

プログラムによる計測・制御を学ぼう

micro:bit 版



わたしたちの生活の中では、コンピュータ制御された機器が数多くあり、これらを便利に使いこなすことで、人にやさしく快適で安全な生活を送ることができます。またエネルギー問題や環境問題に関わって、未来を大切にするために欠かせない技術を多く含んでいます。ここでは、プログラムによる計測・制御のしくみをしっかり学習しましょう。

年 組 番

年 組 番

名前【 】

本書の実習、課題、発展課題の取り扱い

実習 …確かめる内容です。必ず学習しましょう。

課題 …考える内容です。必ず学習しましょう。

発展課題 …発展的な内容で、少し難しい課題です。
時間の余裕があればチャレンジしましょう。

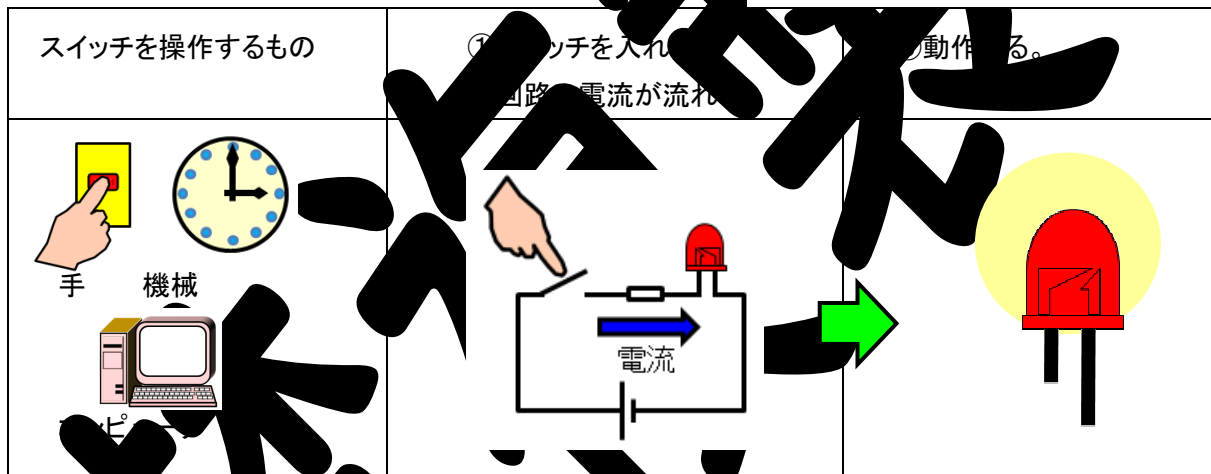
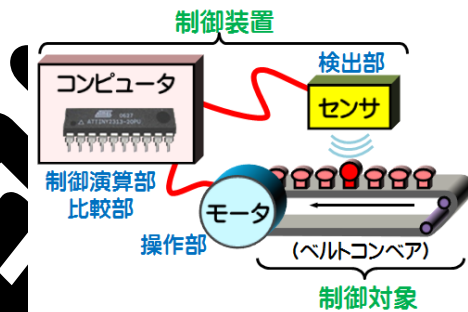
第1章 制御学習の基礎

1. 制御とは

(1) 制御と制御システム

○モータや電灯などのスイッチを ON-OFF するなどして、機械の動作や部屋の環境などを希望するような状態にすることを【 】といいます。

○制御される機器や環境を【 】といい、これを制御するものを【 】といいます。

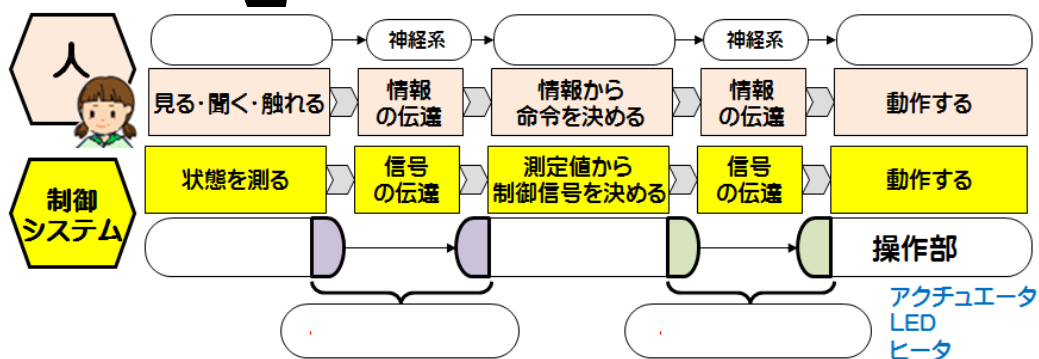


2. 制御システム

○コンピュータ制御する機械やソフトウェアの集まりを【 】と呼びます。

課題1

下の図を完成させよう。



(2) 信号の変換とセンサ

○信号には、なめらかに連続した変化をする【 】と、数値に対応するようにとびとびに変化する【 】があります。

○人間の目や耳の代わりに、制御に必要な信号を取り出す機器を【 】といいます。



(3) センサの種類

センサには、様々なものがあります。

種類	センサ	動作内容
光センサ		明るさにより抵抗値が変化する
		赤外線(人の目には見えない)を検知する
	CCD CMOS イメージセンサ	画像をデジタル化する
温度センサ		温度の増減で抵抗値が変化する
		温度の変化で電圧、接点の通断が変化する

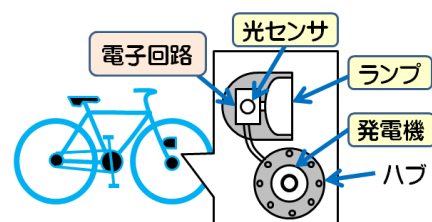
課題2-1

家庭や社会でセンサを使った自動制御されているものを調べよう

制御の例	センサ	制御の内容
(例)電気ポット	温度センサ	湯が沸いたら止める
炊飯器	温度センサ	温度一定になったら加熱を止める
掃除ロボット	距離センサ	位置を記憶して移動する
洗濯機	汚染センサ	汚れの量に応じて洗濯時間を調整する
	明るさ(明暗)センサ	

《参考》センサを使った制御の例

自転車では、常に光センサ(明るさセンサ)が明るさを検知していて、暗くなるとスイッチが自動的に入ってランプが点灯します。



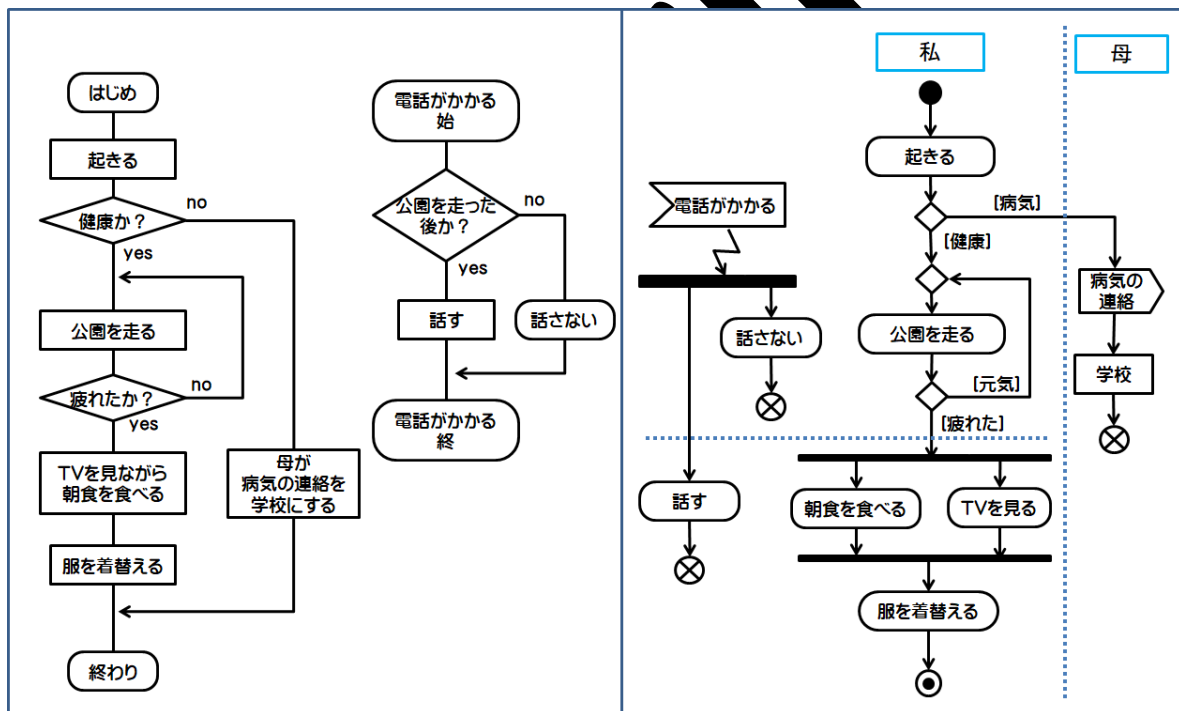
3. 問題解決の方法

(1) 問題解決の手順

○問題を解くためには、まず手順を考えます。この手順を【 】といいます。
○このアルゴリズムを理解しやすくするために【 】や【 】で表します。

問題解決までの流れ

- ①アルゴリズムを考える：図を作成
↓
- ②プログラムの作成：プログラミング言語で記述
↓
- ③プログラムの実行



アクティビティ図

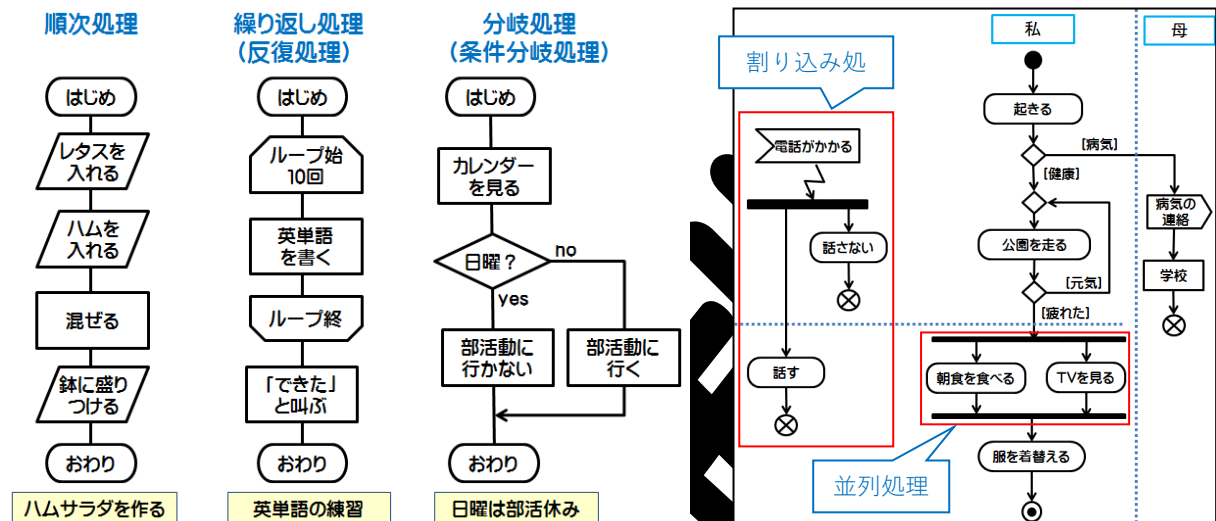
課題2-2

アクティビティ図の作成をまとめましょう。

(2) 基本的な処理

問題解決のための基本的な処理には、

【 】, 【 】, 【 】があります。そのほかに同時に処理を行う【 】や【 】があります。



課題3

日常生活の学習活動など、簡単なフローチャートまたはアクティビティ図で表そう。

Blank area for drawing a flowchart or activity diagram.

4. プログラムとプログラミング言語

(1) プログラム

コンピュータに仕事をさせるための一連の手順を記述したものを【 】といいます。
また、これを作成することをプログラミング、作成する人をプログラマといいます。



(2) プログラムを書く

○プログラムを書くための言葉を【 】といいます。
○プログラミング言語は、プログラムの目的によって色々あります。

5. 第1章のまとめ

(3) 自己評価

コンピュータ制御の基本的な言葉が理解できたか。				D
計測制御システムの基本的な言葉が理解できたか。		C		D
センサの種類や用途が理解できたか。	B	C		D
フィードバックの意味が理解できたか。	B	C		D
問題解決の基本的な3つの処置が理解できたか。	A	B	C	D
フローチャート、グラフ、ビテ図を書くことができたか。	A	B	C	D
プログラミングの基本的な種類について理解できたか。	A	B	C	D
てきぱきと、熱心に作業できたか。	A	B	C	D

(4) 感想

これまで学習して感動したこと、疑問に思ってたことが解決できたこと、将来に役立ちそうなことなどを中心に、感想にまとめよう。

A cartoon illustration of a man with black hair, wearing a blue suit and a red tie, pointing his right index finger upwards.

第2章 micro:bit (マイクロビット) を動かそう

1. マイコンと micro:bit

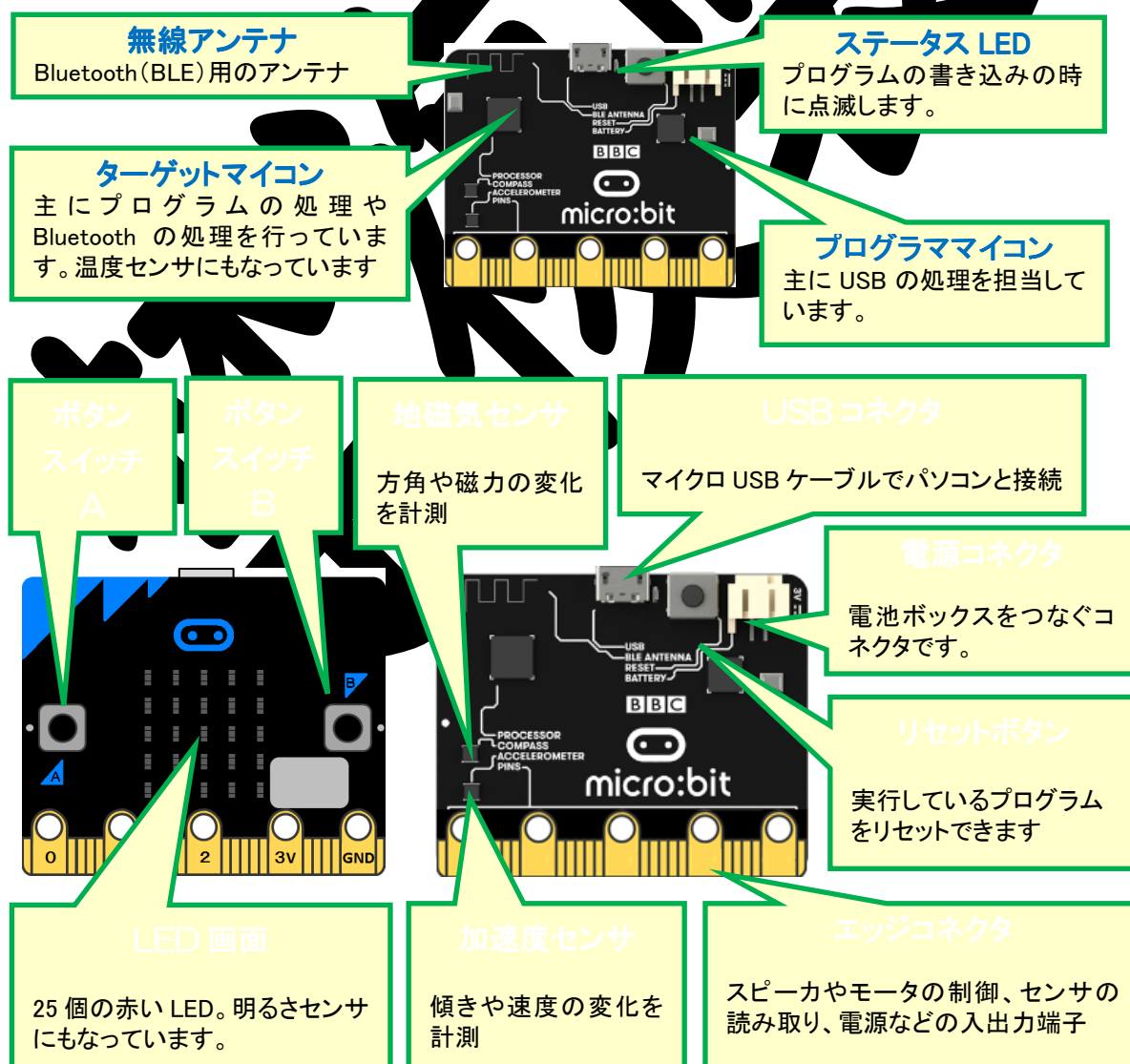
(1) マイコンとは

小さなコンピュータ。家電製品や電子機器などの【 】を必要とする装置に組み込まれています。マイコンという言葉は和製英語。英語では「micro computer(マイクロコンピュータ)」「micro controller(マイクロコントローラ)」または略して「micro」と表現されます。



(2) micro:bit とは

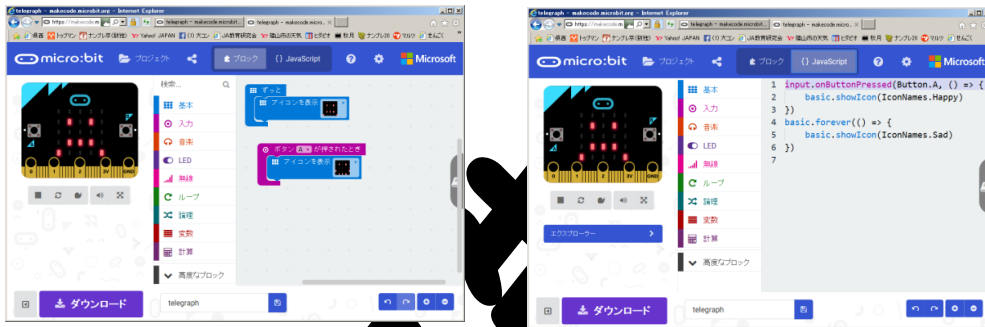
micro:bit とはイギリスの国営放送局である BBC の子供向けの教育のために開発したプログラミングして操作される【 】です。カードサイズの小さなボードで、25個のLED、2つのボタンスイッチがあり、各種センサ(地磁気、加速度、明るさ、温度)と無線通信機能も備わっています。



2. JavaScript（ジャバスクリプト）ブロックエディター

(1) ブロックエディターとは

JavaScript ブロックエディター（以下ブロックエディタ）は、インターネットのブラウザ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。 <https://makecode.micro:bit.org/?lang=ja>



ブロックベース

テキストベース

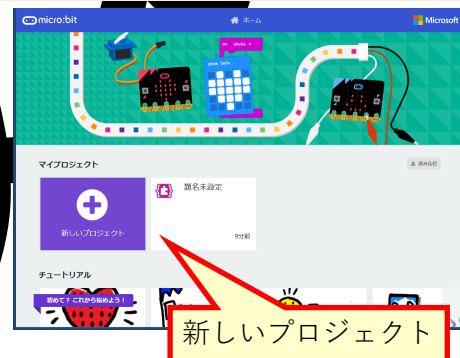
(2) ブロックエディタを開く

ブラウザで下記のURLにアクセス

<https://makecode.micro:bit.org/>

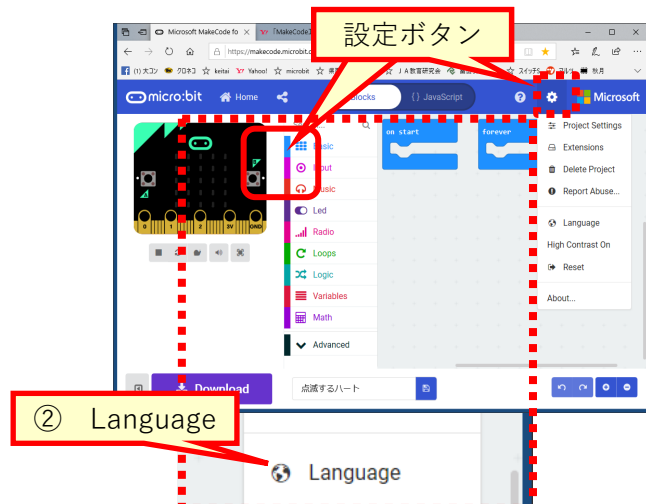
または「makecode」で検索します。

右の「新しいプロジェクト」をクリックします



(3) 言語を変える

ブロックエディターを開いたとき、日本語表記がされていないときは画面左上の「設定ボタン」をクリックし、「Language」から「日本語」を選択します。



(4) ブロックエディターの画面

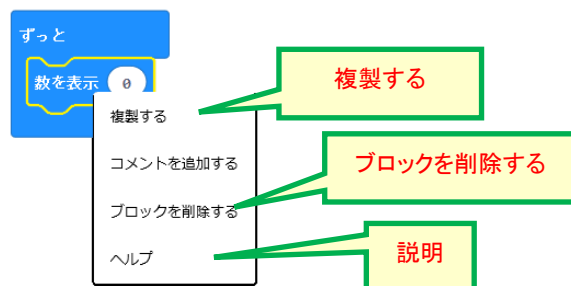


(5) ブロックの貼り付けと削除

- 貼り付け…プログラミングエリアにドラッグ&ドロップ
- 削除…ブロックメニューにドラッグ&ドロップします。



- ブロックを右クリックして表示されるもの



3. micro:bit を使ってみよう

(1) プログラミングの手順（パソコンに保存しない方法）



(2) 簡単なプログラミングの手順例

micro:bit の LED 画面に「0」を表示するプログラムを作成し、プログラミングの手順を覚えましょう。

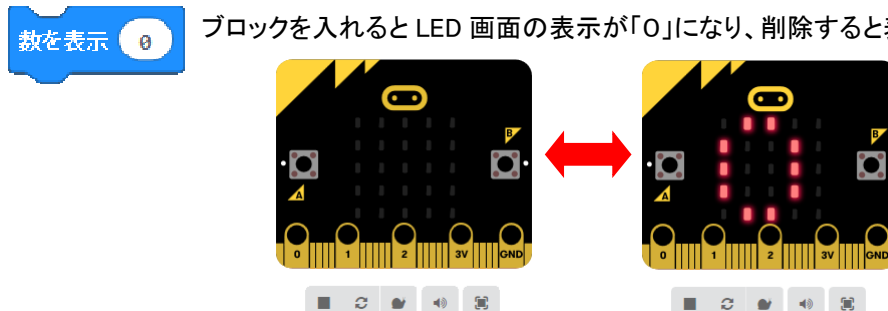
① micro:bit を接続する

② ブロックを組み合わせてプログラミング



③ シミュレーターで動作確認する。

数を表示 [0] ブロックを入れると LED 画面の表示が「0」になり、削除すると表示が消えます。

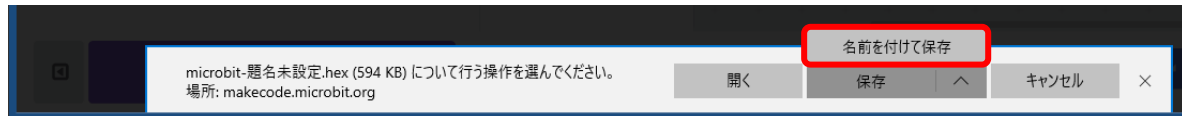


④ プログラムをダウンロード(micro:bit に直接保存)

(1)ダウンロードボタンをクリックします。

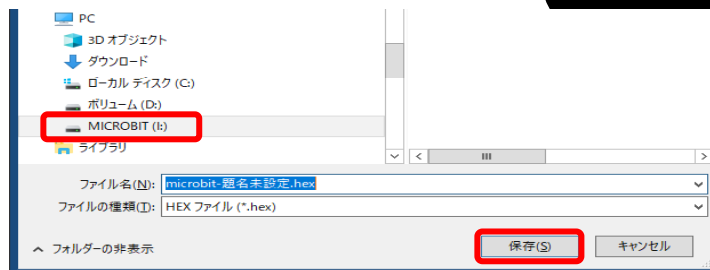


(2)名前を付けて保存をクリックします。

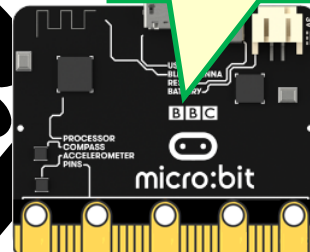


(3)保存先を「MICROBIT」ドライブにして、保存ボタンをクリックします。

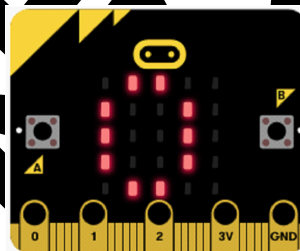
(micro:bit のステータス LED が点滅します)



ステータス LED
プログラムの書き込みの時に点滅します。



⑤ micro:bitで動作確認



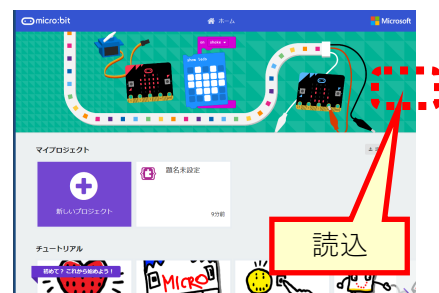
プログラムを一旦保存して micro:bit に書き込む方法

- (1)ダウンロードボタンをクリックします。
- (2)「保存」ボタンを選ぶ。通常は「ダウンロード」フォルダに保存される。
- (3)「フォルダーを開く」ボタンを選ぶ。
- (4)hex ファイルを選んで
- (5)右クリック「送る」を選びを選ぶ。または(5)「MICRO:BIT」ドライブにドラッグ & ドロップする

保存したプログラムを読み込む方法

本ソフトウェアでは、プログラムは Hex ファイル
(拡張子が.hex)の形で保存されます。

- (1)「読み」を選ぶ。
- (2)「ファイルを読み込む」を選び、参照ボタンを押す。
- (3)フォルダを選択する。通常は「ダウンロード」に保存される。
- (4)読み込みたい Hex ファイルを選択し、「つづける」を選ぶ。



4. LED 画面を制御しよう

LED 画面に文字や絵文字を表そう。

実習4-1

文字列を表示

"Hello!"

ブロックを使って“Hello！”と表示されるプログラムを試そう。

※「最初だけ」と「ずっと」のブロックの違いを覚えましょう。



課題4-1

他の文字が表示できるかどうかをシミュレーションし、どんな文字が表示できるかを

確かめよう。

表示できる文字	表示できない文字
ABC, 123, #, \$, %, &, ' ', ~, などの半角文字	漢字, a-z, A-Z, 全角文字

実習4-2

アイコンを表示



ブロックや

LED画面に表示

ブロックを使ったプログラム

をシミュレーションしよう。

ずっと

アイコンを表示



(感想)

実習4-3 「基本」「その他」の中の「表示を消す」ブロックを使って次のプログラムを試そう。

検索...

基本
その他
入力
音楽

その他
表示を消す
矢印を表示 上向き ↑

ずっと
アイコンを表示
表示を消す

(結果)

実習4-4 「基本」の中の「一時停止（ミリ秒）」ブロックを使って、次のプログラムを試そう。

※一時停止の時間は1秒＝1000 ミリ秒とします。

一時停止（ミリ秒） 100

100 ms
200 ms
500 ms
1 second
2 seconds

ずっと
アイコンを表示
一時停止（ミリ秒） 1000
アイコンを表示
一時停止（ミリ秒） 1000

(結果)

課題4-2 実習4-4の「一時停止（ミリ秒）」ブロックの数字を変えてみよう。

(結果)

一時停止の数字を大きくすれば、プログラムの実行速度が速くなる。
1000ぐらいより小さくしても、プログラムの実行速度は変わらない。

課題4-3 なぜ数字を1000ぐらいより小さくしてもドキドキの速さは変わらないのだろうか。

(理由)

※時間に余裕がある人は **発展課題** に挑戦しよう。

発展課題4-1

ハートがだんだん小さくなるプログラムを考えよう。

(プログラム)

発展課題4-2

右のプログラムを参考に、ロボットのボタンを押したとき、LED画面に表示されるプログラムを考えよう。

(結果や感想)



発展課題4-3

右の「明るさを設定する」ブロックを使って、下のプログラムを参考にして、ハートがだんだん小さくなるプログラムを考えよう。



(結果や感想)

5. ボタンAとボタンBで制御しよう

micro:bit のボタンAとボタンBで別の動作を割り当てる【 】をさせることができます。

実習5-1 「入力」の中の

ボタン A ▼ が押されたとき

ブロックを使って、次のプログラムを試そう。

ボタン A ▼ が押されたとき

アイコンを表示

ボタン B ▼ が押されたとき

表示を消す

(結果)

課題5-1

ボタンによって、感情(うれしい、悲しい)などを作るプログラムを考えよう。

(プログラム)

実習5-2

音を鳴らす 高さ (Hz) 真ん中のド 長さ 1 ▼ 拍

ブロックを使って、次の

プログラムを試そう。



ボタン A ▼ が押されたとき

音を鳴らす 高さ (Hz) 真ん中のド 長さ 1 ▼ 拍

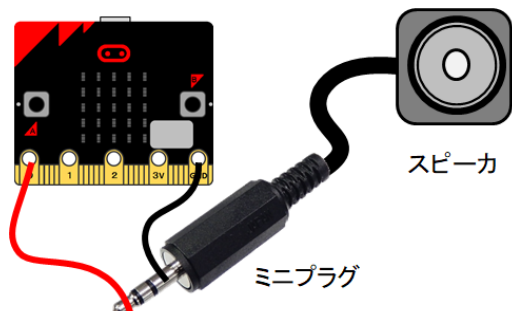
ボタン B ▼ が押されたとき

音を鳴らす 高さ (Hz) 真ん中のミ 長さ 1 ▼ 拍

※micro:bit で音を出すには右図のように、スピーカが別に必要です。

スピーカがない時はシミュレーターで試しましょう。

(結果や感想)



※時間に余裕がある人は **発展課題** に挑戦しよう。

発展課題5-1

横断歩道で鳴っているカッコーの擬音を出してみよう。。

(ヒント) 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。

(感想)

発展課題5-2

(1) 救急車のピーポー音を出してみよう。(ヒント) 周波数は 770Hz と 960Hz です。

(2) A ボタンで近づいてきて、B ボタンで遠ざかる様子を表現してみよう。

(感想)

6. 変数を使ってみよう

変数には【 】や【 文字 】を【 】と【 】き

(1) 変数を作る

実習6-1



変数には何もありませんが「list」「カウンター」「変数」という名前の変数はあらかじめ作られています。



(2) 変数を使う

実習6-2

ボタンAを押すと、カウントアップしながら数字を表示するプログラムを試そう。



(結果や感想)

課題6-1

B ボタンでカウントアップし、ボタンAで数字を表示するプログラムを考えよう。

また、リセットする(変数カウントを0にする)プログラムも考えよう。

(プログラム)

課題6-2

課題6-1

B ボタンのプログラムをいろいろと変えて試してみよう。



(結果と感想)

7. 繰り返しを使う

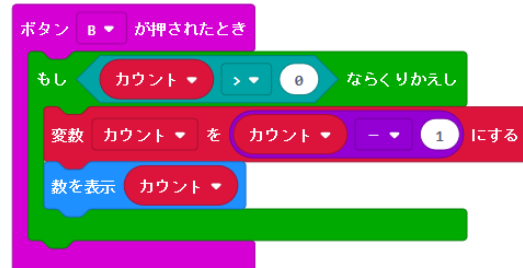
条件が整うまで、ずっと繰り返すことを【**繰り返す**】といいます。

実習7-1

を試してみよう。



ブロックを使った下のプログラム



(結果と感想)

課題7-1

実習7-1

ムを変えましょう。

(プログラム)

「数を表示」ブロックの前に、1 秒停止 20

※時間に余裕がある人は**発展課題**に挑戦しよう。

発展課題7-1

実習7-1

しょう。

(プログラム)

発展課題7-2

課題7-1

正確なカウントダウンプログラムにしよう。(二人で協力して行おう)

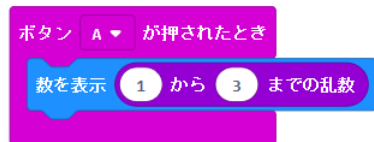
(プログラム)

8. 分岐処理を使う

条件によって行われる処置が異なるようにする処理を【**分岐処理**】といいます。
また、適当な数値が順番なく出てくる数値を【**乱数**】といいます。

実習8-1 「計算」の中の「0 から 10 までの乱数」ブロックを

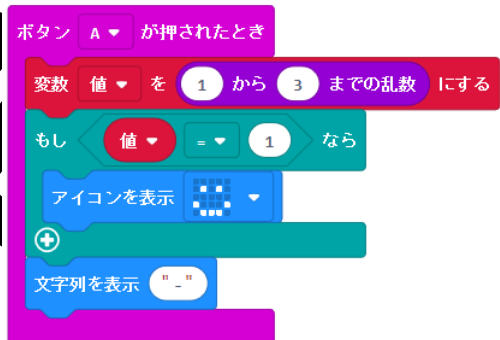
使って、右のプログラムを試そう。



(結果)

実習8-2 「論理」の中の

を使って右のプログラムを試そう

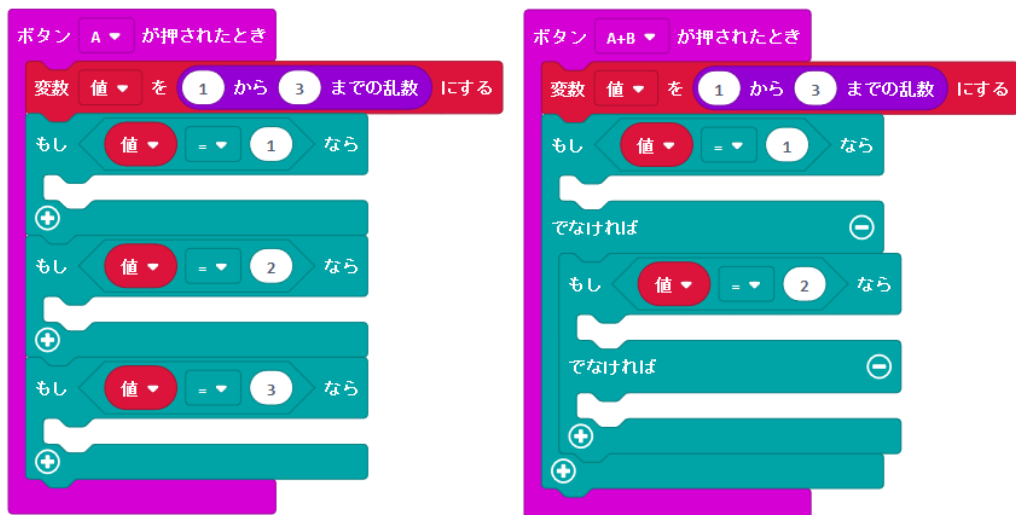


(結果)

課題8-1 **実習8-2** のプログラムを改造して、右のプログラムを作ろう。

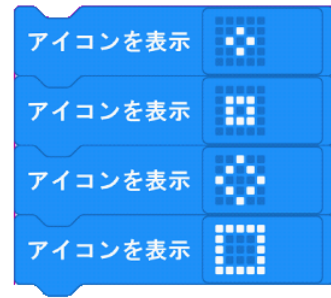
(プログラムの改造)
もし、そうでなければ、変数「値」を、1 から 3 までの乱数にする。
この時は、必ず新しい乱数が出力される。

複数の条件によって処理を行うプログラムの構造です。左と右のプログラムは、同じことができます。



発展課題8-1 ころころサイコロを作ってみよう！

最初に「DISE」「←」と表示。Aボタンを押すと、サイコロが転がるアニメーションの後、さいころの数字が出るプログラムを考えよう。



(ヒント)サイコロが転がるアニメーション

(プログラム)

発展課題8-2 ころころサイコロ2を作ってみよう！

最初に「DISE」「←」と表示。Aボタンを押すと、サイコロが転がるアニメーションの後、さいころの目が出るプログラムを考えよう。

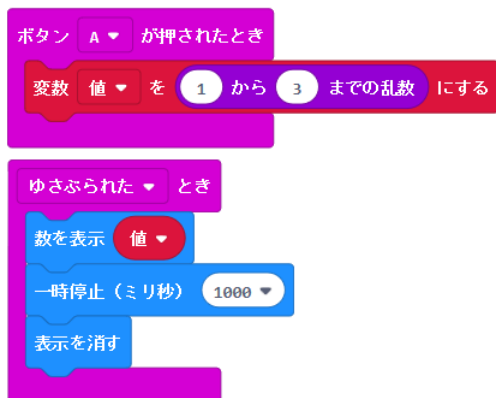
(プログラム)

9. センサで制御する

micro:bit には、【 】 【 】 【 】 が内蔵されています。これにより micro:bit の動きや向き、方角、温度がわかり、別々の動作を割り当てることができます。

(1) 加速度センサを使う

実習9-1 「入力」の中のブロックを使った右のプログラムを試そう。



(結果)

ゆさぶられたとき、micro:bit を振ると「値」が「1」から「3」までの乱数で表示され、1秒間だけ表示された。

課題9-1 micro:bit を使ってじゃんけんができるプログラムを考えよう。

(ヒント) 乱数で出た値は「0」で「パー」、3で「チョキ」の絵文字が出る。

(プログラム)

実習9-2

「右のプログラムを試そう。

スピーカーがない時はシミュレーターで試そう。

ロゴが上になった ▾ とき

メロディを開始する

ジャンプアップ ▾

くり返し

一度だけ ▾

(結果)

課題9-2

課題9-1

のじゃんけんプログラムを、もう少し良いものに改造してみよう。

(例)リセットできるようにする。数字が表示される時にリセットできるなど。

(プログラム)

発展課題9-2

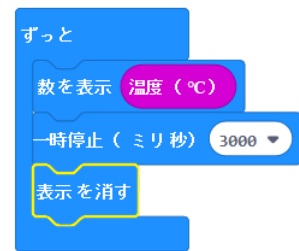
「」の警告音を作ってみよう。

(プログラム)

(2) 温度センサを使う

実習9-3

内蔵された温度センサーで測定した温度が、LED画面に繰り返し表示されるプログラムです。基板の面を触らないように気をつけて、手の中で micro:bit を温めてみよう。



(結果と感想)

発展課題9-3

室温が28度以下だと「○」、それ以上だと「×」を表示するプログラムを考えよう。

(プログラム)

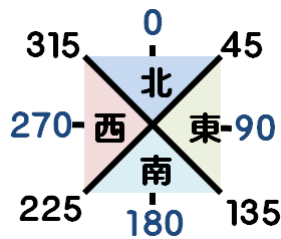
発展課題9-4

温度センサーを使って作ってみよう。

(プログラム)

(3) 地磁気センサを使う

micro:bit に搭載されている地磁気センサーは、micro:bit を水平にした時に向いている方向を、0～359の数値で表し、0(北)・90(東)・180(南)・270(西)となる。

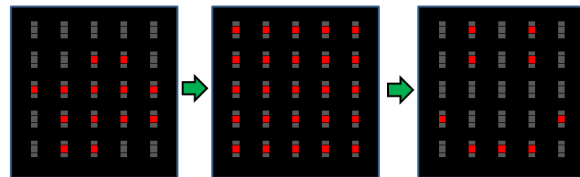


校正 (キャリブレーション)

micro:bit に地磁気センサー関連のプログラムを書き込むと、使用前にコンパスの結果が正確になるように校正プログラムがスタートします。

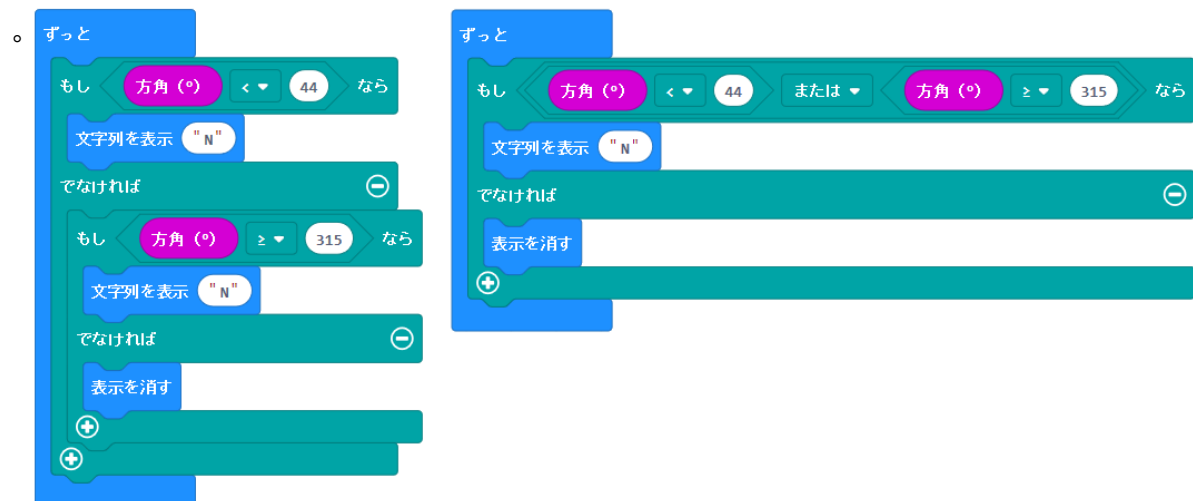
- micro:bit LED画面に「TILT TO FILL SCREEN」と表示される

- micro:bit 本体を回して、LED画面をすべて点灯させる→しばらくすると「笑顔マーク」が出る。



実習9-4

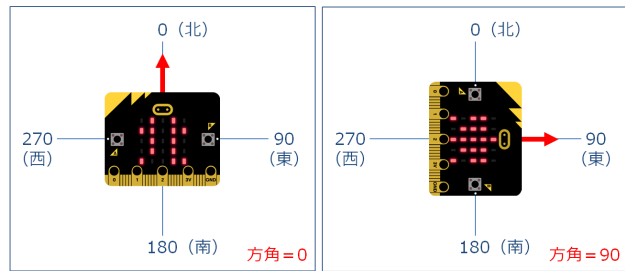
地磁気センサーで測定した方向の角度が、LED画面に表示される。プログラムを試そう。左右とも同じことを行うプログラム。



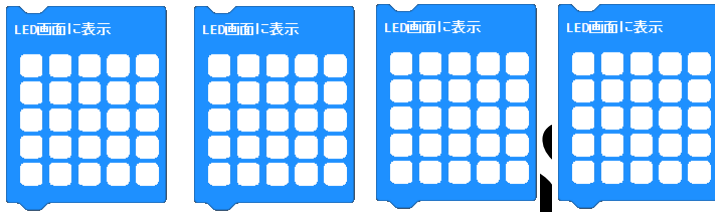
(結果)

発展課題9-5 micro:bit が向いている

方角を、「東・西・南・北」と漢字で表示するプログラムを考えよう。



(プログラム)



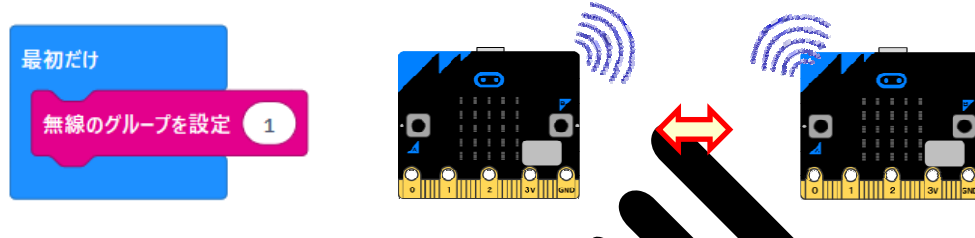
発展課題9-6 micro:bit の画面に、つねに北を示すプログラムを考えよう。



10. 無線通信機能を使って制御する

(1) 無線のグループ

無線を行う前には、必ず無線のグループを指定します。無線のグループ(0~255)が同じなら、無線で数値や文字を複数の micro:bit 間で通信することが可能です。



(2) 送受信内容と割り込み処理

micro:bit では、無線で数値や文字を受信したとき、割り込み処理ができます。

送信	受信(割り込み処理)	送受信内容
数値 	無線で受信したとき 	無線で数値を送信
文字 	無線で受信したとき 	無線で文字列を送信
文字と数値 	無線で受信したとき 	無線で送信 =

実習10-1

このプログラムを使って、グループの人と試そう。

※無線グループは先生の指示に従って決めましょう

(結果や感想)



課題10-1

B ボタンや A+B ボタンでも文字列を送る機能をつけたプログラムを考えよう。

(プログラム)

発展課題10-1

送信した文字列がかわるプログラムを作ってみよう。

(プログラム)

発展課題10-2

micro:bit の無線機能を使って生活に役立つ装置を考えてみよう。

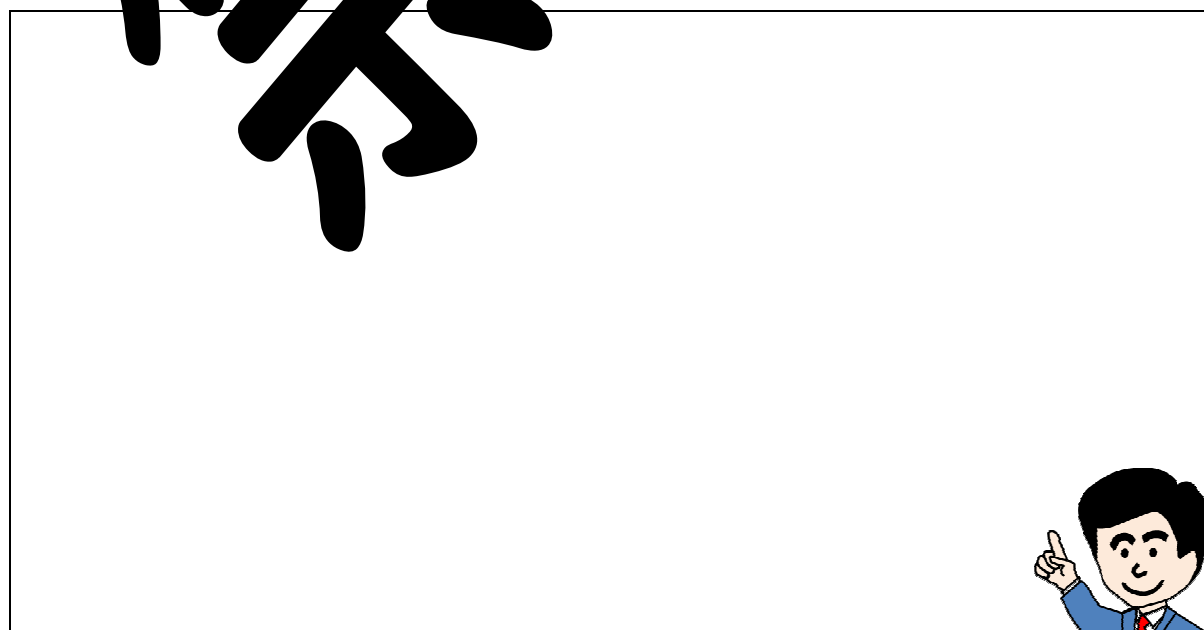
11. 第2章のまとめ

(1) 自己評価

LED 画面の表示の方法を理解し、プログラムを考えることができたか。	A	B	C	D
温度センサによる制御について理解し、プログラムを考えることができたか。	A	B	C	D
ボタンで制御する方法を理解し、プログラムを考えることができたか。	A	B	C	D
乱数について理解し、プログラムを考えることができたか。	A	B	C	D
音声の出し方を理解し、プログラムを考えることができたか。	A	B	C	D
加速度センサによる制御について理解し、プログラムを考えることができたか。	A	B	C	D
地磁気センサによる制御について理解し、プログラムを考えることができたか。	A	B	C	D
無線機能を使った制御について理解し、プログラムを考えることができたか。	A	B	C	D
熱心に授業に参加できたか。	A	B	C	D

(2) 感想

これまで学んだことや、疑問に思っていたこと、将来に役立ちそうなことなどを中心に、感想を述べてください。



第3章 micro:bit と制御回路

1. IoT

(1) IoTとは

【 】の略で「アイオーティー」と読みます。
直訳すると「モノのインターネット」ですが、インターネット経由で【 】
や【 】を持った機器がつながり、互いに【 情報 】をやり取りし
て動作します。



(2) IoTを実現するには

例えば、携帯電話でエアコンのスイッチを操作しますが、インターネットを介した
通信と制御を行うために複雑そうです。最終的に【 】に流れる【 電流 】
することで実現されます。そのため必ず【 制御回路 】が必要なのです。

2. 制御回路の基礎

(1) 制御回路に使われる図記号

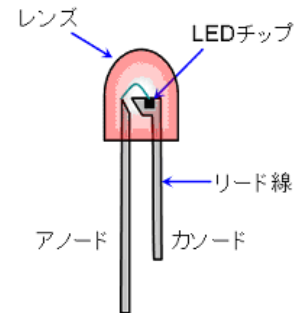
部品名	実体図	部品名	実体図	図記号
直流電源		交流電源		
スイッチ		切換スイッチ		
発光ダイオード (LED)		モータ		
抵抗		可変抵抗 (ボリューム) 半固定抵抗		
CdS セル		スピーカ		

3. 発光ダイオード

(1) 発光ダイオードとは

【 】から【 】に電流が流れるときに発光する半導体の一種で、【 】ともいいます。

Light “光”
Emitting “放出する”
Diode “ダイオード”

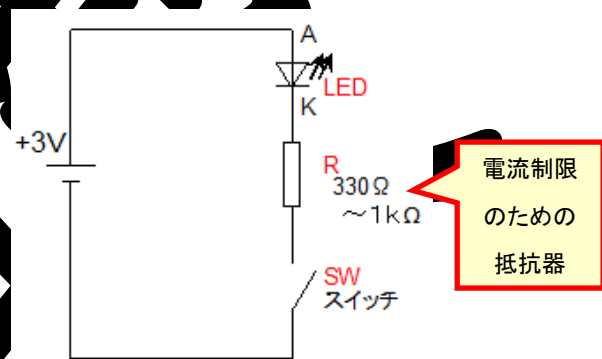


(2) 発光ダイオードと電流制限

LEDは、通常【 】以下の電流で使用します。もし電流が流れすぎると破壊されることがあるため、【 】する必要があります。

電流を制限する主な方法には

- ・【 】を【 】に接続する。
 - ・定電流ダイオードを使う
- などがあります。



(3) 極性

電池の+と-の極性のアノード、カソードのように、電気的性質などが全く正反対なものを【 】といい、極性がある部品は取り付け向きに注意しなければなりません。

4. 抵抗器

(1) 抵抗器とは

電気の流れを妨げる機器で、回路に【 】や【 】を作るはたらきをします。

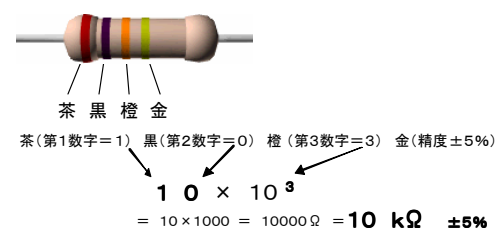
(2) 単位の読み方

Ω【 】、kΩ【 】、MΩ【 】

※抵抗の時だけ、メガオームと呼ぶ習慣があります。

(3) カラーコード

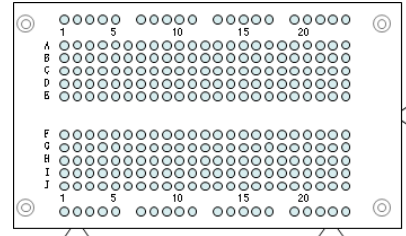
抵抗器の【 】や精度は、【 】で示されています。



5. ブレッドボード

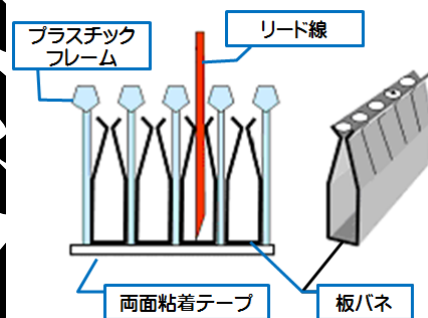
(1) ブレッドボードとは

パン生地をこねる木の板のこと。昔、これに釘を打ち、電気部品や銅線をはんだづけし、電気回路を試行錯誤していたことに由来します。電子素子の交換が容易なので回路変更が自在にでき、このため回路の試作や実験・教育用として広く用いられています。



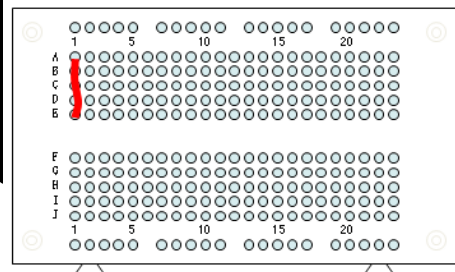
(2) ブレッドボードの穴のつながり

右図のように金属製の板バネがプラスチックフレームに挿入されている。これにより、縦、横のいくつかの穴に電気が通じることができます。



実習5

右の図の穴のつながり方を知るよう
線を記入しよう



(3) 部品のさし方

この穴のつながりを利用して、電子部品やジャンパー線を使って回路を組みます。元々つながっている穴に、同じ部品を差し込まないようにします。

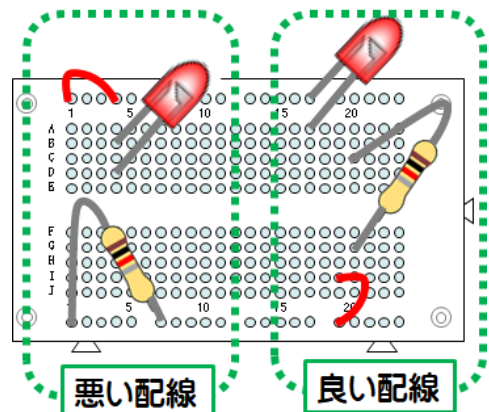
(4) ブレッドボードの穴への部品のさし方

・部品は【 】に、約【 】の深さまでさしこみます。

※短すぎると接触不良に、長いと抜けにくくなります。

※差し込みにくい場合は、先端をやや斜めにカットします。

※リード線が一度曲がると、さらに差し込みにくくなります。



6. 簡単な回路学習

ブレッドボードにLEDや抵抗などの電子部品などをさして回路を作り、動作を確かめましょう。



(1) 実習の注意

①電源の＋と－を直接つなげるような【**ショート回路**】(ショート回路ともいう)ができないようにします。また、電源は全ての回路ができてからつなげるようにします。

※電池から大量の電流が流れ、電池が発熱したり、時に破裂する恐れもあります。

②LEDと抵抗は【**向き**】につなぎ、間に他の線を挟まないようにします。

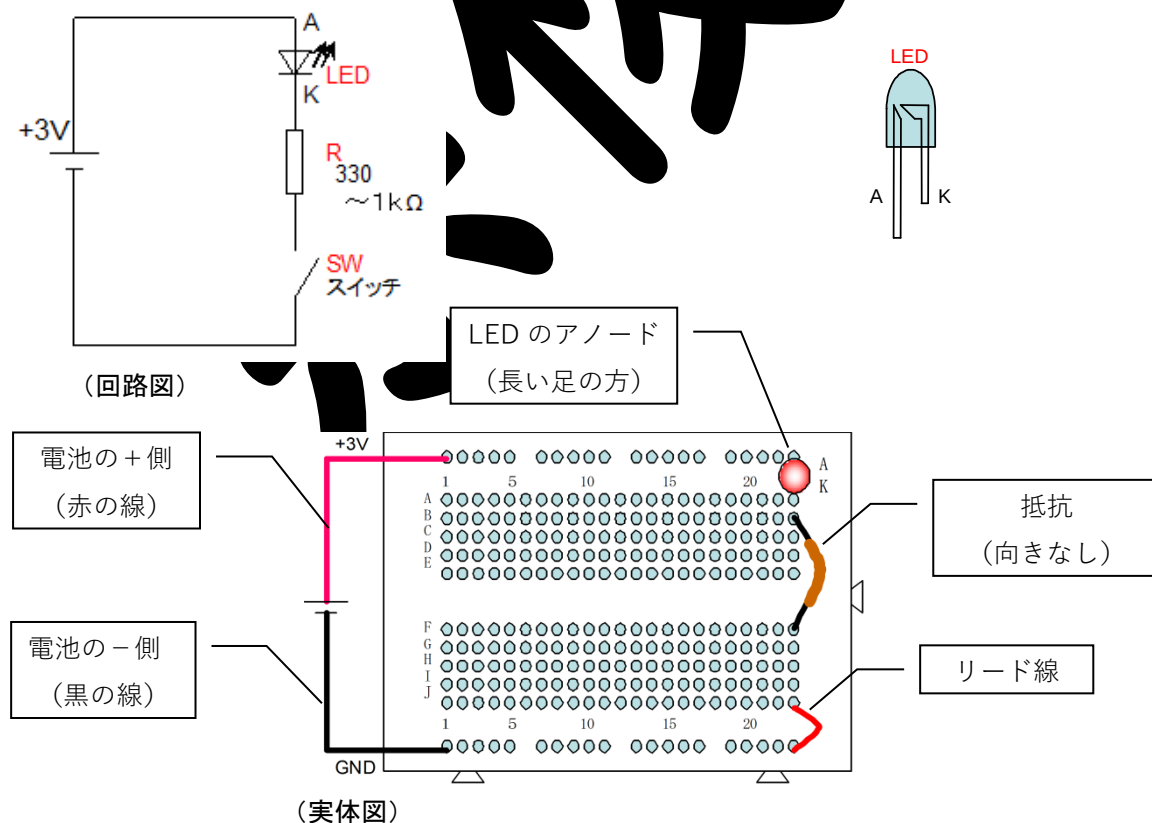
※LEDに電源電圧が直接かかると、LEDが破損することがあります。

③実習終了後は、電池ボックスから電池を抜いておきましょう。

実習6

回路図を見て、下の2枚の図のように必要な部品をブレッドボード上に回路を作って動作を確かめてみよう。スイッチはONとOFFを繰り返して確認します。

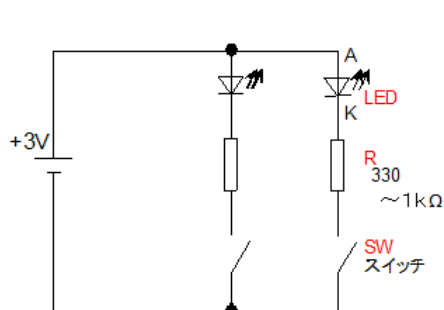
(2) 1 個のLED（発光ダイオード）を点滅する回路



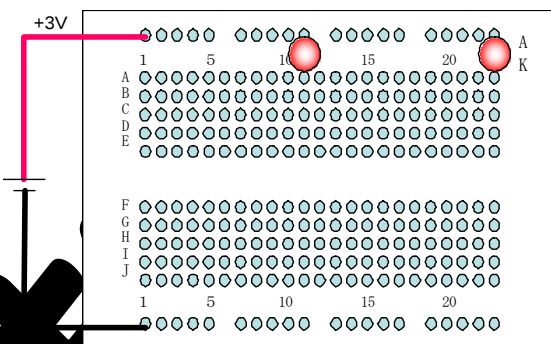
課題6

回路図を見て、実体図に必要な部品を書き込み、ブレッドボード上に回路を作ろう。

(3) 2 個の L E D（発光ダイオード）をそれぞれ点滅する回路

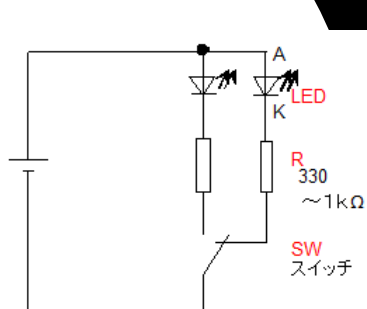


(回路図)

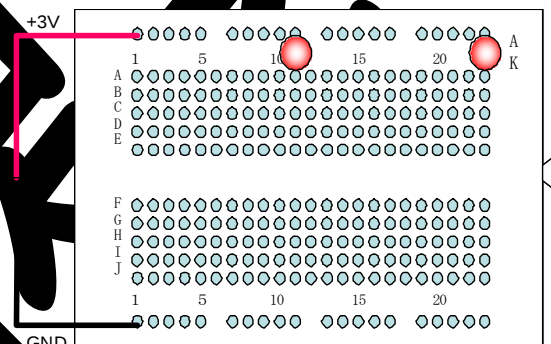


(実体図…部品を追加する)

(4) 2 個の L E D（発光ダイオード）のどちらかを点灯する回路



(回路図)



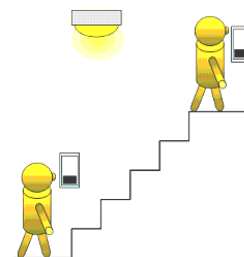
(実体図…部品を追加する)

発展課題6

階段や長い廊下で使われる回路の回路図を考えてみよう。

(5) 階段や長い廊下の電灯の回路【難問】

(ヒント) 切換スイッチ2個を使います。



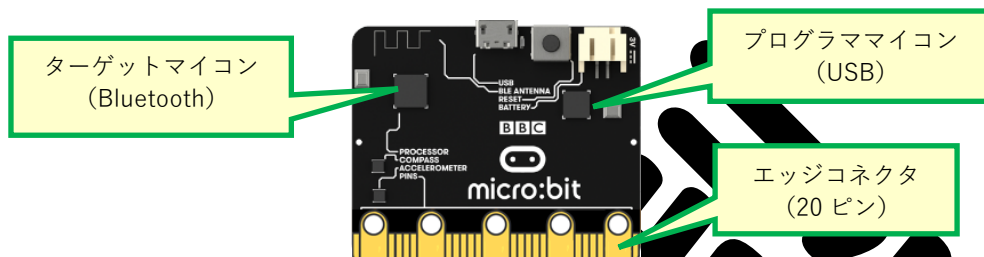
7. micro:bit で L E D を点灯する。

(1) micro:bit のマイコンと入出力ピン

○micro:bit ではターゲットマイコンが【 】の【 (I/O) 】をしています。

○マイコンには【 】と呼ばれる足が数10本あり、これを通じてデータの入出力を行います。

○micro:bit のデータを入出力するピンは【 】につながっています。

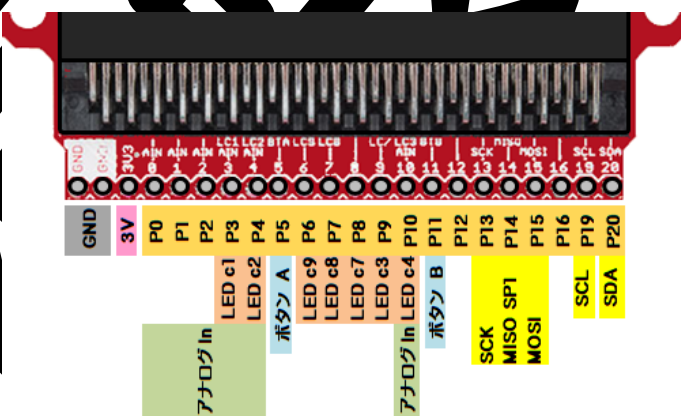
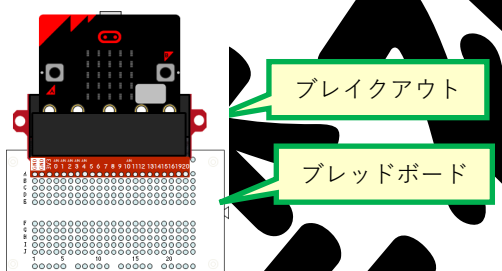


(2) ブレイクアウトとブレッドボード

ブレッドボードで micro:bit を使うために

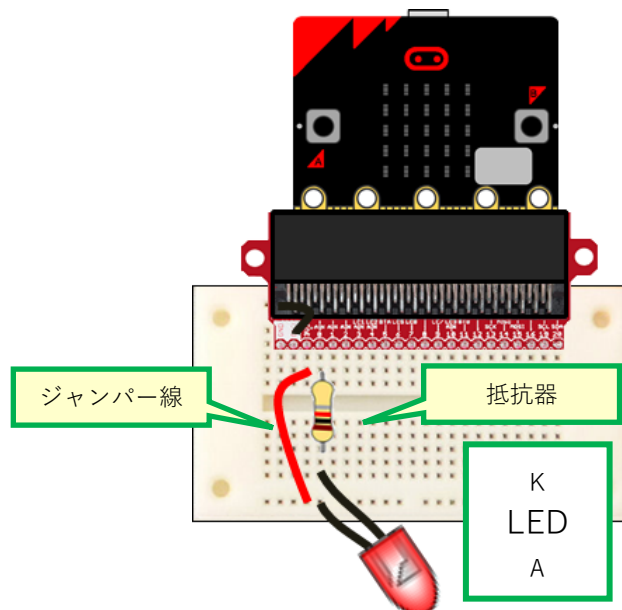
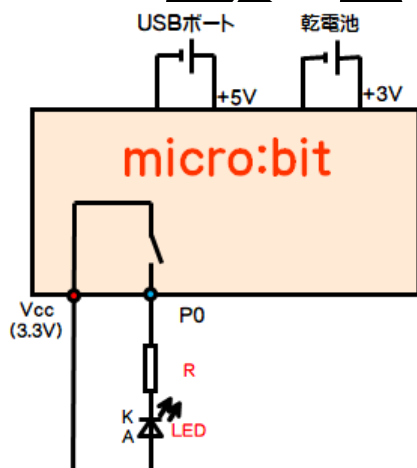
【 】を使います。

ブレックアウトには micro:bit のマイコンのピン番号などが記入されています。



(3) micro:bit と L E D の接続

micro:bit の P0 ピンに LED を接続します。



(4) プログラムで点滅する

実習7-1 「高度なブロック」「入出力端子」

の中の

デジタルで出力する 端子 P0 ▼ 値 0 ブロック

を使って、LED を光らせよう。



(結果と感想)

実習7-2 右のプログラムでLEDの点滅の速く遅くを変えよう。

(結果と感想)

LEDの点滅の速く遅くを変えよう。時間を変える。ゆっ

点滅した。



課題7 点滅の速く遅くを変えよう。

(結果と感想)

発展課題7 LED の数を増やしてみよう。

(結果と感想)

8. 第3章のまとめ

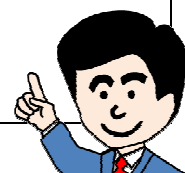
(1) 自己評価

コンピュータ制御の基本的な言葉や回路が理解できたか。	A	B	C	D
発光ダイオードや抵抗のしくみや役割が理解できたか。	A	B	C	D
ブレッドボードの穴のつながりや使い方が理解できたか。	A	B	C	D
回路図を見て、部品を配置することができたか。		B	C	D
てきぱきと、熱心に作業ができたか。			C	D

(2) 感想

これまで学習して感動したことや、知ることができたこと、将来に役立つことなどを中心に、感想にまとめよう。

感想を書くための大きな空白領域。



第4章 制御学習の基礎

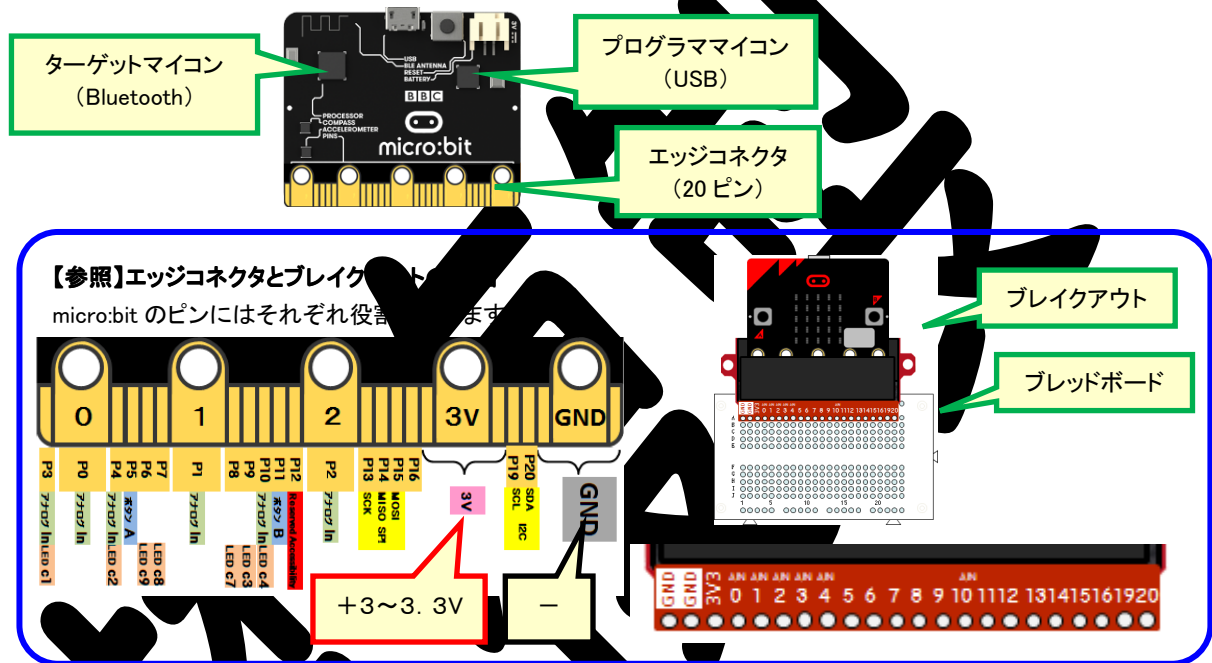
1. マイコンと micro:bit

(1) micro:bit のマイコンと入出力ピン

○micro:bit ではマイコンが【 】の【 】(I/O)をしています。

○マイコンには【 】と呼ばれる足が数10本あり、これを通してデータの入出力を行います。

○micro:bit のデータを入出力するピンはエッジコネクタにつながっています。

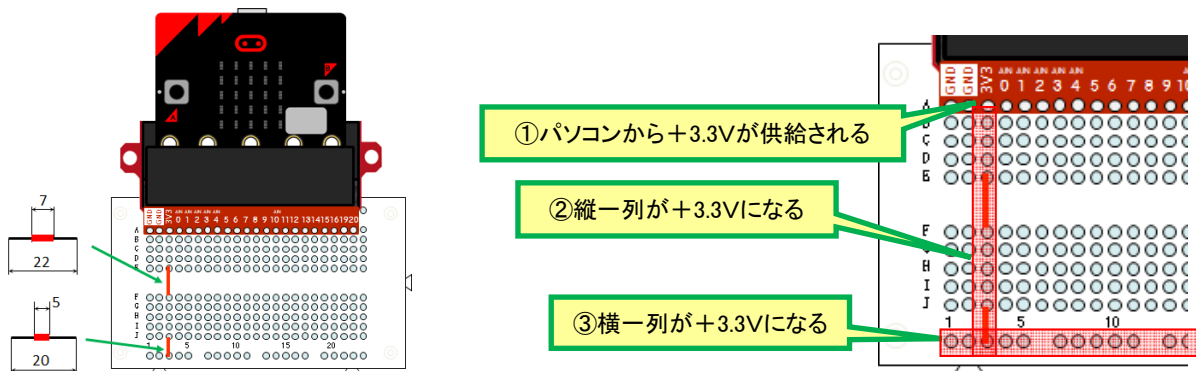


2. 制御実験の準備

(1) LEDの点滅装置の製作

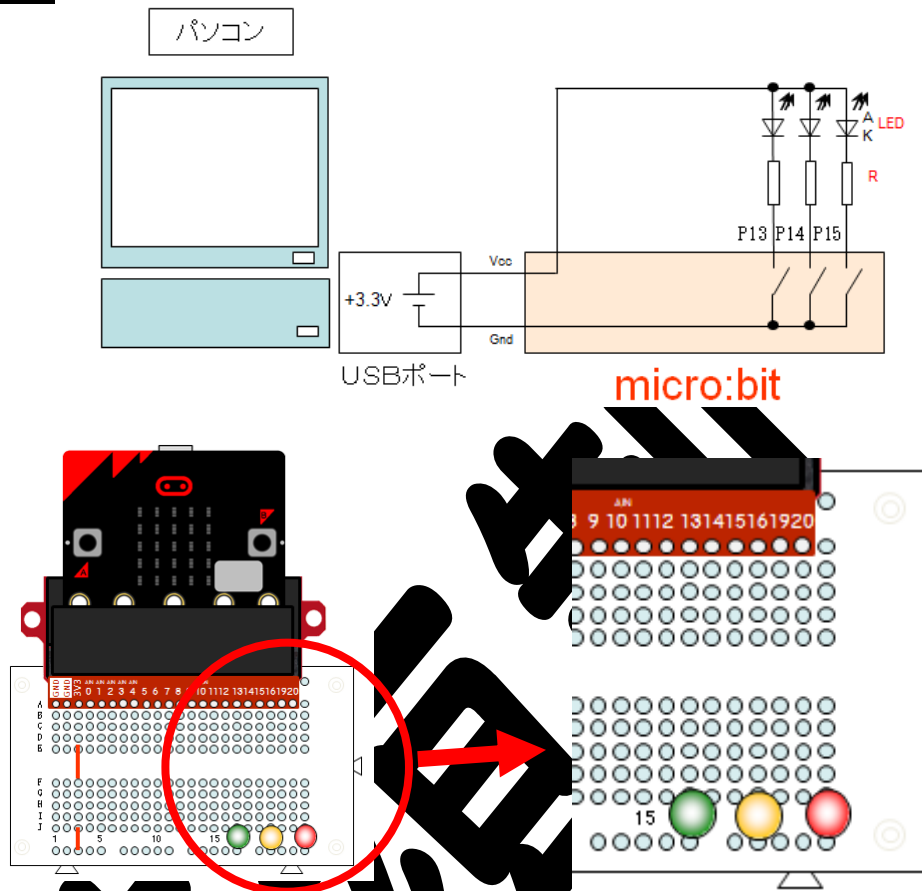
実習2

micro:bit のエッジコネクタをブレイクアウトにさし、さらに【 】にさしてみましよう。同時に【 】の穴を増やすため、2本の【 】(赤色)をさします。



課題2

回路図をみて実体図に抵抗器を書き込んでから、実際にブレッドボードに抵抗器をさそう。



3. LED の点滅制御

(1) LED を点灯する

○「高度なブロック」の入出力端子の中にある「デジタルで出力する」ブロックを利用して、各LEDの値を変えてLEDを点滅します。

○最初は値が「0」点灯、値が「1」点滅、値が「0」点灯、値が「1」点滅と消灯します。



(2) 同じ処理をくり返す

プログラムを何度もくり返し行う処理を【 】処理といいます。



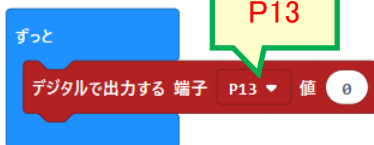
【 】 【 】 【 】



実習3

次のプログラムを作成して、micro:bit にプログラムを書き込もう。

(ア)



P13

(結果)

(イ)



0

1

発展課題3

ブロックエディタとテキストエディタの使い分けの意義を考察してみよう。

ブロックエディタ



テキストエディタ

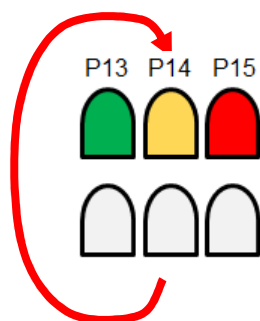
```
1 basic.forever(function () {
2   pins.digitalWritePin(DigitalPin.P13, 0)
3   basic.pause(100)
4   pins.digitalWritePin(DigitalPin.P13, 1)
5   basic.pause(100)
6 })
```

(感想)

例] 最初のブロックの... やりかたは複雑

課題3-1

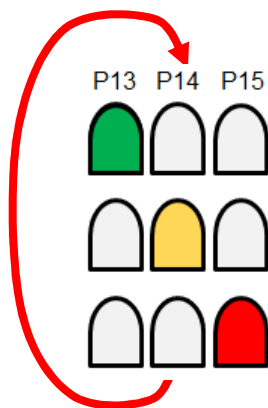
次の動作を実現するプログラムを考えよう。



(プログラムと感想)

課題3-2

次の動作をするプログラムを考えよう

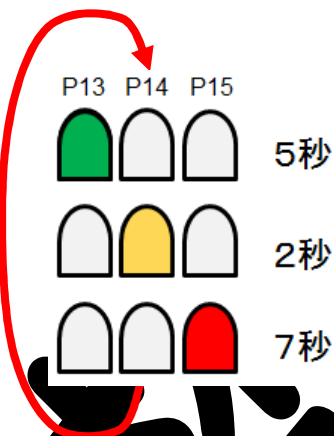


(プログラムと感想)

課題3-3

時間を変えて、信号機のように動作するプログラムを考えよう。

※実習時間の短縮のため、実際の信号機の時間よりも短い時間を使用しよう。



(プログラムと感想)

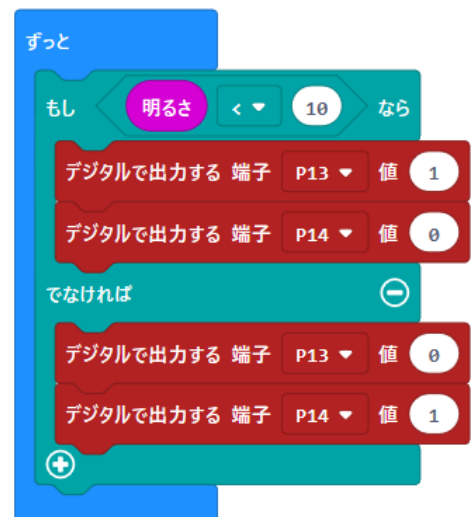
4. 条件によって処理内容を変える

ある条件によってプログラムの処理内容を変えて実行することを【条件分岐】という。

実習4

次のプログラムを作成して、micro:bit の LED ディスプレイを手で隠してみよう。

(結果)



課題4 **課題3**の信号機で、
暗くなると黄色の点滅に変わるよ
うに動作するプログラムを考えよう。

(プログラム)

5. データを一時的に保管する

プログラムの中で、一時的にデータを保管する【変数】というものがあ
る。変数には数字や文字データを入れることができます。



実習5-1

このプログラムを作成して、micro:bitのプログラムを書きこもう。

(ヒント) 変数 変数を追加する... ボタンで、新しい変数 **点滅速さ** をつくろう。

①



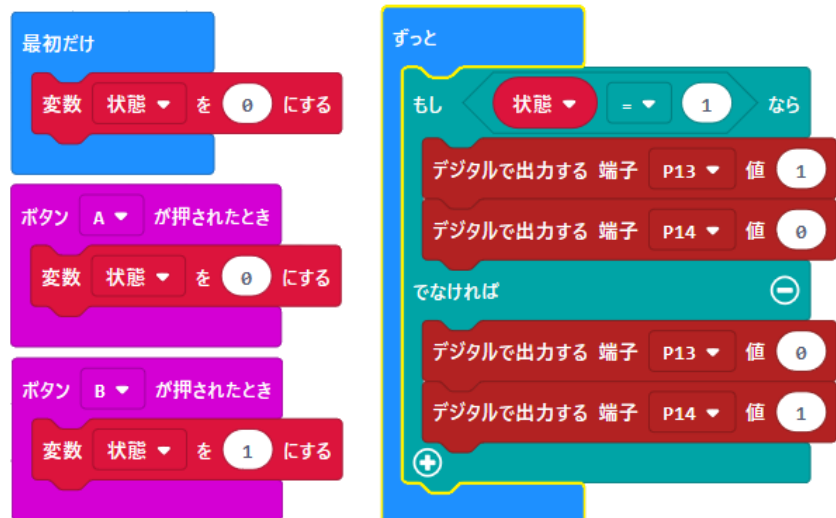
②



(結果)

実習5-2 次のプログ

ラムを作成して、micro:bit に
プログラムを書き込もう。でき
たら micro:bit の A ボタンや B
ボタンを押してみよう。



(結果)

押すと青が点灯する

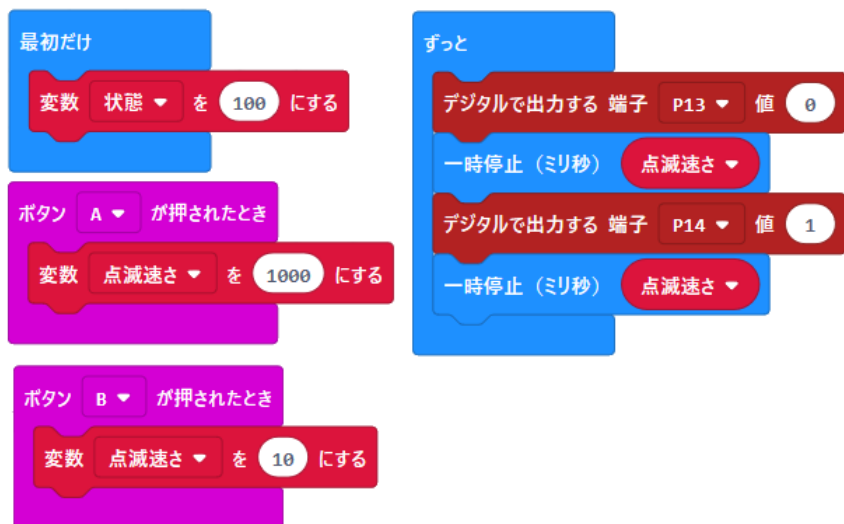
6. 途中に入り込む処理

プログラムが順番に行われる中で、ボタン操作など外部からの操作で、そのプログラムを実行
することを【割り込み処理】と呼ぶ。



実習6

プログラム
を作成して、micro:bit にプ
グラムを書き込もう。でき
たら micro:bit の A ボタンや B
ボタンを押してみよう。



(結果)

micro:bit ではAボタン、Bボタン、P0~1 と GND を使ったタッチセンサ、加速度センサを使った割り込み処理が
簡単にできます。

課題6

他の割り込み処理で、点滅速さを最初の点滅速さ=100 に戻そう。

(プログラム)

7. 同じ処理をまとめる

JavaScript ブロックエディタの中では、ある処理をまとめたものを「関数」(またはブルーチン)といいます。同じ処理を何度も行うときに使用します。



実習7

「課題3」の信号機のプログラムに、関数を使い直して欲しい。

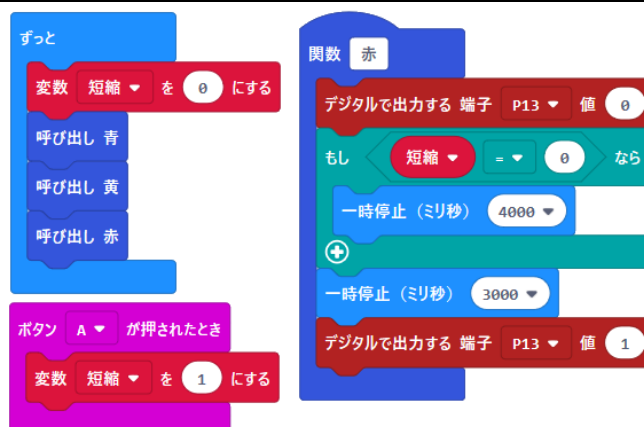


(感想)

が長くかゝりけれ

信号機の歩行者優先ボタンを押すと赤の点滅時間が短くなるプログラムを作ろうと思い、右のようなプログラム(青と黄は略)を組みってみました。

《解説》 普段(青信号の時)は変数「短縮=0」とします。A ボタンを押すと「短縮=1」となります。「短縮=0」の時は赤信号が 4 秒間多く点灯し、「短縮=1」の時は短くなります。



発展課題7

右上のプログラムの問題は何かを考えてプログラムを改造しよう

(問題点と改善方法)

(プログラム) 赤のプログラムだけをかきましょう。

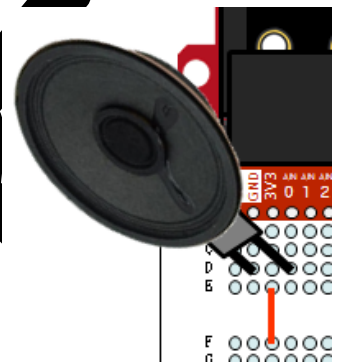
8. 音を鳴らす

スピーカは PO と GND(または3V3)に接続します

実習8

次のプログラムを作ろう。micro:bitにプログラムを書き込もう。

(ヒント)「入出力端子」の「その他」のブロックに「音を鳴らす」ブロックを使います。



課題8

横断歩道で鳴るカッコーの擬音を鳴らしてみよう。

(ヒント) 1200 Hz → 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)

発展課題8

下の信号機のプログラムで、青の時にカッコー音が鳴るようにしてみよう。

(ヒント)カッコー音は「関数」にして、数回鳴るように「有限くり返し」を使います。



(プログラム)改造したところだけを書きまし。

《参考》歩行者用信号の擬音

音響には擬音(ぎおん)式(ピヨ、カッコーなど)とメロディー式(通りゃんせ、故郷の空など)があります。目の不自由な人でを行った実験の結果、渡る方向によって擬音を変える擬音異種鳴き交(か)わし方式が横断方向を理解しやすいことが分かりました。新たに設置されるものは、擬音異種鳴き交わし方式が多くなっています。

9. 無線を使う

- ① micro:bit では、無線で数値や文字を他の micro:bit に送信できます。

無線で数値を送信 0

無線で文字列を送信 " "

- ② micro:bit では無線で数値や文字を受信したときに、割り込み処理ができます。

無線で受信したとき receivedNumber

(数値を受信したとき)

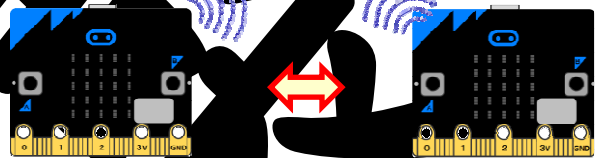
無線で受信したとき receivedString

(文字を受信したとき)

- ③ 無線を行う前には、必ず無線のグループを指定します。無線のグループ(0~255)が同じなら、複数の micro:bit 間で通信が可能です。

最初だけ

無線のグループを設定 1



実習9

無線のグループは自分から番号を決めて、次のプログラムを実行したら、他の人と無線のグループをあわせて、micro:bit のプログラムを互いに入力した micro:bit の A ボタンを押してみよう。

最初だけ

無線のグループを設定 41

最初は
出席番号で

ボタン A が押されたとき

無線で文字列を送信 "A"

無線で受信したとき receivedString

文字列を表示 receivedString

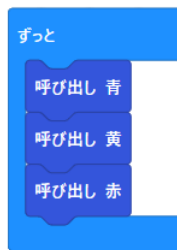
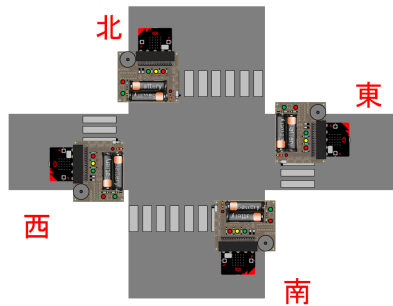
課題9

A ボタン以外の B ボタンや A+B ボタンにも文字列を送る機能をつけて、確かめてみよう。

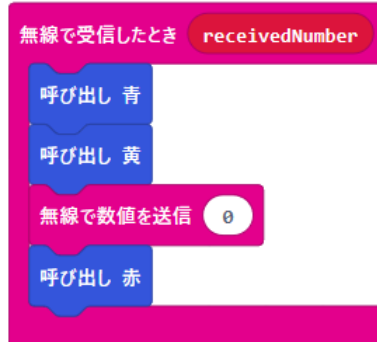
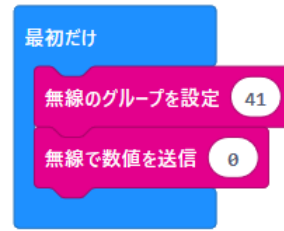
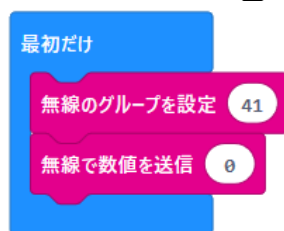
(感想)

発展課題9 2～4人で協力して次の課題に挑戦しよう。

右の信号機のプログラムを元にして、他の人の micro:bit と無線を使って同期する(信号を合わせる)ように改造してみよう。



(ヒント) (同じ向き) 同じになる人のプログラム (進行方向が異なる場合は別のプログラム)



(感想)

10. 歩行者用信号機

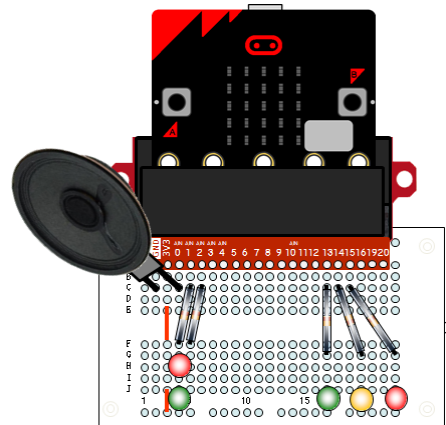
右の図を参考に、歩行者用信号機として P1 に青、P2 に赤の LED を入れ、プログラムを作成してみよう。

(ヒント) 青信号の時に、歩行者用信号機の青は点滅してから赤になります。

課題10

下の表を完成させ、プログラミングしよう。

NO.	車用	歩行者用
1	青	
2		
3	黄	
4	赤	



(プログラム) 「青」の関数を改造しよう。

禁止複製

《参考》 歩行者用信号の点灯時間

歩行者用信号では、一般の人の歩く速さを 1 秒間 1m で計算します。例えば幅 15m の横断歩道は 15 秒で渡れると計算し、これに数秒を足した時間を青点灯時間としています。その後、青点滅を数秒間して赤に変わります。

こうして求められた青点灯時間は一般の人にとっては問題のない長さですが、杖が必要な方や車いすの方の中には渡りきれないことがあります。

11. 交差点を実現しよう 《超難問》

4人(2～3人でも可)で協力して、実際の交差点のように信号機を点滅させよう。

課題11

下の表を完成させよう。

NO.	進行方向		直交方向	
	車用	歩行者用	車用	歩行者用
1	青	青(カッコー音)		
2		青点滅		
3	黄	赤	黄	赤
4	赤			
5				
6				
7				
8				

発展課題11

プログラムを完成せよ

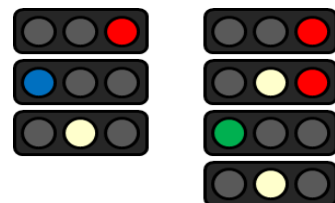
(感想)

たてやうできました ピヨピヨ音は
じのものうできました

《参考》信号機の色と点滅

信号機の色は世界的には「道路標識及び信号に関するウィーン条約」で赤、黄、緑と決められ、点滅順も右のように決められています。

しかし、日本は条約に加盟していないので、ウィーン条約加盟国の信号の点滅方法が異なります。



12. 第4章のまとめ

(1) 【自己評価】

マイコンの役割が理解できたか。	A	B	C	D
micro:bit の入出力ピンについて理解できたか。	A	B	C	D
micro:bit とプログラミング言語の関係が理解できたか。	A	B	C	D
実験回路が正しく組めたか。	A	B	C	D
くり返し処理を理解し、使うことができたか。	A	B	C	D
条件分岐を理解し、使うことができたか。	A	B	C	D
変数を理解し、使うことができたか。	A	B	C	D
割り込み処理や関数を理解し、使うことができたか。	A	B	C	D
音を鳴らしたり、無線を使うことができたか。	A	B	C	D
熱心に授業に取り組めたか	A	B	C	D

(2) 感想

これまで学習して感じたこと、疑問、思ったこと、これから将来に役立ちそうなことなどを中心に、感想にまとめよう。

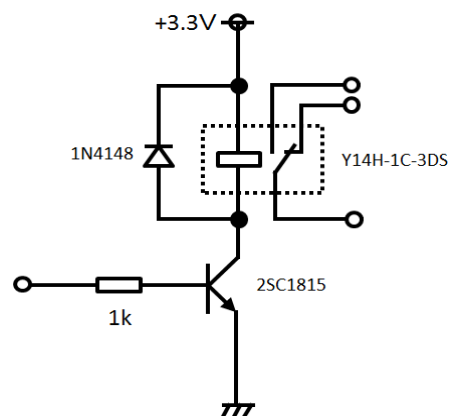


《参考》リレーを使った回路

micro:bit 本体を使った制御では、直流で最大100mAしか制御できません。しかしリレーを使うと、電球・モータ・ヒータなどの大きな電流を使う機器やAC100V機器の制御も行えます。

(使用部品)

- ・リレー(3V仕様)
- ・整流ダイオード
- ・トランジスタ(2SC1815GR)
- ・抵抗器(1kΩ)



第5章 学習のまとめ

1. 人に優しい技術

「人に優しい技術」とは、高齢者や社会的弱者に優しいと同時に、一般の人にも使いやすい技術です。創造性が豊かで、個々の特性に応じる多様性が求められる社会において、技術によって人間が疎外されることなく、一人一人の能力を十分に活かすことのできる技術が求められています。

課題1

人間が疎外されるとは、具体的にどういうことですか。

た。抜明書、専門
る。
、高齢者や障害者、使えない

2. これから期待される技術と社会への影響

さらに安全で快適な社会を実現するため、ロボット開発などの制御に関連する技術への期待が高まっています。

課題2

これから期待される制御に関連する技術は、どのようなものがあるのでしょうか。

を軽減するロボット（介護、福祉、農業など）
の通信を、遠隔地から行うことができる。

3. 技術の発展と自分の将来の生き方

ロボットが人に代わって仕事をすることがますます増えています。そして、ロボットがどんどん進化していけばいくほど、『人ならではの』仕事の価値が高まっていくといわれています。これからの時代を生き抜くためには、「自ら考える」「自ら発言する」「自ら行動する」「自ら反省する」という人間固有の能力を磨くことがより大切になります。

課題3

上の文章を読んで、自分のこれからの生き方を考えてみよう。

目 次

第1章 制御学習の基礎 1

1. 制御とは	1	3. 問題解決の方法	3
2. 制御システム	1	4. プログラムとプログラミング言語	5

第2章 MICRO:BIT を動かそう 6

1. マイコンと micro:bit	6	6. 変数を使ってみよう	15
2. JavaScript ブロックエディター	7	7. 繰り返しを使う	17
3. micro:bit を使ってみよう	9	8. 分岐処理を使う	18
4. LED 画面を制御しよう	11	9. センサで制御する	20
5. ボタンAとボタンBで制御しよう	14	10. 無線通信機能を使って制御する	25

第3章 MICRO:BIT と制御回路 28

1. I o T	28	5. ブレッドボード	30
2. 制御回路の基礎	28	6. 簡単な回路学習	31
3. 発光ダイオード	29	7. micro:bit でLEDを点灯する。	33
4. 抵抗器	29		

第4章 制御学習の基礎 36

1. マイコンと micro:bit	36	7. 同じ処理をまとめる	42
2. 制御実験の準備	36	8. 音を鳴らす	43
3. LED の点滅制御	37	9. 無線を使う	45
4. 条件によって処理内容を変える	39	10. 歩行者用信号機	47
5. データを一時的に保管する	40	11. 交差点を実現しよう 《超難問》	48
6. 途中に入り込む処理	41		

第5章 学習のまとめ 50

1. 人に優しい技術	50
2. これから期待される技術と社会への影響	50
3. 技術の発展と自分の将来の生き方	50

編集：三田市中学校・技術家庭科部会

プログラムによる計測・制御を学ぼう
micro:bit 版

著作：浅田寿展

発行者：JA 教育研究会

【本書の無断転載を禁ず】