
音を鳴らす (Hz) 950 長さ (ミリ秒) 250
一時停止 (ミリ秒) 1000
    
```



音階と周波数

●各音階と周波数の関係を下表に表します。

ド	ド#	レ	レ#	ミ	ファ	ファ#	ソ	ソ#	ラ	ラ#	シ
131	139	147	156	165	175	185	196	208	220	233	247
262	277	294	311	330	349	370	392	415	440	466	494
523	554	587	622	659	698	740	784	831	880	932	988

救急車のピーポー音

●右は救急車が通り過ぎるプログラム例

チャレンジ 時間があれば取り組む課題です。

※遅い児童は、あとで時間があるときにするように指導します。

●横断歩道のカッコー音の例

※プログラムの記述は、略図で描かせるか、画面をハードコピー（キャプチャー）して印刷したものを張らしてもよいでしょう。

(参考)ピヨピヨ音

●横断歩道のピヨピヨ音の例

Scratch script for ambulance siren sound:

- ずっと (Forever loop)
- くりかえし 4 回 (Repeat 4 times)
- 音を鳴らす (Hz) 800 長さ (ミリ秒) 500 (Play sound 800 Hz, 500 ms)
- 音を鳴らす (Hz) 990 長さ (ミリ秒) 500 (Play sound 990 Hz, 500 ms)
- くりかえし 4 回 (Repeat 4 times)
- 音を鳴らす (Hz) 770 長さ (ミリ秒) 500 (Play sound 770 Hz, 500 ms)
- 音を鳴らす (Hz) 960 長さ (ミリ秒) 500 (Play sound 960 Hz, 500 ms)
- くりかえし 4 回 (Repeat 4 times)
- 音を鳴らす (Hz) 740 長さ (ミリ秒) 500 (Play sound 740 Hz, 500 ms)
- 音を鳴らす (Hz) 930 長さ (ミリ秒) 500 (Play sound 930 Hz, 500 ms)
- 一時停止 (ミリ秒) 2000 (Pause 2000 ms)

Scratch script for chirping sound (ピヨピヨ音):

- 関数 ピヨピヨ (Function: Chirp)
- くりかえし 4 回 (Repeat 4 times)
- くりかえし 2 回 (Repeat 2 times)
- くりかえし 4 回 (Repeat 4 times)
- 変数 index を 0 から 200 に変えてくりかえす (Change variable index from 0 to 200 and repeat)
- 音を鳴らす 高さ (Hz) 200 - index × 20 + 2000 (Play sound with height: 200 - index × 20 + 2000 Hz)
- 休符 (ミリ秒) 1/4 拍 (Pause 1/4 beat)
- 一時停止 (ミリ秒) 1000 (Pause 1000 ms)

(2) 警報器を作ってみよう。



チャレンジ 冷蔵庫を開けたら知らせる警報器のプログラムを作ろう。

(プログラムのしくみ) 冷蔵庫を開けたときに、明るくなるので、そのときだけ音が鳴るようにします。

(プログラム)

```

ずっと
もし 明るさ > 0 なら
  メロディを開始する タタダム くり返し 一度だけ
  一時停止 (ミリ秒) 5000
  
```



チャレンジ 居眠り警報器のプログラムを作ろう。

(プログラムのしくみ) ウウトウトしたときに、頭に付けた micro:bit が傾き、音が鳴るようにします。

(プログラム)

```

左に傾けた の時
  音を鳴らす (Hz) 1200 長さ (ミリ秒) 150
  一時停止 (ミリ秒) 200
  音を鳴らす (Hz) 950 長さ (ミリ秒) 250
  一時停止 (ミリ秒) 1000
  
```



警報器を作ってみよう

チャレンジ 時間があれば取り組む課題です。

※遅い児童は、あとで時間があるときにするように指導します。

※プログラムの記述は、略図で描かせるか、画面をハードコピー（キャプチャー）して印刷したものを張らしてもよいでしょう。

追加チャレンジ ※余裕のある児童用の課題

●冷蔵庫が開いたら、離れたところにある別の micro:bit に知らせる警報器



チャレンジ 時間があれば取り組む課題です。

※遅い児童は、あとで時間があるときにするように指導します。

※プログラムの記述は、略図で描かせるか、画面をハードコピー（キャプチャー）して印刷したものを張らしてもよいでしょう。

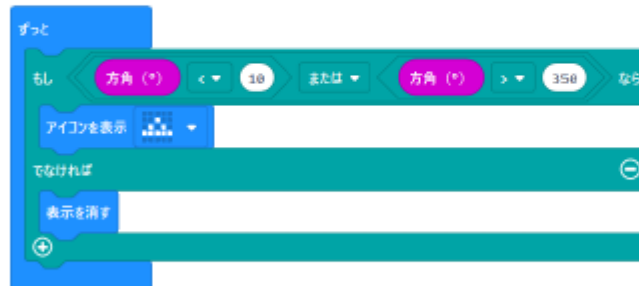
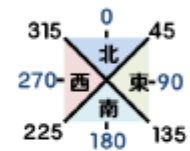
●「傾斜」ブロックを使うとわずかな傾きでも音が鳴ります。



(参考)その他のできること

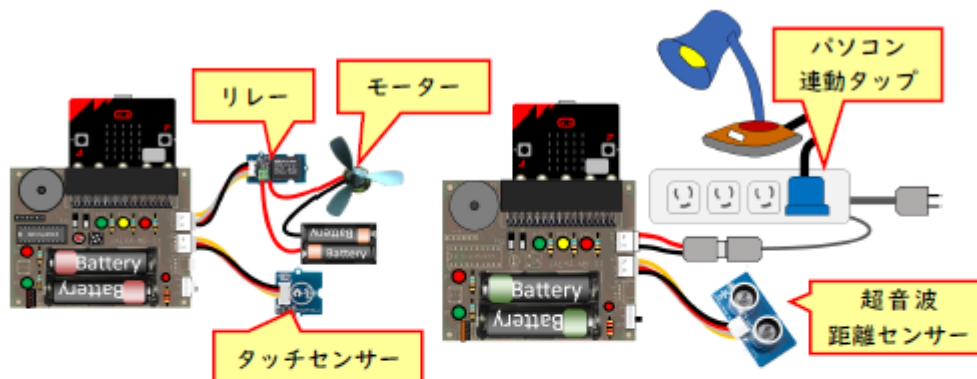
(1) 地磁気センサーを使う

地磁気センサーで測定した方角の角度から方位磁石を作ることができます



(2) ものを動かす (IoT)

micro:bit を使って、LED (発光ダイオード) を点めつしたり、モーターを使ったり、家庭電化製品を動かすことができます。また、micro:bit にないセンサーをつないで測定することもできます。



(3) ゲーム

micro:bit だけ、またはパソコンのソフト (scratch…スクラッチなど) と組み合わせたゲームがインターネット上に数多く紹介されています。

地磁気センサーを使う

地磁気センサーを使ったプログラムを作ったら、最初にキャリブレーション（更正）プログラムが起動します。

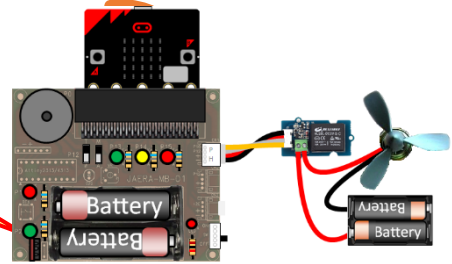
※キャリブレーション…「TILT TO FILM SCREEN」と表示され、LED画面全てのLEDが点灯するよう micro:bit を傾けます。

拡張機能（出力装置）

mbAT は 3 ～ 3.3 V 系の出力装置を接続できます。

※3.3V 系の場合は、別売の DCDC コンバータをはんだ付け使用してください。

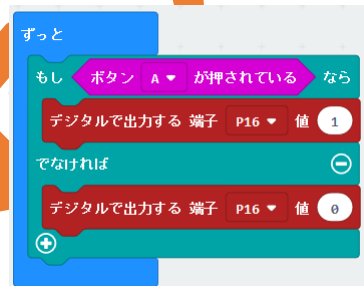
DCDC コンバータ



(1) 3.3～5V 系リレーモジュールを接続した例

※このリレーでは直流のおもちゃなどの制御に使います。家電製品を制御するのは非常に危険ですので、下記の USB 連動タップを使用してください。

- ①ジャンパー線ははずし、DCDC コンバータをはんだ付けします。
- ②PH コネクタをはんだ付けします。
- ③PH コネクタでリレーモジュールを取り付けます。
- ④PI6 ピンを High/Low することで制御できます。



(2) 2 台のリレーモジュールを接続した例

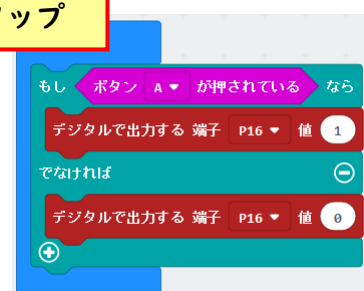
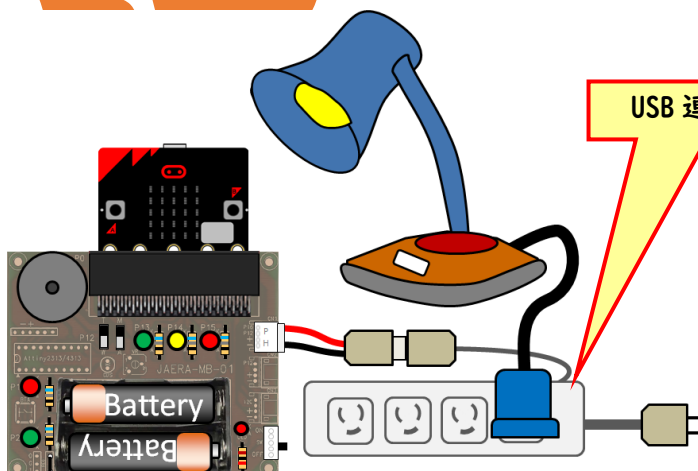
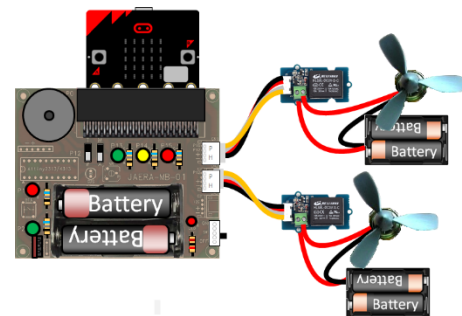
PI6、I2 ピンを High/Low することで制御できます。

(3) 家電製品を制御する場合

- 100V 15A までの家電製品（扇風機や電熱器など）の制御は、USB 連動タップを使います。

※USB 連動タップは 3V で動作するもの

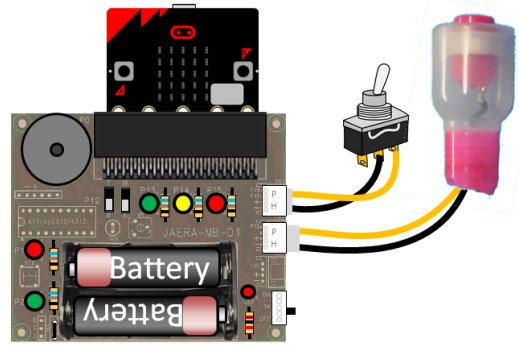
- PI6、I2 ピンの High/Low で制御できます。



拡張機能（入力装置）

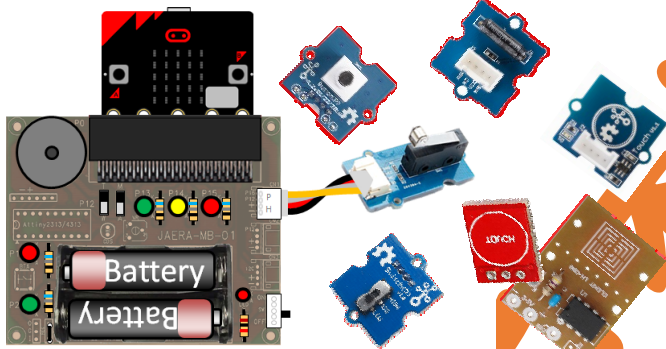
（1）スイッチの接続例

- スライドスイッチ、ボタンスイッチ、マイクロスイッチなどを接続できます。
- P16またはP12とGNDに接続します。
- P16またはP12のHigh/Lowで制御できます。



（2）スイッチモジュールの接続例

- スライドスイッチ、ボタンスイッチ、マイクロスイッチ、傾斜スイッチ、タッチスイッチなどを接続できます。
- P16のHigh/Lowで制御できます。

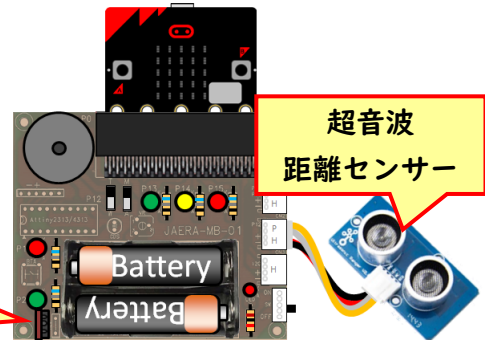


- プログラムでオルタネートロック（押すたびに ON/OFF が切り替わる）などの機能を実現できます。

（3）超音波距離センサー（grove 製を推奨）の接続例

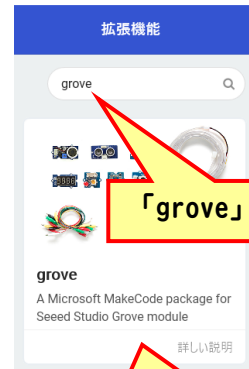
※3.3V 系が少ないので要注意

- 距離を測定（超音波を出し、返ってくるまでの時間で距離を割り出す）し、その距離で制御できます。



「超音波距離センサー」を使ったプログラム

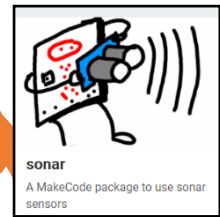
○プログラムする前に、「高度なブロック」の中の「拡張機能」で「grove」で検索し、「grove」メニューを入る必要があります。



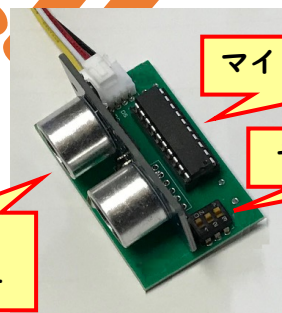
「grove」で検索

「grove」メニュー

○他社製の場合は「sonor」ブロックを使うことが可能ですが、期待通り動かないことがあります。



○あらかじめ設定した距離で P16 や P12 に High/Low を出力するようにプログラムしたマイコンを搭載した「超音波距離センサデジタルモジュール」を使えば、特別なブロックも必要ありません。検知距離は 8 段階設定でき、ディップスイッチで選択します。



マイコン ATtiny4313

ディップスイッチ

超音波
距離センサ

「超音波距離センサデジタルモジュール」

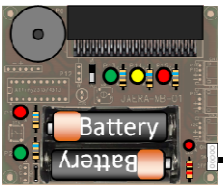
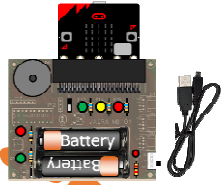
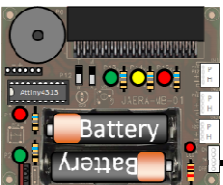
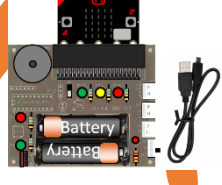


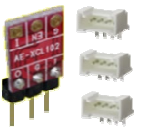
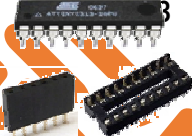

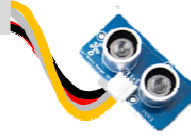
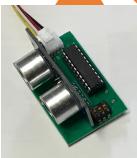

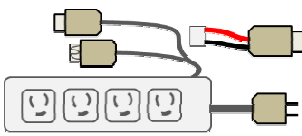
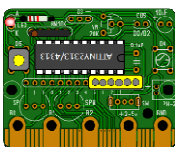
「超音波距離センサデジタルモジュール」を使ったプログラム

《参考》関連製品価格表

(税別、送料別)

●mbAT (エムバット)

	micro:bit と USB ケーブルなし	micro:bit と USB ケーブルあり
基本セット キングエースから販売		
	完成品価格 ¥2000	完成品価格 ¥4150
基本セット JA 教育研究会販売		
	キット価格 ¥2000	キット価格 ¥3000

拡張セット A DCDC コンバータ他	拡張セット B ATtiny4313 他	リレーモジュール タッチセンサなど	超音波距離センサー
			
¥500	¥500	各¥500	¥800
超音波距離センサー デジタルモジュール	micro:bit +USB ケーブル	USB 連動タップ +接続ケーブル	ヒダピオ M ボード (基本完成版)
			
¥1600	¥2500	¥6000	¥1300

《参考》 姉妹冊子

J A教育研究会ではプログラミング学習用の冊子をこれからも発行していく予定です。皆さんのアイデアを形に表したい方は、ご一報下さい。

① micro:bit 入門書

「初めての方への micro:bit 基礎講座」

A 4 サイズ カラー版 40 ページ(表紙込み)

10%税込価格 500 円



② 小学校高学年用学習ノート

「ポンコツロボットと信号機を動かそう！」

micro:bit/mbAT 編」

A 4 サイズ カラー版 32 ページ(表紙込み)

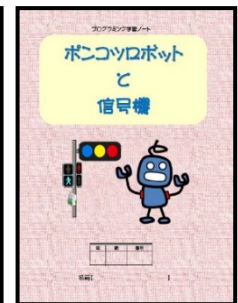
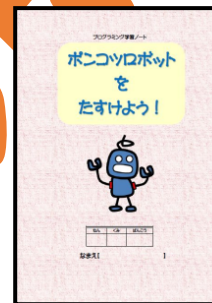
10%税込価格 330 円

③ 小学校低学年用学習ノート

「ポンコツロボットをたすけよう」

A 4 サイズ カラー版 20 ページ(表紙込み)

10%税込価格 200 円



④ 小学校高学年用学習ノート

「ポンコツロボットをたすけよう」

A 4 サイズ カラー版 24 ページ(表紙込み)

10%税込価格 300 円

⑤ スクラッチ入門書

「教師のためのプログラミング講座 3.0」

A 4 サイズ カラー版 48 ページ(表紙込み)

10%税込価格 600 円

「教師のためのプログラミング講座 2.0」

A 4 サイズ カラー版 44 ページ(表紙込み)

10%税込価格 300 円

プログラミング初心者の先生方の簡単な入門書。



複製を禁ず

JAERA	ポンコツロボットのmicro:bitで遊ぼう 教師用指導書	
編集・著作：JA 教育研究会	発行者：JA 教育研究会	【本書の無断転載を禁ず】

非売品 RBMBR02A