

ぐんまデジタルイノベーションチャレンジ

地域ICTクラブ -micro:bit(1)-

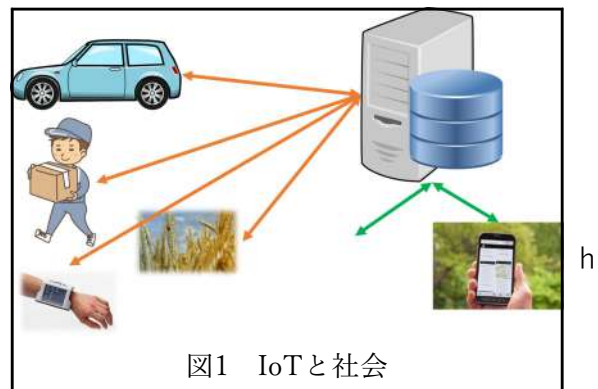
担当：茂木 和弘



1. 組み込みシステムは多くの場所で求められている

1.1. IoT

皆さんの生活の中に、たくさんのコンピュータが利用されています。少ない人や機械で、今以上にモノを作ったり配達するため、コンピュータやセンサを利用した社会作りが進んでいます。「IoT (Internet of Things)」という言葉を見たことがありますか？ IoTを使用することにより、工場の機械、自動車、荷物、商品、植物や動物、人などの状態をセンサで取得し、このデータをネットワークを介して集めて、今以上に上手に機械や人々を活用することを目指しています。



このIoTを使用するには、①機器などの状態を取得するためのセンサと制御用のコンピュータ、②センサからのデータをインターネット上のコンピュータに送信（受信も）するためのネットワーク、③データを集約するためのコンピュータ（サーバと呼んでいます）、そして④集約されたデータを見たり操作するための端末（コンピュータ、スマートフォン、タブレット等）が必要になります。①の「センサとコンピュータ」で使用されるコンピュータは、皆さんが使用しているデスクトップコンピュータやラップトップコンピュータ（クロームブック等）のように文章やスライドを作成できるようになる高機能なコンピュータではありません。例えば、温度センサからのデータ（温度）に応じてスイッチを入れたり切ったりすることのようなある決まった（簡単な）仕事のみを着実に実行することが小型で安価なコンピュータが使用されています。このコンピュータで使用されているのが、「組み込みシステム」と呼ばれるコンピュータシステムです。ここでのコンピュータシステムは、

主に制御を行うシステムを指しています。

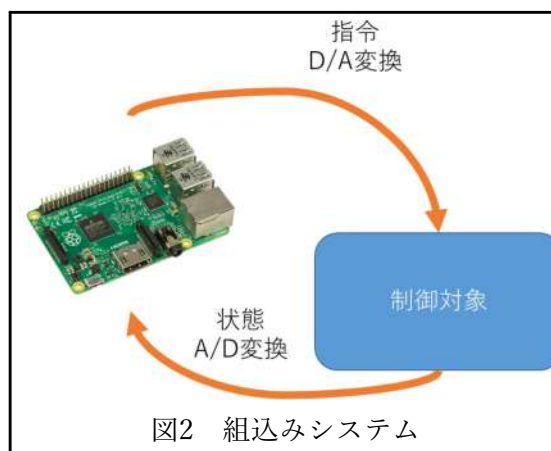
ここでは、micro:bitを使用して組み込みシステムでどのようなことができるかを学びます。また、温度、光、傾き等を検出するセンサからの情報（データ）を、micro:bitに取り込み、そのデータによってLEDを光らせたり、モータを動かすなどしてみます。これらを学んだまとめてして、小型の自動車（マイクロビットカー）を自動走行させます。自動車の自動運転まではできませんが、床に描かれた線をセンサが検出して、進行方向を自動で決めるプログラムを作成します。「えっ！難しそうだな」と思うかもしれませんが、どのようにすればこのような走行ができるようになるかをゆっくりと説明しながら進めますので、ご安心ください。

1.2. 組み込みシステム

組み込みシステムは、「特定の機能」を実現するために機械や装置等に組み込まれるコンピュータシステムである。組み込みシステムに求められる主な特性には以下があります。

- 専用化されたシステムである
- リソース制約が厳しい
- 高い信頼性が必要である
- リアルタイム性が必要である

モータ等の制御対象（どのように動かし



たいか）から回転数などの状態を信号で取得し、この信号に対して適当な動作信号をモータなどに送り、希望した動作をさせる。データが電圧や電流等のアナログ信号である場合は、コンピュータで処理できるデジタル信号に変換する。コンピュータは、基本的で0と1のデジタル信号で計算をするため、デジタル信号で対象となる機器とデータのやりとりが求められる。このため、アナログ信号をデジタル信号に変換するA/D変換器、逆にデジタル信号をアナログ信号に変換するD/A変換器が用いられることもある。最近では、直接デジタル信号で通信できる機器も増えている。スマートファクトリーの用に、工場全体をコンピュータシステムで管理するにはこのような最新の機器が求められるが、まだまだ古い機器が使用されているためデータの取得に大きな問題が残っている。

1.3. センサ

温度や明るさをコンピュータで利用するには、温度や明るさを電気信号に変換し、最終的にはデジタル信号に変換します。この時期、快適な空間を得るためエアコンが大活躍しています。設定した温度になるように、エアコンは涼しい風や暖かい風を送り出しています。そうですね、ここに温度センサが利用されています。室温を温度センサで検知し、設

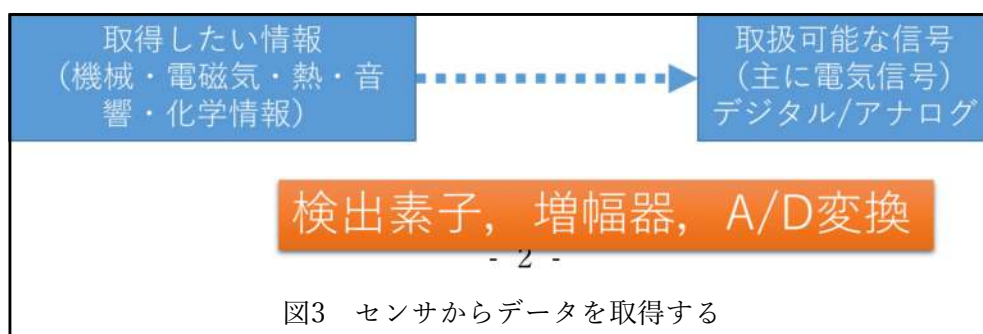


図3 センサからデータを取得する

定温度になるようにエアコンは仕事をしています。ここでも組込みシステムが使用されています。

自動運転をする自動車は、一般的な自動車に搭載されているエンジン、ブレーキ、エアコン、カーナビ用のコンピュータ以外にも、周囲の状況を認識するシステム、自動走行をするための制御システムが搭載されています。これらは組込みシステムというより高性能なコンピュータに分類されるかもしれません。センサも、カメラやLiDARと呼ばれる高性能はセンサが使用されています。

1.4. 効率的なプログラミング勉強法

上手にモータ等の機器を動作（制御）させるには、モータを制御する「プログラム」を上手に組み立てなければなりません。では、どのようにすれば上手はプログラムを作成することができるようになるのでしょうか。難しい問題ですね。プログラムはコンピュータが理解できる命令を組み合わせて作られます。実は、人に人文の考えを伝えるのととても似ています。話す相手がコンピュータか人の違いだけです。自分の考えや、お願いしたいことを上手に伝えることができるようになると、プログラムも上手になります。（と、感じています。）このため、国語や算数はとても重要です。順序よく手順を考えることができるようになると、プログラムの流れをまとめることができるようになります。

組込みシステム以外でも、問題を解決するためにプログラムを作成することがあります。大きなプログラムを作成する場合は、ゼロからプログラムを作成するのは困難であるため、ライブラリやAPIと呼ばれるプログラムを使用します。以下に、プログラム勉強法の1つを示します。参考にしてください。

- 1 作りたいプログラムを考える
- 2 ハードウェアを用意する（パソコン、マイコン等）
- 3 プログラミング言語を決める
- 4 開発環境やライブラリ、API (Application Programming Interface) を整える
- 5 プログラムを作成する
- 6 プログラミング言語を身に着ける（覚えるのではない）
- 7 開発環境になれる
- 8 ライブラリ、APIを活用する
- 9 動作する小さいプログラムからはじめる
- 10 慣れてきたら、アルゴリズムやデータ構造に興味をもとう

前置きはこれくらいにして、お待ちかねのmicro:bitを楽しみましょう。

2. micro:bitを使おう

micro:bitでプログラムの勉強（遊ぶ）をするには、パソコン(Windows, mac os X, linu

x, …), Webブラウザ (google chrome, Edge,…) , BBC micro:bit (3千円くらい), パソコンとmicro:bitを接続するためのマイクロUSBケーブル (数百円) が必要になります. iPadやBluetoothを利用できるパソコンであれば, micro:bitとmicro:bitを動かすための電源 (電池, 電池ボックス, 接続ケーブル) が必要になります.



2.1. micro:bitとは

micro:bitは, 英国放送協会(BBC)が開発し, 英国の11歳と12歳の小学生全員に配布されました. 2020年までに日本の子どもたち30万人へmicro:bitを届ける計画ありましたが, 実際には配布はされませんでした. とても残念でした.

micro:bitは, プログラミング教育用シングルボードコンピュータで以下のような特徴があります.

- プログラム (命令) を処理するCPUコア
- プログラム (命令とデータ) を格納するメモリ
- タイマー
- 外部 (周辺機器など) と通信するための入出力
- 2個のボタンスイッチ, 加速度センサ, 磁力センサ, 25個の赤色LED
- Bluetooth搭載

開発環境がウェブブラウザ上で動作するため, ソフトウェアのインストールが必要あり

ません。もちろん無料で開発環境を利用できます。ウェブ無頼座で作成したプログラムは、

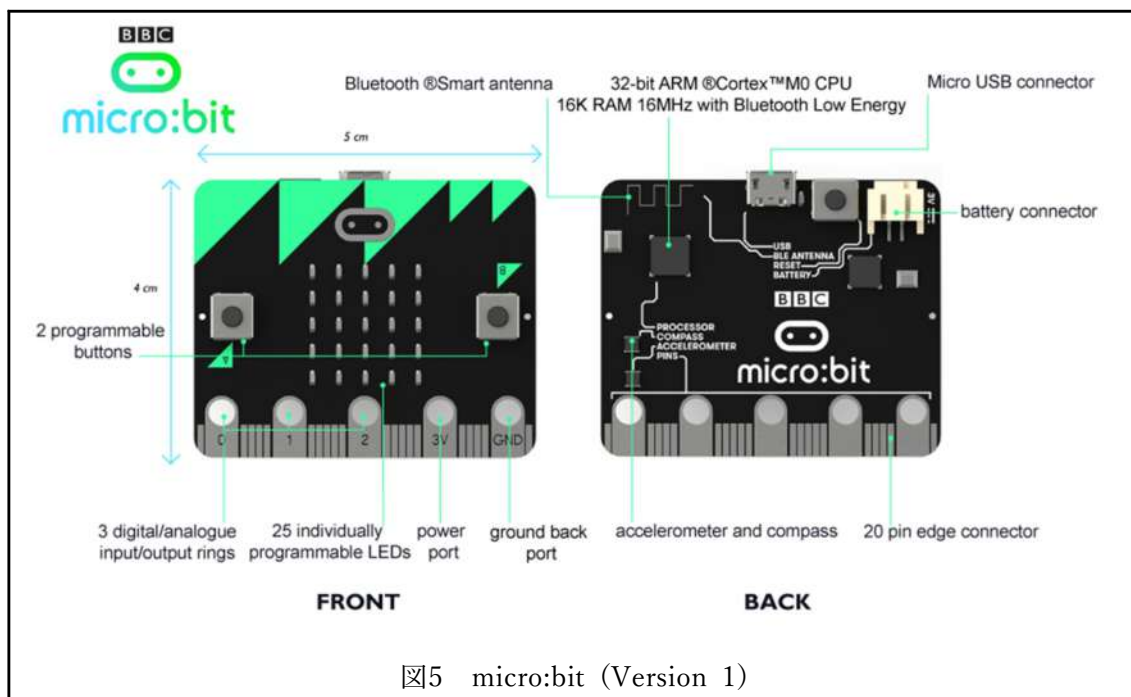


図5 micro:bit (Version 1)

ローカルのコンピュータにダウンロードし、USB経由でmicro:bitに書き込むだけで動作します。単に、プログラムファイル（micro:bit上で動作するように変換されています）をmicro:bitにコピーするだけです。micro:bitはUSBフラッシュメモリの用に扱えます。ファイルのコピーが完了した瞬間に、プログラムの実行が始まります。このため、スタートボタンやストップボタンはありません。プログラムを最初から実行したい場合は、リセット（Reset）ボタンを押してください。



図6 リセットボタン

2.2. micro:bitにプログラムを送信する準備をしよう

2.2.1. パソコン（デスクトップ、ラップトップ）とUSBケーブルで接続

USBケーブルのコネクタ形状に注意してmicro:bitとパソコンを接続しましょう。小さいコネクタの方をmicro:bitに接続します。micro:bitへの電気はUSBケーブルから給電される

ため、電池ボックスを接続する必要はありません。



図7 micro:bitとPCをUSBケーブルで接続する

2.2.2. iPad等とBluetoothで 接続する

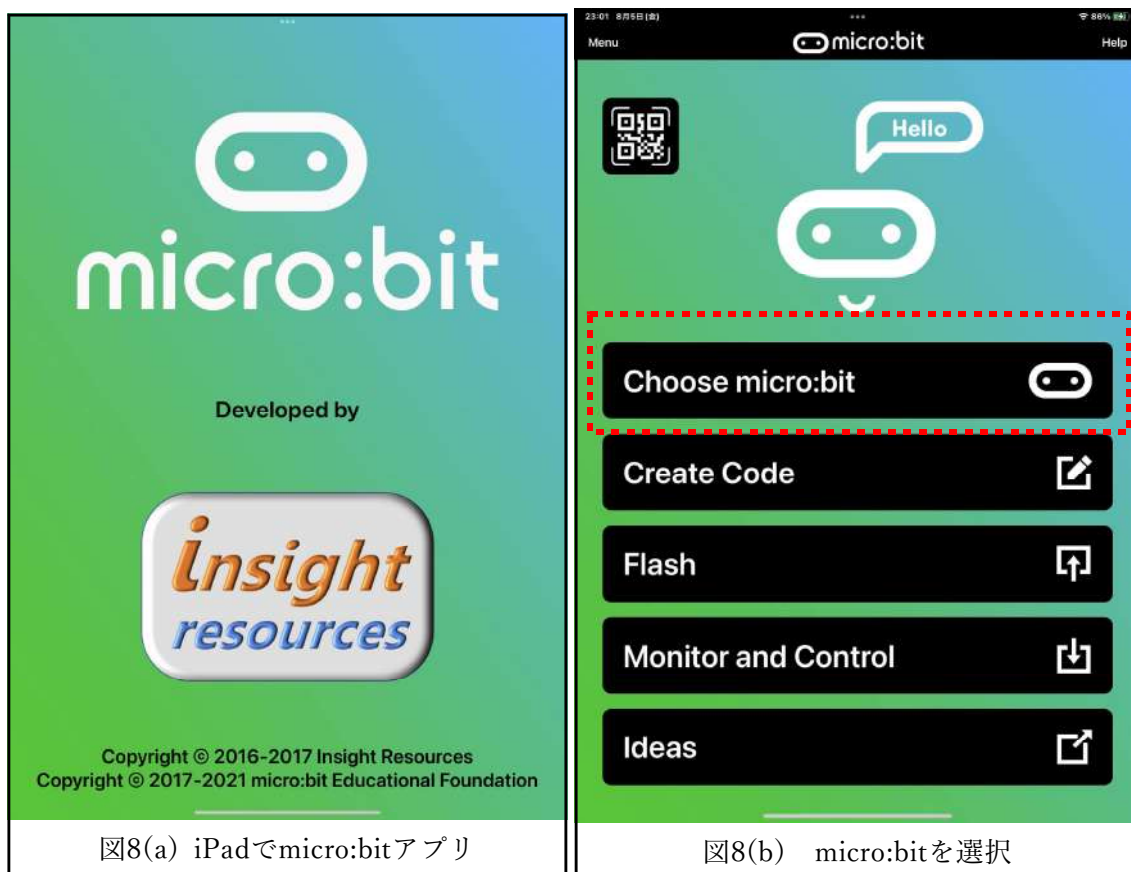


図8(a) iPadでmicro:bitアプリ

図8(b) micro:bitを選択

パソコンを使用する場合はmicro:bitへはUSBケーブルで電気が供給されます。しかし、Bluetoothでmicro:bitと接続する場合は電池ボックスから電気を供給する必要があります。電池を電池ボックスにいれ、電池ボックスをmicro:bitに接続してからペアリングの作業を行ってください。

iPad(macOS)はmicro:bitのアプリをダウンロードします。ダウンロードしたアプリを起

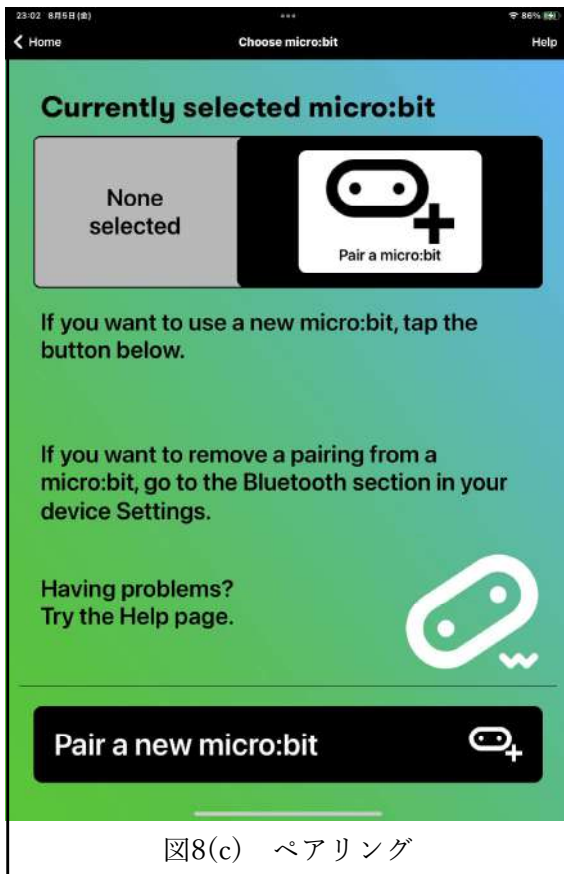


図8(c) ペアリング

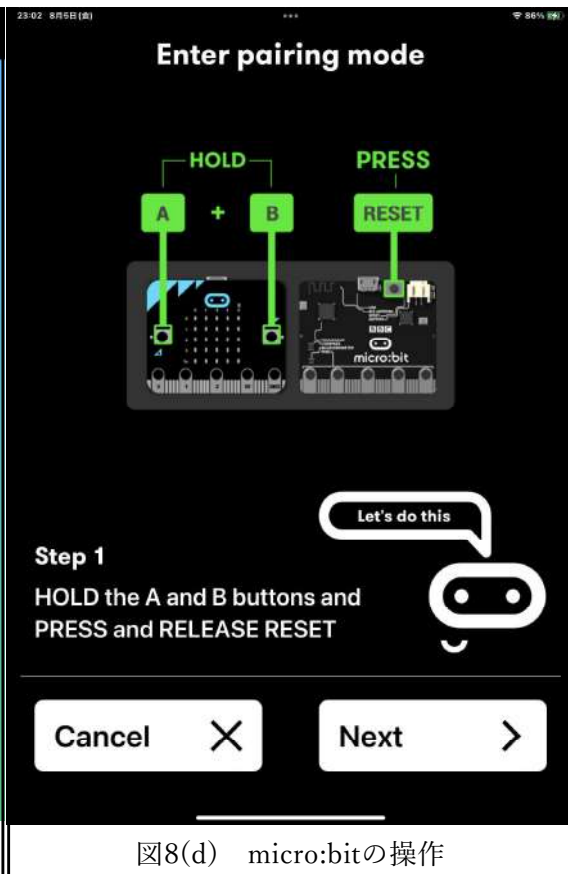
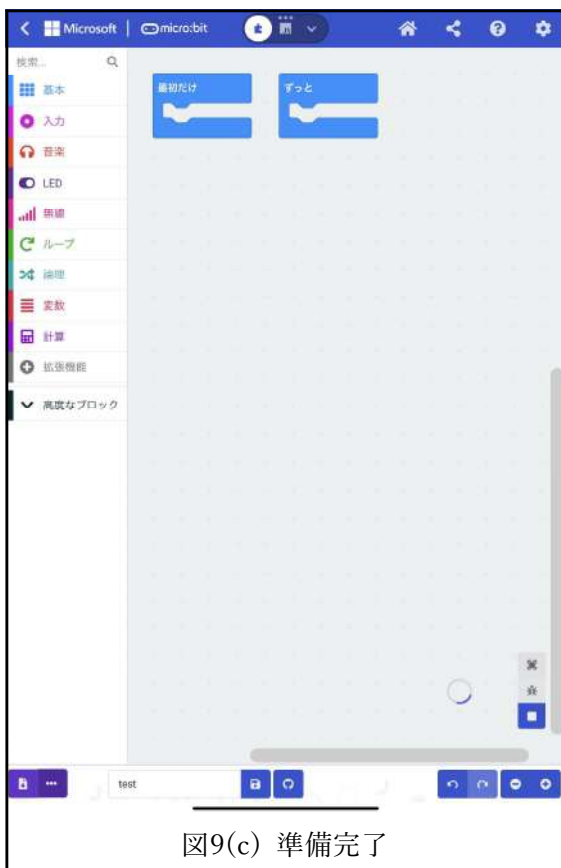
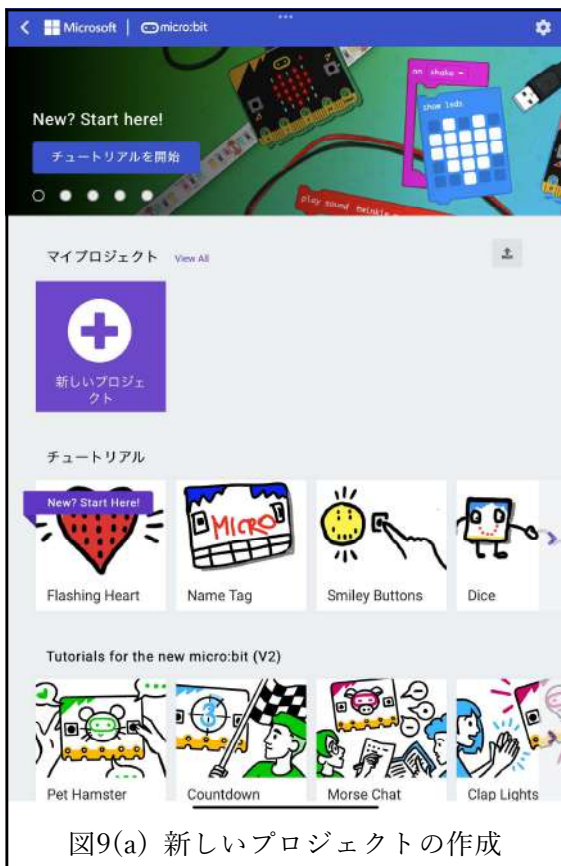


図8(d) micro:bitの操作

動し (図8(a)), 少し待つとmicro:bitを選択する画面 (図8(b)) が表示されます。「Choose micro:bit」をタップして、ペアリングの画面 (図8(c)) を表示します。画面下の「Pair a new micro:bit」をタップしてmicro:bitとペアリングを行います。表示される画面 (図8(d)) に従って、micro:bitを操作してください。うまくペアリングできないときは、何度か試してみましょう。他の人と同時にペアリング作業を行うと、間違ったペアリングが行われるかもしれませんので注意してください。

ペアリングが完了したらホーム画面 (図8(a)) の「Create Code」からプロジェクトの作成に進みます。プロジェクトの名前を入力してください。図9(b)はプロジェクト名として「test」を入力しています。この名前がファイルがダウンロードされます。プロジェクト名でトラブルが発生しないように、プロジェクト名は英字 (aからz) と数字 (0から9) を組み合わせて作成してください。日本語や記号を用いても動作するかもしれませんが、保証できません。プロジェクト名を入力して「作成」ボタンをタップすると、いよいよプログラムの作成画面 (図9(c)) が表示されます。

ここからは、パソコンと同じように操作が行えます。パソコンでは作成したプログラムをダウンロードして、そのファイルをmicro:bitにコピーします。Bluetoothでmicro:bitと接続している場合は、ダウンロードで直接micro:bitにファイルが転送されます。



2.2.3. Webブラウザでプログラミング環境を開く

まず、ChromeやEdge等のWebブラウザを起動して下さい。次に、アドレスバーにURLとして「<https://makecode.microbit.org/>」を入力してEnterキーを押します。または、「Microsoft MakeCode for micro:bit」と入力して検索します。表示される検索結果の(たぶん)一番上に上記URLが表示されます。ここをクリックしてください。

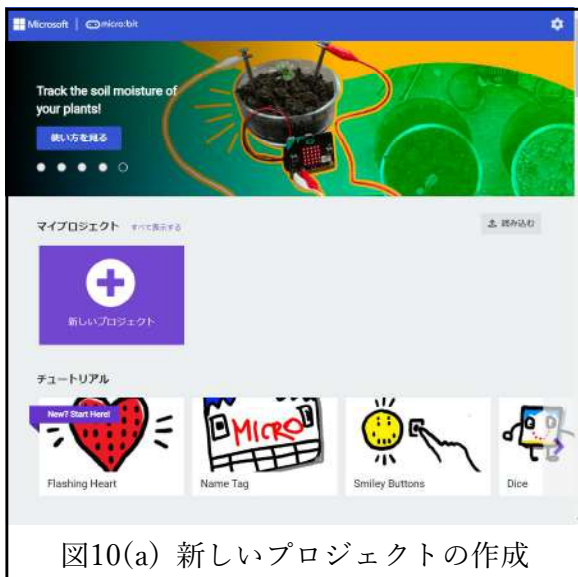


図10(a) 新しいプロジェクトの作成



図10(b) プロジェクト名を入力

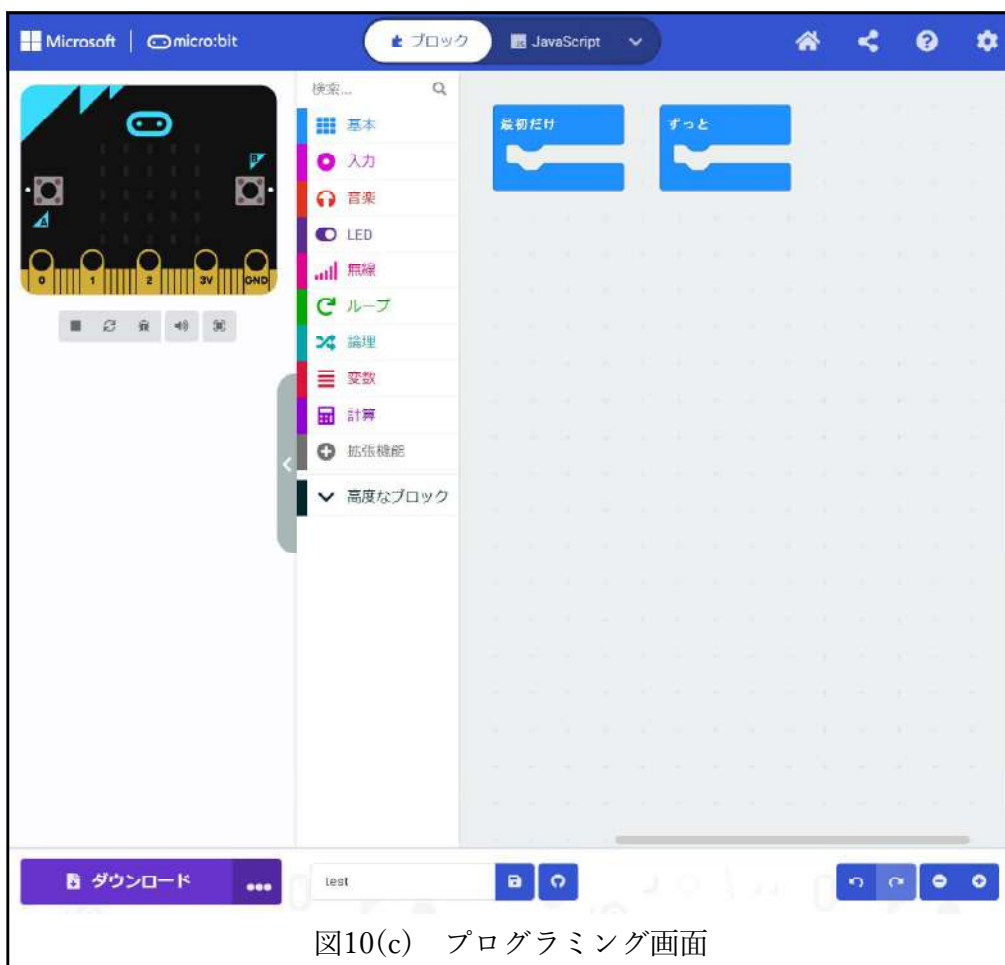
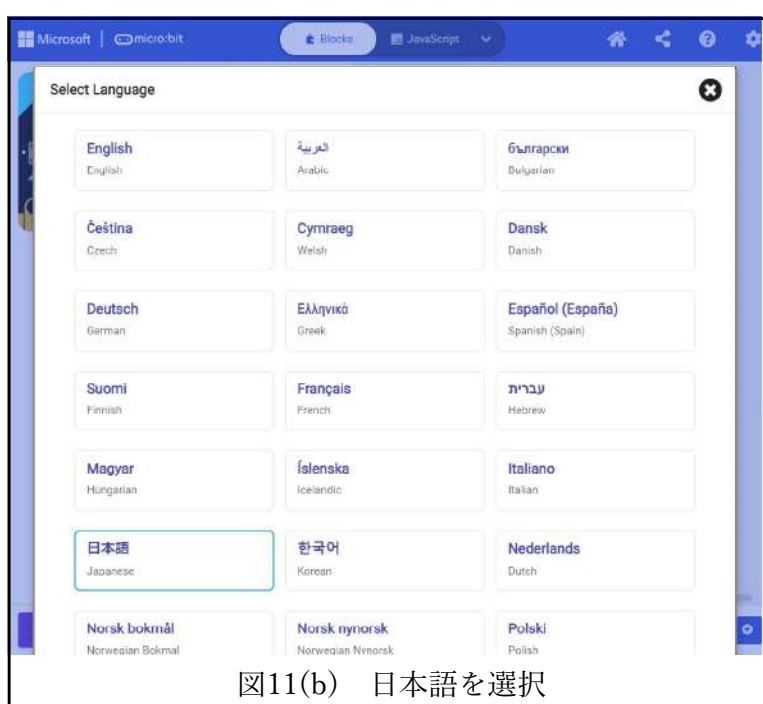
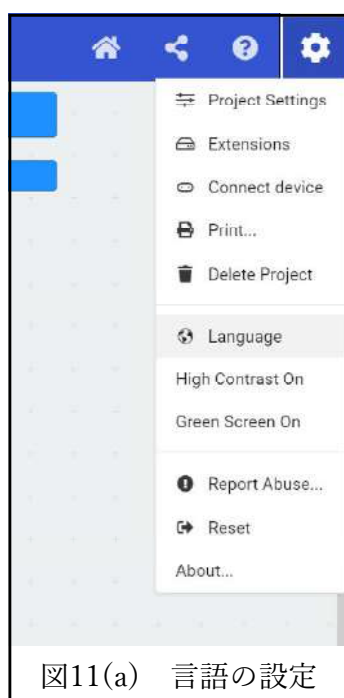


図10(c) プログラミング画面

表示された画面（図10(a)）から新しいプロジェクトの作成に進みます。「新しいプロジェクト」ボタンをクリックし、表示される画面でプロジェクトの名前を入力してください。図10(b)はプロジェクト名として「test」を入力しています。この名前がファイルがダウンロードされます。プロジェクト名でトラブルが発生しないように、プロジェクト名は英字（aからz）と数字（0から9）を組み合わせて作成してください。日本語や記号を用いても動作するかもしれませんが、保証できません。プロジェクト名を入力して「作成」ボタンをタップすると、いよいよプログラムの作成画面（図10(c)）が表示されます。

画面の文字が英語（English）になっている場合は、画面右上の歯車アイコンをクリックして、設定メニューを表示します。設定メニューの中に「Language（言語）」をクリックして、表示される言語一覧から「日本語」を選択してください。もちろん、日本語以外を選択しても問題ありません。



3. micro:bitプログラミングを始めよう

3.1. プログラムの作成と実行手順

- (1) Webブラウザでプログラムを作成する..（図12(a)）
- (2) シミュレーターで動作確認を行う。（図12(a)）
- (3) 作成したプログラムをダウンロードする。
- (4) ダウンロードしたファイル（.hex）をmicro:bitへコピーする。（図12(b)）
- (5) プログラムの動作を確認する。



図12(a) コード作成



図12(b) ファイルのコピー

3.2. ブロックの操作

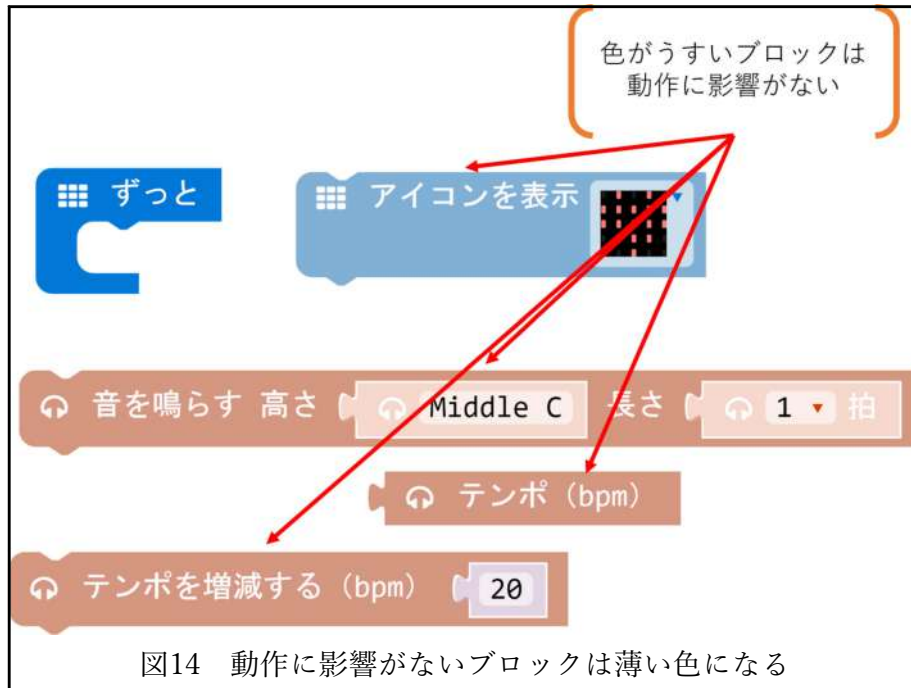
- ブロックの選択
 - ・ マウスで左クリック
- ブロックの移動
 - ・ マウスでドラッグ (左ボタンを押しながら移動する)
- ブロックを消す
 - (1) ブロックを選択し
 - (2) 右ボタンから「ブロックを削除する」を選択
 - (3) または、「Back Space」キーか「Delete」キーを押す
- ブロックを合体
 - (1) ブロックを移動してブロック同士を重ねると、合体できる位置が黄色く光る
 - (2) 合体位置が正しければ、左ボタンから指を離す
 - (3) 間違ったら、ブロックを移動させる
 - (4) より内側、より下のブロックから移動する
 - (5) ブロックの種類 (形) によって、合体できるブロック決まっている



図13 ブロックの合体

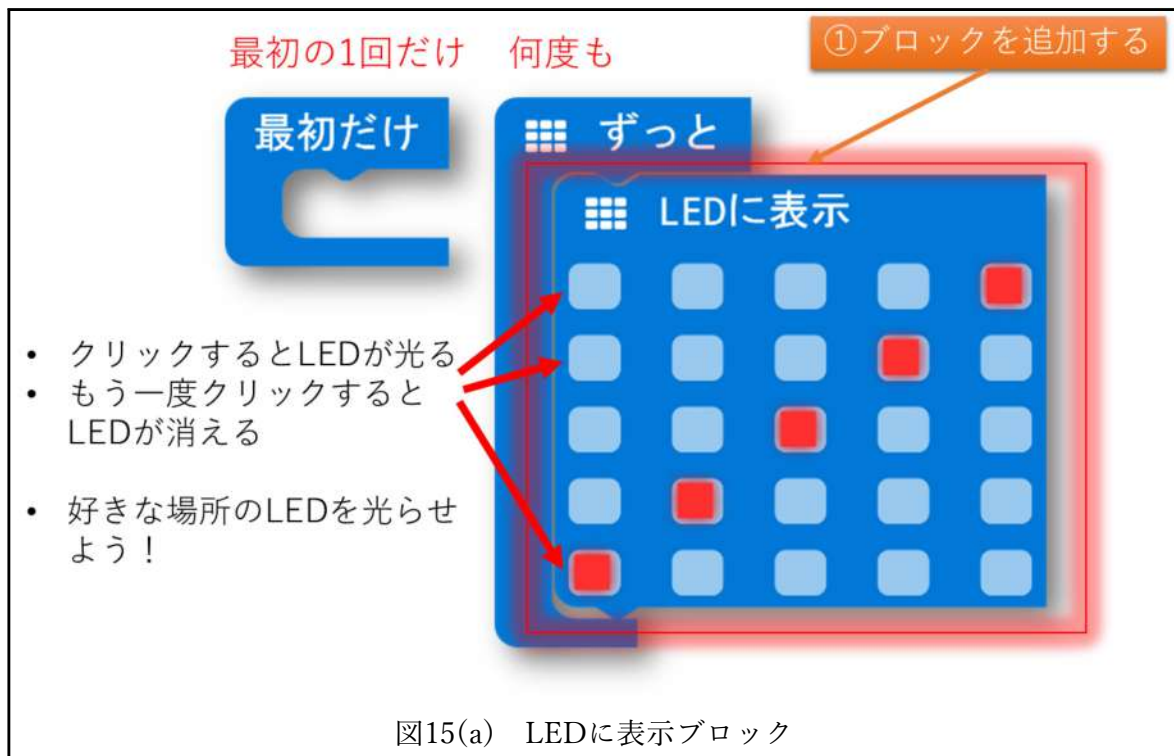
- ブロックのオプション
 - (1) 値 (数字や文字) を選べるブロックがある
 - (2) マウスの右ボタンで選べそうな場所を左クリックする

(3) キーボードから直接入力するか、マウスで選択する



3.3. 基本プログラムの作成

3.3.1. 赤色LEDを点ける (レッスン1)



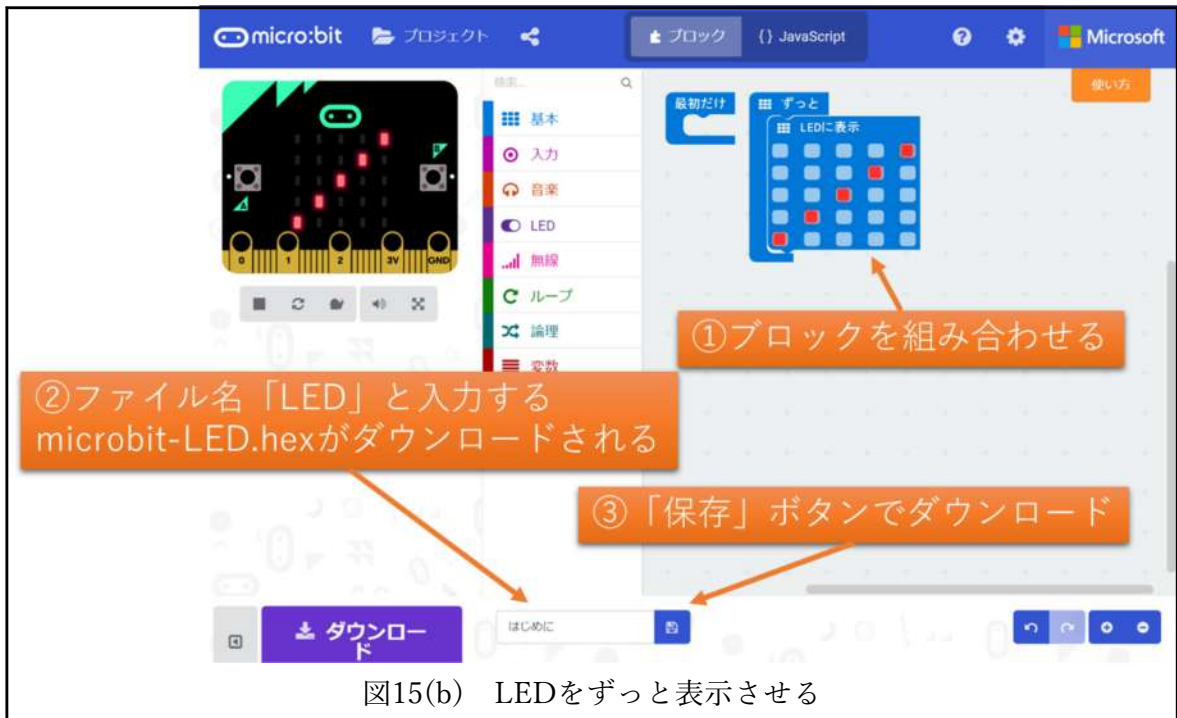


図15(b) LEDをずっと表示させる

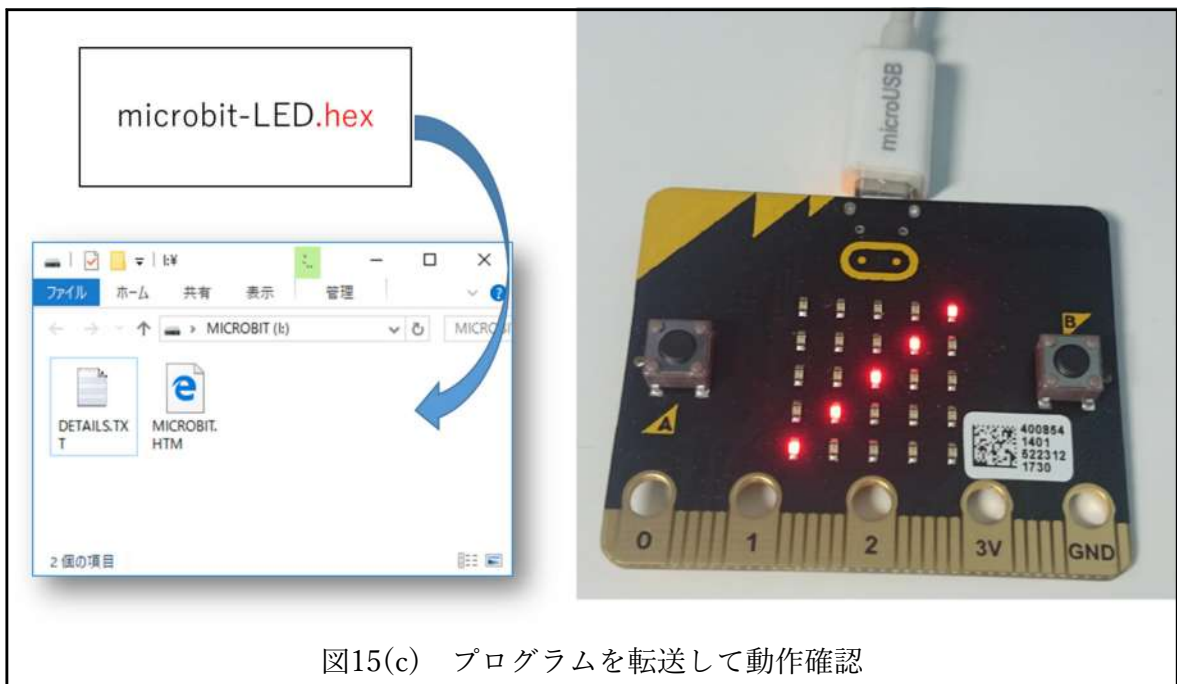


図15(c) プログラムを転送して動作確認

3.3.2. 文字をスクロール (レッスン2)

レッスン1で作成したプロジェクトを修正して次のプログラムを作成します。ブロック「LEDに表示」を削除して、同じところに「文字列を表示 “Hello!”」を追加する。デフォルトの文字列「Hello!」を「ABCDEFG」に変更する。micro:bit シミュレータで「A…G」が表示されるかを確認する。実際のmicro:bitで正しく表示されるかを確認する。



図16 LEDで「ABCDEFGG」を表示する

3.3.3. カウンター（変数の利用がポイント）（レッスン3）

①変数を作成する



図17(a) 変数を作成する

②ボタンAを押すと数が増える

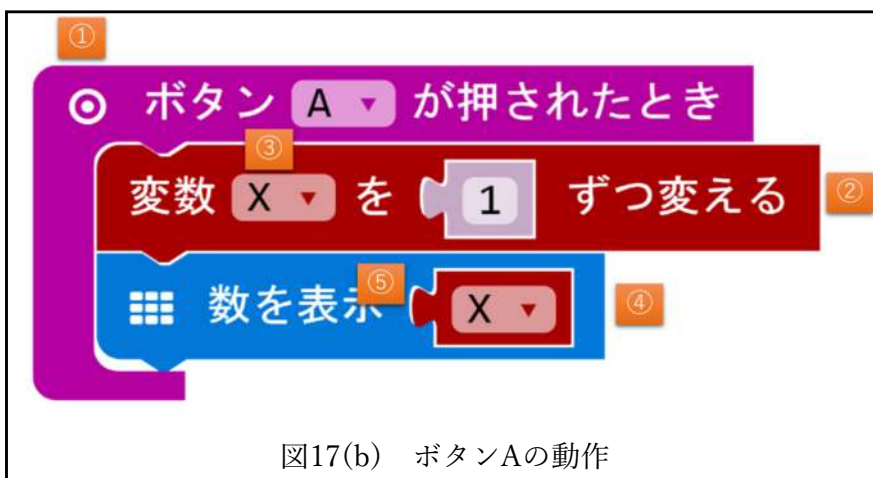


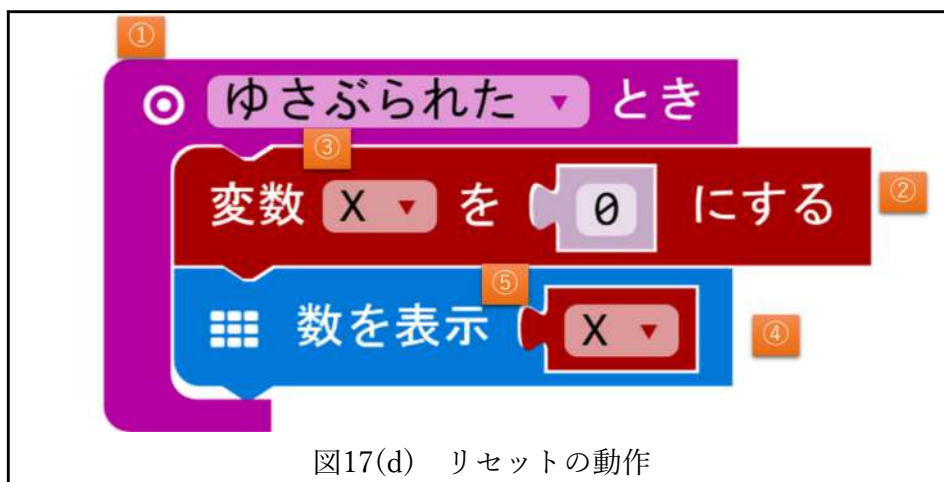
図17(b) ボタンAの動作

③ボタンBを押すと数が減る

ボタンBを押されたときのブロックを，ボタンAと同じように作成する．ボタンBが押されたとき，変数Xを「-1」することで数字が減少（9→8，0→-1等）する．変数Xを表示する．



④振るとリセット（0に戻る）



3.3.4. 繰り返しと場合分け（レッスン4）

①0から9まで繰り返して表示する



②偶数だけ表示する

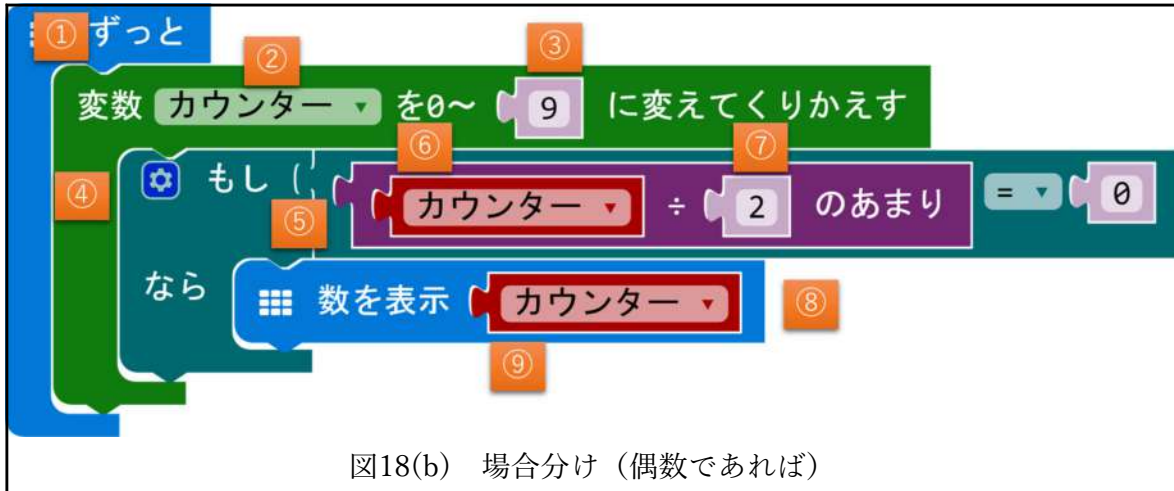


図18(b) 場合分け（偶数であれば）

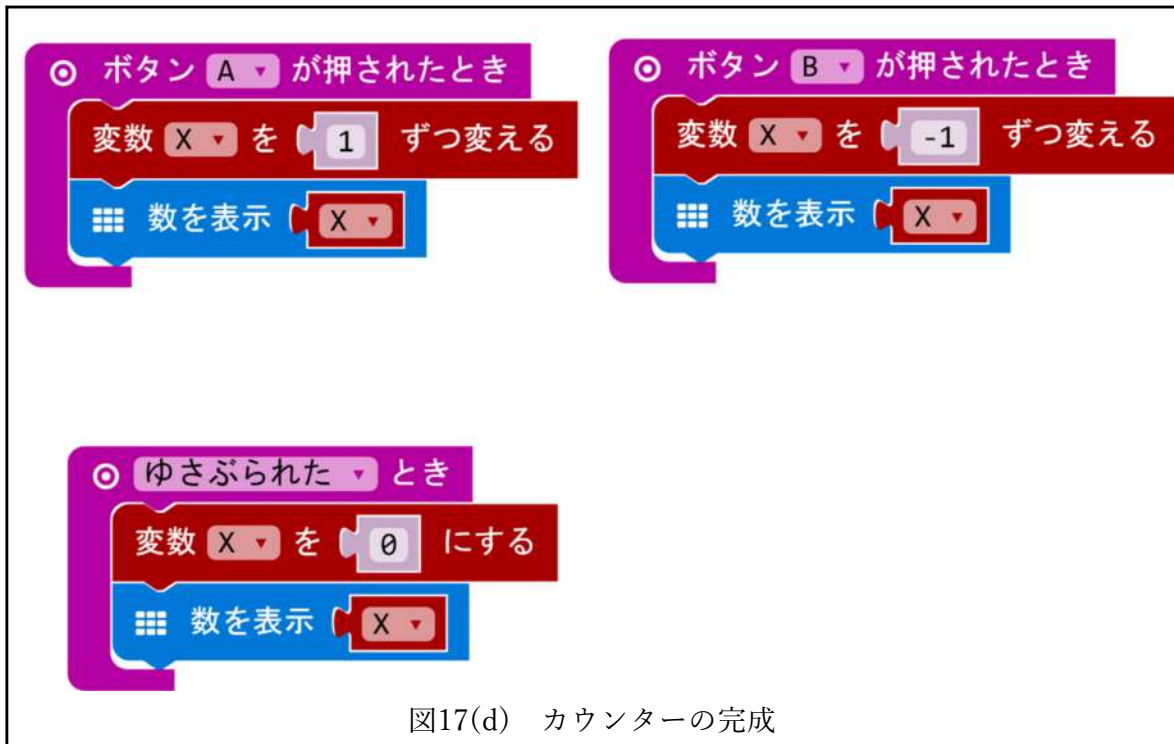


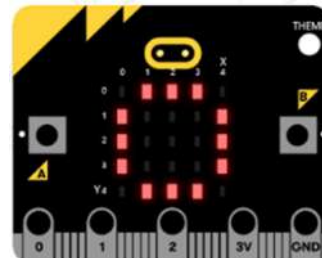
図17(d) カウンターの完成

3.4. センサを使用したプログラムの作成

3.4.1. コンパス (レッスン5)

①最初の1回だけ現在のコンパス値を補正する (キャリブレーション)

- (1) コンパスを使用するとき
- (2) LEDが1つ光る
- (3) micro:bitをかたむけると光る場所が移動する
- (4) 光る場所が、図の場所と同じになると、その場所は光り続ける
- (5) 図のように光るように調節する



②コンパスコンパス値を表示する

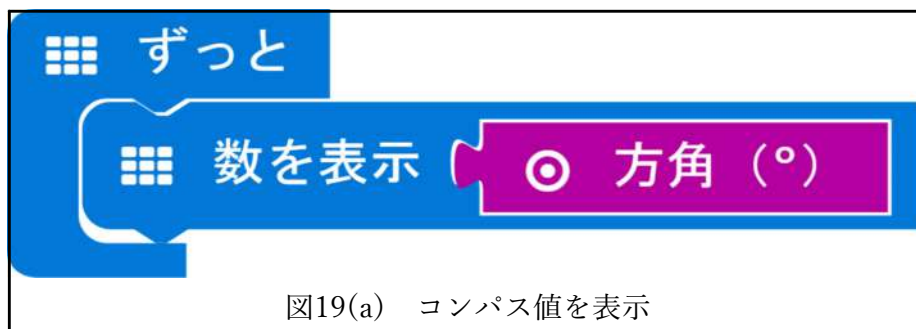


図19(a) コンパス値を表示

シミュレーターでは方角をマウスで変更する。



図19(b) シミュレーターでの操作

③方角が「北」ならば「N」を表示する.



図19(c) 北 (N)を表示する

④他の方角への対応 (場合分け) するため, 論理の「もし なら」を「もし なら でなければ」に変更する.

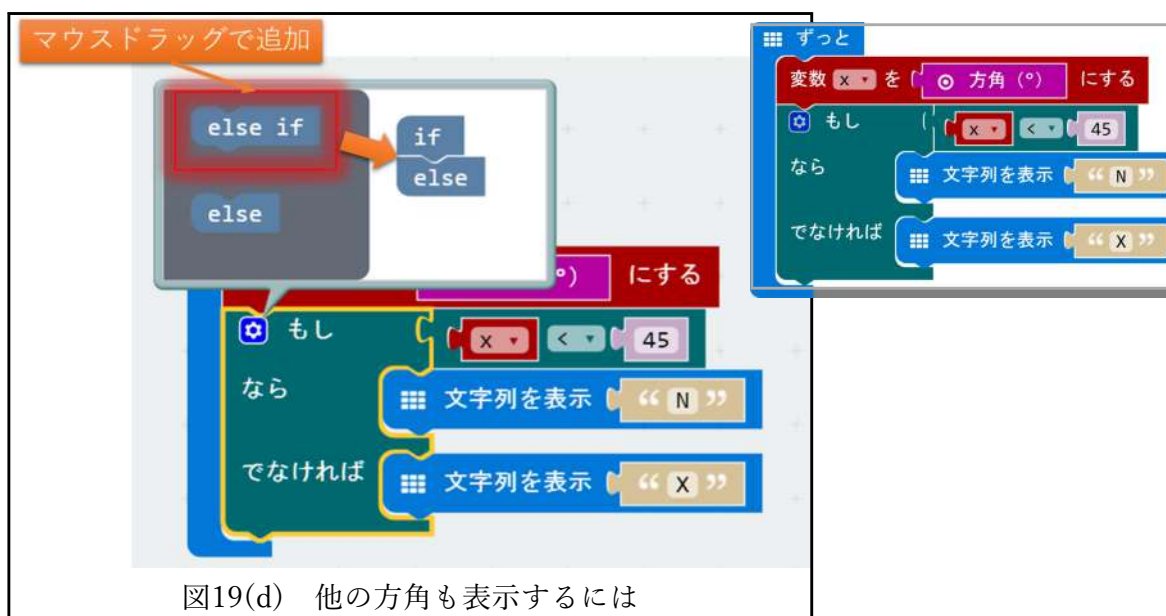


図19(d) 他の方角も表示するには

⑤すべての方角を表示する

The image shows a Scratch script and a compass rose. The script, enclosed in a 'ずっと' (forever) loop, sets a variable 'x' to '方角 (°)' (Direction (°)). It then uses a series of 'もし' (if) blocks to check the value of 'x' and display the corresponding direction: 'N' for 45, 'E' for 135, 'S' for 225, 'W' for 315, and 'N' for values greater than 315. To the right, a compass rose diagram shows the four cardinal directions (N, S, E, W) and the four intercardinal directions (NE, SE, SW, NW) with their respective angles: 45°, 135°, 225°, and 315°.

315° N 45°
W E
S
225°
315より大きい場合

図19(e) すべての方角を表示可能

3.4.2. 温度計 (レッスン6)

The image shows a Scratch script for a temperature gauge. It starts with a 'ゆさぶられた' (shaken) event trigger. The script then sets a variable 'x' to '温度 (°C)' (Temperature (°C)) and displays the value of 'x' as a number.

ゆさぶられた とき
変数 x を 温度 (°C) にする
数を表示 x

図20 温度計のプログラム

3.4.3. 照度計 (レッスン7)

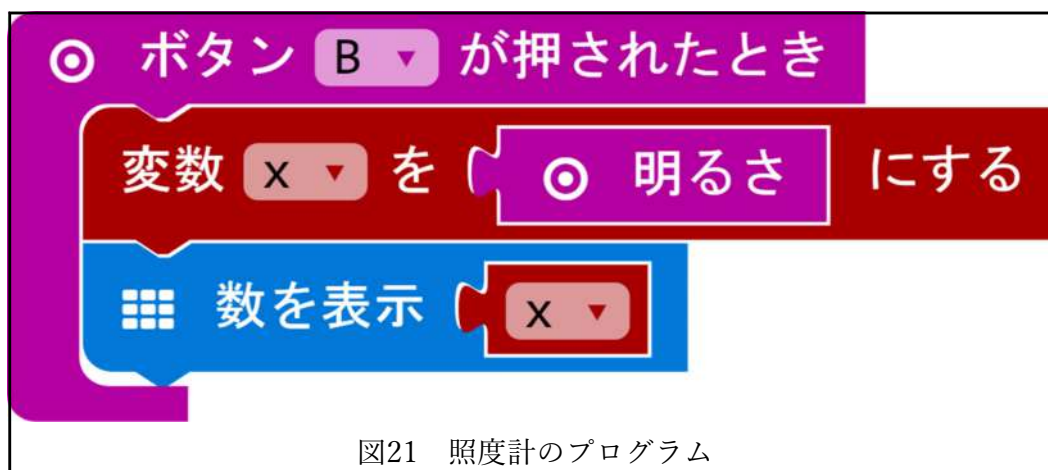


図21 照度計のプログラム