

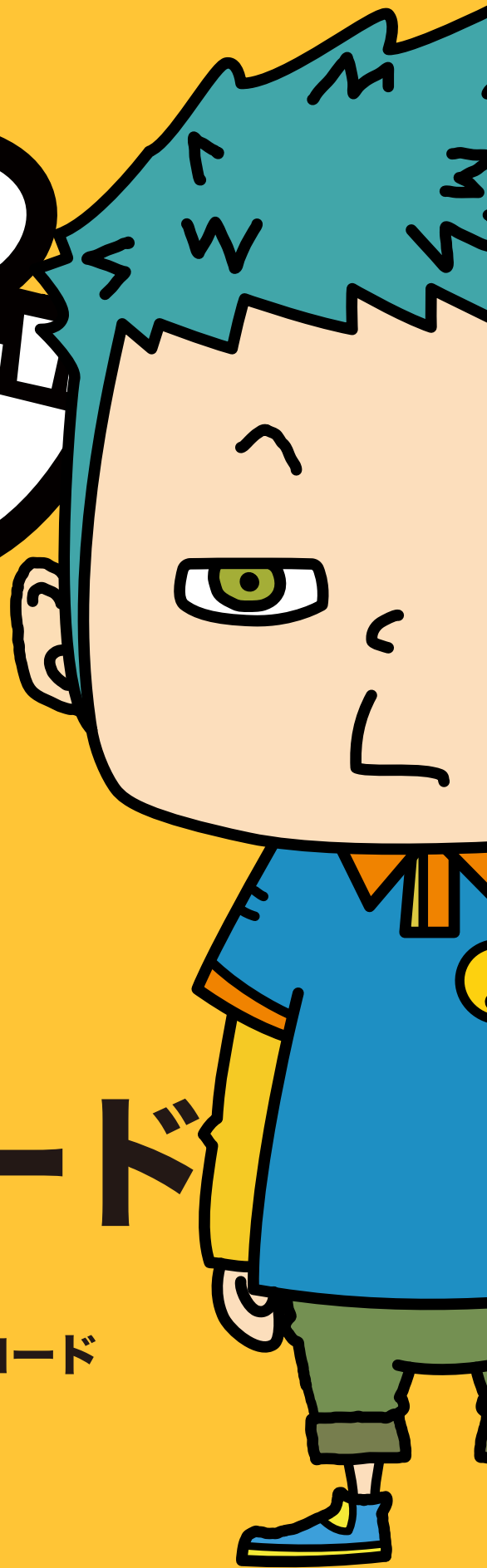
月刊 みんなの タイム

特別企画

はじめての オレンジボード

ピックアップセンサー タッチセンサー

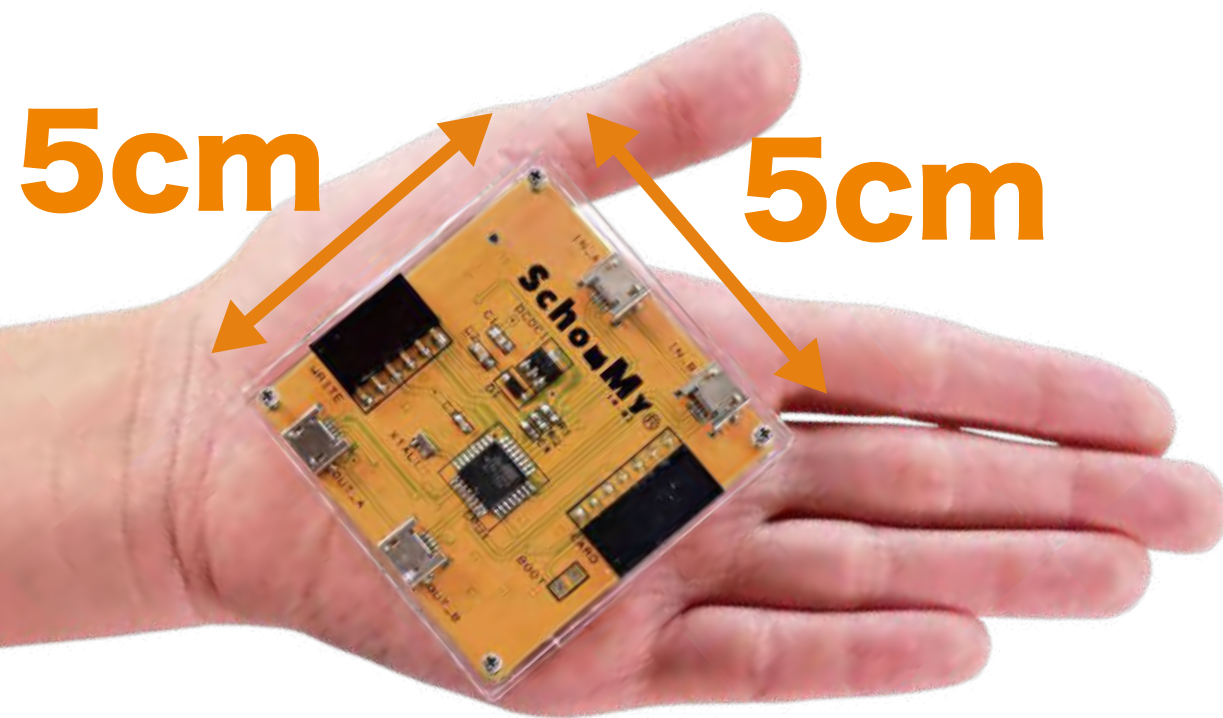
スクーミーブロックエディタのダウンロード
動かない?と思ったらのマニュアル
タッチしたらLEDが光る
いろんなプログラムを使ってみよう



スクーミーオレンジボード

単四電池1本で動き、センサーやアクチュエーターを選んで、
つなげるだけで動く、小型コンピューターです。
動作を指示するプログラムも、指令ブロックを組み合わせるだけで
生成できるため、素早く簡単に思い通りの動作をさせることができる IoT ツールが「スクーミー」です。

小学校・中学生といった初心者から、高等学校の科目となった「情報Ⅰ」「情報Ⅱ」「総合的な探究の時間」で応用的に
活用されるなど、身の回りの課題から社会全体の課題解決まで、さまざまな場所で使われるツールになっています。



書き込み機（書き込み機延長ケーブル）



スクーミーオレンジボードとパソコンを繋げて利用します。
パソコンでプログラムを組み、そのプログラムをスクーミーオレンジボードに書き込むためのツールです。
電源も供給するため USB から電源を取ることもできます。
Unity や他のサービスと連携するとき（シリアル通信）も書き込み機を使うことで手軽に通信させる機能を持っています。
書き込み機延長ケーブルを利用することでデータを取得するときやゲームのコントローラなどで利用することができます。



センサーコネクタ

USB タイプで半田づけや電気回路の知識がなくてもつなげるだけで利用することができます。
様々な種類のセンサーコネクタがあり、1つのボードに同じものをつなげることもできますし、異なったものをつなげることもできます



はじめまして。 オレンジボードです。

School Myself。
これが、私たち SchooMy（スクーミー）の語源です。

こうだったら良いのにな！とか、
こんなこと出来ないのかな？と学校の休み時間に何気なく話している
ことを、そこに行って、すぐに試してみることが出来るモノとして、
このスクーミーボードは生まれました。

誰もが、身の回りにたくさんある小さな気づきや発見したことをそのままにしないで、電池や電球を取り換えるくらい簡単に、試して、改善しに行くことができるツールがこのスクーミーボードです。



スクーミーブロックエディタ

ブロックを組み立てて直感的にプログラミングをすることができます。つくったプログラムを保存して開くこともできますし、ウェブページからダウンロードしたプログラム（sbp ファイル）をここで開くこともできます。シリアルモニタ・グラフモニタを利用することで、センサーコネクタの値をすぐにデータとして見るすることができます。

スクーミー IDE

スクーミーボードに書き込むプログラムをつくる場所のことです。スクーミーブロックエディタでブロックを組み立てて、アップロードをおすことで、IDE 上で言語が生成されます。

測って、比べる。 すぐに試すことができます。

プログラムが書き込み機を通じてスクーミーボードに書き込まれていくのです。もちろん、言語を書きかえてプログラミングすることもできます。

- 本体の大きさは、縦 5 センチ、横 5 センチ
- センサーコネクタは利用者のやりたいことがすぐに試せるよう、温度センサーや明るさセンサーなど 50 種類以上が用意されている
- センサーおよびアクチュエーターを microUSB の端子接続で同時に最大 4 種類接続できる

- センサーやアクチュエーターは半田付けや電気回路の知識がなくても簡単に差し替えができる
- 給電は小型コンピュータに単四電池ソケットが付随しており、電池で動かす際に配線が必要としない
- 給電は本体に内蔵の単四電池での給電以外に、コンピュータと接続することで microUSB 経由で給電、および AC アダプタからの給電も可能である
- スクーミー本体にネジ穴が空いていることで、制作物を板などと固定することができる
- スクーミーを用いて取得された温度データなどの情報は、スクーミー本体に取り付けた SD カードなどの外部記憶装置との接続により保存が可能
- 作成されたプログラムはスクーミー本体および学習端末に保存することができる。（なお、スクーミー本体に保存できるプログラムは 1 つ）



コネクタ延長ケーブル

センサーコネクタをのばして使うことのできるコネクタ専用の延長ケーブルです。



【動作環境】

- Windows、MacOS、Chromebook に開発環境のアプリケーションをインストールすることができ学習端末に対応できる
- 屋外でも利用できるように、開発環境がオフラインでも利用できる
- 開発言語は、独自のオリジナルのビジュアル言語でも可とするが、C 言語、Python、Javascript、に変換が可能である
- 開発環境であるブロックエディタは、日本語対応のほか、英語における学習にも対応できている
- 小型コンピュータの動作は、学習端末で作成したプログラムを Bluetooth 通信の接続で行うのではなく、小型コンピュータに内蔵されている IC チップに学習端末で生成したプログラムソースコードをアップロードし、動作させる

今、スクーミーボードは、教室から外に飛び出して行く子どもたちと一緒に、体育館や校庭、公園から畑、そして自宅やおじいちゃん家、病院や会社といった、あらゆる場所で多くの人に使われ始めています。

小学生が探究するためのコンピュータとして生まれたスクーミーボードが今や、半田付けもおこなずに、そのまま使える便利さが多くの方に受け入れられ、自ら、家庭や地域を改善するツールとして、世の中に新しく広まってきているのです。

スクーミーは、つくることで、すぐ解決できる世界をつくります。



スクーミーブロックエディタのインストール

時間の目安 30分

スクーミー IDE | 総合開発環境 とは

スクーミー IDE は、スクーミーボードにプログラミングをするためのソフトウェアです
パソコンにスクーミー IDE とドライバをインストールして、スクーミーボードを使うための準備をしましょう

01 インターネットで「スクーミー」と検索して、スクーミーの公式サイトにアクセスします



02 スクーミーの公式サイト右上にある「スクーミーボードを使う」をクリックします

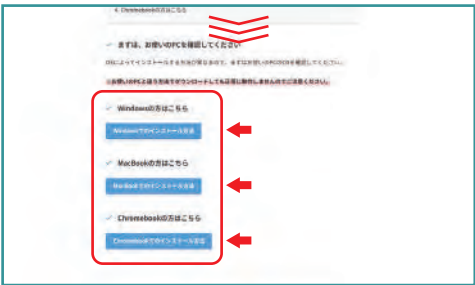
03 「スクーミーボードを使う」をクリックすると、「スクーミー IDE、スクーミーブロックエディタのインストール方法」というハロースクーミーのページが開きます



この画面が開ければ OK

04 ページを下にスクロールすると、OS 別のインストールページへ移動できます
自分のパソコンの OS に合った版の「インストール方法」をクリックします

※自分のパソコンの OS を調べておきましょう



05 インストール方法をクリックすると、インストールページが開きます

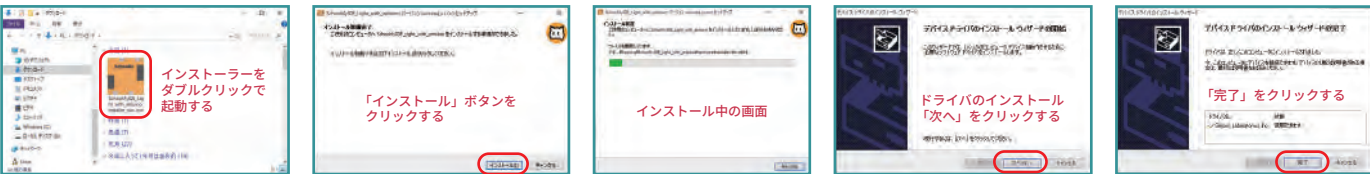
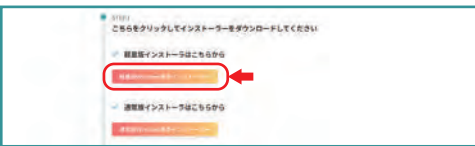
06 インストールページに移動したら、手順を読みすすめるためにスクロールします
インストーラーのダウンロードボタンをクリックしましょう

※インストーラーには「通常版」と「軽量版」があります
授業や部活動で使用する場合は、軽量版をおすすめします

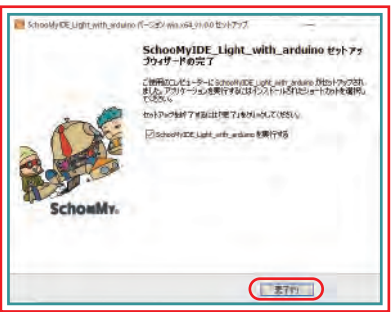
07 インストーラーのダウンロードができれば
インストーラーを起動します（画像は Windows のものです）

※セキュリティソフトからの警告がでる場合がありますが
問題ありませんので、実行してください

インストーラーの指示に従って、
スクーミー IDE とドライバーのインストールしましょう



08 左の画面が表示されたらインストールができました
「完了」ボタンをクリックして、スクーミー IDE が起動するのを確認しましょう



Windows の場合は、デスクトップ画面に
右の 2 つのアイコンができていれば完了です

これで開発環境を準備することができました！



LEVEL
01

スクーミーボードを使うための準備

スクーミーを使うための準備をする

スクーミーボードについて知ろう

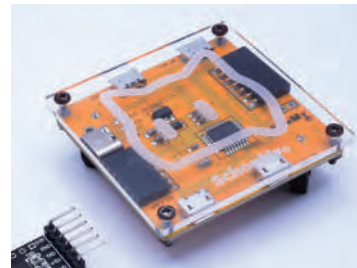
難易度 ★☆☆ 時間の目安 05 分

目標 スクーミーボードの方向を覚える

SchooMy Start Guide Book は、初めてスクーミーボードを使う人向けのガイドブックです
スクーミーボードやブロックエディタについて知り、スクーミーボードを使うための準備をしましょう

01 スクーミーボードを手にとってみましょう

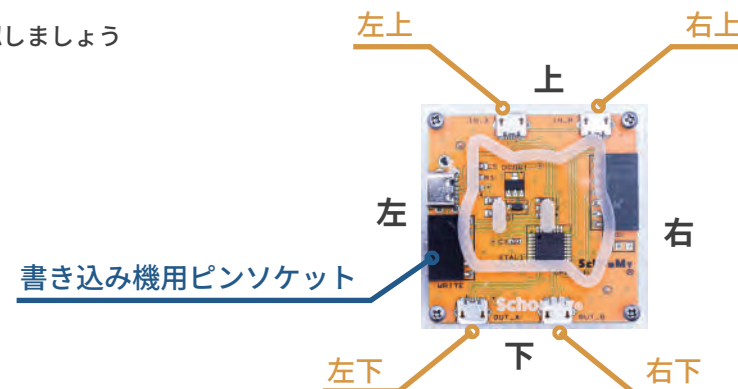
スクーミーボードは、5cm 四方の小型コンピュータ スクーミーボードであり、
プログラムを書き込み、センサーを取り付けることで使うことができます



02 スクーミーボードには方向があります

右の画像を見ながら、スクーミーボードの方向を確認しましょう

- スクーミーボードの表と裏、上下左右の方向
- 左上、右上、左下、右下の micro USB の場所
- 書き込み機用ピンソケットの場所



ここまでできたら LEVEL01 クリア！

スクーミーボードに触れて、
スクーミーボードの方向を理解できた

NEXT ▶▶▶ スクーミーを使うためにパソコンに環境をつくる

LEVEL
02

SchooMyIDE とブロックエディタの立ち上げ

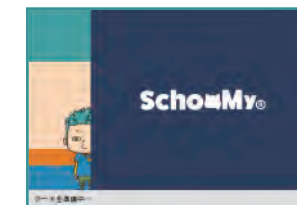
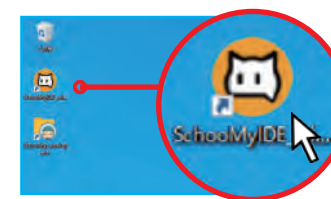
スクーミーを使うためにパソコンに環境をつくる

パソコンでスクーミーブロックエディタを開こう

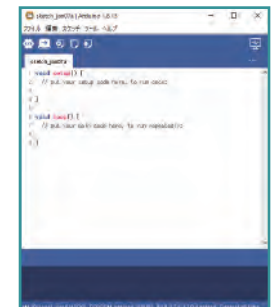
難易度 ★☆☆ 時間の目安 05 分

目標 スクーミーブロックエディタを開いて、画面に見やすく配置する

01 SchooMyIDE_with_arduino のアイコンをダブルクリックして SchooMyIDE を起動します

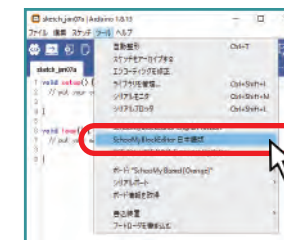


SchooMyIDE 起動中の画面

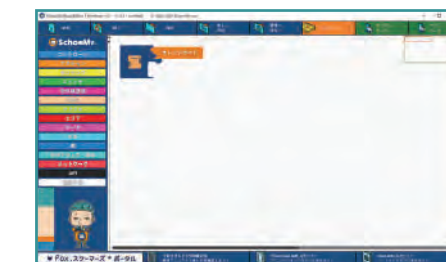


SchooMyIDE の画面

02 SchooMyIDE の [ツール] メニューから [SchooMyBlockEditor 日本語版] をクリックしてブロックエディタを起動します



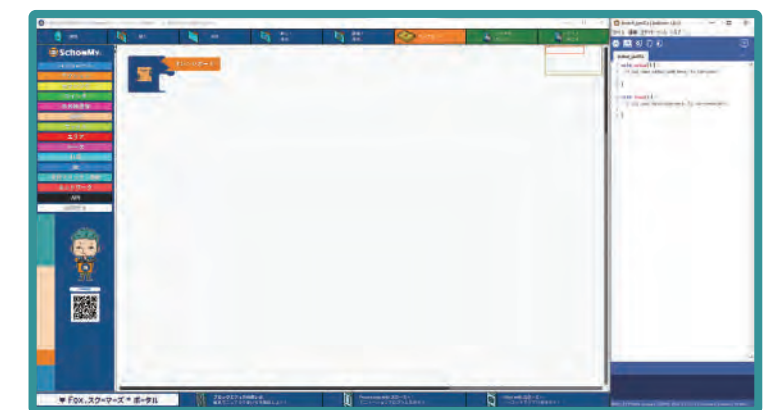
SchooMyIDE の画面



ブロックエディタの画面

03 ウィンドウの位置と大きさを調整して、右画像のような配置にします

SchooMyIDE は横幅を狭くして右側に、
ブロックエディタは大きくして左側に配置します



ここまでできたら LEVEL02 クリア！

スクーミーブロックエディタを開いて、
画面に見やすく配置できた

NEXT ▶▶▶ スクーミーボードとパソコンを接続する

LEVEL 03

スクーミーボードとパソコンの接続

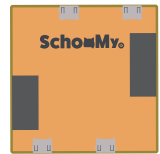
スクーミーボードとパソコンを接続する

スクーミーボードとパソコンを繋げて動作を確認しよう

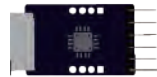
難易度 ★☆☆ 時間の目安 10 分

目標 スクーミーボードとパソコンを接続し、接続されたことを確認する

準備するもの



スクーミーボード



書き込み機



パソコン または タブレット

01 スクーミーボードに書き込み機を挿しこみます



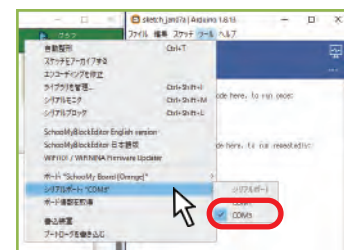
⚠ 書き込み機の挿す場所、挿す方向に気をつけましょう

02 パソコンに書き込み機を挿しこみます



スクーミーボードとパソコンを接続すると、書き込み機が光ります

03 SchooMyIDE の [ツール] メニューから [シリアルポート] にカーソルを合わせます
[COM〇] や [〇〇usbserial〇〇] などが表示され、✓ が付いていることを確認します



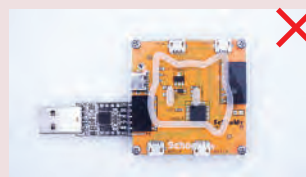
⚠ 正しく表示されない場合や、書き込みがうまくいかない時は
スクーミーボードと書き込み機がしっかりつながっているかを確認しましょう



書き込み機を挿す場所を間違えている



書き込み機を挿す向きを間違えている



書き込み機のピンの位置がずれている

ここまでできたら
LEVEL03 クリア！

スクーミーボードとパソコンを接続し、
シリアルポートが表示された

NEXT ▶▶▶ シリアルモニタに文字を表示させる

LEVEL 04

スクーミーボードでシリアルモニタを使った表示

シリアルモニタに文字を表示させる

シリアルモニタの使い方を知らう

難易度 ★☆☆ 時間の目安 10 分

目標 シリアルモニタを開いて、文字を表示することできる

01 ブロックエディタを開き、[表示する] タブの
シリアルモニタブロックを出してつなげます

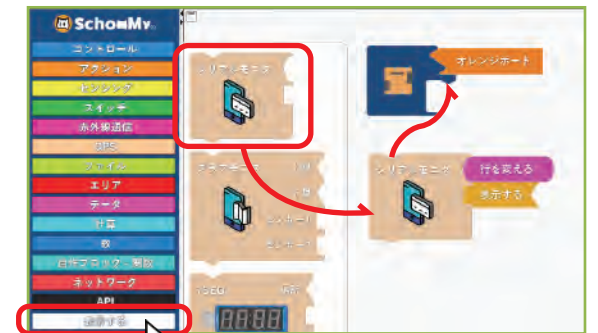
ブロックの凹凸が合うようにつなげましょう
うまくブロックがつながるとカチッと音になります



02 ブロックエディタの [アップロード] をクリックして、
ボードへ書き込みます



ボードへの書き込みが完了しました。



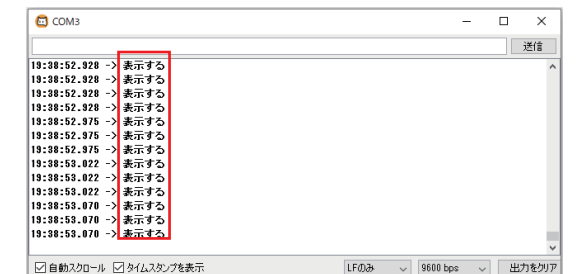
アップロード時に下のようなエラーが出る場合があります
「スクーミーボードが動かない場合の対応について」のページをご覧ください

スキッチの書き込み中にエラーが発生しました エラーメッセージをコピーする

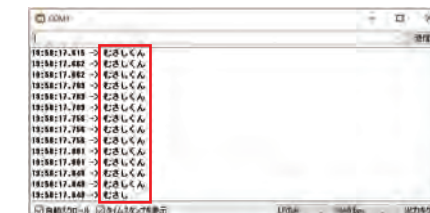
03 ブロックエディタの [シリアルモニタ] をクリックして、
シリアルモニタを出します



「表示する」と文字列が表示されているのを確認しましょう
確認したら、シリアルモニタの [×] ボタンをクリックして閉じます



04 シリアルモニタブロックの黄色い部分をクリックして、文字を打ちこんでみましょう
[アップロード] して、[シリアルモニタ] に表示される文字が変わったのを確認しましょう



ここまでできたら
LEVEL04 クリア！

シリアルモニタに文字を表示できた
表示する文字を変えることができた

NEXT ▶▶▶ 土壌水分センサーの反応を見る

LEVEL 05

スクーミーボードでタッチセンサーを使ったセンシング タッチセンサーの反応を見る

タッチセンサーの使い方を知ろう

難易度 ★★☆☆ 時間の目安 10 分

目標 タッチセンサーに触れたり離したりして、シリアルモニタの値が 0 と 1 で変化するのを確認する

- 01 スクーミーボードの左上に
タッチセンサーを挿します



準備するもの

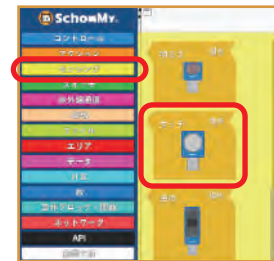
タッチセンサー



- 02 ブロックエディタを開き、[表示する] タブの
シリアルモニタブロックを出してつなげます



- 03 [センシング] のタブからタッチセンサーブロックを出して、
シリアルモニタブロックの右側につなげます



ブロックの凹凸が合うようにつなげましょう
上のようにブロックが繋がれば完成です

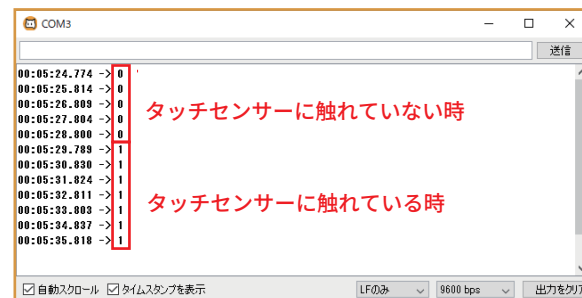
- 04 ブロックエディタの [アップロード] をクリックして、ボードへ書き込みます
ブロックエディタの [シリアルモニタ] をクリックして、シリアルモニタを出します



- 05 タッチセンサーに触れるとシリアルモニタの数値が 1 になり、
タッチセンサーから指を離すと数値が 0 になるのを確認しましょう

ここまでできたら LEVEL05 クリア！

タッチセンサーに触れたり離したりして、
シリアルモニタの数値が変化するのを確認できた



NEXT ▶▶▶ タッチセンサーに触れたら LED が光るようにする

LEVEL 06

タッチセンサーと条件分岐を使った LED のコントロール タッチセンサーに触れたら LED が光るようにする

もし～なら～ブロックの使い方を知ろう

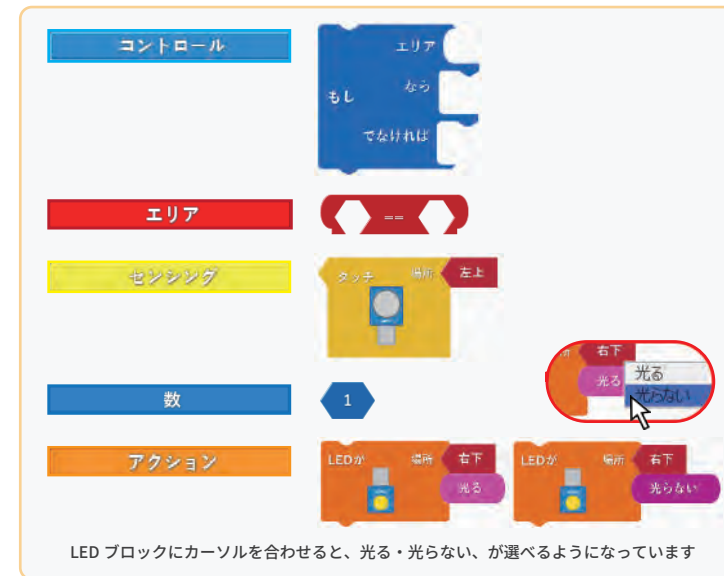
難易度 ★★☆☆ 時間の目安 15 分

目標 タッチセンサーに触れたら LED が光り、離したら LED が消えるようにする

- 01 スクーミーボードの左上にタッチセンサー、右下に LED が挿さっていることを確認しましょう



- 02 それぞれのタブのメニューから、以下のブロックを出します



- 03 下の画像のようにブロックを並べます

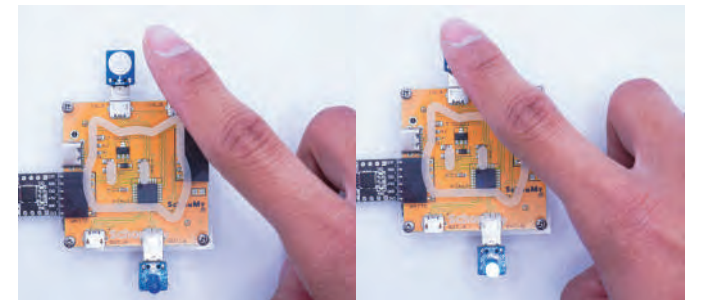


- 04 ブロックエディタの [アップロード] をクリックして、ボードへ書き込みます

- 05 タッチセンサーから指を離している間は LED の光が消え、
タッチセンサーに触れると LED が光るのを確認しましょう

ここまでできたら LEVEL06 クリア！

タッチセンサーに触れたり離したりして、
LED が光ったり消えたりするのを確認できた



NEXT ▶▶▶ 明るさセンサーの反応を見る

LEVEL 07

明るさセンサーを使ったアナログ値のセンシング 明るさセンサーの反応を見る 明るさセンサーの使い方を知ろう

難易度 ★★☆☆ 時間の目安 10 分

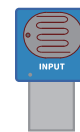
目標 明るさセンサーの周りの明るさを変えて、シリアルモニタの数値が変化するを確認する

01 スクーミーボードの右上に明るさセンサーを挿します



準備するもの

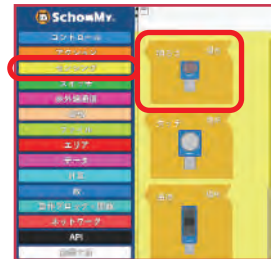
明るさセンサー



02 ブロックエディタを開き、[表示する] タブのシリアルモニタブロックを出してつなげます



03 [センシング] のタブからタッチセンサーブロックを出して、シリアルモニタブロックの右側につなげます



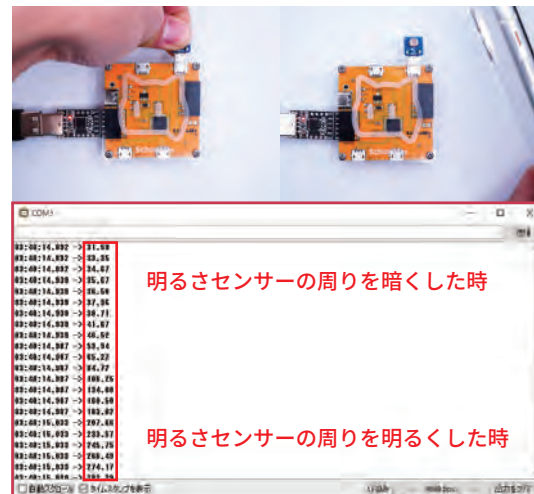
明るさセンサーブロックの場所をクリックして右上を選択します
上のようにブロックが繋がれば完成です

04 ブロックエディタの [アップロード] を押して、ボードへ書き込みます

05 書き込みが完了したら、[シリアルモニタ] をクリックします
明るさセンサーの周りを手で覆って暗くしてみたり、
スマートフォンのライトで照らして明るくしてみましょう
モニタの数値が変化するのを確認しましょう

ここまでできたら LEVEL07 クリア！

明るさセンサーの周りの明るさを変えて、
シリアルモニタの数値が変化するのを確認できた



NEXT ▶▶▶ 暗くなったら LED が光るようにする

LEVEL 08

明るさセンサーと条件分岐を使った LED のコントロール 暗くなったら LED が光るようにする センサーや LED、ブロックエディタを使いこなそう

難易度 ★★☆☆ 時間の目安 15 分

目標 明るさセンサーの周りが暗くなったら LED が光り、明るくなったら LED が消えるようにする

01 スクーミーボードの右上に明るさセンサーを挿し、スクーミーボードの左下に LED を挿します

02 それぞれのタブのメニューから、以下のブロックを出します



明るさセンサーブロックの場所にカーソルを合わせると、場所が選べるようになっています

数ブロックをクリックすると、半角数字で数値を入力できるようになっています

03 下の画像のようにブロックを並べます



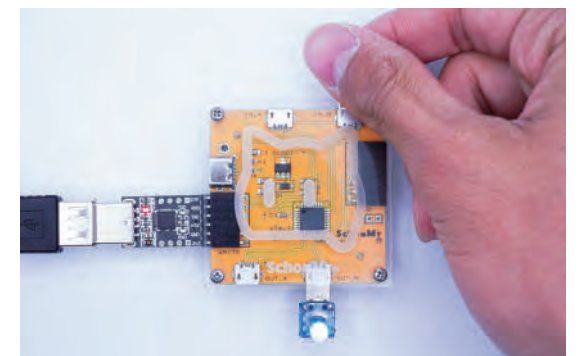
上のようにブロックが繋がれば完成です

04 ブロックエディタの [アップロード] をクリックして、ボードへ書き込みます

05 明るさセンサーの周りを手で覆って暗くしている間は LED の光が消え、
明るさセンサーの周りを明るくすると LED が光るのを確認しましょう

ここまでできたら LEVEL08 クリア！

明るさセンサーの周りを暗くしたり明るくしたりして、
LED が光ったり消えたりするのを確認できた

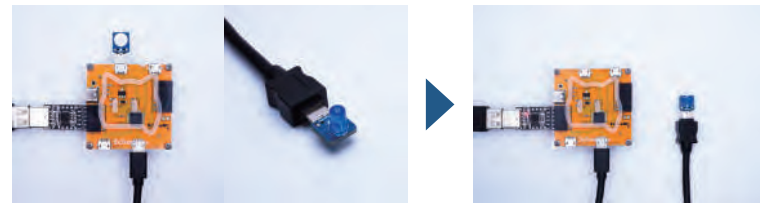


NEXT ▶▶▶ スクーミーボードの活用方法

スクーミーボードの活用方法

01 コネクタ延長ケーブルを使う方法

コネクタ延長ケーブルを使うことで、スクーミーボードから離れたところでセンサーを使うことができます
ものづくりをするときに使ってみましょう



02 スクーミーボードを単四電池で動かす方法

スクーミーボードの裏面にある電池ボックスに単四電池を入れることで、パソコンに接続せずにスクーミーボードを動かすことができます

- スクーミーボードに書き込まれたプログラムが動いていることを確認します
- パソコンから書き込み機を抜き、スクーミーボードからも書き込み機を抜きます
- スクーミーボードを裏返し、プラスマイナスの向きに従って単4電池を取り付けます
- スクーミーボードの裏面のスイッチを [BOX] にするとスクーミーボードが電池で起動します



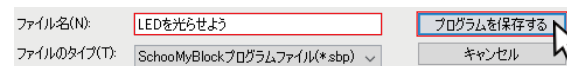
※プログラムをなおすためにパソコンに再接続するときは、必ずスイッチが [PRG USB] に入っていることを確認してください

03 作ったプログラムをパソコンに保存する方法

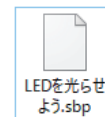


[新しく保存] をクリック

※上書き保存の場合は [保存] をクリック



ファイル名を入力して [プログラムを保存する] をクリック



.sbp ファイルが完成

04 保存したプログラムを開く方法



[開く] をクリック



.sbp ファイルを選択して [プログラムを開く] をクリック

05 新しいプログラムを作成する方法

新しいプログラムを作成するときは、ブロックエディタの [新規] をクリックします



06 ブロックエディタのブロックを複製・削除する方法

ブロックエディタのブロックの上で右クリックをすると、ブロックの複製や削除ができます
ブロックをドラックし画面外でドロップすることでも、ブロックを削除できます



スクーミーボードが動かない場合の対応について

01 ブロックエディタの [アップロード] をクリックした時に「エラーが発生しました」とメッセージが表示され、正常に書き込めない

- スクーミーボードと書き込み機がしっかりつながっていて、パソコンにもつながっているかを確認する



書き込み機を挿す場所を間違えている



書き込み機を挿す向きを間違えている

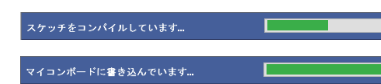


書き込み機のピンの位置がずれている



裏面のスイッチが PRGUSB ではなく BOX に入っている
※PRGUSB の方に入れてください

- 正しくつながっているのにエラーが出る場合は、スクーミーボードをリセットする

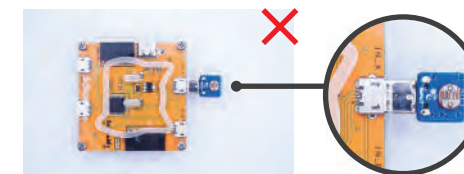


スクーミーボードの裏面のリセットボタンを指で押しながらアップロードをクリックして、「マイコンボードに書き込んでいます ...」の表示になったら指を離すことでリセットできます

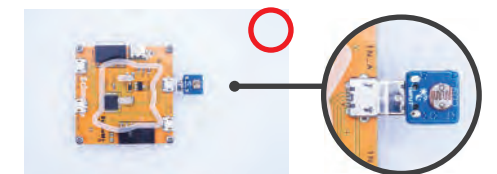
- SchooMyIDE を再起動する

02 エラーは発生せず書き込みはできるが、プログラム通りに正常に動作しない

- スクーミーボードに接続しているコネクタが壊れていないか、正しく接続されているか確認する



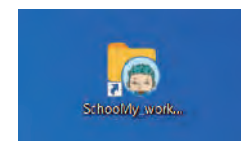
センサーが奥まで挿さっていない



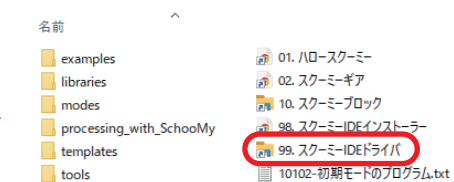
センサーが奥までしっかり挿しこんである

03 アップロードができない、スクーミーボードがパソコンに認識されない

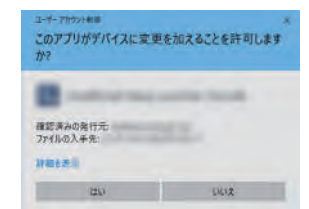
- ドライバーがパソコンにインストールされているか確認する ※Windows のみの対処法



SchooMy_workspace フォルダを開く



99. スクーミー IDE ドライバをダブルクリックする

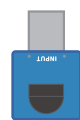


「はい」を選び、手順に沿ってドライバをインストールする

難易度 ★★★ 時間の目安 30分

複数のセンサーコネクタを接続したり、ブロックの繋げ方を工夫することで、身近にある仕組みを再現することができます

このクラスでは温度センサーを使って、高温になったら光と音で知らせてくれる温度計を作ってみましょう

目標 現在の温度を7SEGに表示し、温度の変化をLEDとスピーカーで知らせることができる**準備するもの**

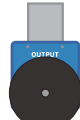
温度センサー



7SEG



フルカラーLED

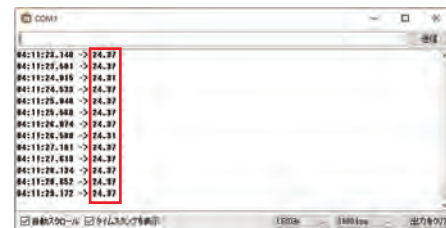
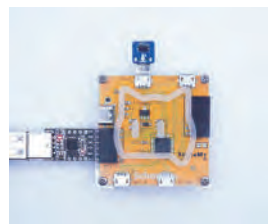


スピーカー

01 スクーミーボードの左上に温度センサーを挿します

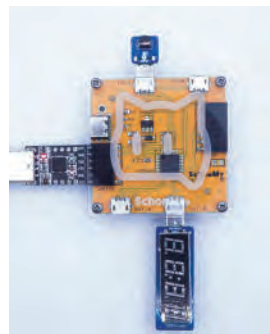
ブロックエディタでシリアルモニタブロック、温度センサーブロックを出してつなげましょう

温度センサーの値をシリアルモニタに表示するプログラムをアップロードし、シリアルモニタで現在の温度を確認しましょう

**02** スクーミーボードの右下に7SEGを挿します

ブロックエディタの[表示する]タブから7SEGのブロックを出してつなげます

アップロードして、現在の温度が7SEGに表示されていることを確認しましょう

**Mission1**

スクーミーボードの右上にスピーカーを挿します

温度が高くなったらスピーカーから音が鳴り、

温度が戻ったらスピーカーの音がとまるようにしましょう

Point

- もし～なら～でなければ～ブロックを正しく使える
- エリアの条件文の不等号の向きや、しきい値を正しく設定できる
- 音ブロックを使い、場所・出力・音階を正しく設定できる

**Mission2**

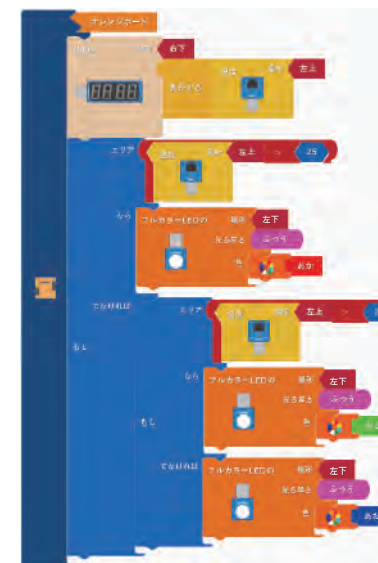
スクーミーボードの左下にフルカラーLEDを挿します

温度に合わせてフルカラーLEDの色を3段階で変わるようにしましょう

25℃以上：赤色 23℃以上 25℃未満：緑色 低温時 23℃未満：青色

Point

- もし～なら～でなければ～ブロックを2つ使って3段階の分岐ができる
- エリアの条件文の不等号の向きや、しきい値を正しく設定できる
- LEDブロックを使い、場所・色を正しく設定できる

**Mission がクリアできたら、Challenge に挑戦！**

現在の温度を7SEGに表示し、温度に合わせてフルカラーLEDの色を3段階で変えることができます

Challenge

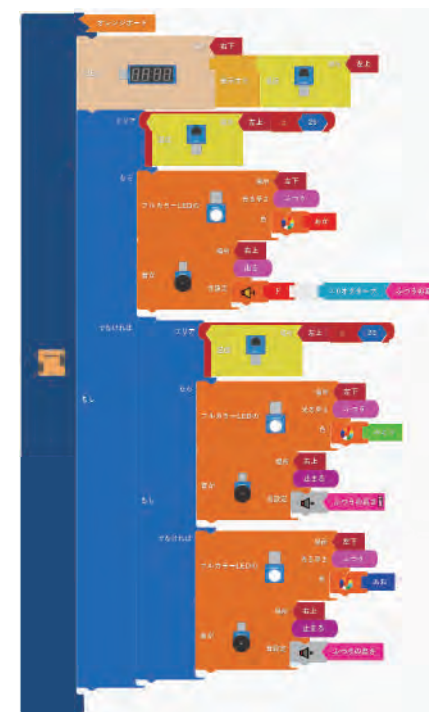
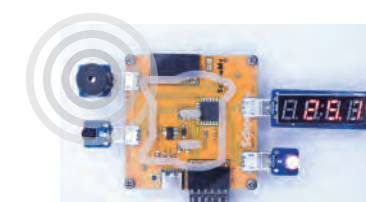
現在の温度を7SEGに表示し、温度に合わせてフルカラーLEDの色を3段階で変わるようにしましょう

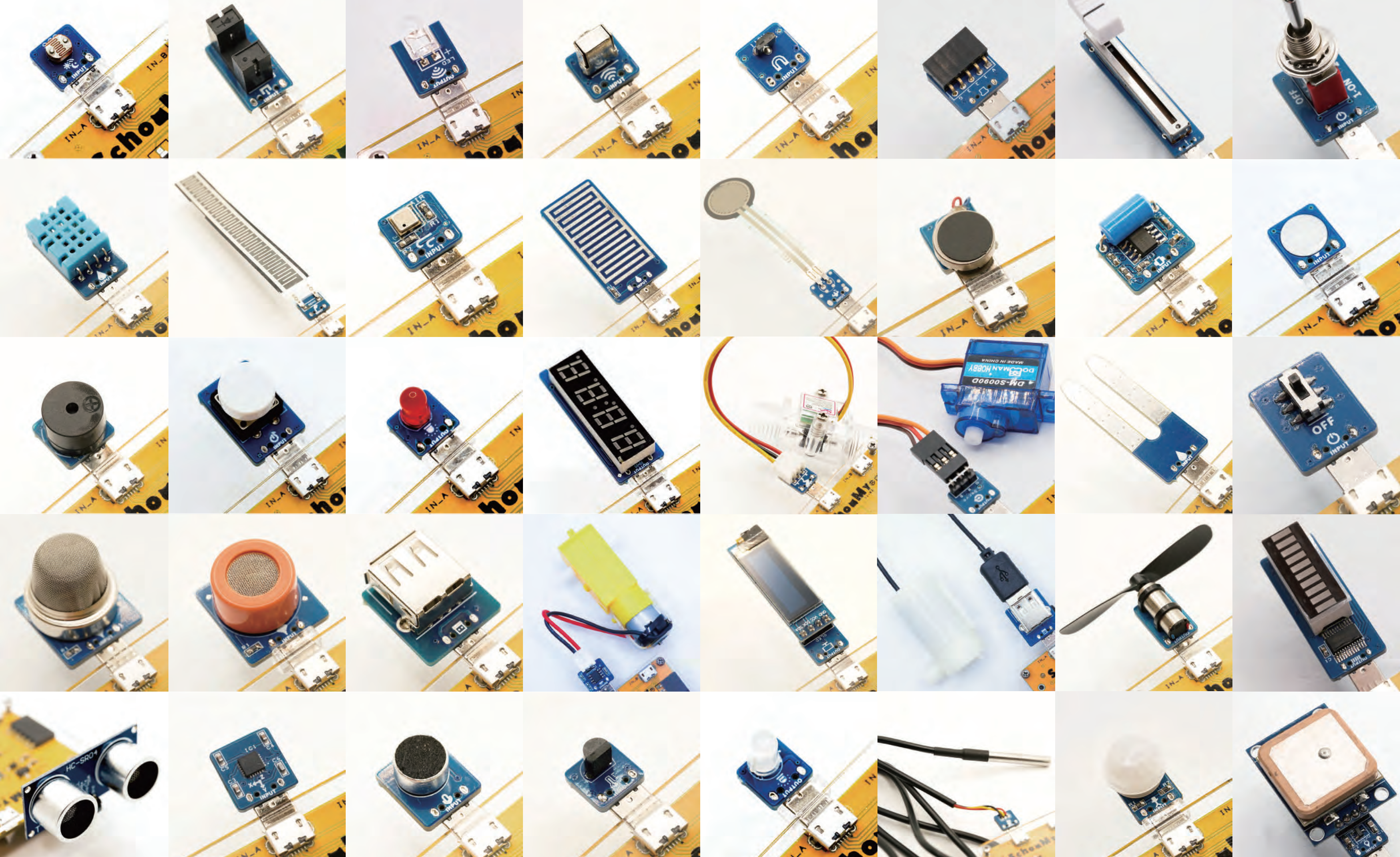
高温の時はフルカラーLEDを赤く光らせながらスピーカーから音が鳴るようにしましょう

25℃以上：赤色・スピーカー

23℃以上 25℃未満：緑色

低温時 23℃未満：青色





OUTPUT

27SEG

数字を表示する（9999まで）



➤

センサーが取得する数字を表示することができる

➤

時間や個数など、カウントアップやダウンができる

仕組みを知って使いこなそう！



センサーの値を表示する

→パソコンにつなげていなくても、センサーがどんな値を取得しているのかを知ることができる

個数を数えることができる

→1、2、3と何かのアクションによって数字が1ずつ増やすことができる

タイマー

→1秒ごとに数字が増えるようにすることができる

7本の光で、数字を表します。

そのため、7SEGといわれています。

個数や時間を見えるようにすることで自分や他の人に、伝えることができる

そろそろ時間だ！
という残り時間を伝える

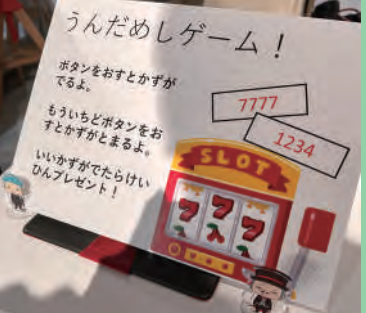
数が減ってきた！
という残りの量を伝える

これだけやった！
という回数を伝える

楽しませるためのモノづくり

温度や明るさを表示させるだけでなく、人の通過の回数も表示させるときに利用する。

その中でもルーレットのように数字を3ケタで表示させ、ゾロ目を出したり、決めておいた数字をランダムに出したりすることでゲーム性を持たせることができる。



1

明るさを7SEGに表示する

オレンジボード

7SEG

場所

右下

表示する

明るさ

場所

左上





2

変数を使って数字を表示する

オレンジボード

数の変数

i

0

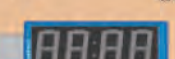
7SEG

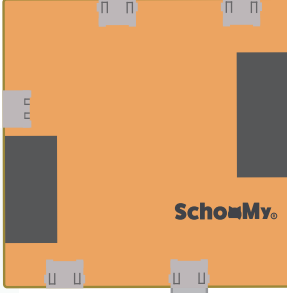
場所

右下

表示する

i





3

ストップウォッチをつくる

スクーミー

オレンジボード

動く

数の変数 値をセット

nowTime

起動からの時間をセット

秒

エリア

タッチ

場所

左上

==

1

なら

エリア

and どちらも

stat == 0

prevSwitch == 0

なら

数の変数 値をセット

stat 1

数の変数 値をセット

startTime

起動からの時間をセット

秒

もし

もし でなければ

エリア

and どちらも

stat == 1

prevSwitch == 0

もし なら

数の変数 値をセット

stat 0

数の変数

prevTime

stopwatch

数の変数 値をセット

prevSwitch 1

でなければ

数の変数 値をセット

prevSwitch 0

エリア

stat == 1

もし なら

数の変数 値をセット

stopwatch

prevTime +

nowTime -

startTime

7SEG

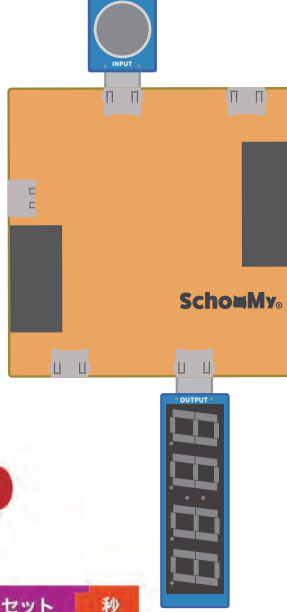
場所

右下

表示する

stopwatch





このコネクターを使う方法

1

スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください

2

スクーミーボードと書き込み機を準備します
イラストのようにつなげてください

3

QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロード
することができます







OUTPUT

1 LED

光ることで知らせたり、楽しませたりする



光ります

ずっと光ったり、ピカピカ光ったりする

仕組みを知って使いこなそう！

暗いときに目立つ

楽しくする

使っていることを知らせる

ピンチを伝える！！

ここです！と伝える

最初は
ゆっくり光っていて、
時間になったら、
はやく光るようにする

光り方を工夫することで、声に出さなくてもメッセージを伝えることができる

- 終わったかどうかを伝える
- 使っているかどうかを伝える
- 時間になりそうなことを伝える

コミュニケーションボード

ボードとLEDを複数個利用し、該当するところを光らせて、文字で書くよりも視覚的にわかりやすく相手に伝える仕組み。

当番表や明日の連絡などを光で伝えることでコミュニケーション問題を解消する。



スイッチで順番に光らせる

1 LEDを点滅させる

オレンジボード

LEDが 場所 右下 光る

待つ ミリ秒 1000

LEDが 場所 右下 光らない

待つ ミリ秒 1000

2 LEDを交互に光らせる

オレンジボード

LEDが 場所 右下 光る

LEDが 場所 左下 光らない

待つ ミリ秒 1000

LEDが 場所 右下 光らない

LEDが 場所 左下 光る

待つ ミリ秒 1000

このコネクターを使う方法

スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください



QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロードすることができます

3 スイッチを押したらLEDが光り、もう一度スイッチを押すとLEDが消える

オレンジボード

エリア

もし

スイッチ 場所 左上 == 1

and どちらも

flg == 0

なら

数の変数 i 1つ増やしてセット

数の変数 flg 1

エリア

i % 2 == 1

なら

LEDが 場所 右下 光る

もし

でなければ

LEDが 場所 右下 光らない

エリア

スイッチ 場所 左上 == 0

もし

なら

数の変数 flg 0



INPUT

② スイッチ

早押しボタンや横断歩道にあるボタンなどの「おしボタン」



➡ おされると0から1に数値が変化します

➡ おされたことで、電気の通り道ができて、電気が流れたことを判定しています

仕組みを知って使いこなそう！



おした！

他にも・・・

おして、はなす

おしつづける

1度おして、もう1度おした

連打！！

それぞれのおしかたで反応させる

人がおすだけではなく、何か起きたことで、ものが動き、その動いたものがスイッチにあたるなどオリジナルスイッチをつくることができます

運動不足を解消するためのモノづくり

腹筋システム

システムの起動や終了をスイッチで行うことでシステムの開始をわかりやすくする

スイッチを押してからシステムを起動させ、距離センサーに足を近づけることでスタートをする仕組み

距離センサー



スイッチ

1 スイッチを押したときモニターに1が表示されることを確認する

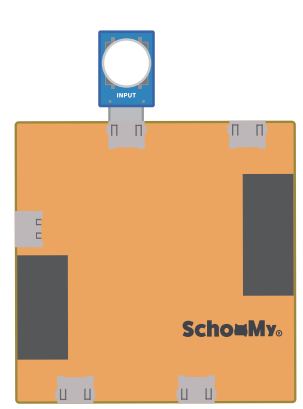
オレンジボード

シリアルモニター

行を変える

表示する

スイッチ 場所 左上



2 スイッチを押したらLEDが光るようにする

オレンジボード

シリアルモニター

行を変える

表示する

スイッチ 場所 左上

エリア

スイッチ 場所 左上 == 1

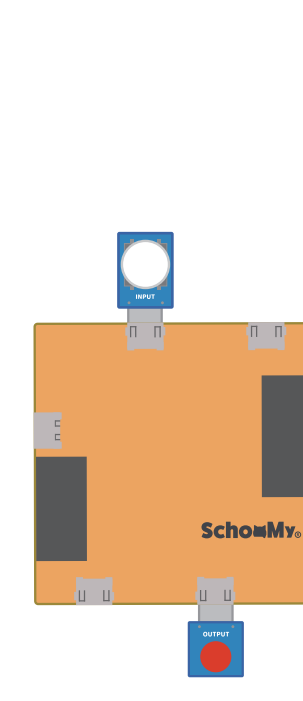
なら

LEDが 場所 右下 光る

もし

でなければ

LEDが 場所 右下 光らない



3 スイッチを押したら光り、もう一度押したら消える

オレンジボード

エリア

スイッチ 場所 左上 == 1

and どちらも

スイッチ 場所 左上

flg == 0

もし

なら

数の変数 i 1つ増やしてセット

数の変数 flg 1

エリア

i % 2 == 1

なら

LEDが 場所 右下 光る

もし

でなければ

LEDが 場所 右下 光らない

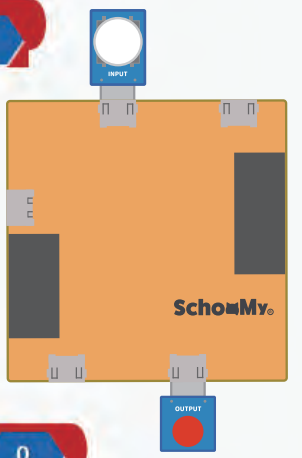
エリア

スイッチ 場所 左上 == 0

もし

なら

数の変数 flg 0



このコネクターを使う方法

1 スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください



2 スクーミーボードと書き込み機を準備します
イラストのようにつけてください



3 QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロードすることができます



OUTPUT

6 スピーカー

知らせる音・注意する音・
楽しませる音などを出す



➤ ドレミファ・・・でつくったメロディを出す

➤ #やオクターブ、長さの設定をしたメロディを出す

仕組みを知って使いこなそう！



ドレミファ
・・・#
オクターブ
長さの設定
をしたメロ
ディを出す

音を鳴らして注意すること・知らせることができる

- ・アラート：受け手が予期しない時に発せられる警告音
ex) 地震速報
- ・アラーム：受け手が事前に知っている場合の警告音
ex) 目覚まし時計
- ・サイン音：警告ではなく何らかの情報を提供することを目的に作られた音
ex) 電子レンジや洗濯機など家電機器の操作確認や終了音など

音の長さやテンポを工夫することで、
センサーでメッセージを伝えることができる

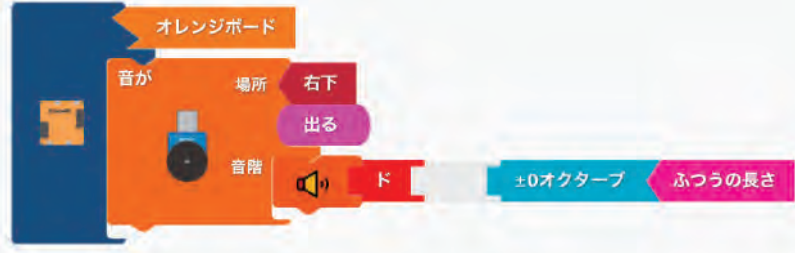
- あぶないことを
アラートで注意する
- 3分タイマーの
アラームで伝える
- サイン音で
楽しくする

農業の問題を解決するためのモノづくり

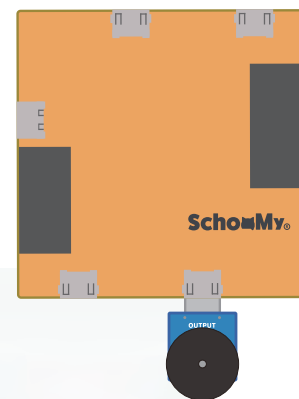
水面に磁石を浮かせておき、底の方に磁気センサーをおいておく状態をつくる。水が無くなってくと浮いている磁石が磁気センサーに近づき、一定の量水が減ったら音が鳴って知らせてくれる仕組み



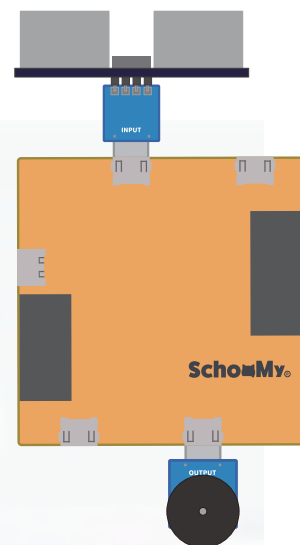
1 ドの音を鳴らす



2 休符を使って、メロディをつくる



3 距離によって、音が変わるようにする



このコネクターを使う方法

1 スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください



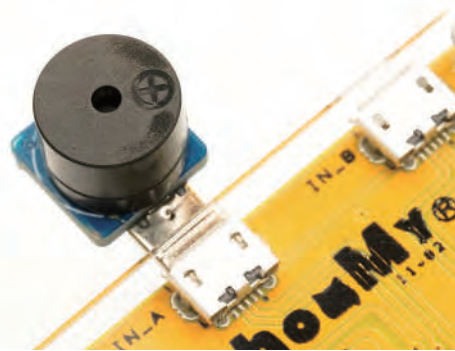
2 スクーミーボードと書き込み機を準備します
イラストのようにつなげてください



3 QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロード
することができます





INPUT

① タッチセンサー

スマートフォンなどのタッチ画面の仕組み



➤ センサーの銀の部分に指がふれることで0から1に数値が変化します

➤ 静電容量(せいでんようりょう)の変化を読み取っています

仕組みを知って使いこなそう！

直接ふれてもOK



電気を通しやすい何かを間に入れ、それにふれても反応します



例えば、アルミホイルを通して指からセンサーに電気が移動します



アルミホイル以外にも反応します

葉っぱ

水分をふくむもの

ハサミ

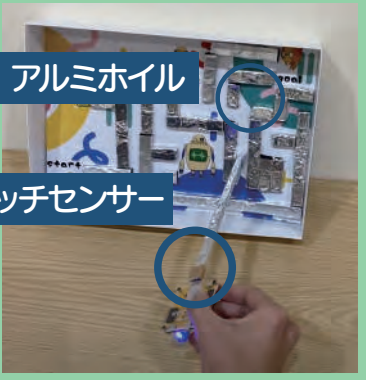
金属部分があるもの

楽しんでもらうためのモノづくり

触れたらアウトなゲーム

タッチセンサーに触れると反応することを利用し、アルミホイルを活用したゲーム

触れたら色が変わりゲームオーバーを伝えたり、制限時間を表示したり、ランダムに振動することなどを入れるなどのパワーアップ



アルミホイル

タッチセンサー

1

タッチしたらランダムに1から5の数字が表示される

オレンジボード

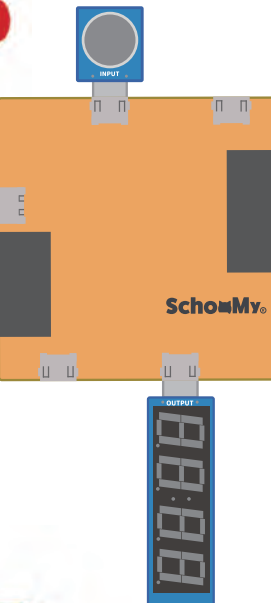
エリア

タッチ 場所 左上 == 1

もし なら

数の変数 a 1~5をランダムにセット

7SEG 場所 右下 表示する a



2

タッチしたらランダムに111から555の数字が表示される

オレンジボード

エリア

タッチ 場所 左上 == 1

もし なら

数の変数 a 1~5をランダムにセット

数の変数 b 1~5をランダムにセット

数の変数 c 1~5をランダムにセット

7SEG 場所 右下 表示する a × 100 + b × 10 + c



3

タッチしたらランダムに111から555の数字が表示されて、ゾロ目がそろったら赤色に光り、そろっていないときは青色に光る

オレンジボード

エリア

タッチ 場所 左上 == 1

もし なら

数の変数 a 1~5をランダムにセット

数の変数 b 1~5をランダムにセット

数の変数 c 1~5をランダムにセット

7SEG 場所 右下 表示する a × 100 + b × 10 + c

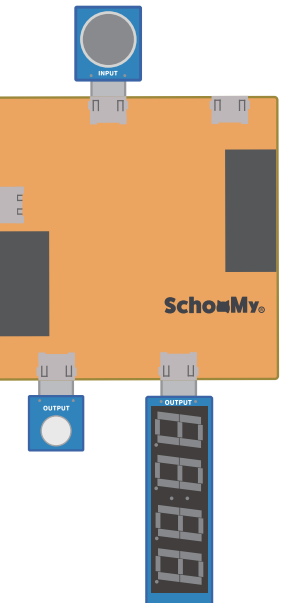
and どちらも a == b b == c

もし

フルカラーLEDの 場所 左下 光る早さ ぶつう 色 あか

でなければ

フルカラーLEDの 場所 左下 光る早さ ぶつう 色 あお



このコネクターを使う方法

1 スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください

2 スクーミーボードと書き込み機を準備します
イラストのようにつけてください



3 QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロードすることができます





OUTPUT

3フルカラーLED

RGBの値を設定し、色を変化させて光らせることができる

フルカラーLEDの値を変えて色を自由につくることができます

色が徐々に変わる光らせ方ができ、その速度も変えることができます

仕組みを知って使いこなそう！

光らない

R,G,B 255,0,0

R,G,B 0,0,0

R,G,B 0,255,0

R,G,B 255,255,255

R,G,B 0,0,255

R,G,B 255,0,255

光の三原色の原理を使って色を表現する手法です。主にスマートフォンやパソコンのディスプレイ、テレビなどの光で映像を映す機器で使用されています。

色の変化や違いで視覚的にわかりやすくします

温度の変化

水分量の変化

正解・不正解

危険度

混雑状況

音の大きさ・波形

感染症対策のためのモノづくり

コロナ対策ボックス

フルカラーLEDの色を変えることで、色の違いで「安全」・「危険」などを視覚的にわかりやすく伝える仕組み
箱の中にはマスクやウェットティッシュを入れるスペースを用意

1赤、青、緑をグラデーションで光らせる

オレンジボード

フルカラーLEDの場所右下

光る早さふつう

色あかあおみどり

2赤と青と緑を交互に光らせる

オレンジボード

フルカラーLEDの場所右下

光る早さふつう

色あか

待つミリ秒500

フルカラーLEDの場所右下

光る早さふつう

色あお

待つミリ秒500

フルカラーLEDの場所右下

光る早さふつう

色みどり

待つミリ秒500

このコネクターを使う方法

スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください

スクーミーボードと書き込み機を準備します
イラストのようにつなげてください

3タッチするたびに、赤、青、緑と光る

オレンジボード

エリアタッチ場所左上==1

数の変数i1つ増やしてセット

エリアi%3==1

フルカラーLEDの場所右下

光る早さふつう

色あか

エリアi%3==2

フルカラーLEDの場所右下

光る早さふつう

色あお

エリアi%3==0

フルカラーLEDの場所右下

光る早さふつう

色みどり

待つミリ秒500

QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロードすることができます

QRコード

回路図

实物写真

INPUT

12 温度センサー

家電製品の温度が上がりすぎないようにするための仕組み



➤ 黒い部分が温度の変化を読み取ることができます

➤ 気温を測ったり、ふれたものの温度を測ることができます

仕組みを知って使いこなそう！



この黒い部分が温度を感知しています

空気の温度変化を感知しますが、ふれたものの温度変化も感知することができます



SchoMy®

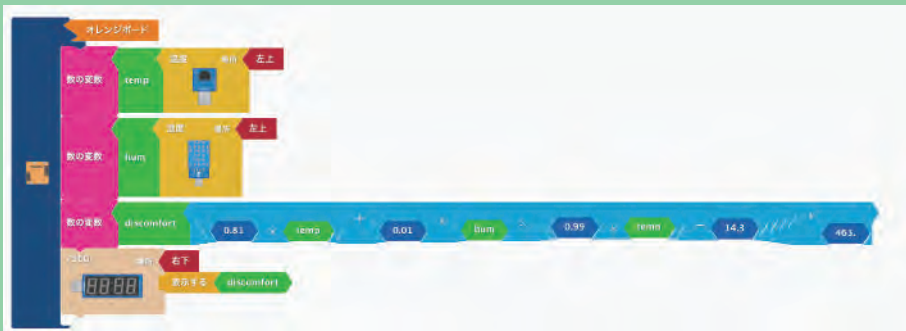
指で黒い部分を持ち、力を入れることで温度が変わっていく様子を見ることができます。

水温を測るときは、水温センサーを使ってください。

温度の変化をグラフで見ることができます

健康管理をするためのモノづくり

取得した温度をそのまま7SEGに表示することもできるが、温度の値と湿度の値をセンサーで取得し、それらの値を計算することで、不快指数や暑さ指数(熱中症指数・WBGT)などを出し、7SEGに表示することもできる



オレンジボード

数値変数 temp 温度 場所 左上

数値変数 hum 湿度 場所 左上

数値変数 discomfort 0.81 × 12002 + 0.01 × hum × 0.99 - 14.3 = 463.1

7SEG 表示 右下

表示する discomfort

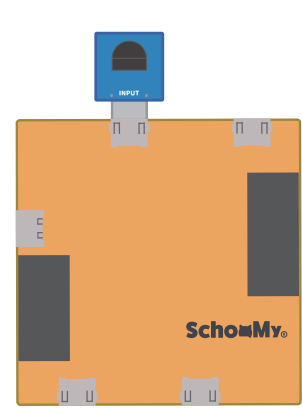
1 センサーを温めて30°Cにしよう



オレンジボード

シリアルモニタ 行を変える 温度 場所 左上

表示する



2 30°Cを超えたらLEDが光るようにする



オレンジボード

シリアルモニタ 行を変える 温度 場所 左上

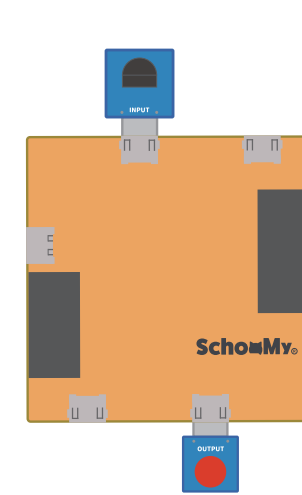
表示する

エリア 温度 場所 左上 > 30

もし

なら LEDが 場所 右下 光る

でなければ LEDが 場所 右下 光らない



3 30°C以上、もしくは、20°C以下のときに、赤色で点滅をする。それ以外のときは緑色に光る。



オレンジボード

シリアルモニタ 行を変える 温度 場所 左上

表示する

エリア

温度 場所 左上 < 30 or どちらか 温度 場所 左上 < 20

なら フルカラーLEDの 場所 右下 光る早さ ふつう 色 あか

待つ ミリ秒 100

フルカラーLEDの 場所 右下 光る早さ ふつう 色 光らない

待つ ミリ秒 100

もし

でなければ フルカラーLEDの 場所 右下 光る早さ ふつう 色 みどり

このコネクターを使う方法

1 スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください

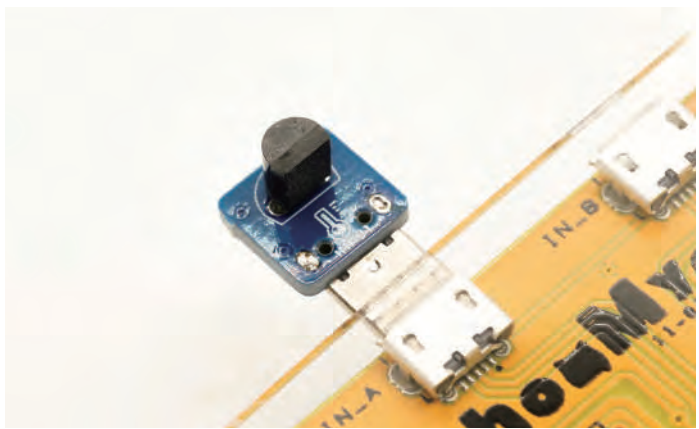


2 スクーミーボードと書き込み機を準備します
イラストのようにつなげてください



3 QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロードすることができます



6 音センサー

スピーカーや音声入力できる仕組み



音の大きさ・小ささを読み取ることができます

音によって生成された音波がセンサーの中の振動版にあたる必要があります

仕組みを知って使いこなそう！

OK！センサーの中の板がふるえる

OK・・・センサーの中の板がふるえにくい
が、ギリギリ反応する

NO！センサーの中の板がふるえない

NO・・・センサーの中の板がふるえにくい

音の大きさ・小ささをグラフで見ることができます
※音の高さ・低さは読み取りにくいです

スクーミーボードをパソコンにつなげて
グラフをパソコンに表示させます

暮らしやすい便利をつくるためのモノづくり

体温計の音を感知するシステム

聞き取りにくいを解決するためのモノづくり
体温計の音は小さく高齢者には聞き取りづらいという課題を、音センサーを活用して、音が鳴ったらLEDが光る仕組みをつくり、体温を測り終わったら光で合図をしてくれる仕組み

音センサー

振動と光



1 センサーの前で話したときのグラフの波形を確認する

オレンジボード

シリアルモニタ

行を変える

表示する

音

場所

左上

2 音が大きかったらLEDが光るようにする

オレンジボード

シリアルモニタ

行を変える

表示する

音

場所

左上

エリア

音

場所

左上

>

100

もし

LEDが

場所

右下

光る

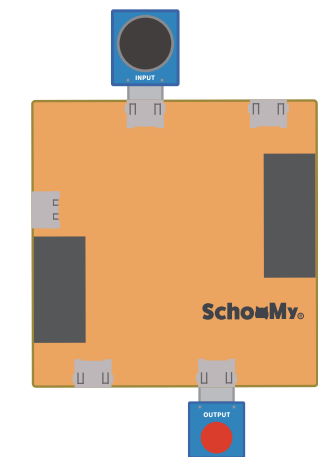
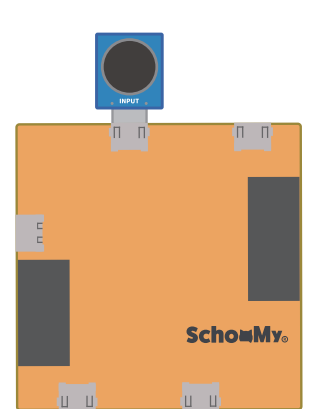
でなければ

LEDが

場所

右下

光らない



3 大きい音が3秒続いたら、LEDが光るようにする

オレンジボード

シリアルモニタ

行を変える

表示する

音

場所

左上

エリア

音

場所

左上

>

100

の間くりかえす

待つ

ミリ秒

1000

数の変数

i

1つ増やしてセット

エリア

i

≥

3

もし

LEDが

場所

右下

光る

数の変数

i

0

LEDが

場所

右下

光らない

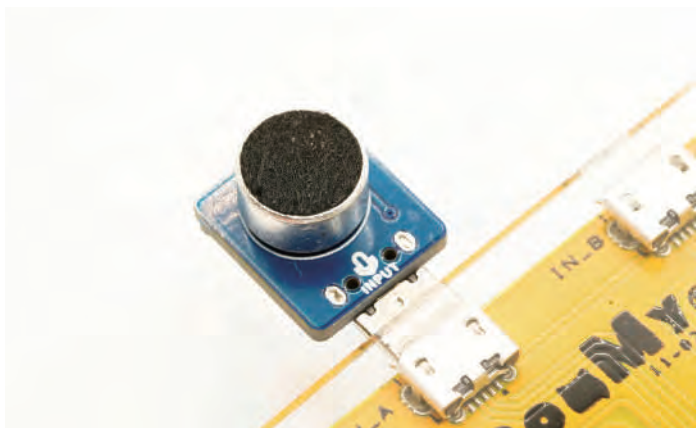
このコネクターを使う方法

1 スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください

2 スクーミーボードと書き込み機を準備します
イラストのようにつけてください

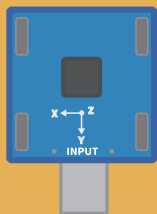
3 QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロードすることができます



13 加速度・角度センサー

ボードがどれだけ動いたかを数値で示すことができる



ボードがどれだけ水平から傾いたかの角度がわかる

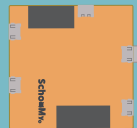


止まっているときを1とした時どれだけ動いたかがわかる。また加速度の大きさを測定できる。

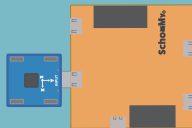
仕組みを知って使いこなそう！

x軸の加速度

+1

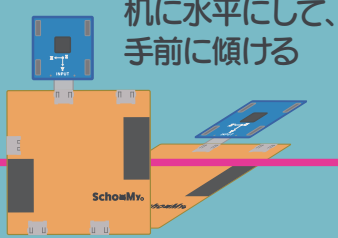


-1



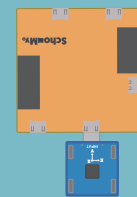
x軸の角度

机に水平にして、手前に傾ける

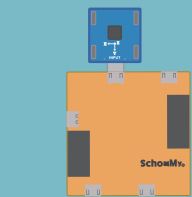


y軸の加速度

+1

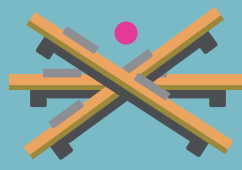


-1



y軸の角度

机に水平にして、左右に傾ける



z軸の加速度

+1



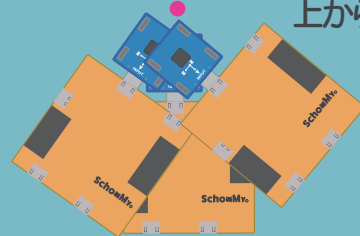
-1



机に水平にして表にするか裏にするか

z軸の角度

机に水平にして上から見た様子



楽しんでもらうためのモノづくり

加速度・角度センサーを動かすことで、テレビゲームのコントローラにすることができる。また自分でUnityなどでゲームをつくり、そのゲームのコントローラにすることもできる。

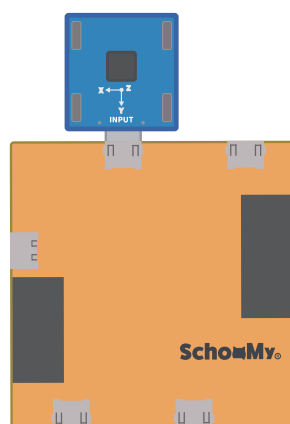
Unityでつくったゲーム



角度を読み取る

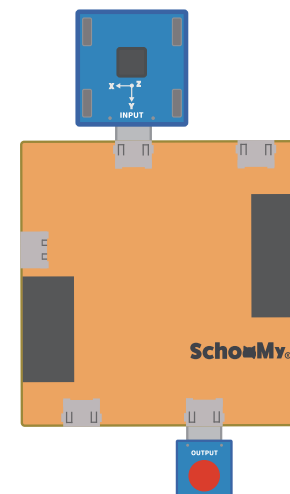
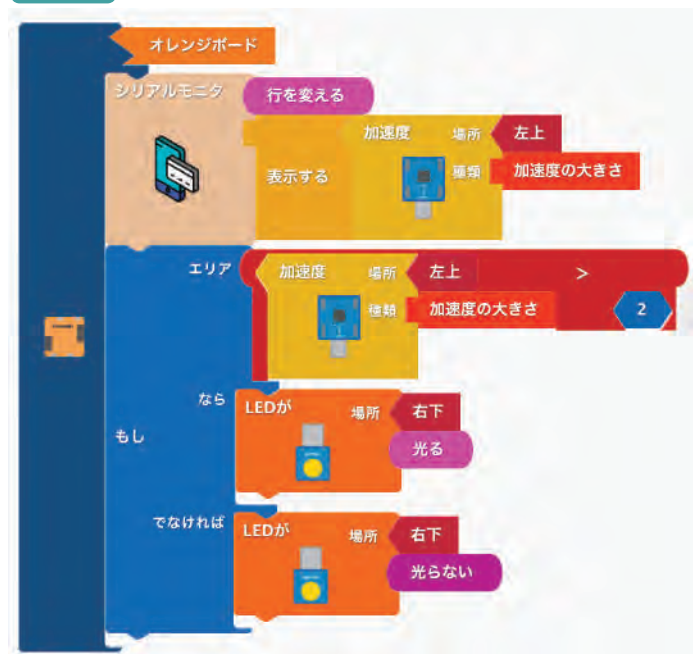
1

加速度の大きさが2を超えるようにセンサーを動かす



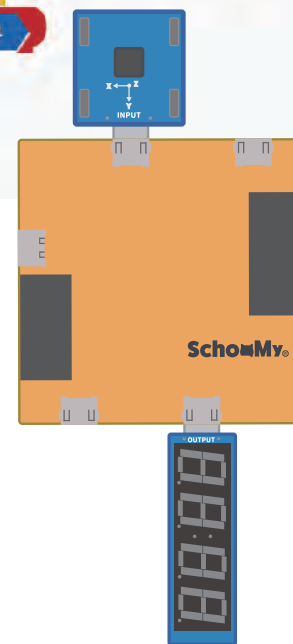
2

加速度の大きさが2を超えたらLEDが光るようにする



3

振った回数をカウントする



このコネクタを使う方法



スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください

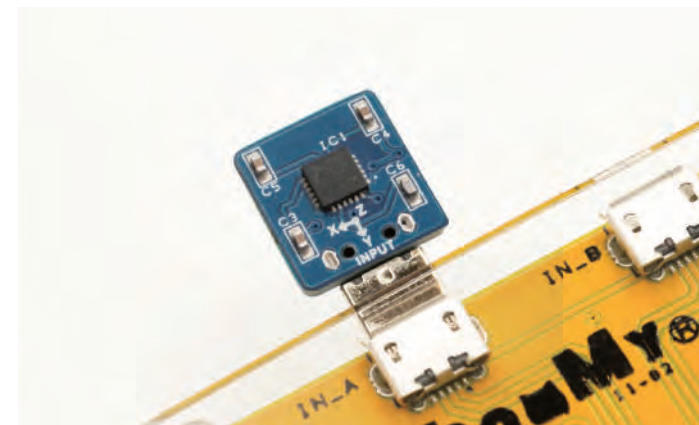


スクーミーボードと書き込み機を準備します
イラストのようにつけてください



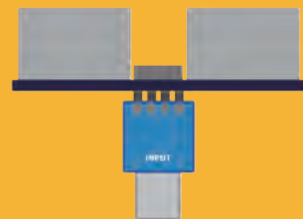
QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロード
することができます



14 距離センサー

自動運転の車の衝突(しょうとつ)を防ぐ仕組み



- およそ2mくらいまでの距離を測ることができます
- 超音波センサーとよばれることもあります

仕組みを知って使いこなそう!

超音波を発し、何かにあたり、はねかえってくることを感知することができます

そのはねかえりまでの時間を計算して、モノまでの距離を計算しています



距離がわかると、こんなこともわかります

量

通過

速さ

高さ

存在

運動不足を解消するためのモノづくり

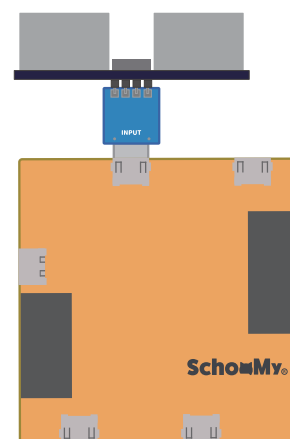
腹筋ローラーと組み合わせて、音が鳴るまで近づくことで、回数を数えてくれ、目標数値まで達するとメロディが鳴る。サボりがちな筋トレを楽しくわかりやすくできるようにした仕組み

腹筋ローラー

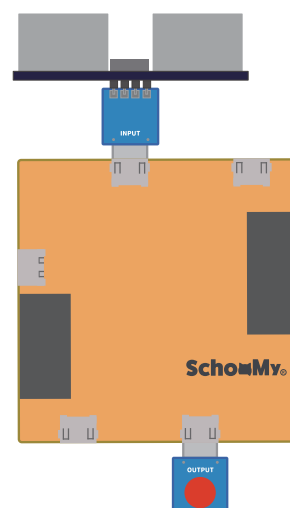
距離センサー



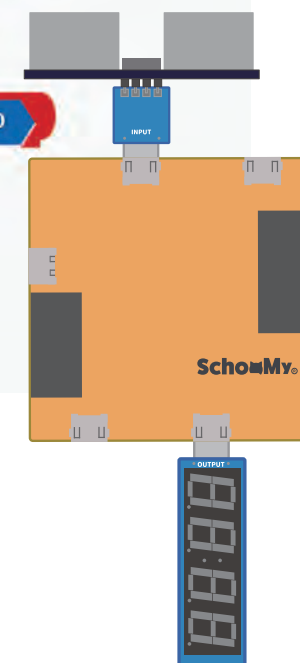
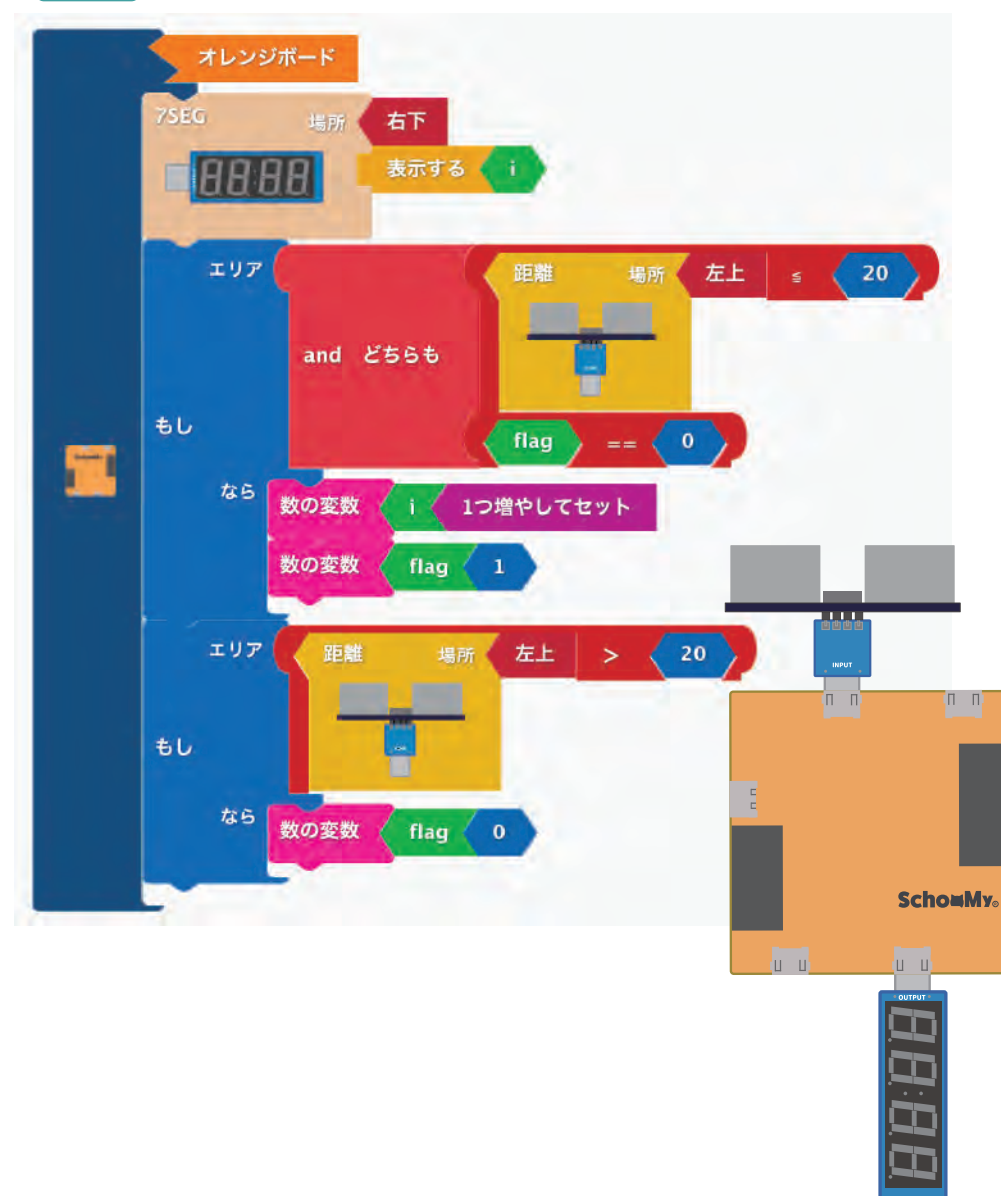
距離が20センチになるようにセンサーの前にモノをおこう



10Luxより小さくなったときに、LEDが光るようにする



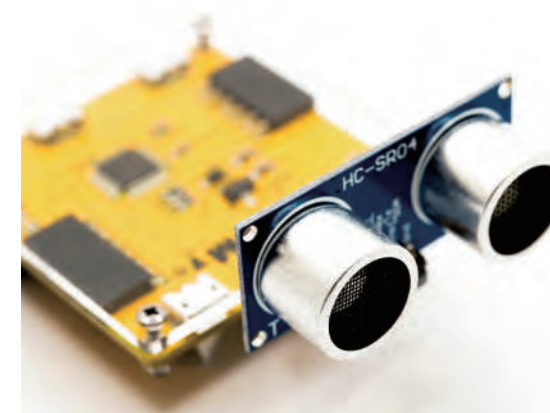
センサーの前を通った人数をカウントする



このコネクターを使う方法

スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてくださいスクーミーボードと書き込み機を準備します
イラストのようにつなげてください

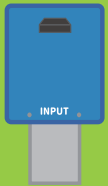
QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロード
することができます

INPUT

4 磁気センサー

エアコンや冷蔵庫などの家電製品で利用される仕組み



➤ 磁力に反応します。N極、S極で反応が変わります。

➤ ホール効果を利用しているので、ホールセンサーともよばれます。

仕組みを知って使いこなそう！



黒い部分に磁力を持つものを近づけることで反応する。

反応はこの3パターンになります

- ▶ N極、S極のどちらかのみを感知する
- ▶ N極、S極の両方を感知する
- ▶ N極からS極、S極からN極への、磁界の変化を感知する

磁力の変化がわかると、こんなこともわかります。

回転

高さ

開閉

暮らしやすい便利をつくるためのモノづくり

あれ？鍵かけたっけ？という外出してから不安を、家の外にいても鍵がかかったかどうかをスマートフォンから確認できる仕組み

鍵の回る部分に磁石をつけて、閉まっているか開いているかをセンサーとの距離で調べる



磁気センサー

システムの説明

14:00:52.132 → 2459	14:01:00.256 → 1751
14:00:52.132 → 2459	14:01:00.256 → 1757
14:00:52.132 → 2454	14:01:00.256 → 1754
14:00:52.132 → 2461	14:01:00.256 → 1747
14:00:52.132 → 2471	14:01:00.256 → 1746
14:00:52.132 → 2467	14:01:00.256 → 1747
14:00:52.132 → 2465	14:01:00.256 → 1756
14:00:52.132 → 2458	14:01:00.256 → 1749

鍵が閉まっている状態 基準値 鍵が開いている状態

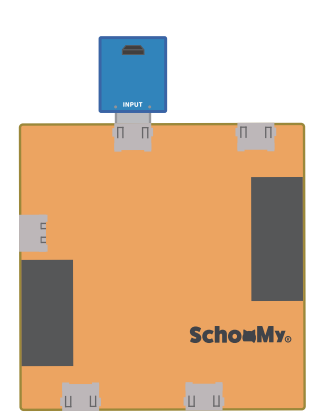
約2400 1900 約1700

1 磁石を近づけたときの数値の変化を確認する

オレンジボード

シリアルモニタ 行を変える 磁力 場所 左上

表示する



2 磁石を近づけたらLEDが光るようにする

オレンジボード

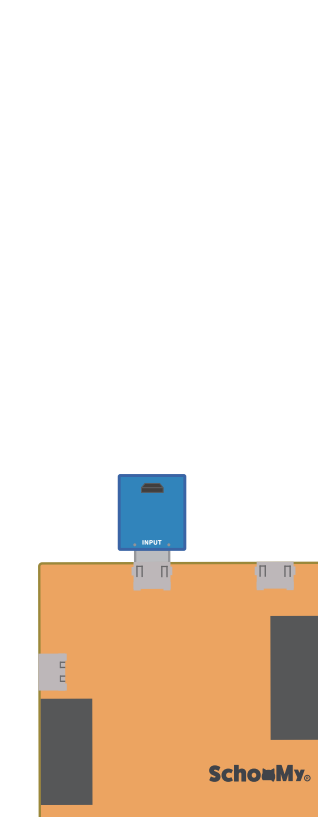
シリアルモニタ 行を変える 磁力 場所 左上

表示する

エリア 磁力 場所 左上 > 600

もし なら LEDが 場所 右下 光る

でなければ LEDが 場所 右下 光らない



3 磁石が5秒間離れたらLEDが光る

オレンジボード

シリアルモニタ 行を変える 磁力 場所 左上

表示する

エリア 磁力 場所 左上 < 600

の間くりかえす 待つ ミリ秒 1000

数の変数 i 1つ増やしてセット

エリア i ≥ 5

もし なら LEDが 場所 右下 光る

数の変数 i 0

LEDが 場所 右下 光らない

このコネクターを使う方法

1 スクーミーのアプリが入ったPCを準備します
スクーミブロックエディタを開いてください

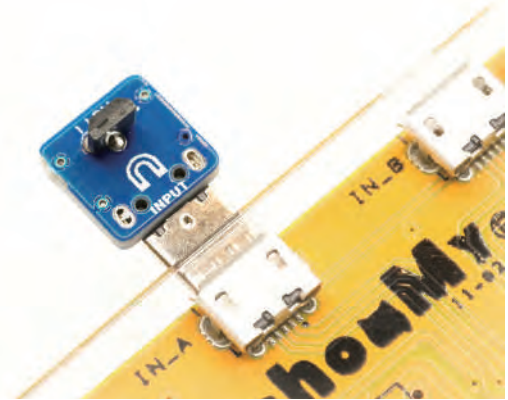
2 スクーミーボードと書き込み機を準備します
イラストのようにつけてください

3 QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロードすることができます







8 赤外線受信

リモコンなどから発せられる
赤外線を読み取る



赤外線通信の受信を
行います



テレビやエアコンのリモコンの
赤外線の情報も読み取
ることができます

仕組みを知って使いこなそう！

プログラムをする



プログラムが完了したら、
シリアルモニターを開き、
赤外線受信コネクタに、
テレビのリモコンを向けて
ボタンをおしてみよう。
受信すると数値の羅列が表示
されます。

赤外線リモコンから発信される信号を受信し、対応する機器に伝えるためのコネクタです。

多くの家電製品や電子機器に使われており、テレビ、エアコン、音響機器など、リモコン操作が可能な製品で広く利用されています。



学校の感染症対策のためのモノづくり

消毒管理システム

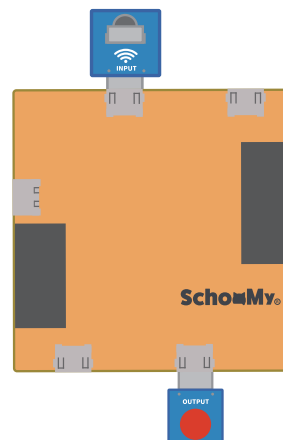
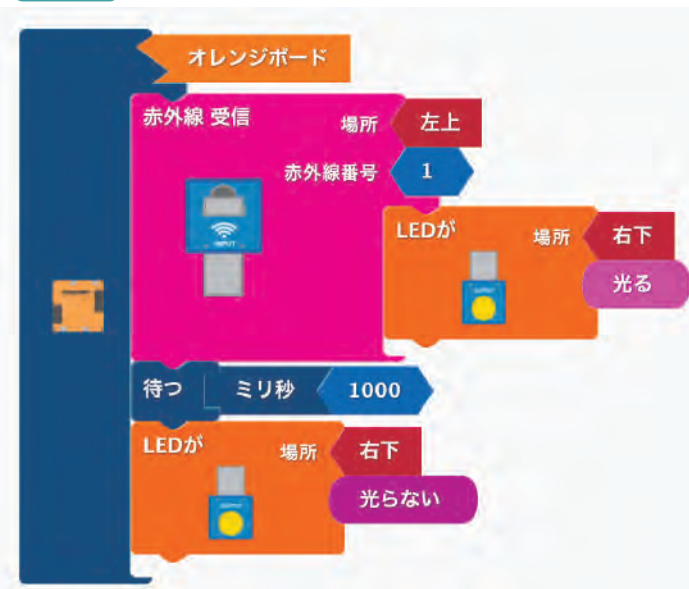
赤外線送信

消毒したことを圧力センサーで感知をし、送信する。

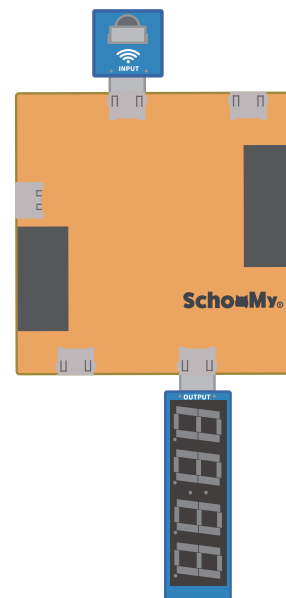
入り口の距離センサーに反応したとき、消毒しているかしていないかを受信した情報をもとに判定し、してなかった場合、注意をする。



赤外線番号1を受信したらLEDが1秒光る
その後、消える



赤外線番号1を受信したら変数iが1増えて、
それが7SEGに表示される



このコネクタを使う方法



スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください



スクーミーボードと書き込み機を準備します
イラストのようにつなげてください

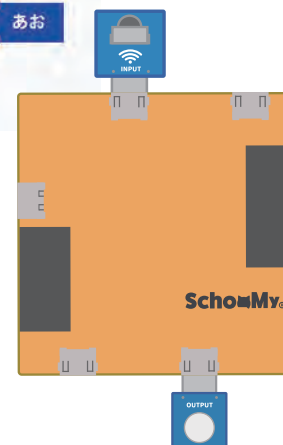


QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロードすることができます



赤外線番号1を受信したら赤が光る
赤外線番号2を受信したら緑が光る
赤外線番号3を受信したら青が光る



5 赤外線送信

同じ赤外線を設定することで
テレビなどを操作できる



- 赤外線の送信を行います
- リモコンと同じ赤外線を発する設定をプログラムすることでテレビなどを操作することができる

仕組みを知って使いこなそう！

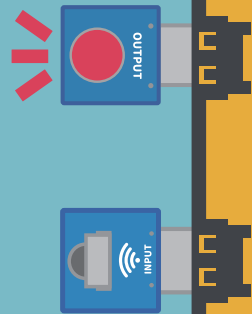
プログラムをする



プログラムをする



2台のスクーミーボードと赤外線受信送信コネクタを使うことで、離れたところにあるスクーミーボードをコントロールすることができる



楽しんでもらうためのモノづくり

スイッチを押すことで、もう一方のスクーミーボードの7SEGの数字が変化するようにプログラムをし、シューティングゲームのように遊べる仕組み。数字に合わせてフルカラーLEDの色も変わる。

赤外線送信

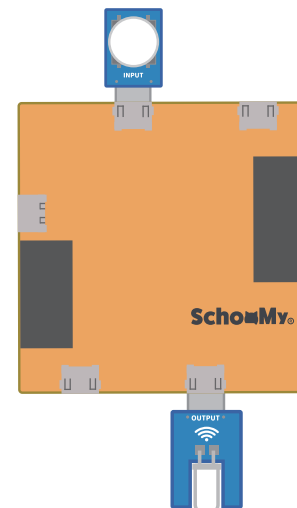
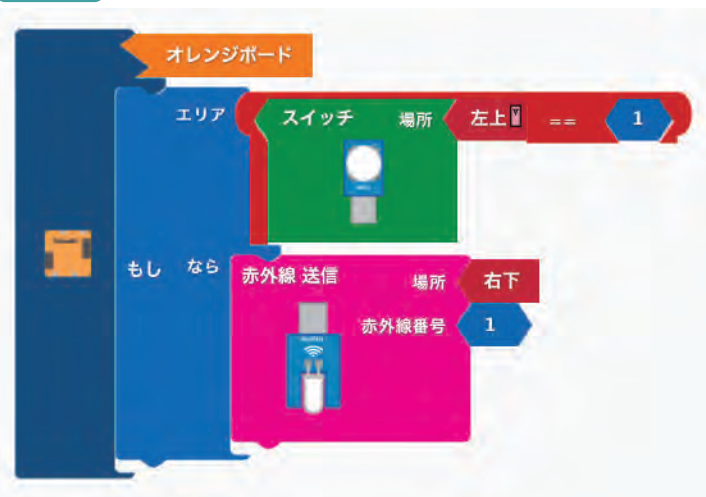
赤外線受信

7SEG



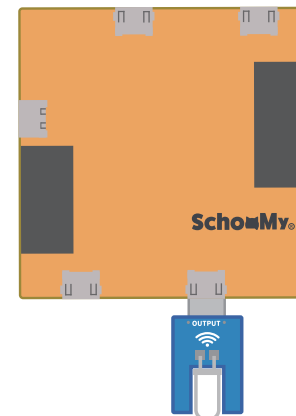
1

スイッチをおしたら、赤外線番号1が送信される



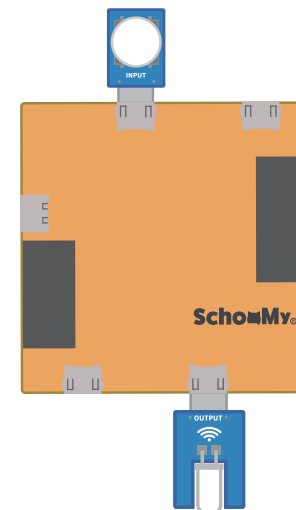
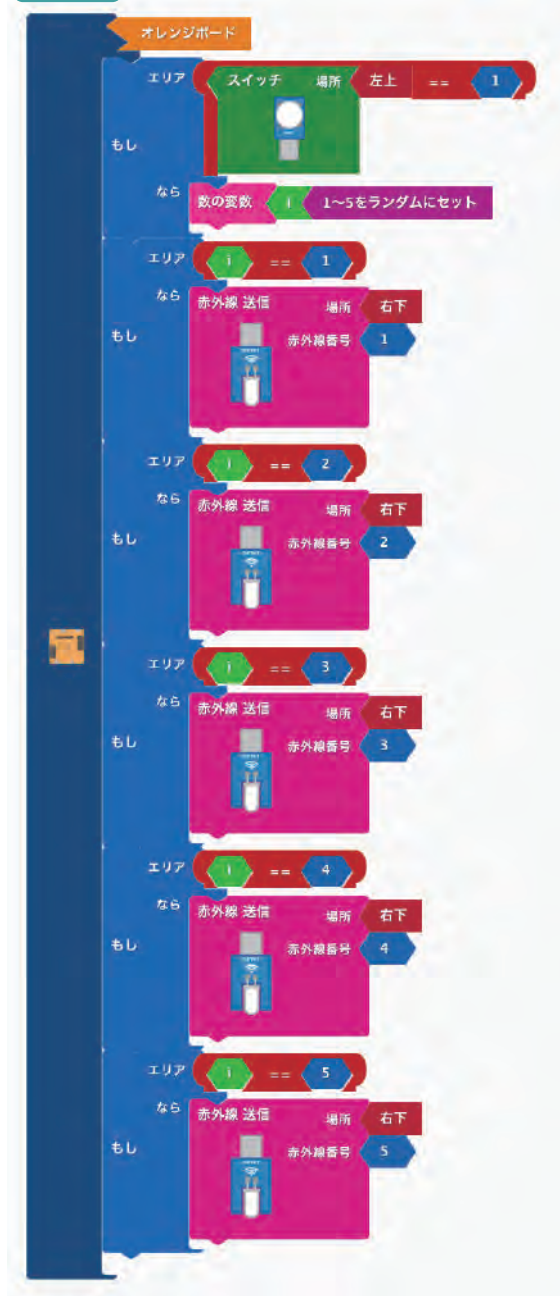
2

赤外線番号1から3までが順番に一定間隔で送信される



3

スイッチを押したら、ランダムで赤外線番号1から5が送信される



このコネクタを使う方法

- スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください

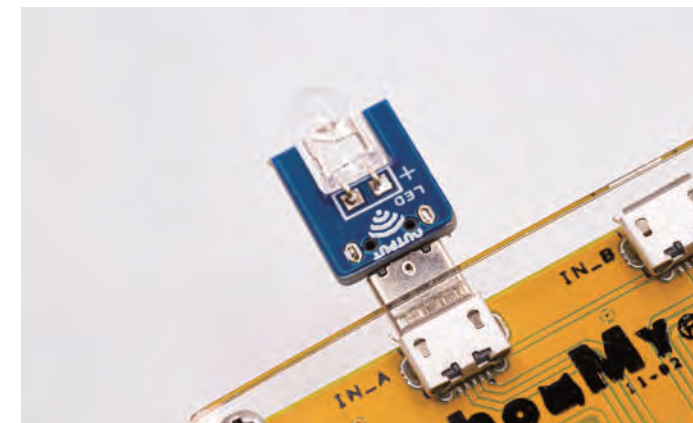


- スクーミーボードと書き込み機を準備します
イラストのようにつけてください



- QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロード
することができます



5 通過センサー

紙やコインなどのうすいものを通過して反応させる



➤ うすいモノの通過を調べることができます

➤ 回転や開閉を感知することができます

仕組みを知って使いこなそう！

通過センサーの簡単な仕組み(横からみた図)



モノがあるとき

すき間にモノがあると、片方のとんがり光をうけとれない

モノがある(通過した)とわかる

光を受けとれない

モノがないとき

すき間にモノがないと、光の受けわたしがおきる

モノがない(通過していない)とわかる

光を受けとる

貯金をするためのモノづくり

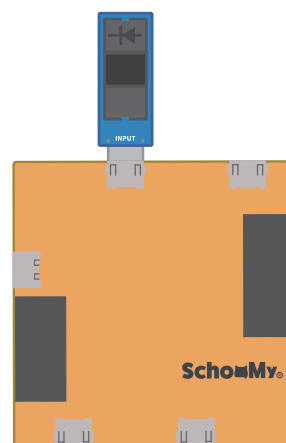
貯金箱のお金を入れる口の部分につけることによって、お金が入った回数やその回数によってアクションを起こしたり、一定時間お金が入っていないことを伝えることができる仕組み

通過センサー



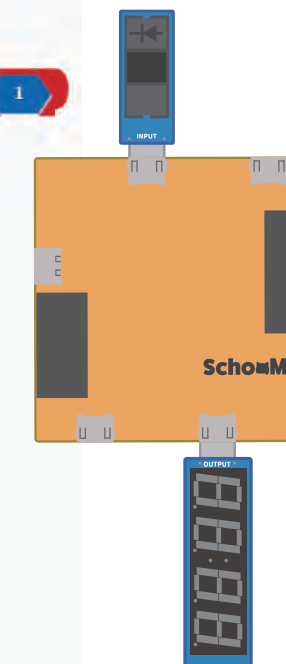
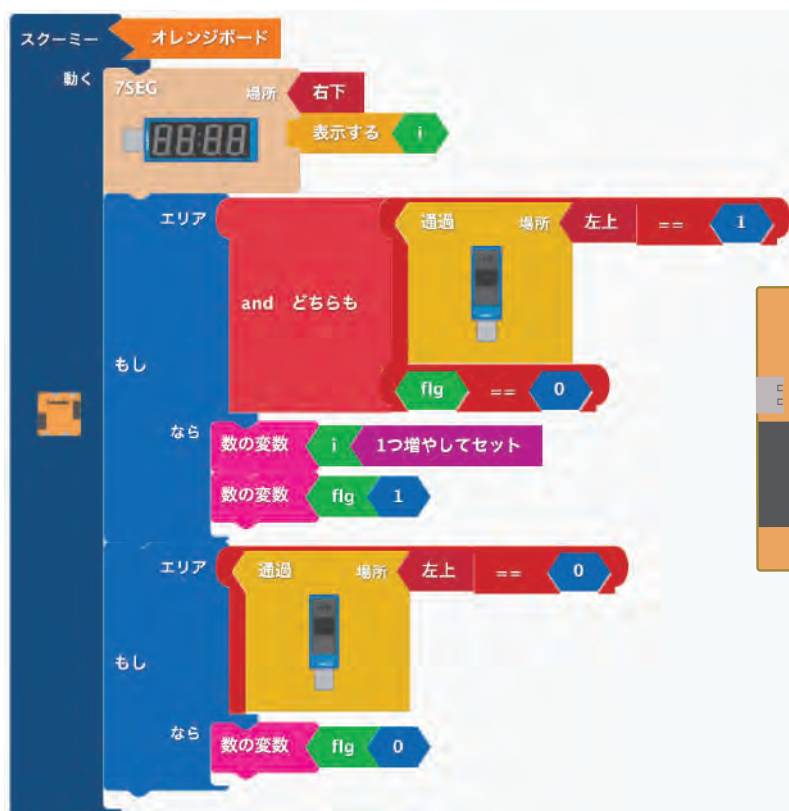
1

センサーにカードを通過させたときモニターに1が表示されることを確認する



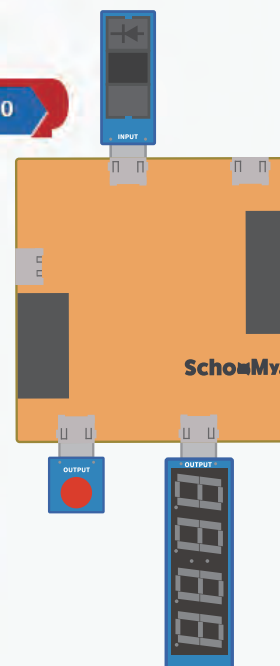
2

通過した回数をカウントする



3

通過した回数が5回以上ならLEDが光る



このコネクターを使う方法



スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください

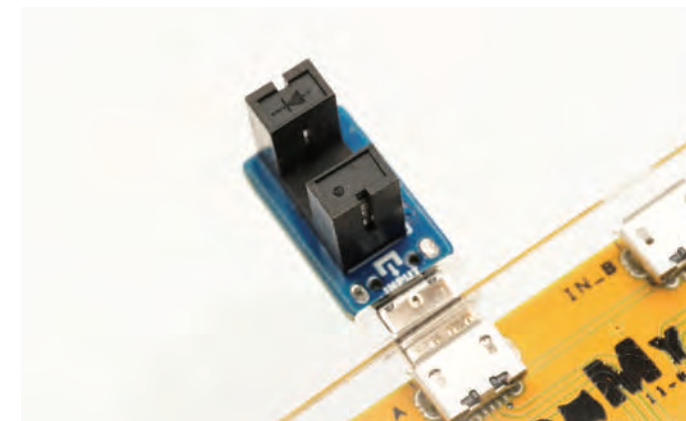


スクーミーボードと書き込み機を準備します
イラストのようにつけてください



QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロード
することができます



INPUT

11 土壌水分センサー

土の水分量を調べて、その土地に適した作物を植える仕組み



➤ 土の中の水分量を測ることができます

➤ 水がない状態を0%
水しかない状態を100%
としています

仕組みを知って使いこなそう！



土の中にさしてぬいた後は、コネクターの銀色の部分に着いた土をキレイにしよう

センサーの銀色の部分を土にさしてください



水分量の変化をグラフで見ることができます

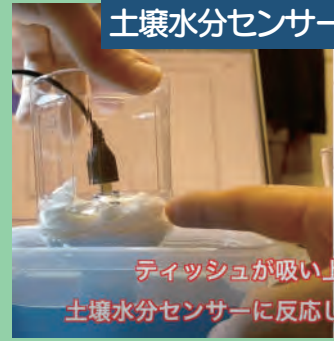


スクーミーボードをパソコンにつなげておくことでグラフをパソコンに表示させます

実験とモノづくり

毛細管現象

水が上がってくる仕組みを視覚的にわかりやすくするために土壌水分センサーとティッシュペーパーを使って実験のサポートとしてセンサーを活用



ティッシュが吸い上げた水分が土壌水分センサーに反応し数値が大きくなる

1 センサーを土にさし、水をかけたときの数値の変化を確認する

オレンジボード

シリアルモニタ 行を変える

表示する

土壌水分 場所 左上

種類 実測値

2 水分量の割合が20%以下だったときにLEDが光るようにする

オレンジボード

シリアルモニタ 行を変える

表示する

土壌水分 場所 左上

種類 割合(%)

エリア < 土壌水分 場所 左上

種類 割合(%)

もし なら LEDが 場所 右下

光る

でなければ LEDが 場所 右下

光らない

このコネクタを使う方法

1 スクーミーのアプリが入ったPCを準備します
スクーミブロックエディタを開いてください



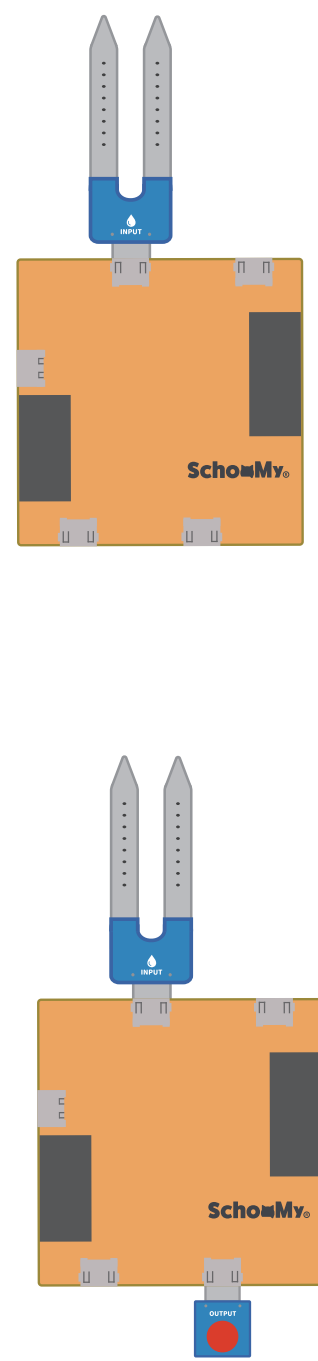
2 スクーミーボードと書き込み機を準備します
イラストのようにつけてください



3 QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロードすることができます





3 水分量の割合が30%より大きい場合は赤色、10%から30%のときは緑色、それ以下の時は青色に光る

オレンジボード

数の変数 i

7SEG 場所 右下

表示する i

エリア 30 ≤ i

もし なら フルカラーLEDの 場所 左下

光る早さ ふつう

色 あか

エリア and どちらも i < 30

10 ≤ i

もし なら フルカラーLEDの 場所 左下

光る早さ ふつう

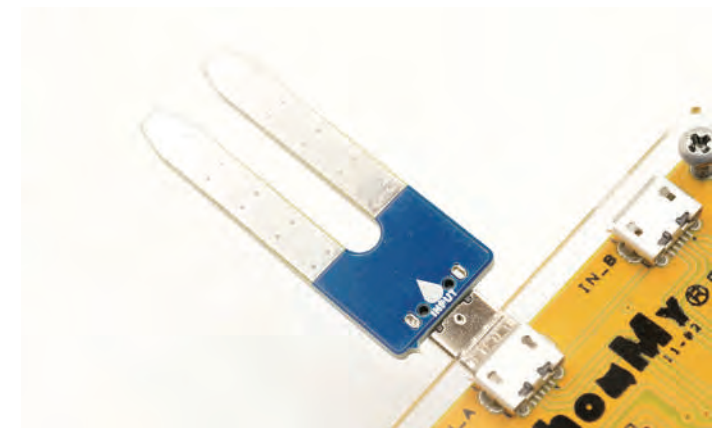
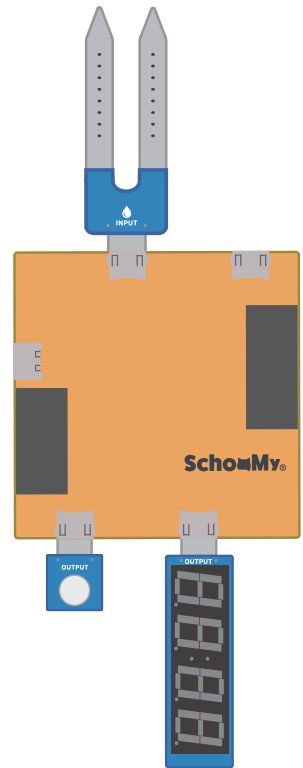
色 みどり

エリア i < 10

もし なら フルカラーLEDの 場所 左下

光る早さ ふつう

色 あお



INPUT

3 明るさセンサー

パソコンなどで、周りが暗いと画面が明るくなる仕組み



どのくらいセンサーに光が当たっているかを見ることができます

CdSセルとよばれることもあります

仕組みを知って使いこなそう！



SchoMy®

このグレーの部分に光が当たると、スクーミーボードが読み取る値が変化します

読み取った数値を、LUX(ルクス)という単位になるように計算しています。パソコンに50と表示されていたら、50LUXということです。

明るくなると数値は大きくなり、暗くなると数値は小さくなります。

明るさの変化がわかると、こんなこともわかります

- 明暗
- 透過度
- 回転
- 開閉
- 影
- 色

暮らしやすい便利をつくるためのモノづくり

暗い部屋で段差に近づいたら光って教えてくれる

明るさセンサーと距離センサーの2つを活用し、部屋が暗いときに段差に近づいたら、足元が光り注意を促す仕組み



距離センサー

LED

1 センサーの周りを10Luxになるように暗くしよう

オレンジボード

シリアルモニタ

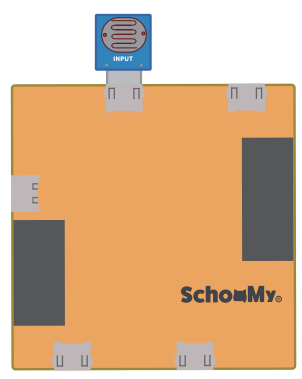
行を変える

表示する

明るさ

場所

左上



2 10Luxより小さくなったときに、LEDが光るようにする

オレンジボード

シリアルモニタ

行を変える

表示する

明るさ

場所

左上

エリア

明るさ

場所

左上

< 20

LEDが

場所

右下

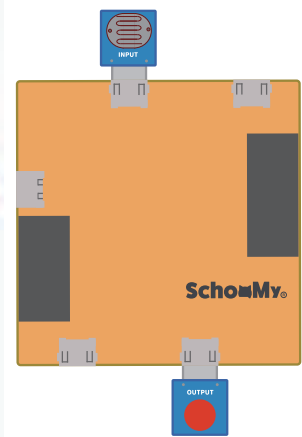
光る

LEDが

場所

右下

光らない



このコネクターを使う方法

スクーミーのアプリが入ったPCを準備します
スクーミーブロックエディタを開いてください



QRコードを読みこんで、使い方を学んでください

QRコードを読みこむことでプログラムをダウンロードすることができます

3 5秒間暗くなったらスピーカーから音が鳴る

オレンジボード

シリアルモニタ

行を変える

表示する

明るさ

場所

左上

エリア

明るさ

場所

左上

< 50

の関わりかえす

待つ

ミリ秒

1000

数の変数

i

1つ増やしてセット

エリア

1

=

5

なら

音が

場所

右下

出る

もし

音が

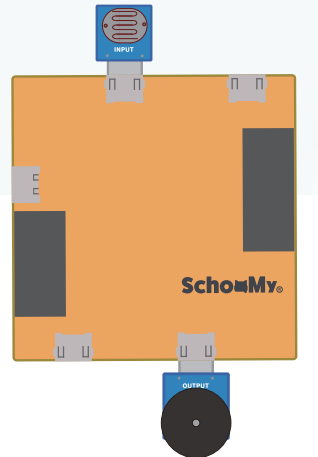
場所

右下

止まる

±0オクターブ

ふつうの長さ



コントロール

ブルーボード

オレンジボード

ずっと動くブロックを使ったプログラミング

同じプログラムを何回も繰り返すときに必要なブロックです。

- このブロックで LED をずっと光らせることができます。
- 2つ使うとうまくプログラムができなくなります。

LED がずっと光るようにしよう

「スクーミー」をリアルモニタに表示させよう



コントロール

ブルーボード

オレンジボード

最初一回だけ動くブロックを使ったプログラミング

最初に1回だけ動かすときに必要なブロックです。

- ずっと動くブロックよりも先に動きます。
- 最初に1回だけ動かしたいときに使います。

LED が 5 回点滅したあと、点滅をやめて光り続けるようにしよう



下のプログラムでは、一回だけ動くブロックのなかの指定回数くりかえすブロックは 5 回くりかえすため、LED は 5 回点滅します。指定回数くりかえすブロックをぬけると、一回だけ動くブロックからずっと動くブロックへ移行して LED が光り続けるようになります。

一回だけ動くブロックのなかで LED が点滅した数だけ、ずっと動くブロックのなかで LED をはやく点滅させよう



下のプログラムでは、タッチセンサーにふれるとエリアの間くりかえすブロックをぬけて、一回だけ動くブロックからずっと動くブロックへ移行します。一回だけ動くブロックのくりかえしの中で変数 i を増やしていたことで、ずっと動くブロックでは i の数だけくりかえします。



もしならブロックを使ったプログラミング

ブロックが動くための条件を指定するときに必要なプログラムです。

- 条件が満たされたときの動作を設定できます。
- 明るさセンサーが反応したときにシリアルモニターに文字を表示させることができます。

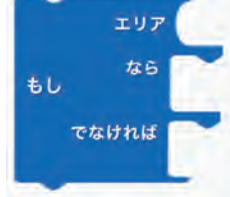
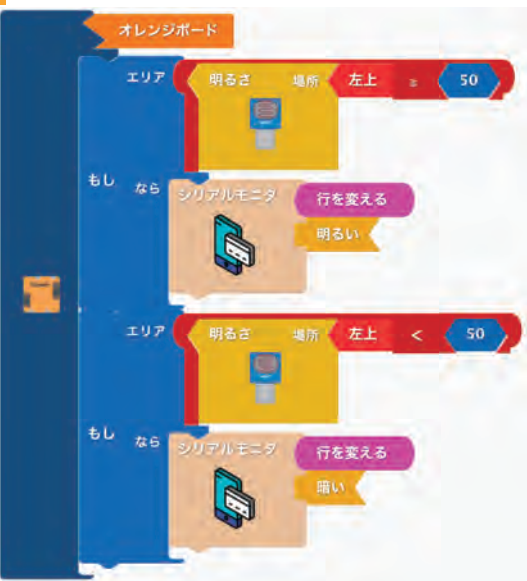
明るさの値が 50 より大きいなら、「明るい」とシリアルモニタに表示しよう



明るさの値が 50 以上（50 をふくむ）なら、「明るい」とシリアルモニタに表示しよう



明るさの値が 50 以上なら、「明るい」とシリアルモニタに表示して、50 より小さいなら、「暗い」とシリアルモニタに表示しよう



もしならでなければブロックを使ったプログラミング

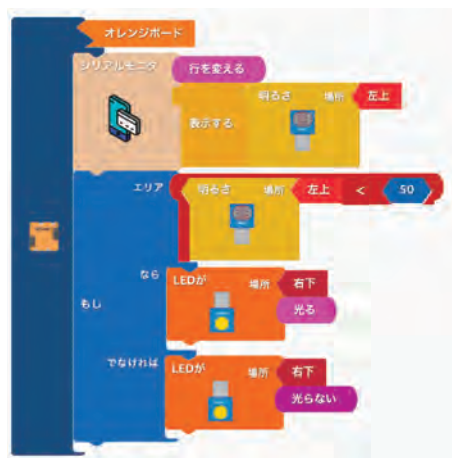
ブロックが動くための条件を指定するときに必要なプログラムです。

- 条件が満たされたときと満たされていないときで別の動きを設定できます。
- 明るさセンサーの値によって LED を光らせたり消したりできます。

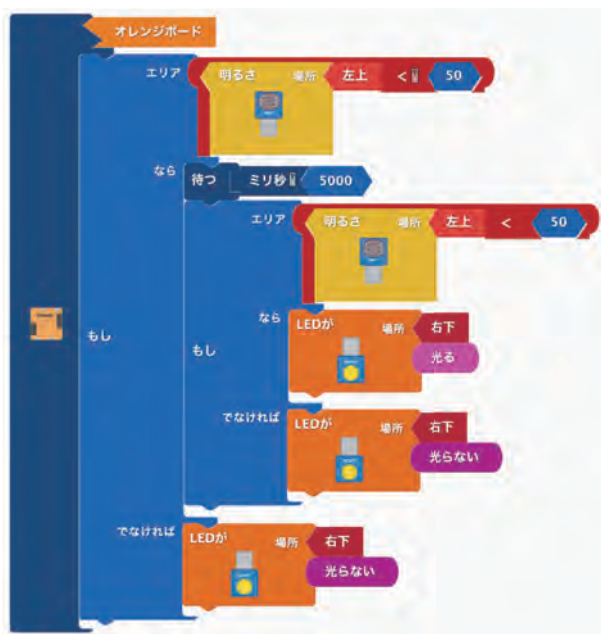
明るさの値が 50 より大きいなら、LED が光り、 そうでなければ LED が消えるようにしよう



明るさセンサーの値をシリアルモニタに表示させながら、明るさの値が 50 より小さくになったら LED が光るようにしよう



明るさの値が 50 より小さいなら、5 秒待ち、まだ明るさの値が 50 より小さいなら、 LED が光るようにしよう。そうでなければ LED が消えるようにしよう



指定した回数繰り返すブロックを使ったプログラミング

指定した回数繰り返すときに必要なプログラムです。

- 中にあるブロックを複数回繰り返して実行できます。
- 繰り返す回数を指定することができます。

タッチセンサーに触れたら、LED が 5 回だけ光るようにしよう

タッチセンサーに触れると、指定した回数くりかえすブロックが実行されます。下のプログラムでは counter が5になっているため、LED は 5 回だけ点滅します。



タッチセンサーが 1 になるごとに、3 段階で LED の点滅の速さや回数が変わるようにしよう

下のプログラムではタッチセンサーに触れた回数によって変数 i が 1 ずつ増えていき、変数 k には i を 3 で割った余りが代入されます。変数 k は 0、1、2 のいずれかであるため、それぞれの場合で LED の点滅の間隔と回数を設定しています。



エリアの間繰り返すブロックを使ったプログラミング

指定した条件の間繰り返すときに必要なプログラムです。

- 中にあるブロックを複数回繰り返して実行できます。
- 繰り返すための条件を指定することができます。

LED を 5 回だけ光るようにしよう

一回だけ動くブロックは、ずっと動くブロックが実行される前に一度だけ実行されます。変数 i は 0 が初期値として設定されているため、エリアの間くりかえすブロックの中身は 5 回実行されます。



明るさの値が 50 より小さいときだけ LED が点滅するようにしよう

くりかえす条件にセンシングを使用している例です。明るさセンサーから 50 未満の数を受け取ったときのみ、エリアの間くりかえすブロックの中身がくりかえし実行されます。



繰り返しを抜けるブロックを使ったプログラミング

繰り返しから抜けるときに必要なプログラムです。

くりかえしを抜ける

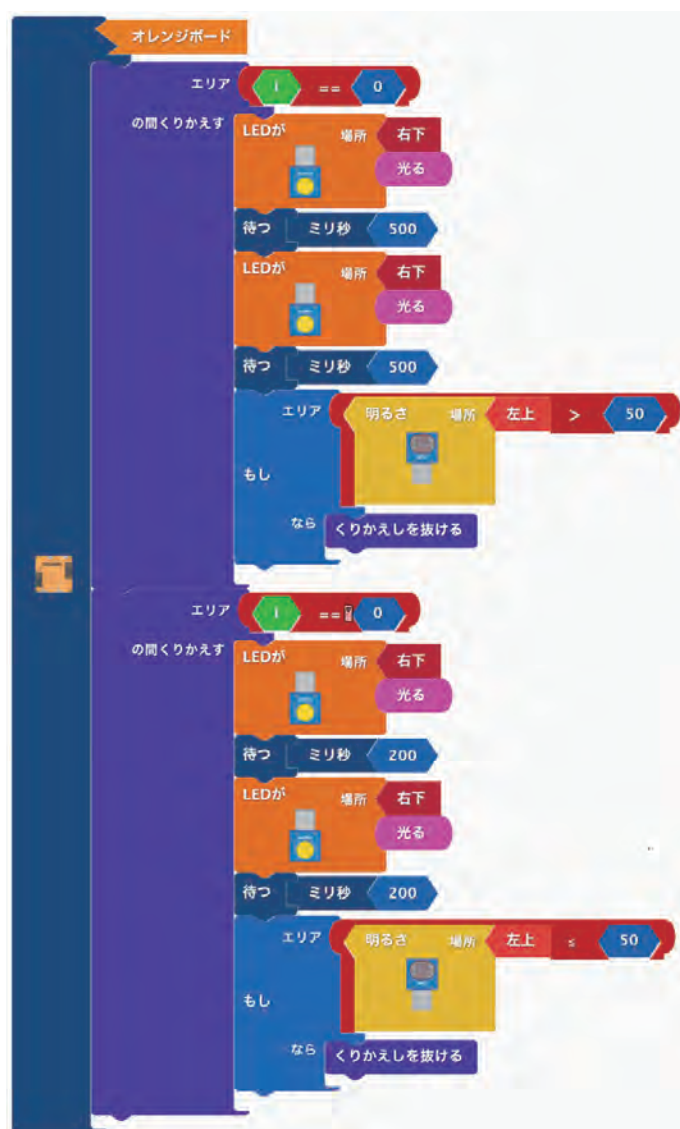
- 繰り返すブロックの中に入れることで繰り返しを抜けることができます。
- もしならブロックと組み合わせてセンサーが反応したら繰り返しを抜けるといったことができます。

タッチセンサーを触ったら繰り返しを抜けて、LED の点滅が止まるようにしよう



暗いときと明るいときの点滅の速さを変えよう

下のプログラムは、 明るいときに点滅が 0.5 秒間隔、暗いときに 0.2 秒間隔で LED が光るプログラムです。
i ブロックは実行時に 0 で初期化されるため、繰り返しを抜けるブロックでのみ繰り返しを抜けることができます。



待つブロックを使ったプログラミング

時間を調整して待つときに必要なプログラムです。

待つ

- 時間を決めてプログラムを待つときに使います。
- 時間の単位も変更できます。

1 秒ごとに「スクーミー」をシリアルモニタに表示させよう



1 秒ごとに「スクーミー」と「ボード」を表示させよう



2 秒ごとに「スクーミー」をシリアルモニタに表示させよう



表示する

ブルーボード

オレンジボード

シリアルモニタブロックの使い方

シリアルモニタに表示するときに必要なプログラムです。

- シリアルモニタにセンサーで測った値を表示することができます。
- 値だけでなく文字なども表示することができます。

「表示する」をシリアルモニタに表示しよう



「スクーミー」をシリアルモニタに表示しよう



スクーミーと表示させたあとに、ボードと表示させよう



表示する

ブルーボード

オレンジボード

7SEG ブロックを使ったプログラミング

7SEG に表示するときに必要なプログラムです。

- センサーの値を 7SEG に表示することができます。
- 小数点やマイナスも表示することができます。

「1」を 7SEG に表示しよう



「1031」を 7SEG に表示しよう



1 秒ごとのカウントダウン（3→2→1→0）を 7SEG に表示しよう



表示する

ブルーボード

オレンジボード

グラフモニタブロックの使い方

グラフモニタに表示するときに必要なプログラムです。

- センサーで測った値をグラフモニタに表示することができます。
- 2つのセンサーの値を表示して、視覚的に比較することができます。

グラフモニタブロックの使い方

ブロックメニューの「表示する」から「グラフモニタ」をドラッグアンドドロップし「ずっと」につけます。
表示したコネクタのブロックを選んで、「グラフモニタ」につけます。



上限値と下限値を設定してみよう

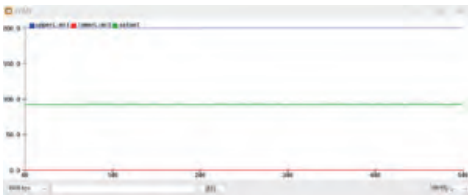
STEP.1 グラフのスケールを設定する

上限の値と下限の値を調整します。（画像では上限の値を 200 に変えています）
「アップロード」をクリックします。



STEP.2 シリアルモニタでグラフを確認する

書き込みが完了したら「シリアルモニタ」を押します。グラフに明るさの値（緑線）が表示されています。



STEP.3 上限値を変えて表示を確認する

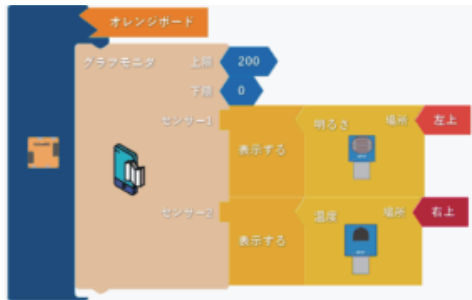
上限の値を 600 に変えてグラフに表示してみます。
上限の値が 600（青線）、下限の値が 0（赤線）と表示されグラフの大きさが変わりました。



2つのセンサーの値をグラフモニタに表示してみよう

STEP.1 センサー-1 とセンサー-2 にブロックをつなぐ

「センサー-1」と「センサー-2」にそれぞれ表示させたいブロックをつけてアップロードをクリックします。



STEP.2 複数のセンサー値を確認する

今回の例では、以下のように明るさセンサーの数値が緑線、温度センサーの数値がオレンジ線で同時に表示されます。



表示する

ブルーボード

オレンジボード

OLED ブロックを使ったプログラミング

OLED に表示するときに必要なプログラムです。

- OLED にセンサーの値を表示することができます。
- アルファベットも表示することができます。

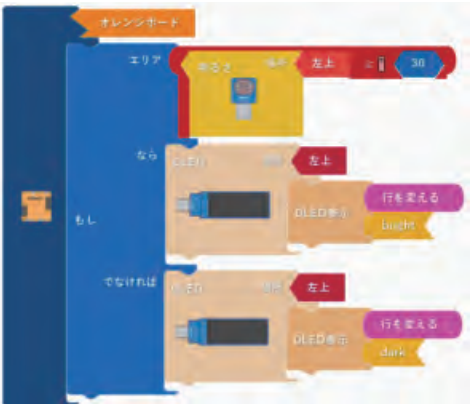
OLED で、部屋の明るさを表示しよう

下のプログラムでは、明るさセンサーから取得した値を 1 秒ごとに表示します。



部屋が明るかったら「bright」、暗かったら「dark」と表示しよう

OLED で文字を表示するプログラムです。OLED 表示ブロックにセンシングなどをセットせずに使用することで、文字を表示することができます。（初期は「display」、半角文字に限る）。
下のプログラムでは、明るさセンサーから受け取る値が 30 以上のときに「bright」、30 より小さいときには「dark」と表示します。



アクション

ブルーボード

オレンジボード

LED ブロックを使ったプログラミング

LED を光らせたり、 消したりするときに必要なプログラムです。

- LED を光らせて、 ずっと点灯させることができます。
- LED を光らせたり消したりして、 点滅させることができます。

LED を右下につけて光るようにしよう

LED を左下につけて光るようにしよう

LED をピカピカ光らせよう



アクション

ブルーボード

オレンジボード

スピーカーブロックを使ったプログラミング

スピーカーの音を鳴らしたり、 音を消すことができるプログラムです。

- 音の高さや長さを決めて、 好きな音を鳴らすことができます。
- スピーカーブロックをつなげると、 音楽をつくることができます。

ドの音を鳴らそう



やオクターブの設定を変えてアラートを作ろう



明るさが 10 より小さいなら、 シの音が鳴るようにしよう



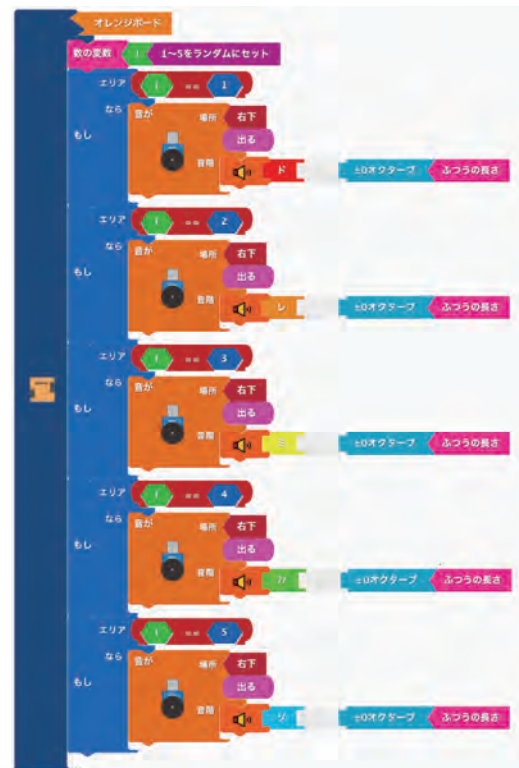
ド・レ・ミの音を鳴らそう



タッチしたらドの音が鳴るようにしよう



ドレミファソの音をランダムに出そう



アクション

ブルーボード

オレンジボード

フルカラー LED ブロックを使ったプログラミング

フルカラー LED をいろいろな色に光らせたり、消したりするときに必要なプログラムです。

- R (あか)、G (みどり)、B (あお) にそれぞれ数を入れて光らせることができます。
- RGB の数の大きさのバランスで光る色を決めることができます。

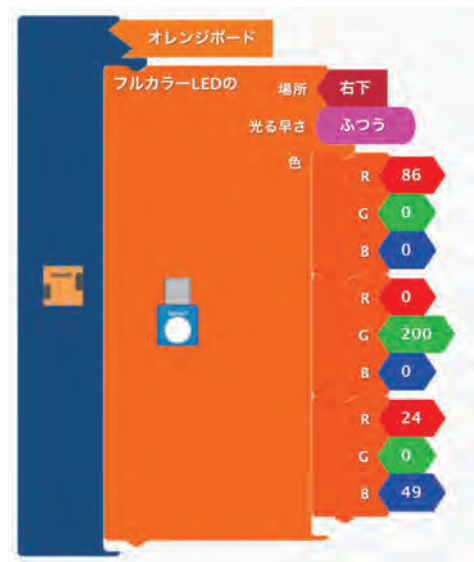
フルカラー LED を赤色に光らせよう



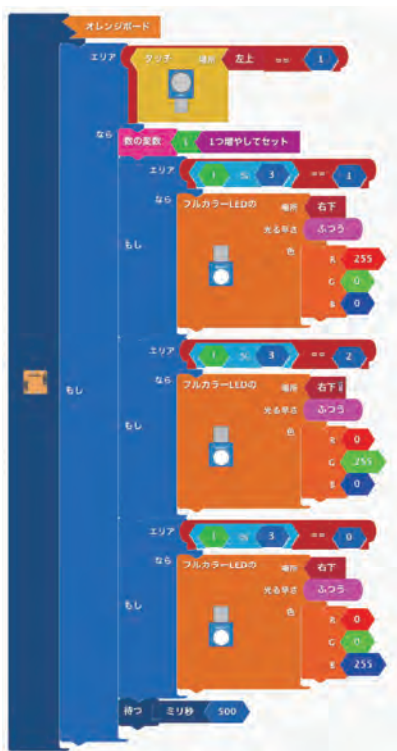
フルカラー LED が赤色に 1 秒光り、1 秒消える動きをくりかえそう



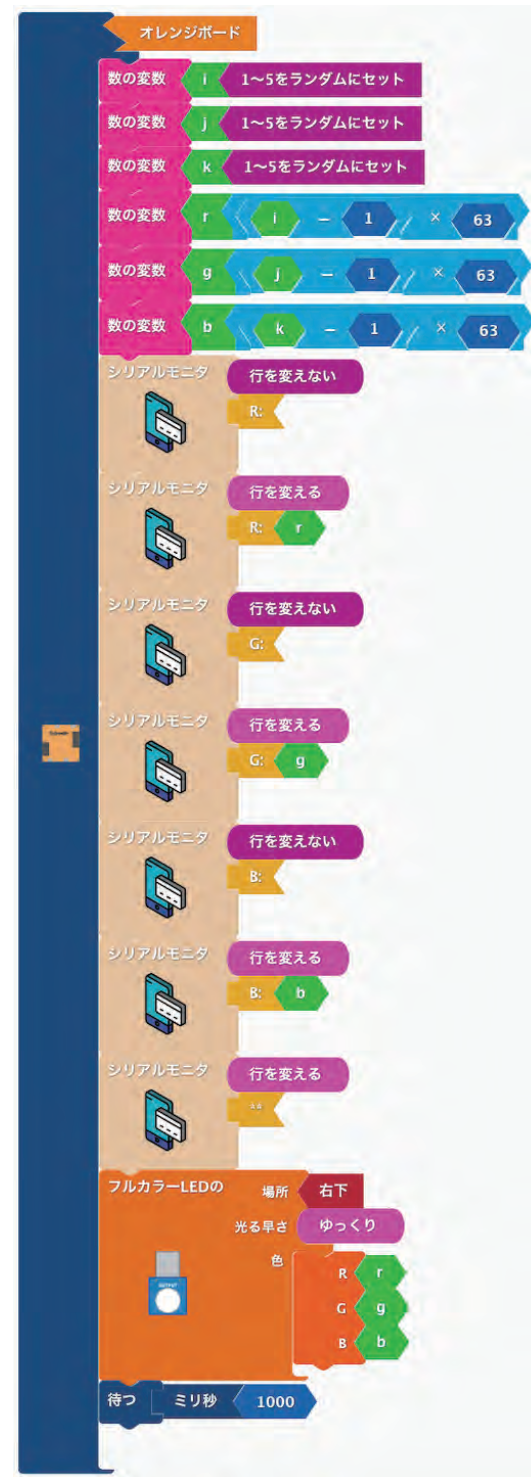
フルカラー LED を色々な色に変化させよう



タッチするたびにフルカラー LED の色が赤⇒緑⇒青にかわるようにしよう



フルカラー LED の色が 1 秒ごとランダムに変わるようにしよう



アクション

ブルーボード

オレンジボード

フルカラー色選択ブロック

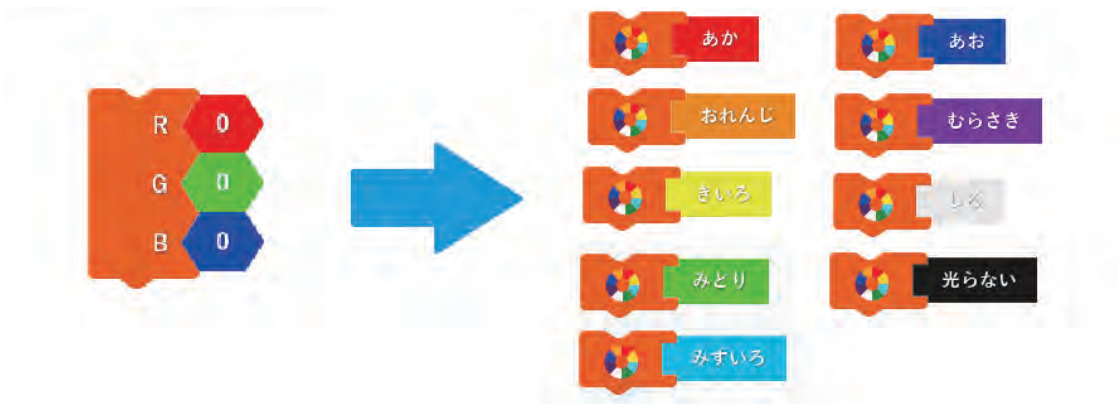
フルカラー LED の色をかんたんにえらぶことができるプログラムです。

- 「あか」「おれんじ」「きいろ」などの色を決めて光らせることができます。
- 8 色の色と、「光らない」からえらぶことができます。

フルカラー色選択ブロックとは

RGB ブロックでは R・G・B それぞれに数を入れて、色を作っていました。

フルカラー色選択ブロックでは、あらかじめ作られた色をえらぶ事で簡単に使うことができます。



使い方

STEP.1

- ①「アクション」メニューから
- ②フルカラー色選択ブロックを取り出す



STEP.2

フルカラー LED ブロックの「色」につなげる

フルカラー LED ブロックの「色」と書かれた部分の切り欠きに色選択ブロックをつなげます。



STEP.3

プルダウンメニューで色をかえる

ブロックにマウスカーソルを合わせると下三角のアイコンが出てくるのでクリックします。

色のメニューが出てくるので、選択して使用します。



イルミネーションを作ろう

フルカラー色選択ブロックを上下につなげる事で、フルカラー LED が様々な色に変化します。



振動モーターブロックを使ったプログラミング

振動モーターを振動させるときに必要なプログラムです。

- 振動モーターをずっとふるえさせることができます。
- 交互にふるえさせて、リズムをきざむことができます。

振動モーターを ON にしよう

振動モーターがずっと振動します。



タッチセンサーで振動モーターの ON と OFF を切り替えよう

下のプログラムでは、タッチセンサーに触れると振動モーターが ON、タッチセンサーに触れていないときは OFF が出力されます。



明るさによって振動するリズムを変えよう

下のプログラムでは、明るいとき（明るさセンサーが 50 以上の数値を示すとき）は 1 秒間隔で振動し、暗いとき（明るさセンサーが 50 未満の数値を示すとき）は 0.3 秒間隔で振動します。



DC モーターブロックを使ったプログラミング

DC モーターを回したり止めたりするときに必要なプログラムです。

- 方向を決めて「回る」「逆に回る」のどちらかにモーターをずっと動かすことができます。
- 「止まる」ことでモーターを止めることができます。

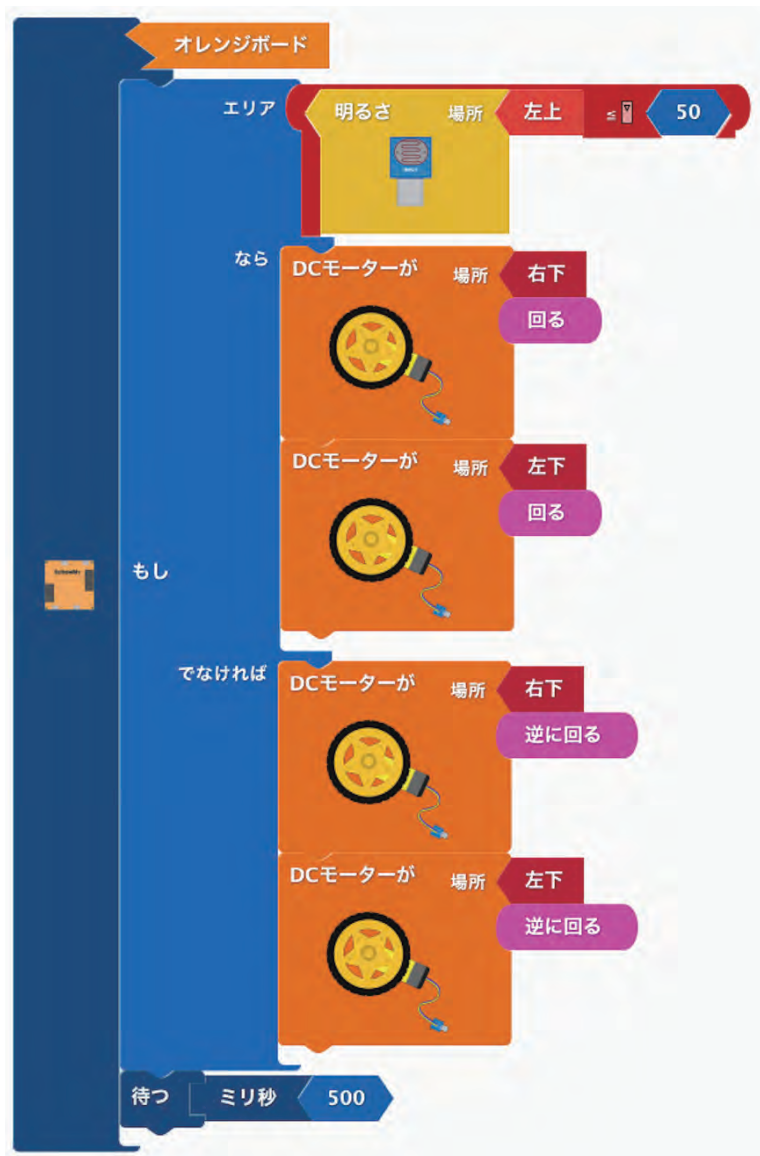
モーターを回してみよう

DC モーターブロックは、「回る」、「逆に回る」、「止まる」のいずれかを選択することで、モーターを制御します。



明るいところと暗いところを往復する車をつくろう

下のプログラムは、二輪の自動車が暗い場所と明るい場所を往復するような想定で作成しています。暗い（明るさセンサーから 50 以下の値を取得した）場合にモーターが回転し、明るい（明るさセンサーから 50 より大きい値を取得した）場合にモーターが逆向きに回転します。待つブロックの時間を長くすることで、往復距離を大きくすることができます。



回転サーボモーターブロック

回転サーボモーターの回転のさせ方を制御するときに必要なプログラムです。

- 回転する速さと方向、 回る時間を決めてモーターを回すことができます。
- ある位置で角度を決めて回転サーボモーターを止めることもできます。

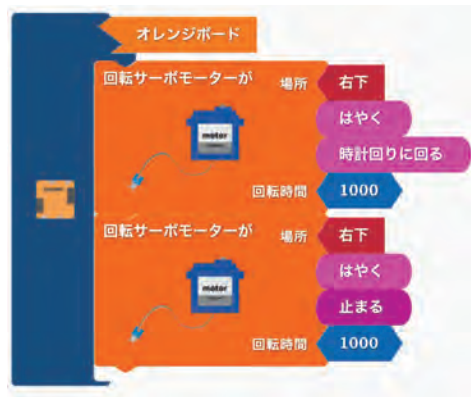
サーボモーターをはやく時計回りにまわそう

サーボモーターを早く時計回りにまわすプログラミングです。



1 秒回って、 1 秒止まらせよう

サーボモーターを 1 秒回らせて、 1 秒止まらせるプログラミングです。



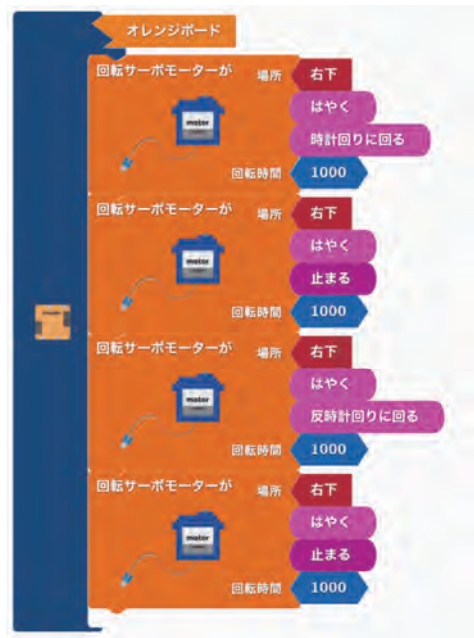
振り子の動きを作ってみよう

振り子の動きを作るプログラミングです。



1 秒ごとに時計回り⇒止まる⇒反時計回り⇒止まる をくりかえそう

1 秒ごとに時計回り→止まる→反時計回り→止まる をくりかえすプログラミングです。



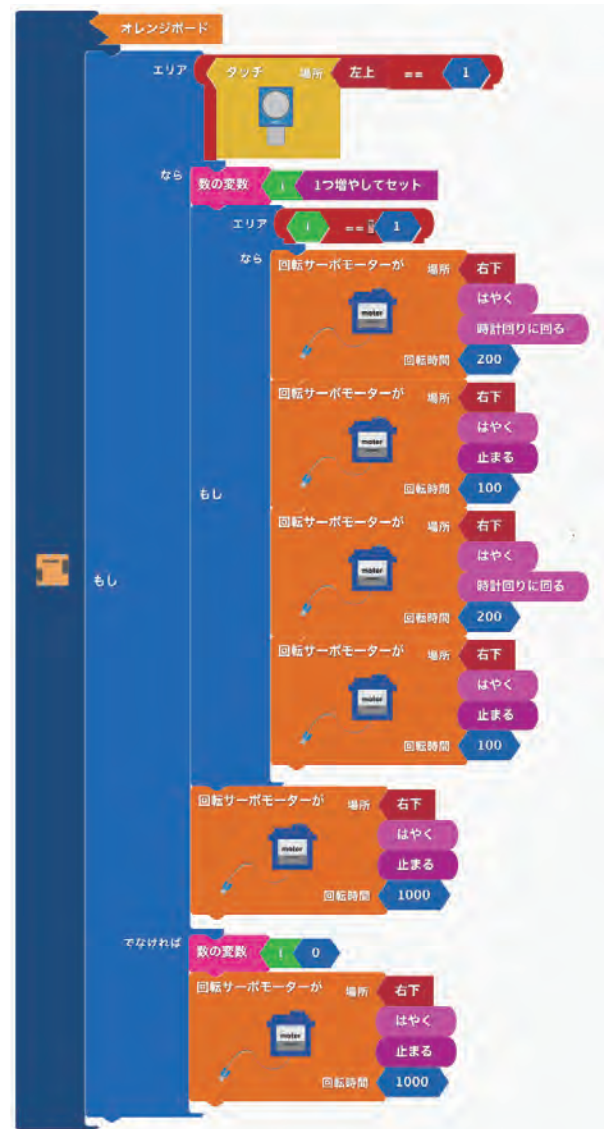
温度が 25 度より大きくなったらサーボモーターを時計回りに回転させよう（温度感知扇風機）

温度が 25 度より大きくなったらサーボモーターを時計回りに回転させるプログラミングです。



タッチコネクタにふれたら、 セットした紙飛行機が飛ぶセンサーを作ろう

タッチコネクタにふれたら、 セットした紙飛行機が飛ぶプログラミングです。



スペック

トルク：0.147Nm(4.8V 時)
 回転速度：130RPM(6V 時)、100RPM(4.8V 時)
 動作電圧：4.8V ~ 6V

180° サーボモーターブロック

180° サーボモーターの角度を制御するときに必要なプログラムです。

- サーボモーターの角度を 0 度から 180 度まで決めて動かすことができます。
- ある位置で角度を決めて回転サーボモーターを止めることもできます。

サーボモーターの角度を 90 度に見よう

サーボモーターの角度を 0 度から 180 度まで、以下の 5 つの角度に変更することができます。

0 度、45 度、90 度、135 度、180 度



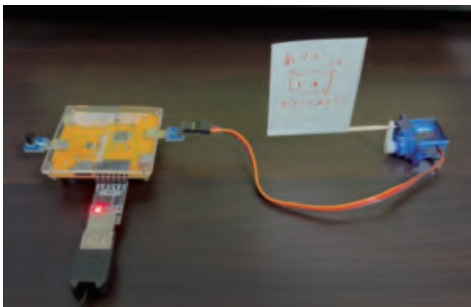
振り子の動きをさせてみよう

下のプログラムでは、振り子のようにサーボモーターの角度が一秒ごとに 45 度と 135 度に切り替わります。



温度が上がったら知らせる装置をつくろう

下のプログラムでは、温度センサーの温度が、28 度より小さい場合に 0 度で旗が横になっている状態、28 度より大きい場合に 90 度で旗が縦の状態になります。



USB-A ブロックを使ったプログラミング

スクーミーボードに USB-A を接続して制御することができるプログラムです。

- USB-A を接続して動かすことができる機器を制御できます。
- LED 照明や小型扇風機の電源をつけたり、消したりできます。

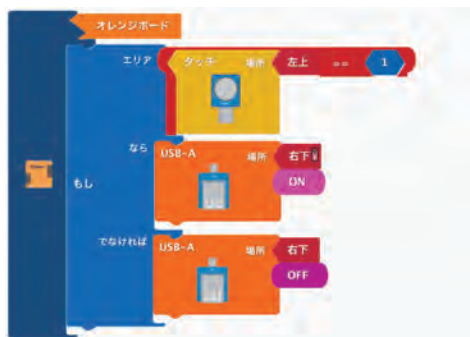
接続した USB-A を ON にしよう

USB-A コネクタに接続するだけでは USB-A は ON にはなりません、下のプログラムをスクーミーボードに書き込むことで ON にすることができます。



タッチセンサーに触れているときだけ USB-A を ON にしよう

タッチセンサーに触れているときは USB-A を ON にし、離れたときは OFF にするプログラムです。
ただし、PC の制御外で ON と OFF をくりかえすため、USB メモリなどのデータ破損を引き起こす可能性がある記憶媒体の使用は避けてください。



プロペラブロックを使ったプログラミング

プロペラを回したり止めたりするときに必要なプログラムです。

- プロペラをずっと回し続けることができます。
- 「止まる」 ことでプロペラを止めることができます。

プロペラを回してみよう

5 秒間隔でプロペラが回ると回らないを繰り返すプログラムです。



暑いときにプロペラが回るようにしよう

下のプログラムでは、温度センサーが 25 以上の値を示したときプロペラが回り、25 未満の値を示したときプロペラが回らなくなります。シリアルモニタで温度を確認しながら、基準とする数値を設定してみましょう。



ポンプブロックを使ったプログラミング

ポンプを動かして水を出したり止めたりするときに必要なプログラムです。

- ポンプををずっと動かして、 水を出し続けることができます。
- センサーを使って、 自動で水を供給することができます。

タッチセンサーで、 流れる水をコントロールしよう

タッチセンサーに触れると 1 が入力されるため、タッチセンサーに触れるとポンプから水が出て、 離すと水が止まります。



土の水分量が減ったときに、 水を追加するプログラムを作ろう

土壌の水分量が減少した時に、 自動で水を供給するプログラムです。 下のプログラムでは、土壌水分コネクタの値と 30 を比較していますが、シリアルモニタを見て適切な数に変更しましょう。



アクション

ブルーボード

オレンジボード

LED バー範囲指定ブロックを使ったプログラミング

10 メモリある LED バーの光らせる範囲を指定するときに必要なプログラムです。

- 最小値と最大値を指定することで、その範囲の LED を光らせることができます。
- 「すべて光らない」を選択すると、範囲に関係なくすべてが光らせないことができます。

LED バー範囲指定ブロックとは？

1 番目から 10 番目の指定した範囲の LED（例えば、5 番目から 10 番目の LED など）をまとめて光らせます。



LED バー範囲指定ブロックの使い方

最小値と最大値を指定することで、その範囲の LED を光らせます。範囲は、**最小値が「1」、最大値が「10」**になっています。それ以外の値（1 ～ 10 以外）が設定されると、**すべて光らない状態**になります。LED バーの位置がスクリーンボード側なのが「**最大値**」、その反対側が「**最小値**」です。なお、最小値とは一番小さい値のことで、最大値とは一番大きい値のことです。



1 番目から 10 番目の LED を光らせよう

下のプログラムでは、指定した範囲の LED バーが光った状態になります。このプログラムの場合、最小値を 1、最大値を 10 に指定しているため、1 番目から 10 番目のすべてが光った状態になります。



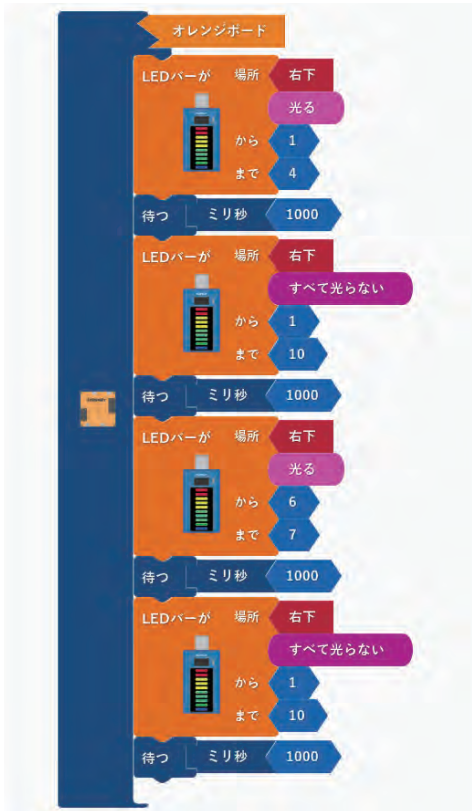
1 番目から 10 番目の LED を光らせよう

下のプログラムでは、LED バーが光らない状態になります。「すべて光らない」を選択しているので、範囲に関係なくすべてが光らない状態になります。



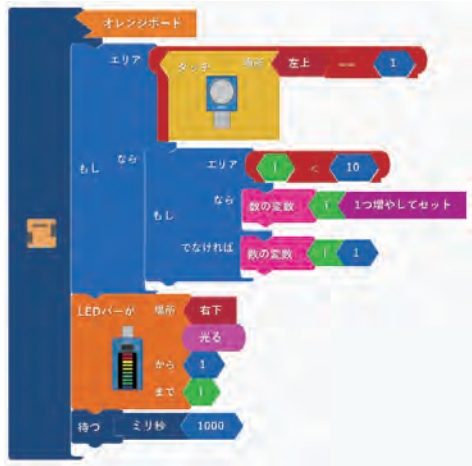
「待つ」ブロックを使って、光る範囲を変化させよう

下のプログラムでは、複数の LED バー範囲指定ブロックを並べることで、1 秒ごとに光る範囲が変化していきます。はじめに、1 番目から 4 番目の範囲が光り、次に 6 番目から 7 番目の範囲が光り、それが繰り返されます。



タッチセンサーをタッチすると点灯範囲が増えていくようにしよう

下のプログラムでは、タッチセンサーをタッチすると、LED の光る範囲が一つずつ増えていきます。最大値の 10 番目まで光った後は、最小値の 1 番目から光る範囲が一つずつ増えていくことを繰り返します。



アクション

ブルーボード

オレンジボード

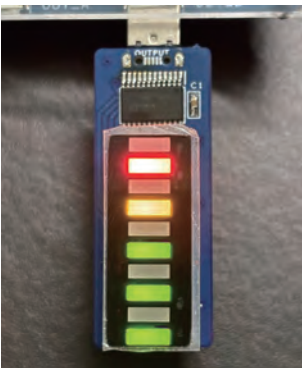
LED バー個別制御ブロックを使ったプログラミング

10 メモリある LED バーの光らせる場所を指定するときに必要なプログラムです。

- 奇数番目のメモリの LED を光らせたり消したりすることができます。
- LED バーの「位置」は、ボードに接続する側の遠い場所から 1 ～ 10 と数えます。

LED バー個別制御ブロックとは？

指定した位置の LED(例えば、1 番目の LED や 9 番目の LED、奇数番目の LED など) を光らせたり消したりすることができます。



LED バー個別制御ブロックの使い方

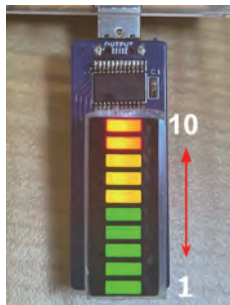
LED バー個別制御ブロック専用の光る・光らないブロックを使って、LED バーを制御することができます。



光るブロック

光る・光らないブロックに繋がっている制御位置ブロックを使って、LED バーを制御する位置を操作することができます。

なお、LED バーの「位置」は、ボードに接続する側の遠い場所から 1 ～ 10 と数えます。

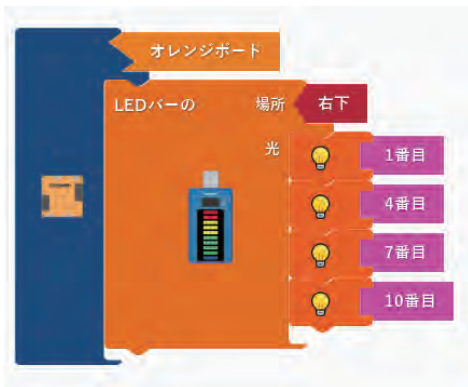


また、LED バー個別制御ブロック専用の待つブロックを使って、次の動作に移るまでの待機時間を設定することができます。



複数の LED を同時に光らせよう

下のプログラムでは、1 番目、4 番目、7 番目、10 番目の LED が同時に光ります。もしも順番に光らせたい場合は、それぞれの光るブロックの間に待つブロックを入れてください。



タッチするたびにランダムな位置を光らせよう

下のプログラムでは、タッチセンサーに触れるたびにランダムな位置の LED が光ります。タッチセンサーに触れ続けた場合、1 秒ごとに新たなランダムな位置の LED が光ります。既に光っている位置が選ばれた場合は、その位置の LED は光った状態のままです。タッチセンサーから手を離すと、すべての LED が消えます。



1 番目の LED を点滅させよう

下のプログラムでは、1 番目の LED が 1 秒ごとに点滅します。待つブロックを入れていないと、1 番目の LED がずっと光っているように見えるので、点滅させたい場合は待つブロックを忘れないようにしてください。



奇数番目と偶数番目の LED を交互に光らせよう

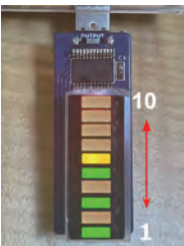
下のプログラムでは、奇数番目の LED と偶数番目の LED が交互に光ります。まず、奇数番目 (1, 3, 5, 7, 9) の LED が 1 秒間光り、その後、偶数番目 (2, 4, 6, 8, 10) の LED が 1 秒間光ります。



LED バー制御パターン例

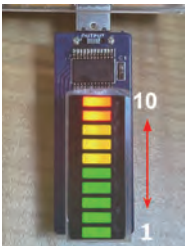
1 番目～ 10 番目

選択した番号の位置の LED を操作する



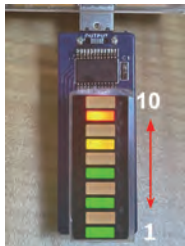
すべて

すべての LED を操作する(すべて光る・すべて光らない)



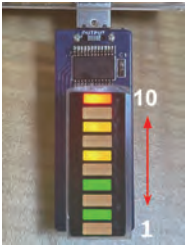
奇数番目

奇数番号 (1, 3, 5, 7, 9) の LED を操作する



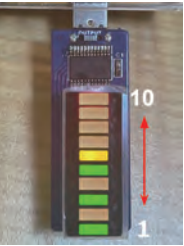
偶数番目

偶数番号 (2, 4, 6, 8, 10) の LED を操作する



ランダム

ランダムに選ばれた番号の LED を操作する



距離センサーブロックを使ったプログラミング

距離センサーを使用して距離を測るときに必要なプログラムです。

- 距離センサーから物や壁までの距離を測ることができます。
- センチメートル単位でシリアルモニタに距離を表示することができます。

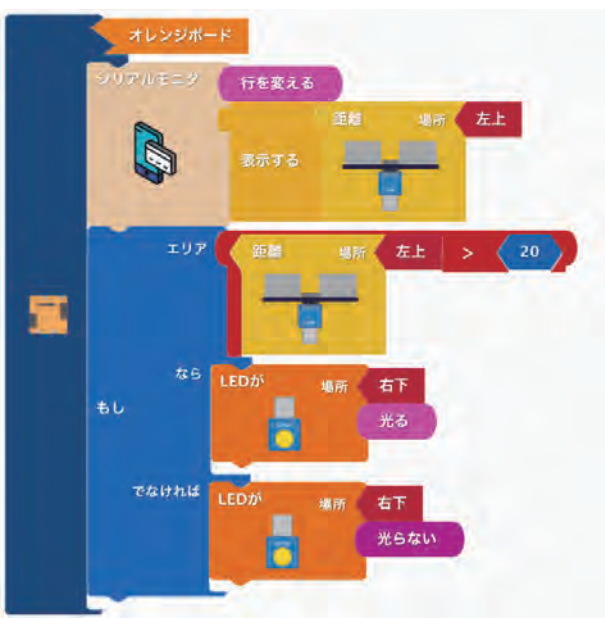
距離をモニターに表示する



距離を 7SEG に測定する



センサーからものまでの距離を 7SEG に表示し、
20cm より大きくなると LED が光る



ものがセンサーの前を通ると 7SEG の値が
1 ずつ増えるようにする



明るさセンサーブロックを使ったプログラミング

明るさセンサーを使用して明るさを測るときに必要なプログラムです。

- 明るさセンサーの置いてある場所の明るさを測ることができます。
- 明るいところほど数値が大きくなり、 暗いと数値が小さくなります。

明るさの値をシリアルモニタに表示しよう



明るさの値を 5 秒ごとに シリアルモニタに表示しよう



明るさの値をグラフに表示しよう



明るさの値に「LUX」という単位をつけて、
シリアルモニタに表示しよう



タッチセンサーブロックを使ったプログラミング

タッチセンサーを使用してセンサーに触れたかどうかを判断するときに必要なプログラムです。

- タッチセンサーにものが触れているか触れていないかを判別することができます。
- シリアルモニタで表示すると触れていたら 1、 触れていなかったら 0 と表示されます。



タッチセンサーにふれ、 0 から 1 になることを
シリアルモニタに表示しよう



タッチセンサーが 1 なら、 LED を光る
ようにしよう



タッチセンサーにふれ、
0 から 1 になることをグラフで表示しよう



明るさの値が 50 より小さい時に、 タッチセンサーが 1 なら、
LED が光るようにしよう



通過センサーブロックを使ったプログラミング

通過センサーを使用してセンサーの間を何かが通過したかどうかを判断するときに必要なプログラムです。

- 通過センサーをものが通過したかどうかを測ることができます。
- シリアルモニタでは通過センサーでものを検知したときは 1、 検知しないときは 0 と表示されます。



通過センサーの間を通過したとき、
モニターに 1 を表示させよう



通過したときに音を鳴らすプログラムを
作ろう

下のプログラムでは、 通過センサーが反応したとき
に音コネクタからドの音が出ます。



通過センサーの間を通過したときの
様子を、 グラフにしよう



通過センサーの間を通過したとき、
7SEG に 1 を表示させよう



通過した回数によって
フルカラー LED の色を変化させよう

通過センサーが反応すると変数 i が増加して、フルカラーLEDの RGB の値が変化していきます。
RGB の i にかける数を自由に変えて、 色の変化を見てみましょう。



センシング

ブルーボード

オレンジボード

加速度センサーを使ったプログラミング

加速度センサーを使用してセンサーの動く加速度を測るときに必要なプログラムです。

- 加速度センサーでは物の加速度を測ることができます。
- 加速度が大きくなることによって数値が大きくなります。

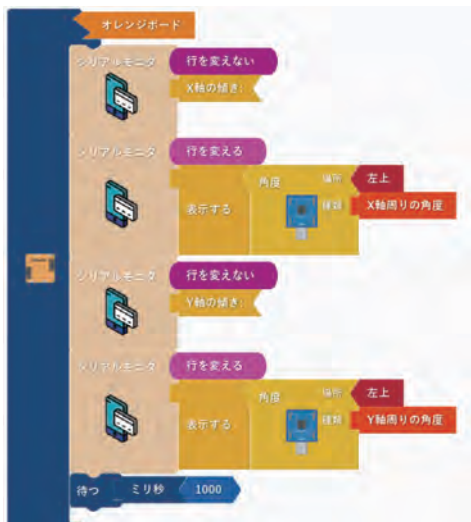
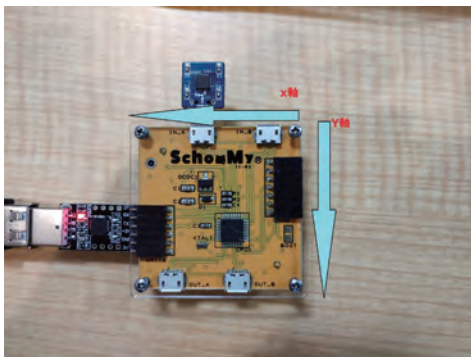
値の動きを見てみよう

シリアルモニタブロックに加速度センサーブロックにつなげて値を見てみましょう。

シリアルモニタブロックに何もつながらない時は入力した文字が表示されます。

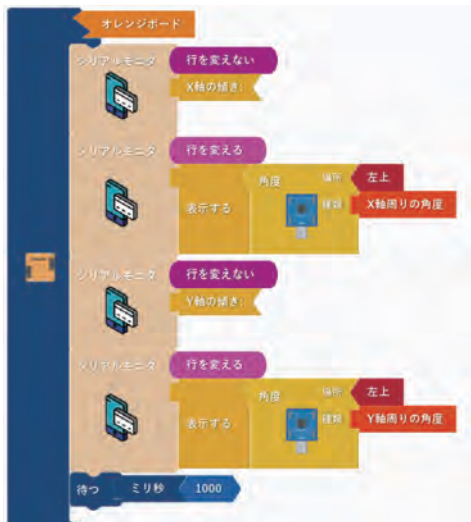
ボードを平らな場所に置いてシリアルモニタを見てみましょう。 x 軸、y 軸はほとんど値が動かず、z 軸は大体 1.00 前後の値を示すかと思います。これが重力加速度です。

ボードを横 ・ 縦に移動してみましょう。
横方向はx軸、 縦方向は y 軸の値が動きます。



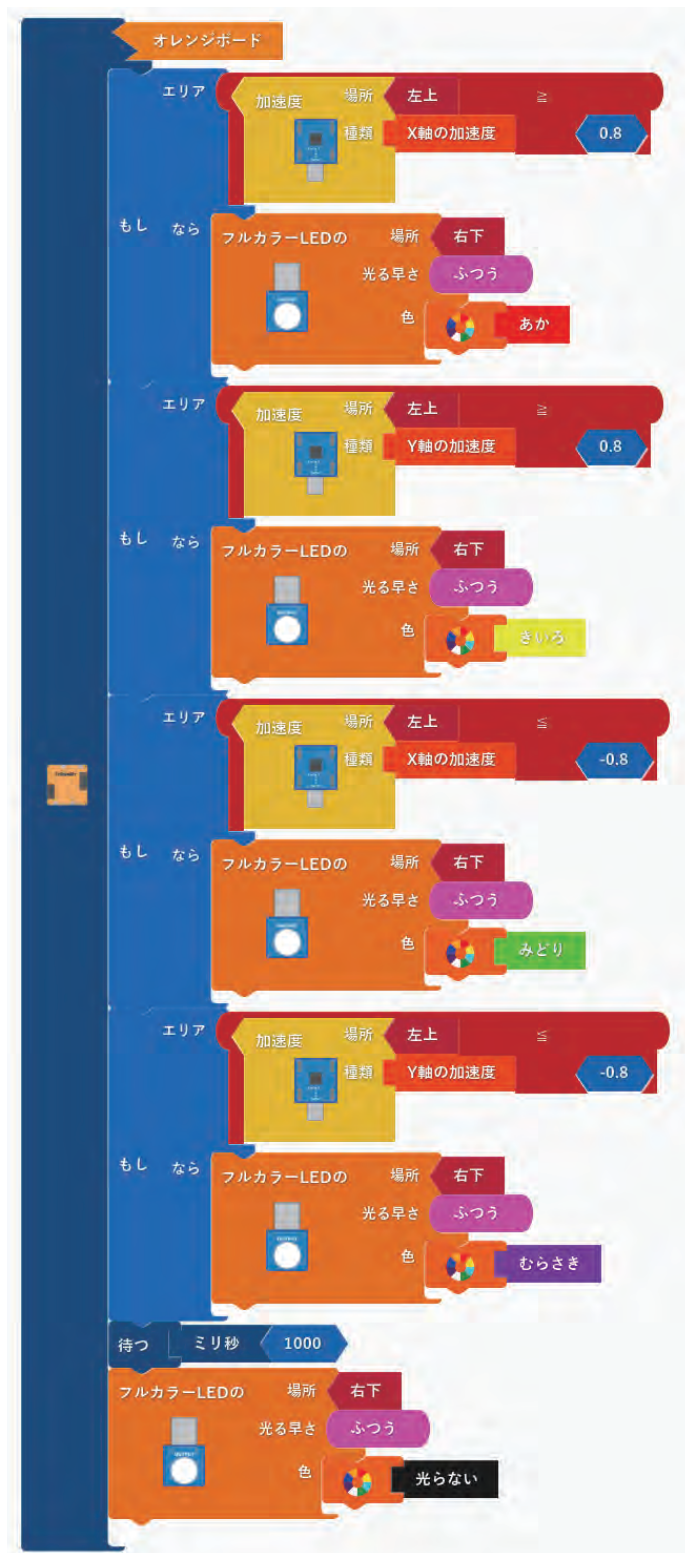
動いたら光るプログラムを作ろう

下のプログラムは、 加速度の大きさが 2 より大きくなった時に LED が光り、 2 以下になったら消えるようになっています。
動かない時の加速度は 1.00 なので、 1.00 より大きい加速度の値を設定すれば「動いたら光る」プログラムが出来ます。



ボードを傾ける方向によって4色に光るプログラムを作ろう

下のプログラムは、 ボードを水平から垂直に傾け上下左右どちらかの辺を下にしたかで光る色が決まります。
上だと黄色、 下だと紫、 左だと緑、 右だと赤色に光ります。



角度センサーブロックを使ったプログラミング

角度センサーを使用してセンサーの向いている角度を測るときに必要なプログラムです。

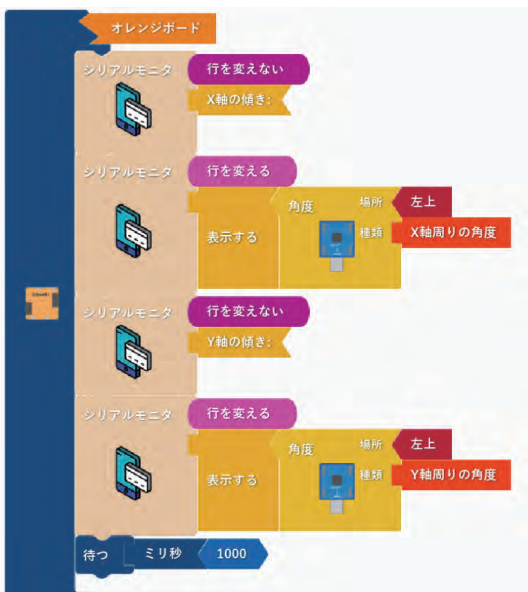
- 角度センサーを傾けた角度によって数字が変化します。
- X 軸方向、Y 軸方向、Z 軸方向のそれぞれの傾きを測ることができます。

オレンジボードが傾くのを検知しよう



オレンジボードの平面の傾きを X 軸、Y 軸に分けて取得することができます。
また、シリアルモニタはセンシングや数をセットしない場合、文字を表示することができます。

オレンジボードが水平じゃないとき LED が光るようにしよう



角度センサーはある方向へ回すとプラスに、逆の方向へ回すとマイナスとなります。
つまり、角度センサーを用いて水平を保つときにはプラス、マイナスになることを考える必要があります。

下のプログラムでは、X 軸周りの角度が 10 以上または -10 以下になると LED が光り、
また Y 軸周りの角度が 10 以上または -10 以下になるときも LED が光ります。

温度センサーブロックを使ったプログラミング

温度センサーを使用して温度を測るときに必要なプログラムです。

- 温度センサーのある場所の温度を測ることができます。
- シリアルモニタにはセルシウス温度で表示されます。

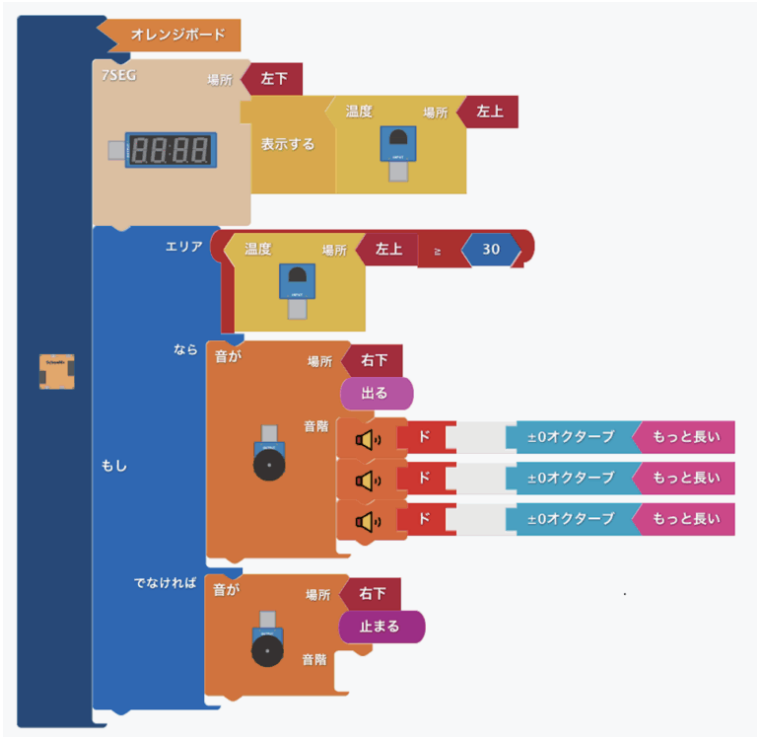
温度をシリアルモニターに表示する



温度をグラフで表示する



温度が 30 度以上になったら音が鳴る



温度の違いによって、鳴らすメロディを変える (今回の場合は、28 度以上か、30 度以上か)





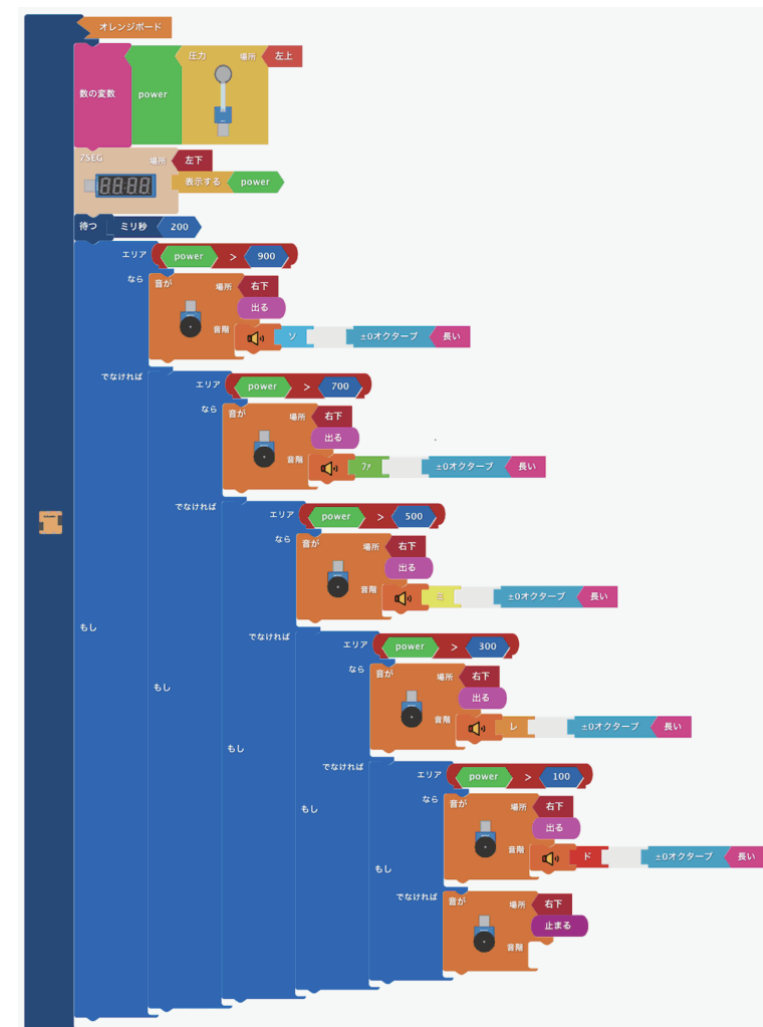
- 磁力センサーで磁力の大きさを測ることができます。
- N 極で数値が減少し、S 極で数値が増加します。

磁力をグラフで表示する



- センサーにかかる圧力を測ることができます。
- 抑える強さが強くなるほど数値が大きくなり、弱いと数値が小さくなります。

圧力を加えたら LED が光るようにしよう





センシング

音ブロックを使ったプログラミング

音センサーを使用して音の大きさを測るときに必要なプログラムです。

- 音の大きさを測ることができます。
- 大きな音を鳴らすと数値が大きくなり、 静かなときは数値が小さくなります。

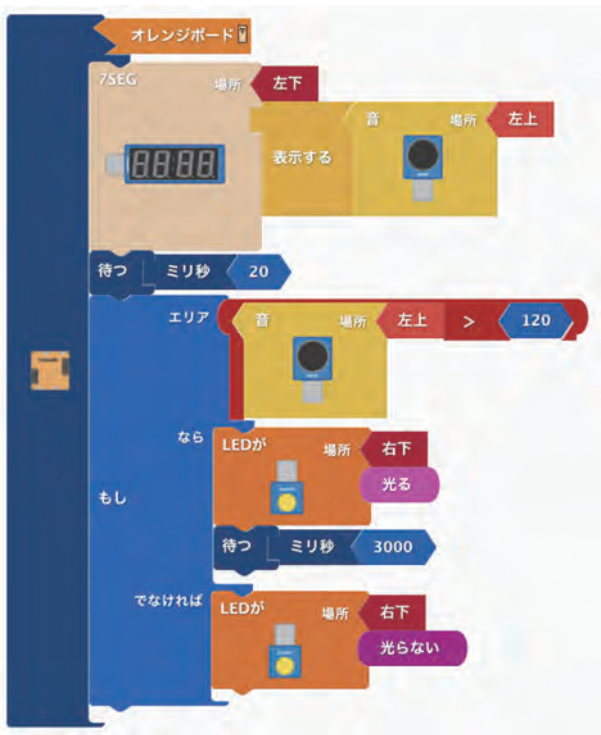
音の大きさをモニターに表示しよう



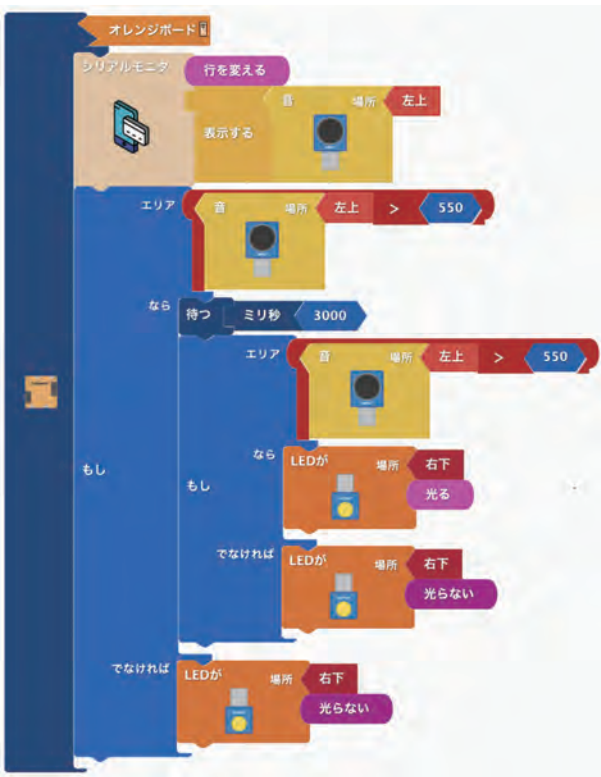
音の大きさをグラフで表示しよう



音の大きさが 120 より大きくなったら LED が光るようにしよう



音が一定の間鳴り続けたら LED が光るようにしよう



ブルーボード

オレンジボード

センシング

曲げセンサーブロックを使ったプログラミング

曲げセンサーを使用してセンサーの曲がり具合を測るときに必要なプログラムです。

- センサーの曲がり具合を測ることができます。
- 大きく曲がれば曲がるほど数値が大きくなります。

どれだけ曲げたかを数値で表示しよう

下のプログラムで、 曲げセンサーが取得した値をシリアルモニタで表示することができます。



曲げすぎると音が鳴るようにしよう

曲げセンサーが 800 より大きい値を取得したときに音コネクタから音が出るプログラムです。 また、 比較する数をほかの数に変えて試してみましょう。



曲げ具合によって、 鳴る音の高さが変わるようにしよう

曲げセンサーが取得する値によって音の高さが変化するプログラムです。
以下のプログラムでは、 300、 500、 700 の値と比較しており、 大きい値を取得するほど高い音が鳴ります。
それぞれの比較する数を変更して試してみましょう。





センシング

人感センサーブロックを使ったプログラミング

人感センサーを使用して周りに人がいるかどうかを判断するときに必要なプログラムです。

- 人感センサーの周囲に動きがあるかどうかを測ることができます。
- 人感センサーの近くで人が動くと 1、じっとしていると 0 を示します。

人が近くにいたら LED が光るようにしよう

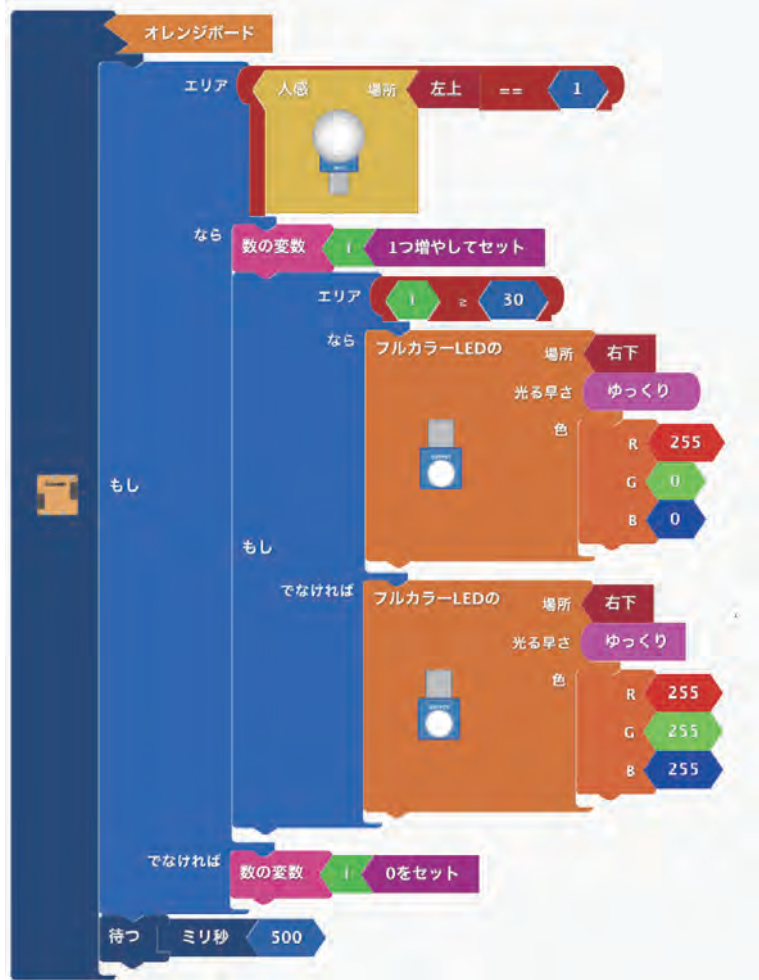
人感センサーは人を感知すると1、感知しなければ 0 を取得するため、人感センサーが 1 のときのみ LED が光るようにします。



人がずっと近くにいると、フルカラー LED の色が変わるようにしよう

下のプログラムでは、人感センサーが 1 を取得したときにフルカラー LED が光ります。また、ずっとブロックでプログラムを繰り返すたびに変数 i を 1 ずつ増加させており、i が 30 以上となったときフルカラー LED が赤く光ります。

変数 i と比較する数を調整することで赤く光るまでの時間を調整することができます。



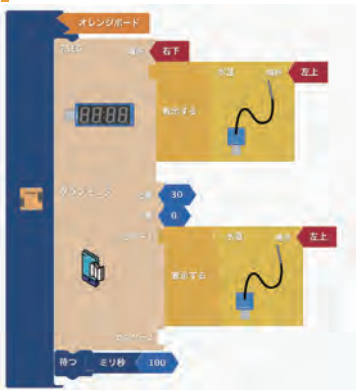
センシング

水温センサーブロックを使ったプログラミング

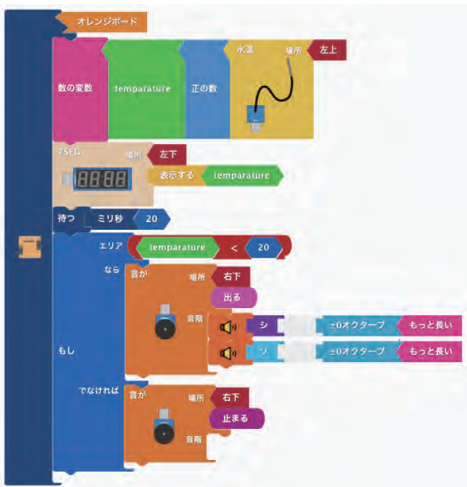
水温センサーを使用して水温を測るときに必要なプログラムです。

- 水の中に水温センサーを入れると水温を測ることができます。
- 水温が下がるとセンサーの数値も小さくなります。

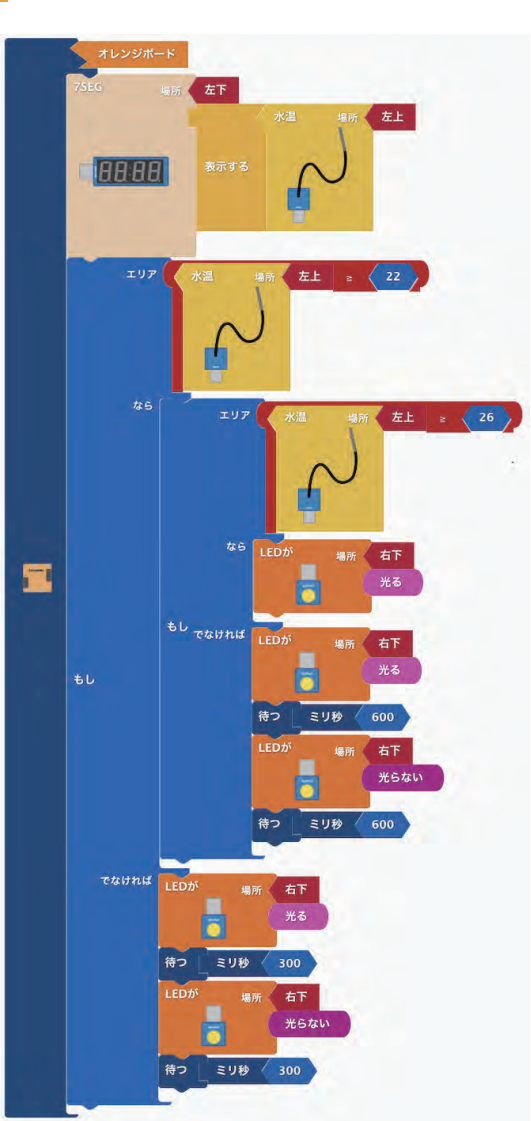
水温を 7SEG に表示し、さらにグラフモニタで表示する



水温が 20 度以下になったら音が鳴る



水温の違いによって、光り方を変える (今回の場合は、22 度以上か、26 度以上か)



センシング

ブルーボード

オレンジボード

土壌水分センサーブロックを使ったプログラミング

土壌水分センサーを使用して土の中の水分量を測るときに必要なプログラムです。

- 土壌水分センサーを土に刺して、土の水分量を測ることができます。
- 水分量が多ければ多いほど、土壌水分センサーが示す数値が大きくなります。

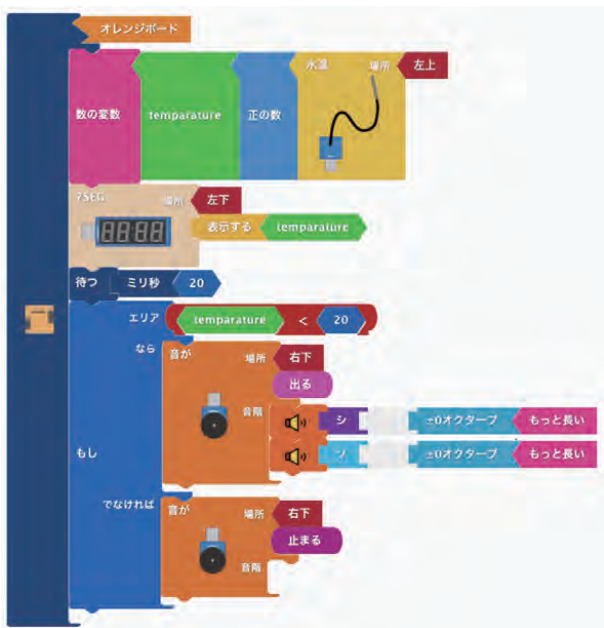
土の中の水分量をモニターに測定する



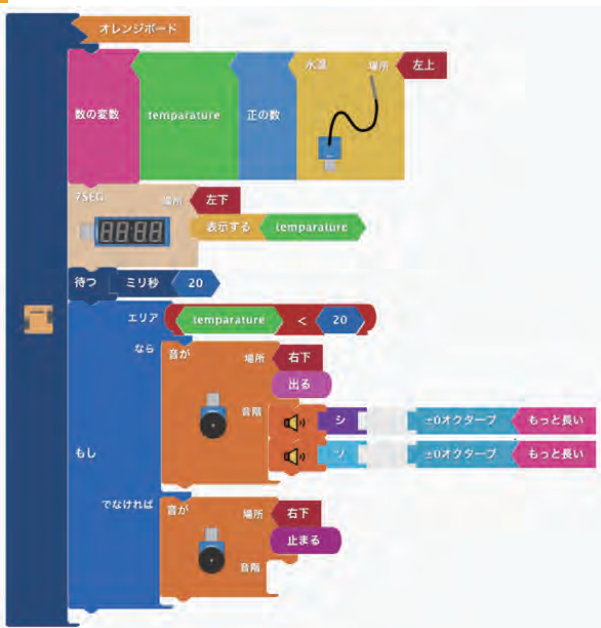
土の中の水分量をグラフに測定する



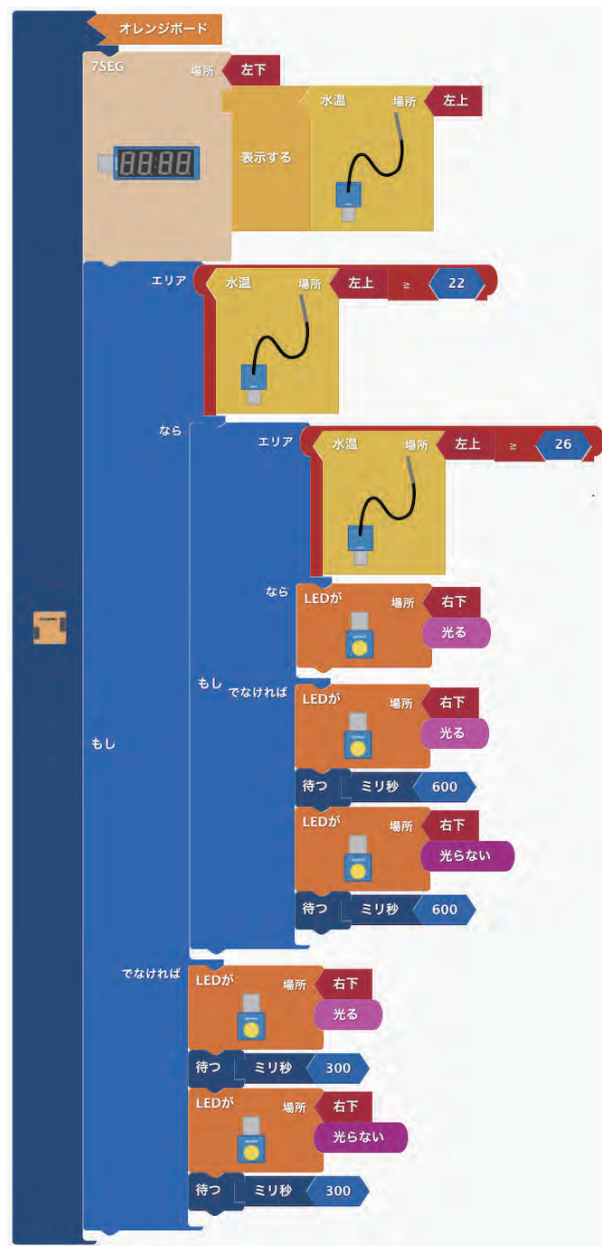
土の中の水分量を測定する



**土の中の水分量が 7SEG に表示して、
水分量が少なくなったら音が鳴る**



土の中の水分量によってLEDの光りかたを変える
(55%以上 ⇒ 点灯 50 ~ 54% ⇒ ゆっくり点滅 49%以下 ⇒ はやい点滅)



気圧センサーブロックを使ったプログラミング

気圧センサーを使用して周囲の気圧を測るときに必要なプログラムです。

- 気圧センサーのある場所の気圧を測ることができます。
- 気圧が大きいほど気圧センサーが示す数値も大きくなります。

気圧を測定しよう



気圧が下がったら音が鳴るようにしよう



気圧が上がったら一定の間 LED が消えるようにしよう

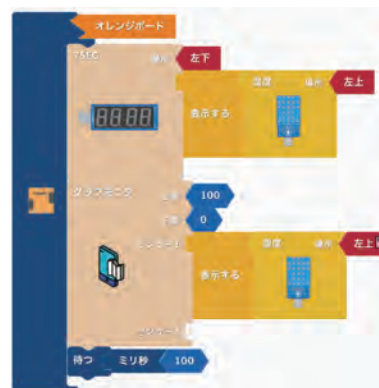


湿度センサーブロックを使ったプログラミング

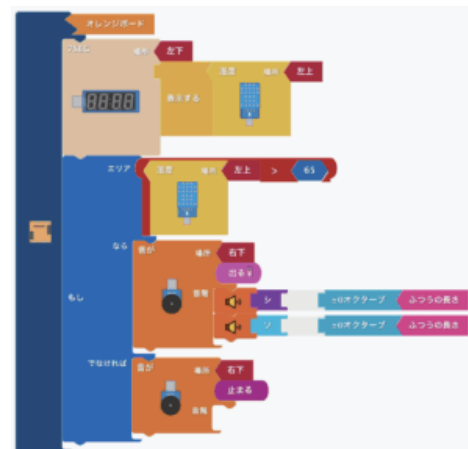
湿度センサーを使用して湿度を測るときに必要なプログラムです。

- 湿度センサーの周囲の湿度を測ることができます。
- 湿度が高くなると湿度センサーの数値も大きくなります。

湿度を測定しよう



湿度が上がったらアラートを鳴らそう



湿度によって LED の光り方を変えよう



雨センサーブロックを使ったプログラミング

雨センサーを使用して雨が降ったことを判断するときに必要なプログラムです。

- 雨センサーに雨の水滴が当たったときに表面の水を検知して値を取得します。
- 雨センサーを使用するときは、 オレンジボードは雨にあたらないようにしてください。

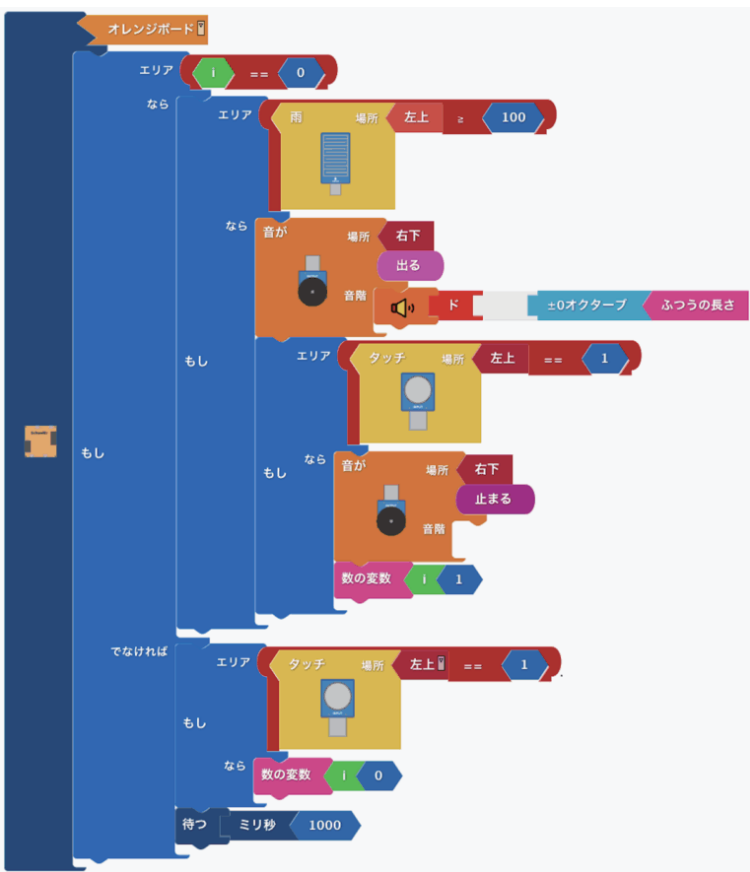
雨が降ったかどうかを シリアルモニタで確認しよう

雨センサーは雨が当たったときに値を取得します。
雨センサーを屋外に設置して、 雨が降っているときに
値をシリアルモニタで確認してみましょう。



雨が降っていることを音でおしえるプログラムを作ろう

雨センサーは雨が当たったときに値を取得します。 雨センサーを屋外に設置して、
雨が降っているときに値をシリアルモニタで確認してみましょう。



アルコールセンサーブロックを使ったプログラミング

アルコールセンサーを使用してアルコール濃度を測るときに必要なプログラムです。

- アルコールセンサーでアルコールの呼気中濃度、 血中濃度を測ることができます。
- 息に含まれるアルコールの濃度によって音が変わるセンサーをつくることができます。

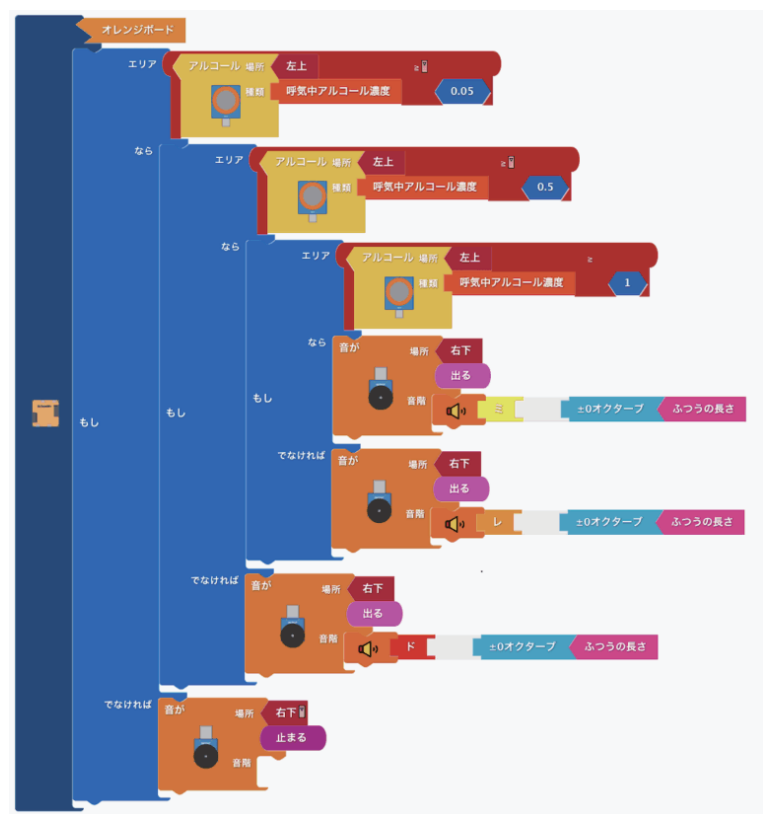
息にアルコールが含まれていないか確認しよう

アルコールセンサーが取得した値をシリアルモニタに
表示するプログラムです。また、 下のプログラムで
はアルコールブロックの種類にアナログ実測値を選
択しています。



呼気中アルコール濃度によって鳴る音の高さを変えよう

呼気中アルコール濃度の大きさによって音コネクターから鳴る音を変えるプログラムです。
アルコールセンサーブロックと比較している数を変更するなどして、 オリジナルのプログラムを
作成してみましょう。



アナログブロックとデジタルブロック

センサーのアナログ値やデジタル値を表示するときに必要なプログラムです。

- アナログブロックは、 センサーの値を 0~1023 までのアナログ値で表示します。
- デジタルブロックは、 アナログブロックの値を 0 と 1 だけのデジタル値で表示します。

タッチセンサーがどんな値を取得しているか見てみよう

STEP.1 タッチセンサーをアナログブロックで見よう

タッチセンサーをアナログブロックで見よう。
このプログラムは、アナログブロックとシリアルモニタブロックだけを使います。
これは、 どのセンサーのアナログ値を見るときも同じプログラムです。



タッチセンサーに触ってみると、0 が 1023 の値がシリアルモニタに表示されます。

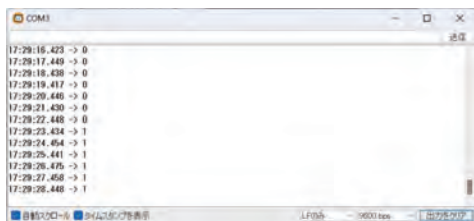


STEP.2 タッチセンサーをデジタルブロックで見よう

今度は、 デジタルブロックでタッチセンサーを見よう。



タッチセンサーに触ってみると 0 と 1 の値がシリアルモニタに表示されます。



このように、 センサーの値をアナログ値やデジタル値で表示することができます。
ほかのセンサーでは違った数字が見られるかもしれません。 いろいろなセンサーの値を見てみましょう！

ロードセルセンサーを使ったプログラミング

重さセンサーを使用してセンサーに乗っている物の重さを測るときに必要なプログラムです。

- 重さセンサーで上に乗っている物の重さを測ることができます。
- 重さセンサーの上に乗っている物の重さを g で計測します。



重さを量りたいモノを、
茶色い丸の土台部分に載せてください。



重さセンサーでは、 茶色い土台の真ん中にある金属がひずむことによって、 重さを量っています。
なお、 金属に書かれている重さを超えるモノは載せないように注意してください。
また、 プログラムを実行した直後は重さを量るための計算をしているので、
シリアルモニタ等に計測値が表示されるまでは、 モノを載せないでください。

モノの重さを量ろう

重さセンサーが取得した値をシリアルモニタに表示するプログラムです。
下のプログラムでは、 単位の種類として g(グラム) を選択しています。



モノが重たすぎると音が鳴るようにしよう

重さセンサーが 1000g(=1kg) より大きい値を取得したときに、
音コネクタから音がでるプログラムです。



スイッチブロックを使ったプログラミング

スイッチを使用してスイッチが押されているかどうかを判断するときに必要なプログラムです。

- スイッチを押したか押していないかを測ることができます。
- スイッチを押すと 1、押されていないと 0 を示します。

スイッチを押したとき、押していないときの様子をモニターに表示する



スイッチを押したとき、押していないときの様子をグラフで表示する



おす前は 0 が 7SEG に表示され、おされた後にそれを感知して値が 1 になることを確認する

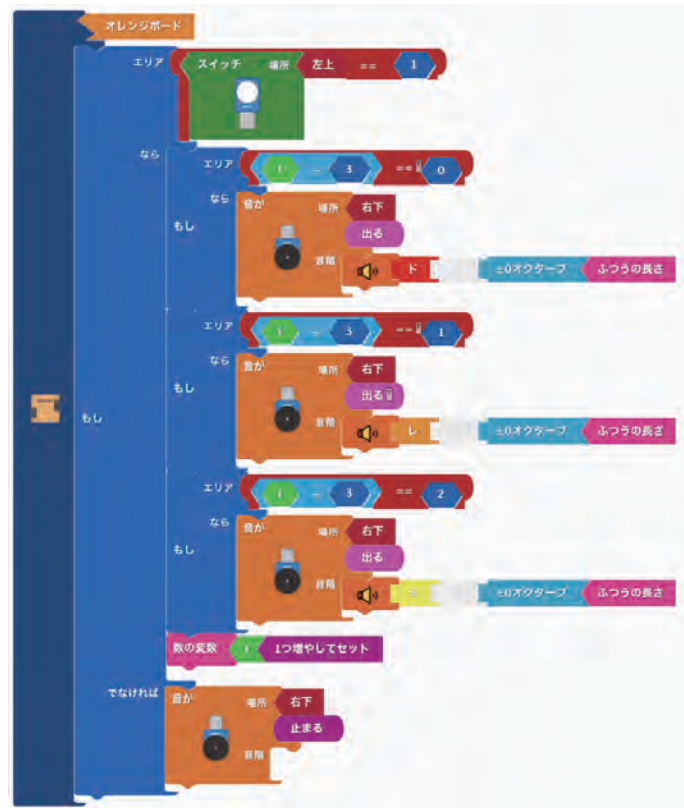


スイッチを押すと音が出る



スイッチを押す回数によって鳴る音を変化させる

下のプログラムでは、スイッチを押した回数によって異なる高さの音が鳴ります。i ブロックは初期値が 0 であり、スイッチを押すと 1 つ増やしてセットされます。



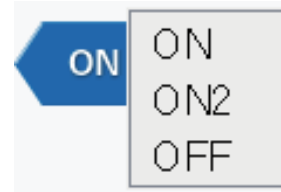
トグルスイッチブロックを使ったプログラミング

トグルスイッチを使用してオンオフ、 オン 2 を判断するときに必要なプログラムです。

- トグルスイッチでオンオフを測ることができます。
- オンにはオン 1、 オン 2 があり、それぞれでプログラムできます。

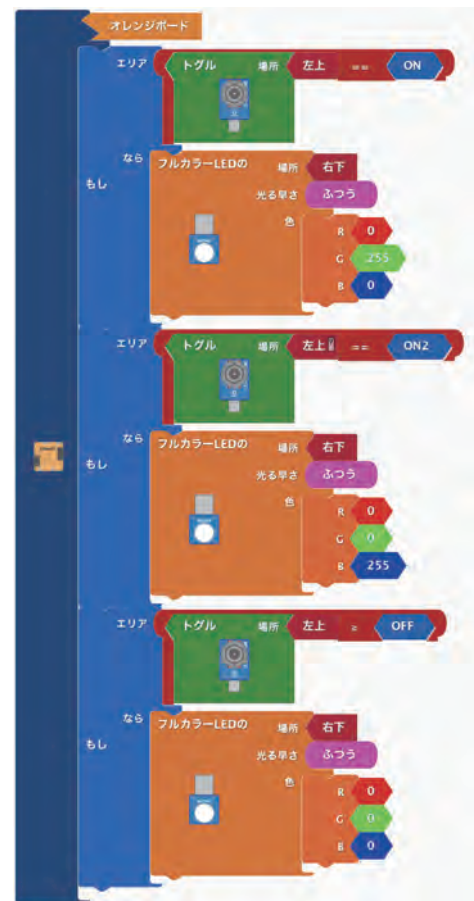
スイッチ状態ブロックの使い方

エリアブロック・もしならブロックと組み合わせることで、スイッチの状態を指定する事が出来ます。



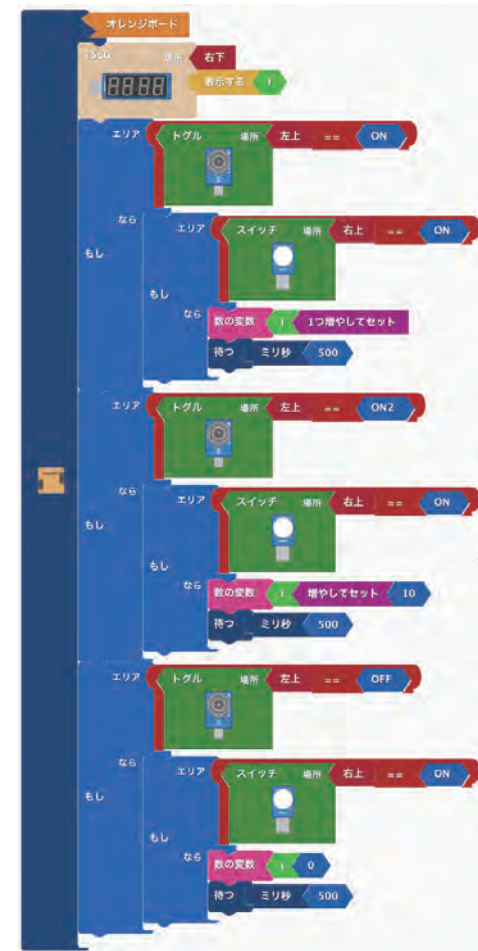
ON にすると緑に光り、 ON2 にすると青く光る

トグルスイッチを ON に切り替えると緑色に、 ON2 に切り替えると青く光り、 OFF にすると消えます。



ON にしてスイッチを押すと 1 増えて、 ON2 にしてスイッチを押すと 10 増えるカウンター

下のプログラムでは、スイッチを押すと 7seg に表示されている値が増えていくカウンターです。トグルスイッチを ON に切り替えてスイッチを押すとスイッチを押すと 1 増え、 ON2 に切り替えてスイッチを押すと 10 増えていきます。OFF に切り替えてスイッチを押すと 0 にリセットされます。



振動場所 左上

振動スイッチを使用して振動しているかどうかを判断するときに必要なプログラムです。

- ## スクーミーボードを振動させると LED が光るプログラムを作ろう



The script is designed to create a color wheel animation. It starts with a 'when green flag clicked' event, followed by a 'say Hello for 2 sec.' speech bubble. A 'when green flag clicked' event is also present at the bottom of the script. The main loop begins with a 'when green flag clicked' event, followed by a 'say Hello for 2 sec.' speech bubble. The loop then branches into three 'if' conditions based on the 'color' variable (0, 1, 2). Each condition sets the 'color' variable to a specific value (0, 1, or 2) and then sets the 'color' variable to a specific value (0, 1, or 2). The 'color' variable is then used to determine the 'color' of the 'color wheel' (0, 1, or 2). The 'color' variable is then used to determine the 'color' of the 'color wheel' (0, 1, or 2). The 'color' variable is then used to determine the 'color' of the 'color wheel' (0, 1, or 2).

Figure 1 shows a sequence of operations in the 'Robot Factory' game. The sequence starts with a '750C' block, followed by a '表示する' (Show) block, then a '移動' (Move) block to the '左上' (Top Left) position. This is followed by an 'エリサ' (Erika) block, then a '移動' (Move) block to the '左上' (Top Left) position, and finally a 'なら' (If) block. The 'なら' block has a '750C' block, a '表示する' (Show) block, and a '移動' (Move) block to the '左上' (Top Left) position. The sequence ends with a '待つ' (Wait) block for 1000 milliseconds.

```

when green flag clicked
  say Hello for 2 sec
  set Count to 0
  repeat (7) times
    move Count up by 1
    if Count = 7
      say Hello for 2 sec
    else
      say Goodbye for 2 sec
  wait 1000 ms
  
```

```

when green flag clicked
  set 7SEG to 0
  set location to bottom-left
  show 7SEG
  set count to 1

  loop
    if area is 1
      set count to count + 1
      if count is 1
        set 7SEG to 1
        show 7SEG
        wait 50 ms
      else
        set 7SEG to 0
        show 7SEG
        wait 50 ms
    else
      set count to count - 1
      if count is 0
        set 7SEG to 0
        show 7SEG
        wait 50 ms
      else
        set 7SEG to 1
        show 7SEG
        wait 50 ms
  end loop
  set count to 0

```

when green flag clicked

7SEG 場所 bottom-left

show 7SEG

count 1

loop

if area is 1

count count + 1

if count is 1

7SEG 1

show 7SEG

wait 50 ms

else

7SEG 0

show 7SEG

wait 50 ms

else

count count - 1

if count is 0

7SEG 0

show 7SEG

wait 50 ms

else

7SEG 1

show 7SEG

wait 50 ms

end loop

count 0

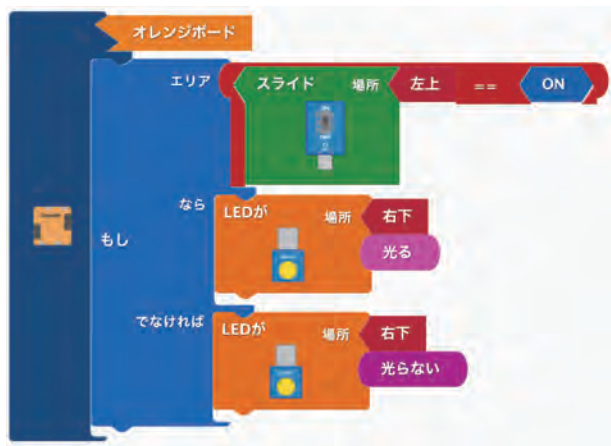
スライドスイッチブロックを使ったプログラミング

スライドスイッチを使用してオンオフを判断するときに必要なプログラムです。

- スライドスイッチでスイッチのオンオフを測ることができます。
- スwitchをオンにして LED を光らせることができます。

スライドスイッチが ON の時だけ LED が光り、 OFF になると消える

スライドスイッチを ON に切り替えると LED が光り続けます。



スライドスイッチが ON の時の秒数の合計を 7SEG に表示する

スライドスイッチを ON に切り替えると、7SEG の表示が 1 秒ごとに 1 ずつ増えていきます。



プッシュスイッチブロックを使ったプログラミング

プッシュスイッチを使用してプッシュスイッチが押されているかどうかを判断するときに必要なプログラムです。

- プッシュスイッチでスイッチが押されているか押されていないかを測ることができます。
- プッシュスイッチが押されていれば 1、押されていなければ 0 を示します。

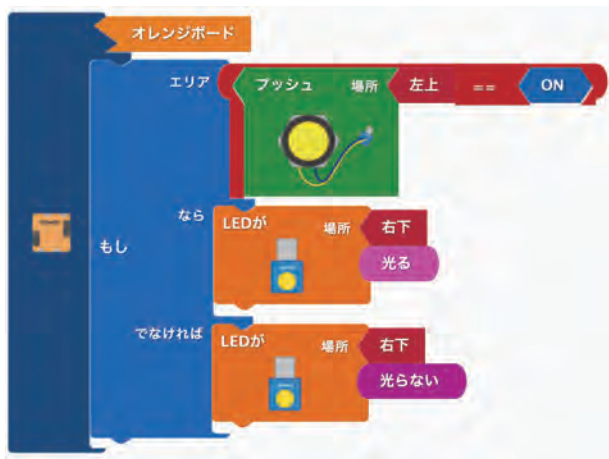
シリアルモニタを見てみよう

スイッチを押すことでシリアルモニタの値が0から1に変わります。1の状態が ON、0の状態が OFF になります。



プッシュスイッチを押すと LED が光り、 はなすと消える

下のプログラムは、 プッシュスイッチを押すと LED が光り、 はなすと消えます。ON ブロックは、 ON は1、 OFF は0という値を表しており、押すと ON＝1状態になるので LED が光ります。スイッチをはなすと OFF＝0となるので LED が消えます。



プッシュスイッチを押すと7SEG のカウントが減り、 0になると LED が光る

下のプログラムでは、 プッシュスイッチを押すと 1 秒ごとに 7SEG の数字が1ずつ減り、 0になると LED が光ります。



フェーダースイッチを使ったプログラミング

フェーダースイッチを使用してフェーダーの位置によって数字を出力するときに必要なプログラムです。

- 最小値と最大値を指定することで、 フェーダーの位置に応じてその範囲の値を指定することができます。
- つまみが下の方だと最小値、 上の方だと最大値を出力します。

フェーダースイッチとは？

フェーダースイッチとはツマミ式のスイッチで、 フェーダーの位置に応じて値を出力します。

フェーダースイッチブロックの使い方

最小値と最大値を指定することで、 フェーダーの位置に応じてその範囲の値を出力します。 ブロックを取り出した直後は、 最小値が 0、 最大値が 100 になっています。 フェーダーの位置がスクーミ-ボード側なのが「最小値」、その反対側が「最大値」です。 なお、 最小値とは一番小さい値のことで、 最大値とは一番大きい値のことです。



シリアルモニタに数値を出力させよう

下のプログラムでは、 指定した範囲の値が、 フェーダーを動かした位置に応じてシリアルモニタに出力されます。 このプログラムの場合、 最小値を 0、 最大値を 100 に指定しているため、 フェーダーの位置に応じて 0~100 の値がシリアルモニタに出力されます。



フェーダーを動かして、フルカラー LED の色を変化させよう

下のプログラムでは、 フェーダーをスライドするとフルカラー LED の色が変化します。 最小値が 0、 最大値が 255 になっているので、 フェーダーの位置に応じてフルカラー LED の色の R 要素が 0~255 の値になります。 このプログラムの場合、 フルカラー LED の色は青から紫に変化します。



フェーダーを動かして、音を変化させよう

下のプログラムでは、 フェーダーを動かすと音が変わります。 フェーダーの位置が中心より左（1~50）の場合はドの音が、 中心より右（51~100）の場合はラの音が鳴ります。



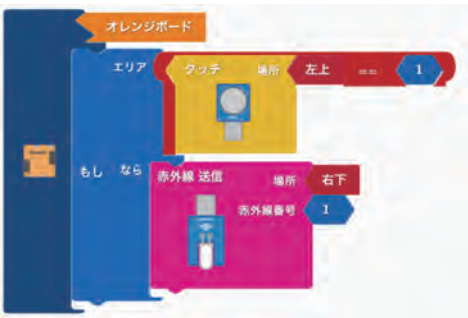
赤外線送信 ・ 受信ブロックを使ったプログラミング

赤外線送信 ・ 受信センサーを使用して 2 つのボード間の通信をするときに必要なプログラムです。

- 赤外線の送信 ・ 受信センサーを使って、 2 つのボード間で通信ができます。
- 赤外線を受信したら、 LED が光るというようなプログラムができます。

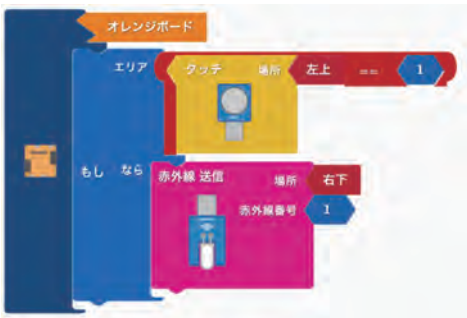
タッチセンサーに触れると、 別のオレンジボードに接続している LED が点滅するようにしよう

赤外線送信側のオレンジボードにアップロードするプログラム



タッチセンサーに触れることで、
赤外線送信ブロックが実行されます。
赤外線番号が1であるため、 赤外線受信側の赤外線
受信ブロックの赤外線番号は1にする必要があります。

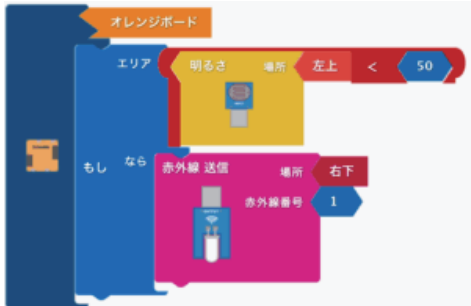
赤外線受信側のオレンジボードにアップロードするプログラム



赤外線を受信すると LED が点滅するプログラムです。
赤外線受信ブロックは、 赤外線番号がおなじ赤外線送信ブロックからの
赤外線を受け取ることができます。
また、 赤外線を受信したときのみ、 赤外線受信ブロック内のプログラムが実行されます。

ある部屋に電気がついているかによって、 リビングにあるフルカラー LED の色が変わるようにしよう

赤外線送信側のオレンジボードにアップロードするプログラム



明るさセンサーが 50 より小さい値を受け取ったときのみ、
赤外線を送信します。

赤外線受信側のオレンジボードにアップロードするプログラム



赤外線を受信していないときはフルカラー LED が青色に、
受信しているときはフルカラー LED が赤色に光るプログラムです。
変数ブロックを使用することによって、 下のプログラムのように受信している場合と
していない場合で実行するプログラム内容を変えることができます。

SD カードブロックを使ったプログラミング

SD カードにセンサーの値などを書き込むときに必要なプログラムです。

- センサーから受け取った値を SD カードに書き込むことができます。
- 書き込んだファイルを消去することができます。

SD カードに、 変化する明るさを記録しよう

明るさコネクタから取得する値を、 SD カードに書き込むプログラムです。



SD カードに、 回数のデータを記録しよう

変数をつけることによって、 センサーの値以外にもデータを記録することができます。



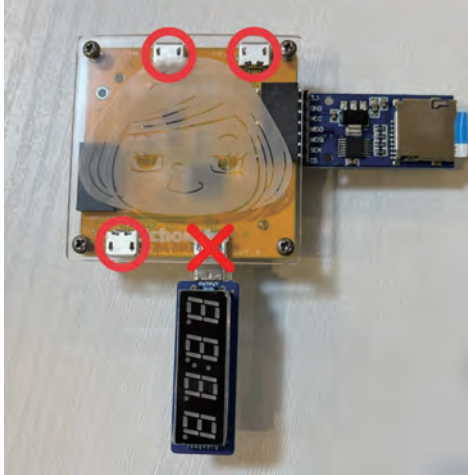
確認したファイルを削除しよう

SD カードからファイルを削除するプログラムです。 下のプログラムでは、
「1」と名前の付いたファイルを削除します。



7SEG と SD カードを一緒に使う場合の注意点

SD カードと 7SEG を一緒に使う場合は、ハードウェアの制約があるため、
7SEG は右下以外で使ってください。





GPS

ブルーボード

オレンジボード

GPS センサーブロックを使ったプログラミング

GPS センサーを使用して緯度と経度を測るときに必要なプログラムです。

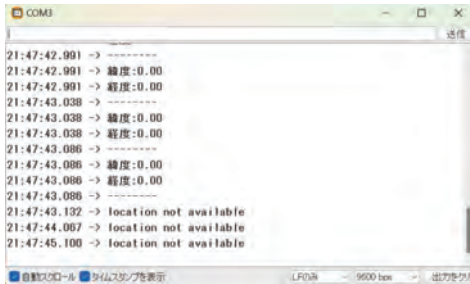
- GPS センサーで緯度と経度を測ることができます。
- GPS が取得されていない場合、 シリアルモニタ 『location not available』と表示されます。

GPS センサーでスクーミ-ボードの緯度、 経度を取得しよう

GPS センサーは屋根のない屋外や、 パルコニーやベランダなど遮るものがない見通しの良いところで使用ください。

GPS センサーが正しく緯度、 経度を取得するには 3 ～ 5 分近く時間がかかります。

また GPS が取得されていない場合、 シリアルモニタには以下のように 『location not available』と表示されます。



現在位置を緯度、 経度で表示しよう

GPS センサーを使用することで緯度、 経度を取得することができます。

下のプログラムでは緯度は変数 lat 、 経度は変数 lng に代入され、 変数はシリアルモニタで表示しています。



GPS

ブルーボード

オレンジボード

2 点間距離ブロックを使ったプログラミング

2 地点の緯度、 経度から 2 点間の距離を測るときに必要なプログラムです。

- 2 地点の緯度、 経度から 2 点間の距離を取得することができます。
- 緯度、経度は Degree 形式 (123.456...のような数字のみの表記) で入力します。

特定の 2 か所の緯度と経度からその距離を取得しよう

下のプログラムはスカイツリーと甲府駅の距離をキロメートル (km) 単位で表示します。

変数 KofuStation_lat に甲府駅の緯度、 変数 KofuStation_lng に甲府駅の経度、 変数 SkyTree_lat にスカイツリーの緯度、 変数 SkyTree_lng にスカイツリーの経度を入れていきます。

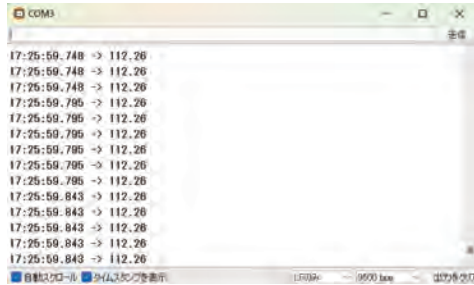
変数に好きな場所の緯度、 経度を入れて、 実際に 2 地点の距離を取得してみましょう。

※緯度と経度は Google map などから取得することができます。

STEP.1 変数に好きな場所の緯度、 経度を入れる



STEP.2 シリアルモニタには 下のように表示される



STEP.3 「キヨリ測」を用いて 距離を測定する

実際に「キヨリ測」を用いて甲府駅とスカイツリーの距離を測定すると、 おなじ 112.26km を取得することができました。



エリア

ブルーボード

オレンジボード

エリアブロックを使ったプログラミング

もしならブロックのエリアを指定するときに必要なプログラムです。

- 常に行われるプログラムではなく、もしならブロックで条件を指定するときに使います。
- 左辺と右辺の数値の比較して条件にすることができます。

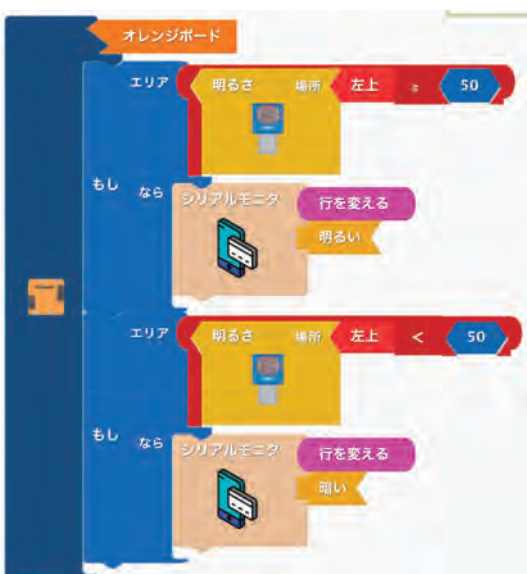
明るさの値が 50 より大きいなら、「明るい」とシリアルモニタに表示しよう



明るさの値が 50 以上（50 をふくむ）なら、「明るい」とシリアルモニタに表示しよう



明るさの値が 50 以上なら、「明るい」とシリアルモニタに表示して、50 より小さいなら、「暗い」とシリアルモニタに表示しよう



エリア

ブルーボード

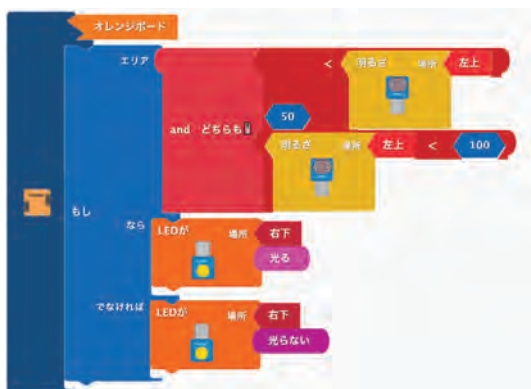
オレンジボード

どちらもブロックの使い方

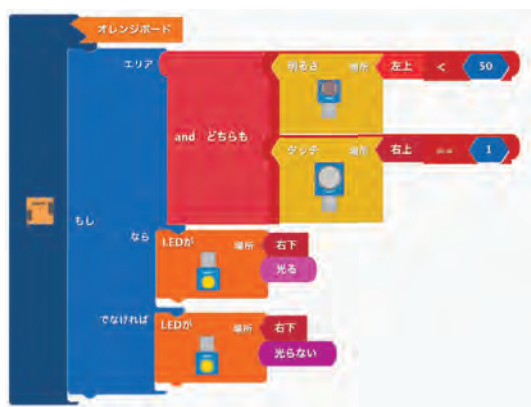
複数のエリアが同時に満たされているときに必要なプログラムです。

- 2 つの条件を両方満たすことを条件として指定することができます。
- 2 つだけでなく、3 つ以上でも条件を指定することができます。

明るさの値が 50 より大きくて 100 より小さいなら、LED が光るようにし、そうでなければ LED が消えるようにしよう



明るさの値が 50 より小さくて、タッチセンサーが 1 なら、LED が光るようにしよう



明るさの値が 50 より大きく 100 より小さくて、タッチセンサーが 1 なら、LED が光るようにしよう





エリア

どちらかブロックを使ったプログラミング

複数のエリアのどちらかが満たされているときに必要なプログラムです。

- 2つの条件のどちらかを満たすことを条件として指定することができます。
- 2つだけでなく、3つ以上でも条件を指定することができます。

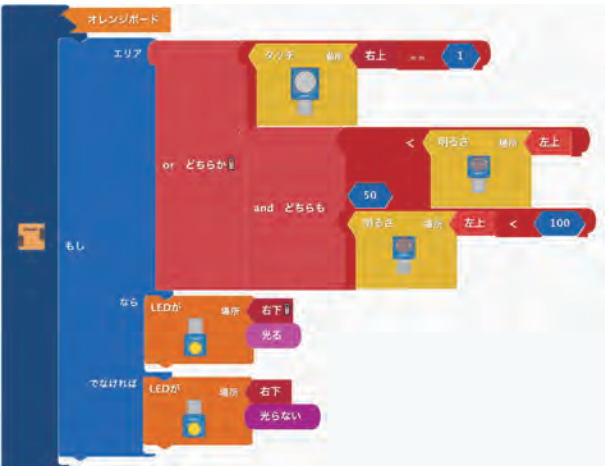
明るさの値が 50 より小さいか、100 より大きいどちらかなら LED が光り、
そうでなければ LED が消えるようにしよう



タッチセンサーが 1 か、明るさの値が 50 より小さいかのどちらかなら、
LED が光り、 そうでなければ LED が消えるようにしよう



タッチセンサーが 1 か、明るさの値が 50 より大きくて 100 より小さいどちらかなら、
LED が光り、 そうでなければ LED が消えるようにしよう



ブルーボード

オレンジボード

エリア

ではないブロックを使ったプログラミング

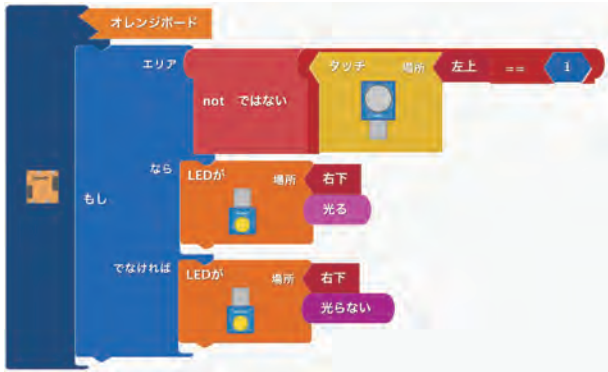
エリアが満たされているかどうかを逆転させるときに必要なプログラムです。

not ではない

- 条件の否定を条件として指定することができます。
- どちらもブロックやどちらかブロックの否定を条件にすることができます。

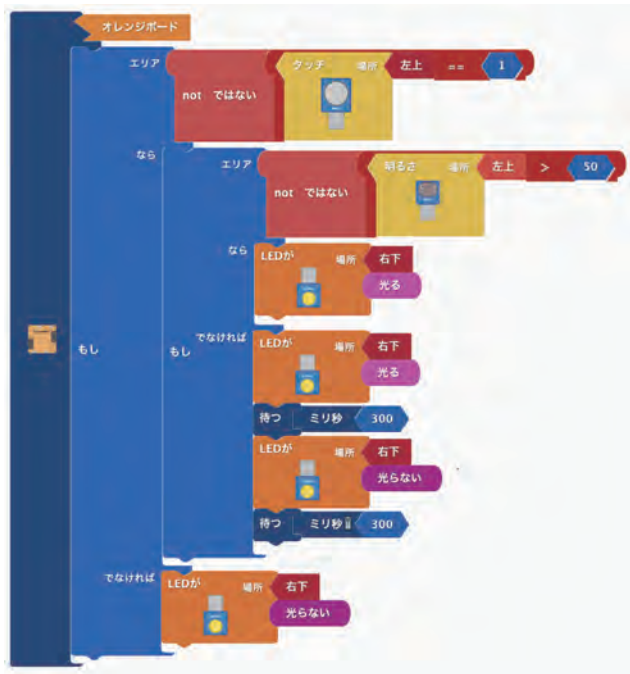
ではないブロックを使ったとき、
タッチセンサーに触ったときの LEDの光り方を確認しよう

ではないブロックをもしならでなければブロックのエリアで使用することで、 エリアが成立していない場合に LED が光ります。



ではないブロックを使ったプログラムがどんな動作
をするか確認しよう

下のプログラムではではないブロックを 2 つ使用しています。
タッチセンサーに触れず明るさコネクタが 50 以下の値を取得するときに LED が光り、
タッチセンサーに触れず明るさコネクタが 50 より大きい値を取得するときは LED が点滅し、
タッチセンサーに触れたとき、 LED は光らなくなります。



数の変数ブロックを使ったプログラミング

変数を使用するときに必要なプログラムです。

- 変数を宣言することができます。
- 数ブロックを使い、変数の値を変更することができます。

変数 **i** の中に「1031」を入れて、
変数 **i** をパソコンに表示しよう



変数 **i** の中にタッチセンサーの値を入れて、
変数 **i** をパソコンに表示しよう



変数 **i** の中に明るさの値を入れて、
変数 **i** が 10 より小さいなら、
LED が光るようにしよう



数の変数名ブロックを使ったプログラミング

変数名を使用するときに必要なプログラムです。

- 変数名は数値を保存する箱の名前です。変更することができます。
- 変数を様々なブロックで 사용할 ことができます。

数の変数名

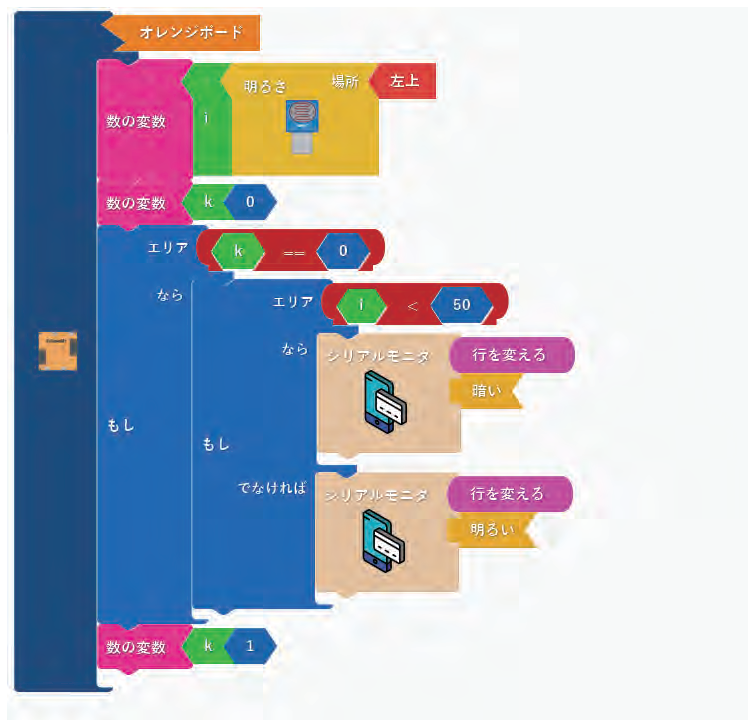
変数 **i** と変数 **k** を使い、変数 **i** が 50、変数 **k** が 100 の時、**i** と **k** を比べて、**k** の方が大きいなら、
LED が光るようにしよう



変数 **i** と変数 **k** を使い、変数 **i** が 50、変数 **k** が明るさセンサーの値にすると、**i** と **k** を比べて、**k** の方が大きいなら、
LED が光るようにしよう



変数 **i** を使い、明るさセンサーに 1 回だけ反応させて、50 より小さくになったら、
「暗い」とパソコンに表示して、そうでないときは「明るい」とパソコンに表示しよう



iブロックを使ったプログラミング

変数 i を使用するときに必要なプログラムです。

- iブロックは、 数の変数名ブロックと同じように数値をセットして使います。
- iブロックは、 数の変数ブロックで代入した数が入ります。

iブロックに数を代入して、 その数だけ LED を点滅させよう

iブロックは、 数の変数ブロックで代入した数が入ります。（変数ブロックの初期値は 0）

下のプログラムでは数の変数ブロックで i に 5 を代入しているため、 5 回指定回数くりかえすブロックの中身をくりかえします。



iブロックに数を複数回代入して、 動きを確認しよう

上のプログラムを 2 つ連結した形式です。

変数 i に 5 を代入してはじめの指定回数くりかえすブロックをぬけた後、 i に 3 を再代入しています。

したがって 2 つ目の指定回数くりかえすブロックは 3 回くりかえされます。



起動からの時間をセットブロックを使ったプログラミング

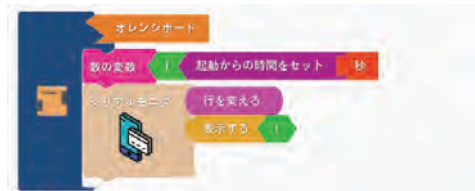
ボードを起動してからの時間を計測するときに必要なプログラムです。

起動からの時間をセット 秒

- スクーミーボードをパソコンに接続したとき、 または電池をいれたときからの時間をセットします。
- 待つブロックなどの一時的にプログラムが見た目上動かない場合にも値を変化させ続けることができます。

スクーミーボードを起動してからの時間をモニターに表示させよう

シリアルモニタに起動からの時間をセットブロックを使用することで、 スクーミーボードを起動してからの時間を使用することができます。



5 秒間隔で音を鳴らそう

数の変数ブロックに対して起動からの時間をセットブロックをセットすることができます。

下のプログラムでは、 起動からの時間を変数 i に代入して、 i が 5 の倍数になるたびに音ブロックから音が出ます。

5 秒間隔のほかにも、 いろいろな秒間隔に変更してみましょう。



データ

ブルーボード

オレンジボード

0 と 1 を入れ替えてセットブロックを使ったプログラミング

変数の値を 0 と 1 を交互に入れ替えるときに必要なプログラムです。

0と1を入れ替えてセット

- 変数の値の 0 と 1 を交互に入れ替えることができます。
- ずっとブロックが動くたびに値が入れ替わります。

変数に 0 と 1 を交互に代入して、LED を点滅させよう

下のプログラムでは、0 と 1 を入れ替えてセットブロックによって変数 i に 0 と 1 が交互に入ります。
ずっと動くブロックをくりかえすたびに i の値が変わるため、1 秒間隔で LED が点滅します。



変数に 0 と 1 を交互に代入してフルカラー LED を 2 色に光らせよう

数の変数ブロックに対して起動からの時間をセットブロックをセットすることができます。
下のプログラムでは、起動からの時間を変数 i に代入して、i が 5 の倍数になるたびに音ブロックから音が出ます。
5 秒間隔のほかにも、いろいろな秒間隔に変更してみましょう。



データ

ブルーボード

オレンジボード

1 つ増やしてセットブロックを使ったプログラミング

変数の値を 1 だけ増やすときに必要なプログラムです。

1つ増やす

- 変数の値を 1 だけ増やすことができます。
- ずっとブロックが動くたびに変数の値が 1 ずつ増えていきます。

変数 i の数値を増やして 7SEG に表示しよう



タッチセンサーが 1 になると、数値が増えていくことを 7SEG に表示しよう



明るさの値が 50 より小さいなら時間（時間の単位は秒）を 7SEG に表示して、50 以上なら、0 を 7SEG に表示しよう



増やしてセットブロックを使ったプログラミング

変数の値を好きな数だけ増やすときに必要なプログラムです。

増やしてセット

1

- 変数の値を好きな数だけ増やすことができます。
- ずっとブロックが動くたびに変数の値が増えていきます。

増やしてセットブロックをいれた変数の値を 7SEG に表示させよう

変数 *i* は初期値として 0 がセットされています。

はじめて増やしてセットブロックに 5 を指定しているため、 はじめて数の変数ブロックを実行したときには *i* に 5 が入り 7SEG には 5 が表示されます。

ふたたび数の変数ブロックを実行したときには *i* には 5 が足されるため、 7SEG には 10 が表示されます。



フルカラー LED が時間経過でいろいろな色に光るようにしよう

変数 *red*、*green*、*blue* はそれぞれ増える数が異なります。これにより、 それぞれの変数を 255 で割ったときの余りは一致せず、 くりかえすごとに比率も異なっていくため、 フルカラー LED の色が変化していきます。



1つ減らしてセットブロックを使ったプログラミング

変数の値を 1 だけ減らすときに必要なプログラムです。

1つ減らしてセット

- 変数の値を 1 だけ減らすことができます。
- ずっとブロックが動くたびに変数の値変数の値が 1 ずつ減っていきます。

一秒ごとに、 数を 1 ずつ減らすプログラムを作ろう

変数 *i* には初期値として 0 がセットされています。 したがって以下のプログラムでは一秒ごとに *i* が

-1, -2, -3, -4, ...

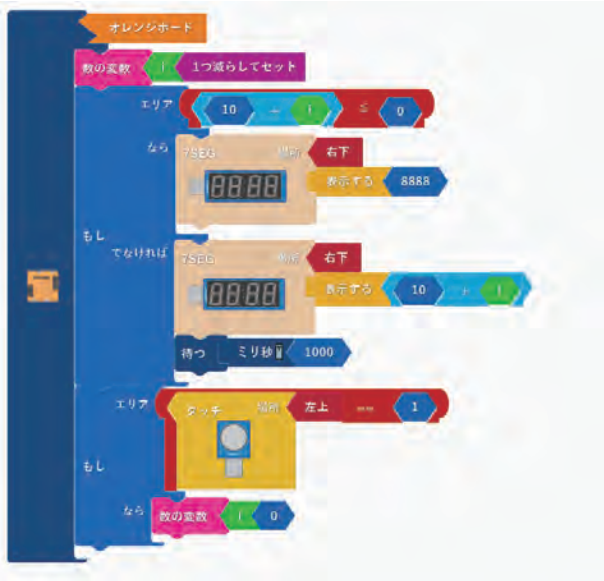
のように変化していきます。



7SEG でカウントダウンしよう

変数 *i* は、 ずっと動くブロックでくりかえすたびに初期値 0 から 1 ずつ小さくなります。 したがって、 もしならでなければブロックのエリアの左辺は 10 から 1 ずつ小さくなります。

i が -10 になると、 もしならでなければブロックのエリアが正しくなるため、 7SEG には「8888」が表示されます。 タッチセンサーに触れると *i* が 0 に初期化されるため、 ふたたびカウントダウンを開始します。



減らしてセットブロックを使ったプログラミング

変数の値を好きな数だけ減らすときに必要なプログラムです。

減らしてセット

1

- 変数の値を好きな数だけ減らすことができます。
- ずっとブロックが動くたびに変数の値が減っていきます。

一秒ごとに、数を5ずつ減らして表示させよう

減らしてセットブロックに5がセットされているため、数の変数ブロックを実行するたびに、iは5ずつ減少していきます。

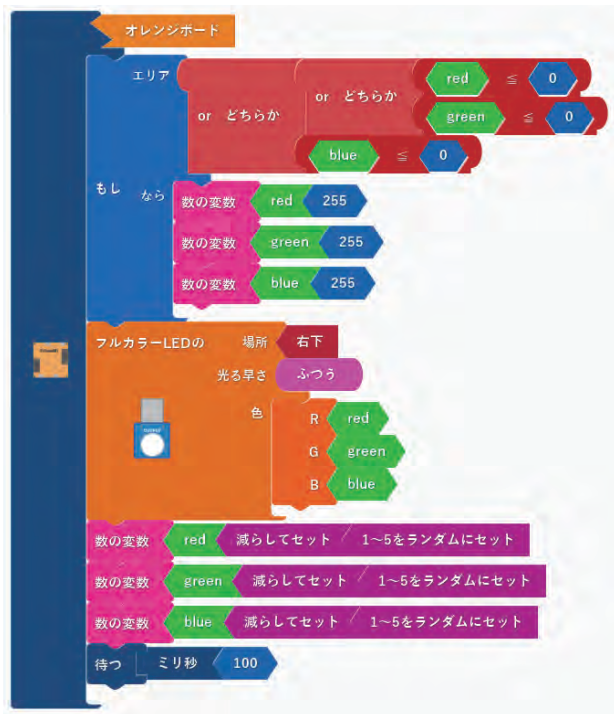


フルカラーLEDをいろいろな色に光らせてみよう

下のプログラムでは、変数 red、green、blue が用意されており、それぞれがフルカラーLEDのRGBに対応しています。

それぞれの変数には255がはいり、それぞれの変数から1～5の数がランダムに減っていきます。ランダムであるためフルカラーLEDの色は変化していきます。

また、いずれかの変数が0より小さくなった場合、それぞれの変数は255に初期化されます。



0をセットブロックを使ったプログラミング

変数の値に0をセットするときに必要なプログラムです。

0をセット

- 変数の値を0にセットすることができます。
- ずっとブロックが動くたびに変数の値に0がセットされていきます。

変数を0にセットしてみよう

数の変数ブロックを使ってiに0をセットしてみましょう。

iが0になっていることはシリアルモニタや7SEGで確認することができます。



変数iの数字を増やしていった、100になったら0にもどることを7SEGに表示しよう

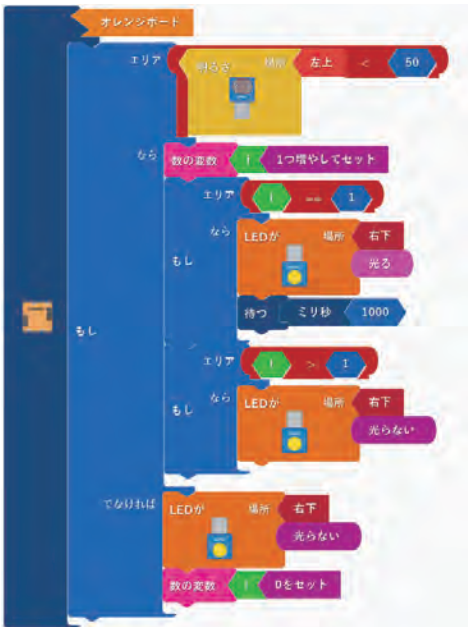


i変数が5をこえたら0にもどすプログラムを作ろう

下のプログラムでは、ずっと動くブロックの中身がくりかえされるたびにiが1ずつ増えていきます。iが5を超えるとiが0になるように、もしならブロックを使用しましょう。



変数iの数字を増やしていった、100になったら0にもどることを7SEGに表示しよう



明るさの値が50より小さいなら、変数iを1ずつ増やしていくことを7SEGに表示しよう



ランダムブロックを使ったプログラミング

変数の値に 1 から 5 をランダムにセットするときに必要なプログラムです。

1～5をランダムにセット

- 変数の値に 1 から 5 をランダムにセットすることができます。
- ずっとブロックが動くたびに変数の値に 1 から 5 がランダムにセットされます。

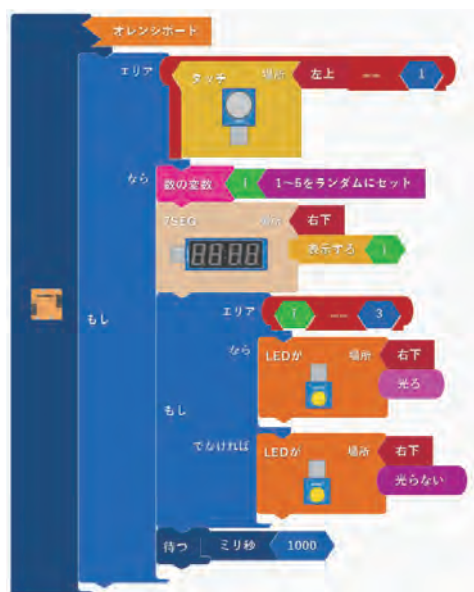
変数 **i** に 1 秒ごとにランダムに数値を入れ、変数 **i** が変化していく様子をグラフで表示しよう



変数 **i** に 1 秒ごとにランダムに数値を入れ、7SEG に表示される変数 **i** が 2 の時だけ LED が光り、変数 **i** が 2 でなければ、LED が消えるようにしよう



タッチセンサーが 1 なら、7SEG に変数 **i** に 1 秒ごとにランダムに数値を表示し、変数 **i** が 3 の時に LED が光るようにして、3 でないときは消えるようにしよう



からまでランダムにセットブロックを使ったプログラミング

2 点間距離ブロックの変数に入れる緯度、経度は Degree 形式 (123.456…のような数字のみの表記) で入力します。

ランダムにセット

- 変数の値に好きな範囲でランダムにセットすることができます。
- ずっとブロックが動くたびに変数の値に好きな範囲でランダムにセットされます。

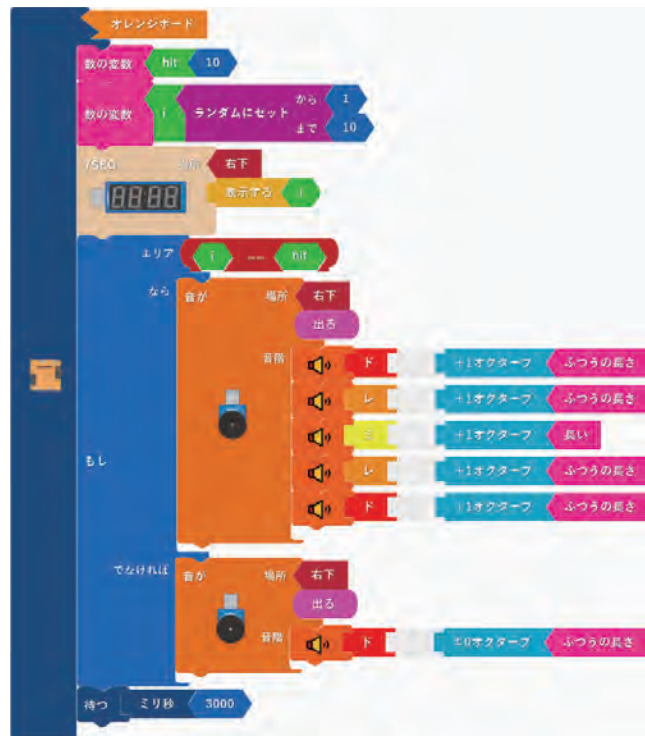
シリアルモニタに 1 から 10 までの数をランダムに表示させよう

ランダムにセットブロックを使用することで、i にランダムの指定範囲の数を代入することができます。



特定の数がセットされたときに、当たりの音を鳴らすプログラムを作ろう

ランダムにセットブロックによって、変数 **i** にランダムな数が代入されます。変数 **hit** に適当な当たりの数をセットしてプログラムを動作させてみましょう。
また、下のプログラムでは待つブロックを使用していますが、タッチセンサーなどのセンシングを用いて好みのタイミングで当たりの音を鳴らすことができます。



小数点ブロックを使ったプログラミング

小数点の桁数を調整するときに必要なプログラムです。

- シリアルモニタに表示する小数点以下の桁数を調整することができます。
- 数だけでなく、 変数やセンサーの値の小数点以下の桁数を調整することができます。

シリアルモニタに 1 から 10 までの数をランダムに表示させよう

STEP.1 センサの値をシリアルモニタに表示する

例として明るさセンサを使用しています。 明るさセンサから取得した値は数の変数 i に入り、 変数 i を下のプログラムのようにセットすることでシリアルモニタに表示されます。

下のプログラムでは、 シリアルモニタには小数第二桁まで表示されています。



STEP.2 表示する桁数を調整する

ここで小数点ブロックを使用することで、 シリアルモニタに表示する小数の桁数を調整して表示することができます。

小数点以下の桁数を 0 にすることで、 シリアルモニタには整数で表示することができます。



上のプログラムでは小数点以下の桁数を 0 にセットしていますが、 これを変更することで表示される小数点以下の桁数を自由に変更することができます。

正の数ブロックを使ったプログラミング

数値の絶対値を用いるときに必要なプログラムです。

- マイナスの数を正の数に変換することができます。
- 数だけでなく、 変数やセンサーの値を絶対値に変換することができます。

マイナスの数を正の数にしてシリアルモニタに表示しよう

正の数ブロックを使用して、 マイナスの数をプラスの数にすることができます。 プラスの数はそのまま表示されます。



変数の数を正の数にしてシリアルモニタに表示しよう

下のプログラムのように変数を正の数字にすることもできます。



大きい方ブロックを使ったプログラミング

2 つの数値の内、 大きいほうを用いるときに必要なプログラムです。

- 2 つの数を比較して、 大きいほうをシリアルモニタに表示することができます。
- 数だけでなく、 変数やセンサーの値を比較することもできます。

2 つの数を比較して、 大きい方の数をシリアルモニタに表示させよう①

大きい方ブロックを使用することで、 2 つの数を比較し大きい方の数を用いることができます。



2 つの数を比較して、 大きい方の数をシリアルモニタに表示させよう②

上のプログラムでは数を比較しましたが、 下のプログラムのように変数を比較することもできます。



小さい方ブロックを使ったプログラミング

2 つの数値の内、 小さいほうを用いるときに必要なプログラムです。

- 2 つの数を比較して、 小さいほうをシリアルモニタに表示することができます。
- 数だけでなく、 変数やセンサーの値を比較することもできます。

2 つの数を比較して、 小さい方の数をシリアルモニタに表示させよう①

小さい方ブロックを使用することで、 2 つの数を比較し小さい方の数を用いることができます。



上のプログラムでは、 小さい方ブロックに 100、 50 の数字が入っているため、 小さい方の 50 がシリアルモニタに表示されます。

2 つの数を比較して、 小さい方の数をシリアルモニタに表示させよう②

上のプログラムでは数を比較しましたが、 下のプログラムのように変数を比較することもできます。



数の範囲ブロックを使ったプログラミング

変数のある範囲内の比率を同じ比率のまま、別の範囲に変更するときに必要なプログラムです。

- 数の範囲ブロックの一番上には変数が入り、上の 2 つの数は変更前の上限値と下限値が入ります。
- 数の範囲ブロックの下 の 2 つの数は、変更後の上限値と下限値が入ります。

計算ブロックを使ったプログラミング

変数や数、センサーの値などを計算するときに必要なプログラムです。

- 変数や数、センサーの値と、複数の四則演算を組み合わせて計算することができます。
- %はわり算のあまりを表します。

10 段階でフルカラー LED の色を変化させよう



下のプログラムでは、変数 number に代入する数を 0 から 9 のいずれかに設定すると、数の範囲ブロックによって変数 red に 0 から 255 の数が代入されます。たとえば、変数 number に 0 をセットすると変数 red には 0 が代入され、変数 number に 9 をセットすると変数 red には 255 が代入されます。変数 number に 0 から 9 の好きな数をセットして、フルカラー LED の色の変化を確認しましょう。

10 ÷ 3 の余りを 7SEG に表示しよう



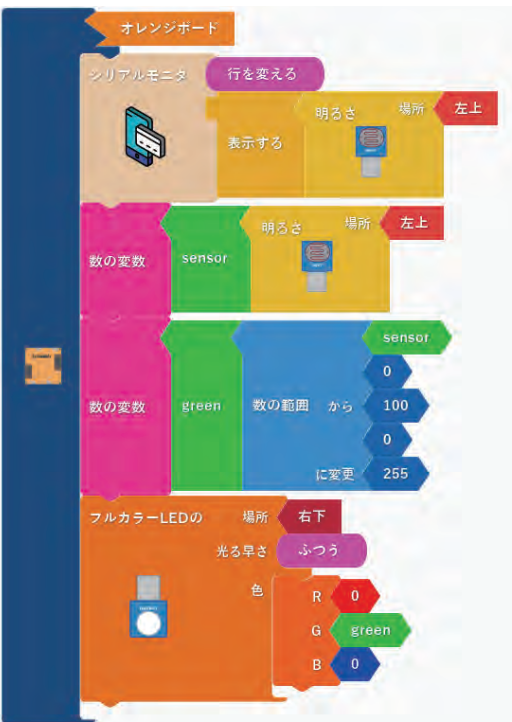
変数 i を 1 秒ごとに 1 つずつ増やしていき、その値を 7SEG に表示して、その値が奇数なのか偶数なのかをパソコンに表示しよう



タッチセンサーが 1 なら LED が光るようにし、1 度はなくても LED が光ったままで、もう 1 度タッチセンサーが 1 になったら LED が消えるようにしよう



明るさによってフルカラー LED の光る色を変えよう



下のプログラムでは、明るさセンサーが取得した値を 0 から 255 の範囲に変更してフルカラー LED の緑色の値（変数 green）に代入しています。明るさセンサーが取得する値をシリアルモニタで確認し、取得した最小値と最大値を数の範囲ブロックにセットしてみましょう。

自作ブロックを使ったプログラミング

自作ブロックを使用するときに必要なプログラムです。

- 「自作ブロック」に作ったプログラムを、「自作ブロックが動く」ブロックで動かすことができます。
- 複数の自作ブロックを使用できます。

自作ブロックで、LED を光らせてみよう

自作ブロックの動作を確認してみましょう。

ずっと動くブロックの中身には LED はありませんが、自作ブロックの中に LED を光らせるプログラムがあるため、実行すると LED が光ります。

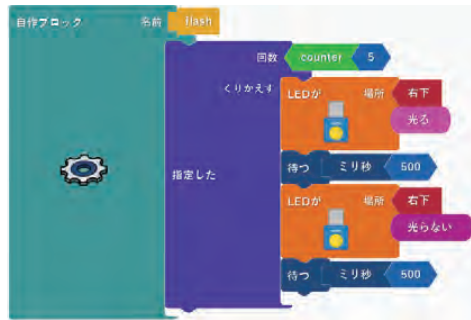
※自作ブロックと自作ブロックが動くブロックの名前を一致させなければ動作しません。



タッチセンサーを触ると LED が点滅するプログラムを作ろう

プログラムのなかに自作ブロックを組み込んだプログラムです。

以下のプログラムでは、タッチセンサーに触れると自作ブロック「flash」が実行されるため、LED が 5 回点滅します。



LED の点滅を、タッチセンサーに触れたときだけはやくしよう

自作ブロックは名前を変更することで、複数作成することができます。役割ごとにまとめると、見やすくまとめたプログラムを作成することができます。

下のプログラムでは、タッチセンサーに触れていないときには自作ブロック slow が実行され、LED が 1 秒間隔で点滅します。タッチセンサーに触れたときは自作ブロック quick が実行され、LED が 0.3 秒間隔で点滅します。



自分でもいろいろな自作ブロックを作成して組み合わせてみましょう。