

# PIECEプログラミングモジュール

## 操作説明書



PIECEプログラミング  
モジュール ZZ-05



## ようこそ、プログラミングの世界へ

私たちの身の回りにはエアコンや炊飯器、スマートフォンや自動販売機や信号機など、便利な機械や大事な機械の多くには、センサーやボタンなどの入力装置、モーターやLEDなどの出力装置と、コンピューターが使われています。コンピューターは入力の情報をもとに出力装置を制御して動かしています。

コンピューターは超高速で動いて疲れませんが、コンピューターはプログラムによって命令された通りしか動けません。機械に希望する動きをやってもらいたいときは、その動作を達成するまでの手順を細かく伝えなければいけません。

この手順を考えて、コンピューターに分かる言葉で伝えることがプログラミングです。

プログラミングで、センサーやLEDを自由に制御できるようになると、もっと便利な装置が作れたり、もっと安全に動かせる機械が作れるようになります。

PIECEプログラミングモジュールの中にも小さなコンピューターが使われていて、プログラムで動かすことができます。

PIECEでプログラミングを体験して自分だけの装置を発明してみましょう！

# 1. プログラミング・ポータル

## 👉 プログラミングの準備



ブラウザには Google Chrome または Microsoft Edge を使います。

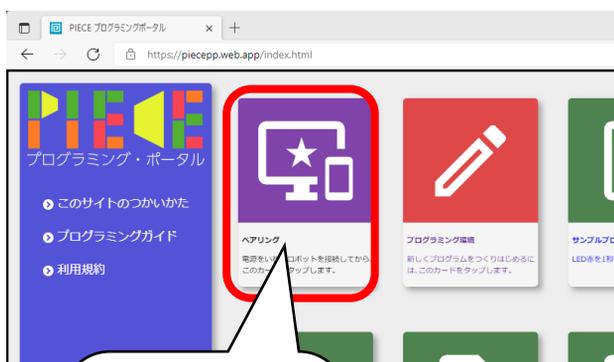


## PIECEプログラミングポータルにアクセスする

- 1 ブラウザーのアドレスバーに下記URLを入力してサイトに移動します。

**piecepp.web.app**

- 2 PIECEプログラミングポータルのページが開きます。



➡ ペアリングのカードがあるときは、  
3 へ進みます。

➡ ペアリングのカードがないときは、  
4 へ進みます

# 👉 プログラミングの準備

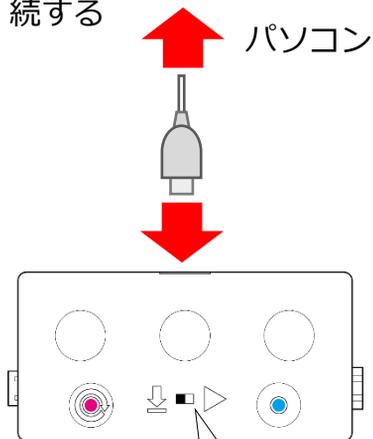
## ③ ペアリングする



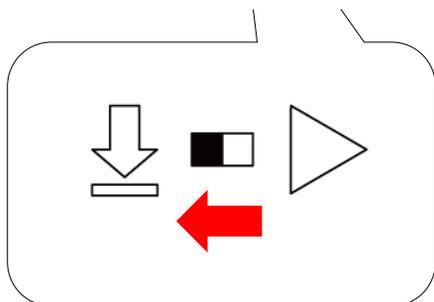
PIECEプログラミングモジュールを始めてパソコンにつないだときは「ペアリング」をします。ペアリングは最初に一回だけで、次に使うときは必要ありません。

※使用するブラウザーが変わったときや、ブラウザーを使用するログインユーザーが変わったときにはペアリングが必要ことがあります。

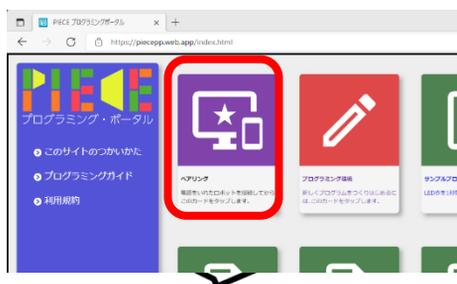
- ① プログラミングモジュールとパソコンをUSBケーブルで接続する



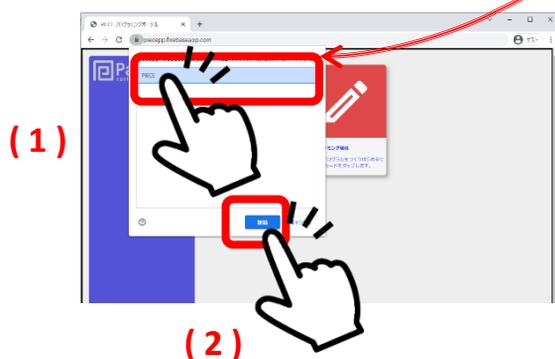
- ② モード切替スイッチを左にする



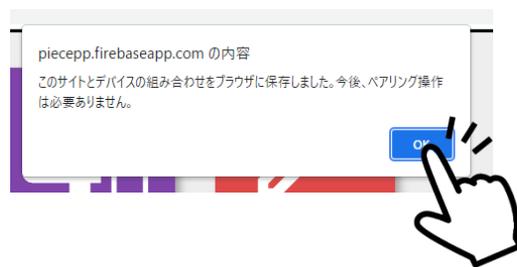
- ③ 「ペアリング」をタップ



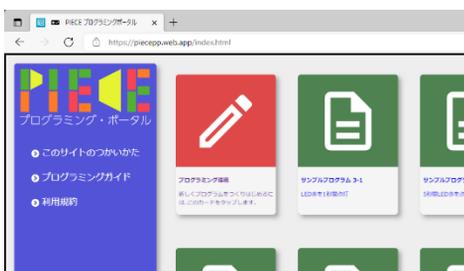
- ④ (1) PIECE をタップして選ぶ  
(2) **接続** を押す



- ⑤ **OK** をタップ



- ⑥ ペアリング終了

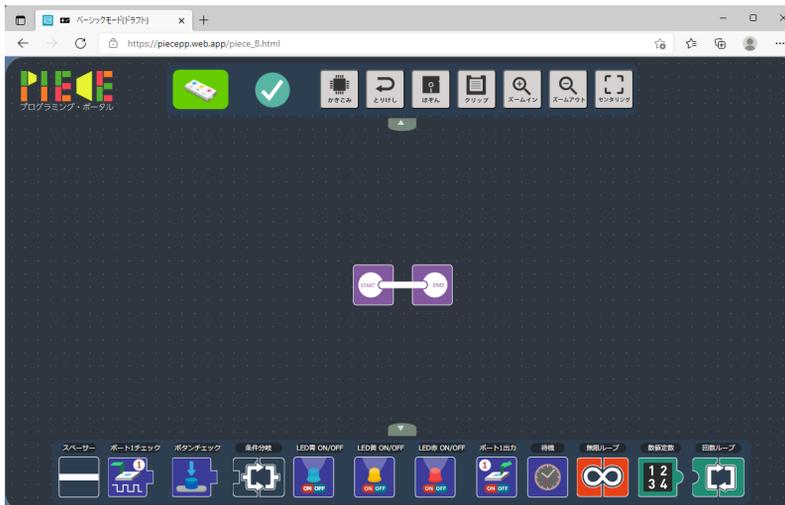
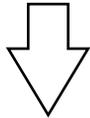




## プログラミングの準備

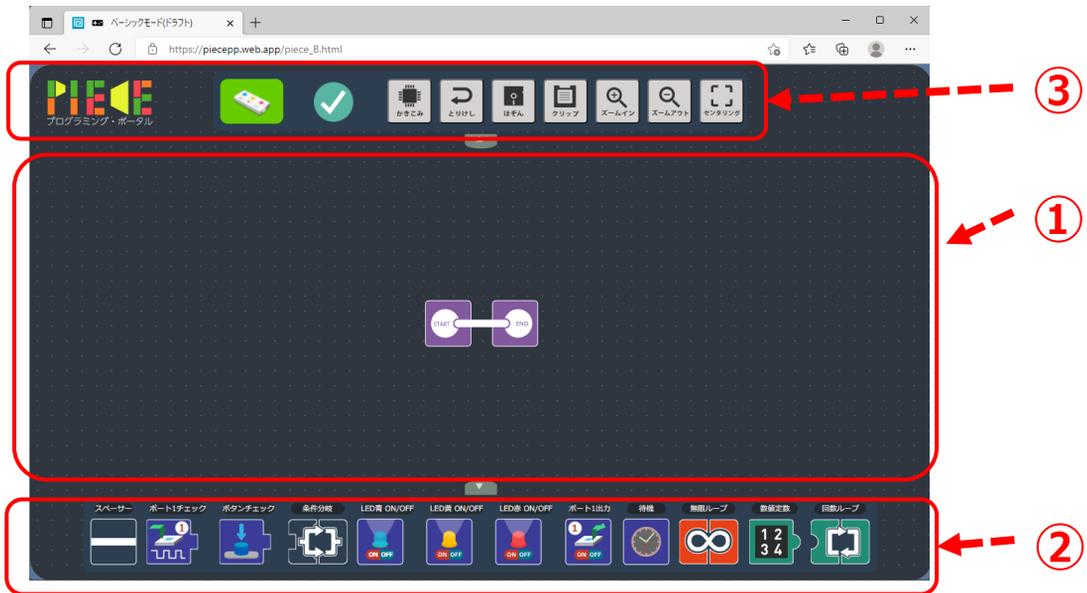
### 4 プログラム編集画面を開く

プログラムを新しく作るときは、プログラミングポータル画面の、プログラミング環境のカードをタップします。



プログラム編集画面が開きます。  
さあ、プログラミングをはじめよう！

## プログラム編集画面



### ①ワークスペース

この場所にプログラムを作成します。  
スタートとエンドのアイコンはあらかじめ配置されています。

### ②ドック

プログラムで使用する命令アイコンが並んでいます。  
LEDをオンオフするアイコンや、プログラミングモジュールの入力や出力を制御するためのアイコン、分岐やループのアイコンなどが用意されています。

### ③スイッチボード

接続の状態を表示したり、プログラムの保存や、プログラムの書込みなどの操作をするためのアイコンが並んでいます。



## アイコンの説明 : スイッチボードのアイコン



### USB チェック

プログラミングモジュールと正しく接続されているときに緑色のチェックマークが表示されます



### かきこみ

プログラムをプログラミングモジュールに書き込みます



### とりけし

プログラム作成操作を1つ前の状態に戻します



### ほぞん

作成したプログラムを保存します



### クリップ

現在プログラムを表示しているブラウザ画面のURLアドレスを、クリップボードに保存します。このURLをメールやメモ帳に貼り付けて他の人に伝えることで、別のパソコンのブラウザで同じ画面が開きます。



### ズームイン ズームアウト

ワークスペース表示の拡大、縮小をします



### センタリング

ワークスペースをセンタリングします



### ホーム

プログラム編集画面を閉じて、プログラミングポータル画面に戻ります



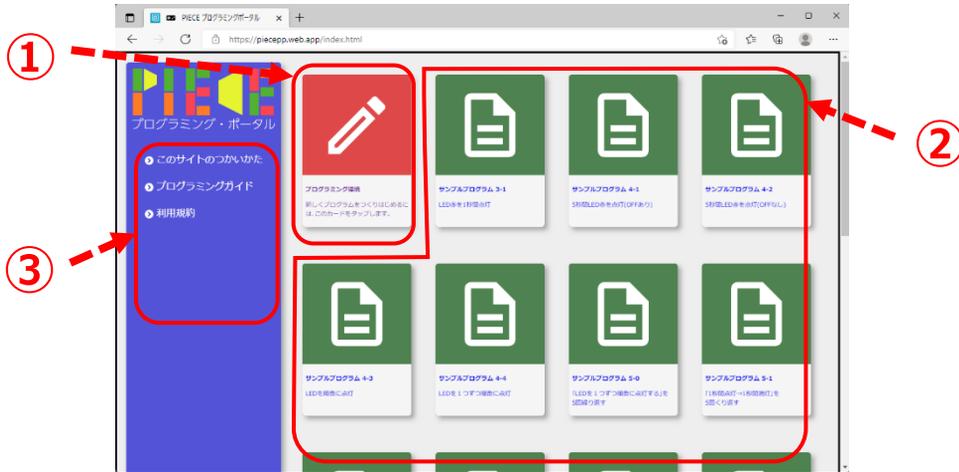
## アイコンの説明 : ドックのアイコン

ドックのアイコンの詳細な説明は、「13. アイコンの説明」で説明しています。次ページ以降の「やってみよう」でプログラミングに慣れて、オリジナルのプログラムを作るときに活用してください。



# プログラミングポータル画面の説明

## プログラミングポータル画面



### ①プログラミング環境

新しくプログラムを作成するときにタップします。

### ②ワークスペース

この場所に作成したプログラムや、サンプルプログラムが表示されます。

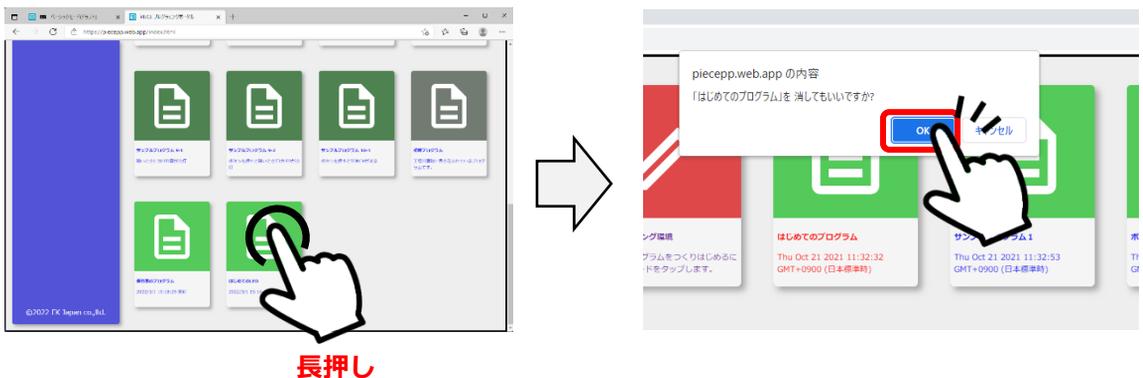
### ③情報スペース

プログラミングモジュールに関連する情報が表示されます。



# プログラムを削除する

削除したいプログラムのカード上で長押しするとメッセージが表示されるので、OKを押します。



※ あらかじめ用意されている、「プログラミング環境」と「サンプルプログラム」のカードは消すことができません。

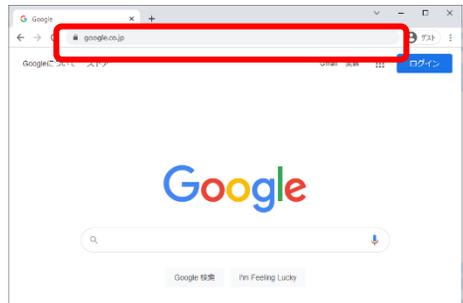
# 👉 プログラムのコピー

プログラムを作ってうまく動作したときは、プログラムのファイルをコピーして保存しておくの良いかもしれません。試行錯誤しているときに「昨日のプログラムに戻りたいけど上書き保存してしまって戻れない…」となってしまう前にコピーをしておきましょう。

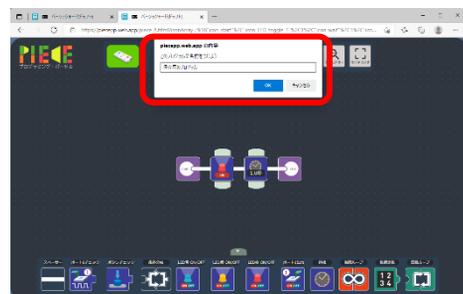
- ① 保存したいプログラムを開き、  
クリップ  をタップします。  
(URLがコピーされます)



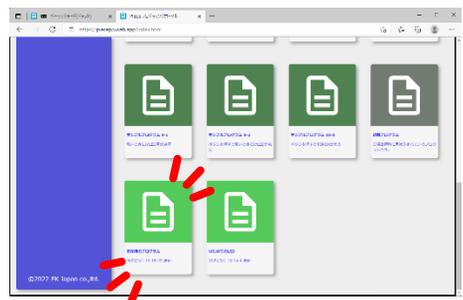
- ② ブラウザーの新規ページを開き、  
アドレスバーにURLを張り付けて、移動します。



- ③ プログラムの画面が開いたら、  
ほぞん  をタップし、  
名前をつけてOKをタップします。



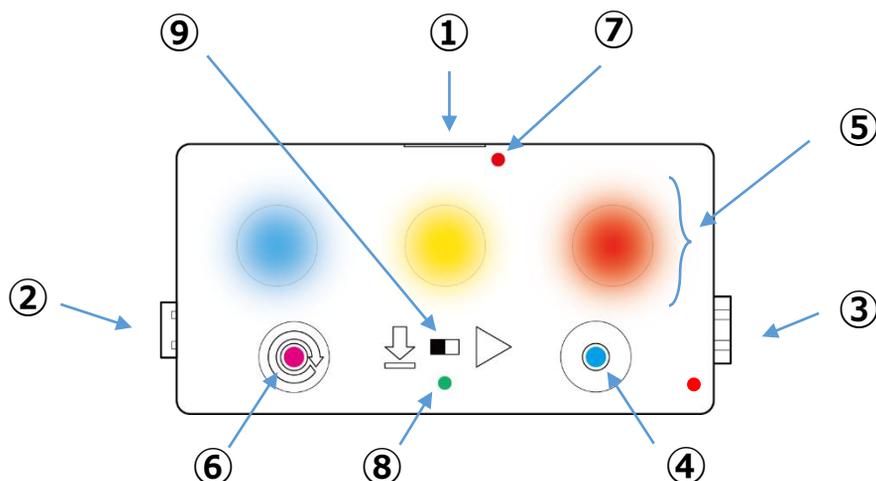
- ④ プログラムが保存されます。



## 2. プログラミングモジュールを知ろう



### 各部のなまえ



①	USB	付属のUSBケーブルをつなぎます。パソコンとUSBをつなぐことでプログラミングモジュールの電源になります。作成したプログラムを書き込むときにも利用されます。
②	入力1	別売のPIECEのオレンジモジュール(入力)や黄色モジュール(処理/論理)をつなぎます。
③	出力1	別売のPIECEの緑色モジュール(出力)や黄色モジュール(処理/論理)をつなぎます。プログラミングモジュールがオン信号を出している間はコネクタ近くの赤色LEDが点灯します。
④	ボタン	プログラムで自由に使用できるボタンです。
⑤	LED赤/黄/青	プログラムで自由に点灯/消灯できるLEDです。赤色、黄色、青色の3つがあります。
⑥	リセットボタン	ボタンを押すとプログラミングモジュールがリセットされ、プログラムを最初から実行します。
⑦	ステータスLED	パソコンと接続中や、リセット直後、プログラム実行が終了したときに点灯・点滅するLEDです。
⑧	電源表示LED	プログラミングモジュールに電源が接続されているときに点灯します。
⑨	モード切替スイッチ	動作モードを切り替えます。右にスライドすると「実行モード」左にすると「書き込みモード」になります。 <b>実行モード</b> : プログラムを実行します。 <b>書き込みモード</b> : プログラムを書き込みできます。

# 3. プログラムで動かしてみよう



## ここで使うアイコン



### スタート

このアイコンからプログラムが始まります。



### LED赤

プログラミングモジュールのLED赤をON/OFFします。



### エンド

全ての動作を停止し、プログラムを終了します。



### 待機

指定された秒数だけプログラムを一時停止します。



## アイコンを配置してプログラム

命令アイコンを画面下のドックからドラッグして、スタートとエンドの間の置きたい場所で離すとプログラム上にアイコンが配置されます。



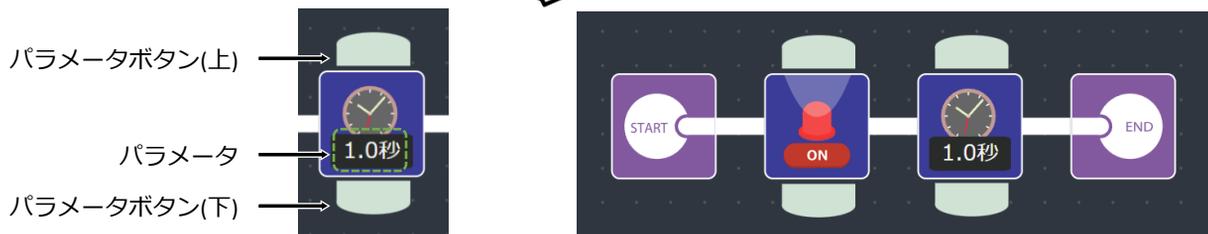
### 命令アイコンを消すとき

命令アイコンを消すときは、消したいアイコンをドラッグして、スタートとエンドの間ではない、アイコンを何も置いていない場所でドロップすると（手を離すと）アイコンを消すことができます。



# LED赤を1秒間点灯するプログラム

下図のような「LED赤を1秒間点灯する」プログラムをつくります。  
 待機アイコンの秒数は1秒に設定しましょう。アイコンの上下についている  
 パラメータボタン  をタップすると秒数が変わります。



(例3-1) [LED赤を1秒間点灯](#)



## パラメータの便利な入力方法

待機アイコンは0秒～10秒の範囲で時間を設定できます。  
 パラメータボタンをタップして値を設定することもできますが、  
 アイコンのパラメータを短くタップすると、入力ウィンドウが表示され、  
 キーボードで値を入力することもできます。

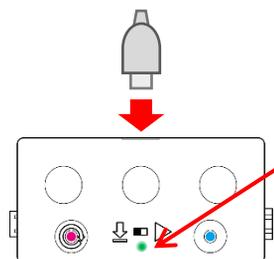




## プログラムを書き込もう

プログラムができたら、プログラムを書き込みましょう。

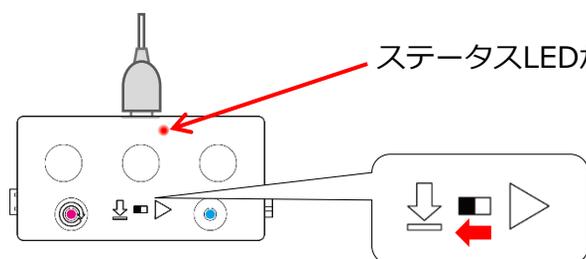
①プログラミングモジュールとパソコンをUSBケーブルで接続します。



電源表示LEDが点灯します。

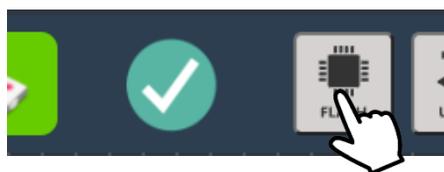
**充電専用USBケーブルでは書き込みできません。**  
付属のUSBケーブルを使います。

②プログラミングモジュールのモード切替スイッチを左側にスライドして書き込みモードにします。

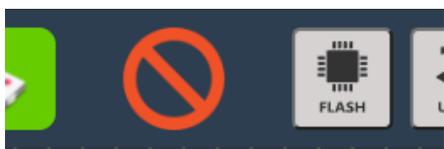


ステータスLEDが点灯することを確認します。

③FLASH(フラッシュ)ボタンをタップしてプログラムを書き込みます。



チェックマーク  が表示されているときにプログラムを書き込むことができます。



**未接続マーク**  が表示されているときはプログラムを書き込むことができません。  
ケーブルの接続とモード切替スイッチの位置を確認し、**プログラミングモジュールのリセットスイッチを一回押してみます。**

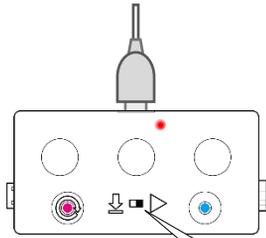
④画面にプログラムを転送しました と表示されれば書き込み完了です。  
OKをタップするとメッセージが消えます。



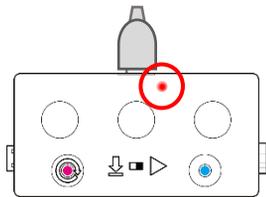
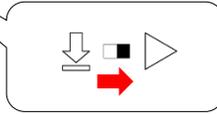


## 動作チェックをしよう

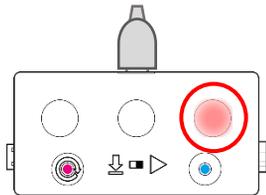
書き込んだプログラムを実行してみましょう。



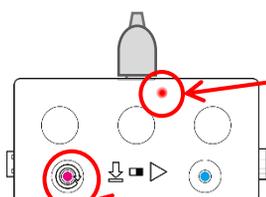
モード切替スイッチを右側にスライドして、実行モードにします。



実行モードにするとステータスLED赤が数回すばやく点滅したあとに、プログラムを実行します。



1秒間LED赤が点灯！



プログラムが終了するとステータスLED赤がゆっくり点滅します。

もう一度プログラムを実行するときは、リセットスイッチを一回押します。

## 4. 順次処理：順番に実行する

プログラムがスタートすると、命令された順番どおりに命令を実行していきます。このことを順次処理といいます。PIECEのプログラムもスタートアイコンから始まって、線につながっているアイコンを順番に処理していきます。ここでは順次処理をやってみましょう。



### この章で使うアイコン



#### LED黄

プログラミングモジュールのLED黄をON/OFFします。  
パラメータボタンを押してONとOFFを切り替えることができます。



#### LED青

プログラミングモジュールのLED青をON/OFFします。  
パラメータボタンを押してONとOFFを切り替えることができます。



## 5秒間LED赤を点灯する

下図のような「5秒間LED赤を点灯」するプログラムを作ります。待機アイコンのパラメータを5.0秒に設定しましょう。



パラメータボタンを押してOFFにします

(例4-1) [5秒間LED赤を点灯](#)



### 動作チェックをしよう

- START → LED赤(ON) → 待機 (5秒間) → END  
の動きはうまくできましたか？



### ワンポイント

下図のようなプログラムを作って、動きを確かめてみましょう。



(例4-2) [LEDをOFFしないで終了](#)

さっき作ったプログラムとは少し違い、LED赤をOFFにする命令がないのに全く同じ動きかたをします。

これは「END」アイコンが「すべての出力をOFFにしてプログラムを終了する」というアイコンなので、LED赤をわざわざOFFにしなくても勝手にOFFになってしまうためです。どちらのプログラムでも間違いではありませんが、作成するプログラムが大きく複雑になってくると、ちゃんとプログラムの中にOFFにする処理を書いておいたほうが後で見直しをするときにミスを発見しやすく、他の人が見たときにも分かりやすくなります。



## LEDを順番に点灯する

3つのLEDが順番に点灯するプログラムをつくってみましょう。

LED赤 → LED黄 → LED青 の順番で点灯させます。

順番に点灯していくように見せるために、LEDが点灯して次のLEDが点灯するまでに1秒間待つことにします。

下図のようなプログラムを作ります。待機アイコンのパラメータを1.0秒に設定しましょう。



(例4-3) [LEDを順番に点灯](#)



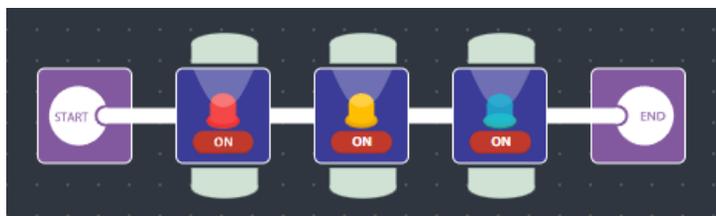
### 動作チェックをしよう

- LEDは順番に点灯しましたか？



### ワンポイント

プログラムはアイコンを順番に処理していくので、下図のように待機アイコンを使わなくても、LEDは順番に点灯しそうですね。



でもプログラムを書き込んで動かしてみると、ほんの一瞬全部のLEDが点灯するだけです。その理由は、プログラムはちゃんとスタートして、LEDを点灯させるアイコンを実行しているのですが、すぐに最後のENDアイコンに到着してしまい、プログラムが終了してLEDがOFFになってしまうからです。

このようにプログラムに期待する動きと実際の動きが違う場合は、プログラムにどこがおかしいところがないか、考え方が間違っていないか、アイコンの動作を間違えて覚えていないかなどを確認をするようにしましょう。



## LEDを1つずつ順番に点灯する

3つのLEDが順番に1つずつ点灯するプログラムをつくってみましょう。さっき作ったプログラムでは最後に3つ全部点灯しましたが、今度は、1つ点灯させたらそのLEDを消して、次のLEDを点灯させるようにします。

下図のようなプログラムを作ります。待機アイコンのパラメータを1.0秒に設定しましょう。



(例4-4) [LEDを1つずつ順番に点灯](#)



## 動作チェックをしよう

- LEDが1つ点灯して消えて、次のLEDが点灯というように順番に点灯しましたか？

## 5. くり返し処理：同じ動きをくり返そう

### 📖 同じ動きのアイコンをたくさん並べる？

例えば、(例4-4)で作った「LEDを1つずつ順番に点灯する」を5回繰り返すプログラムを作ろうとしたとき、どうすれば実現できるでしょうか？



(例5-0) [「LEDを1つずつ順番に点灯する」を5回繰り返す](#)

これで動かすことができます。でもこのプログラム、同じことを繰り返すだけの単純な動きなのにとっても長くて読みにくいですね。それに「繰り返す回数を100回にしてください」なんて言われたらもう大変ですね。この章ではこういった単純な動作を繰り返し行うときに便利な「くり返し処理」について学んでいきます。



### この章で使うアイコン



#### 回数ループ

囲まれた部分のプログラムを数値定数で指定した回数だけ繰り返します。



#### 数値定数

回数ループなどに使う数値を指定します。  
パラメータボタンによって数値を増減できます。



#### 無限ループ

囲まれた部分のプログラムを無限に繰り返します。



## 「1秒間点灯 → 1秒間消灯」を5回くり返す

「1秒間点灯→1秒間消灯」を5回くり返すプログラムをつくります。  
くり返したい動作を「回数ループ」のはじめと終わりのアイコンの間に置き、  
くり返す回数を「数値定数」で設定します。数値定数アイコンのパラメータボ  
タンの上をタップして値を5にします。



(例5-1) [「1秒間点灯→1秒間消灯」を5回くり返す](#)

### ✓ 動作チェックをしよう

- 「LED赤(ON) →待機(1秒間)→LED赤(OFF) → 待機(1秒間)」を5回くり返して動きますか？



#### パラメータの便利な入力方法

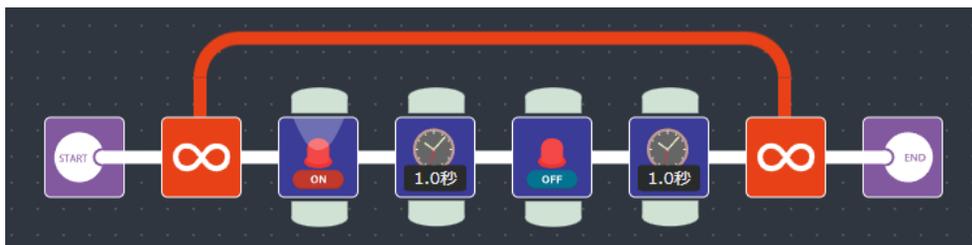
数値設定アイコンは-100～100の範囲で数を設定できます。  
パラメータボタンをタップして値を設定することもできますが、  
アイコンのパラメータを短くタップすると、入力ウィンドウが表示され、  
キーボードで値を入力することもできます。





## 「1秒間点灯 → 1秒間消灯」をずっとくり返す

自動で装置を動かす場合、プログラムが終了することなく、ずっと動き続けてほしいことがほとんどです。「無限ループ」を使って、さっき作ったプログラムをずっとくり返してみましょ。



(例5-2) [無限ループ](#)



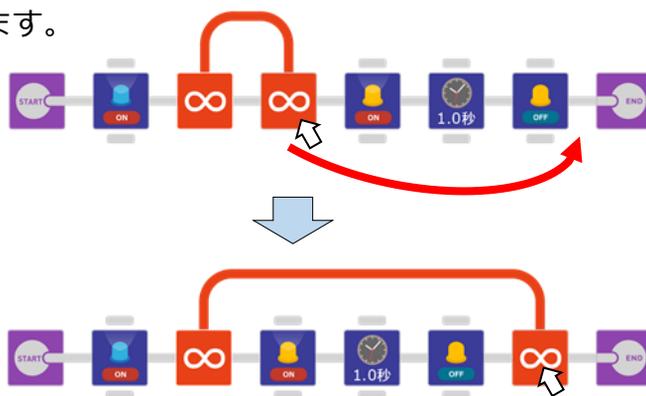
### 動作チェックをしよう

- 「LED赤(ON) → 待機(1秒間) → LED赤(OFF) → 待機(1秒間)」をずっとくり返して動きますか？

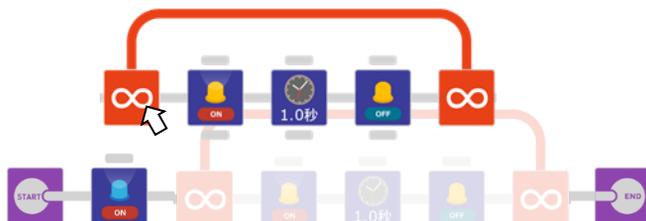


#### 無限ループ/回数ループの操作方法

ループアイコンの後ろ側のアイコンをドラッグして囲む範囲を変えることができます。



ループアイコンの先頭側をドラッグするとループアイコンで囲まれている範囲を一緒に移動することができます。



## 6. 条件分岐：条件によって動きをかえよう

### 条件で動きをかえる

人は道を歩いているときに、目の前に急にボールがとびだしてきたら、すぐに止まることができますよね。これは「ボールがとびだしてきたら」止まる、「ボールがないときは」歩く。というように条件によって動きをかえているということです。

このように条件によって動きをかえることを、条件分岐といいます。ここでは条件分岐のプログラムを学びましょう。



### この章で使うアイコン



#### 条件分岐

このアイコンの前につながっているアイコンの状態分で岐します。アイコンの状態が[○]なら分岐の○側、アイコンの状態が[×]なら分岐の×側のプログラムを実行します。



#### ボタン

プログラミングモジュールについているボタンの状態をチェックするときに使います。ボタンが押されているときは[○]、押されていないときは[×]になります。

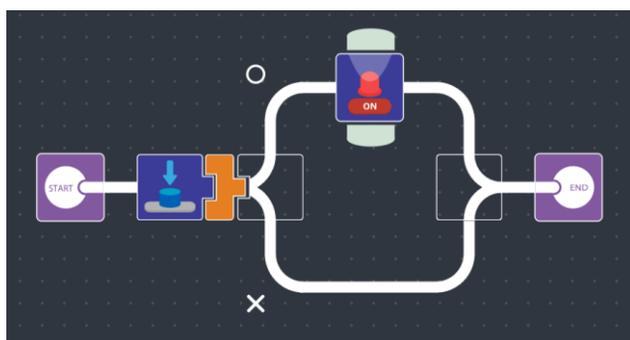


## ボタンを押すとLED赤が点灯するプログラム（1）

「ボタンを押すとLED赤が点灯する」プログラムをつくります。

ボタンが押されているかどうかをチェックするためには、ボタンアイコンを使います。ボタンアイコンはプログラミングモジュールのボタンが押されていると、次につながっているアイコンに[○]を渡し、ボタンが押されていないときは[x]を渡します。

[○]と[x]で動きを変えるには条件分岐アイコンを使います。条件分岐アイコンは、前につながっているアイコンから[○]を受け取ったら上の線へ、[x]を受け取ったら下の線へ進みます。



(例6-1) ボタンを押すとLED赤が点灯(1)



### 動作チェックをしよう

期待した通りに動きましたか？

実はこのプログラムは**期待した通りに動きません**。電源をONにしてプログラムを実行しても、すぐにステータスLEDが点滅するだけで、ボタンを押しても何の変化もありません。なぜでしょうか。

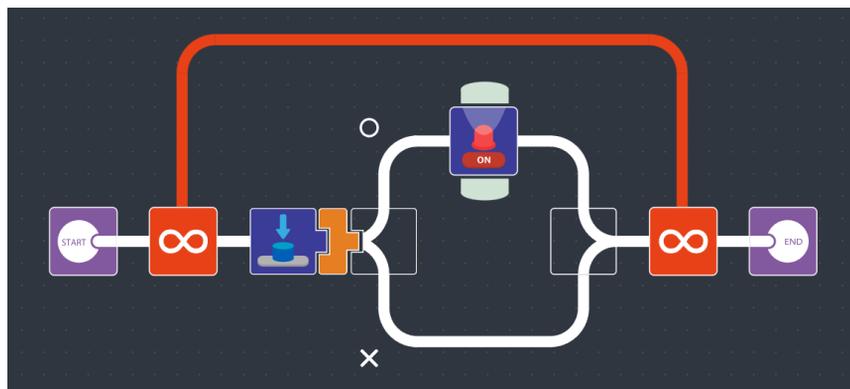
これは、プログラムがスタートすると、すぐにボタンの状態をチェックしてどちらかの道を進んで、すぐにENDに到着してしまうからです。ほんの一瞬で終わります。つまりプログラムが終わった後にボタンを押している訳です。

ずっとボタンの状態をチェックするためには、プログラムがずっと動き続けなといけません。**ずっと動くプログラム**を作るときは**無限ループ**を使います。



## ボタンを押すとLED赤が点灯するプログラム（2）

「ボタンを押すとLED赤が点灯する」を無限ループで囲んだプログラムをつくれます。



(例6-2) [ボタンを押すとLED赤が点灯\(2\)](#)



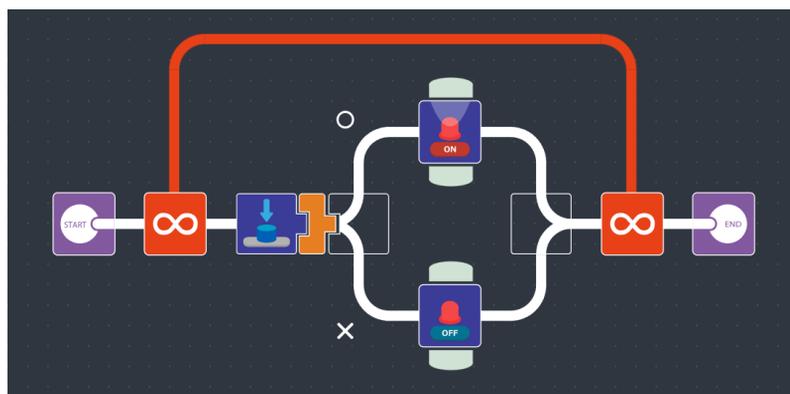
### 動作チェックをしよう

- ボタンを押すとLED赤が点灯しますか？



### やってみよう

さっきのプログラムは、ボタンを押すとLED赤が点灯しましたが、今度は、ボタンを押している間はLED赤が点灯して、押していないときはLED赤が消えているプログラムを作ってみましょう。



(例6-3) [ボタンを押すと点灯、離すと消灯](#)



## 信号機を作ってみよう

身近にある、押しボタン式の歩行者用信号機の動きを再現してみましょう。信号機の動きをよく観察してどうやって光っているのか詳しく調べましょう。

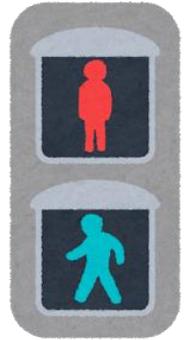
- ・ ボタンを押すまで信号機は何色？
- ・ ボタンを押してから何秒後に青に変わる？
- ・ 青から赤に変わるときはどんな光り方をしている？



### こんな信号機を作ってみましょう

- ・ 普段は(ボタンが押されていないとき)は赤色が点灯。
- ・ ボタンを押すと、「しばらくお待ちください」が光る。
- ・ ボタンを押してから3秒後に青色が点灯する。
- ・ 青色が6秒間点灯して、点滅を始める。
- ・ 青色は点滅を5回繰り返して消えてから、赤色が点灯する。

※「しばらくお待ちください」ランプはないので、ここではLED黄を点灯させることにします。



今まで学んだ、順序処理、くり返し処理、条件分岐の組み合わせでプログラムを作ってみましょう！



### 考えてみよう

さっき作ったプログラムには「しばらくお待ちください」が点灯する機能を入れましたが、この機能は本当に必要なのでしょうか？ないとどうなるのでしょうか？「しばらくお待ちください」機能をなくしたプログラムを作って実際に使ってみましょう。

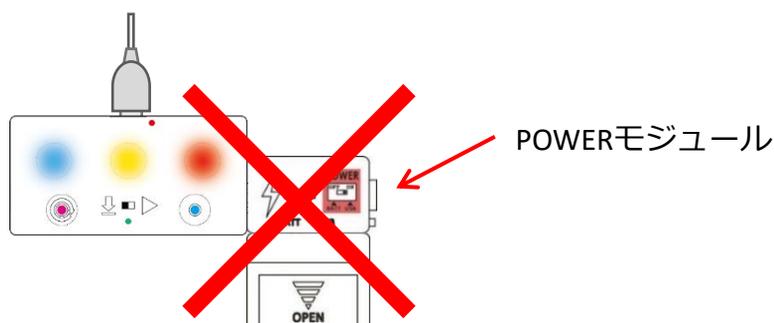
実際に使ってみると、ボタンを押したかどうか分からなくて何度もボタンを押してしまいます。使いづらいですね。一見必要なさそうな機能でも、使いやすくするためには大事な機能であることがわかりますね。

## 7. PIECEのモジュールつないでみよう

プログラミングモジュールの入力端子や出力端子に、PIECEのモジュールをつないでつなぐことができます。ここではセットに入っている、LIGHTセンサーモジュールを入力1につないだときの使い方を説明しています。

別売の他のPIECEセットのモジュールの利用する場合は、以下の点に注意してお持ちのモジュールを接続してください。

PIECEプログラミングモジュールがUSBの電源で動いているときには、絶対に**PIECEの電源モジュール(POWERモジュール)**をつながないでください。発熱の恐れがあり、故障の原因となります。



### 準備しよう

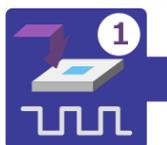
プログラミングモジュールに下図のようにLIGHTセンサーモジュールをつないでください。



## 8. プログラミングモジュールの入力をつかってみよう



### ここで使うアイコン



#### ポート1チェック

プログラミングモジュールの入力1に接続したPIECEモジュールの状態を調べるときに使います。

PIECEモジュールがオン信号を出しているときは[○]に、信号をだしていないときは[×]になります。



#### ポートとは？

ポート（PORT）とは、日本語に訳すと「港（みなと）」です。

港は、船が街に出たり入ったりするために必要な場所です。電気の世界でもこれになぞらえて、電気信号を出したり受け取ったりする端子のことをポートといいます。受け取り専用の端子のことを入力ポート、信号を出す専用の端子を出力ポートとよんだりします。

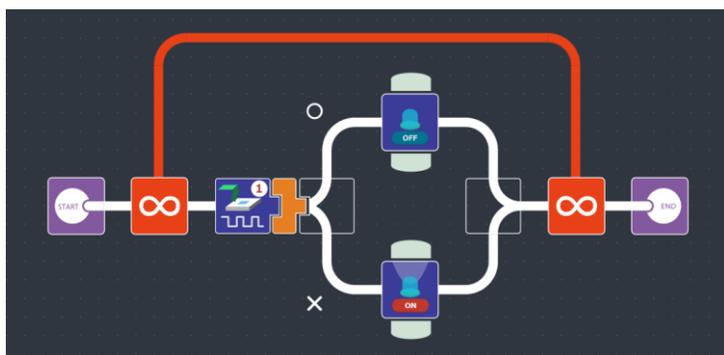


### 暗いときにはLED青が点灯する

「暗いときにはLED青が点灯する」プログラムを作ってみましょう。考えるときのポイントは以下の点です。

- ・明るさをチェックしたいので「LIGHT」センサーを使う。
- ・“暗いとき”ということは“明るくないとき”ということ。
- ・LIGHTセンサーは入力1につながっている。
- ・明るいときと暗いときで動きがかわるので、条件分岐を使う。
- ・ずっと動き続けて欲しいので無限ループを使う。

プログラムを作成すると下図のようになります。



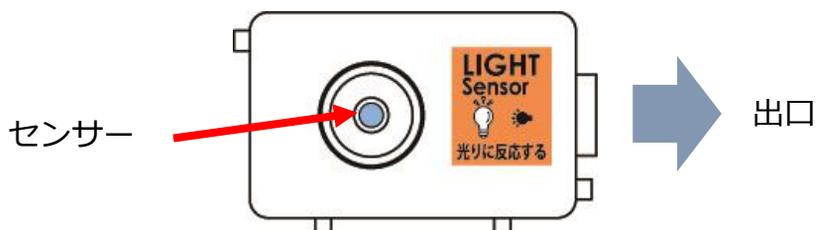
(例9-1) [暗いときにはLED青が点灯する](#)

## ☑ 動作チェックをしよう

- センサーの周囲を暗くすると、LED青が点灯しましたか？

知っておこう：

### LIGHT センサーモジュール



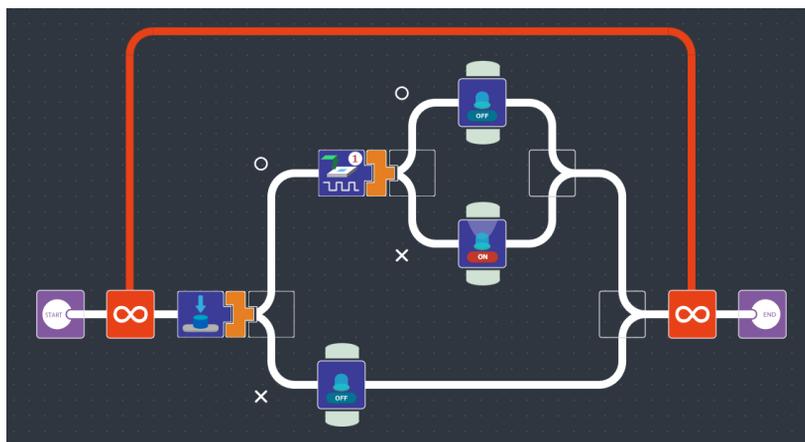
センサーが光を感じると出口からオン信号が出ます。



## ボタンを押すと暗いときにだけLED青が点灯する

入力を2つ使って、条件が2つになった場合のプログラムを作ってみましょう。  
考える方のポイントは以下ようになります。

- ・ ボタンが押されて、かつ、明るくないとき(暗いとき) は点灯
- ・ それ以外のときは消灯する



(例9-2) ボタンを押すと暗いときにだけLED青が点灯



### 動作チェックをしよう

- 暗いときにボタンを押すとLED青が点灯し、ボタンから手を離すと消灯し、明るいときはボタンを押しても、押していなくてもLED青が消灯したままになりましたか？



### プログラムを観察してみよう

(例9-1)で光センサーのプログラムを作ったときは、光センサーの条件分岐で○と×の2通りだけ動きを設定しましたが、ここでは「光センサーの条件分岐」が「ボタンでの条件分岐」の中に入っています。このように条件分岐の中にまた条件分岐を入れている構造のことを「**ネスト (入れ子)**」といいます。ネストを使うことで条件分岐を3通り以上にすることができます。

## 9. プログラミングモジュールの出力をつかってみよう

ここでは別売の、「いろんなセンサーでライトをつけようセット」(ZZ-AC01)に用意してある出力モジュール(TORCH)を使って説明しています。

他のセットをお持ちの場合は、お手元の出力モジュールに合わせて文章を読みかえるなどしてご利用くださいますようお願いいたします。



### ここで使うアイコン



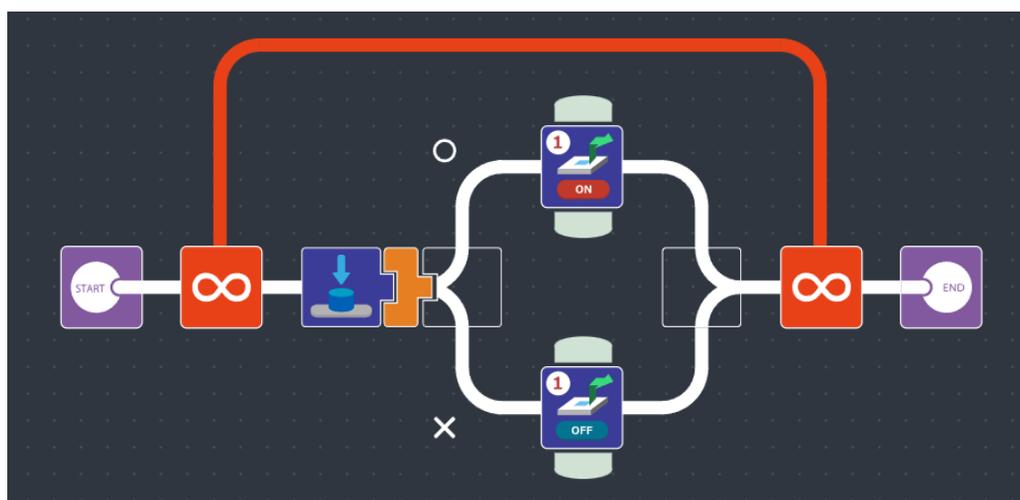
#### ポート1出力

プログラミングモジュールの出力1をオン/オフします。パラメータボタンを押してオンとオフを切り替えます。



### ボタンを押している間TORCHが光る

プログラムを作成すると下図のようになります。



(例10-1) [ボタンを押すとTORCHが光る](#)



### 動作チェックをしよう

- ボタンを押すとTORCHが光り、離すと消灯しましたか？

# 10. いろいろな装置をつくってみよう

プログラミングモジュールのLEDや入力1/出力1を、順序処理、くり返し処理、条件分岐のプログラムで動かす方法を学びました。これらを組み合わせて、身の回りがあったら便利になるもの、安全になるものを作ってみましょう。プログラムをつくって、使ってみて、使いにくいのか、もっと省エネできないかなどを考えてより便利な装置にすることに挑戦してみましょう。

## 玄関が暗くて危ないなあ・・・

暗くなると自動で点灯するライト

↓もっと便利にしてみる

暗くなると自動で点灯して、ボタンを押すと消灯するライト



## 部屋の電気を消してベッドまで行くときに暗いなあ・・・

暗くなると1分間点灯してから消えるライト

↓もっと楽しく

暗くなると1分間、ライトやLED赤黄青が賑やかに点滅する



いろいろな課題をみつけて、自分の装置をつくることに挑戦してみましょう。

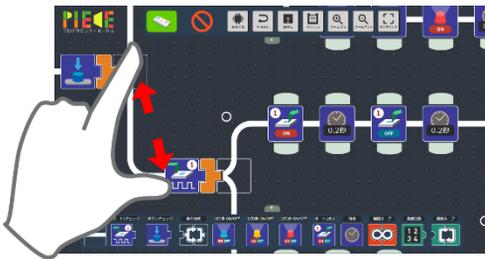
# 1 2. 覚えると便利な操作テクニック



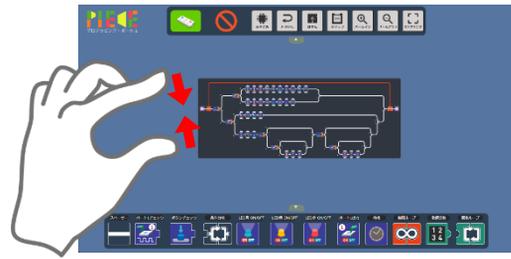
## 拡大・縮小

ピンチアウト・ピンチイン操作でプログラムの拡大・縮小をすることができます。

ピンチアウトで拡大



ピンチインで縮小



## 画面のスクロール・センタリング

プログラムを作成するとき、何も無い(アイコンが置かれていない)位置でドラッグ操作を行うとワークスペース全体のスクロール操作ができます。また、何も無い位置でダブルタップ(画面を軽くたたく操作を2回繰り返す)をするとワークスペースをセンタリングします。

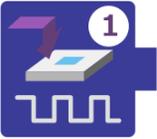


## プログラムのCLIP (クリップ)

現在プログラムを表示しているブラウザー画面のURLアドレスを、クリップボードに保存します。

このURLをメールやメモ帳に貼り付けて他の人に伝えることで、別のパソコンのブラウザーで同じ画面が開きます。

# 13. アイコンの説明

アイコン	アイコンの名前・説明
	<p> <b>&lt;LED赤 オン/オフ&gt;</b>  <b>&lt;LED黄 オン/オフ&gt;</b>  <b>&lt;LED青 オン/オフ&gt;</b> </p> <p>プログラミングモジュールのLED赤/黄/青をオン/オフします。パラメータボタンを押してオンとオフを切り替えることができます。</p>
	<p><b>&lt;ボタン&gt;</b> プログラミングモジュールについているボタンの状態をチェックするときに使います。ボタンが押されているときは[○]、押されていないときは[×]になります。</p>
	<p><b>&lt;ポート1チェック&gt;</b> プログラミングモジュールの入力1に接続したPIECEモジュールの状態をチェックするときに使います。PIECEモジュールがオン信号を出しているときは[○]に、信号をだしていないときは[×]になります。</p>
	<p><b>&lt;ポート1出力&gt;</b> プログラミングモジュールのポート出力1をオン/オフします。パラメータボタンを押してオンとオフを切り替えます。</p>
	<p><b>&lt;スタート&gt;</b> このアイコンからプログラムが始まります。</p>
	<p><b>&lt;エンド&gt;</b> 全ての動作を停止し、プログラムを終了します。</p>

アイコン	アイコンの名前・説明
	<p><b>&lt;待機&gt;</b>            パラメータによって設定した時間プログラムの処理を待機させます。待機している間、直前の動作が継続されます。設定できる時間は最小0秒で、0.1秒ずつ増やすことができ最大で10秒です。初期値は1秒です。</p>
	<p><b>&lt;条件分岐&gt;</b>            真偽値を受け取り、[○]を受け取った場合は上の経路へ、[×]を受け取った場合は下の経路へ処理を進めます。            ※「条件分岐」アイコンはデータ(真偽値)を受け取るアイコンであるため、その左側には必ずデータ(真偽値)を渡すアイコンが配置されなくてはなりません。</p>
	<p><b>&lt;スペーサー&gt;</b>            プログラムを読みやすくするための目的で、アイコンとアイコンの間にスペースを挿入するためのアイコンです。プログラムの動作に与える影響はありません。</p>
	<p><b>&lt;無限ループ&gt;</b>            [無限ループ]内の処理を実行します。ループの最後に到達した場合は、[無限ループ]の先頭に戻ります。</p>
	<p><b>&lt;回数ループ&gt;</b>            数値を受け取り、その回数、[回数ループ]内の処理を実行します。</p>
	<p><b>&lt;数値定数&gt;</b>            パラメータボタンで設定した数値を右のアイコンに渡します。パラメータボタンをクリックまたはドラッグして数値を設定します。設定できる数値は最小-100、最大100の範囲内の整数です。</p>

# 14. トラブルシューティング

USBチェックアイコンが緑色にならず書き込めない。



- ・ 充電専用USBケーブルを誤って使っていないか確認します。プログラムの転送には付属のUSBケーブルを使います。
- ・ プログラミングモジュールのモード切替スイッチの位置を確認します。
- ・ プログラミングモジュールのリセットスイッチを押してみます。
- ・ パソコン側のUSB端子が不調である可能性があります。パソコンに別のUSB端子があれば、そちらの端子を使って確認してみます。

保存したプログラムが見つからない。

- ・ プログラムを保存したときに使用したブラウザーであることを確認します。
- ・ ブラウザーにログインしているユーザーがもそのときのもあることを確認します。  
※ブラウザーが同一であっても、ログインユーザーが違っているときもプログラムは表示されません。  
※ブラウザーのシークレットモードを使っていた場合はプログラムは保存されていません。