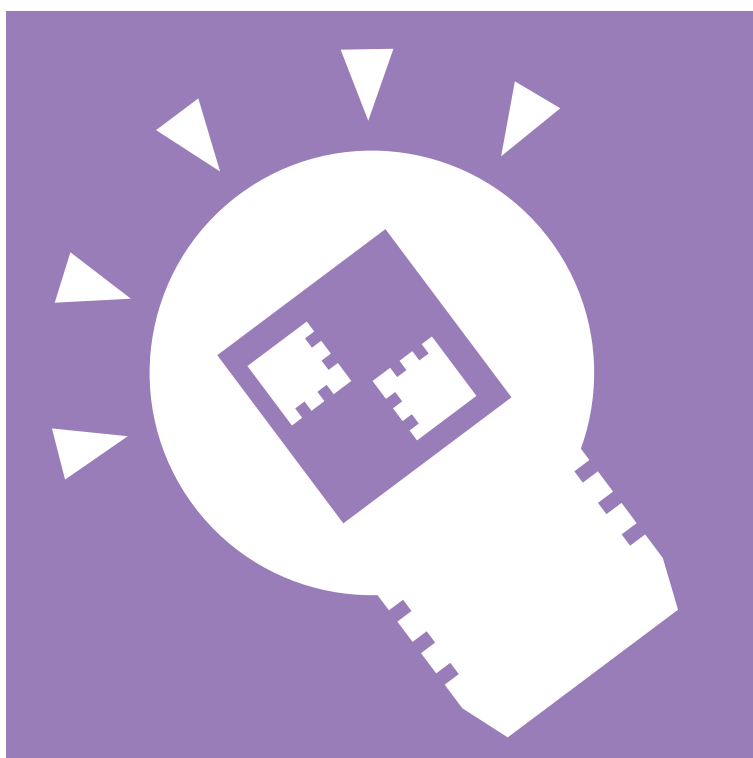


小学校理科 電気の利用 プログラミングガイド

応用編

教師用



- 1章 コンピューターとプログラム
- 2章 プログラムによる電気の流れの制御
- 3章 夜間ライトの制作
- 4章 感温扇風機 of 制作
- 5章 非常灯の制作

はじめに

■ テキストの使いかた

このテキストでは、理科6年生「電気の利用」の単元に沿って、プログラムによる制御について学びます。はじめに1章と2章でプログラミングの基本について学習し、その後は授業で取り上げたいテーマに応じて章を選んで実施してください。

■ 目次

1章 コンピューターとプログラム	2
身のまわりにあるコンピューターの役割や、それを動かすプログラムについて学びます。	
2章 プログラムによる電気の流れの制御	10
順次処理のプログラムをつかって、豆電球の点灯・消灯を制御します。	
3章 夜間ライトの制作	17
光センサーを使用して、周囲が暗くなると点灯するライトをつくり、電気の効率的な利用について学びます。	
4章 感温扇風機の制作	24
温度センサーを使用して、周囲の温度によってONとOFFを切りかえる扇風機をつくり ます。	
5章 非常灯の制作	31
加速度センサーを使用して、揺れを検知すると点灯するライトをつくりま	

※スタディーノビットを使用しない場合は、スタディーノビットを電池ボックスから外すようにしてください。
ボタンの押下の確認に電力を消耗し、一週間ほどで電池切れになります。

※本テキストに記載されている製品やサービスの名称は、各社の商標または登録商標です。

・Windowsは、米国およびその他の国で登録されたMicrosoft Corporationの商標です。

・Macは、米国およびその他の国で登録されたApple Inc.の商標です。

・iOSは、Apple Inc.のOS名称です。IOSは、Cisco Systems, Inc.またはその関連会社の米国およびその他の国における商標または登録商標であり、ライセンスに基づき使用されています。

・Apple Storeは、Apple Inc.のサービスマークです。

・Android、Chromebook、Google playは、Google LLCの商標です。

・ArtecRobo2.0およびStuduino:bitは、株式会社アーテックの商標です。

1章 コンピューターとプログラム

身のまわりにあるコンピューターの役割や、このテキストで取り組むプログラミングの方法について学びます。

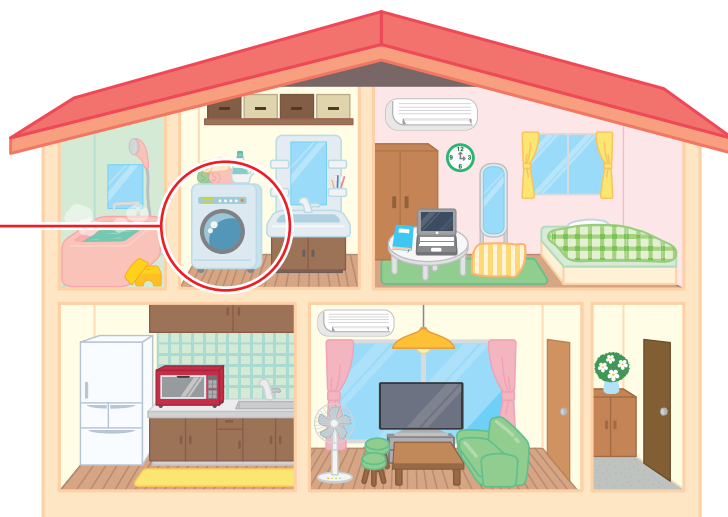
1-1. コンピューターを使った身近な道具

くらしのなかで利用している電気を使った道具のなかには、コンピューターが入っているものたくさんあり、そのような道具は人にかわって自動で働くことができます。



れい じどうかいさつき
例1) 自動改札機

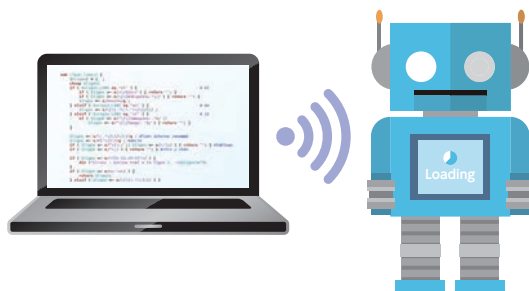
れい せんたくき
例2) 洗濯機



つか 使われている道具	じどう 自動でどのような働きをしているか
れい じどうかいさつき 例1) 自動改札機	きつぷ 切符やICカードから読み取った情報に応じて、ゲート ひら を開いたり閉じたりする。
れい せんたくき 例2) 洗濯機	お じどう せんたくもの りょう しら みず そそ ボタンを押すと自動で洗濯物の量を調べて水を注ぐ。

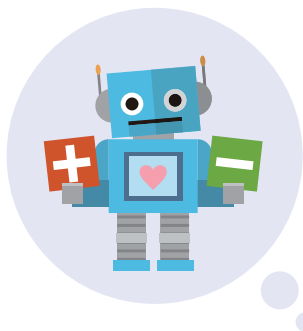
1-2. コンピューターを動かすためのプログラム

コンピューターを動かすためには「プログラム」が必要です。プログラムはコンピューターが働く手順を書いたもので、コンピューターはプログラムを読んで、その通りに動いています。

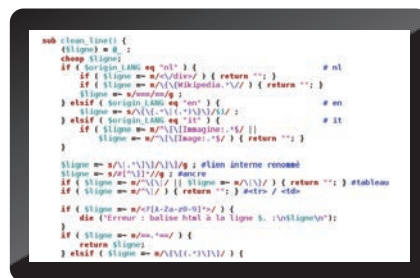


プログラムは人がコンピューターにさせたいことを考えて、そのために必要な手順を整理してつくりまします。プログラムをつくることを「プログラミング」といい、コンピューターが理解できる「プログラミング言語」という言葉を使って書きます。

コンピューターにどんなことをさせるのか整理する。



整理した手順をもとにプログラミングしてプログラムをつくる。



プログラミング言語で書かれたプログラム



プログラミング言語は主に「テキスト言語」と「ビジュアル言語」の2つに分けることができます。ここでは、より表現の平易なビジュアル言語を使ってプログラミングをつくりまします。

テキスト言語

文字で記述してコンピューターへの命令をつくる。

```
def thread_0():
    img = Image('11111:11111:11111:11111:11111')
    img.set_base_color((31,0,0))
    display.show(img, delay = 0)
    time.sleep(1)
    display.clear()

def start():
    thread_0()
```

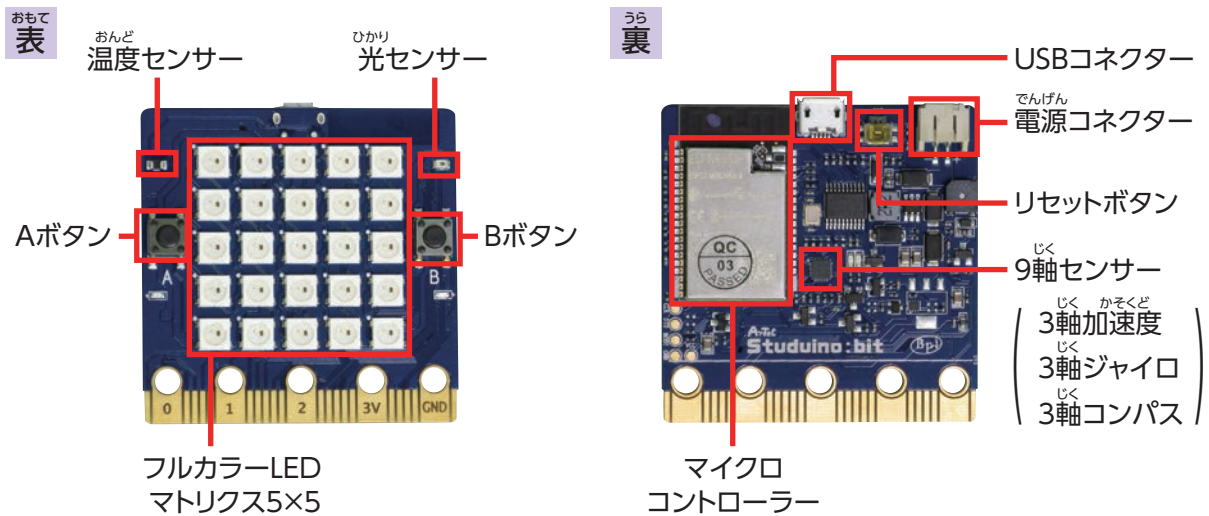
ビジュアル言語

アイコンなどを組み合わせてコンピューターへの命令をつくる。



1-3. Studuino:bit(スタディーノビット)について

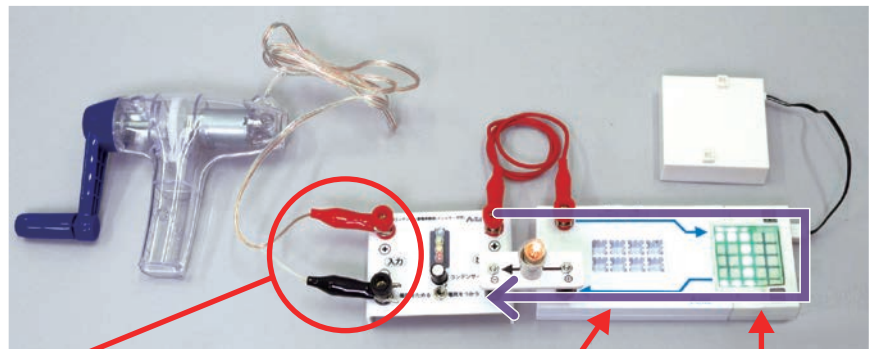
今回プログラムで動作させるのは「Studuino:bit」(スタディーノビット)というコンピューターです。カバーを外すと以下のようにになっています。



スタディーノビットは「電気の利用拡張ボックス」と合わせて回路の中に組み込みます。これらがスイッチの役割を果たし、プログラムでONとOFFを命令することで電気の流れを制御できます。

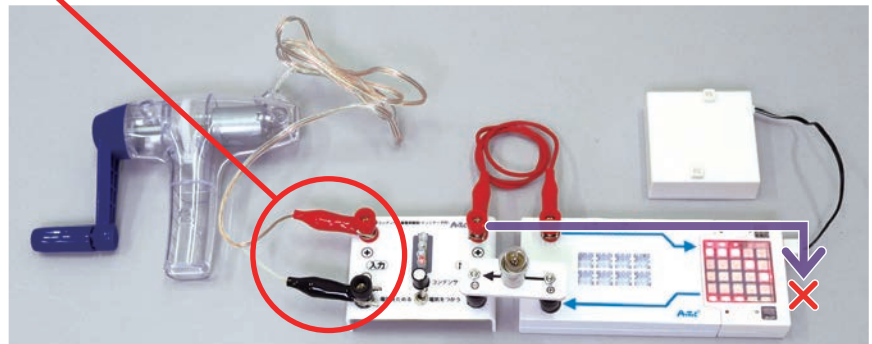
スイッチがONのとき

拡張ボックスの赤色のコネクタに電源の+を、黒色のコネクタに電源の-を接続してください。
接続を間違えると、正常に動作しません。



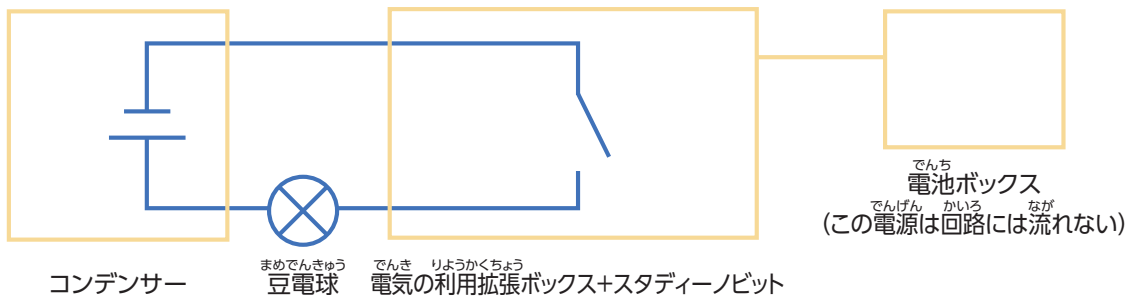
電気の利用拡張ボックス スタディーノビット

スイッチがOFFのとき

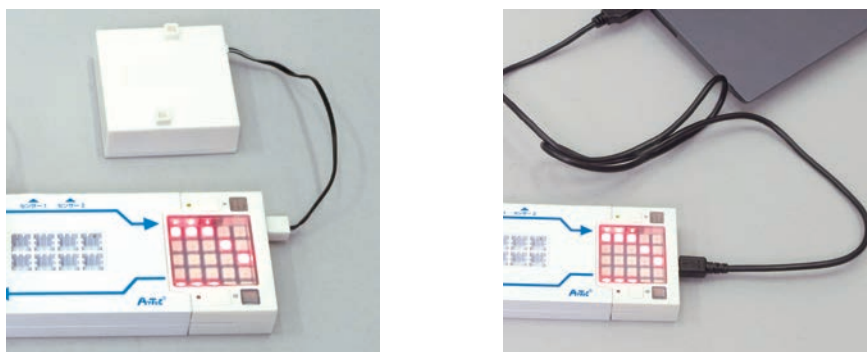


拡張ボックスにかかれた矢印は電気の流れをあらわしています。スイッチをONにすると、スタディーノビットのLEDが緑色に点灯し、矢印がつながって電気が通ります。スイッチをOFFにすると、LEDが赤色に点灯して電気の流れが止まります。

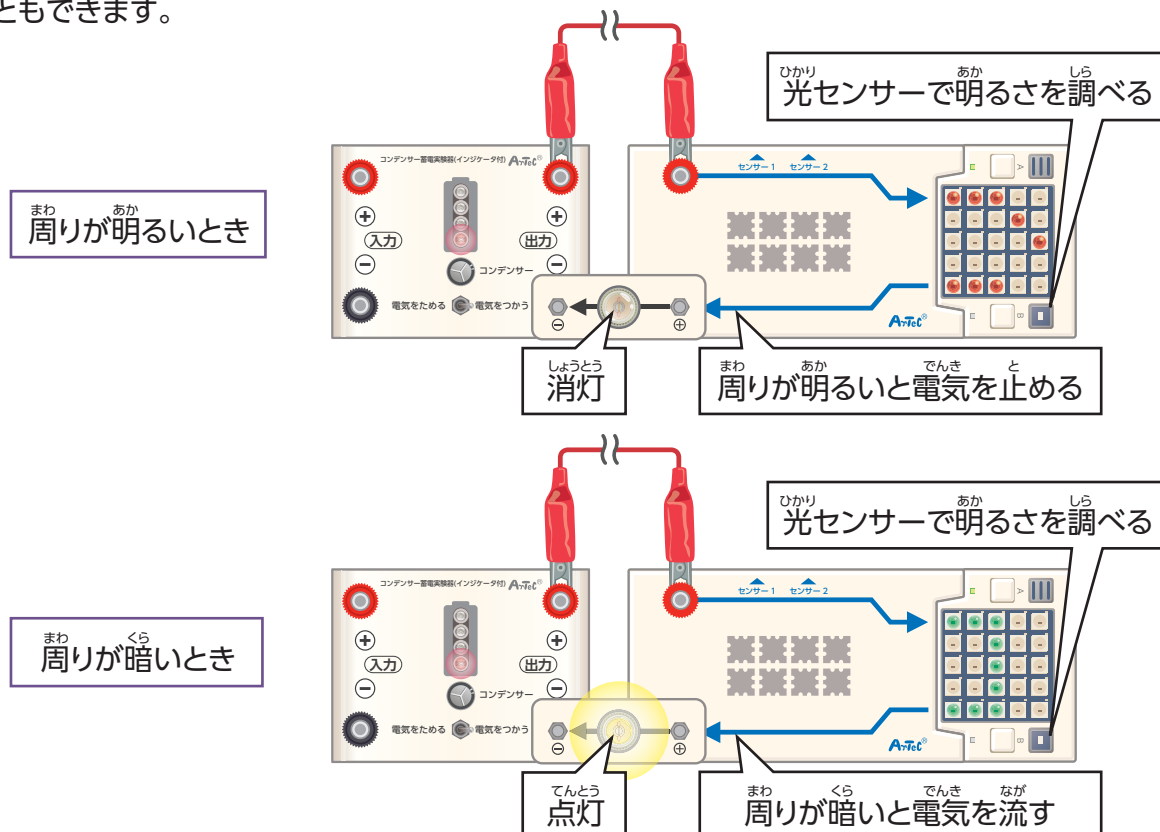
これを回路図にあらわすと以下ようになります。



手回し発電機で発電し、コンデンサーにためた電気がこの回路の電源になります。ただし、スイッチのONとOFFを切りかえる(スタディーノビットを動作させる)ための電源として、電池ボックスまたはパソコンをスタディーノビットにつないでおく必要があります。



また、センサーを利用することで、周りの状況に応じて自動で判断して、電気の流れを制御することもできます。



1-4. ソフトウェアについて

プログラミングには専用のソフトウェアを使います。このテキストではStuduino:bit Software (スタディーノビットソフトウェア)を使用してプログラムをつくります。

1-4-1. ソフトウェアのインストール

お使いの端末のOS (オペレーションシステム)を確認し、ソフトウェアをインストールしてください。

● Windowsの場合

以下のウェブサイトアクセスし、setupファイルをダウンロードします。



<https://www.artec-kk.co.jp/artecrobo2/ja/software/>

ダウンロードしたsetupファイルを実行して、「Studuino:bit Software」をインストールします。

● Macの場合

以下のウェブサイトアクセスし、ソフトウェアをダウンロードします。



<https://www.artec-kk.co.jp/artecrobo2/ja/software/>

ダウンロードしたファイル内の「Studinobit.app」をアプリケーションフォルダに移動してください。

● Android／ChromeBookの場合

Google Playで「Studuino:bit」と検索して、ソフトウェアをインストールします。

● iOSの場合

App Storeで「Studuino:bit」と検索して、ソフトウェアをインストールします。

インストールが完了すると、画面上に以下のようなアイコンが表示されます。



1-4-2. プログラム作成画面の準備

次の手順でプログラム作成画面を表示してください。

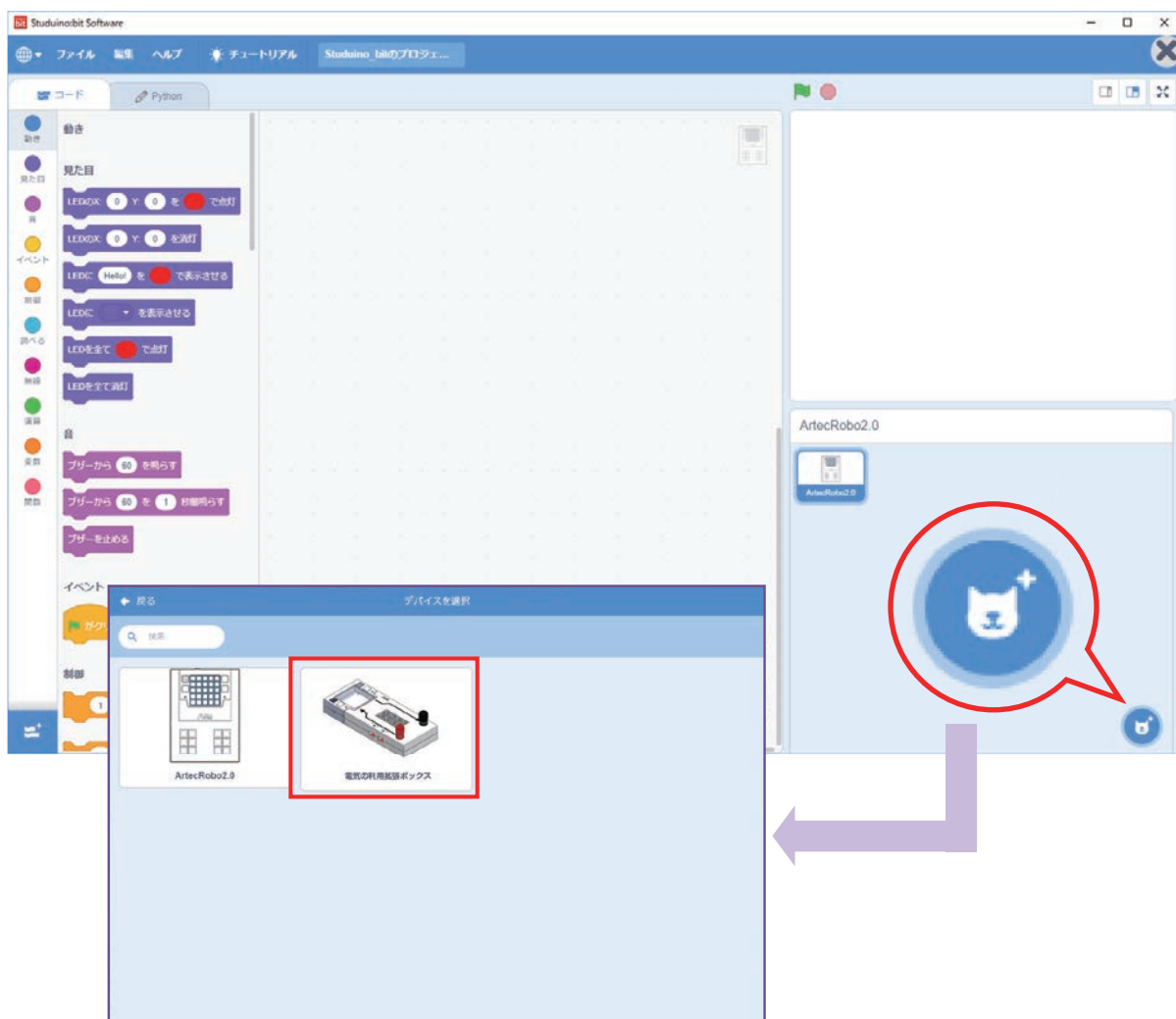
- 1 Studuino:bit Softwareのアイコンをクリックして、ソフトウェアを立ち上げます。



- 2 ソフトウェアが立ち上がったら、ロボットモードを選択します。



- 3 画面右下のスプライトエリアから、電気の利用拡張ボックスを選択してください。



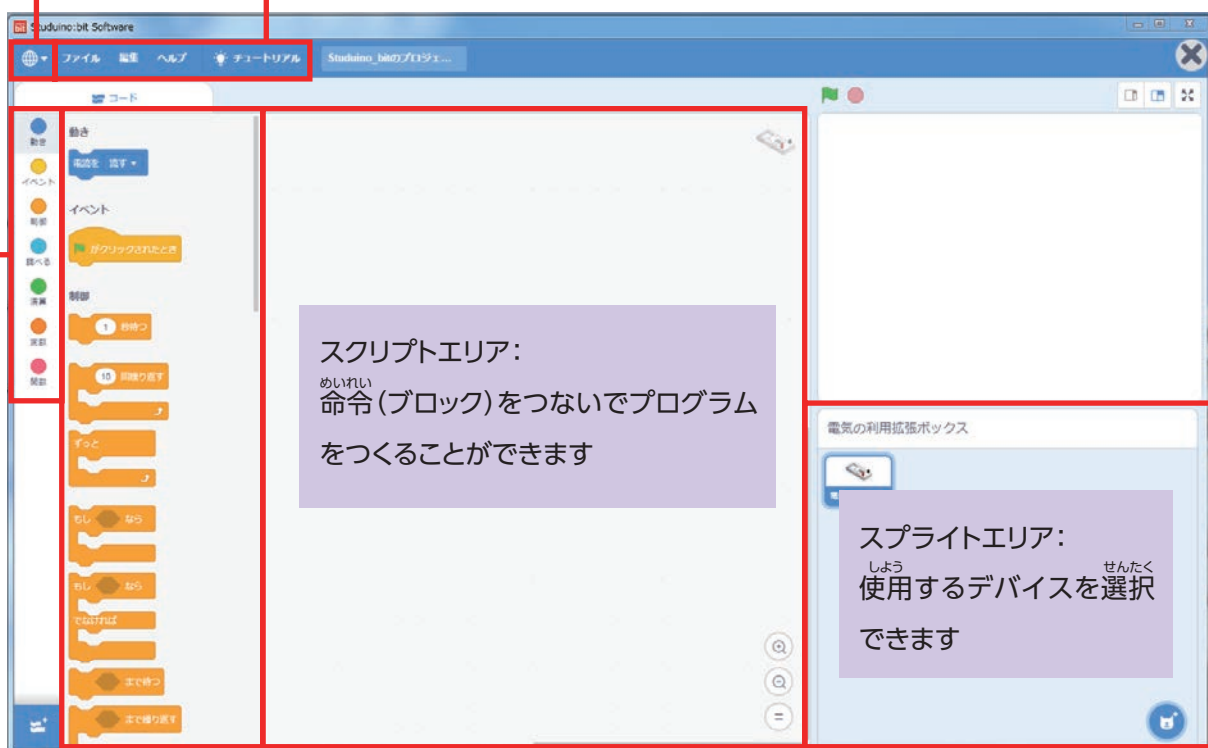
4 電気の利用拡張基板に対応した、プログラム作成画面があらわれます。

カテゴリ：命令(ブロック)の種類を選ぶことができます

(各カテゴリのアイコンをクリックすると、ブロックパレットがスクロールされて表示されるブロックが変わります)

言語の選択

メニュー



スクリプトエリア:

命令(ブロック)をつないでプログラム
をつくることができます

スプライトエリア:

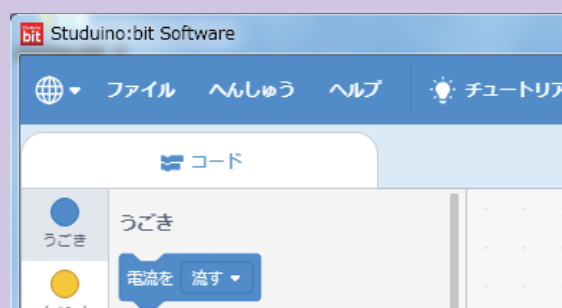
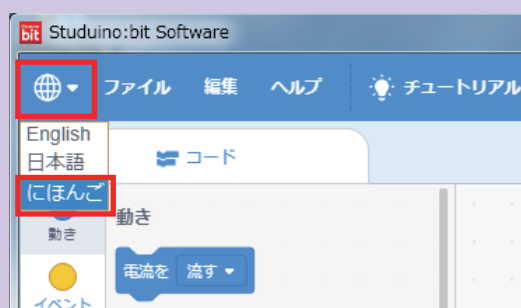
使用するデバイスを選択
できます

ブロックパレット: プログラムをつくるために必要な命令(ブロック)が表示されます

参考

言語の選択

言語を「にほんご」に設定すると、画面がひらがな表記になります。

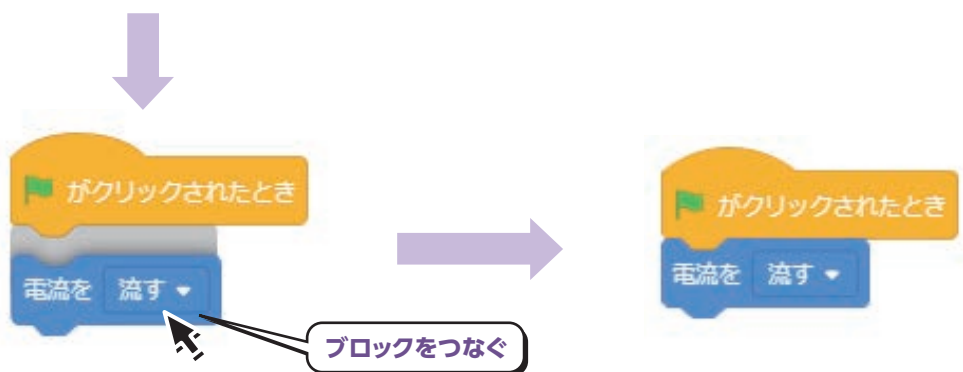


1-4-3. 操作方法

●プログラムの作成

ブロックパレットにある命令(ブロック)をスクリプトエリアに移動させてつなぐことでプログラムをつくれます。

❗電気の利用拡張ボックスを使用するときのプログラムは、必ず「がクリックされたとき」から始まるようにつくります。



●ブロックの削除

スクリプトエリアからブロックパレットにブロックを移動させると、削除することができます。

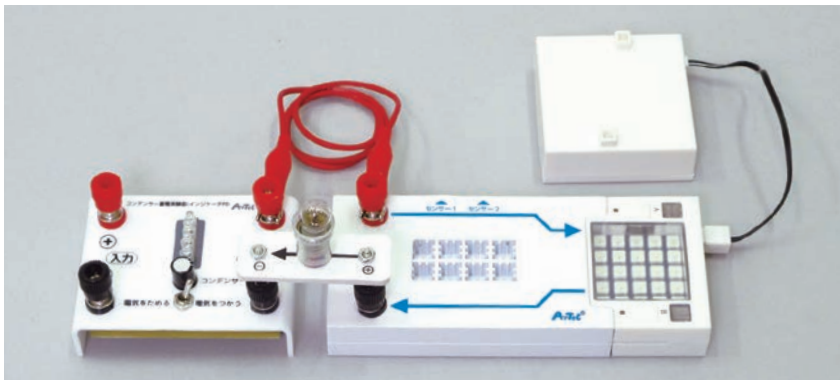


2章 プログラムによる電気の流れの制御

実際にコンピューターをプログラミングして、くらしの中で身近な明かりをコントロールします。
この章では豆電球の点灯と消灯を自動で切りかえるプログラムを作成し、スタディーノビットを使って回路に流れる電気が制御できることを確認します。

2-1. 実験準備

スタディーノビットとコンデンサー、豆電球を図のようにつなぎます。

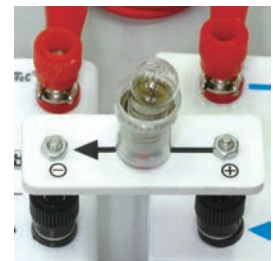


2-2. 一定時間点灯するプログラムの作成

下の3つのブロックを組み合わせ、豆電球を5秒間点灯させるプログラムをつくりま



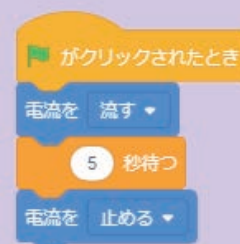
びょうご
5秒後
》



- | | | |
|------|------------|----------------------------|
| イベント | がクリックされたとき |プログラムを開始するブロック |
| 動き | 電流を 流す |電気の流しかたを制御するブロック |
| 制御 | 1 秒待つ |次の命令まで指定された時間だけ待つブロック |

プログラム例

ブロックの命令は上から順番に実行されます。



2-3. プログラムの転送^{てんそう}

つくったプログラムをスタディーノビットに送^{おく}ります。

Windows/Macの場合^{ばあい}

- ① USBケーブルでスタディーノビットとパソコンをつなぎます。



- ② メニューの「編集^{へんしゅう}」から「転送^{てん}」を選択します。



- ③ プログラムの転送先を0～9のいずれかから選択^{せんたく}してください。プログラム名もここで変更^{へんこう}できます。



- ❗ スタディーノビットは、最大10個までプログラムを書き込むことができます。

- ④ プログラムは転送完了後、スタディーノビット上で自動的に実行^{じっこう}されます。

- ❗ ただし、コンデンサーからの電源の供給が無ければ豆電球は点灯しません。13ページに進み、動作の確認をしてください。

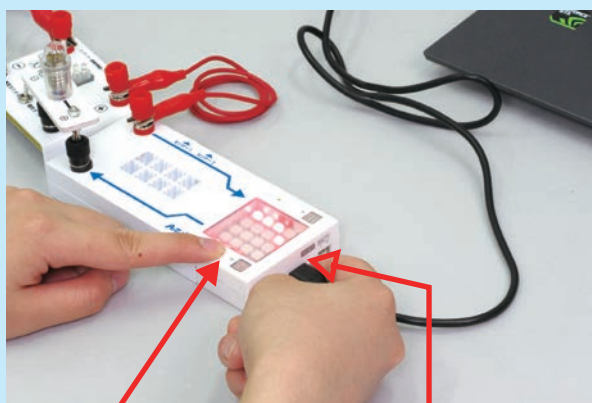
他の番号に転送したプログラムを実行する場合は、16ページを参照してください

Android(Chromebook)/iOSの場合^{ばあい}

- ① メニューの「編集」から「転送」を選択します。^{へんしゅう てんそう せんたく}



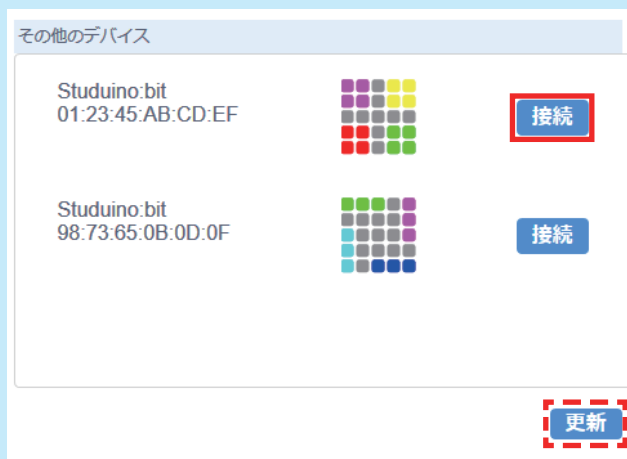
- ② 下のような画面が表示されたら、スタディーノビットのBボタンを長押ししながらリセットボタンを1回押してください。^{した がめん ひょうじ}



Bボタン

リセットボタン

- ③ スタディーノビットのLEDが点灯したらBボタンを離します。^{てんとう はな}
画面の中から同じ点灯パターンのデバイスを探し、「接続」を選択してください。^{がめん なか おな てんとう さが せつぞく せんたく}



- ❗ スタディーノビットと同じ点灯パターンが画面に表示されない場合は、「更新」を押してください。^{おな てんとう がめん ひょうじ ばあい こうしん お}

- ④ プログラムの転送先を0～9のいずれかから選択してください。
プログラム名もここで変更できます。



❖ スタディーノビットは、最大10個までプログラムを書き込むことができます。

- ⑤ プログラムは転送完了後、スタディーノビット上で自動的に実行されます。
ただし、コンデンサーからの電源の供給が無ければ豆電球は点灯しません。このあとの動作の確認をしてください。

❖ 転送が成功した場合は、LEDの右上が青く点灯します。失敗した場合は白く点灯するので、リセットボタンを押して再度転送してください。

他の番号に転送したプログラムを実行する場合は、16ページを参照してください

2-4. 動作の確認

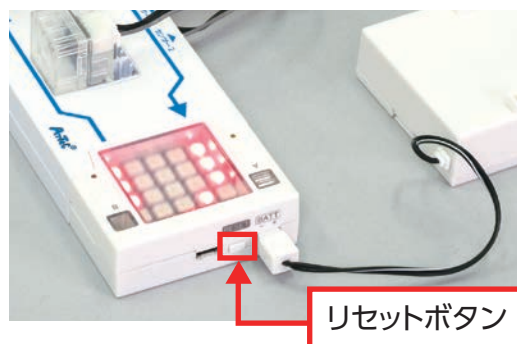
コンデンサーにためた電気を使って、豆電球がプログラム通りに点灯するかを確認します。

- ① 手回し発電機のハンドルを回して電気をつくり、コンデンサーに蓄えます。

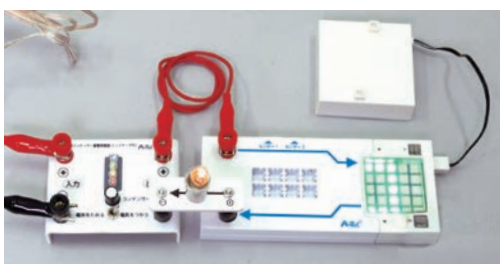


❖ 蓄電するときはスイッチを「電気をためる」にします。

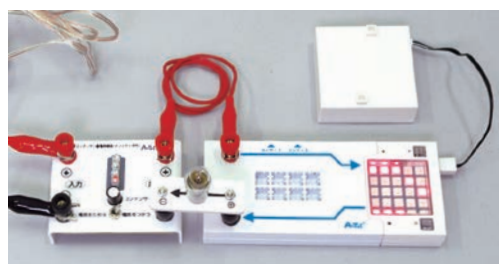
- ② コンデンサーのスイッチを「電気をつかう」にして、回路に電気を流します。スタディーノビットのリセットボタンを押してプログラムを始めから実行し、豆電球が5秒間点灯することを確認します。



❖ リセットボタンを押してからプログラムが実行されるまで、2秒ほど時間がかかります。



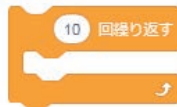
5秒後
》



2-5. 点滅を繰り返すプログラムの作成

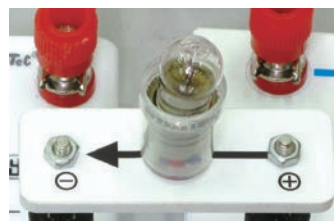
身近で光を制御して動いているものの例として、信号機が挙げられます。ここでは、歩行者用信号機のように光を点滅させる動作をプログラミングします。

10ページで使ったブロックに、「制御」カテゴリの



を追加して、豆電球が1秒間

隔の点滅を5回繰り返すプログラムを考えます。



1回だけ点滅する動作は右のプログラムでつくることができます。しかし、このプログラムを5回繰り返しても、豆電球はうまく点滅しません。

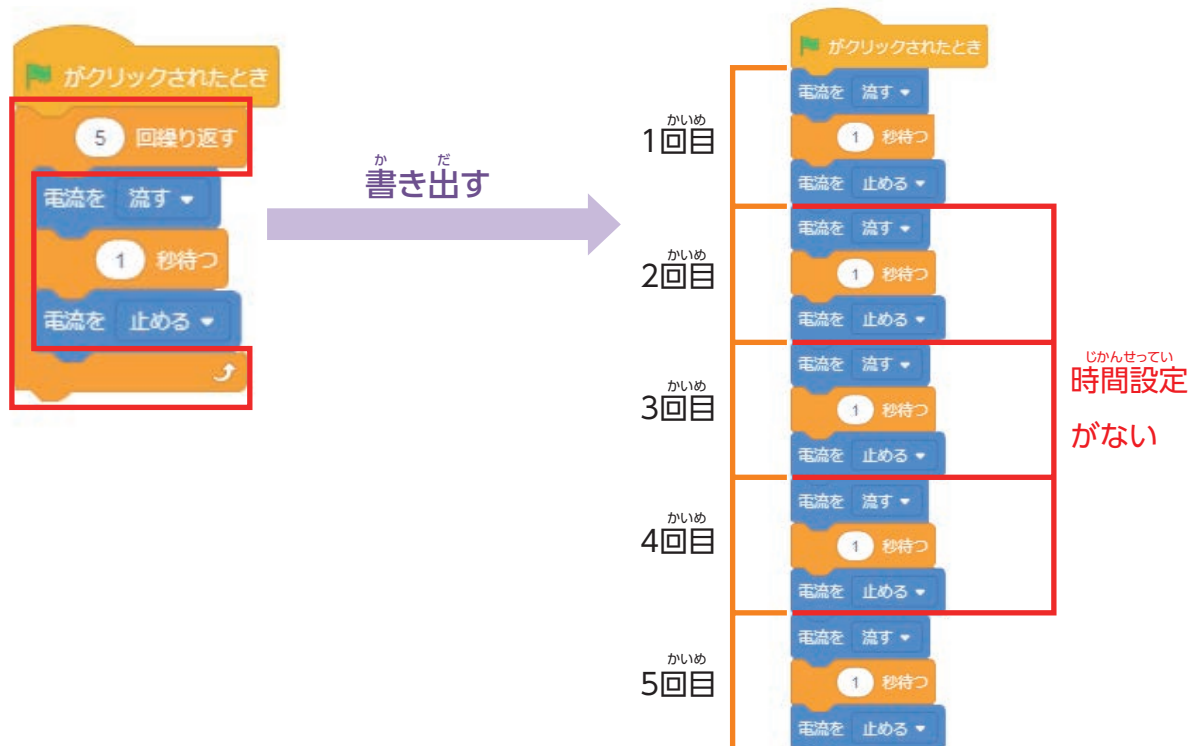


下の図のように



を使わずにプログラムを書き出してみると、その理由がわかります。

コンピューターはプログラムを高速で処理しています。そのため、「電流を止める」の後に時間設定がないと、すぐに「電流を流す」の命令が実行されてしまい、点滅になりません。



そのため、「電流を止める」の後にも **1 秒待つ** を入れてプログラムをつくと、豆電球が正常に点滅を繰り返します。

プログラム例



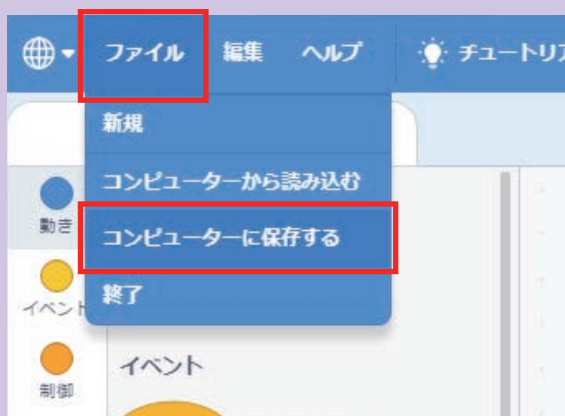
5 回繰り返す
の代わりに
ずっと
を使うと、同じ
命令を何度も繰り返すこと
ができます。



参考

プログラムの保存方法

作成したプログラムをパソコンやタブレットに保存したい場合は、メニューの「ファイル」から「コンピューターに保存する」を選択してください。



以下の方法で転送済みのプログラムの番号を指定して実行することができます。

① 電源を接続する(通常起動)

- 1) 電池ボックスもしくはUSBケーブルをスタディーノビットに接続します。
- 2) 電源ランプが点灯し、電源が自動的にONになり起動します。通常起動時は前に実行したプログラムが自動的に実行されます。



電源ランプ

② プログラム選択モードへの移行

- 1) Aボタンを長押ししたまま、リセットボタンを押してください。再起動のため一時的に電源ランプが消灯します。
- 2) リセットボタンを離して3秒後に電源ランプが再点灯したら、Aボタンを離します。
- 3) スタディーノビットに0が表示されると、プログラム選択モードへ移行しています。



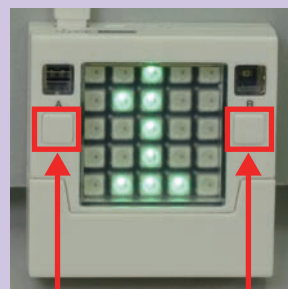
リセットボタン

Aボタン



③ プログラムの選択・起動

- 1) プログラム選択モード時にAボタンを押すと、表示が0,1,2...と切り替わります。9まで切り替わると0に戻ります。
- 2) 選択した番号でBボタンを押すと、各番号に割り当てられたプログラムが実行されます。



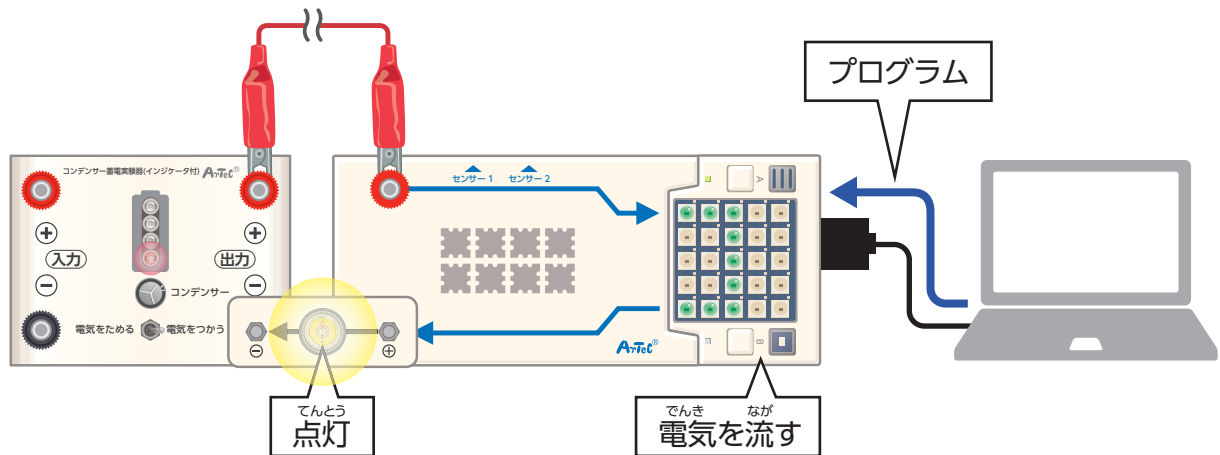
番号の選択

決定(実行)

- ❗ 初期状態では、各番号にサンプルプログラムが書き込まれています。
- ❗ サンプルプログラムは新たなプログラムを転送すると上書きされます。

3章 夜間ライトの制作

コンピューターを使うと、プログラムによって電気の流れを制御することができました。



さらに**センサー**を追加することで、周りの情報を調べて、ある状況になったときに自動で電気を流すプログラムをつくることができます。この章では光センサーを利用して、暗くなると自動で点灯する夜間ライトをつくります。

まわあかしょうとう
周りが明るいときライトは消灯

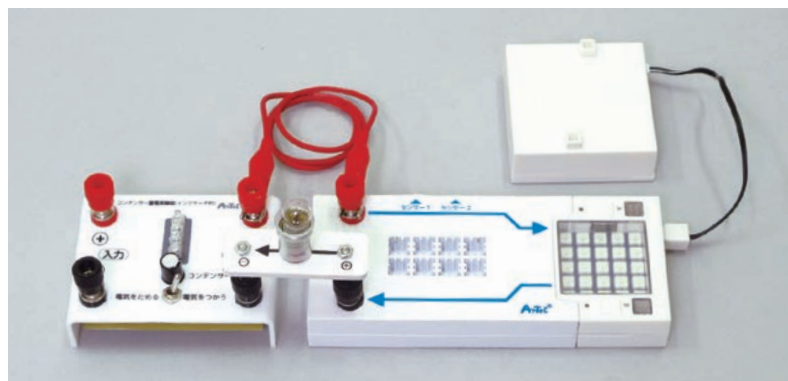


まわくらてんとう
周りが暗いときライトが点灯



3-1. 実験準備

スタディーノビットとコンデンサー、豆電球を図のようにつなぎます。



3-2. 光センサーの使いかた

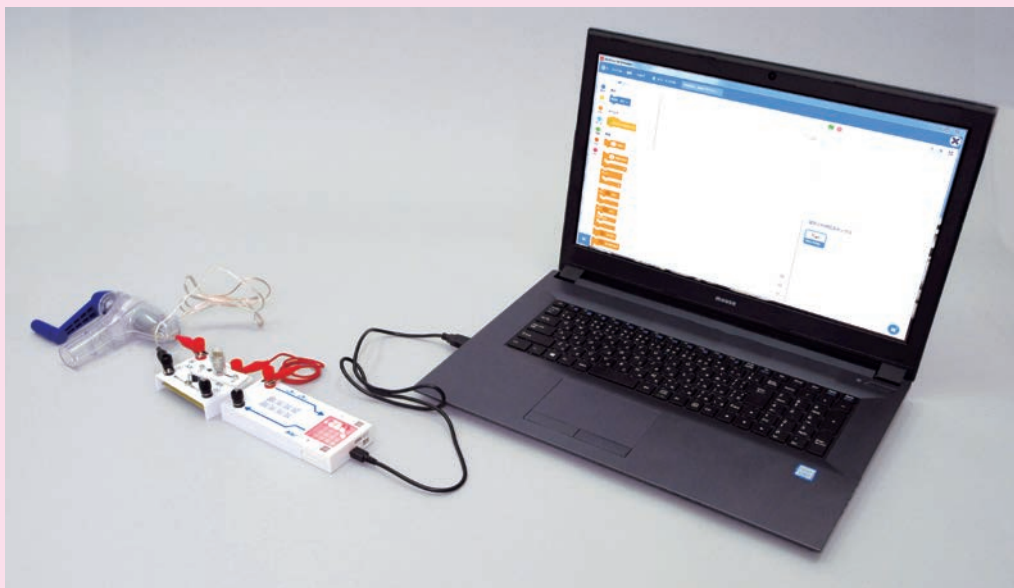
スタディーノビットは光センサーを内蔵しています。
このセンサーを使用すると、周りの明るさを調べることができます。



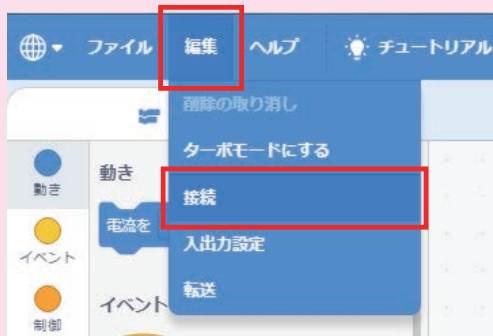
センサーが調べた情報は数値であらわされます。この数値は、スタディーノビットを通信状態にすることで、ソフトウェアから確認することができます。

Windows/Macの場合 (USBケーブルによる通信)

- 1 USBケーブルでスタディーノビットとパソコンをつなぎます。



- 2 メニューの「編集」から「接続」を選択すると、画面右上にセンサーボードが表示されます。



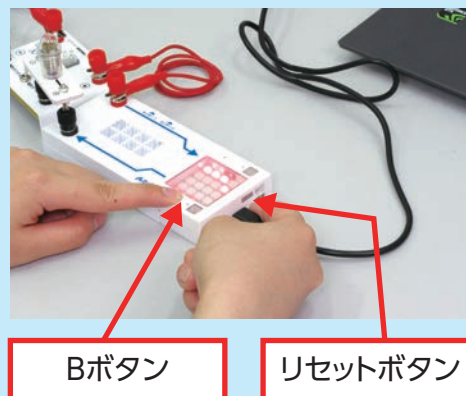
センサーボード	
Studuino:bit	
ボタンA	1
ボタンB	1
光センサー	20
温度センサー	24.53
加速度センサー X	-0.55
加速度センサー Y	0.81
加速度センサー Z	-0.03
ジャイロセンサー X	0
ジャイロセンサー Y	-4
ジャイロセンサー Z	2
磁気センサー X	-49
磁気センサー Y	-54
磁気センサー Z	4
電気の利用拡張ボックス	

Android(Chromebook)/iOSの場合 (Bluetoothによる通信)

- ① メニューの「編集」から「接続」を選択します。



- ② 下のような画面が表示されたら、スタディーノビットのBボタンを長押ししながらリセットボタンを1回押してください。



- ③ スタディーノビットのLEDが点灯したらBボタンを離します。
画面の中から同じ点灯パターンのデバイスを探し、「接続」を選択してください。



- ❗ スタディーノビットと
同じ点灯パターンが画
面に表示されない場
合は、「更新」を押して
ください。

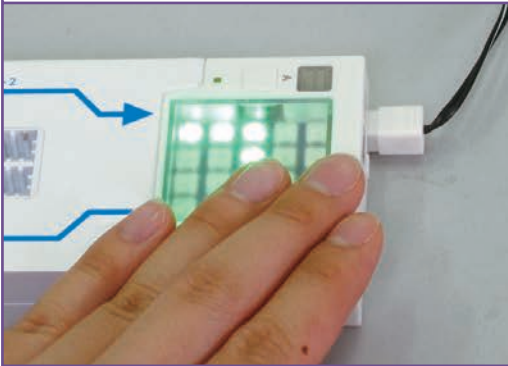
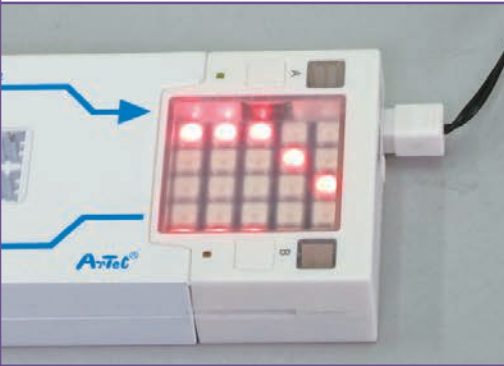
- ④ スタディーノビットの通
信ランプが青色に点灯
し、画面上にセンサー
ボードが表示されます。



センサーボード	
Studino bit	
ボタンA	1
ボタンB	1
光センサー	20
温度センサー	24.53
加速度センサー X	-0.55
加速度センサー Y	0.81
加速度センサー Z	-0.03
ジャイロセンサー X	0
ジャイロセンサー Y	-4
ジャイロセンサー Z	2
磁気センサー X	-49
磁気センサー Y	-54
磁気センサー Z	4
電気の利用拡張ボックス	

まわ^{まわ}が^{あか}明るいときと^{くら}暗いときとで、センサーボードの「光センサー」の^{あたい}値がどのように^{へんか}変化するかを調べます。

センサーボード	
Studuino:bit	
ボタンA	1
ボタンB	1
光センサー	20
温度センサー	24.53

まわ ^{まわ} の ^{じょうたい} 状態	くら ^{くら} 暗いとき(手でかくしたとき)	あか ^{あか} 明るいとき(手でかくさないとき)
	 <p>およそ0~10</p>	 <p>およそ30~70</p>
センサーの ^{あたい} 値		

❗ さえぎ^{さえぎ} じょうたい^{じょうたい} しょうめい^{しょうめい} つよ^{つよ} あたい^{あたい} か^か
遮るものがない状態であっても、照明の強さなどによって値は変わります。

ひかり^{ひかり} あたい^{あたい} へんか^{へんか} 光センサーの値の変化

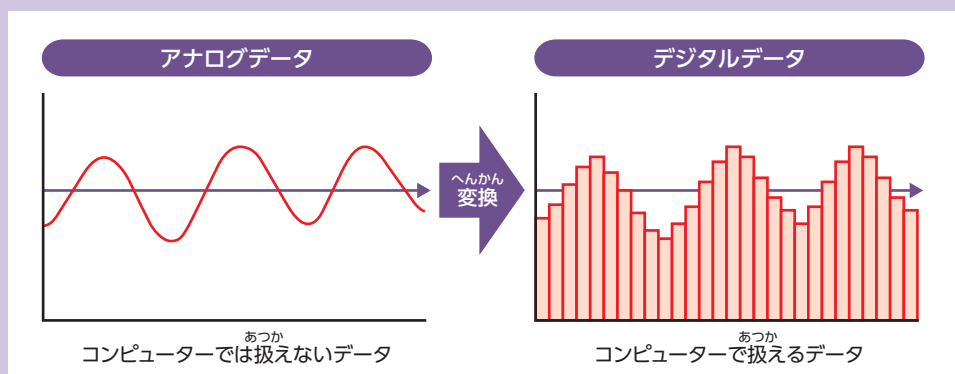
まわ^{まわ} くら^{くら}暗いとき → ひかり^{ひかり} あたい^{あたい} ちい^{ちい}
光センサーの値は小さくなる

まわ^{まわ} あか^{あか}明るいとき → ひかり^{ひかり} あたい^{あたい} おお^{おお}
光センサーの値は大きくなる

さんごう^{さんごう} 参考

じょうほう^{じょうほう} あつか^{あつか} センサー情報の扱いかた

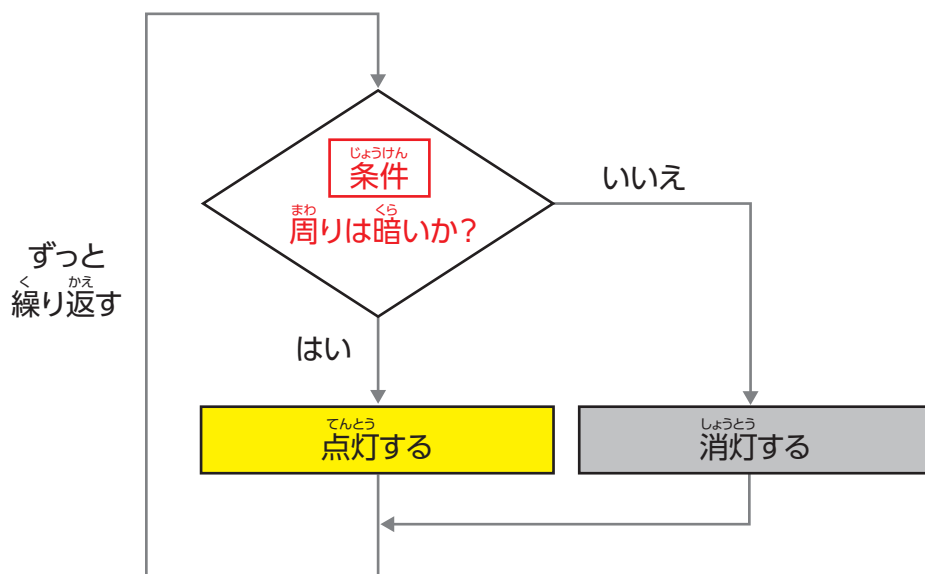
コンピューターでは、あか^{あか}明るさなどのアナログ^{じょうほう}情報を、コンピューターが^{しり}処理するデジタル^{じょう}情報に変換^{へんかん}して扱っています。



たと^{たと}例えば、スタディーノビットソフトウェアでは、ひかり^{ひかり} けんち^{けんち} あか^{あか}
光センサーが検知した明るさを0~100のデ^ち
ジタル値であらわしています。

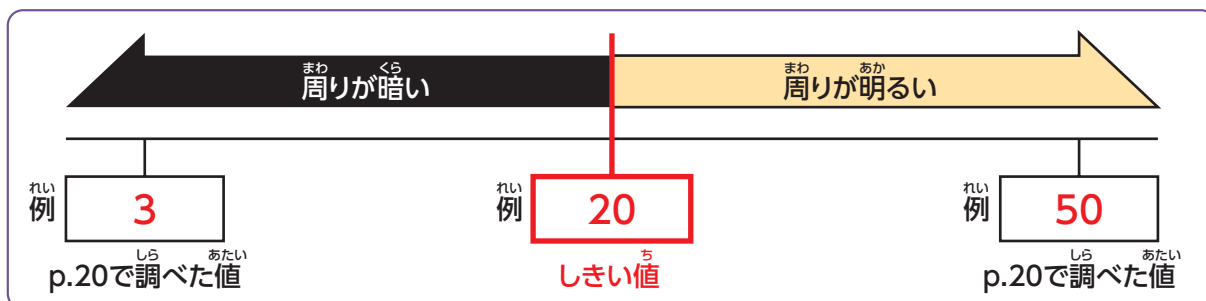
3-3. センサーの値を利用したプログラムの作成

光センサーの値を使って、周りが暗いときは豆電球が点灯し、明るいときには消灯するという動作をさせるためのプログラムを考えます。この動作をフローチャートであらわすと以下のようになります。



3-3-1. しきい値の設定

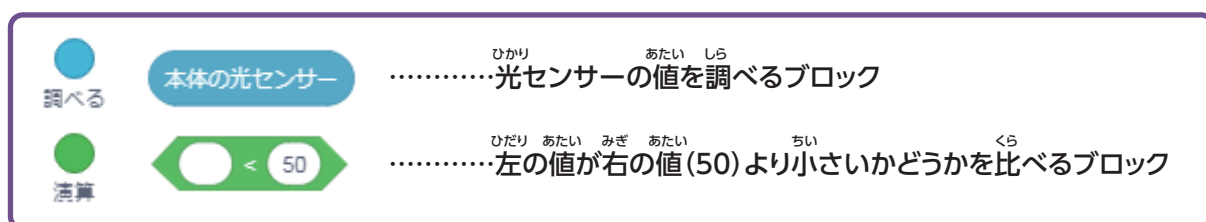
20ページで調べたセンサーの値から、「周囲が暗い」と判断するための基準となる値を決めます。この値のことを「しきい値」といいます。



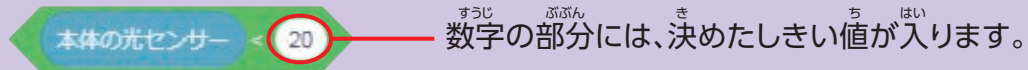
光センサーの値は、周囲が暗くなるほど小さくなります。よって、「周囲は暗いか?」という条件は、「光センサーの値がしきい値よりも小さいか?」と言い換えることができます。

3-3-2. 条件によって動作を分けるプログラムの作成

「光センサーの値がしきい値よりも小さい」という条件を、下のブロックを使ってつくります。



条件の例

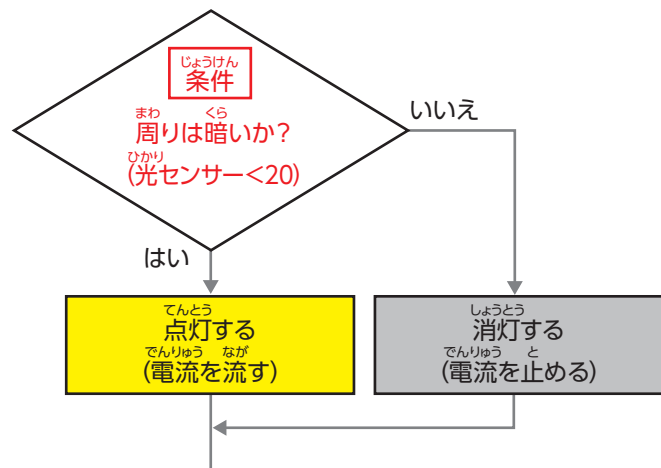


このブロックを使い、21ページのフローチャートの通りに動作する夜間ライトのプログラムを考えます。

「制御」カテゴリの



を使うと、条件によって動作を2つに分けることができます。



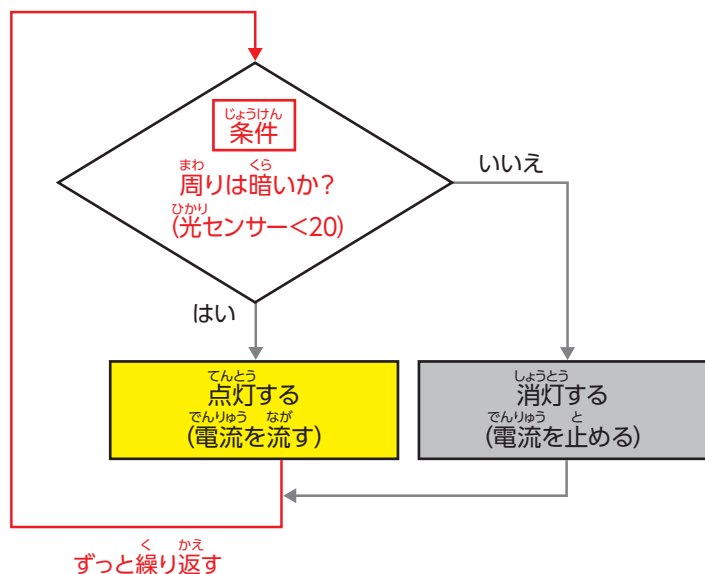
しかし、上のプログラムではうまく動作しません。コンピューターは非常に高速で処理をするため、実行した瞬間に光センサーの値を確認するプログラムが終了してしまうからです。

そのため、プログラムを



で囲むと、光センサーの値を常に確認することができるようになります。

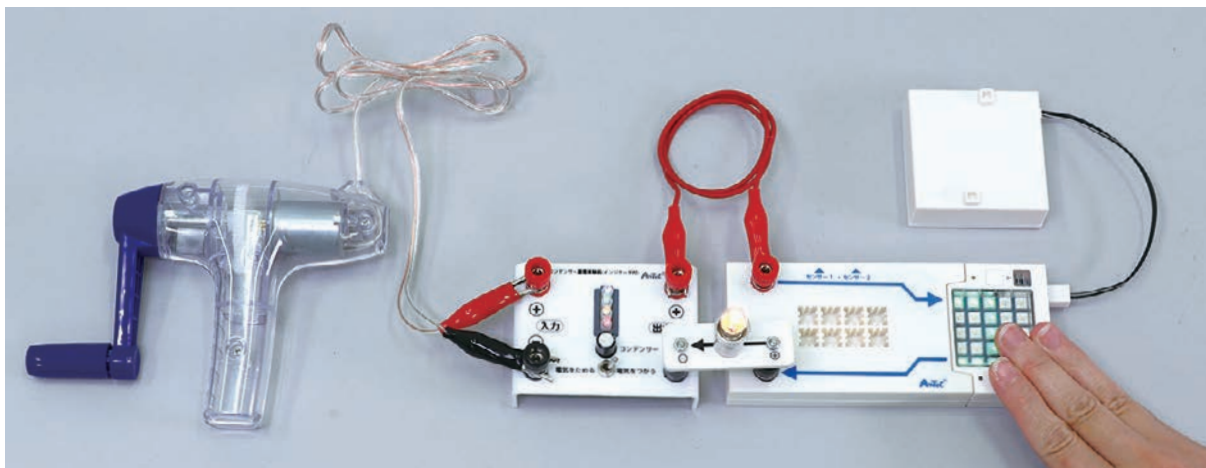
プログラム例



3-3-3. プログラムの実行

完成したプログラムをスタディーノビットに転送してコンデンサーから電気を送り、動作を確認します。

❗ 光センサーの値はスタディーノビットのLEDの光からも影響を受けるため、光センサーを手で隠しても値がそれほど小さくなりません。そのためしきい値が低すぎると、暗いと判断することができません。実際に手をかざして値を確かめながらしきい値を調整してください。



3-4. 電気の効率的な利用

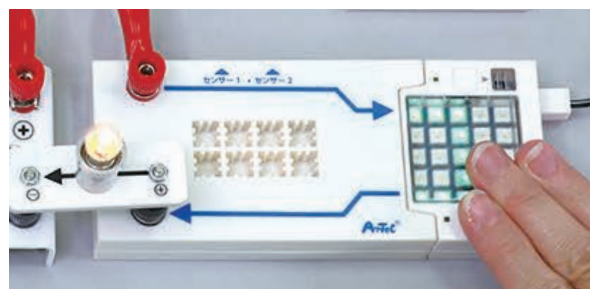
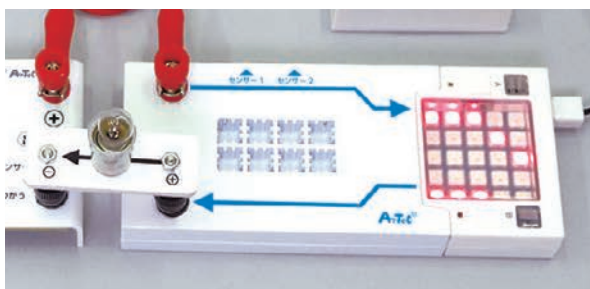
発電機とコンデンサーを使うと、自分たちでつくった電気で豆電球を点灯させることができます。しかし、電気は使い続けていると少しずつ減っていきます。

夜間ライトのように、センサーを利用して必要な時にだけ電気をを使うようにすることで、電気を無駄にせず、効率的に利用することができます。

まわが明るいときライトは消灯

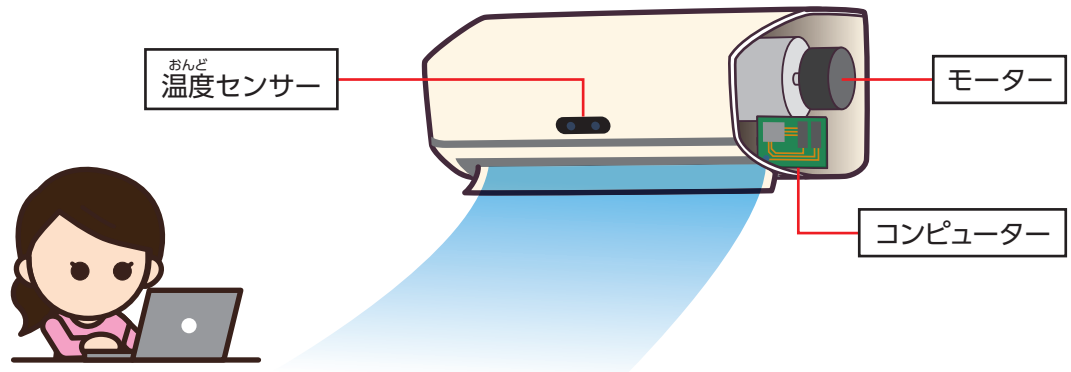


まわが暗いときライトが点灯

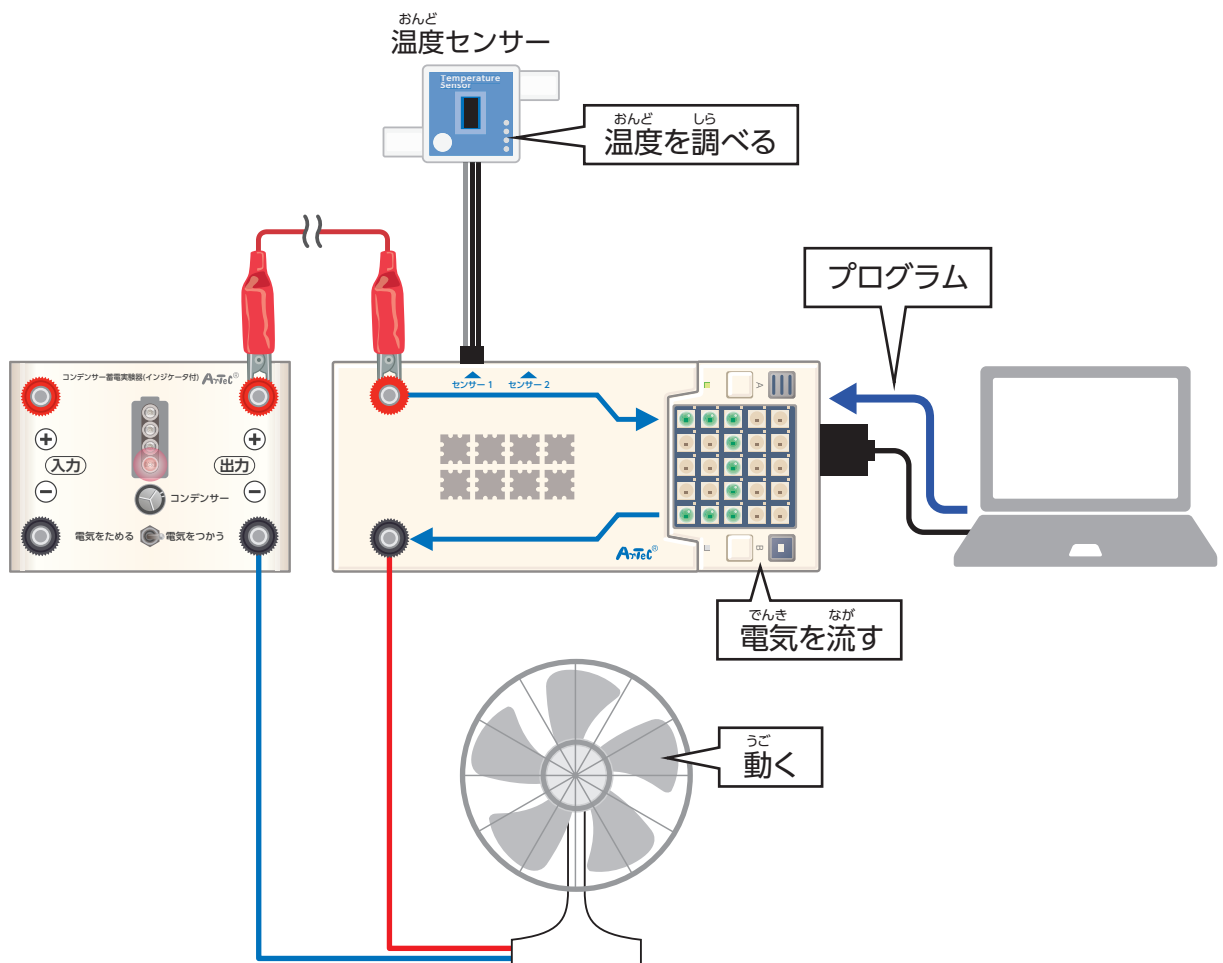


4章 感温扇風機^{しやう かんおん せんぶうき せいさく}の制作

エアコンは、内部^{ないぶ}のコンピューター^{おんど}が温度センサー^{げんざい}によって現在の温度^{おんど}を計測^{けいそく}し、設定温度^{せっていおんど}に近づく^{ちか}ようにモーター^{うご}の動き^{せいぎょ}を制御^{そうふう}して送風^{へや}することで、部屋の温度^{おんど}を調整^{ちやうせい}しています。



この章では、エアコンの代わりに扇風機^{せんぶうき}をつくって、周りの温度^{まわ}に応じて自動^{おんど}でONとOFF^{おう}が切りかわる^{じどう}ようにプログラミング^きします。



❗ スタディーノビットにも温度センサー^{おんど}は内蔵^{ないぞう}されていますが、基板^{きばん}の温度^{おんど}の影響^{えいきやう}を受けるため、気温^うを正確^{きおん}に測る^{せいさく}ことはできません。そのため今回は、外部^{はか}接続^{かんおん}の温度センサー^{せんぶうき}を使用して感温扇風機^{かんおん}をつくります。

4-1. 感温扇風機の組み立て

❖ プロペラパーツなどが付属していないセットを使用されている場合は、お手持ちのモーターに紙でつくったプロペラをつけるなどして代用してください。

つか 使うパーツ



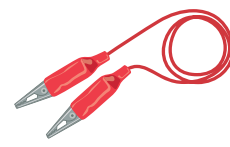
スタディーノビット × 1



電気の利用拡張
ボックス × 1



電池ボックス × 1



みのむし
リード線 赤 × 1



コンデンサー蓄電器 × 1



手回し発電機 × 1



温度センサー × 1



センサーコード × 1



モーター × 1



モーターカバー × 1



モーター台 × 1



ブロック ハーフC
(薄水) × 10



プロペラ × 1

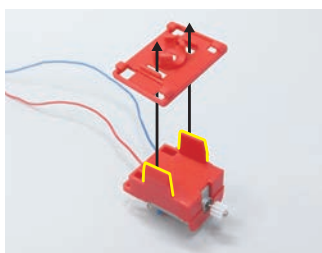
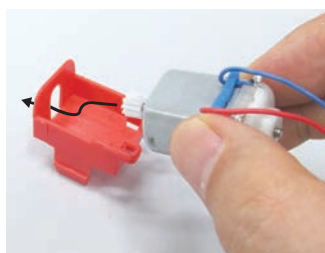


プロペラ台 × 1

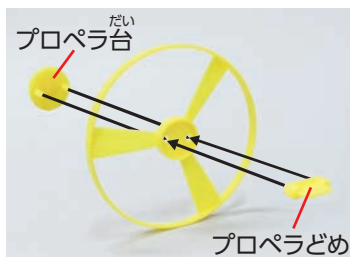


プロペラどめ × 1

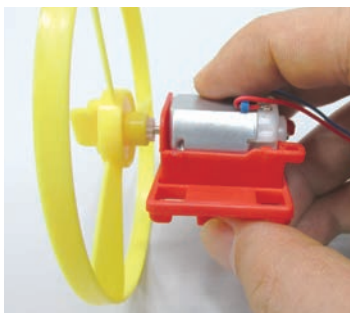
① モーター部分を図のように組み立てます。



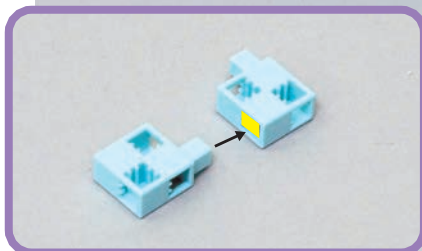
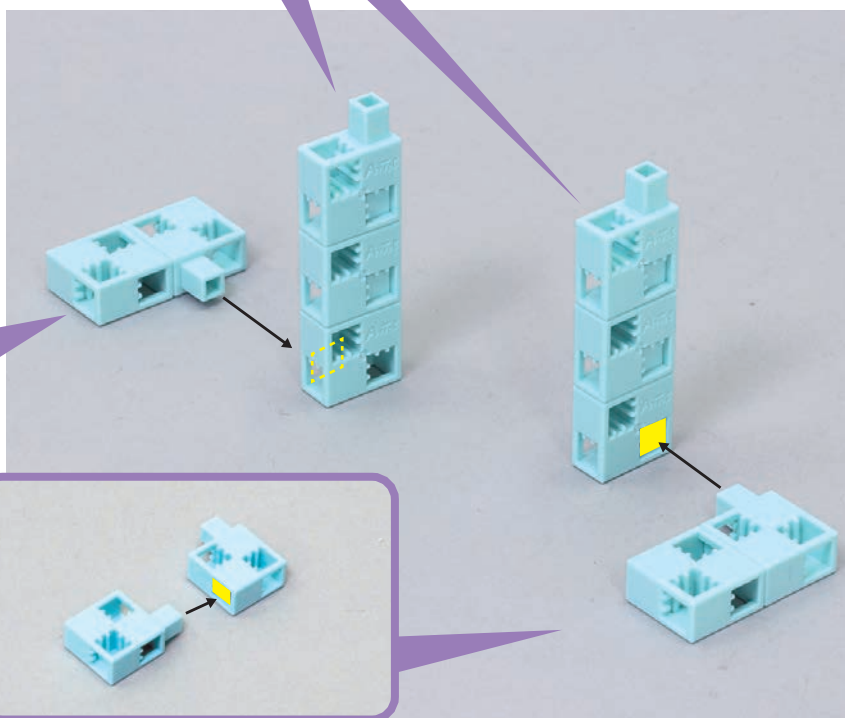
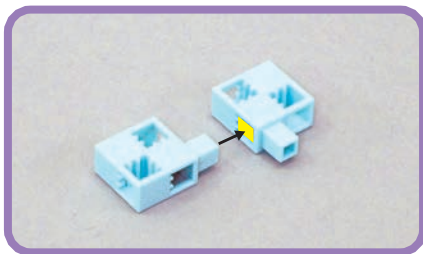
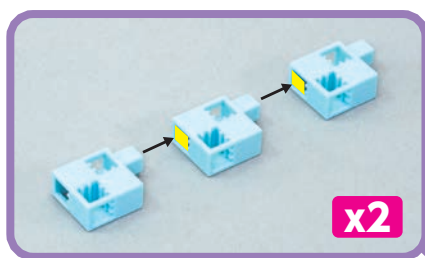
② プロペラ部分を図のように組み立てます。



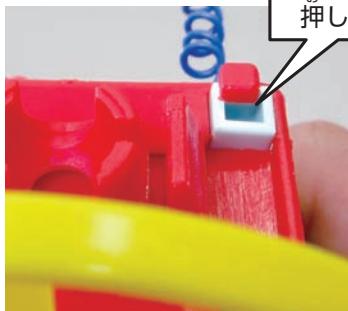
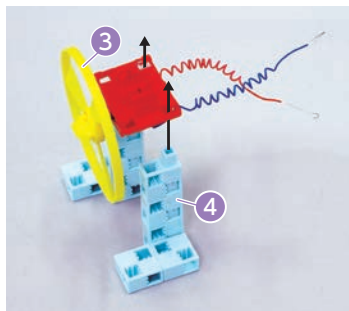
③ ①と②を組み合わせます。



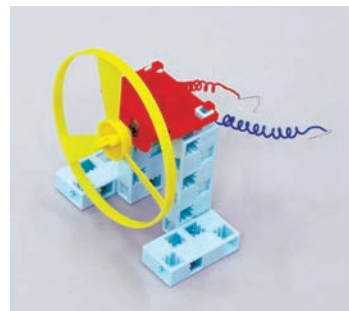
④ ブロックを図のように組み立てます。



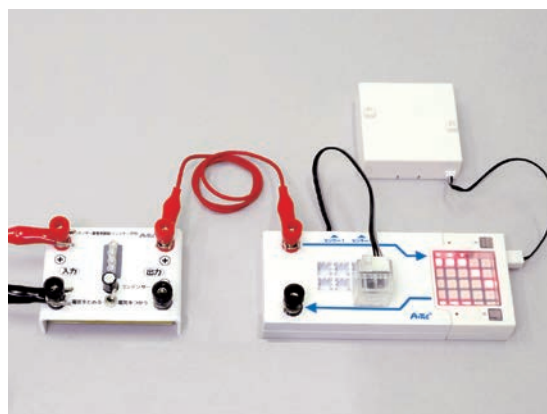
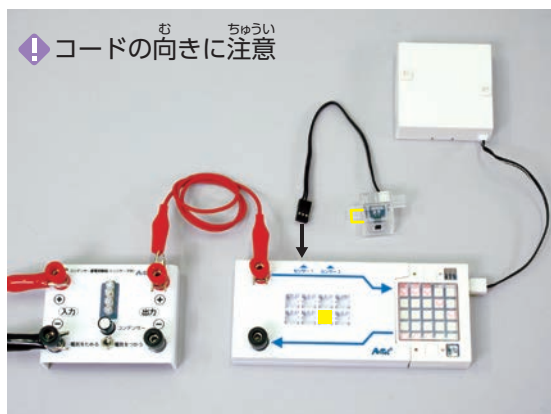
⑤ ④に③を組みあ合わせます。



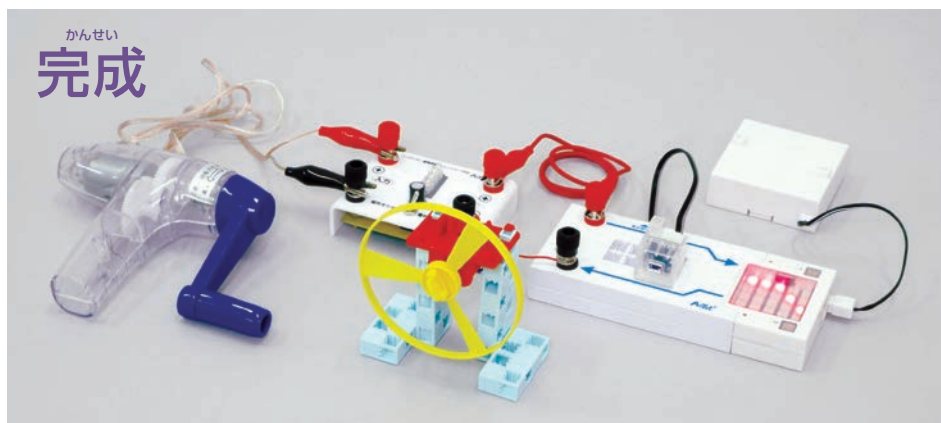
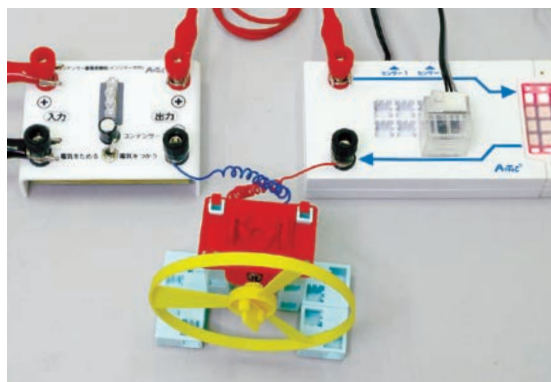
あおとつき
青い突起を
お押し込む



⑥ 電気の利用拡張ボックスのセンサー1に温度センサーを取り付けます。



⑦ モーターの導線を図のようにつなぎます。

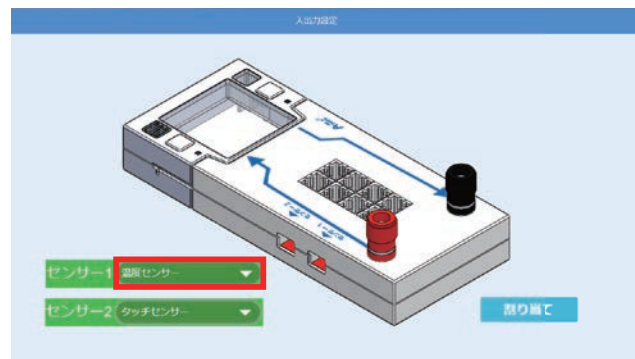


かんせい
完成

4-2. 温度センサーの使いかた

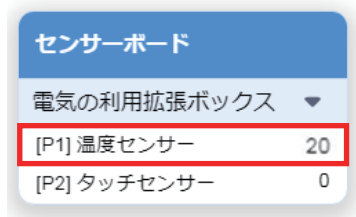
周りの温度によって、温度センサーの値がどのように変化するかを調べます。

メニューの「編集」から「入出力設定」を選択し、センサー1を「温度センサー」にします。

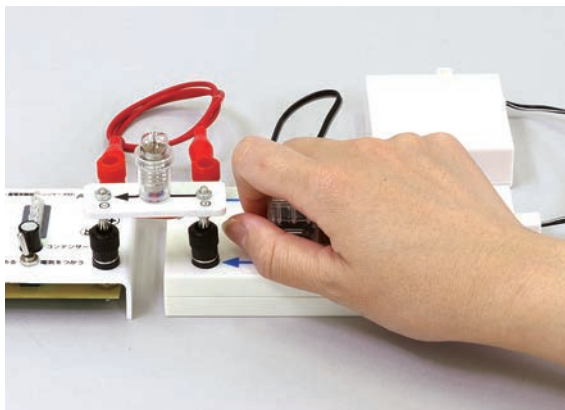


スタディーノビットを通信状態(≫18・19ページ)にして、センサーボードの「[P1]温度センサー」の値を確認します。温度センサーの値は、現在の温度[°C]を表しています。

❗ 本体の温度センサーと間違えないように注意してください。



温度センサーを手で握り続けると、センサーボードの値が上昇します。



❗ 実験に関する注意

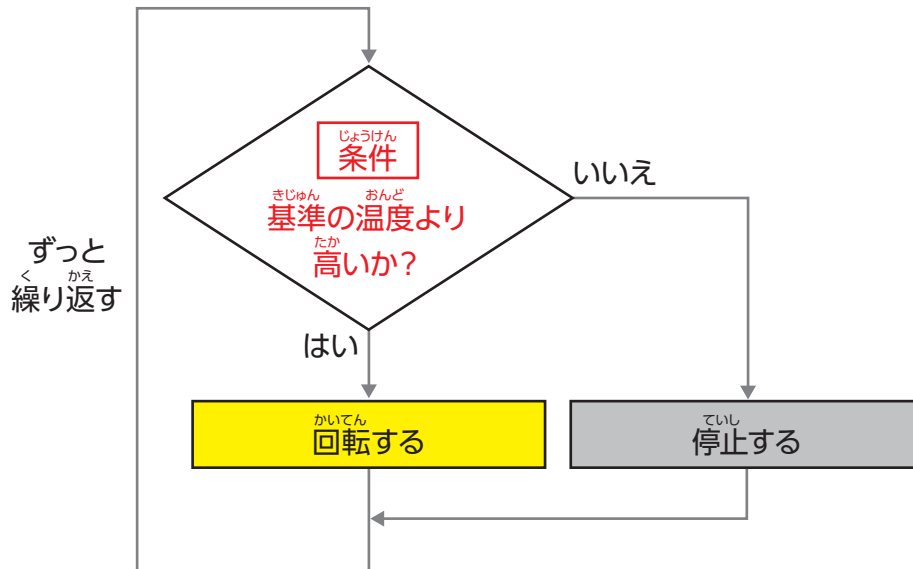
- ・夏場などの暑い環境だと温度が上がりにくいので、涼しい環境で行ってください。
- ・温度を下げるために水をかけないでください。機器の故障につながります。

温度センサーの値の変化

温度が上がる → 温度センサーの値は**大きくなる**
 温度が下がる → 温度センサーの値は**小さくなる**

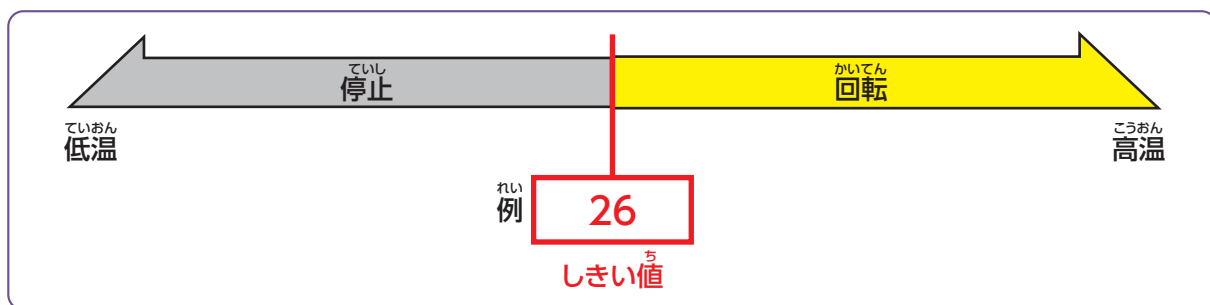
4-3. 感温扇風機のプログラムの作成

温度センサーの値を使って、温度が高いときはモーターが回り、低いときにはモーターが止まるという動作をさせるためのプログラムを考えます。この動作をフローチャートであらわすと以下のようにになります。



4-3-1. しきい値の設定

28ページで調べたセンサーの値を参考に、モーターの動きを変えるための基準となる温度(しきい値)を設定します。



4-3-2. 条件によって動作を分けるプログラムの作成

「温度センサーの値がしきい値よりも大きい」という条件を、下のブロックを使ってつくります。

調べる

センサー 1

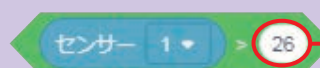
演算

> 50

……電気の利用拡張ボックスの「センサー1」につないだセンサーの値を調べるブロック

……左の値が右の値(50)より大きいかどうかを比べるブロック

条件の例



数字の部分には、決めたしきい値が入ります。

このブロックを使い、29ページのフローチャートの通りに動作する感温扇風機のプログラムを完成させます。

プログラム例

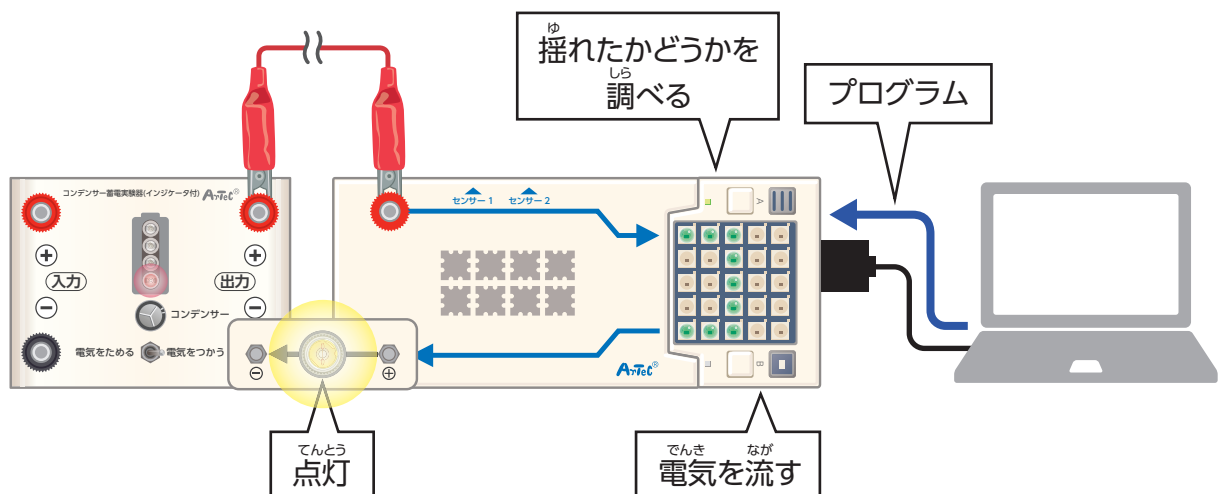


5章 非常灯の制作

マンションや商業施設には、火災や地震が起こった際、安全に避難できるようにするためのライトが設置されています。

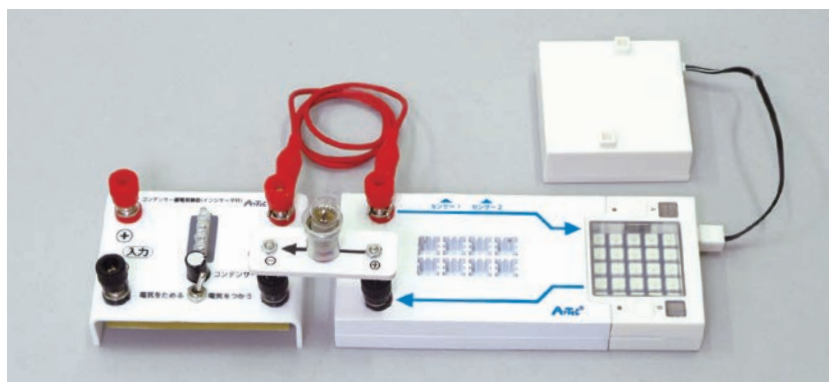


この章では、スタディーノビットに内蔵されている加速度センサーを利用して、揺れを検知すると点灯するライトをつくります。



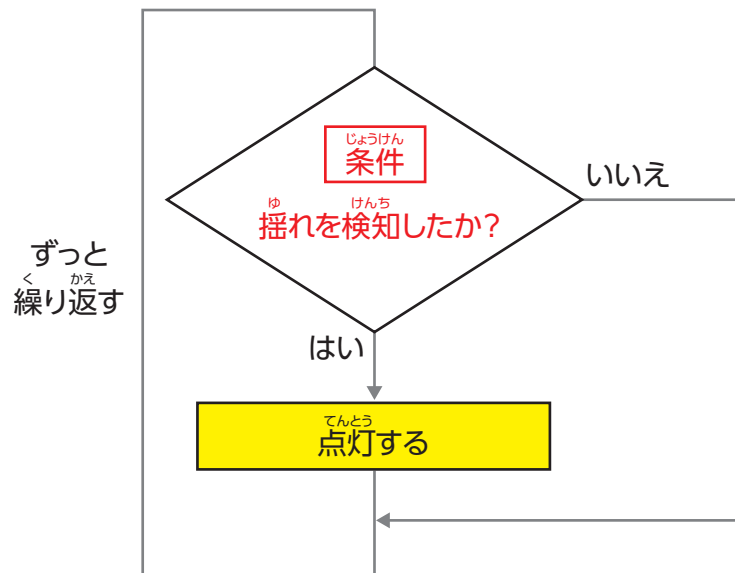
5-1. 実験準備

スタディーノビットとコンデンサー、豆電球を図のようにつなぎます。



5-2. 非常灯のプログラムの作成

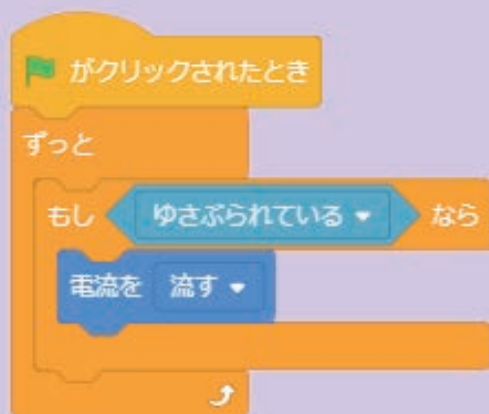
非常灯は、揺れを検知している(地震が起きている)最中だけでなく、揺れが収まったあとも点灯させておく必要があります。そのため、ライトの点灯はプログラムで制御し、消灯は手動で行う(任意のタイミングで電気の供給を止める)ようにします。プログラムで行われる動作をフローチャートであらわすと以下ようになります。



加速度センサーが揺れを検知したかどうかを調べるには、「調べる」カテゴリの「ゆさぶられている」を使います。

加速度センサーの詳しい使いかたについては33ページを参照してください

プログラム例

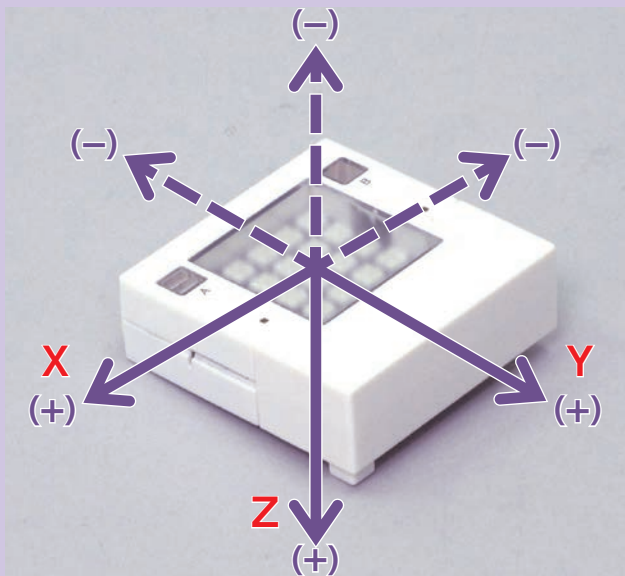


かそくど かそくど そくど いっていいかん へんか けいそく
加速度センサーは、加速度（速度が一定時間あたりどれだけ変化したか）を計測するセン
サーであり、X・Y・Zのそれぞれの方向に動かしたときの加速度を調べることができます。
スタディーノビットをつうしんじょうたい かたむ ふ すうち へんか
スタディーノビットを通信状態にすると、傾けたときや振ったときに数値が変化することを
かくにん すうち へんか
確認できます。（数値は-2.00から2.00のあいだで変化します。）

すうちけんしゆつ ほうほう こうこうぶつり ないよう しょうがっこう
! 数値検出の方法については、高校物理の内容となりますので、小学校では

ゆさぶられた ▼ とき

や ゆさぶられている ▼ のブロックを使ってください。



センサーボード

Studuino:bit ▼

ボタンA	1
ボタンB	1
光センサー	20
温度センサー	24.53
加速度センサー X	-0.55
加速度センサー Y	0.81
加速度センサー Z	-0.03
ジャイロセンサー X	0
ジャイロセンサー Y	-4

無断複製・転載を禁じます