

次世代キッズ プログラミング教室 2023

第1回目：2023年11月12日（日）

micro:bitとプログラミングの基礎、センサーの利用、外部出力、変数

第2回目：2023年11月25日（土）

外部入力、工作のヒント、活用例

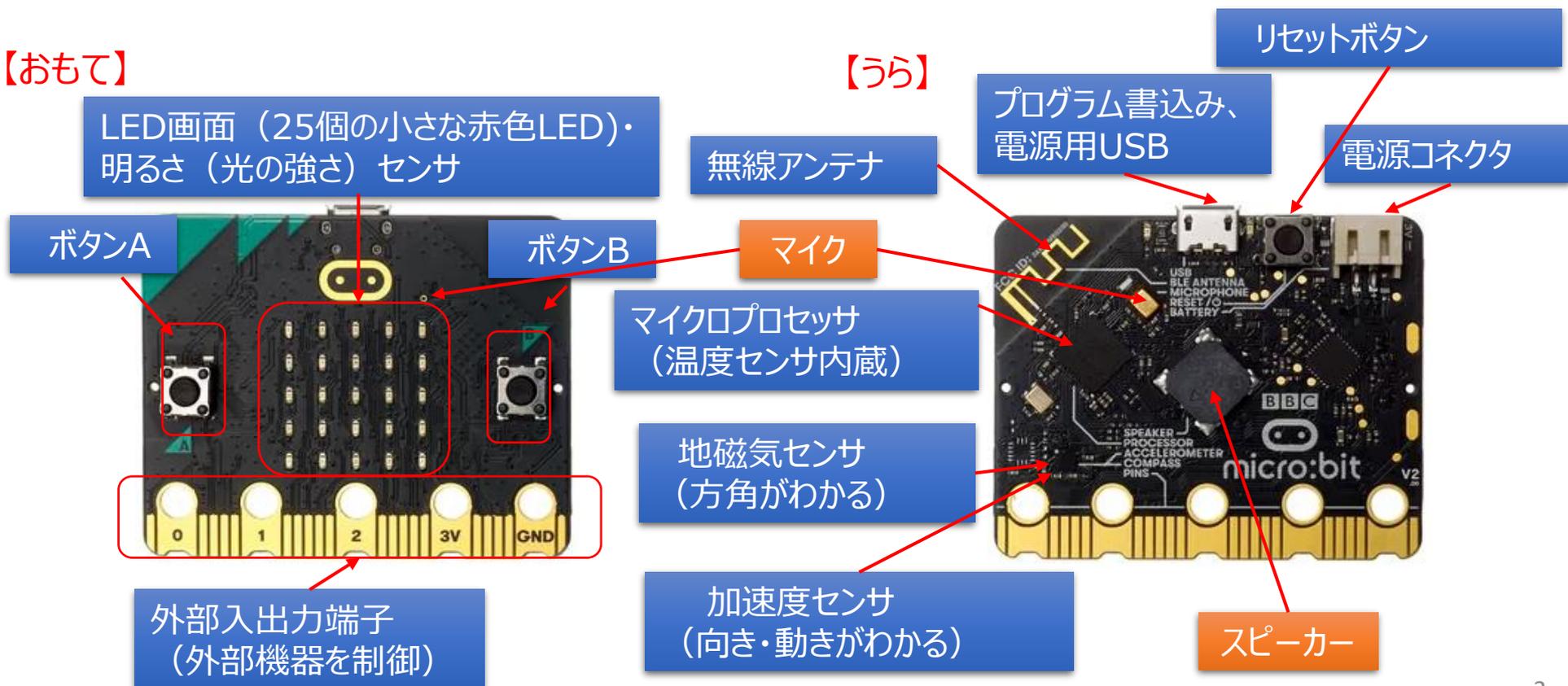
第3回目：2023年12月10日（日）

無線通信、作品発表、モーターと関数の利用

(はじめに)

micro:bit v2

- 2016年に、イギリス公共放送局(BBC)が中心となって開発
- LED、スイッチ、センサー、スピーカー、無線通信機能などを搭載
- プログラム学習のための**小型の教育用コンピューター**



micro:bitを使ってみよう！

1. 接続編

Youtube動画

「はじめてのマイクロビット講座 1.接続編」

<https://youtu.be/hhyLya92x2U>



2. プログラミング編

Youtube動画

「はじめてのマイクロビット講座 2.プログラミング編」

<https://youtu.be/CVYaBHGH9uc>

トム君のマイクロビット講座

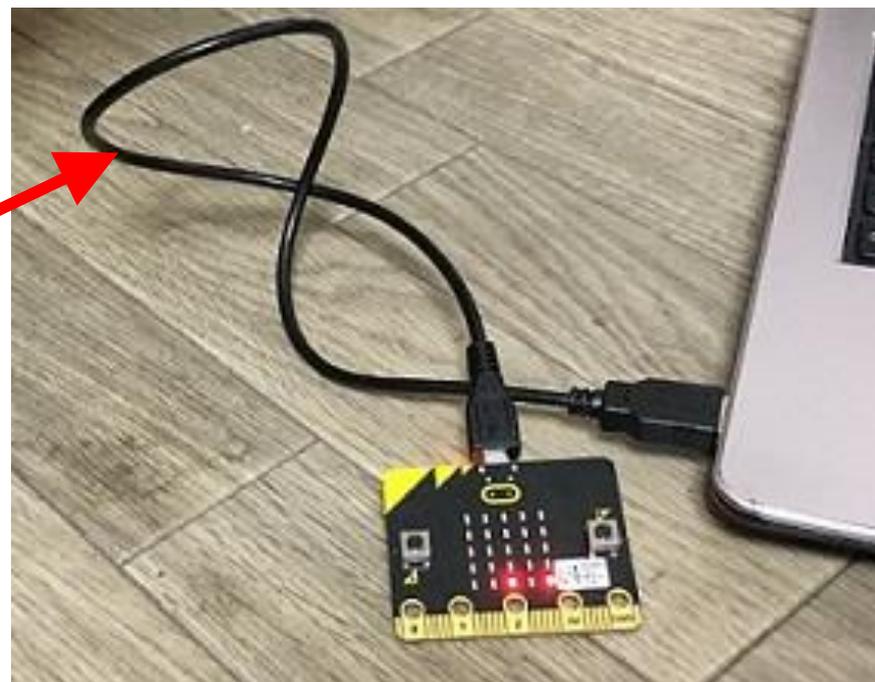
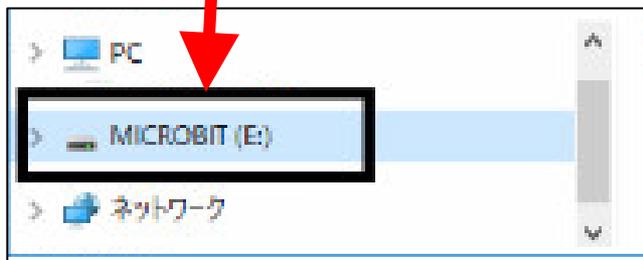
micro:bitをパソコンに接続しよう！

micro:bitをパソコンにつないで使うとき

プログラムするときは、
この形で使います。

①ケーブルでつなぎます

②「MICROBITドライブ」
ができます。



プログラミングの準備をしよう！ ①

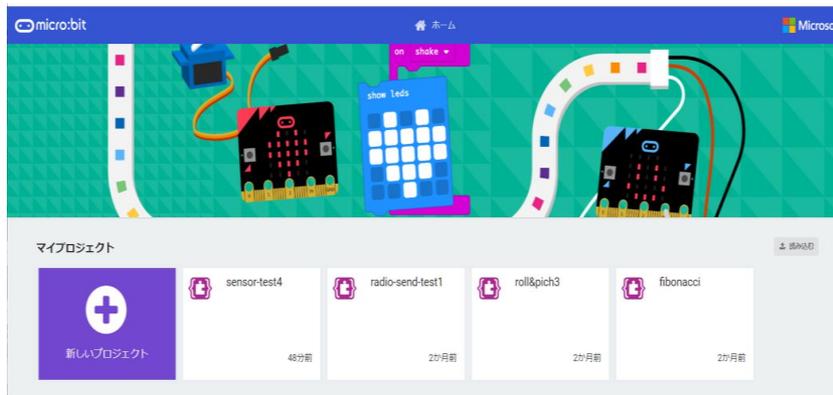
【準備：MakeCodeエディタを立ち上げます】

【パソコン（WindowsやMac）の場合】

ブラウザを開いて「makecode microbit」で検索

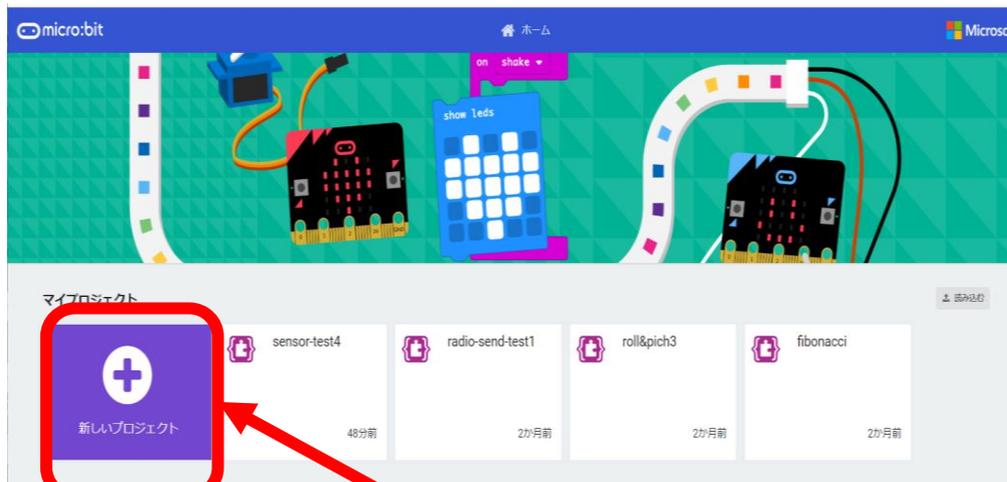
Microsoft MakeCode for micro:bit

A Blocks / JavaScript code editor for the **micro:bit** powered by Microsoft **MakeCode**.



「**MakeCode for micro:bit**」を
ダウンロードしインストールする方法もあります。

プログラミングの準備をしよう！ ②



ここをクリックし「新しいプロジェクト」を作ります

適当な名前をつけます

例： LESSEN1

あとで変えることができます

作成をクリック

【ブロック・プログラミング】

言語を使わず
ブロックを並べてプログラムを作ります。

ツールボックス

いろいろな命令ブロックメニュー
がそろっている

プログラミングエリア(ワークスペース)
命令ブロックをならべてプログラム
を組むところ

micro:bitシミュレータ(エミュレータ)
micro:bitの動作が確認できる

ダウンロードボタン

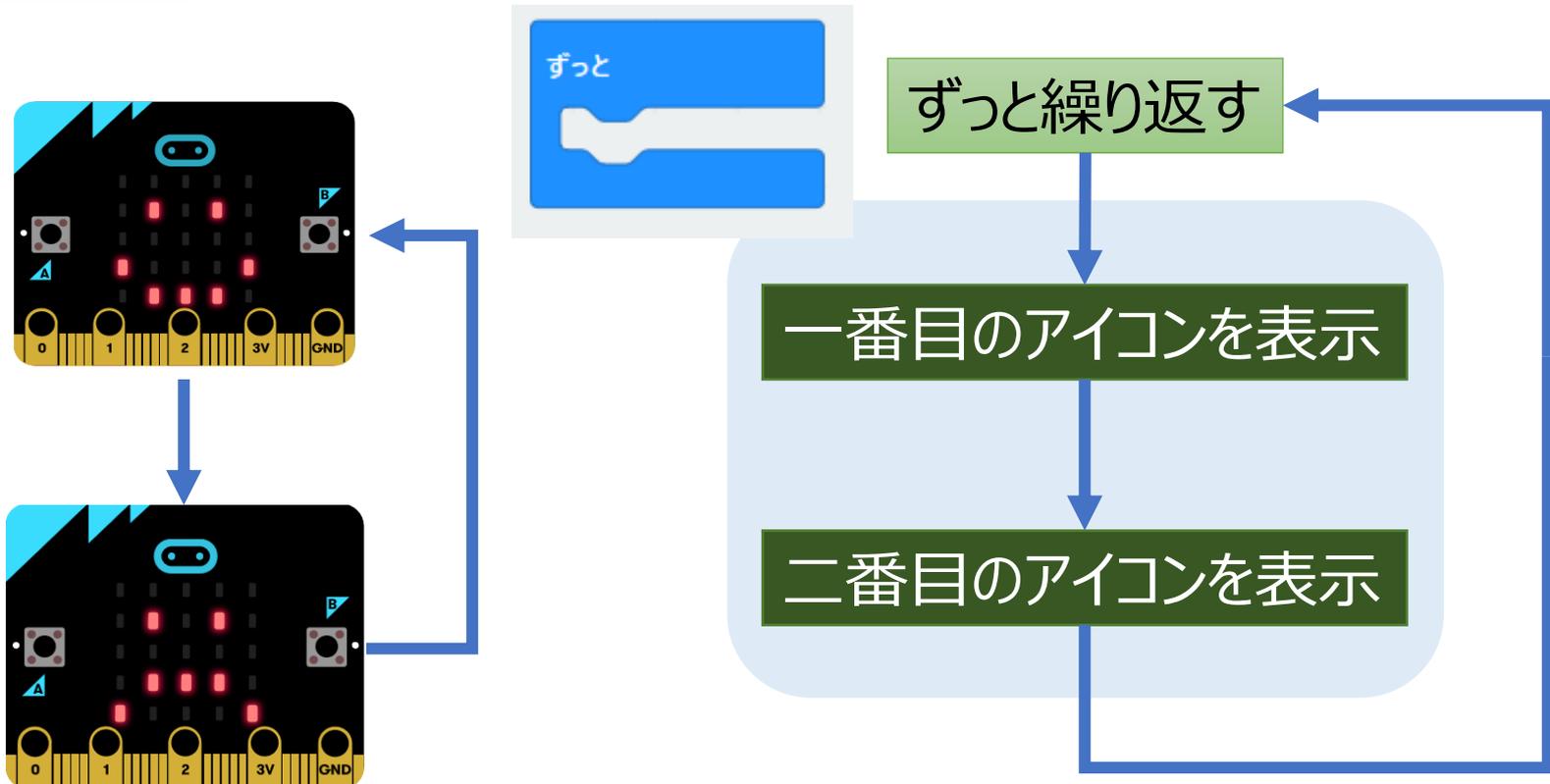
作ったプログラムをパソコンに保存します。



プログラミングを始めよう

目標 プログラムを作って、micro:bitを動かすまでを体験してみよう

やりたいこと 用意されている顔文字（アイコン）を、LED画面に交互に表示させる



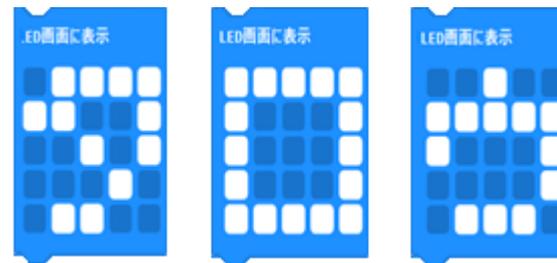
プログラミングを始めよう

LED画面への表示方法



①文字列や数を表示
※「文字列を表示」はアルファベット
しか表示できない。

②LED画面に表示
アイコンを自由にデザインできる。

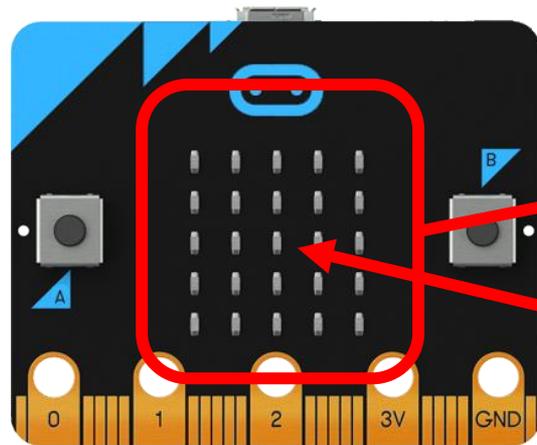


③一時停止 (表示時間を変える)
※「1000」→ 1秒停止

センサーを使ってみよう

目標 部屋などの明るさを測ってみよう

やりたいこと micro:bitのLED画面センサーがとらえた明るさを、LED画面に表示させる



LED画面
明るさセンサ

入力

出力

ずっと繰り返す

明るさセンサーで
明るさを測定

LED画面に
明るさを表示

センサーを使ってみよう

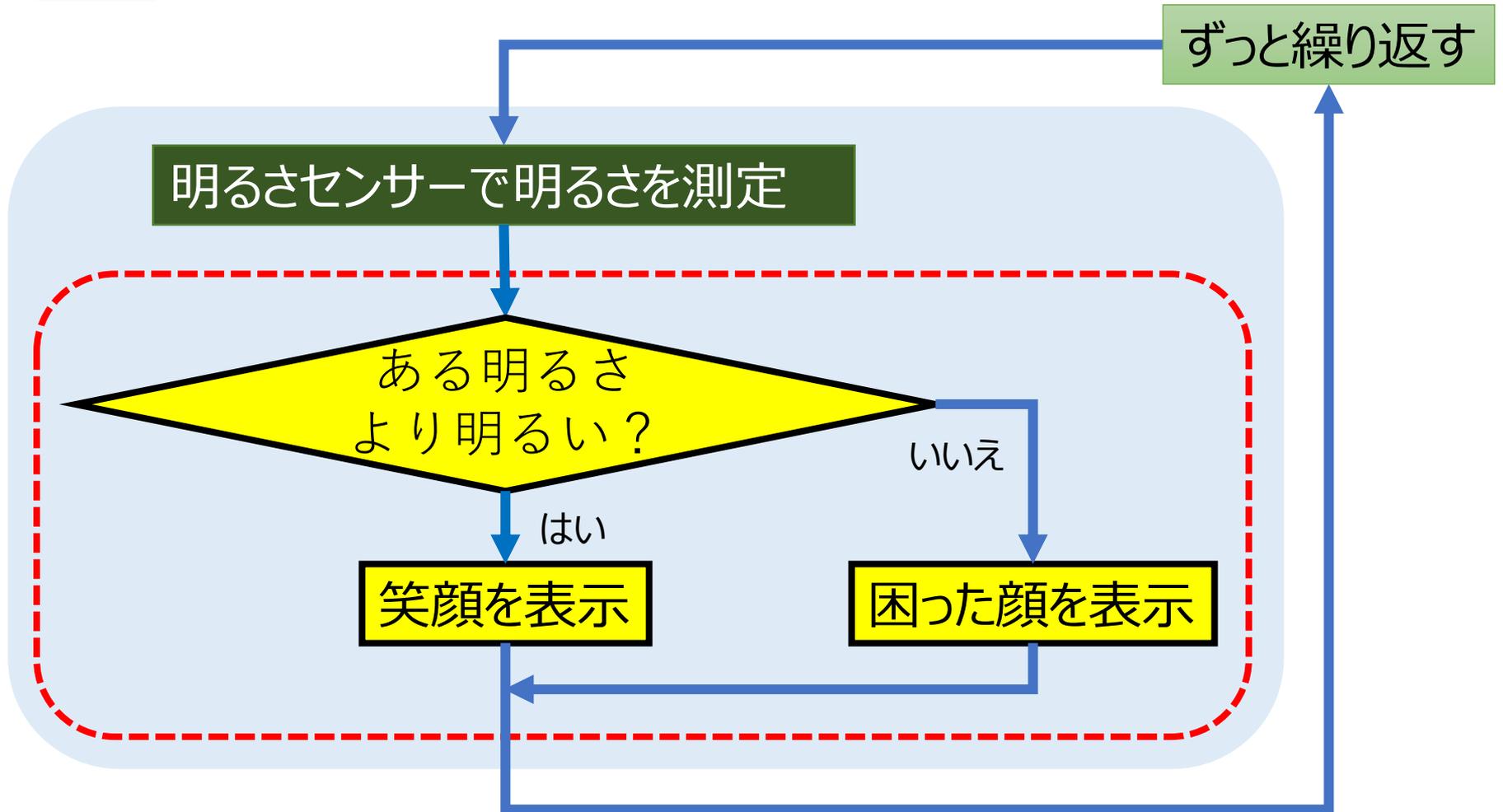


手でかくして，明るさの変化をみてみよう

条件で変わるプログラムを作ろう

目標

部屋などの明るさによって表示が変わるプログラムを作ろう



検索...

基本

入力

音楽

LED

無線

ループ

論理

変数

計算

高度なブロック

条件判断

もし 真 なら

もし 真 なら

でなければ

くらべ

0 = 0

0 < 0

" " = " "

Scratch script showing a blue 'ずっと' (Forever) loop block containing a teal 'もし 真 なら' (If true) block. The 'もし 真 なら' block contains a teal 'でなければ' (Else) block. A red circle highlights the '真' (true) dropdown in the 'もし' block. A red arrow points from this circle to the '0 > 128' comparison block in the 'でなければ' block.

A teal comparison block with '0' in the left field, '>' in the operator dropdown, and '128' in the right field. A red arrow points from the '真' dropdown in the 'もし' block above to this comparison block.

明るさ

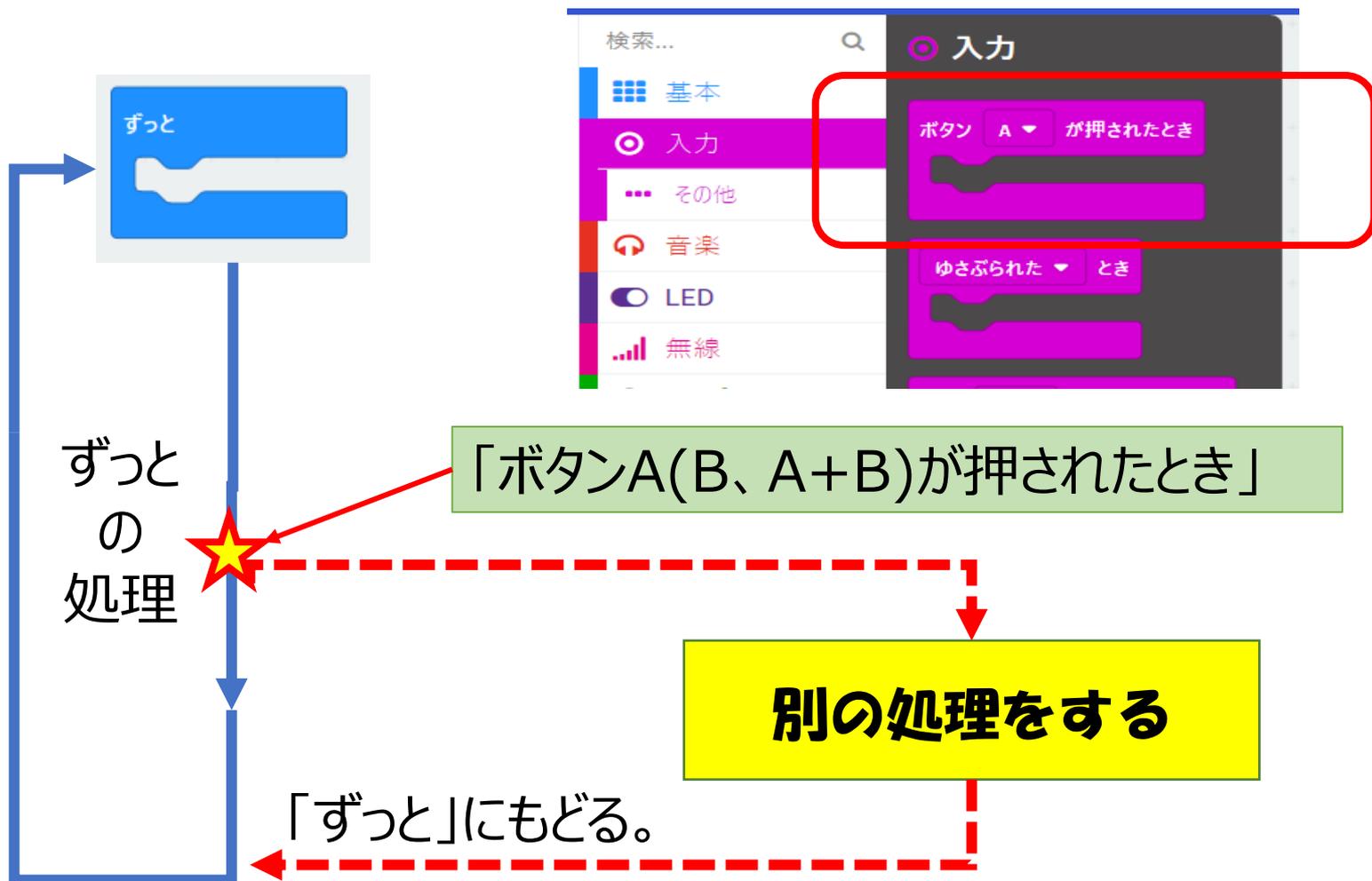
選べる

A teal comparison block with '明るさ' in the left field, '>' in the operator dropdown, and '128' in the right field. A red box surrounds the entire block. A black arrow points from the text '選べる' to the operator dropdown.

条件: 明るさが 128 より大きい

ボタンで電球をつけてみよう

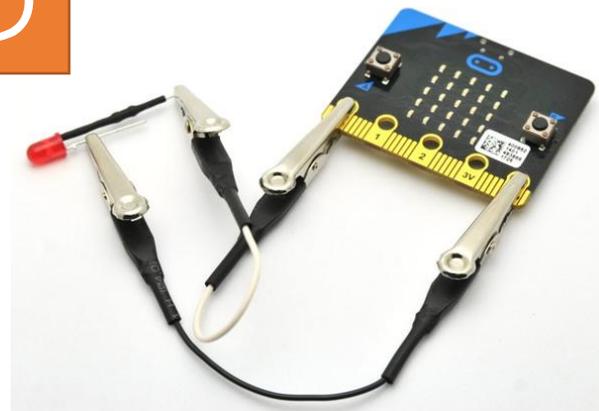
ボタンの働き



ボタンで電球をつけてみよう

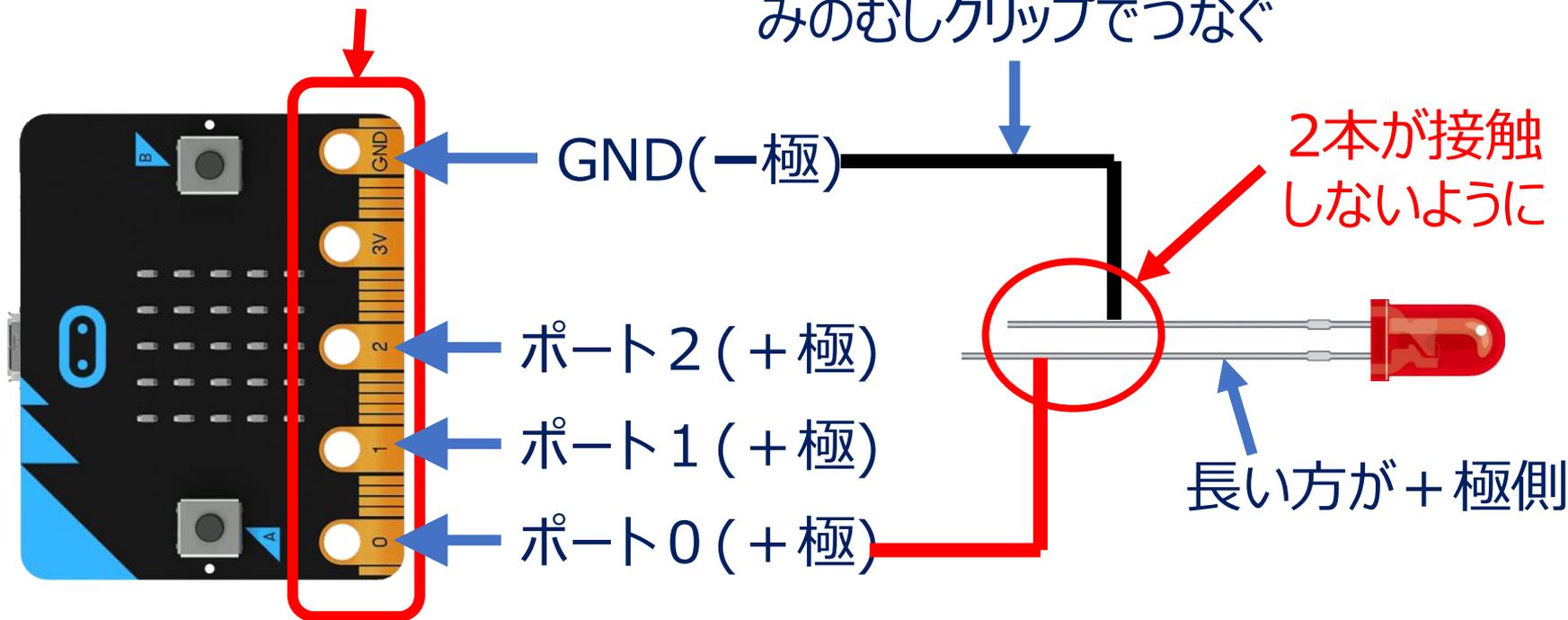
やりたいこと

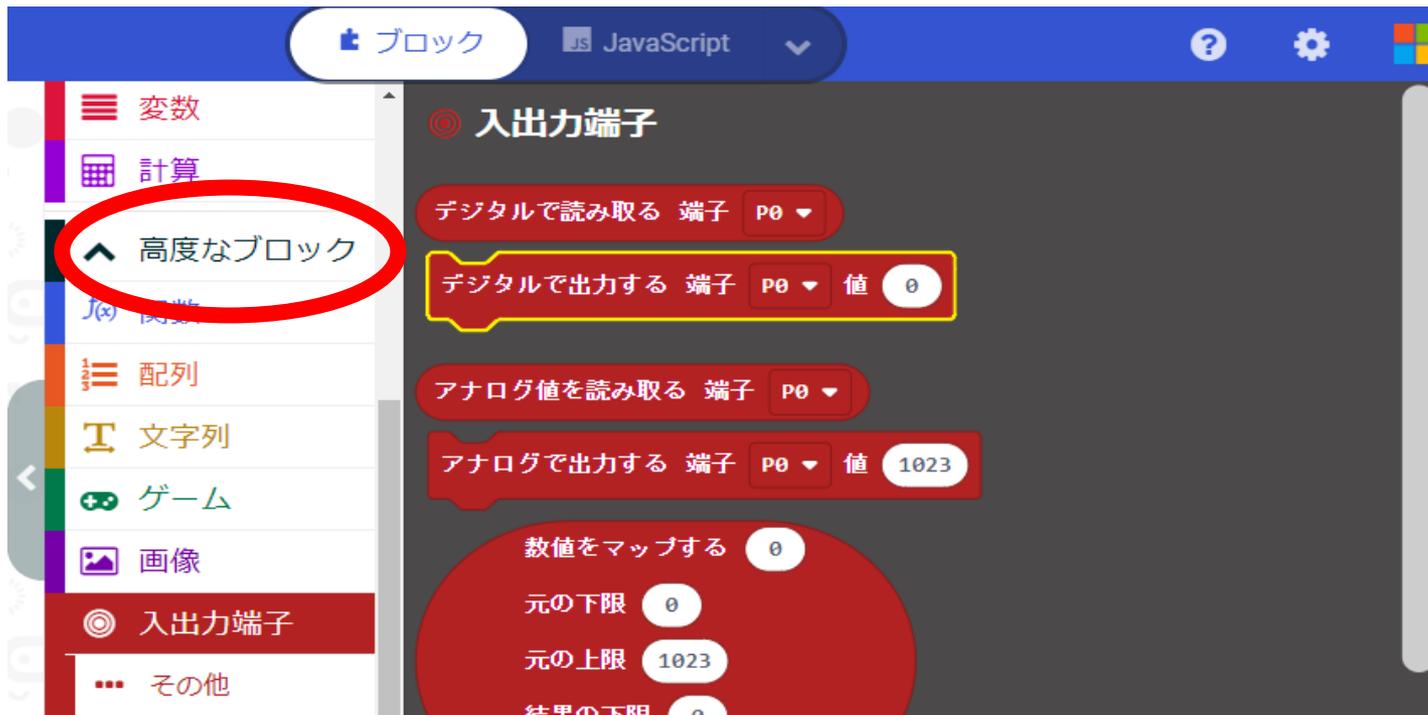
Aボタンを押すと明かりがつき,
Bボタンを押すと明かりが消える



外部出力端子

みのむしクリップでつなぐ





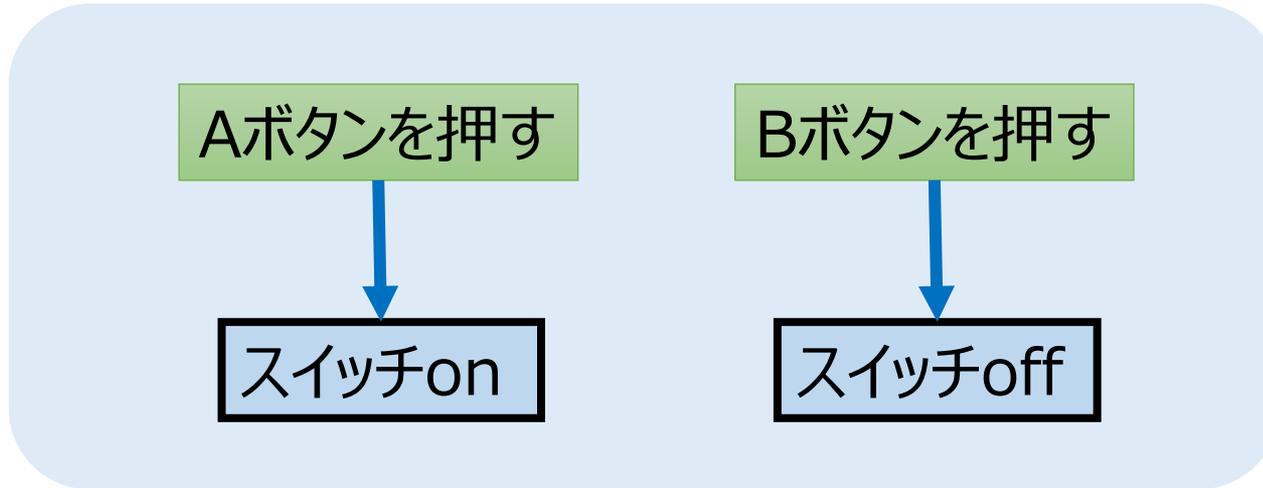
1 = オン
0 = オフ

出力先（ポート）を選ぶ

Aボタンで点灯、Bボタンで消灯するプログラムを作ろう

やりたいこと

Aボタンを押すと明かりがつき,
Bボタンを押すと明かりが消える



1 = オン
0 = オフ

出力先 (ポート) を選ぶ

Aボタンで点灯、Bボタンで消灯するプログラムを作ろう

電球をつけてみよう（改良）

やりたいこと

Aボタンを押すと明かりがつき,
Bボタンを押すと明かりが消える

Aボタンを押す

スイッチon

Bボタンを押す

スイッチoff

やりたいこと

Aボタンだけでon/offさせてみよう

Aボタンを押す

消えている？

はい

スイッチon

いいえ

スイッチoff

やりたいこと

Aボタンだけでon/offさせてみよう

はじめだけ

変数←off

今の状態(on/off)を覚えておくために
変数を用意する。

Aボタンを押す

変数はoff?

はい

スイッチon

いいえ

スイッチoff

変数←on

変数←off

新しい状態
を覚える。

検索...



変数

基本

入力

音楽

LED

無線

ループ

論理

変数

計算

変数を追加する...

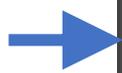
「LED」という名の変数を作る

作成する変数の名前 :

LED

OK

LEDという変数



LED ▾

変数 LED ▾ を 0 にする

変数 LED ▾ を 1 だけ増やす

1 = オン
0 = オフ

やりたいこと

Aボタンだけでon/offさせてみよう

はじめだけ

LED ← 0

変数LEDに覚えさせる
(1 = オン、0 = オフ)

Aボタンを押す

LEDは0 ?

はい

スイッチon

LED ← 1

いいえ

スイッチoff

LED ← 0

新しい状態
を覚える。

今日のまとめ

みじかなものにも プログラムは使われている。

例：外灯は 暗くなるとつき，明るくなると消える。

クーラーは 暑いと冷風を出し，寒いと切れる。

※みじかなものの動きを再現する

「再現工作」 がおもしろい！

次回，オルゴールを再現してみよう！

次世代キッズ プログラミング教室 2023

第1回目：2023年11月12日（日）

micro:bitとプログラミングの基礎、センサーの利用、外部出力、変数

第2回目：2023年11月25日（土）

外部入力、工作のヒント、活用例

第3回目：2023年12月10日（日）

無線通信、作品発表、モーターと関数の利用

オルゴールを作ろう

やりたいこと Aボタンを押したら、音楽が流れるようにしよう

The screenshot shows the Scratch interface for a micro:bit project. The top navigation bar includes 'micro:bit', 'ホーム', '共有', 'ブロック', and 'JavaScript'. On the left, a sidebar lists categories: '基本', '入力', '音楽', 'LED', '無線', 'ループ', and '論理'. The '音楽' (Music) category is selected and highlighted in red. The main workspace shows the '音楽' block palette with sections for 'メロディ' (Melody) and 'トーン' (Tone). A 'メロディ' block is placed on the stage, set to play a melody at 120 bpm. A script block 'ボタン A が押されたとき' (When button A is pressed) is attached to the 'メロディ' block. A red arrow points from the 'メロディ' block in the palette to the 'メロディ' block on the stage.

A close-up of the 'メロディ (上級)' (Melody Advanced) block. It features a 'メロディを開始する' (Start melody) button, a melody selection dropdown menu currently showing 'ダダダム', a 'くり返し' (Repeat) dropdown menu set to '一度だけ' (Once), and a 'メロディを開始する' button.

オルゴールを作ろう

やりたいこと Aボタンを押したら、音楽が流れるようにしよう

シミュレーターを使ってみる



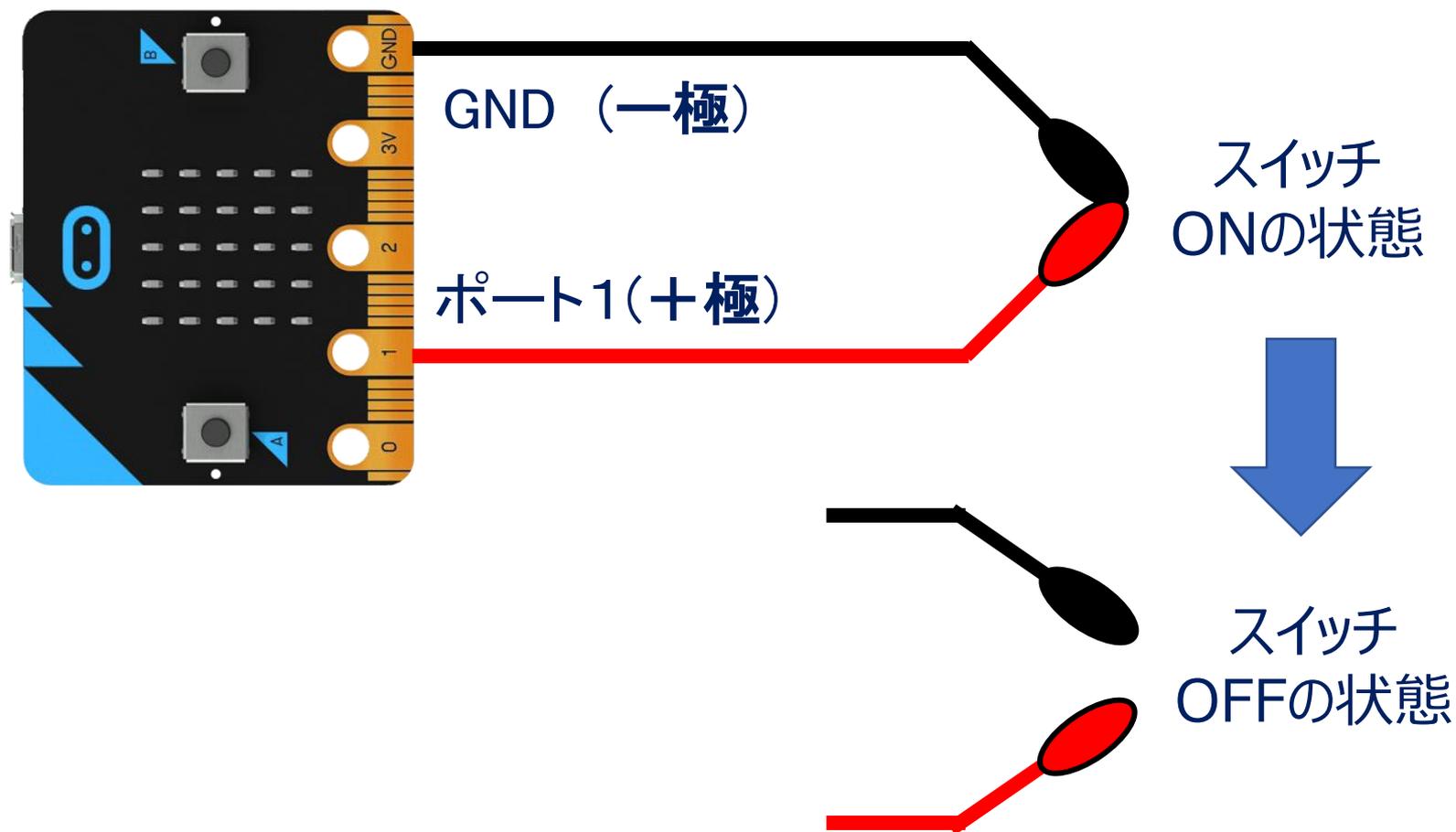
シミュレーターの
Aボタンをクリック

曲を変えて
ためしてみよう

曲が決まったら、
マイクロビットに
ダウンロードしよう

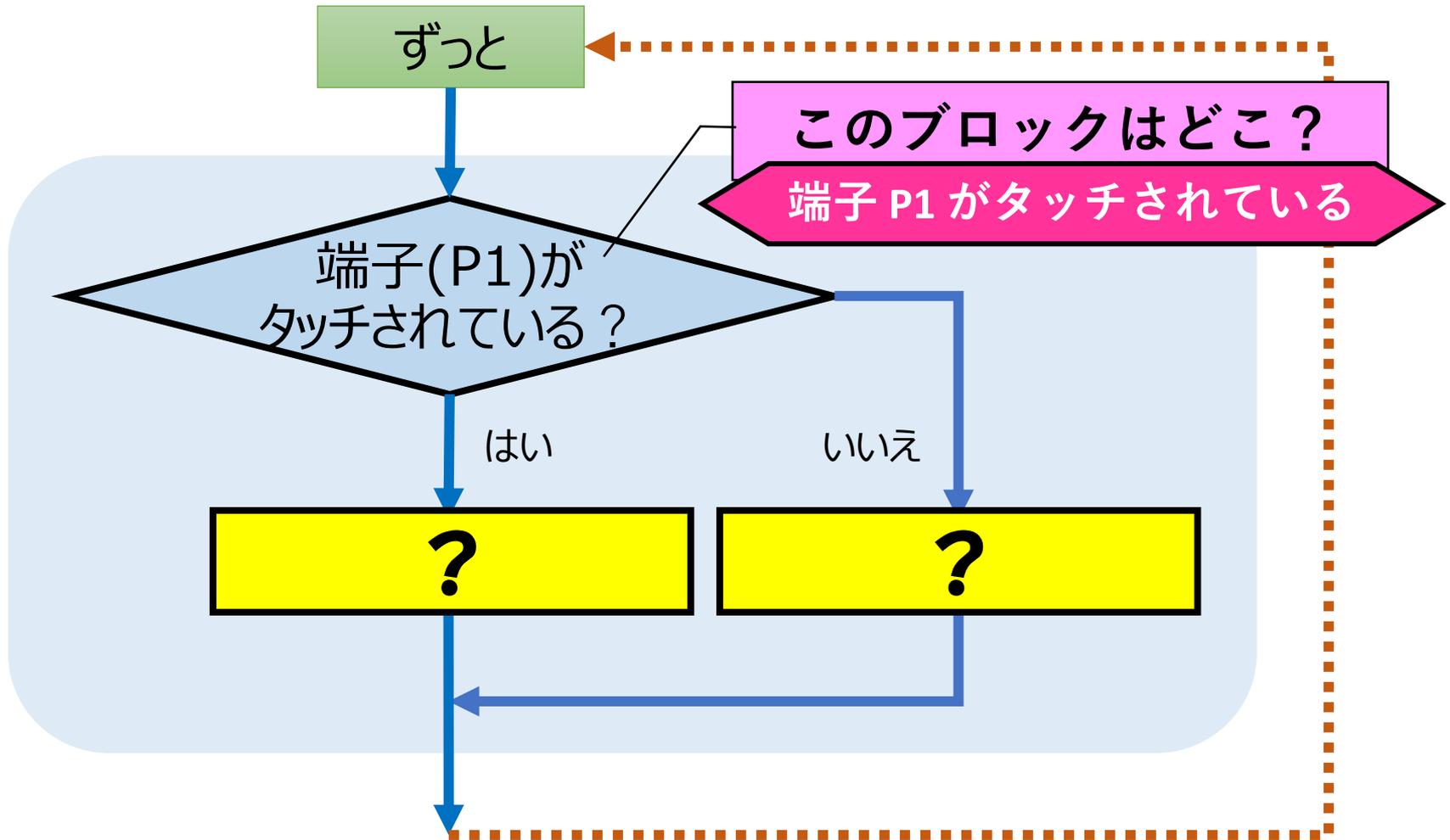
やりたいこと スイッチをオフにすると、音楽が流れるようにしよう

とりあえず、みのむしクリップだけでつないでおく

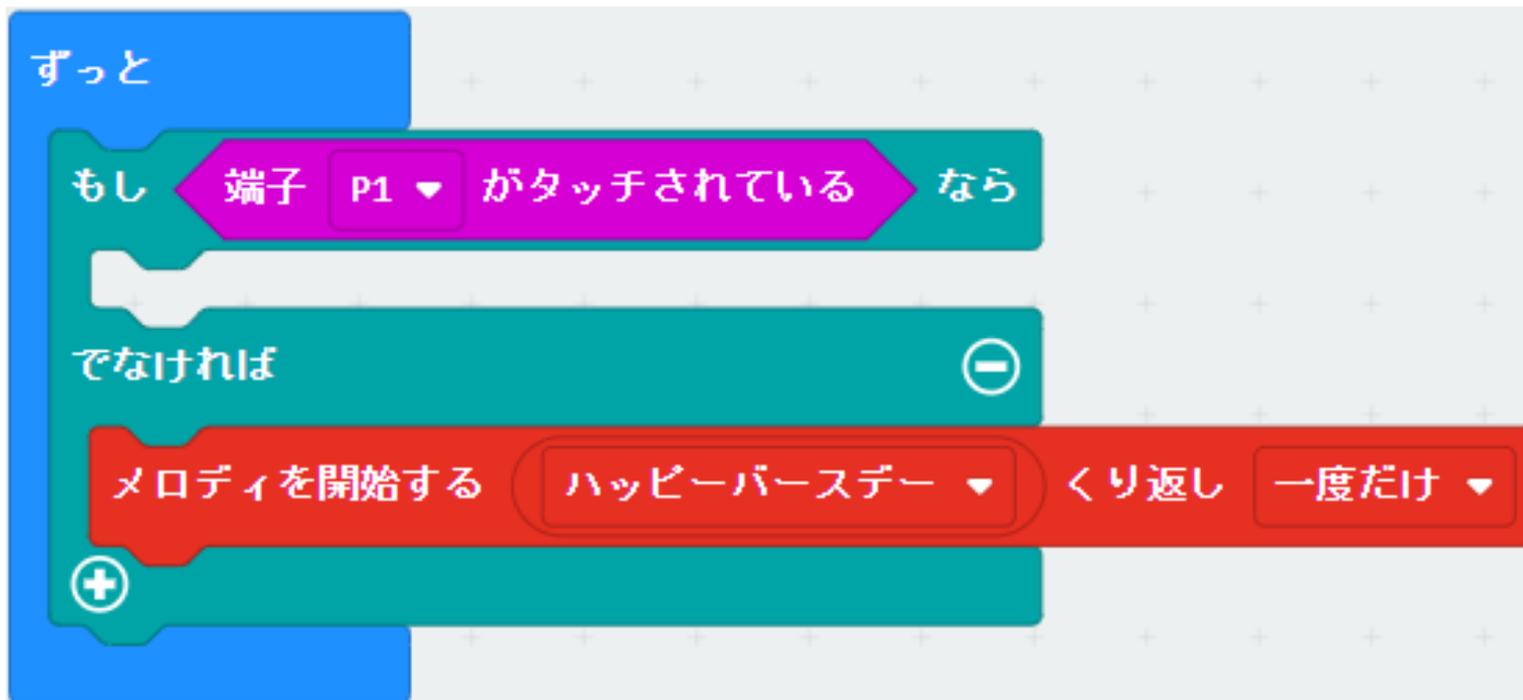


やりたいこと

スイッチをオフにすると、音楽が流れるようにしよう

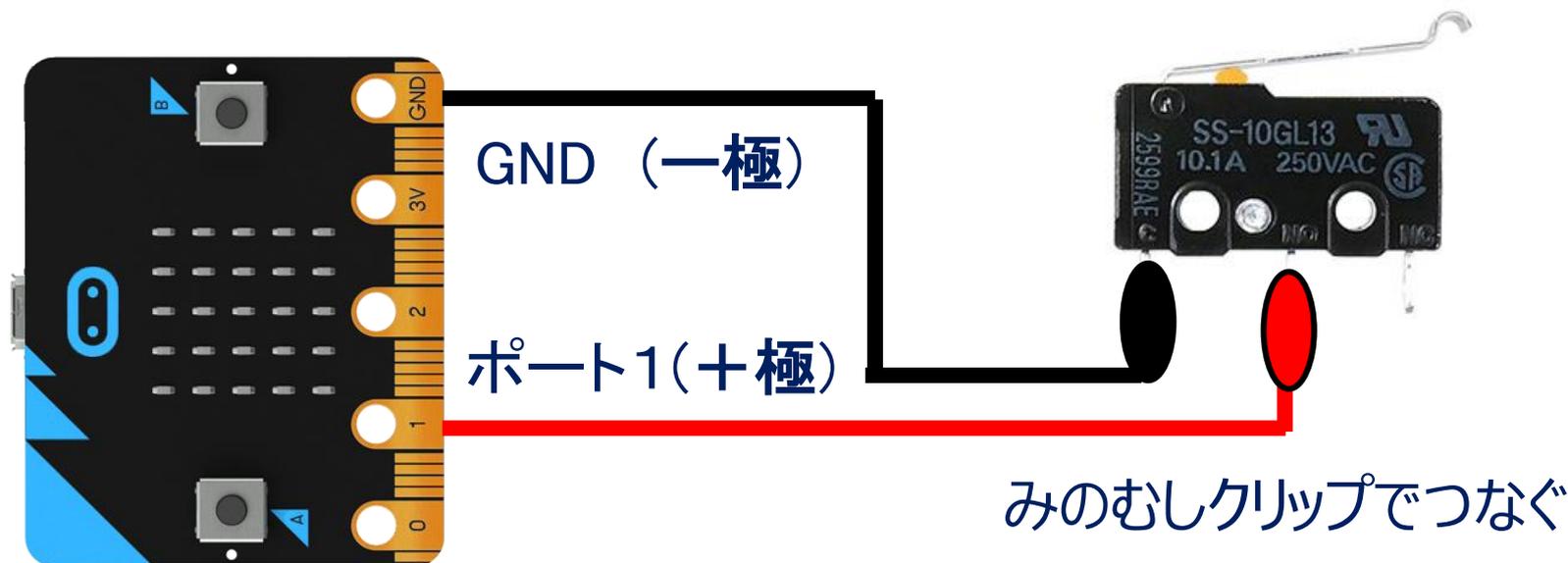
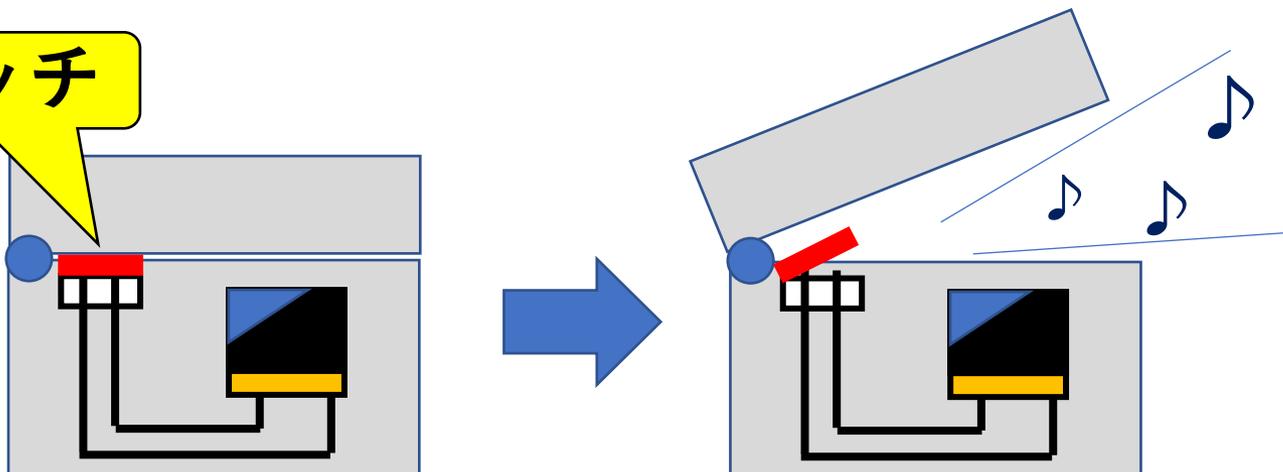


やりたいこと スイッチをオフにすると、音楽が流れるようにしよう



やりたいこと スイッチをオフにすると、音楽が流れるようにしよう

スイッチ



オルゴールを作ろう

マイクロスイッチを箱に取り付けよう。

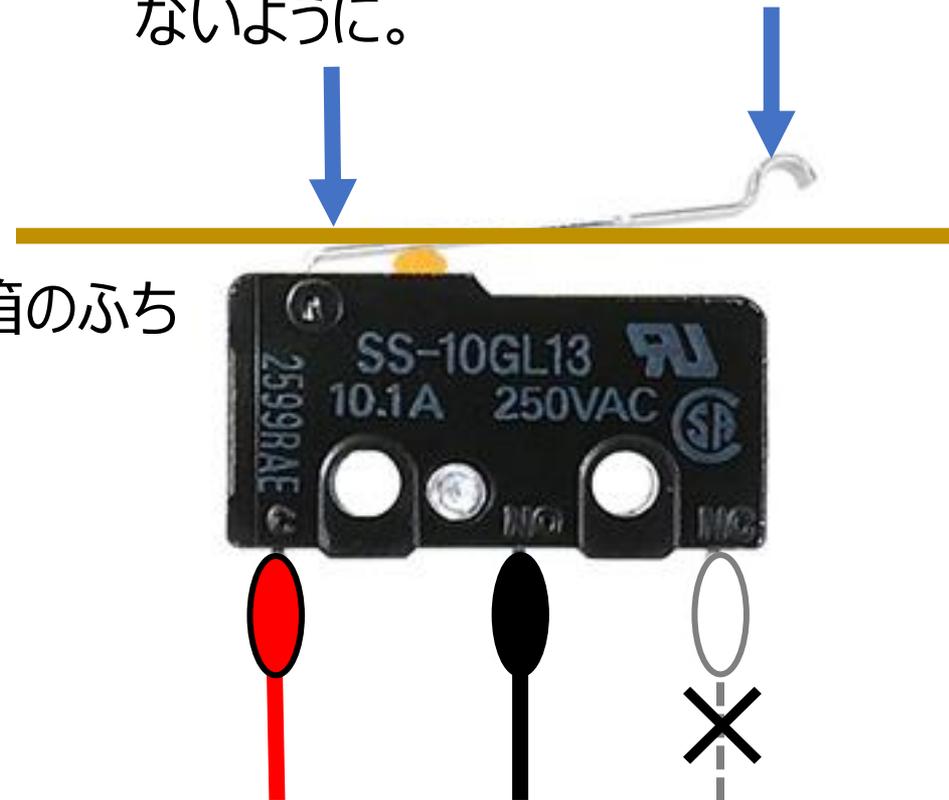
つける高さに注意！

レバーの根元が箱のふちを越えないように。

レバーの先を箱のふちより高くする。



箱のふち



自由にメロディーを作ってみよう

ここをクリック

音を鳴らす 高さ (Hz) 真ん中のド 長さ 1 拍

音程を変える

音を鳴らす 高さ (Hz) 262 1 拍

下のレ

音量を設定する
テンポを設定する

音量を設定する 127

テンポを設定する (bpm) 120

音を鳴らす 高さ (Hz) 真ん中のド 長さ 1 拍

音を鳴らす 高さ (Hz) 真ん中のミ 長さ 1 拍

音を鳴らす 高さ (Hz) 真ん中のソ 長さ 1 拍

音を鳴らす 高さ (Hz) 上のド 長さ 2 拍

長さを変える

工作のヒント

いろいろな工作にチャレンジしてみよう。

鞍手高校SSH部

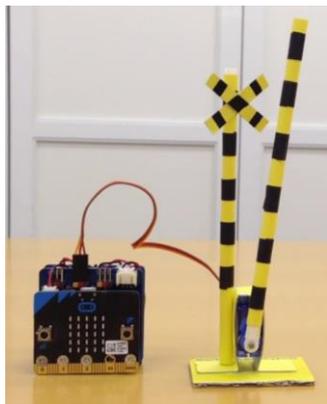
講師の先生

工作のヒント

いろいろな工作にチャレンジしてみよう。



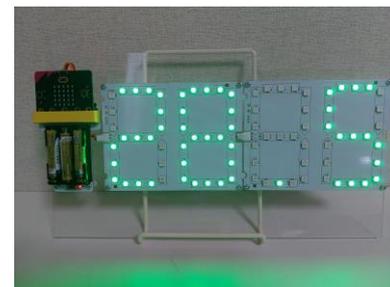
ドアチャイム



踏切



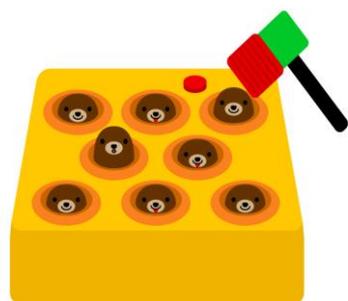
信号機



温度計



腕時計型
コントローラー



モグラたたき

ピンポーン



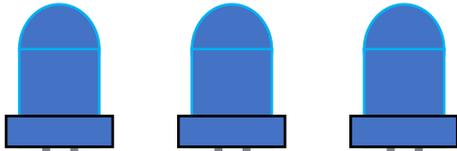
クイズの早押しボタン



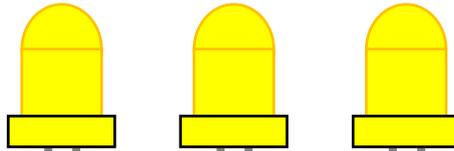
イライラ棒

LEDライトの応用

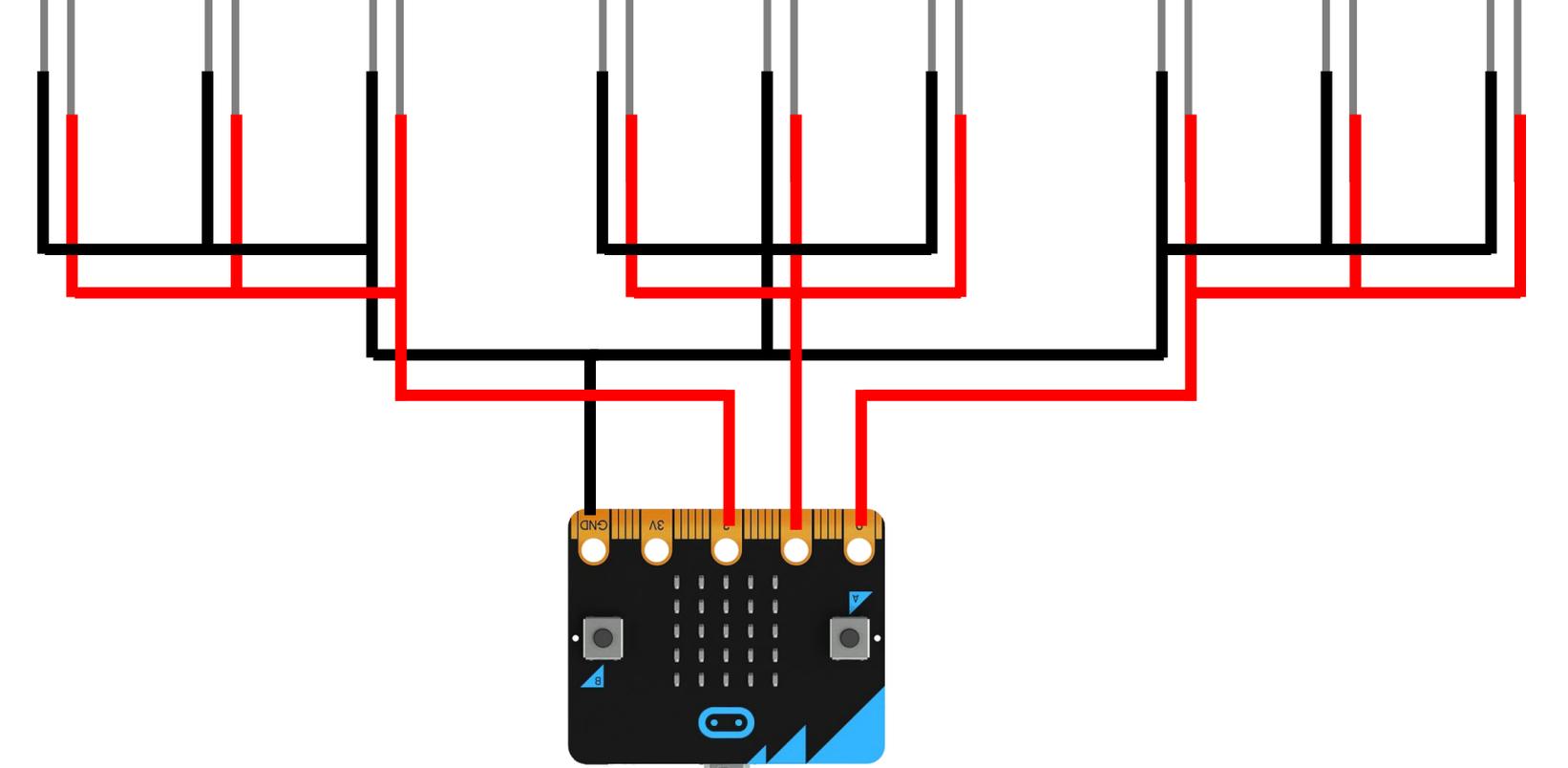
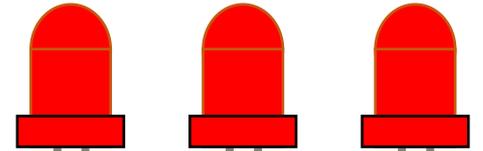
ポート 2



ポート 1



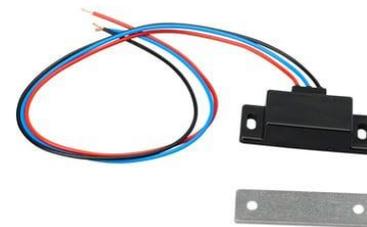
ポート 0



スイッチの応用

さらにセンサーを使うと・・・

磁気近接センサー

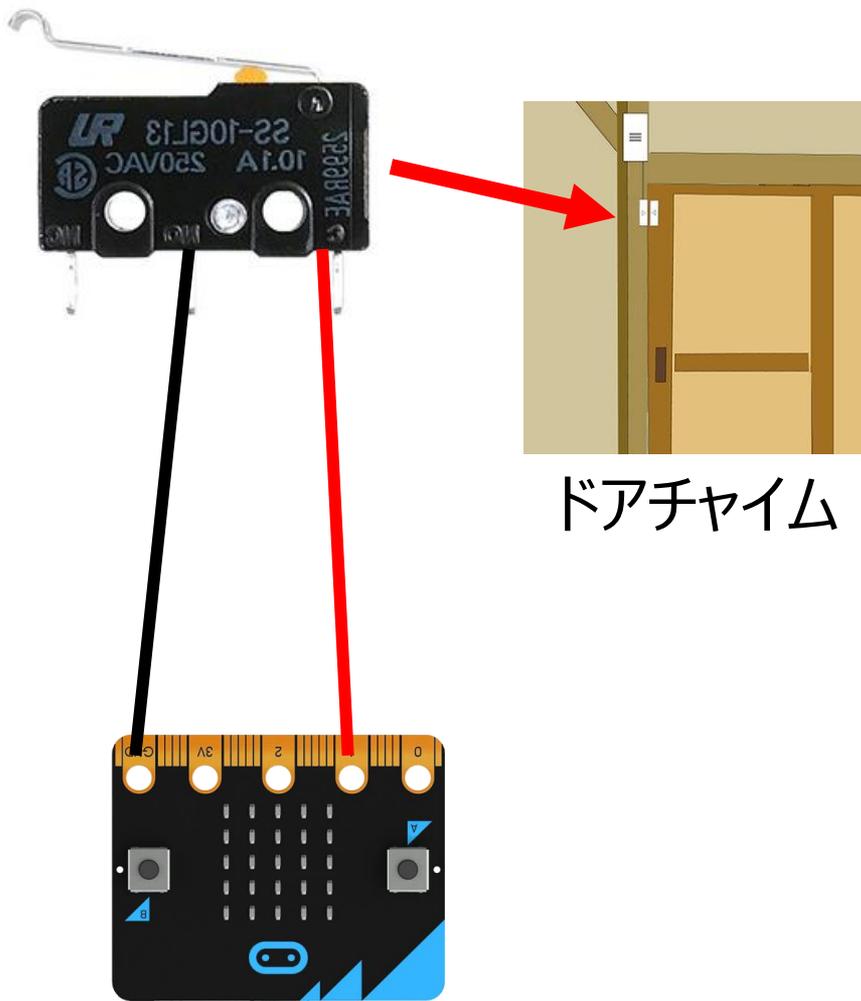


人感センサー

(赤外線
距離センサー)



通過センサー



ドアチャイム

モーターの応用



DCモーター
・・通常のモーター

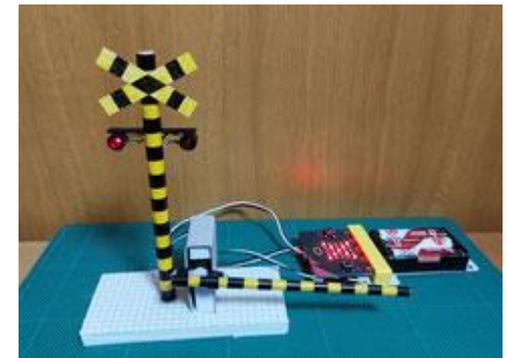


注意

※Microbitだけでは
動かない。



サーボモーター FS90等
・・指定した角度に回転



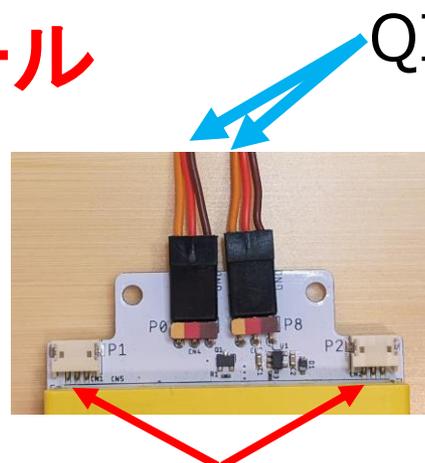
※見た目では
区別できない

回転サーボモーター FS90R等
・・速さと向きを変えられる



あると便利なもの

ワークショップモジュール



QIコネクタ

サーボモーターなど

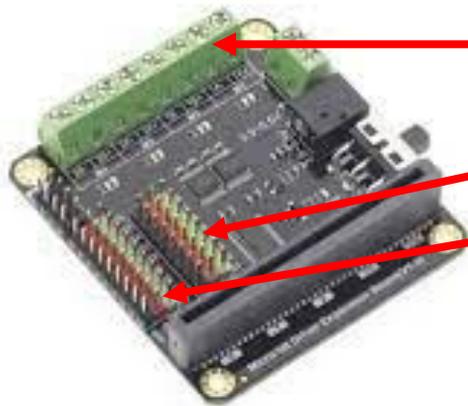
※通常のモーターは
使えない。

ZHコネクタ

センサー, スイッチなど

モータードライバ

おすすめ!



通常のモーター (DCモーター)

サーボモーター

センサー, スイッチなど

※電池ボックスなし

(モバイルバッテリーも使える)

参考になるサイト



トム君の マイクロビット講座



工作に役立つマイクロビットの基礎知識や
部品の入手先を紹介しています。

いろいろな工作にチャレンジしてみよう。

夏休みの工作で応募してみよう！

次世代夢プロジェクト2023
わたしの将来!!まちや生活!!
10年後 はこうすれば
よくなる!
ドキドキ!!ワクワク!!な未来をつくる
アイディア大募集!!

①今の問題や将来の目標 → ②どんな工夫やアイデアで

(例えば) 泥棒をなくす 高齢者の事故 将来の夢
ATM オレオレ詐欺をなくす 見守りを増やす ユーチューバーになる
自然破壊を止める 空き家を減らす 家族が仲良くなる

③どんな未来になる

(例えば) ユーチューブの撮影や編集スタジオがあったら練習できる。
家族や親戚との交換日記をする 大切に、いつかまたみんなで見る

詳しくはこちらで WEBからも応募できます

みんなの応募お待ちます!!全員に参加賞もあるよ!

■募集のあんない
このチラシの裏面又はWEBのフォーマットにて必要事項を必ず記入して応募ください。10年後はこうすればよくなるのか「ドキドキ!!ワクワク!!」な未来が作れるのか写真やイラストをつけて内容をわかりやすく書いてください。
■応募 記入した用紙を各学校に置かれている応募専用ボックスに入れてください。直轄産業振興センターに郵送、またはFAX 0949-22-2700にて受け付けます WEBではそのまま送信してください
■賞格 小学校1年生から中学校3年生まで随時 令和5年9月15日(金)発表 最優秀作品・入賞作品を表彰します
■問い合わせ 直轄産業振興センター(ADOX福岡)所在地 直方市大字榎木849-1 電話 0949-22-0575
■主催 直轄産業振興センター(ADOX福岡) 直轄次世代産業研究会共催 直方市
■後援 宮若市 小竹町 鞍手町 直方商工会議所 宮若商工会議所 トヨタ自動車九州株 直轄工業人クラブ

令和5年度 ADOX福岡 発明工夫展

ADOX福岡では、毎年夏休み期間中に直方市内の小・中学校に『発明くふう展』の募集をしています。

今年は、5校26作品の応募がありました。9月14日に審査会を行い、各賞5点、奨励賞5点が選ばれました。全応募作品はADOX福岡別館に展示公開していますので、みなさまぜひご覧ください。

展示期間 9月24日(日)～9月26日(火)
10:00～17:00
場 所 直轄産業振興センター
ADOX福岡別館 1階 ロビー

入賞作品は、10月に開催される『福岡県児童生徒発明くふう展』に応募します。

チャレンジ

自分でいろいろ考えながら、
何かを作って、
みんなに見せてみよう！！

次回、発表会！

次世代キッズ プログラミング教室 2023

第1回目：2023年11月12日（日）

micro:bitの基礎、プログラムの基本、センサーの利用、外部出力

第2回目：2023年11月25日（土）

変数、外部入力、活用例

第3回目：2023年12月10日（日）

無線通信、作品発表、モーターと関数の利用

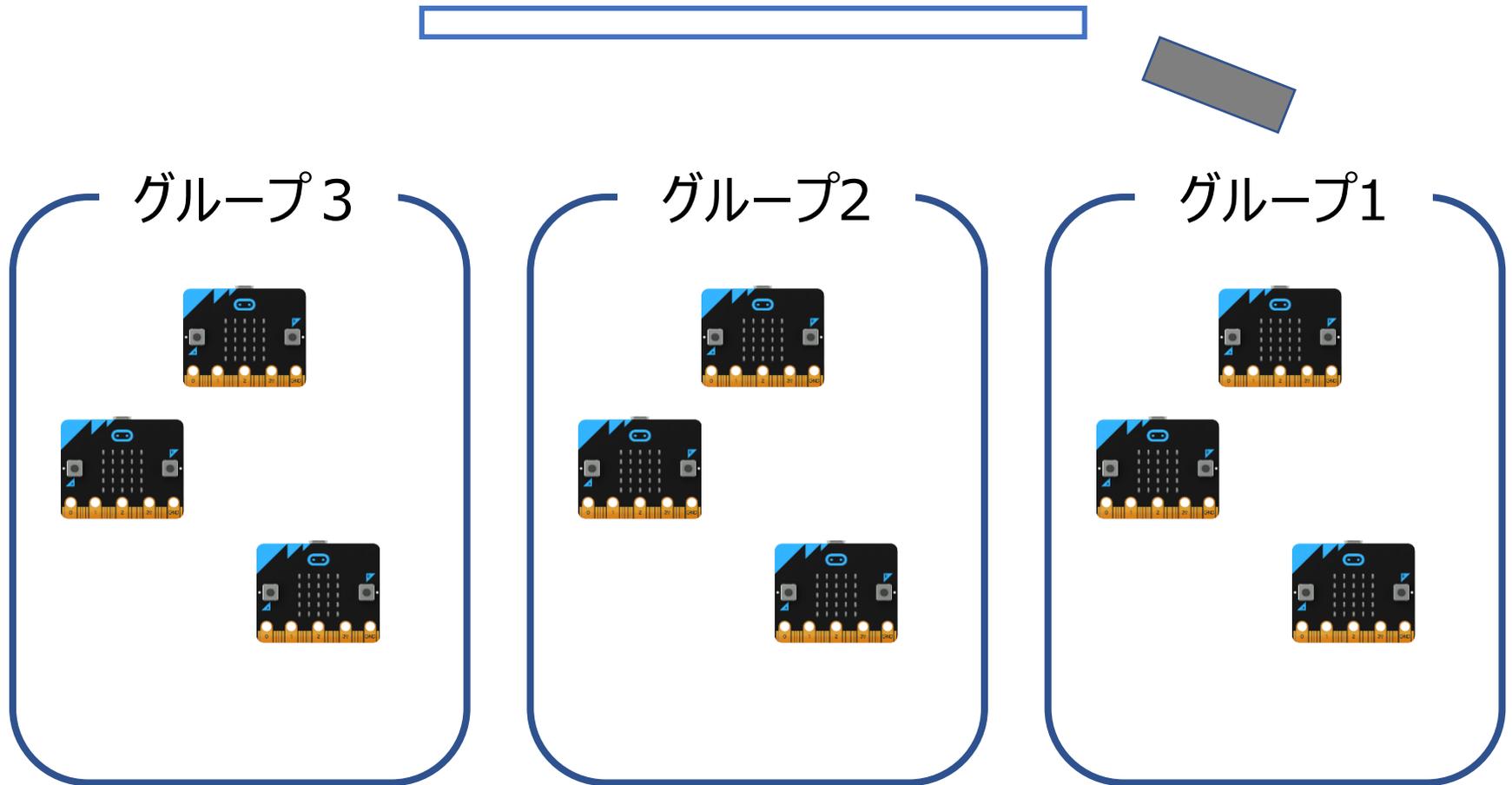
作品を発表したいひと

※作品を発表してくれる人は
SSH部の部員に声をかけてください。

発表する作品は
自分のマイクロビットにいらしてください。
今日作る「いいねボタン」は
SSH部のマイクロビットにいきます。

無線通信に挑戦

同じグループ内で情報をやりとりする。



無線グループの設定

The screenshot shows the Microsoft MakeCode IDE interface. At the top, there's a blue header with 'ブロック' (Blocks), 'JavaScript', and the Microsoft logo. On the left, a sidebar contains a search bar and a list of categories: '基本' (Basic), '入力' (Input), '音楽' (Music), 'LED', '無線' (Wireless), 'その他' (Other), and 'ループ' (Loops). The '無線' category is selected and highlighted in pink. The main workspace shows a '無線' block palette with several blocks: '無線のグループを設定' (1), '無線で数値を送信' (0), '無線で送信' (name = 0), and '無線で文字列を送信' ("). A blue box labeled '最初だけ' (Only at first) highlights the '無線のグループを設定' block, with a red arrow pointing to it and the text 'グループを選択' (Select group).

The screenshot shows a workspace with a 'ボタン A が押されたとき' (When button A is pressed) block connected to a '無線で文字列を送信' (Wireless send text) block. The text 'TARO' is entered in the input field of the '無線で文字列を送信' block.

Aボタンで名前を送ってみよう

情報を送受信しよう

※Aボタンで情報を送る場合

文字のやりとり

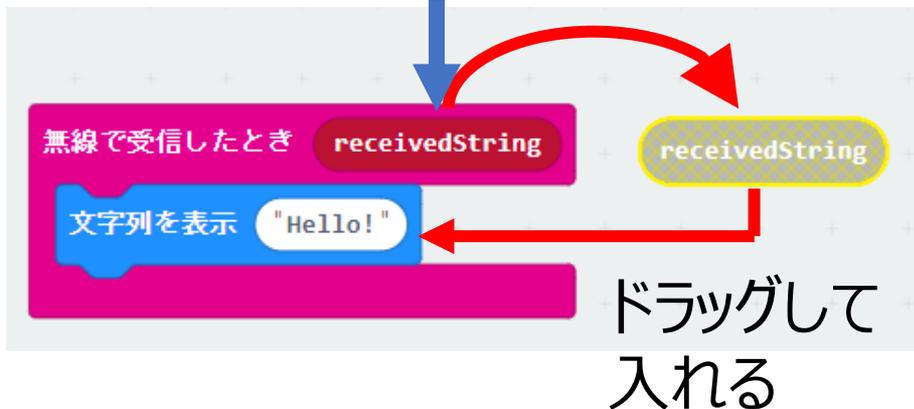
送信側



文字列

receivedString
受け取った文字

受信側



ドラッグして
入れる

数字のやりとり



数字

receivedNumber
受け取った数字



情報を送受信しよう

文字と数字をセットで送る場合

送信側



受信側



Name value
名前 値

発表準備

「いいねプログラム」を作ろう。

「**いいね**」という変数を作っておきます。

グループは
全員1

最初だけ

- 無線のグループを設定 1
- 変数 いいね を 0 にする
- 文字列を表示 "ok!"

ボタン A が押されたとき

- 無線で数値を送信 1

ボタン B が押されたとき

- 表示を消す
- 変数 いいね を 0 にする
- 文字列を表示 "ok!"

無線で受信したとき receivedNumber

- もし いいね < 25 なら
- 点灯 x いいね ÷ 5 y remainder of いいね ÷ 5
- 変数 いいね を receivedNumber だけ増やす
- でなければ
- くりかえし 5 回
- 表示を消す
- 一時停止 (ミリ秒) 200
- アイコンを表示
- 一時停止 (ミリ秒) 200

5で割った余り

「ループ」の中
にあります。

作品発表

- ※番号順に発表してもらいます。
作品はスクリーンに表示します。
- ※発表を見た人は、がんばって発表した人に「いいね」を送ろう。

もっとMicrobitで楽しもう

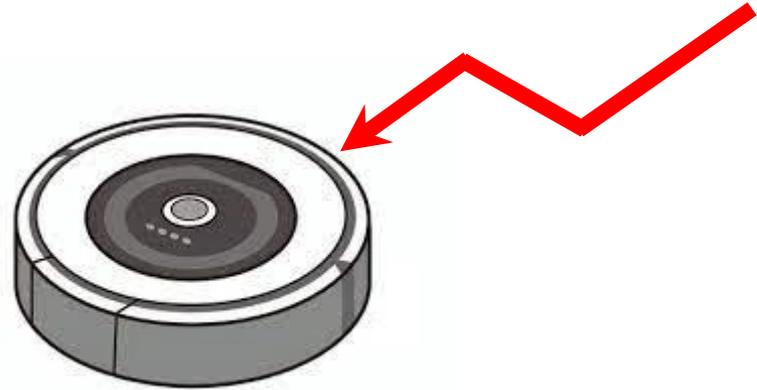
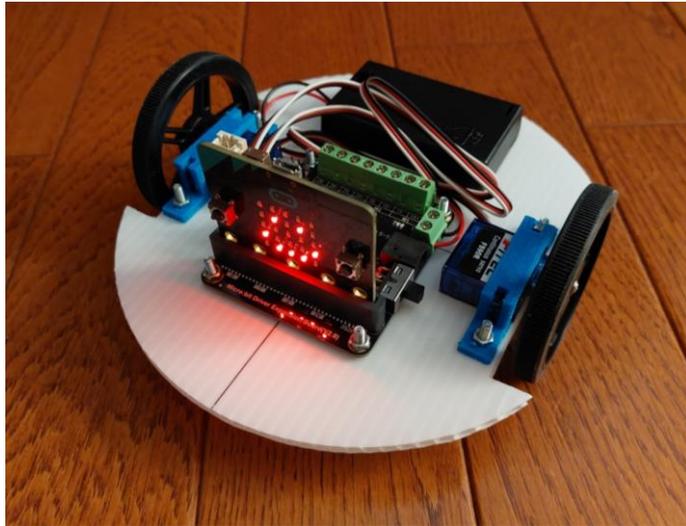
2つのグループに分かれて、交代で行います。

	Aグループ	Bグループ
10:35 11:10	①自動走行車 (ミーティングルーム)	②ラジコンカー (研修室)
	休憩・移動	
11:15 11:50	②ラジコンカー (研修室)	①自動走行車 (ミーティングルーム)

※密にならないように
お互いに気を付けましょう。

もっとMicrobitで楽しもう

① 自動走行車を作ろう。

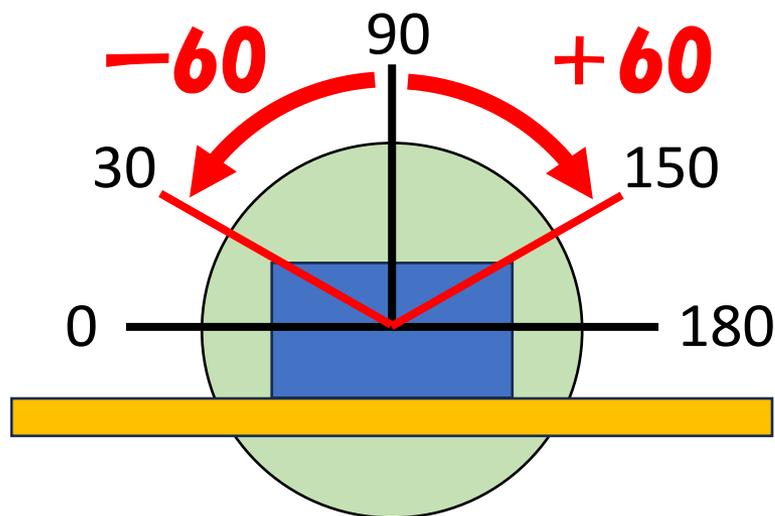


モーターの種類

〔サーボモーター
回転サーボモーター

- ・ 指定した角度になる
- ・ 回転の速さと向きが指定できる

サーボモーターのしくみ



回転サーボモーター

90で停止，
90以下と90以上では逆方向，
0と180で回転が最速になる。

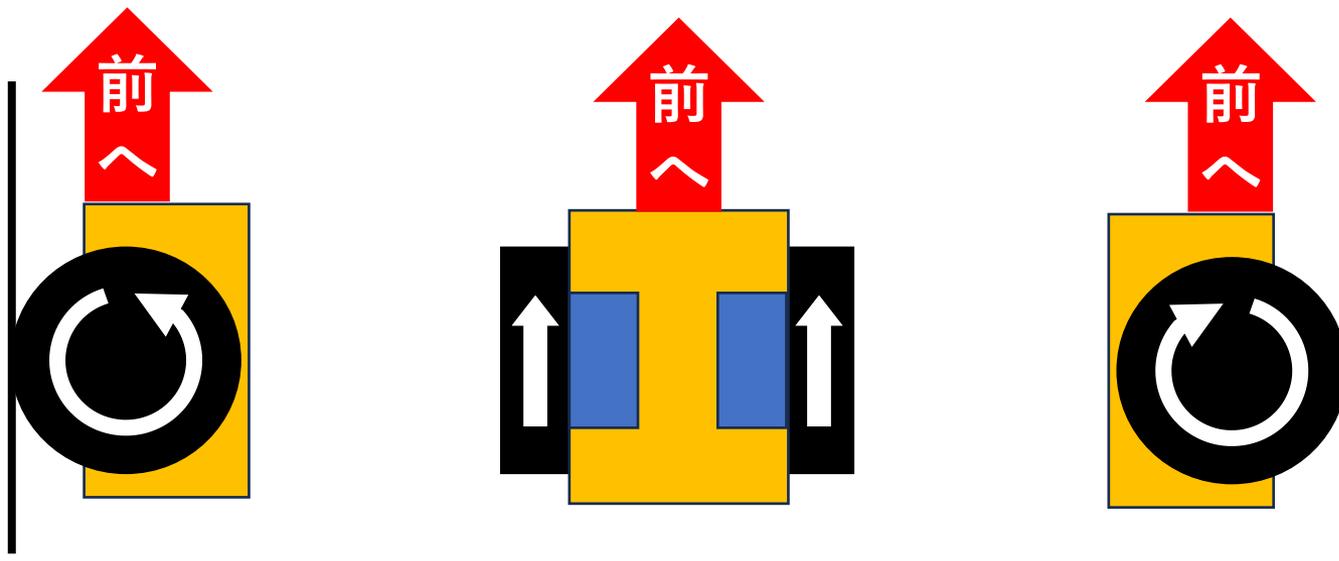
※初めて使うときは，裏のネジで
90で停止するように調整が必要

※0より小さい値や180より大きな値を入れると
モーターが壊れることがあるので，
安全のために10～170の範囲に制限する。

サーボ 出力する 端子 角度 を 以上 以下の範囲に制限

車輪の回転方向

左から見ると・・・ 上から見ると・・・ 右から見ると・・・



※前に進む場合、左右の車輪は逆回転になる。

例) 左が $+20$ ($90+20=110$) なら
右は -20 ($90-20=70$) にする

プログラムの作り方

左車輪のモーター ⇒ P0ポート

右車輪のモーター ⇒ P8ポート

につないだ場合

ボタンAを押すと

左 + 20 (90 + 20 = 110) , 右 - 20 (90 - 20 = 70) で
1秒間 (1000ミリ秒間) 前進する

ボタン A が押されたとき **前進する**

サーボ 出力する 端子 P0 角度 110 を 10 以上 170 以下の範囲に制限

サーボ 出力する 端子 P8 (出力のみ) 角度 70 を 10 以上 170 以下の範囲に制限

一時停止 (ミリ秒) 1000

サーボ 出力する 端子 P0 角度 90 を 10 以上 170 以下の範囲に制限

サーボ 出力する 端子 P8 (出力のみ) 角度 90 を 10 以上 170 以下の範囲に制限

止まる

プログラムの作り方

左車輪のモーター ⇒ P0ポート

右車輪のモーター ⇒ P8ポート

につないだ場合

ボタンAを押すと

左 + 20 (90 + 20 = 110), 右 - 20 (90 - 20 = 70) で
1秒間 (1000ミリ秒間) 前進する

ボタン A が押されたとき **前進する**

サーボ 出力する 端子 P0 角度 110 10 以上 170 以下の範囲に制限 20

サーボ 出力する 端子 P8 (出力のみ) 角度 70 10 以上 170 以下の範囲に制限 20

※ 「左20, 右20 ⇒ 前進」
つまり, 左右同じ数なら前進

これを繰り返すと色々な動きをプログラムできる。

関数の利用

何度も使うプログラムは関数にすると便利。

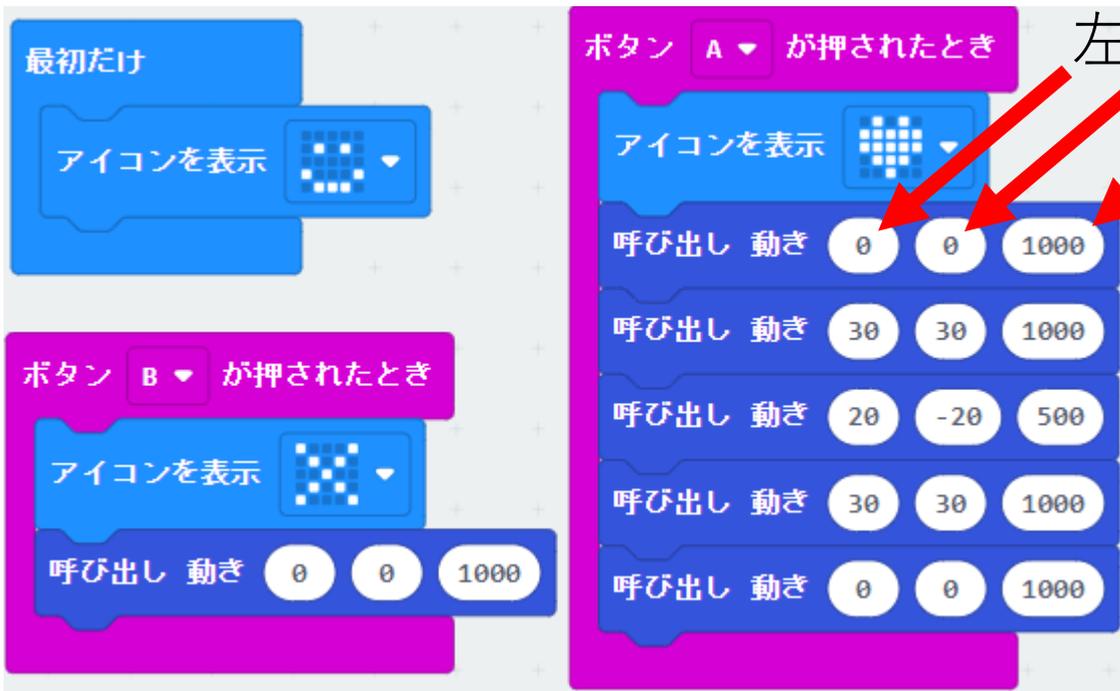
The image shows a programming environment interface with a sidebar on the left containing various blocks like '無線', 'ループ', '論理', '変数', '計算', 'Neopixel', '拡張機能', '高度なブロック', '関数', and '配列'. The main workspace shows a '関数' (Function) block being created. A red arrow points from the '関数を作成する...' button to the '関数' block. Another red arrow points from the '関数' block to the '動き' (Motion) block. Three red arrows point from the '動き' block to the '左' (Left), '右' (Right), and '時間' (Time) blocks. The '関数' block is labeled '関数' and '動き'. The '動き' block is labeled '動き', '左', '右', and '時間'. The '左', '右', and '時間' blocks are labeled '左', '右', and '時間'.

①関数を作成して
関数名「動き」を付ける

②数値（引数）
を作成して
名前を付ける

関数の利用

- ①プログラムを関数に移動する
- ②数値をドラッグして入れる

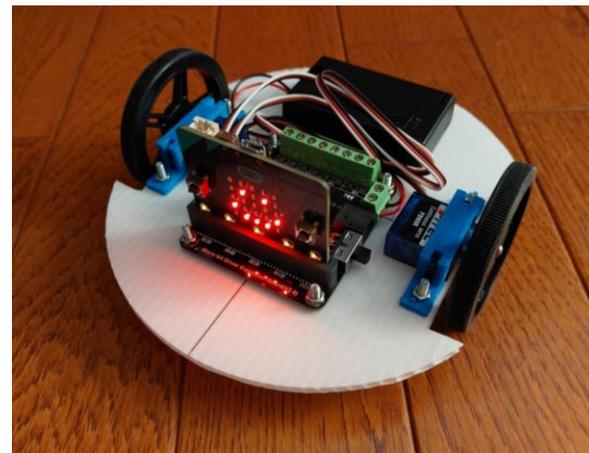


- 1000ミリ秒止まる
- 1000ミリ秒前進
- 500ミリ秒右回転
- 1000ミリ秒前進
- 1000ミリ秒止まる

※緊急停止

走らせてみよう

車にマイクロビットをさして、
電源を入れて、Aボタンを押す。



**※マイクロビットを抜き差しするときは、
必ず電源を切った状態で行う。**

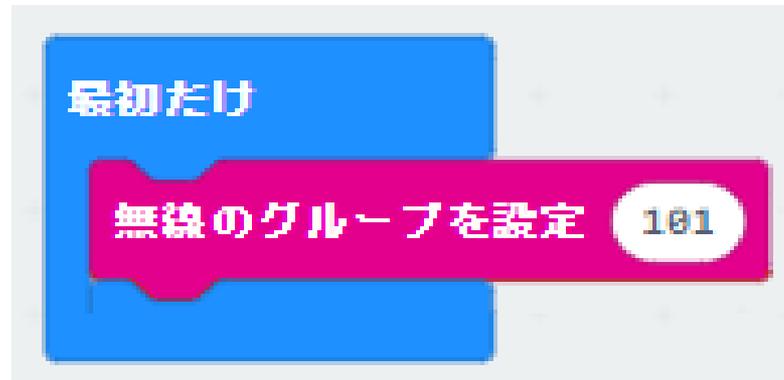
**※絶対に机の上で動かさない。
(必ず床の上で動かす)**

思った通りに動かないときは、
左，右，時間に入れる数字を変えてみる
どう変えればいいのか考えて， やってみよう！

もっとMicrobitで楽しもう

②ラジコンカーのコントローラーを作ろう。

無線グループの設定



※無線グループは、

1号車=101、2号車=102、...16号車=116です。
各グループのラジコンカーに書かれた番号を
確認してください。

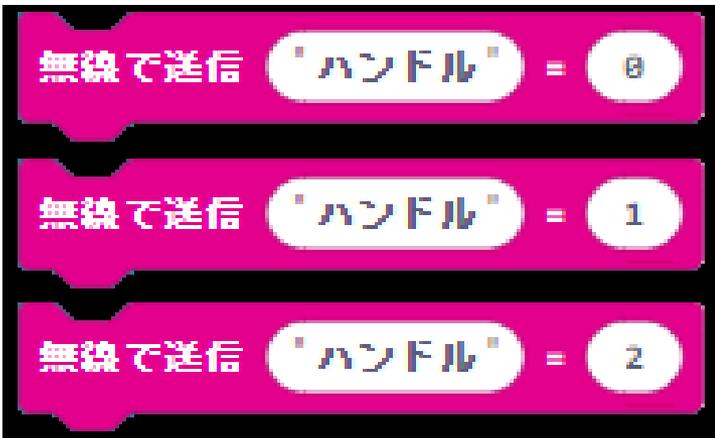
もっとMicrobitで楽しもう

ハンドルの設定

正面 無線で送信 'ハンドル' = 0

右に 無線で送信 'ハンドル' = 1

左に 無線で送信 'ハンドル' = 2

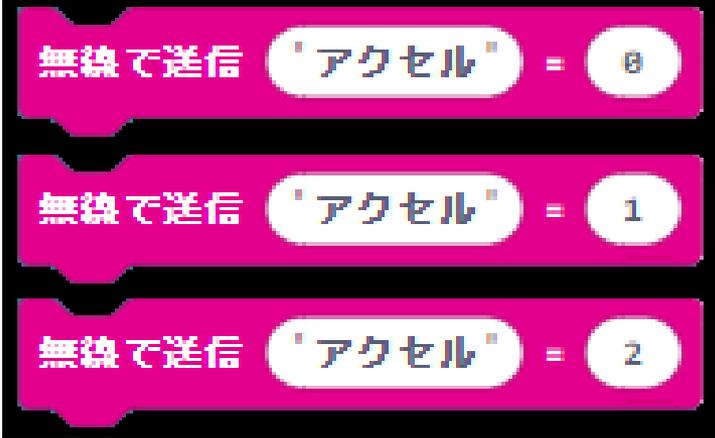
Three Scratch '無線で送信' (Wireless Send) blocks are stacked vertically. The first block is labeled '正面' (Front) and has the variable 'ハンドル' (Handle) set to 0. The second block is labeled '右に' (To the right) and has the variable 'ハンドル' set to 1. The third block is labeled '左に' (To the left) and has the variable 'ハンドル' set to 2.

アクセルの設定

停止 無線で送信 'アクセル' = 0

前に 無線で送信 'アクセル' = 1

後に 無線で送信 'アクセル' = 2

Three Scratch '無線で送信' (Wireless Send) blocks are stacked vertically. The first block is labeled '停止' (Stop) and has the variable 'アクセル' (Accelerator) set to 0. The second block is labeled '前に' (Forward) and has the variable 'アクセル' set to 1. The third block is labeled '後に' (Backward) and has the variable 'アクセル' set to 2.

※どの動きに割り当てるかは自由！
どうすれば使いやすくなるかな？

自分だけのコントローラを作ろう！

もっとMicrobitで楽しもう

ラジコンカーのコースは2つあります。
時間を見て移動してください。

	名前	最初の班
1	自動車学校コース	グループ1
2	サッカー場コース	グループ2

※密にならないように
お互いに気を付けましょう。