

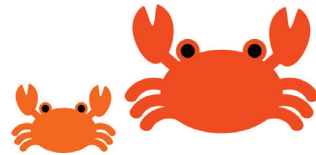


鳥大ものづくり教室×TIA KIDS SCHOOL

フィジカル・コンピューティング 体験教室



2017/08/19



案内

- トイレは「実習工場」のものをお使い下さい
- この部屋は飲食可能ですが、ゴミは分別して捨てるようにして下さい
- お車をご利用の方には駐車場が無料になるサービス券を配布致します
- ご不明な点やご質問があれば、挙手してスタッフにお声がけ下さい

写真撮影について

今日の教室の様子を写真に撮らせて頂きます。
写真は、下記のような用途に使用する予定です。
不都合のある方はスタッフまでお声がけ下さい。

- 公式SNS、WEBサイトへの掲載
- 報告書等への掲載
- 学会発表等の資料に掲載
- 新聞報道への提供



工学部附属ものづくり教育実践センター

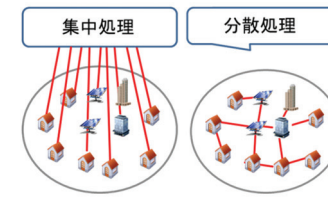


ものをつくりながら、実践的に学ぶ

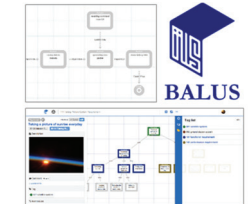
三浦 政司 (みうら まさし) <http://m-miura.jp/>



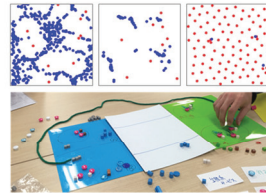
飛行体の誘導・制御



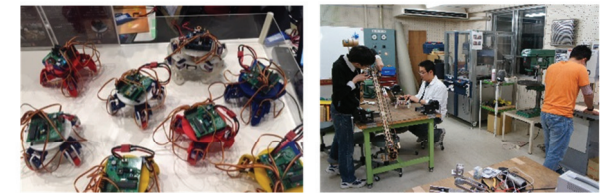
分散協調制御



システム設計プロセス



エージェントシミュレーション



PBL型工学教育

学生の自己紹介

影山 智明 (かげやまともあき)

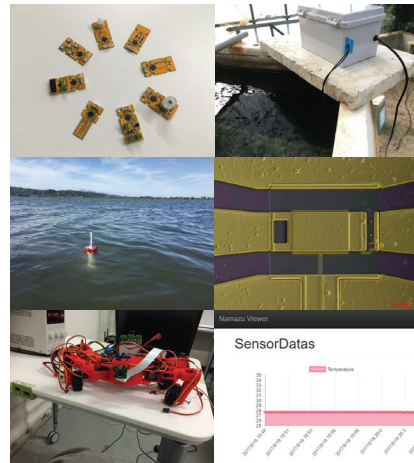
- モジュール設計、製造
- 通信機能 (TURIP) 開発

- 専門
- マイクロマシニング (MEMS)
 - センシングシステム

牧田 岳大 (まきたたけひろ)

- Scratch連携アプリ (Tucratch) 開発
- Tucratchブロック設計

- 専門
- センサデータ処理
 - Webアプリケーション



TIA KIDS SCHOOLの先生紹介

河原 央実 (かわはらひろのり)

- TIA KIDS SCHOOL講師

TIA KIDSの先生としてスクールで教えるほか、公民館などで出張プログラミング教室を実施。TIA KIDSの運営全般を担う。
趣味：パラグライダー



梶岡 大晃 (かじおかひろあき)

- TIA KIDS SCHOOL補助講師

TIA KIDSの先生であり、裏のキャラ。TIA KIDS運営補助担当。
趣味：写真、ドライブ



TIA KIDS SCHOOLの紹介

TIA
KIDS
SCHOOL

全ての子どもたちに
次の世代で
活躍できる力を。

TIA KIDS SCHOOLの紹介



子どもたちの主体性を尊重し、将来活躍できる力を育む

今日の流れ

- フィジカルコンピューティングって何？
- プログラミングの練習 (Scratch)
- 入力と出力について
- フィジカルコンピューティングにチャレンジ
- まとめ

TIA KIDS



フィジカルコンピューティングって何？

- ・身の回りにあるモノに命令を与え、動かすためのプログラムを作ることです。

(例) プログラムで動いているもの

- エアコン → 人が温度設定をすることで、エアコンに命令を送る
- テレビ → リモコンのボタンを押すことで、テレビが動く
- お風呂 → 人が水の温度を設定することで、お湯になる

クイズ

Q1. 台所にあるプログラムで動いているものは何でしょう？



クイズ (答え)

- ・ 冷蔵庫
 - ・ 電子レンジ
 - ・ IH
 - ・ 食器洗い機
- ・・・など



プログラミングって・・・

なんだかややこしい・・・。

難しそう・・・？

```
public static void main(String[] args) {
    // 1000 自動生成したユーザーを作成
    Connection conn = null;
    try {
        Class.forName("org.postgresql.Driver");
        // データベースに接続
        conn = DriverManager.getConnection("jdbc:postgresql://localhost:5432/postgres", "user", "password");
        String sql = "select user_id, user_name, birthday from user_m";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery();
        while (rs.next()) {
            String id = rs.getString("user_id");
            String name = rs.getString("user_name");
            Date birthday = rs.getDate("birthday");
            System.out.println("ID:" + id);
            System.out.println("名前:" + name);
            System.out.println("誕生日:" + birthday);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



「こんなにたくさんの文字を打たないといけないの?!」

「しかも英語だし・・・」って？

ビジュアルプログラミング

マウスだけでプログラミングしてみよう！

いきなりさっきみたいな物を作ろうなんて無理ですよね。
あんなにたくさんの文字を打つのは難しそうです。
でも、マウスでパズルのようにプログラムが作れたら・・・？
しかも英語じゃなく、日本語で。

プログラムをつくる

それができちゃうのがビジュアルプログラミング言語というもの。
というわけで、まずはマウスだけでプログラムを簡単に作ることをしてみましょう！

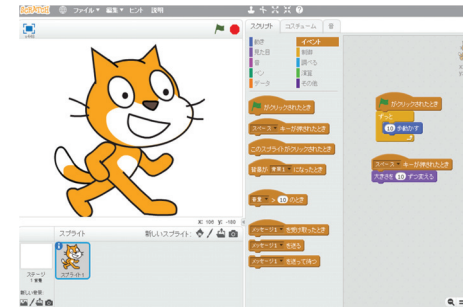
Scratchで プログラミングをしてみよう

Scratchってなに？

世界一有名なビジュアルプログラミング言語

今回使うのは、「Scratch」というビジュアル言語。

あのMIT（マサチューセッツ工科大学が開発した言語です）。



Made by MIT

さまざまな命令が書かれた
ブロックを組み合わせることで、

簡単にプログラミングを
体験することができます。

キャラクターに自由に命令できます。

Scratchでできること

Scratchで実際にどんなことができるのか見てみましょう。

ここでは、歩いて鳴かせるものを作ります。



猫をこのブロック
の通り動かす



←こんな風にブロックを組み合わせると、猫が・・・

- ・ずっと
 - ・10歩動いて
 - ・「ニャー」と言う
- プログラムになります。
実際に動かすと・・・
猫が動きながら「ニャー」と言います。

Scratchを使ってみよう

画面の説明

①ステージ (スプライトが動く場所)

②スプライト (キャラクター)

③スプライト (キャラクター) を変更できます

④ブロックパレット (命令ブロックの種類)

⑤ブロック (スプライトに命令するパーツ)

⑥スクリプトエリア (プログラムを組み立てる場所)

⑦スクリプト (Scratchで作るプログラムのこと)

ブロックの動かし方

がクリックされたとき

がクリックされたとき

基本的なブロックの使い方

失敗例

ぴったりはまらないと使えない・・・
だけど、もう1つのブロックなら!!

成功例

形が合うことでぴったりはまった!!

ブロックの外し方

がクリックされたとき

こんにちは & say

10 歩動かし

うーん・・・ & think

がクリックされたとき

こんにちは & say

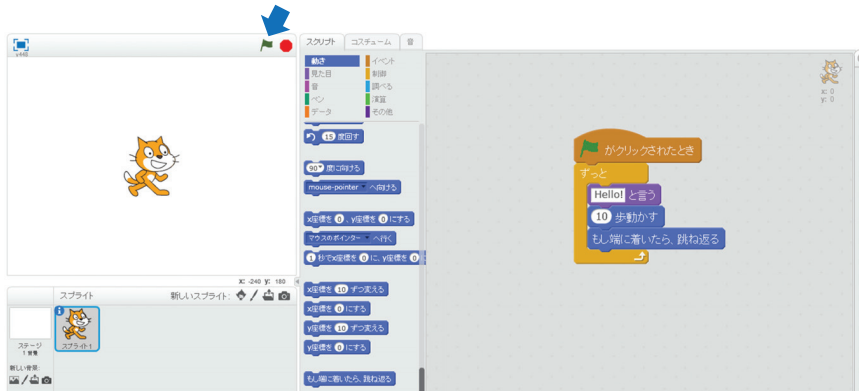
10 歩動かし


うーん・・・ & think

外したいブロックをつかんで離してみましょう!

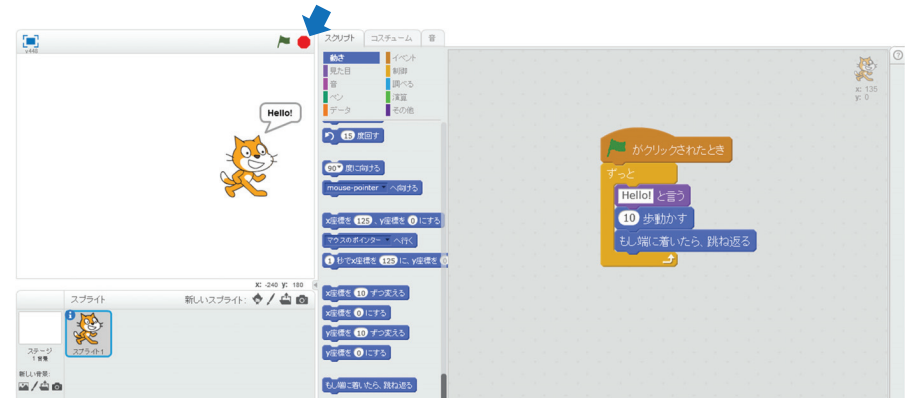
一番下のブロックをつかんだら1つだけ外れます!


ブロックを使ってみよう(実行の仕方)



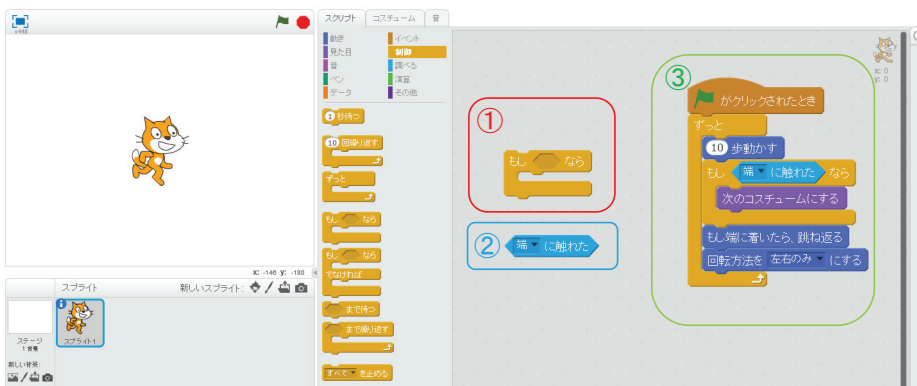
最初に行わせたいブロックを持ってきて、合わせてみましょう！
その後に  ボタンをクリックして実行しましょう！

ブロックを使ってみよう(停止の仕方)



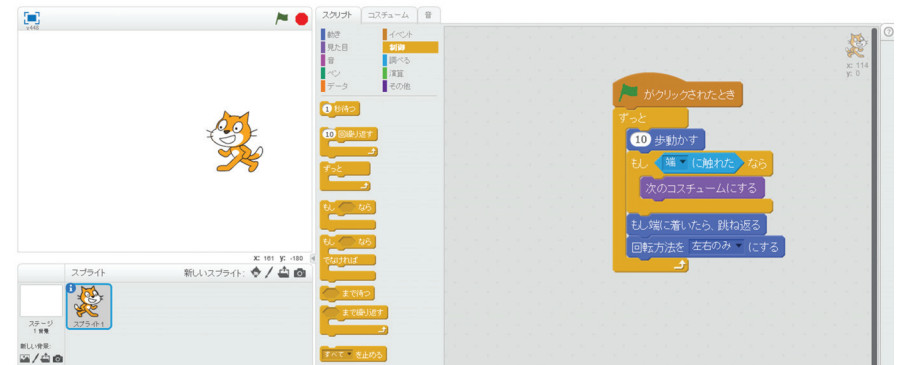
実行すると「Hello!」と言いながらスプライト君が右と左に歩き続けます。
ただこのままじゃずっと歩くので、ボタン  を押して停止させてあげましょう！



条件によって動作を変えてみよう



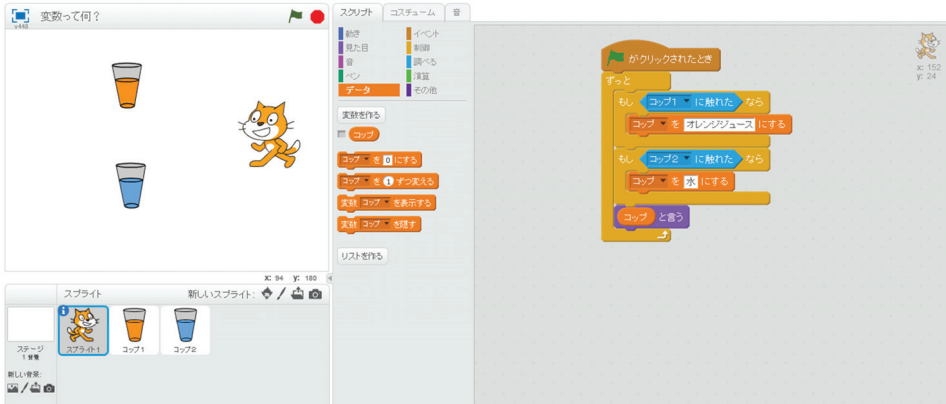
- ①のブロックは条件によって動きが変わるブロックです。「もしもブロック」と呼びましょう。
- ②のブロックは条件の内容を決めるブロックです。今回は「端」に触れたら...という条件を書きましょう。
- ①と②を合わせて使ったものが、③のプログラムとなります。
内容は「スプライト君が画面の端(はし)まで歩くと向きが逆になり、また歩き続ける動作を行う」となっています。

条件によって動作を変えてみよう



前の画面と見比べて見ましょう。
スプライト君が画面の端に触れたことで、見た目が変わり、向きが逆になっていますね。
このように  にある  このブロックを使えば、スプライト君を自分の考えたとおりに動かせます。
この他にも色々なブロックがあるので、試してみましょう！

変数を使った動作



ここでは変数を使ったプログラムを見てみましょう。
皆さん、右のブロックを見たときに「どうなっているのか良く分からないなあ...」と思いませんか？
でも大丈夫！ 変数を知ればこのプログラムが簡単だと、すぐ思えるようになりますよ！

変数って何？

- 変数とは何らかの値をしまう**入れ物**のようなもの

身の回りのものに例えると・・・

例：コップ

コップという**入れ物**の中にジュースという値が入る



← 同じコップ(入れ物)でも中身(値)が違いますよね？

変数を使ったプログラムの説明①



もしもブロックの中に条件を入れましょう。今回は「コップ1に触れたなら」という条件を付けてあげて、コップの中身をオレンジジュースにしましょう。
同じように、コップ2に触れたら中身を水にしましょう

変数を使ったプログラムの説明②



最後にコップの中身を言う動きをさせましょう。
ここではコップはオレンジジュースか、水のどちらかになりますよね？
なぜならコップという**入れ物**にオレンジジュースか水という**値**が入っているからです。

変数を使ったプログラムの説明③

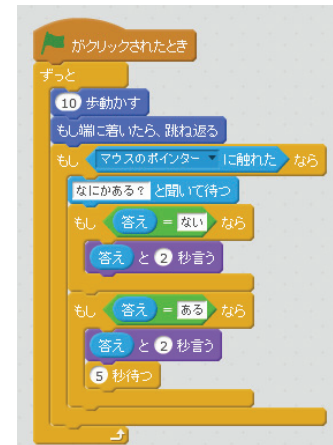
それではどんな動きをするのか確認してみましょう。



左の図ではコップという変数(入れ物)にオレンジジュース(値)が入っているため、オレンジジュースと言います。

右の図ではコップという変数(入れ物)に水(値)が入っているため、水と言います。

自動運転のようなネコのプログラム



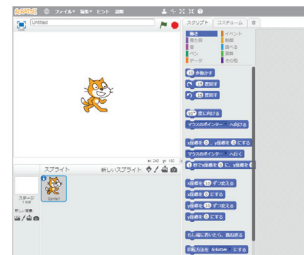
- ・ネコが10歩動く。
- ・マウスポインター(なにか)に触れた(感知した)ときになにかあるのかを調べる。
- ・もしも、調べた結果なにも「ない」場合は、「ない」と2秒言う。
- ・もしも、調べた結果なにか「ある」場合は、5秒待つ。

※答えのブロックは中身が変わる変数のブロックです！

家でもScratch!

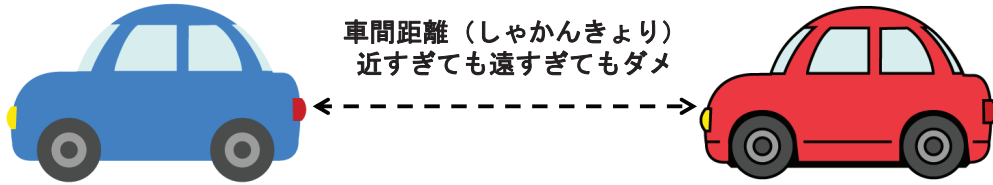
このページまでの内容はうちのパソコンでもチャレンジできます。

- ①【URL】 <https://scratch.mit.edu/> ←このURLで検索!
- ②「やってみる」をクリックすると、Scratchの画面が出てきます。



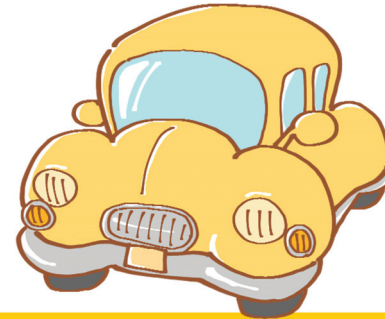
今日、学んで欲しいこと 「入力」と「出力」

自動運転車を考えてみよう



人間が運転する場合

- ①目で距離をはかって
- ②頭で「近すぎるからスピードを落とさなければ」と考えて

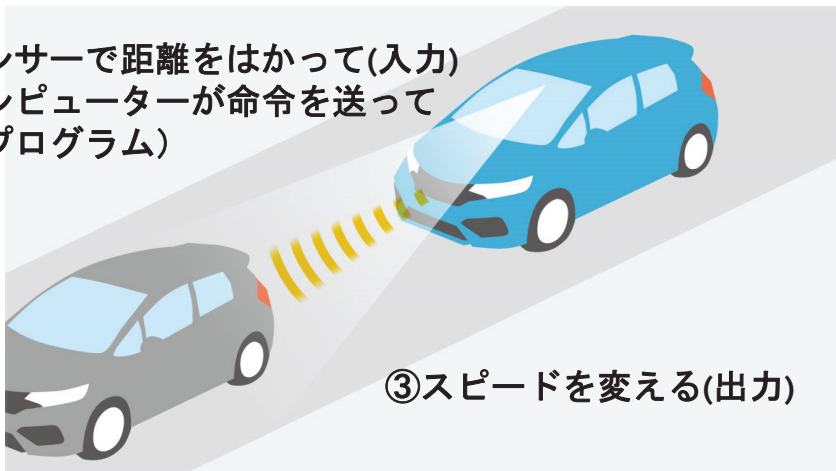


- ③ブレーキを踏む



自動運転車の場合

- ①センサーで距離をはかって(入力)
- ②コンピューターが命令を送って(プログラム)



入力と出力

運転手

<入力>
目ではかった
距離



脳
(考える)



<出力>
ブレーキやアクセルを踏む
(足の筋肉)

自動
運転
車

<入力>
センサーではかった
距離



コンピューター
(プログラム)



<出力>
スピードを変える
(例: モーターのはやさを変える)

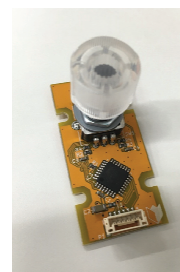
みなさん自身の行動の
入力と出力の例を考えてみよう！

身の回りのもの（家電など）で
入力と出力の例を考えてみよう！

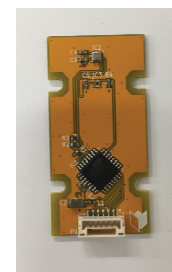
Tucratch (TURIP for Scratch)
をつかってみよう

Tucratchとは？

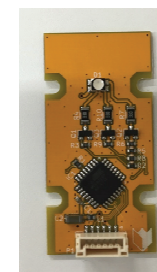
Scratchでプログラミングすることで
温度や明るさを測ったり、モーターを動かしたりすることができます



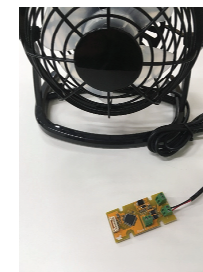
ボタン&つまみ



環境センサー
(温度、気圧、明るさを測る)



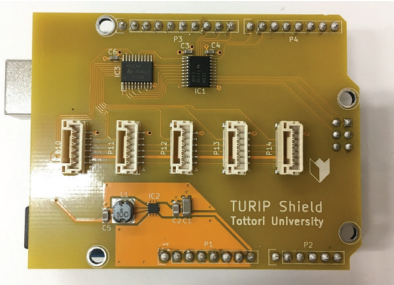
LEDライト



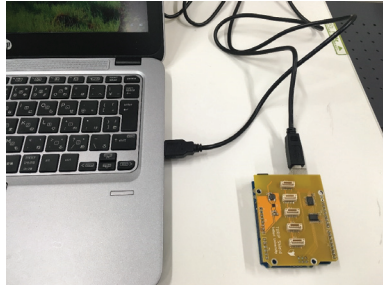
扇風機
(モーターをまわす)

Tucratchをつかう準備①

まず、「TURIPシールド」をパソコンにつなぎます



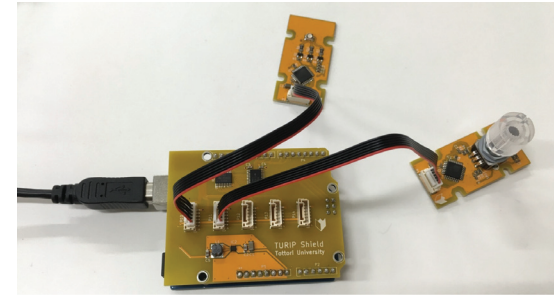
TURIPシールド



PCとTURIPシールドを接続

Tucratchをつかう準備②

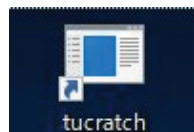
「TURIPシールド」に「TURIPモジュール」をつなげます。
まずは「ボタン&つまみ」と「LEDライト」のモジュールをつなげましょう。



TURIPシールドにTURIPモジュールをつないだようす

Tucratchをつかう準備③

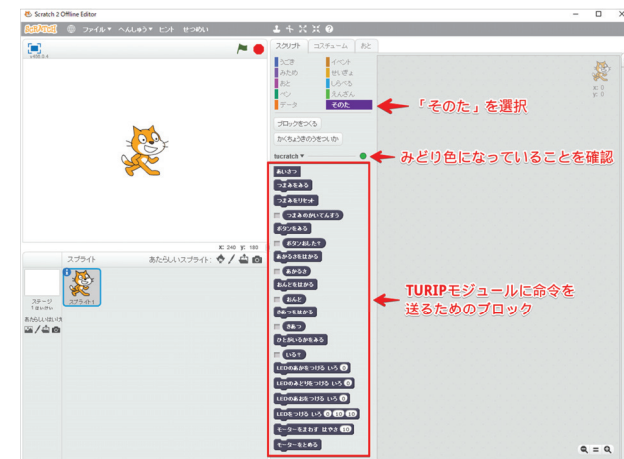
①デスクトップにある「tucratch」のアイコンをダブルクリック



②tucratchというウィンドウで「COM0」を選択



Tucratchをつかう準備④



練習 1. 赤いライトをつける

- ① 「LEDのあかをつける」というブロックをつかうとLEDライトが赤色にひかります。
明るさを0~255の数字で指定することができます。
※10くらいでも十分に明るくなります



- ② ライトをつける命令だけだと、ずっとつけっぱなしになってしまうので、明るさで0を指定することでライトを消してみよう
※「○秒待つ」のブロックは「制御」のところにあります



練習 2. ボタンを押している間、赤いライトがつく

- ① 「ボタンをみる」のブロックを実行した後に「ボタンをおした？」のブロックに入っている値をしらべるようにします。
ボタンを押している間「ボタンをおした？」には、「True」が入ります。
※押していない間は「False」が入ります。
- ② 「ボタンをみる」を「ずっと」の中に入れることで常にボタンが押されたかどうかをチェックするようにします
- ③ 「もし〜でなければ」のブロックを使うことで、ボタンが押されたときと押されていないときの命令を変えます



練習 3. ボタンを押している間は赤色に、 押していないときは白色に光る

ヒント：
「あか」「みどり」「あお」が全部同じ強さで混ざったとき、白色に光ります

他にも...

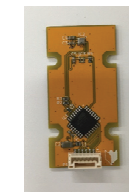
- ・ 黄色：「あか」と「みどり」が同じ強さで「あお」がゼロ
- ・ 紫色：「あか」と「あお」が同じ強さで「みどり」がゼロ
- ・ いろんな色を試してみよう！



↑ どちらでもOK ↓



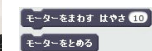
他のモジュールもつないでみましょう



環境センサ



扇風機



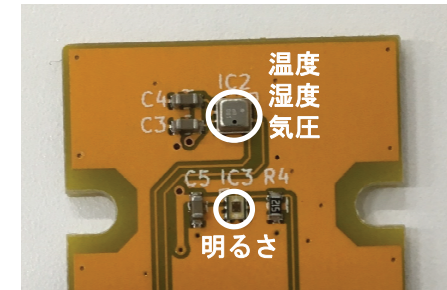
※はやくさは0~255で指定する

センサーの値を表示する

「あかるさ」や「おんど」などのブロックにチェックを入れるとそれぞれの値が表示されるようになります。



環境センサについて



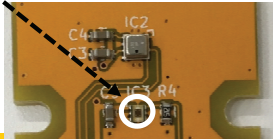
センサー、つまみ、ライト、
モーター(扇風機)を試してみよう

<チャレンジ1>

つまみをまわすことで回転のよさを調節できる扇風機をつくってみよう！

<チャレンジ2> 明るいときだけ動く扇風機を つくってみよう！

- 明るいときはつまみをまわすことで回転のよさを調節できるようにしておこう
- 暗くなったら止まって、つまみをまわしても動かないようにしておこう
- 明るさをはかる部分を指で隠すことで、暗くすることができるよ



<チャレンジ3> 自分だけのオリジナルの装置を つくってみよう！

例えば：

- 気温が上がるとまわりはじめる扇風機
- 気温が高くて暗いとき（ねぐるしい夜）に自動でまわる扇風機
- つまみの回転や気温を色で表す機能がついている扇風機
- 一度ボタンをおすと(指を離しても)まわりはじめて、もう一度ボタンをおすと止まる扇風機（難しい！）
- 運にまかせて動いたり止まったりする扇風機
※「演算」の「0から□までの乱数」ブロックを使ってね

ブロックの説明

イベント スクリプト

- 旗がクリックされたとき 旗がクリックされたときに動作を開始する。
- スペース キーが押されたとき 指定したキーを押したときに動作を開始する。
- このスプライトがクリックされたとき 画像がクリックされたときに動作を開始する。
- 音量 > 10 のとき 音量が指定した数字より大きいとき動作を開始する。
- メッセージ1 を受け取ったとき メッセージ1を受け取ったとき動作を開始する。
- メッセージ1 を送る メッセージ1を送る。

ブロックの説明

動き スクリプト

- 10 歩動かす 画像を10横に動かす
- x座標を 0、y座標を 0 にする 画像の位置を自分が指定した数字にする
- x座標を 10 ずつ変える 画像の横の位置だけ変更する
- y座標を 10 ずつ変える 画像の縦の位置だけ変更する
- もし端に着いたら、跳ね返る 画像が端に行ったら跳ね返る
- 回転方法を 左右のみ にする 画像の回転方向を決める

ブロックの説明

見た目 スクリプト

Hello! と 2 秒言う	画像に指定した秒数だけ好きな文字を吹き出させる。
Hello! と言う	画像に文字を吹き出させる。
表示する	画像を表示する。
隠す	画像を隠す。
大きさを 10 ずつ変える	画像の大きさを指定した数字ずつ変える。
次のコスチュームにする	画像を変える。

ブロックの説明

音 スクリプト

pop の音を鳴らす	popな音を鳴らす。
終わるまで pop の音を鳴らす	終わるまでpopな音を鳴らす。
1 のドラムを 0.25 拍鳴らす	ドラム1を0.25拍鳴らす。
楽器を 1 にする	楽器1にする。
音量を -10 ずつ変える	音量を指定した数字ずつ変える。
テンポを 20 ずつ変える	テンポを指定した数字ずつ変える。

ブロックの説明

制御 スクリプト

1 秒待つ	1秒待つ。
ずっと	ブロックに囲まれた動作をずっと行う。
もし ×× なら でなければ	もし××なら○○をして、××でなければ△ △ をする動作を指定する。
×× まで待つ	××するまで待つ。
×× まで繰り返す	××するまで繰り返す。

ブロックの説明

調べる スクリプト

「調べる」のブロックは他のブロックに組み込んで使うのはほとんどです。

端 に触れた	「端に触れた」かどうかを調べる。
色に触れた	「指定した色に触れた」かどうかを調べる。
Spaceship までの距離	「指定したもののまでの距離」を調べる。
What's your name? と聞いて待つ	指定したコメントを聞いて動作を待つ。
スペース キーが押された	指定したキーが押されたかどうかを調べる。
マウスが押された	マウスが押されたかどうかを調べる。