

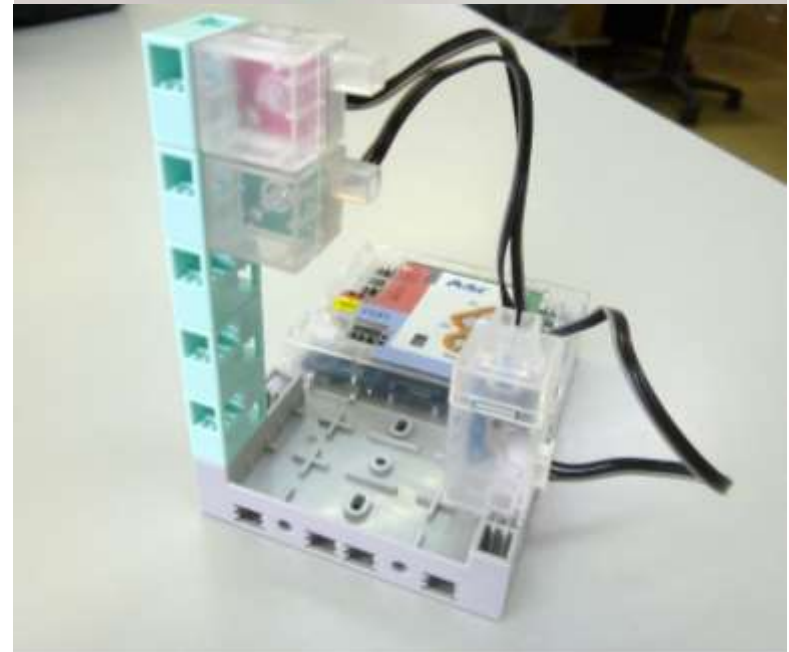
PGCon (ピジコン) 1

# PGCon (ピジコン) とは？

LEDの制御等を体験するプログラミング教材



PGCon (ピジコン)



Stduino (スタディノ)

# PGCon (ピジコン) の活用例

小学校 6 年生 理科 電気の利用

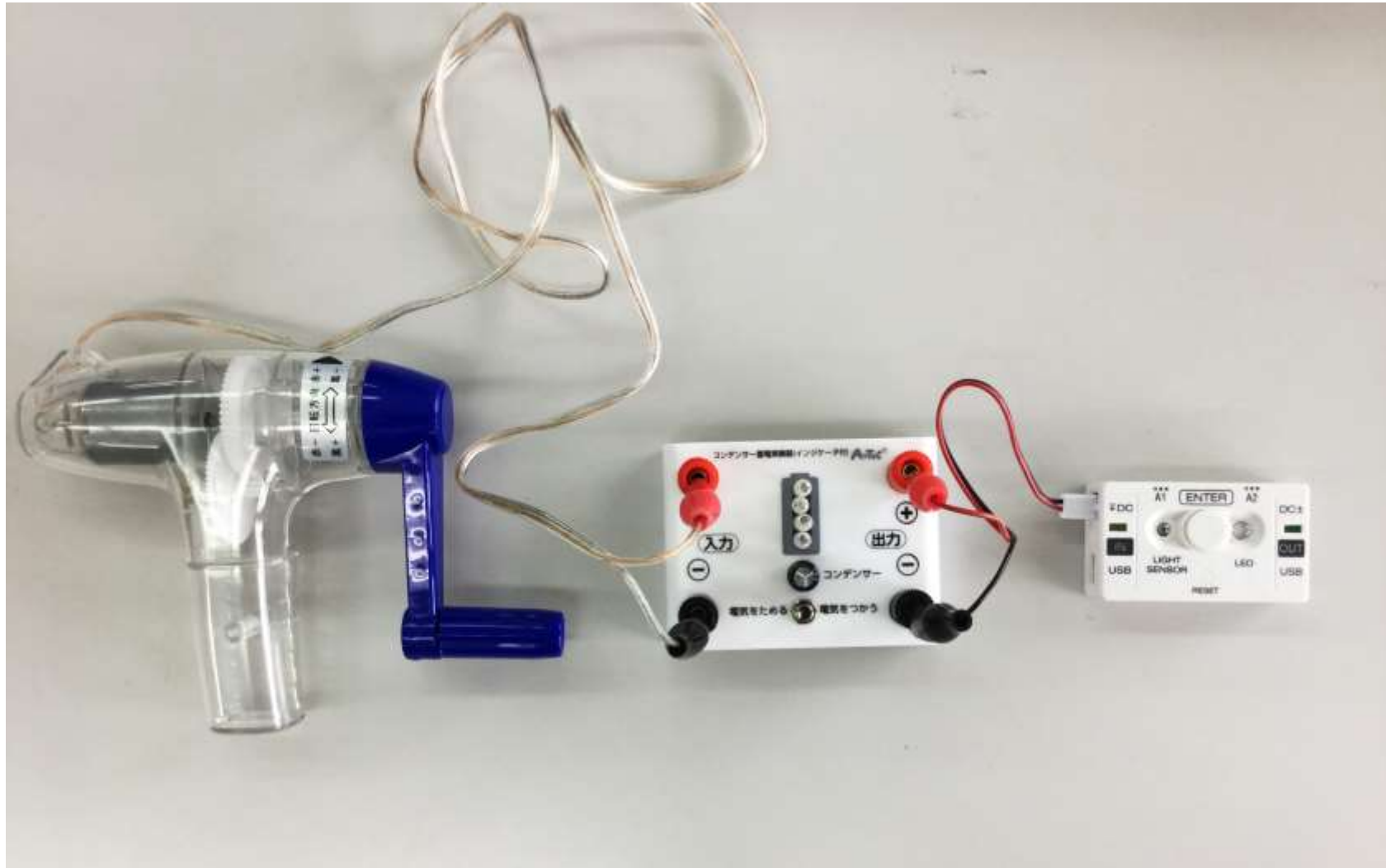
電気を効率的に利用する仕組み

- 暗くなったら点灯する
- 人が近づいたら点灯する
- 一定の時間が経てば自動的に消える

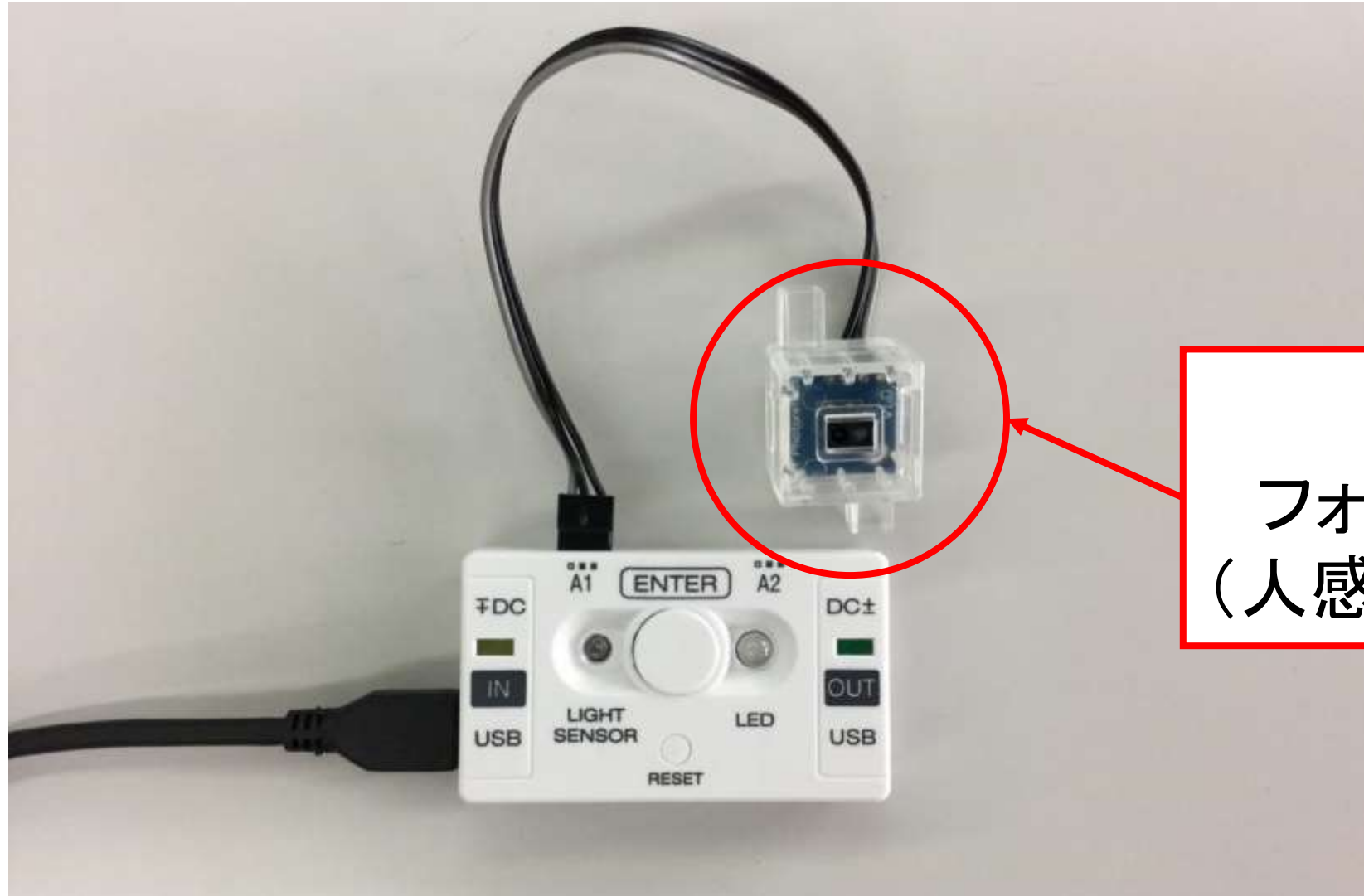


**プログラミングの体験を通して制御の仕組みを学ぶ**

# 発電機と蓄電池をつなげることも可能



# Studinoの部品をつなげることも可能



赤外線  
フォトリフレクタ  
(人感センサー用)

# USB給電型の道具もつなげることが可能



デスクライト

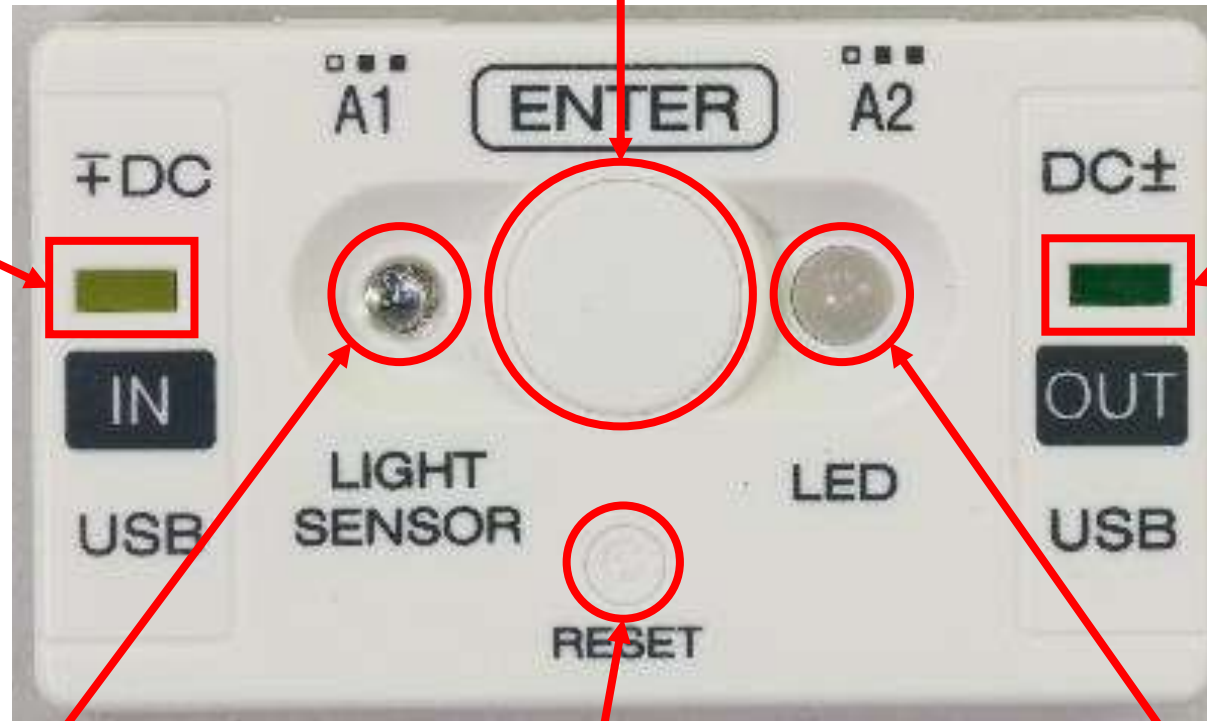


扇風機

# PGCon (ピジコン) の各部の名称と役割 1

ENTERボタン

電源入力  
確認用LED  
(黄)



電源出力  
確認用LED  
(緑)

光センサー

リセットボタン

フルカラーLED

# PGCon (ピジコン) の各部の名称と役割 1

- 光センサー  
明るさを0～100の値で計測。明るいほど値は大きくなる。
- ENTERボタン  
押しているときは0、押していないときは1の値を示す。
- フルカラーLED  
RGBをそれぞれ0～255で指定して色を設定。点灯・消灯の制御が可能。
- リセットボタン  
転送されたプログラムをはじめから実行。
- 電源入力確認用LED(黄)  
正常に電源が入力されていると黄色に点灯。
- 電源出力確認用LED(緑)  
正常に電源が出力されていると緑色に点灯。



# PGCon (ピジコン) の各部の名称と役割 2

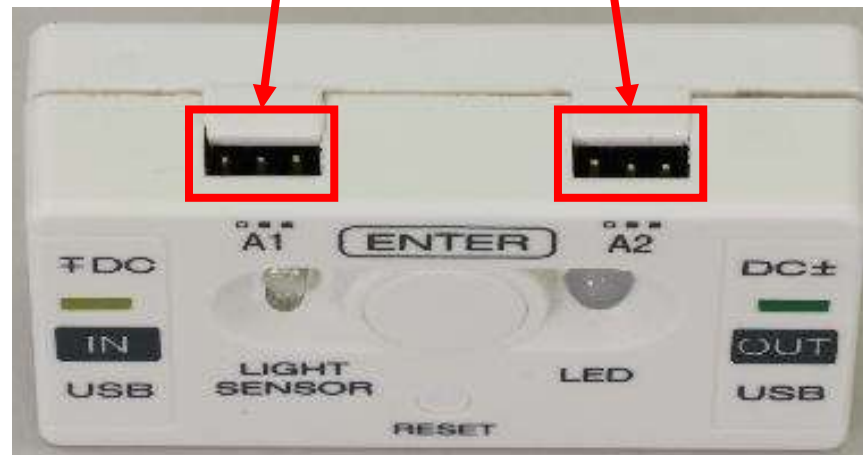
左側面



入力端子  
(USB)

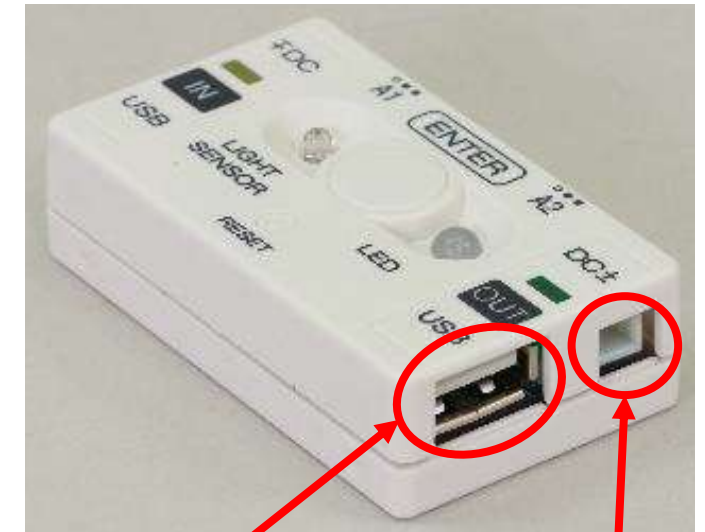
入力端子(DC)

外部アナログ  
入出力



上面

右側面



出力端子  
(USB)

出力端子(DC)

# PGCon (ピジコン) の各部の名称と役割 2

- 入力端子 (USB)  
PCとの通信ケーブルのコネクタ。USB microB規格。電源供給も可能。
- 入力端子 (DC)  
外部電源 (乾電池等) を入力するコネクタ。XHタイプ。
- 出力端子 (USB)  
USB給電で動作する機器を接続するコネクタ。USB A規格。
- 出力端子 (DC)  
USB機器以外の外部機器に出力するコネクタ。XHタイプ。
- 外部アナログ入出力  
アーテック製のロボット用の機器が接続できます。音センサー・光センサー・赤外線フォトリフレクタ・タッチセンサー・電子ブザー・LED・温度センサー

# PGCon (ピジコン) の制御に必要なもの

- コンピュータ (Windows)
- PGConソフトウェア
- USBケーブル (microBタイプ)



コンピュータで作成したプログラムをUSBケーブルで転送

# PGConソフトウェア

- Scratch1.4をベースにPGCon制御用に開発
- ウェブサイトからダウンロード(無料)  
サイトはこちら <http://artec-kk.co.jp/dl/153019/>
- インストールは不要  
ダウンロードしたファイル(pgcon.zip)を展開  
展開されたフォルダ(PGCon)内のpgcon.exeを起動



# カテゴリー

PGCon

ファイル 編集 実行 ヘルプ

aA aA aA

- 動き
- 制御
- 調べる
- 演算
- 変数

```
制御スタート  
ずっと  
 本体のLEDを 赤 で点灯する  
 1秒待つ  
 本体のLEDを消灯する  
 1秒待つ
```

# スプライトエリア

# ブロックパレット

- ブザー A1 から 60 を出力する
- ブザー A1 の出力を停止する
- LED A1 点灯
- 本体のLEDを 黒 で点灯する
- 本体のLEDを 赤: 255 緑: 255 青: 255
- 本体のLEDを消灯する
- OUTから電気を 100 % 流す
- OUTからの電気を止める
- LEDテープA2の 1 番目を 黒 で点灯する
- LEDテープA2の 1 番目を 赤: 255 緑: 255 青:
- LEDテープA2のすべてを 黒 で点灯する
- LEDテープA2のすべてを 赤: 255 緑: 255 青: 255
- LED

# プログラムの作成から実行まで

1 ブロックパレット内のブロックをスプライトエリアに置く



ドラッグ & ドロップ

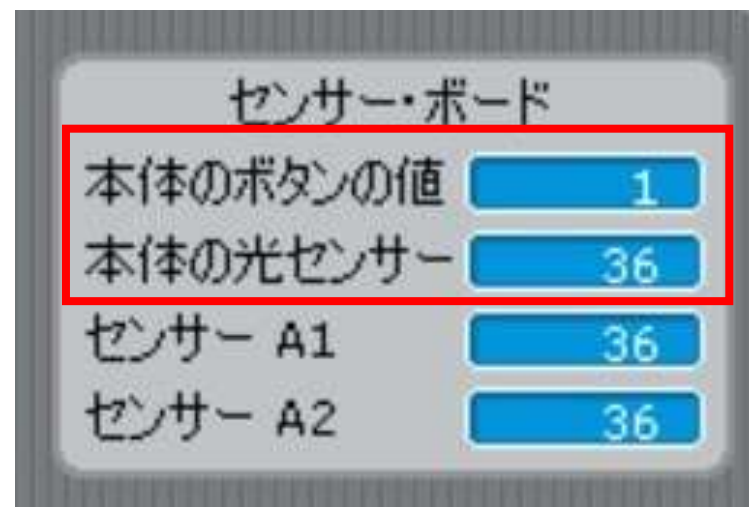
白い線を見て

ピッタリくっつける



# プログラムの作成から実行まで

## 2 光センサーやENTERボタンの値はテストモードで確認



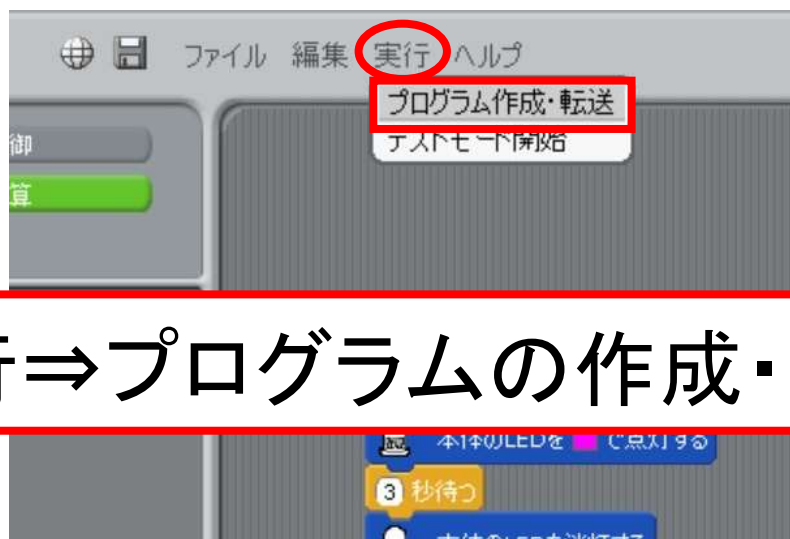
実行⇒テストモード開始

ボタン 押す…0 押さない…1  
光センサー 明るいほど値は大



# プログラムの作成から実行まで

## 3 プログラムの作成・転送で実行



実行⇒プログラムの作成・転送

思い通りの動作をしなかった場合はプログラムを修正する



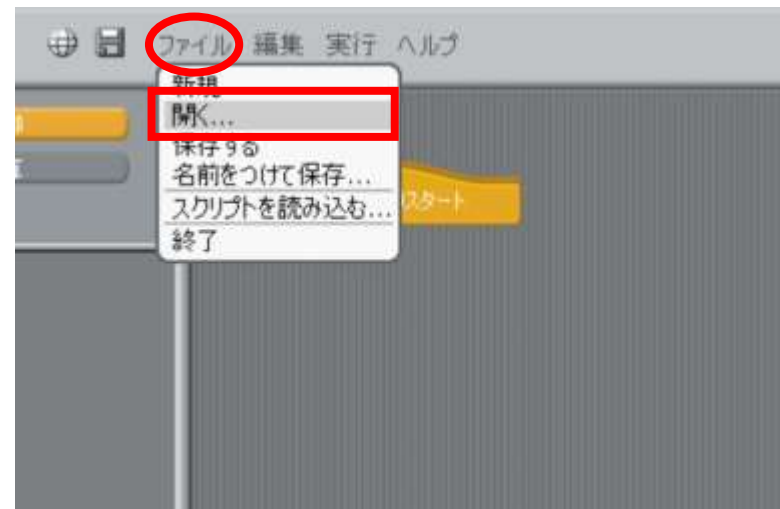
不要なブロックがあればブロックパレットに戻す



# プログラムをコンピュータに保存可能



ファイル⇒名前をつけて保存



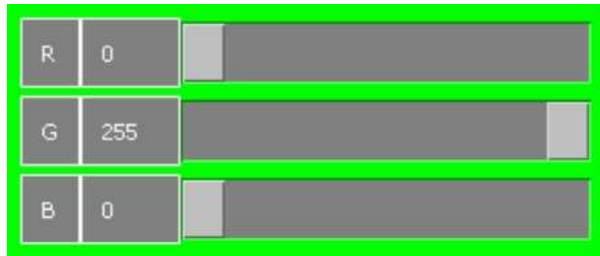
ファイル⇒開く

保存してすることで再利用が可能

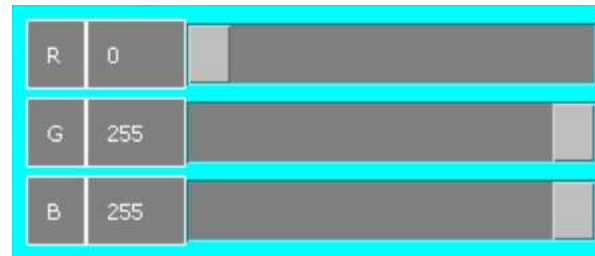
# 主な「動きブロック」

 <p>The image shows a Scratch 'Turn on LED' block with a small LED icon. Below the block is a screenshot of the color picker interface. The interface has a title '本体のLEDを ■ で点灯する' and three rows of sliders for Red (R), Green (G), and Blue (B). The R slider is set to 255, G to 0, and B to 0. A red rectangle highlights the R, G, and B rows.</p>	<p>色の部分をクリックし、表示されたスライダーでRGBの値を調整してLEDを点灯</p>
 <p>The image shows a Scratch 'Turn on LED' block with a small LED icon. The text inside the block is '本体のLEDを 赤: 255 緑: 255 青: 255 で点灯する'.</p>	<p>RGBの値を指定してLEDを点灯</p>
 <p>The image shows a Scratch 'Turn off LED' block with a small LED icon. The text inside the block is '本体のLEDを消灯する'.</p>	<p>本体のLEDが点灯していれば消灯</p>

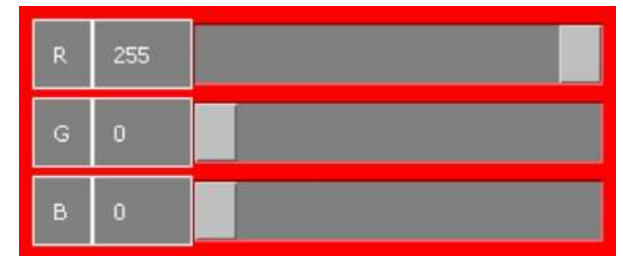
# RGBの値でLEDの色を設定



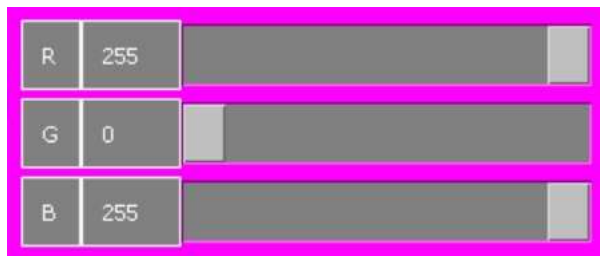
明るい緑



水色



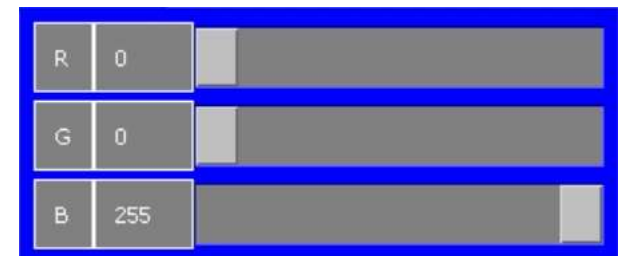
赤



ピンク



黄



青

※ 設定画面の色の通りにLEDが光らない色があります

# 主な「制御ブロック」

	指定した秒数を待つ		条件が真のときに中のブロックを処理
	中のブロックの処理をずっと繰り返す		条件が真の場合は「もしなら」の直下のブロックを処理 偽の場合は「でなければ」の直下のブロックを処理
	中のブロックの処理を指定した回数繰り返す		

# その他のブロック

	それぞれの値を示す
	関係式を満たした場合に真を返す
	両方の条件が真の場合に真を返す
	どちらかの条件が真の場合に真を返す
	条件が偽の場合に真を返す

# 実習 LEDを〇〇色に点灯させる

【動き】 ブロック

