

# Stduino

## ブロックプログラミング環境

---

### 入門ガイド 前編

【LED ブザー プッシュスイッチ/センサー】

Ver1.0.0.0

2014/11/1



本資料は、Stduino(スタディーノ)プログラミング環境のブロックプログラミング環境のチュートリアルになります。Stduino プログラミング環境の変更に伴い、加筆・修正が加えられる可能性があります。

## 更新履歴

| バージョン      | 前バージョンからの変更内容                 |
|------------|-------------------------------|
| Ver0.9.7.1 | 誤植修正しました。P34, P56             |
| Ver1.0.0.0 | 入出力設定に、「チェックを全て外す」ボタンを追加しました。 |

## もくじ

|  |    |
|--|----|
| 1. はじめに .....                          | 1  |
| 1.1. Studuino のセットアップ.....             | 1  |
| 1.2. 接続方法.....                         | 1  |
| 2. LED.....                            | 4  |
| 2.1. Studuino 基板と LED との接続.....        | 4  |
| 2.2. 入出力ポート情報の設定 .....                 | 4  |
| 2.3. LED 点灯プログラムの作成.....               | 6  |
| 2.4. LED 点滅プログラムの作成.....               | 8  |
| 2.5. LED の点滅回数を指定する.....               | 10 |
| 2.6. 点滅の速さを変更する .....                  | 13 |
| 3. ブザー .....                           | 15 |
| 3.1. Studuino 基板とブザーの接続 .....          | 15 |
| 3.2. 入出力ポート情報の設定 .....                 | 15 |
| 3.3. ブザーを鳴らす .....                     | 16 |
| 3.4. 連続で音を鳴らす .....                    | 18 |
| 3.5. リストを使って連続で音を鳴らす.....              | 21 |
| 4. プッシュスイッチとセンサー .....                 | 30 |
| 4.1. プッシュスイッチ .....                    | 30 |
| 4.1.1. 入出力ポート情報の設定.....                | 31 |
| 4.1.2. プッシュスイッチの動作確認.....              | 31 |
| 4.1.3. プログラミング.....                    | 34 |
| 4.2. 光センサー .....                       | 41 |
| 4.2.1. Studuino 基板と光センサーの接続.....       | 41 |
| 4.2.2. 入出力ポート情報の設定 .....               | 41 |
| 4.2.3. 光センサーの動作確認.....                 | 42 |
| 4.2.4. プログラミング.....                    | 42 |
| 4.3. 赤外線フォトリフレクタ .....                 | 46 |
| 4.3.1. Studuino 基板と赤外線フォトリフレクタの接続..... | 46 |
| 4.3.2. 入出力ポート情報の設定 .....               | 46 |
| 4.3.3. 赤外線フォトリフレクタの動作確認.....           | 47 |

|        |                             |    |
|--------|-----------------------------|----|
| 4.3.4. | プログラミング.....                | 48 |
| 4.4.   | タッチセンサー.....                | 53 |
| 4.4.1. | Studuino 基板とタッチセンサーの接続..... | 53 |
| 4.4.2. | 入出力ポート情報の設定.....            | 53 |
| 4.4.3. | タッチセンサーの動作確認.....           | 54 |
| 4.4.4. | プログラミング.....                | 55 |
| 4.5.   | 音センサー.....                  | 58 |
| 4.5.1. | Studuino 基板と音センサーの接続.....   | 58 |
| 4.5.2. | 入出力ポート情報の設定.....            | 58 |
| 4.5.3. | 音センサーの動作確認.....             | 59 |
| 4.5.4. | プログラミング.....                | 59 |

# 1. はじめに

## 1.1. Studuino のセットアップ

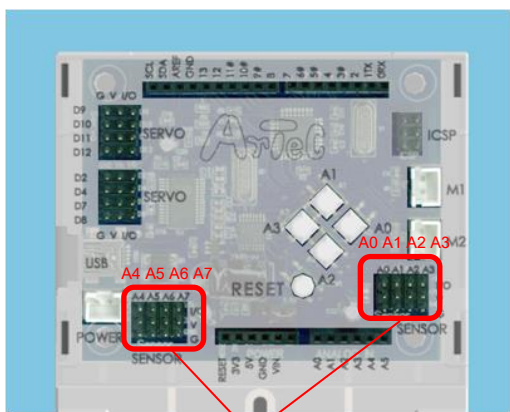
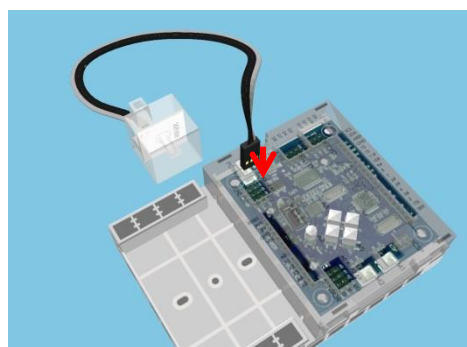
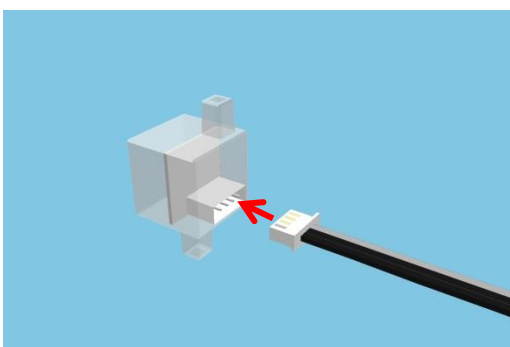
<http://www.artec-kk.co.jp/studuino> より Studuino プログラミング環境セットアップ方法をご確認のうえ、セットアップを行ってください。

## 1.2. 接続方法

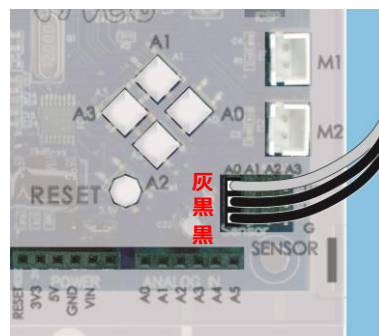
### ① 各種センサー/LED/ブザーの接続方法

センサーパーツに下図のようなセンサー接続コードを接続し、Studuino 基板のセンサー/LED/ブザー用コネクタに接続します。

※加速度センサー以外のセンサーはセンサー接続コード（3芯）を使用し、加速度センサーはセンサー接続コード（4芯）を使用してください。



センサー接続コードは下図のように灰色のコードが上側にくるように接続します。



センサー/LED/ブザー用コネクタ

※音センサー、光センサー、赤外線フォトリフレクタは A0～A7 に接続できます。

※タッチセンサー、LED、ブザーは A0～A5 に接続できます。

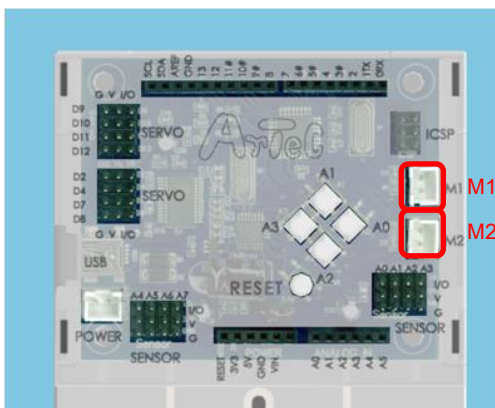
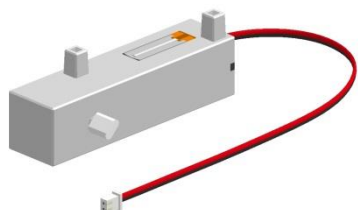
※加速度センサーは、A4 と A5 にまたがって接続します。

※A0～A3 のコネクタ使用時は、プッシュスイッチ A0～A3 が使用できなくなります。

## ② DC モーターの接続方法

DC モーターは 2 つまで接続可能で、DC モーター用コネクタ（M1、M2）に接続します。

※DC モーターのコネクタは逆方向に接続できない構造になっています。

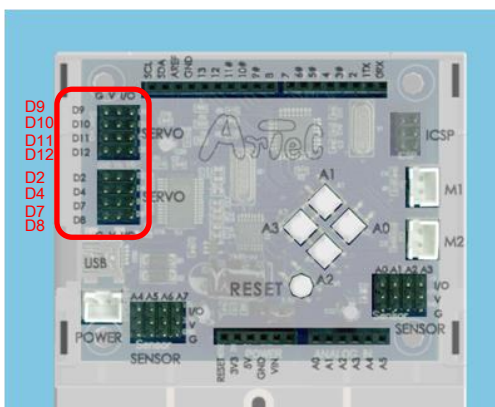
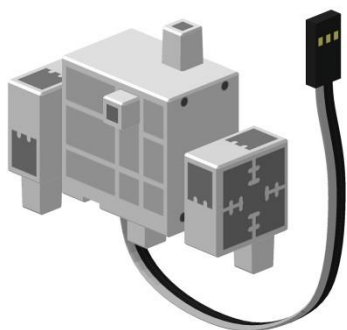


※DC モーター用コネクタ-M1 とサーボモーター用コネクタ-D2,D4 は同時に使用できません。

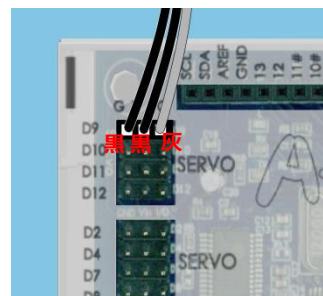
※DC モーター用コネクタ-M2 とサーボモーター用コネクタ-D7,D8 は同時に使用できません。

## ③ サーボモーターの接続方法

サーボモーターは 8 つまで接続可能で、サーボモーター用コネクタに接続します。



サーボモーターのコードは右図のように  
灰色のコードが右側にくるように接



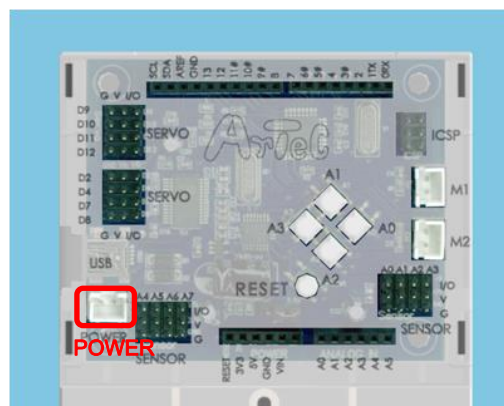
※DC モーター用コネクタ-M1 とサーボモーター用コネクタ-D2,D4 は同時に使用できません。

※DC モーター用コネクタ-M2 とサーボモーター用コネクタ-D7,D8 は同時に使用できません。

#### ④ 電池ボックスの接続方法

電池ボックスは電源コネクタ（POWER）に接続します。

センサー/LED/ブザー/スイッチは、USB からの供給電源で動作しますが、DC モーターやサーボモーターを動かす場合は電源コネクタから電力を供給する必要があります。また、USB ケーブルを外して動作させるときにも電源コネクタから電力を供給する必要があります。

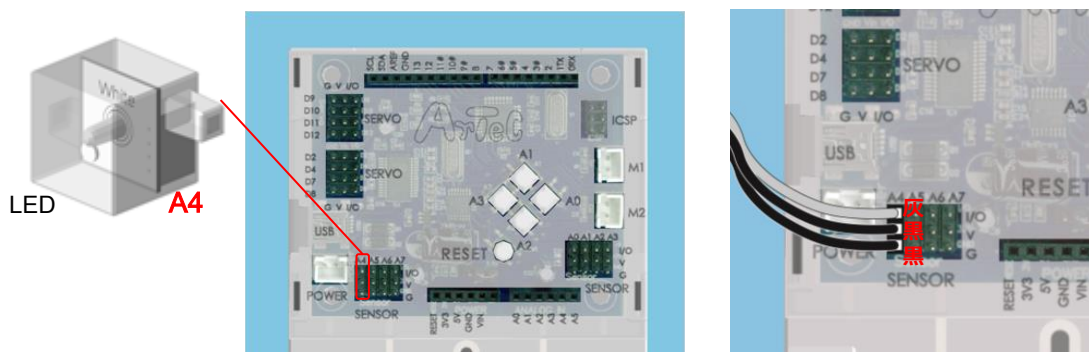


## 2. LED

LEDを使ったプログラムの作成を通して、Stduino 基板とパーツの接続、プログラミング環境への入出力設定の方法、ブロックプログラミング環境でのプログラミング方法、プログラムの作成転送方法を学習します。

### 2.1. Stduino 基板と LED との接続

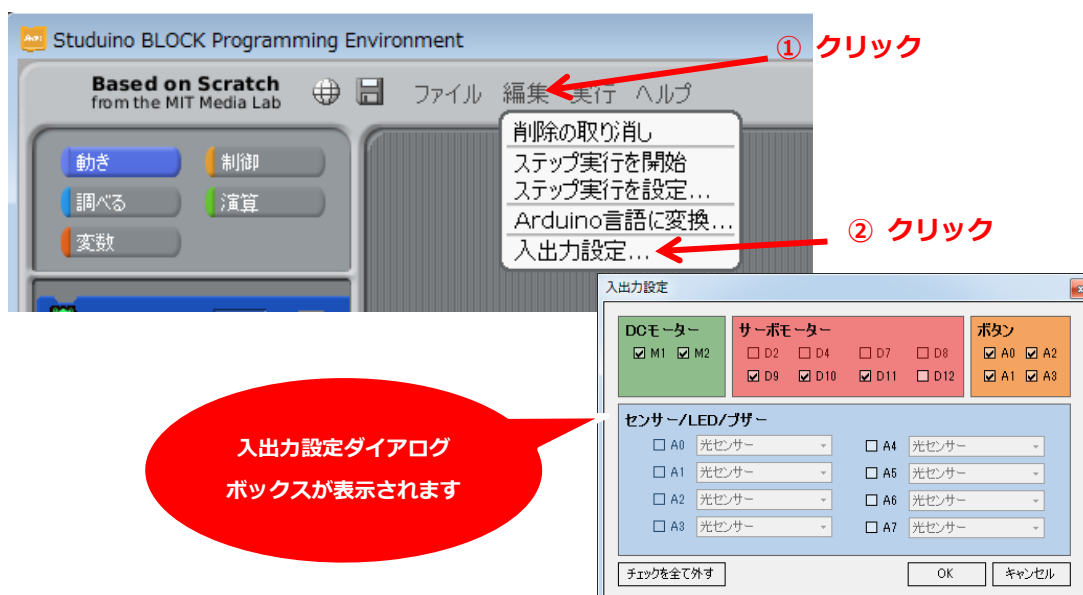
Stduino 基板のセンサー/LED/ブザー用コネクターの A4 に LED を接続します。



センサー接続コードの向きに注意！  
上側に灰色のコードがくるように。

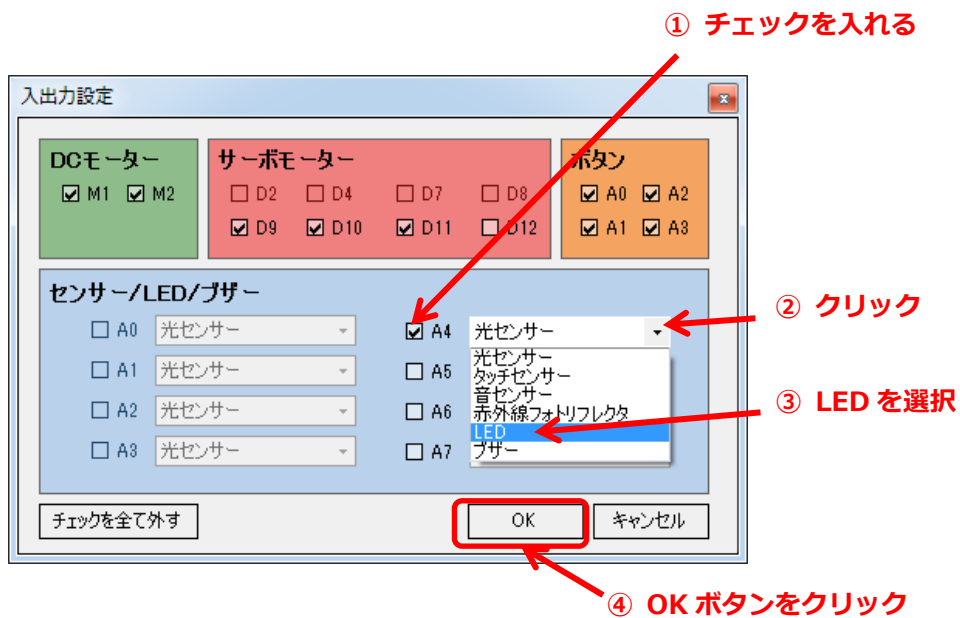
### 2.2. 入出力ポート情報の設定

プログラムを始める前に、ブロックプログラミング環境に Stduino 基板のポート情報を設定します。ブロックプログラミング環境を起動し、メニューバーの「編集」から「入出力設定...」を選択して、入出力設定ダイアログボックスを開きます。







入出力設定ダイアログボックスの「センサー/LED/ブザー」エリアの A4 のチェックボックスをチェックし、LED を選択して下さい。



以上でブロックプログラミング環境に Studuino 基板の入出力ポート情報を設定したことになります。

## 2.3. LED 点灯プログラムの作成

 ブロックを使って、ブロックプログラミングの基本的な操作を学習します。

- ① コマンドグループパレットの「動き」ボタンをクリックし、「動き」パレット上の  ブロックをスクリプトエリア内にドラッグし、 ブロックにくっつけます。

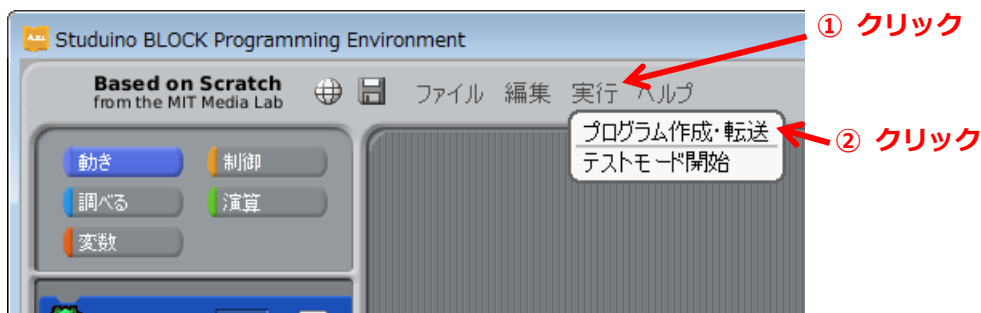


 **ブロックどうしがくっついていることを確認しましょう**

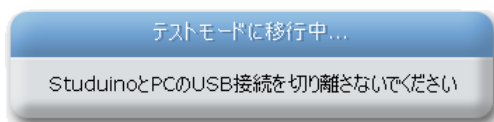


※スクリプトエリア内にドロップされたブロックは自由にドラッグ&ドロップにより移動させることができます。また、ドラッグ&ドロップさせる際、下にくっついたブロックがある場合は、それらが一つのかたまりとして移動されます。

- ② Studuino 基板と PC を USB コードで接続し、メニューバーの「実行」から「プログラム作成・転送」を選択し、プログラムを転送してください。



下のメッセージが表示され消えると、プログラムが Studuino 基板に正常に転送されたこととなります。転送が完了すると、LED が自動的に点灯します。



下記のメッセージが表示された場合、プログラムが Studuino 基板に転送されていないので、対策を実施し、再度「プログラム作成・転送」を実行して下さい。

| メッセージ | 対策  |
|-------|---|
|       | Studuino 基板と PC が USB コードでしっかり接続されていることを確認してください  |
|       | ブロックに接続されていないブロックがあります。 に繋がれない余分なブロックを削除してください。   |
|       | Studuino 基板と通信しているアプリケーションがある場合は、そのアプリケーションを終了してください。通信しているアプリケーションがない場合は、USB コードを Studuino 基板から抜いて再度接続し、Studuino 基板のリセットボタンを押してください。 |

## 2.4. LED 点滅プログラムの作成

2.3 LED 点灯プログラムの作成で作成したプログラムを使って、LED を点滅させるプログラムを作成します。本プログラムで、無限ループを表す「ずっと」ブロックを使って、リピート処理を学習します。

- ① コマンドグループパレットの「制御」ボタンをクリックし、「制御」パレット上の「1秒待つ」ブロックをスクリプトエリア内にドラッグし、LED 点灯プログラムにくっつけます。



- ② 「LED A4 を点灯」ブロック上で右クリックし、コンテキストメニューを表示し、「複製」を選択し、ブロックを複製します。



- ③ 複製されたブロックを「1秒待つ」ブロックの下にくっつけます。



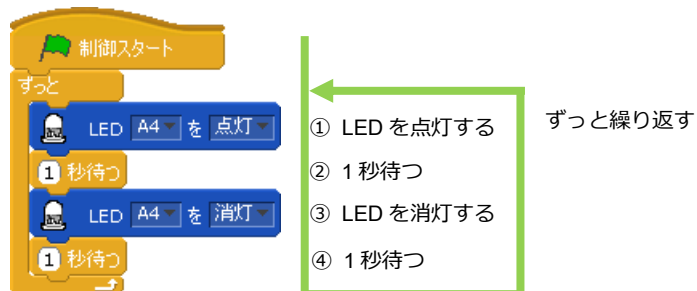
- ④ 「制御」パレット上の「ずっと」ブロックをスクリプトエリア内にドラッグし、プログラム全体を囲う形でドロップします。



- ⑤ 2つ目の「LED A4 を点灯」ブロックの「点灯」を「消灯」に変更します。





以上で、LED を 1 秒間隔でずっと点滅するプログラムができました。



Studuino 基板と PC を USB コードで接続し、メニューバーの「実行」から「プログラム作成・転送」を選択してください。転送が完了すると、LED が自動的に 1 秒間隔で点滅します。

## 2.5. LED の点滅回数を指定する

2.4 LED 点滅プログラムの作成で作成したプログラムを使って、LED の点滅回数を制御するプログラムを作成します。本プログラムで、ループ回数を表す  ブロックを使って、リピート処理を学習します。

- ① 「制御」パレットから  ブロックをスクリプトエリアにドラッグ&ドロップします。




- ②  ブロックの中のブロックを  ブロックの開いているところに移動します。



- ③  ブロックをブロックパレットにドラッグ&ドロップします。



 ブロックを除く、スクリプトエリア上のブロックは、ブロックパレットにドラッグ&ドロップすることでスクリプトエリア上から削除できます。

※ブロックの上で右クリックして現れるメニューからも削除できます。

※間違えて削除してしまったブロックは、メニューバーの編集から「削除の取り消し」で元に戻すことができます。


- ④ ②で作成したブロックを  ブロックにくっつけます。




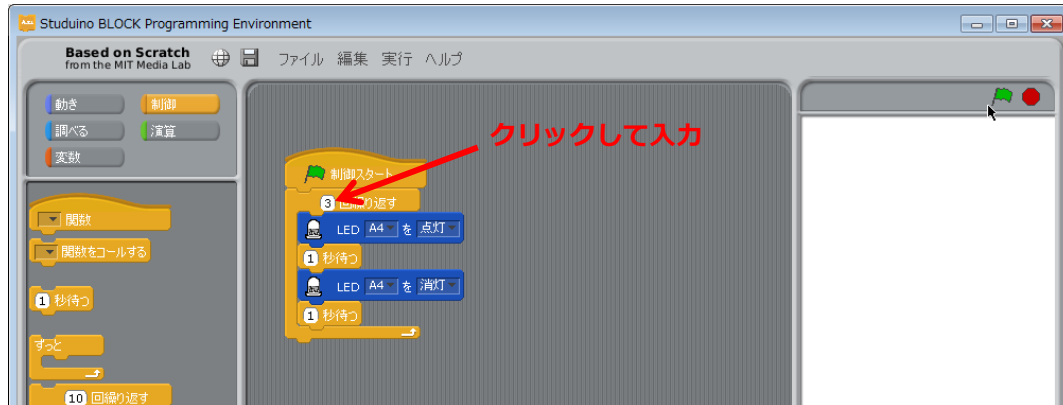
以上で、LED を 1 秒間隔で 10 回点滅するプログラムができました。



Studino 基板と PC を USB コードで接続し、メニューバーの「実行」から「プログラム作成・転送」を選択してください。転送が完了すると、LED が自動的に 1 秒間隔で 10 回点滅します。

次に、 → ブロックのリPEAT回数を変更することで点滅回数を変えます。

⑤  → ブロックの「10」をクリックし LED を点滅させたい回数を入力します。



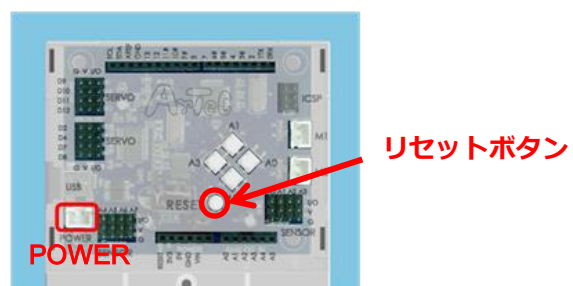
Studuino 基板と PC を USB コードで接続し、メニューバーの「実行」から「プログラム作成・転送」を選択してください。転送が完了すると、LED が自動的に 1 秒間隔で指定回数点滅します。

Studuino 基板に転送したプログラムを、再度実行する場合は、メニューバーの「実行」から「プログラム実行」を選択してください。



※Studuino 基板への転送完了後は、電池ボックスからの電源供給によりプログラムを実行することができます。Studuino 基板から USB コードを外し、電池ボックスを電源コネクタ（POWER）に接続し、電源を ON にし、LED が 1 秒間隔で指定回数点滅することが確認できます。

※終了したプログラムを再度実行する場合は、Studuino 基板のリセットボタンを押すことで再度実行できます。





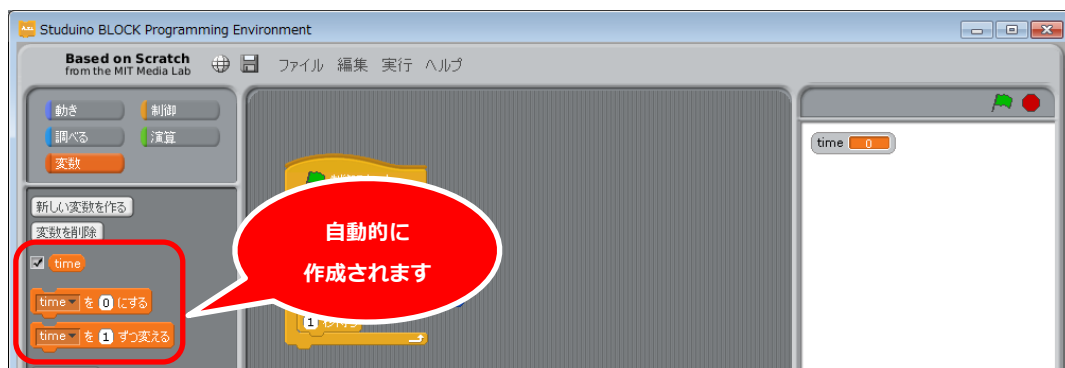
## 2.6. 点滅の速さを変更する

2.5 LED の点滅回数を指定するで作成したプログラムを使って、LED の点滅の速さを制御するプログラムを作成します。本プログラムで、変数ブロックを使って、変数を使用するプログラムを学習します。

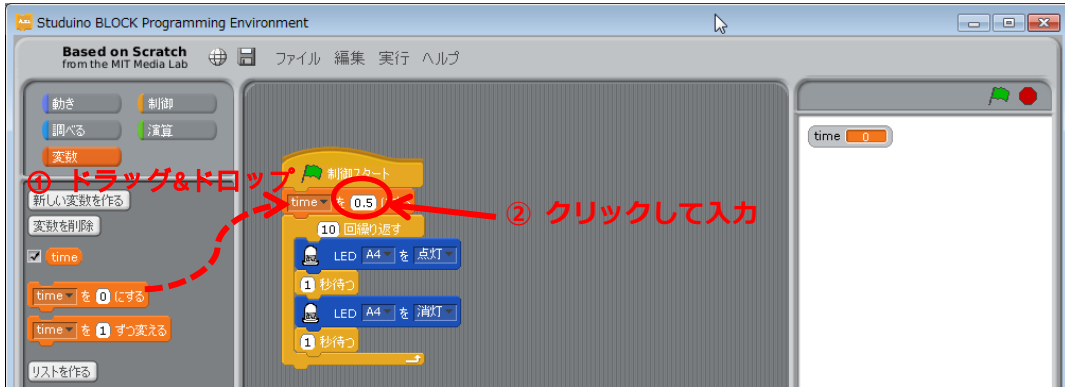
- ① コマンドグループパレットの「変数」ボタンをクリックし、ブロックパレットの「新しい変数を作る」ボタンをクリックします。変数名設定ダイアログボックスが表示されるので、ここでは「time」と入力し、OK ボタンをクリックします。



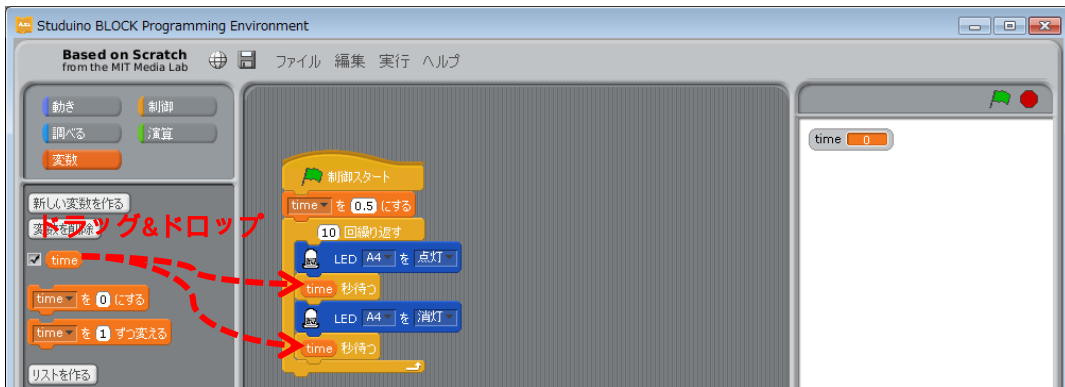
- ② time 変数を制御するブロックが作成されます。



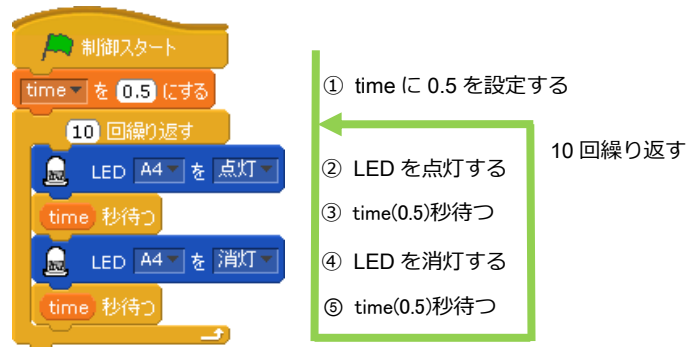
- ③ **time** を 0 にする ブロックを **制御スタート** ブロックの下にくっつけて、**time** を 0 にする ブロックの値を 0.5 に設定します。



- ④ **time** ブロックを **1 秒待つ** ブロックにくっつけます。



以上で、LED を 0.5 秒間隔で 10 回点滅するプログラムが作成できました。



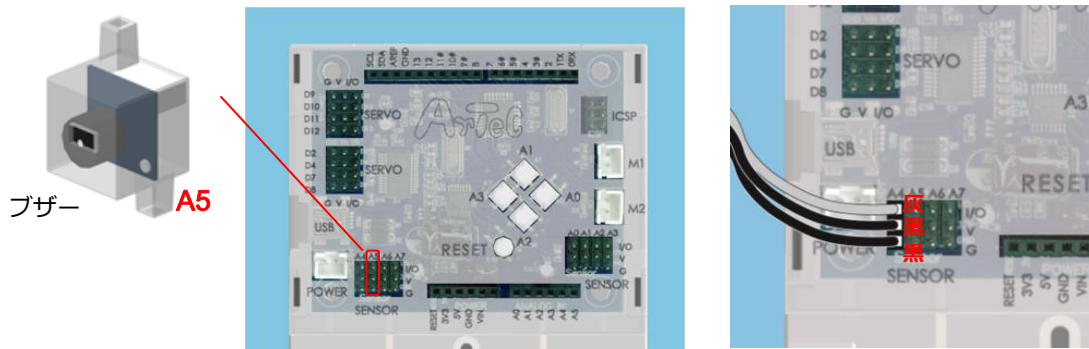
Studuino 基板 と PC を USB コードで接続し、メニューバーの「実行」から「プログラム作成・転送」を選択してください。転送が完了すると、LED が自動的に 0.5 秒間隔で 10 回点滅します。

### 3. ブザー

ブザーを使ったプログラムの作成を通して、ブザーの鳴らし方、止め方、リストブロックと関数ブロックの使い方を学習します。

#### 3.1. Studuino 基板とブザーの接続

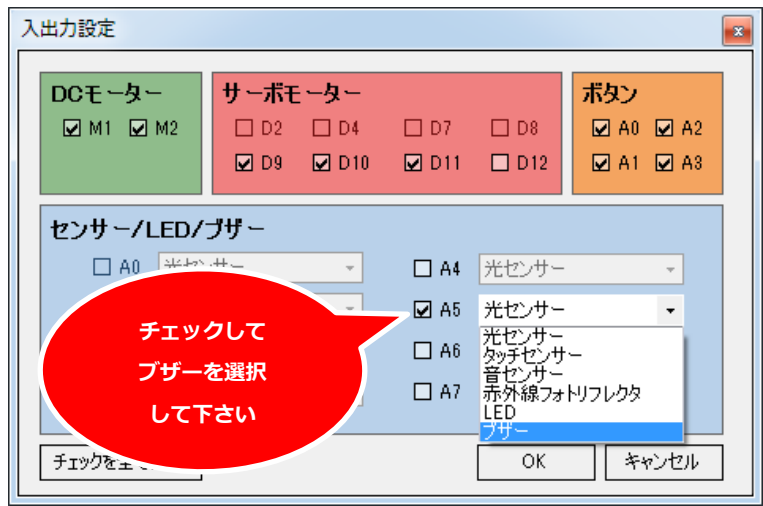
Studuino 基板のセンサー/LED/ブザー用コネクターの A5 にブザーを接続します。





センサー接続コードの向きに注意！  
上側に灰色のコードがくるように。

#### 3.2. 入出力ポート情報の設定



ブロックプログラミング環境に Studuino 基板のポート情報を設定します。ブロックプログラミング環境のメニューバーの「編集」から「入出力設定...」を選択して、入出力設定ダイアログボックスを開きます。入出力設定ダイアログボックスの「センサー/LED/ブザー」エリアの A5 をチェックしてブザーを選択して下さい。

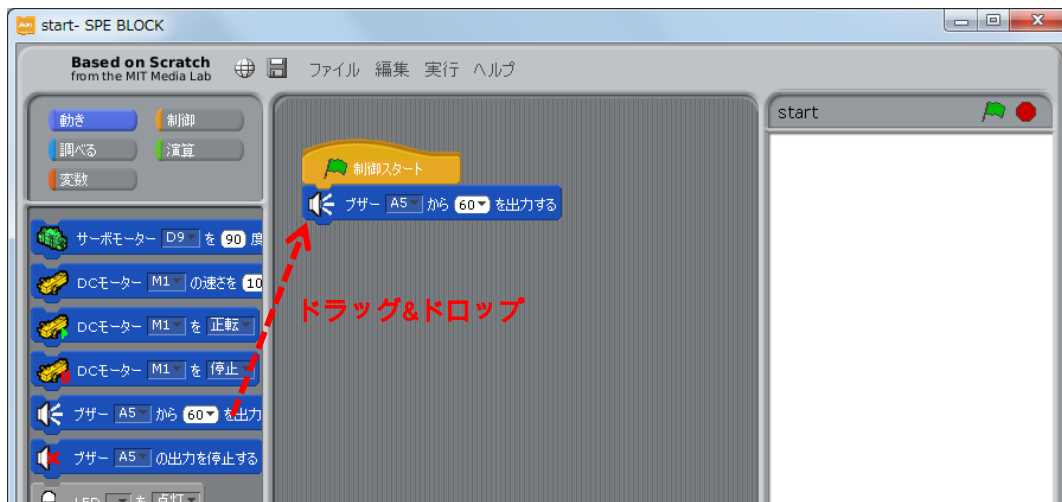



### 3.3. ブザーを鳴らす

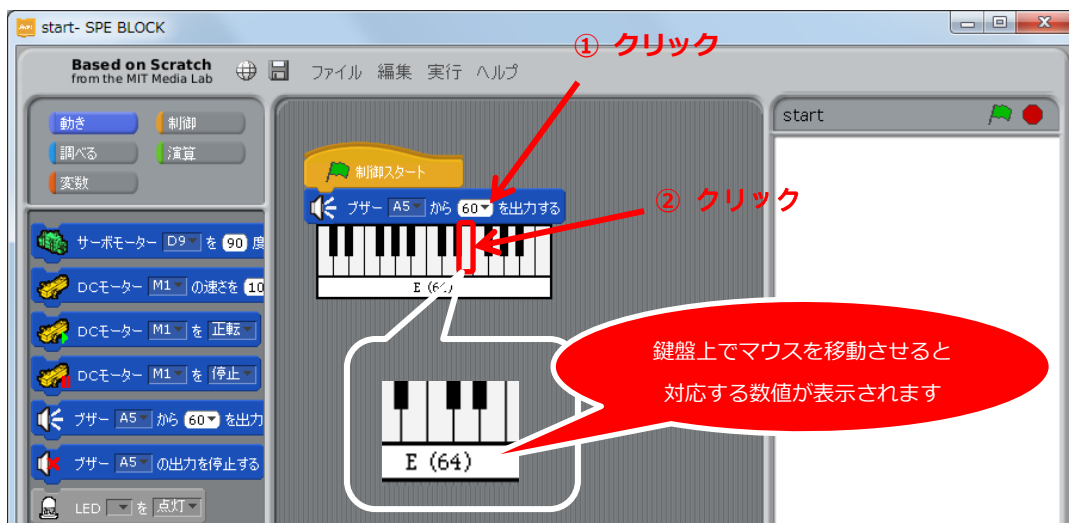
ブザーから 1 秒間音を鳴らすプログラムを作成します。本プログラムで、ブザーから音を制御する  ブザー [A5] から 60 を出力する ブロックと  ブザー [A5] の出力を停止する ブロックを使って、音を制御するプログラムを学習します。

※プログラムを新たに作る場合は、メニューバーの「ファイル」から「新規」を実行してください。

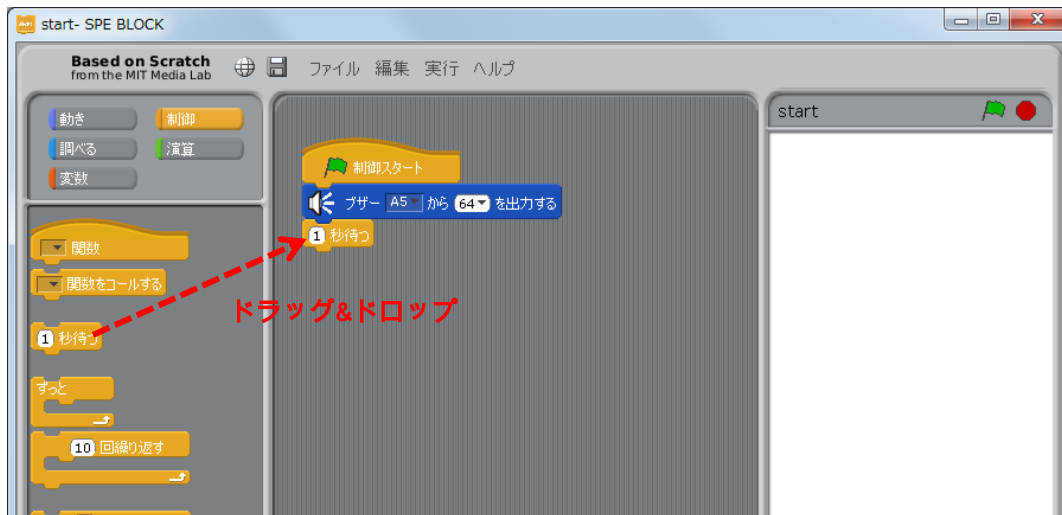
- ① 「動き」パレットから  ブザー [A5] から 60 を出力する ブロックを  制御スタート ブロックにくっつけます。



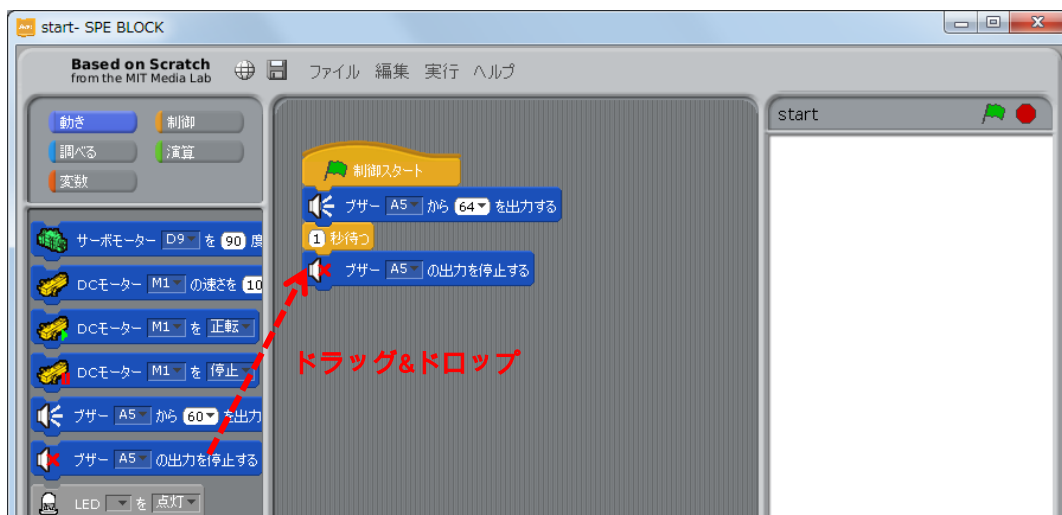
- ②  ブザー [A5] から 60 を出力する ブロックの▼をクリックし、鍵盤を表示します。表示された鍵盤をマウスでポイントすると、対応する音の数値が表示されるので、ここでは E (64) (※ミの音を表す)をクリックして選択します。



- ③ 「制御」パレットから「1秒待つ」ブロックを「ブザー A5 から 64 を出力する」ブロックの下にくっつけます。



- ④ 「動き」パレットの「ブザー A5 の出力を停止する」ブロックを「1秒待つ」ブロックの下にくっつけます。



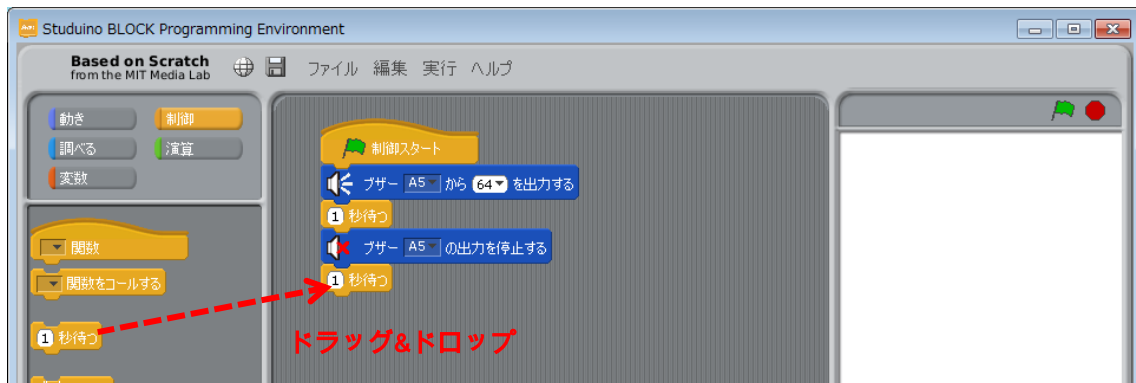
以上で、ブザーから“ミ”の音を 1 秒間鳴らすプログラムが作成できました。

Studuino 基板と PC を USB コードで接続し、メニューバーの「実行」から「プログラム作成・転送」を選択してください。転送が完了すると、ブザーから“ミ”の音が 1 秒間出力されます。

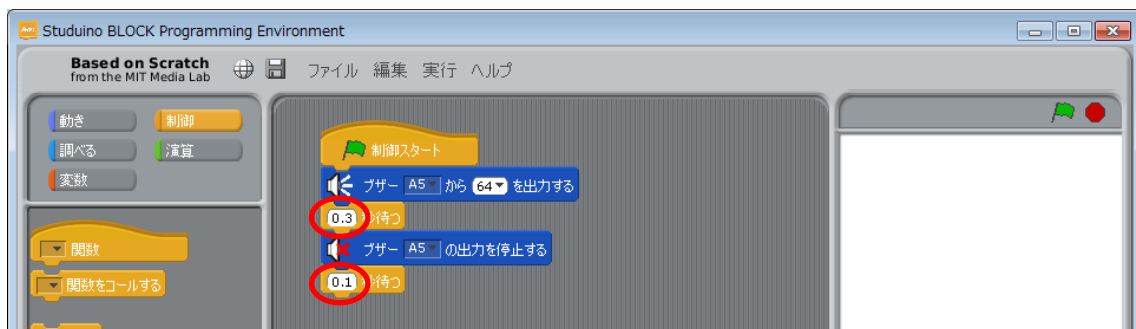
### 3.4. 連続で音を鳴らす

「きらきらぼし」(ド・ド・ソ・ソ・ラ・ラ・ソ)を出力するプログラムを作成します。ブザーから音を連続で出力することで、「きらきらぼし」の最初のメロディができます。

- ① 3.3 ブザーを鳴らすで作成したプログラムに **1秒待つ** ブロックを追加します。



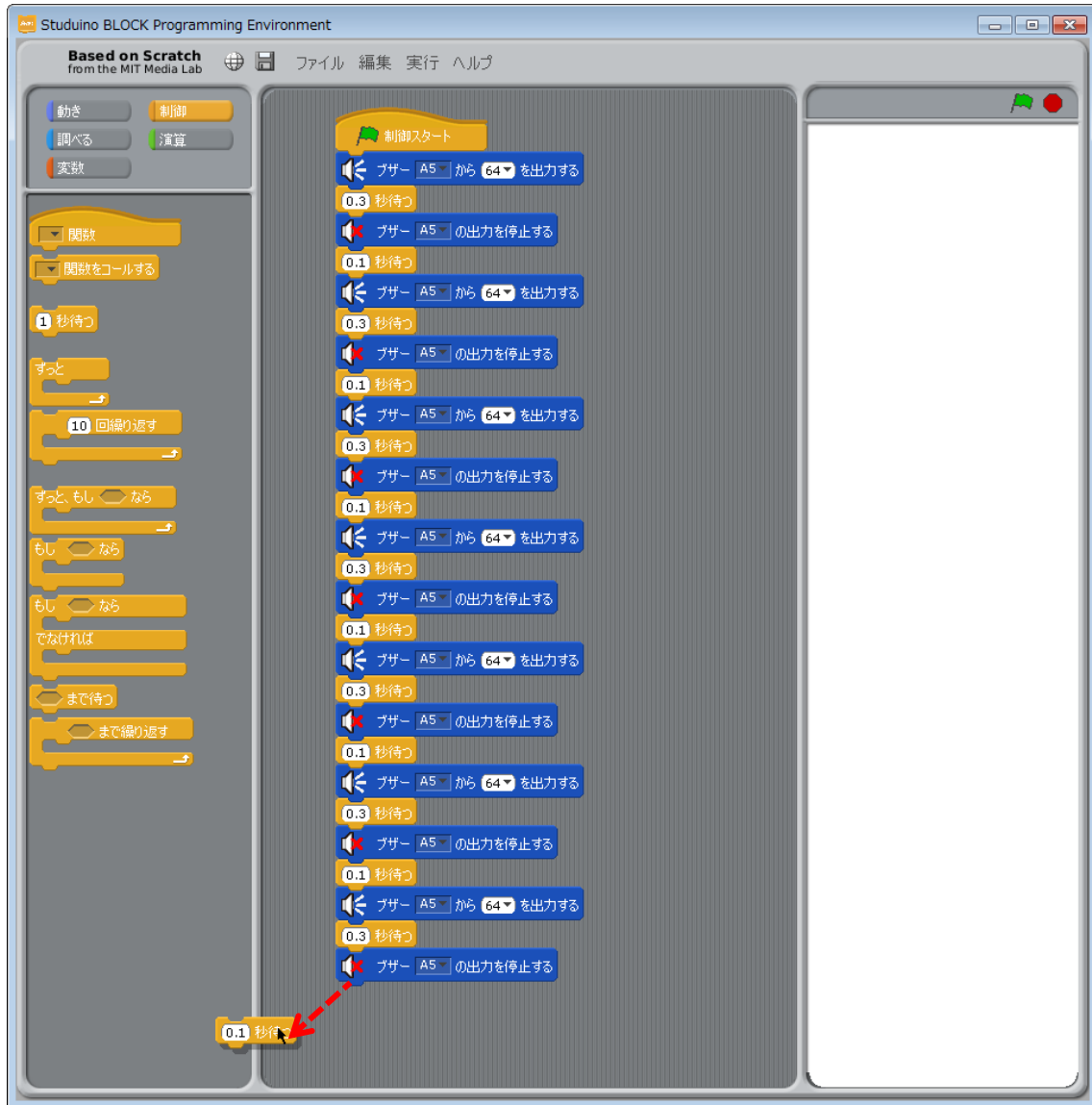
- ② **1秒待つ** ブロックの待ち時間を 0.3 秒と 0.1 秒に設定します。




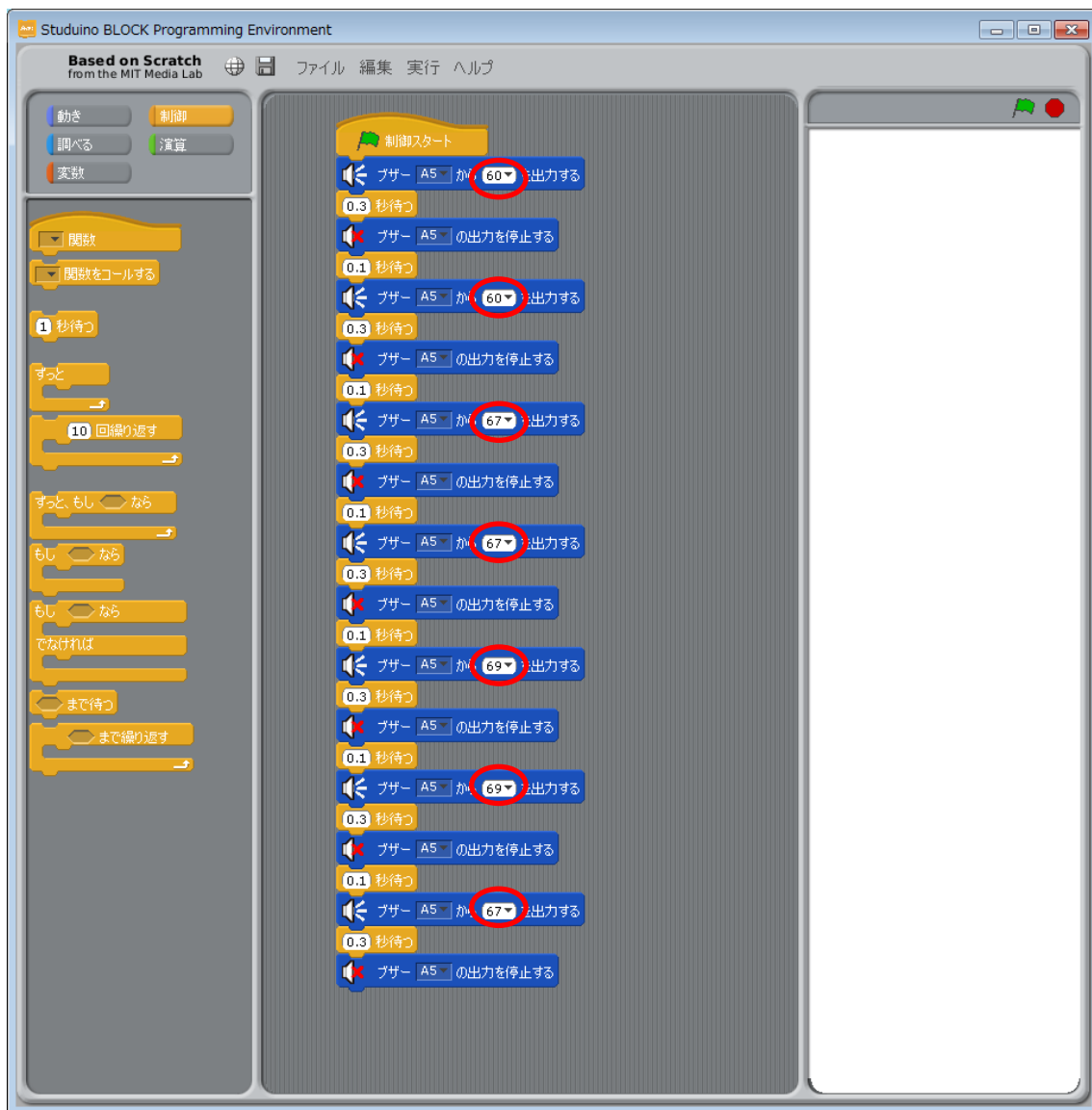
- ③ **ブザー A5 から 64 を出力する** ブロック上で右クリックして、複製を選択し、複製されたブロックを複製元のブロックとくっつけます。



- ④ ③の手順を繰り返してブロックの組を7つ作ります。最後の **0.1 秒待つ** ブロックを削除します(不要なブロックはコマンドパレットに戻すことで削除できます)。



- ⑤ 各  プザー A5 から 64 を出力する ブロックにド(60)・ド(60)・ソ(67)・ソ(67)・ラ(69)・ラ(69)・ソ(67)を設定します。



以上で、ブザーから「きらきらぼし」の最初のメロディを鳴らすプログラムが作成できました。

Studuino 基板と PC を USB コードで接続し、メニューバーの「実行」から「プログラム作成・転送」を選択してください。転送が完了すると、ブザーから「きらきらぼし」が 1 小節分出力されます。

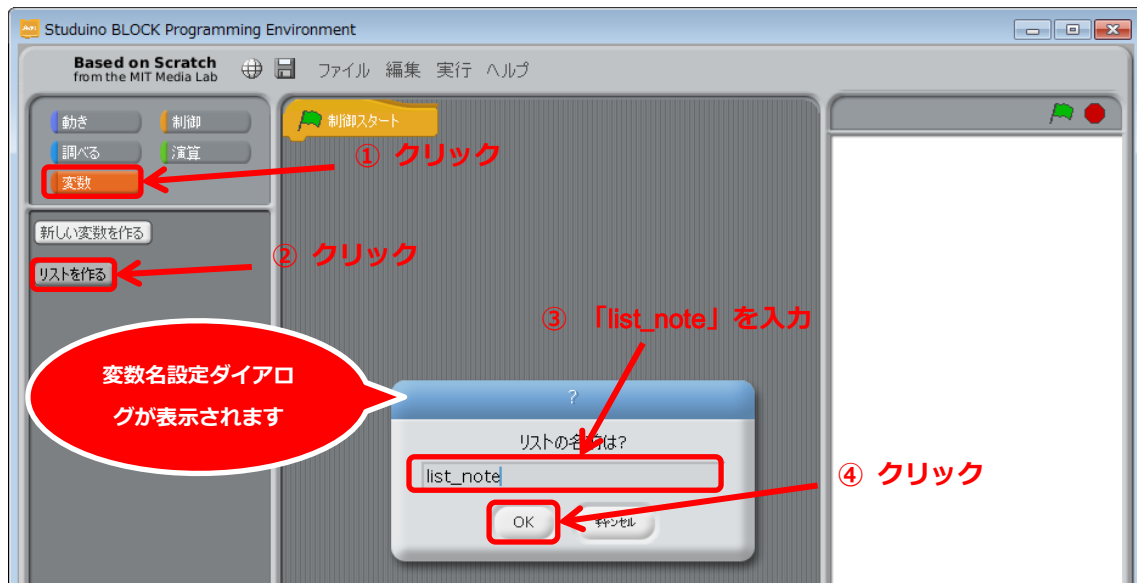


### 3.5. リストを使って連続で音を鳴らす

3.4 連続で音を鳴らすで作成したプログラムは、リストを使用する事で制御を簡単にする事ができます。まずはリストを作ってリストを使ってみましょう。

#### ● リストを使ってみる

- ① 変数パレットで、「リストを作る」ボタンをクリックして、ここではリスト名「list\_note」というリストを作成します。





- ② リストを制御するブロックと空のリストが作成されます。



以下で作成したリストを制御するブロックを説明します。

- ③ **0** を `list_note` に追加する ブロックを 3 つスクリプトエリアにドラッグしてくっつけ、0、12、15 を設定します。




- ④  をクリックすると、 ブロックに繋がっているブロックが上から順番に実行されます。空のリストに 0、12、15 が追加されます。



次に④で作ったリストを使用してリスト処理を説明します。

- ⑤ ③で作ったブロックを削除し、`list_note` の 1 番目に 0 を挿入する ブロックを  にくっつけ、「2」番目に「30」を挿入すると設定します。




⑥  をクリックします。



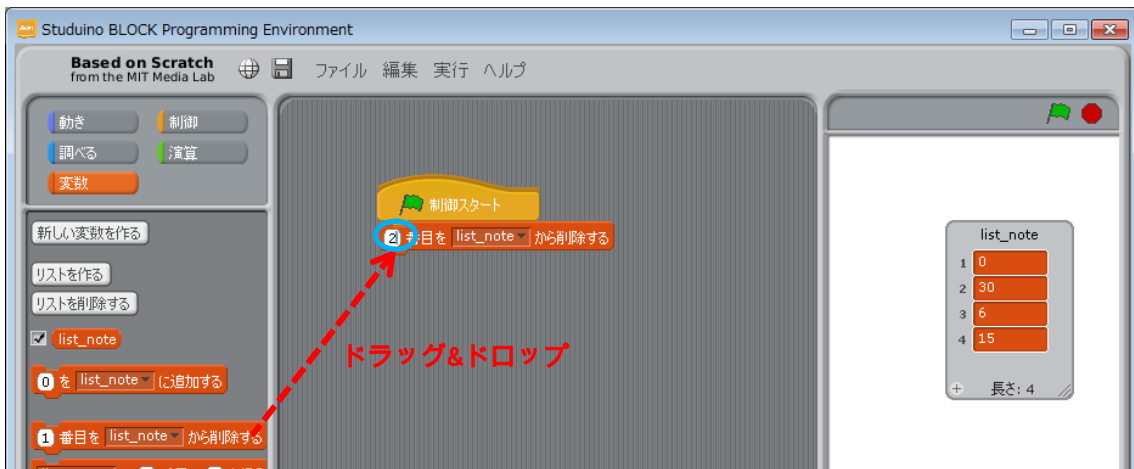
⑦ 次に、**list\_note** の 2 番目に 30 を挿入する ブロックを削除し、**list\_note** の 1 番目を 0 で置き換える ブロックをスク립トエリアにドラッグしてくっつけ、「3」番目を「6」で置き換えると設定します。



⑧  をクリックします。



- ⑨ `list_note` の 3 番目を 6 で置き換える ブロックを削除し、1 番目を `list_note` から削除する ブロックをスクリプトエリアにドラッグしてくっつけ、「2」番目を削除すると設定します。



- ⑩ 緑の旗をクリックします。

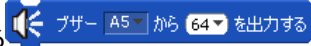


以上のようにリストを利用すれば、値を一系列で管理(追加・挿入・置換・削除)することができます。また、下記のブロックを利用してリストの情報にアクセスできます。


| ブロック                             | 処理                        |
|----------------------------------|---------------------------|
| <code>list_note</code> の 1 番目    | ブロックで指定された番号の値を取得します      |
| <code>list_note</code> の長さ       | リストの長さ(要素の数)を取得します        |
| <code>list_note</code> に 0 が含まれる | リストに指定された値が含まれるかどうかを判断します |

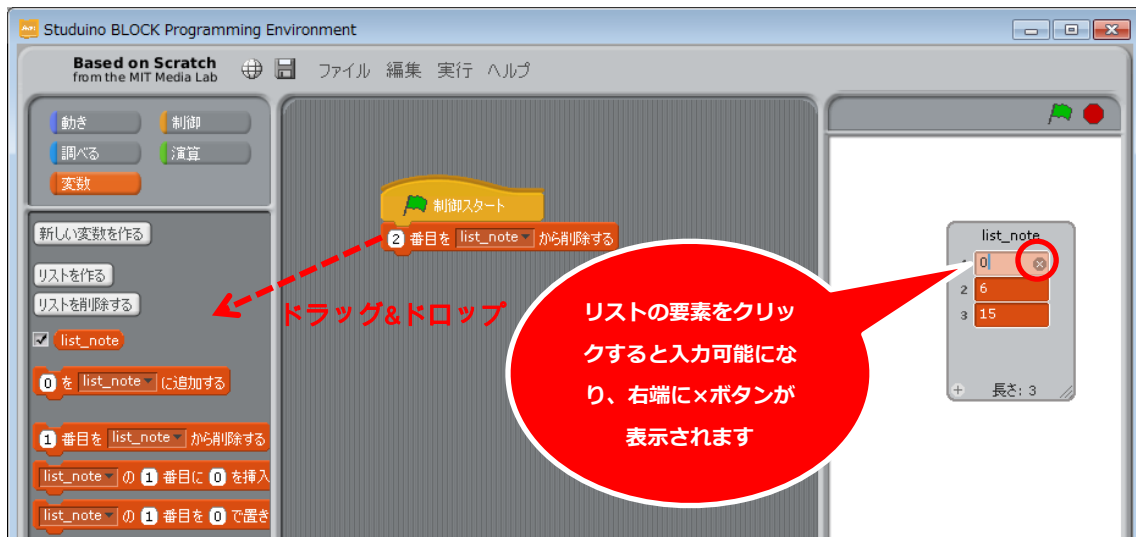
次に、3.4 連続で音を鳴らすで作成したプログラムを、リストを使って作成します。


## ●リストを使って音を鳴らす

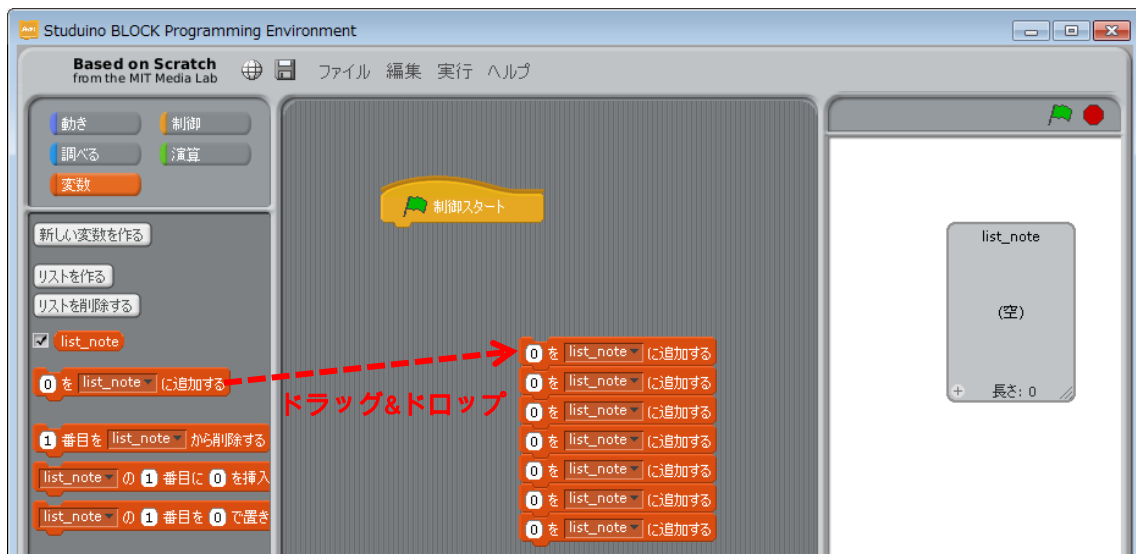
3.4 連続で音を鳴らすで作成したプログラムは、下図のブロックを音の高さを変えて、7回実行しています。音の高さの情報をリストで管理して、リストから  ブロックに音の高さを設定することで、メロディを作りやすくなります。



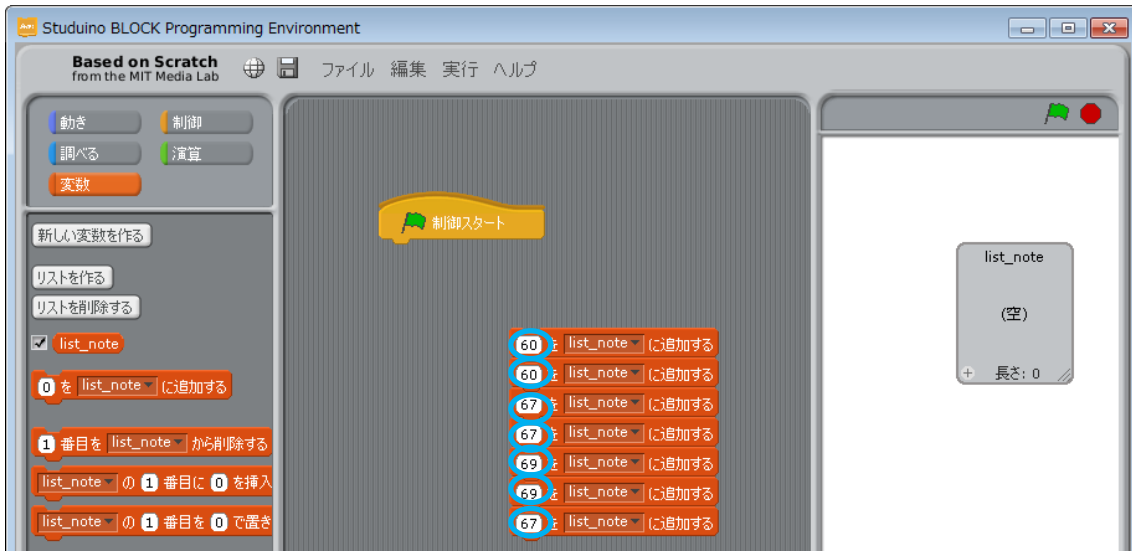
- ⑪  ブロックを削除し、リストの要素をクリックし、右端に出る×ボタンをクリックしてリストからすべての要素を削除します。



- ⑫  ブロックを7つドラッグしそれぞれくっつけます。

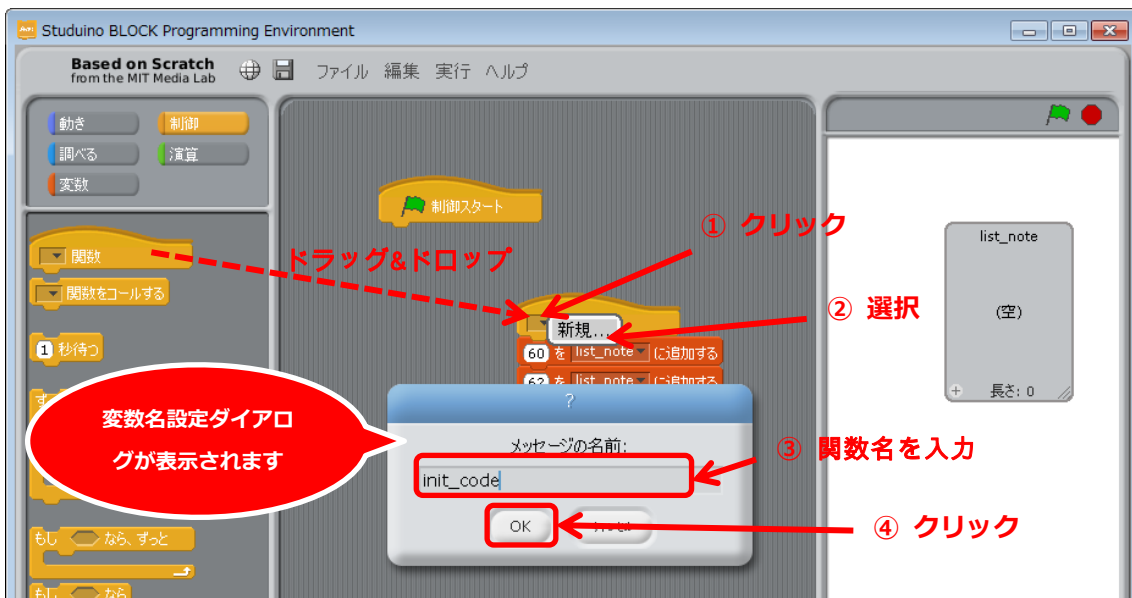


- ⑬ **0** を `list_note` に追加するブロックに値をそれぞれ設定します。



ここで設定した値は、音の高さの値になります。

- ⑭ 「制御」パレットから `関数` ブロックをドラッグし、⑬で作成したブロックの上にくっつけます。 `関数` ブロックの▼をクリックし、「新規」を選択して、関数名「init\_code」を入力します。



以上で、リストに「きらきらぼし」の最初のメロディを登録する関数 `init_code` が完成しました。 `関数` ブロックにくっつけられたブロックは、 `関数をコールする` ブロックが実行されるタイミングで実行されます。今回の場合、 `init_code` `関数をコールする` ブロックが実行されるときに `init_code` `関数` ブロックが実行され、リストへのコードの登録が行われます。

- ⑮ 「制御」パレットから **init\_code をコールする** ブロックをドラッグし **制御スタート** ブロックにくっつけ、▼をクリックし、init\_code 関数を選択します。



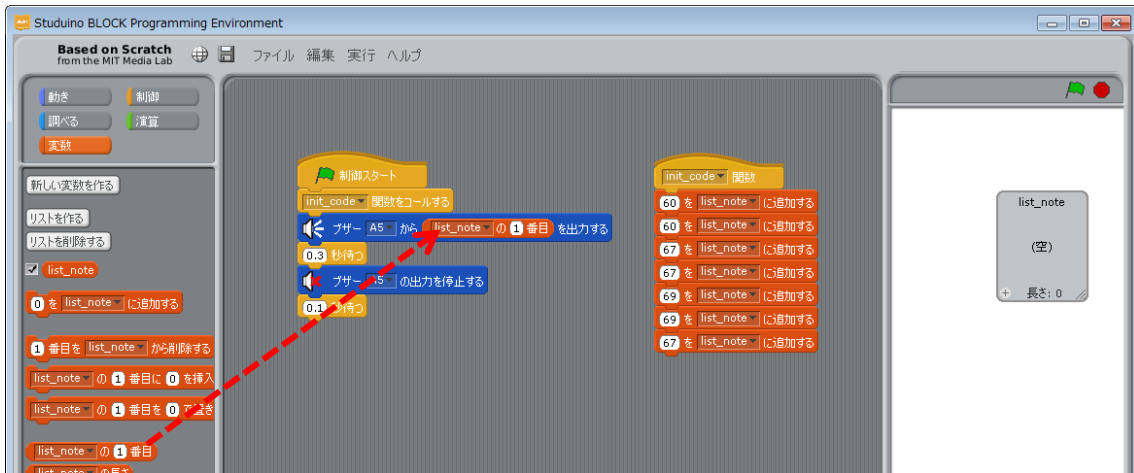
- ⑯ 「動き」パレットから **プザー A5 から 60 を出力する** ブロックと **プザー A5 の出力を停止する** ブロックを下にくっつけます。



- ⑰ 「制御」パレットから **1 秒待つ** ブロックをくっつけ、0.3 と 0.1 にそれぞれ設定します。



- ⑱ 「変数」パレットから `list_note` の 1 番目 ブロックを `ブザー A5 から 60 を出力する` ブロックにくっつけます。



- ⑲ 「変数」パレットの「新しい変数を作る」ボタンをクリックし、変数 `number` を作成します。



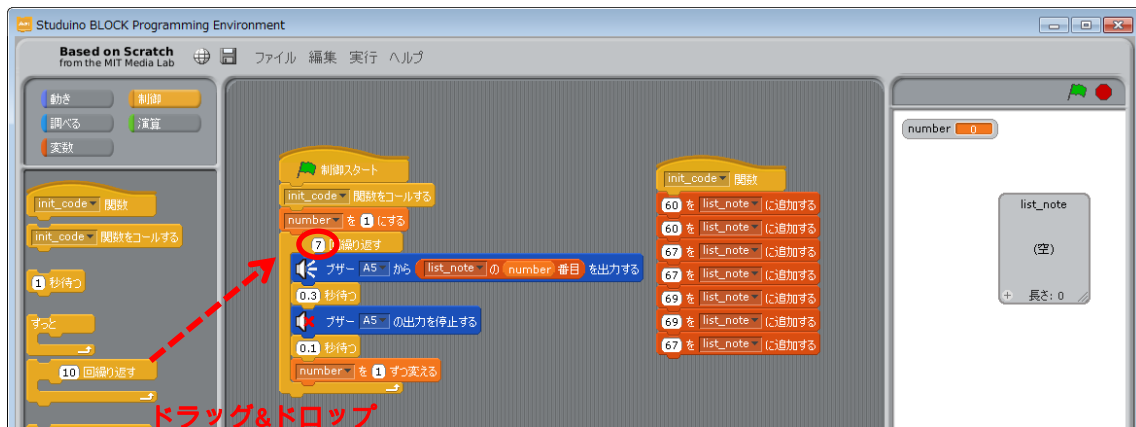
- ⑳ 作成された「変数」ブロックを下図の位置にそれぞれくっつけます。

`number` を 0 にする ブロックの設定値を 1 に設定します。

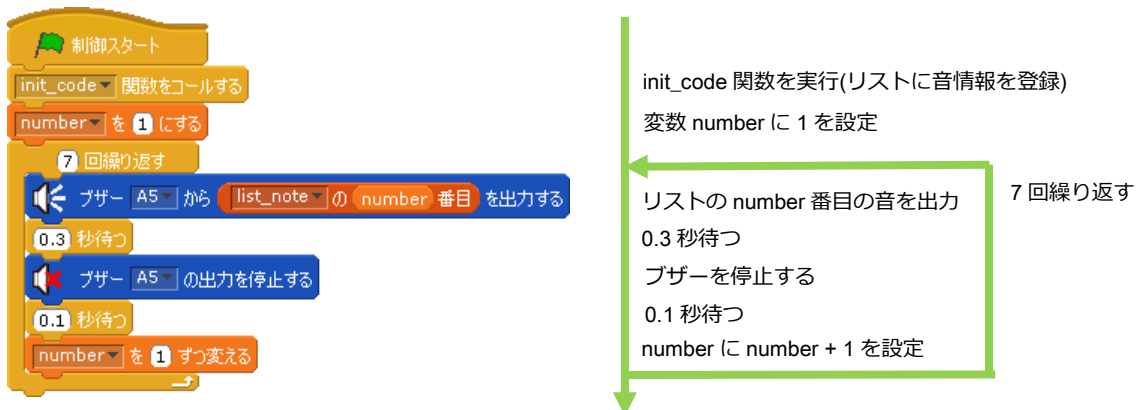




④ 「制御」パレットの  → ブロックをくっつけます。7 に設定します。



以上で、リストを使って、ブザーから「きらきらぼし」の最初のメロディを鳴らすプログラムが作成できました。作成したプログラムは下記のような処理を行います。

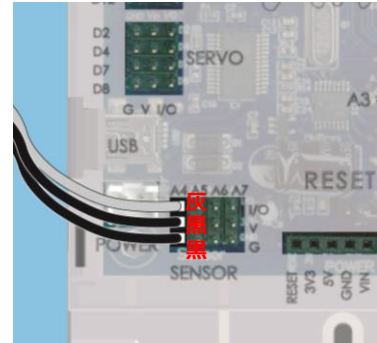
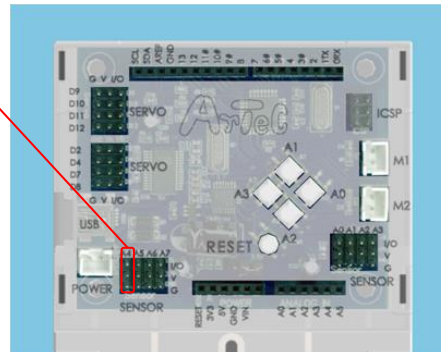
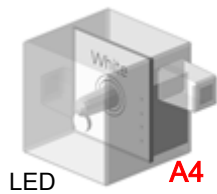


1 回のループ処理ごとに変数 number の値を 1 ずつ増やし、変数 number を使って、リストに登録された音を順番に出力します。

Studuino 基板と PC を USB コードで接続し、メニューバーの「実行」から「プログラム作成・転送」を選択してください。転送が完了すると、ブザーから「きらきらぼし」の最初のメロディが出力されます。

## 4. プッシュスイッチとセンサー

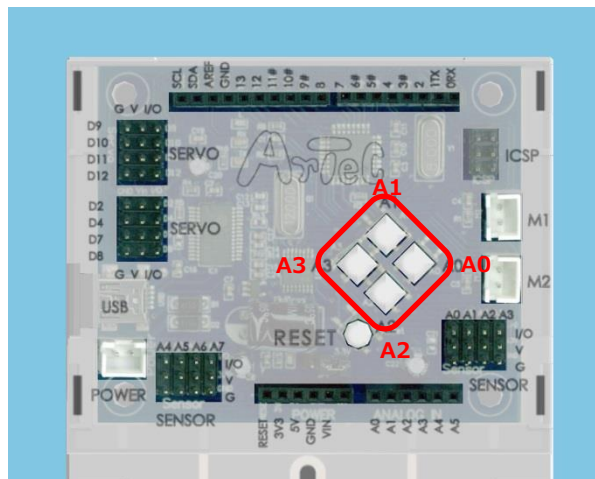
プッシュスイッチやセンサーを使ったプログラムを通して、プッシュスイッチやセンサーの使い方、条件ブロックの使い方、テストモードについて学びます。また、プログラムに LED を使用しますので、LED を Studuino 基板のセンサー/LED/ブザー用コネクタの A4 に接続します。



センサー接続コードの向きに注意！  
上側に灰色のコードがくるように。

### 4.1. プッシュスイッチ

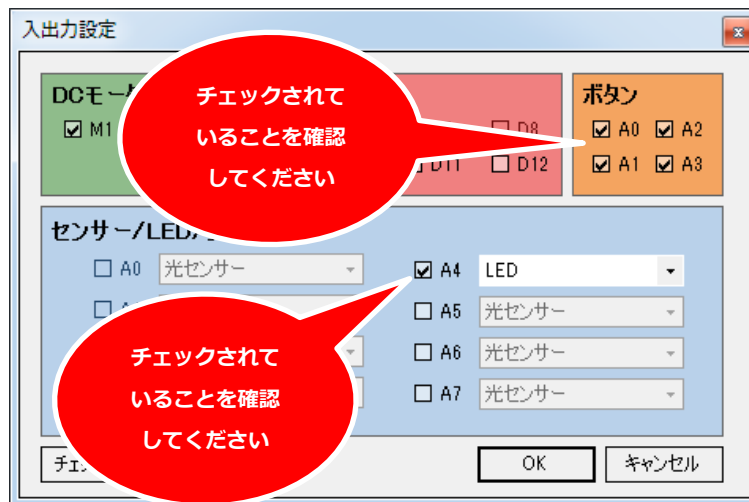
プッシュスイッチは、Studuino 基板に付いているボタンです。A0～A3 の信号ピンが割り当てられています。



#### 4.1.1. 入出力ポート情報の設定

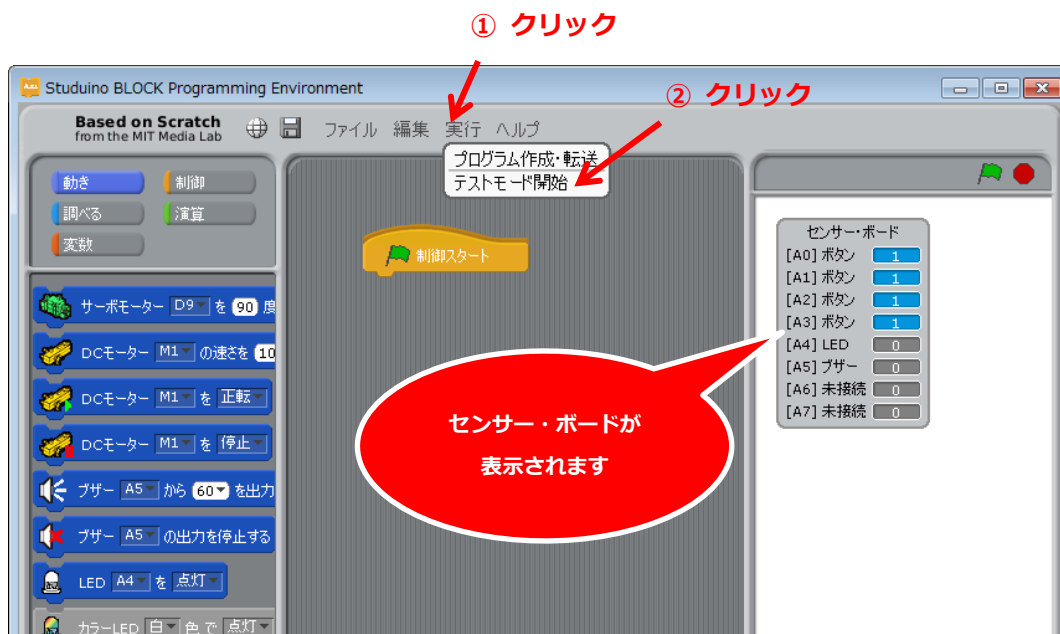
ブロックプログラミング環境に Studuino 基板のポート情報を設定します。ブロックプログラミング環境のメニューバーの「編集」から「入出力設定...」を選択して、入出力設定ダイアログボックスを開きます。

入出力設定ダイアログボックスの「ボタン」エリアの A0～A3 がチェックされていることと、「センサー/LED/ブザー」エリアの A4 に LED が選択されていることを確認してください(チェックされていない場合は、チェックを入れてください)。

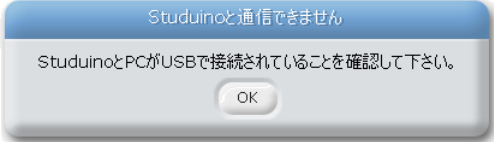
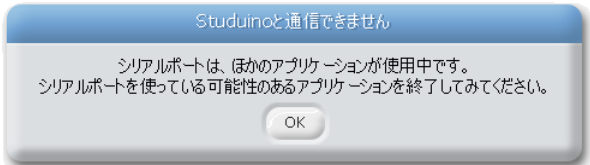


#### 4.1.2. プッシュスイッチの動作確認

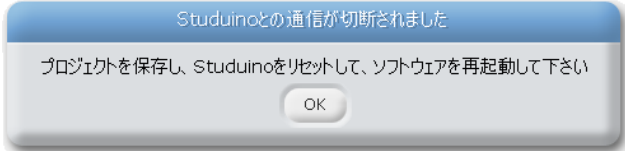
プッシュスイッチの動作を確認します。Studuino 基板と PC を USB コードで接続し、メニューバーの「実行」から「テストモード開始」を選択してください。



下記のメッセージが表示された場合、テストモードが開始されていませんので、対策を実施し、再度「テストモード開始」を実行して下さい。

| メッセージ   | 対策  |
|---|---|
|  | <p>Stduino 基板と PC が USB コードでしっかり接続されていることを確認してください。</p>   |
|  | <p>Stduino 基板と通信しているアプリケーションがある場合は、そのアプリケーションを終了してください。通信しているアプリケーションがない場合は、USB コードを Stduino 基板から抜いて再度接続し、Stduino 基板のリセットボタンを押してください。</p> |

テストモード中に誤ってリセットボタンを押してしまった場合や、Stduino 基板と PC の通信が切断された場合に、下記のメッセージが表示されます。

| メッセージ   | 対策  |
|---|---|
|  | <p>テストモードを一度終了することで再度プログラミングを続けることができますが、通信状態が不定のため、メッセージにあるように本ソフトウェアを再起動することを推奨します。</p> |

プッシュスイッチは、押されていない状態の時は 1 で、押されている状態の時は 0 になります。Studuino 基板のプッシュスイッチ A0～A3 を押して、センサー・ボードの対応するボタンの値が変化することを確認してください。

| センサー・ボード |   |
|----------|---|
| [A0] ボタン | 0 |
| [A1] ボタン | 1 |
| [A2] ボタン | 1 |
| [A3] ボタン | 1 |
| [A4] LED | 0 |
| [A5] ブザー | 0 |
| [A6] 未接続 | 0 |
| [A7] 未接続 | 0 |

Studuino 基板のプッシュスイッチを押すと 0 に変化します

確認ができれば、メニューバーの「実行」から「テストモード終了」を選択し、テストモードを終了してください。

### 4.1.3. プログラミング

プッシュスイッチを使ってボタンを押したら LED を点滅するプログラムを作成します。

本プログラムで **もし...なら** ブロックを使ったプログラムを学習します。

- ① 「動き」パレットから **LED A4 を点灯** ブロックをドラッグします。



- ② 「制御」パレットから **1 秒待つ** ブロックを下にくっつけます。



- ③ **LED A4 を点灯** ブロック上で右クリックし、「複製」を選択します。



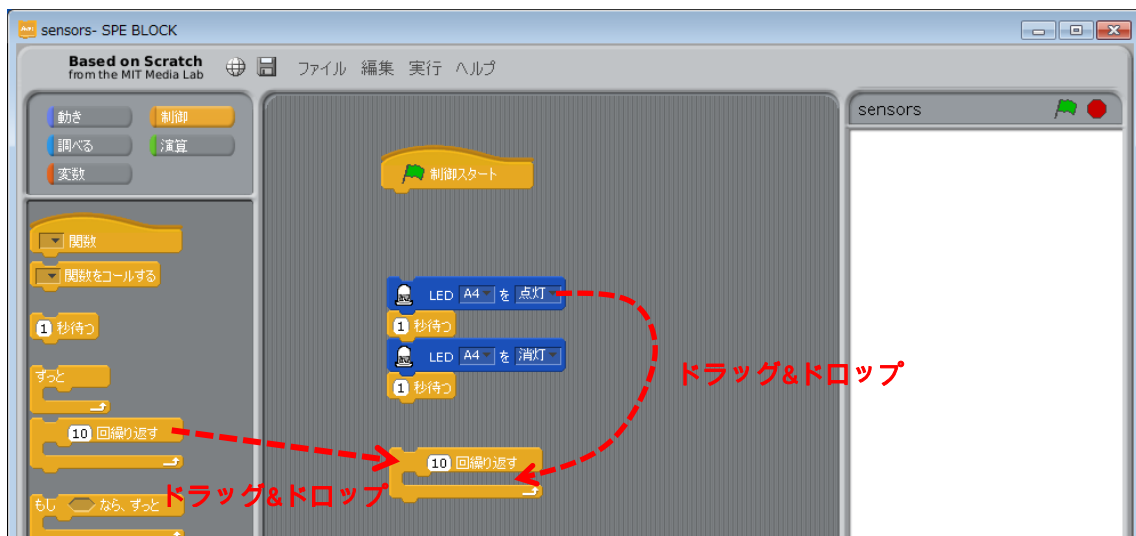
- ④ 複製した「LED A4 を点灯」ブロックの「点灯」を「消灯」に変更します。



- ⑤ 複製したブロックを複製元のブロックとくっつけます。



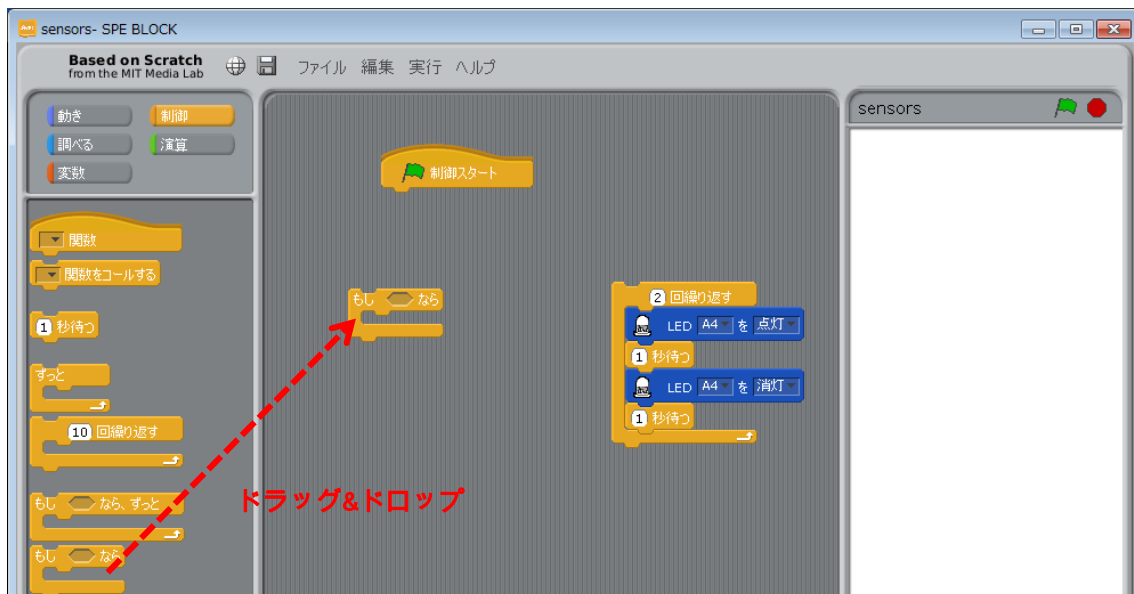
- ⑥ 「制御」パレットから「10 回繰り返す」ブロックをドラッグします。④で作成したブロックを「10 回繰り返す」ブロックの開いているところに入れます。



⑦ 回数を2回に設定します。



⑧ 「制御」パレットから「もし〜なら」ブロックをドラッグします。

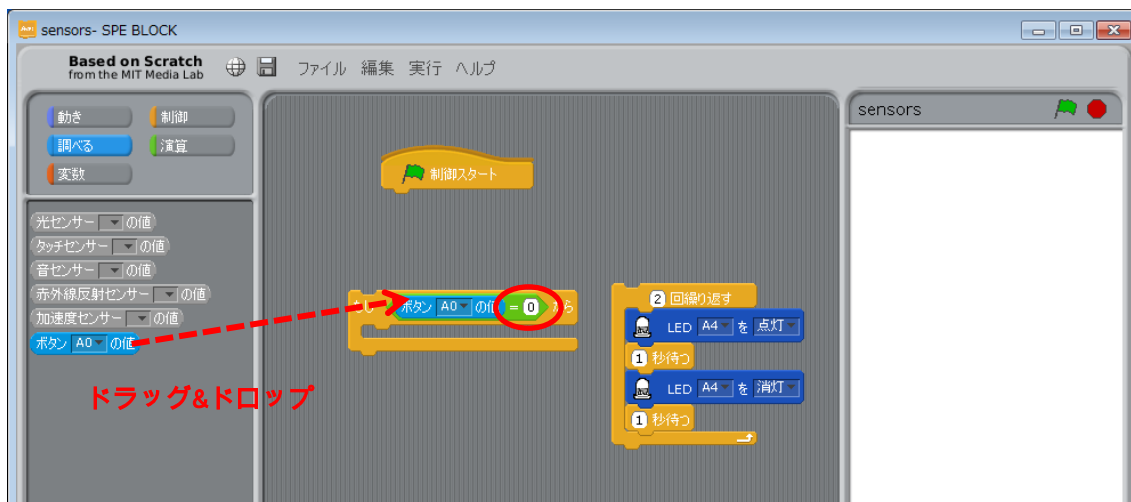


⑨ 「演算」パレットから「=」ブロックをくっつけます。





- ⑩ 「調べる」パレットから「ボタン A0 の値」ブロックを「=」ブロックの左辺にくっつけ、右辺に 0 を設定します。





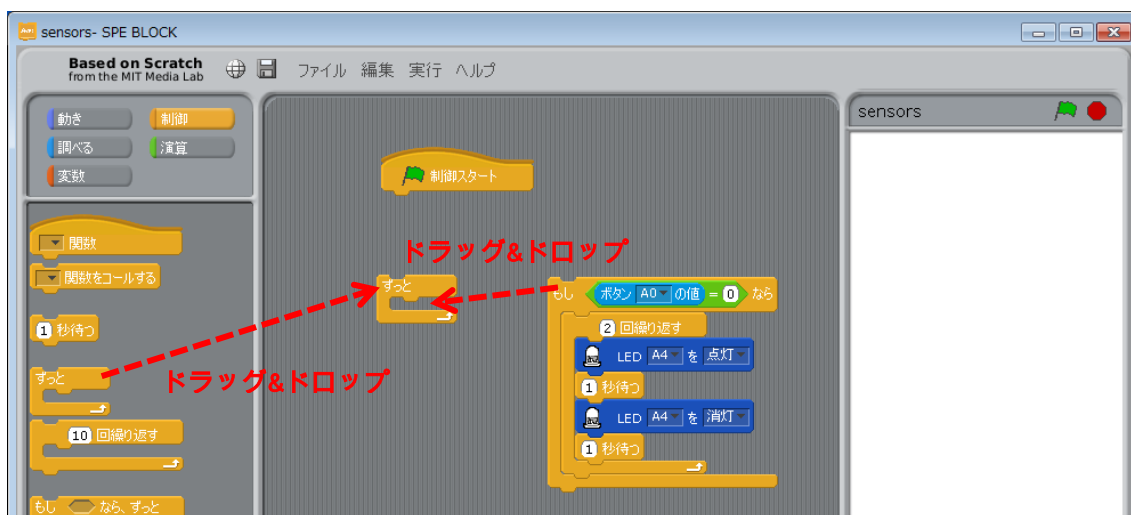
- ⑪ ⑦で作成したブロックを「もし」ブロックの開いている所に入れます。



作成したブロックは、A0 のプッシュスイッチが押されている(0)場合、LED を 2 回点滅する処理を表します。



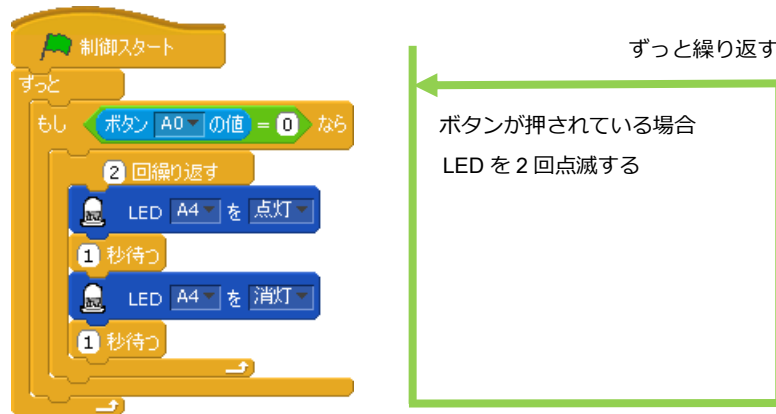
⑫ 「制御」パレットから  ブロックをドラッグします。⑪で作成したブロックを  ブロックの開いているところに入れます。



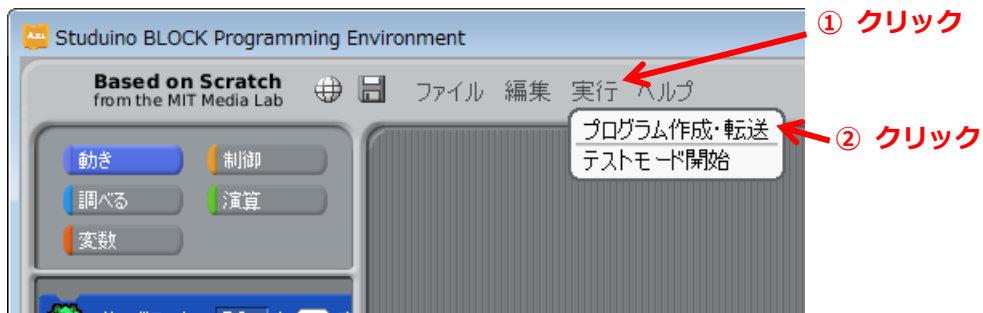
⑬ ⑫で作成したブロックを  ブロックにくっつけます。



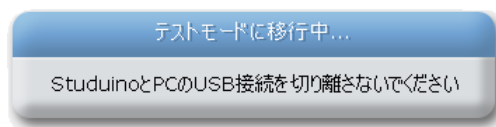
以上で、プッシュスイッチ A0 が押されたときに LED が 2 回点滅するプログラムができました。



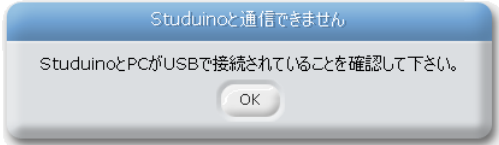
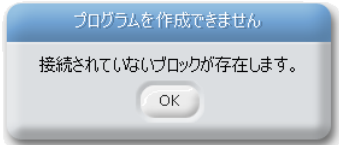


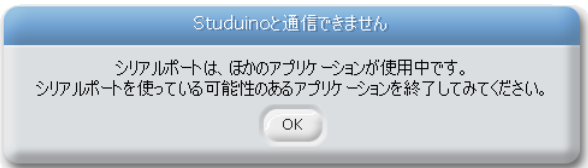
Studuino 基板 と PC を USB コードで接続し、メニューバーの「実行」から「プログラム作成・転送」を選択してください。



下のメッセージが表示され消えると、プログラムが Studuino 基板に正常に転送されたこととなります。転送が完了すると、プッシュスイッチ A0 を押すと LED が 2 回点滅します。



下記のメッセージが表示された場合、プログラムが Studuino 基板に転送されていませんので、対策を実施し、再度「プログラム作成・転送」を実行して下さい。

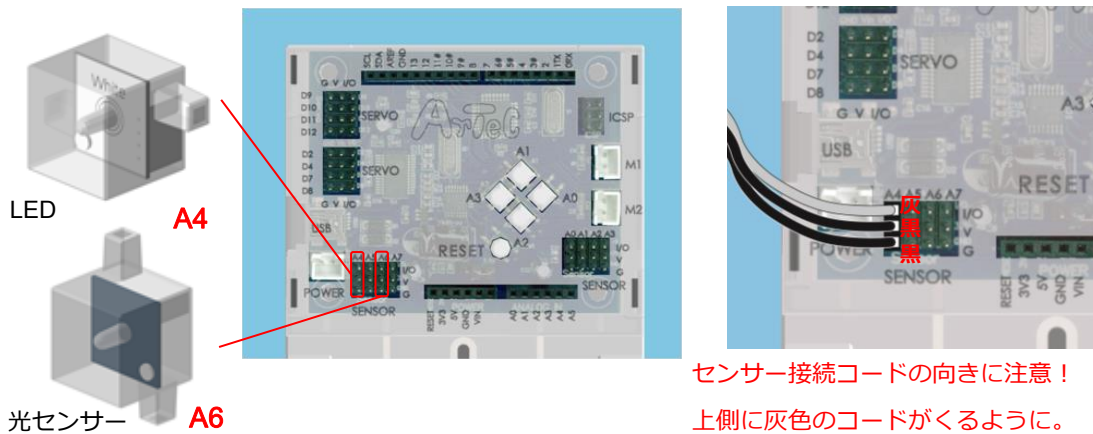
| メッセージ  | 対策  |
|--|---|
|   | <p>Studuino 基板と PC が USB コードでしっかり接続されていることを確認して下さい。</p>   |
|   | <p> ブロックに接続されていないブロックがあります。 に繋がれない余分なブロックを削除して下さい。</p> |
|  | <p>Studuino 基板と通信しているアプリケーションがある場合は、そのアプリケーションを終了してください。通信しているアプリケーションがない場合は、USB コードを Studuino 基板から抜いて再度接続し、Studuino 基板のリセットボタンを押して下さい。</p>   |

## 4.2. 光センサー

光センサーは明るさを計測するセンサーです。

### 4.2.1. Studuino 基板と光センサーの接続

Studuino 基板のセンサー/LED/ブザー用コネクターの A4 に LED を、A6 に光センサーを接続します。



### 4.2.2. 入出力ポート情報の設定

ブロックプログラミング環境に Studuino 基板のポート情報を設定します。ブロックプログラミング環境のメニューバーの「編集」から「入出力設定...」を選択して、入出力設定ダイアログボックスを開きます。入出力設定ダイアログボックスの「センサー/LED/ブザー」エリアの A6 をチェックして光センサーを選択して下さい。また、「センサー/LED/ブザー」エリアの A4 に LED を選択して下さい。



### 4.2.3. 光センサーの動作確認

光センサーの値をチェックします。Studuino 基板と PC を USB コードで接続し、メニューバーの「実行」から「テストモード開始」を選択してください。テストモードが開始されるとセンサー・ボードが表示され、「[A6] 光センサー」が表示されます。

光センサーの値は、明るさを 0~100 で表し、暗い状態の時は 0 で、明るい状態の時は 100 になります。Studuino 基板に接続した光センサーを手で囲うなどして、センサー・ボードに表示される光センサーの値が変化することを確認してください。


| センサー・ボード   |    |
|------------|----|
| [A0] ボタン   | 1  |
| [A1] ボタン   | 1  |
| [A2] ボタン   | 1  |
| [A3] ボタン   | 1  |
| [A4] LED   | 0  |
| [A5] 未接続   | 0  |
| [A6] 光センサー | 96 |
| [A7] 未接続   | 0  |

明るさを変えることで  
光センサーの値が変化します

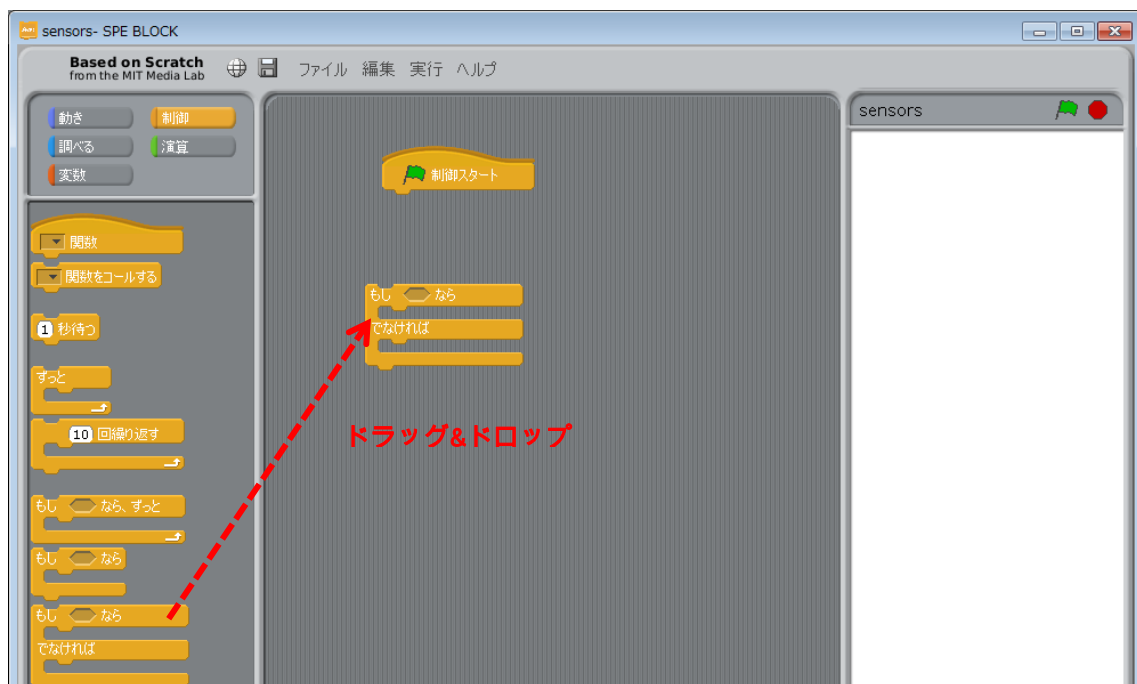
確認ができれば、メニューバーの「実行」から「テストモード終了」を選択し、テストモードを終了してください。

### 4.2.4. プログラミング

光センサーを使って暗くなったら LED を点灯するプログラムを作成します。


本プログラムで  ブロックを使ったプログラムを学習します。

① 「制御」パレットから  ブロックをドラッグします。






- ② 「演算」パレットから  ブロックをくっつけます。



- ③  ブロックの左辺に、「調べる」パレットから  の値 ブロックをくっつけ、右辺に 50 を設定します。



- ⚠ 光センサーの値は、部屋の明るさによって変わるため、右辺の値は、センサー・ボードの値を参考に決めてください。

- ④ 「動き」パレットから  LED A4 を点灯 ブロックを  ブロックの開いているところに入れます。下の段の  LED A4 を点灯 ブロックの設定を消灯に設定します。



作成したブロックは、暗い場合(光センサーの値が 50 より小さい場合)、LED を点灯し、明るい場合(50 より大きい場合)、LED を消灯します。



- ⑤ ブロックをスクリプトエリアにドラッグ&ドロップします。④で作成したブロックを ブロックの開いているところに入れます。



- ⑥ ⑤で作成したブロックを ブロックにくっつけます。







以上で、暗いときにLEDが点灯し、明るくなるとLEDが消灯するプログラムができました。



⑦ 明るさで LED が点灯・消灯するかを確かめます。Studuino 基板と PC が USB コードで接続されていることを確認し、メニューバーの「実行」から「テストモード開始」を選択してください。

⑧ テストモードに移行したら、をクリックしてください。



 **テストモード中は、Studuino 基板と PC を接続する USB コードを抜かないでください**  
実行中のブロックが白枠で囲まれて表示されます。センサー・ボードを見ながら光センサーを手で囲うなどして、暗くなった時に LED が点灯することを確認してください。

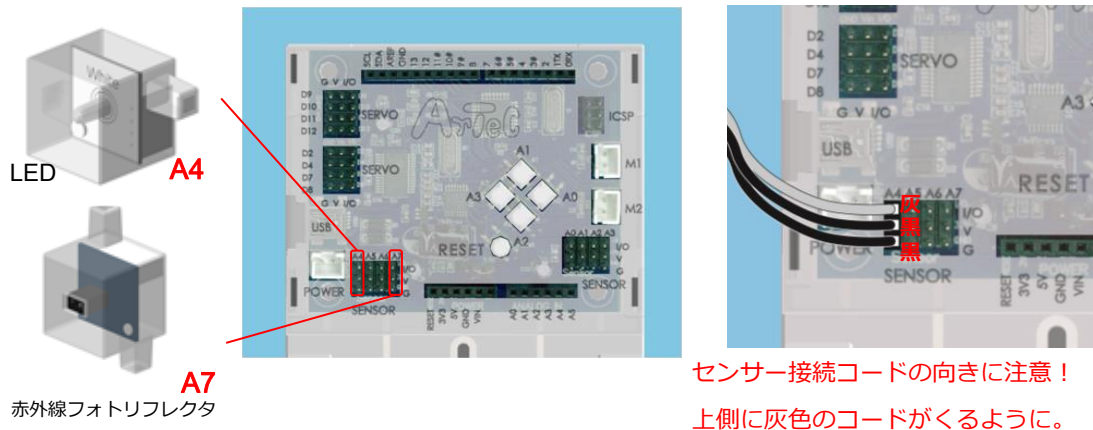
メニューバーの「実行」から「プログラム作成・転送」を選択し、Studuino 基板にプログラムを転送することで、USB コードを外し、電池ボックスでプログラムを実行することができます。

### 4.3. 赤外線フォトリフレクタ

赤外線フォトリフレクタは、照射した赤外線の影響を受光することで、物体との距離を計測します。

#### 4.3.1. Studuino 基板と赤外線フォトリフレクタの接続

Studuino 基板のセンサー/LED/ブザー用コネクタの A4 に LED、A7 に赤外線フォトリフレクタを接続します。



#### 4.3.2. 入出力ポート情報の設定

ブロックプログラミング環境に Studuino 基板のポート情報を設定します。ブロックプログラミング環境のメニューバーの「編集」から「入出力設定...」を選択して、入出力設定ダイアログボックスを開きます。入出力設定ダイアログボックスの「センサー/LED/ブザー」エリアの A7 をチェックして赤外線フォトリフレクタを選択して下さい。また、「センサー/LED/ブザー」エリアの A4 に LED を選択して下さい。



### 4.3.3. 赤外線フォトリフレクタの動作確認

赤外線フォトリフレクタの値をチェックします。Studuino 基板と PC を USB コードで接続し、メニューバーの「実行」から「テストモード開始」を選択してください。テストモードが開始されるとセンサー・ボードが表示され、「[A7] 赤外線フォトリフレクタ」が表示されます。

赤外線フォトリフレクタの値は、照射した赤外線の反射を受光する量を 100 段階で数値化しており、対象物が近づくにつれて 100 に近づき、6~10mmの位置が最大値となり、さらに近づくると再び数値は小さくなります。

また、対象物の色や表面性状によっても異なった数値となります。(黒い物体を近づけた時より、白い物体を近づけたときの方が値は大きくなります。)

Studuino 基板に接続した赤外線フォトリフレクタに物体を近づけたり話したりして、センサー・ボードの赤外線フォトリフレクタの値が変化することを確認してください。


| センサー・ボード         |    |
|------------------|----|
| [A0] ボタン         | 1  |
| [A1] ボタン         | 1  |
| [A2] ボタン         | 1  |
| [A3] ボタン         | 1  |
| [A4] LED         | 0  |
| [A5] 未接続         | 0  |
| [A6] 未接続         | 0  |
| [A7] 赤外線フォトリフレクタ | 40 |

対象とセンサーの距離を変えることで値が変化します

確認ができたなら、メニューバーの「実行」から「テストモード終了」を選択し、テストモードを終了してください。

#### 4.3.4. プログラミング

赤外線フォトリフレクタを使って物体が近づいたら LED を点滅するプログラムを作成しま

す。本プログラムで  ブロックを使ったプログラムを学習します。

- ① 「動き」パレットから  ブロックをドラッグします。



- ② 「制御」パレットから  ブロックを下にくっつけます。



- ③  ブロック上で右クリックし、「複製」を選択します。



- ④ 複製した LED A4 を点灯 ブロックの「点灯」を「消灯」に変更します。



- ⑤ 複製したブロックを複製元のブロックとくっつけます。






- ⑥ 「制御」パレットから  ブロックをドラッグします。



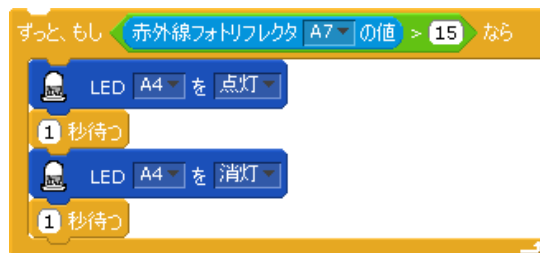
⑦ 「演算」パレットから  ブロックをくっつけます。



⑧  ブロックの左辺に「調べる」パレットから  ブロックをくっつけ、右辺に 15 を設定します。⑤で作成したブロックを  ブロックの開いているところに入れます。



作成したブロックは、物体が近くにある場合(赤外線フォトリフレクタの値が 15 より大きい場合)、LED を点滅します。



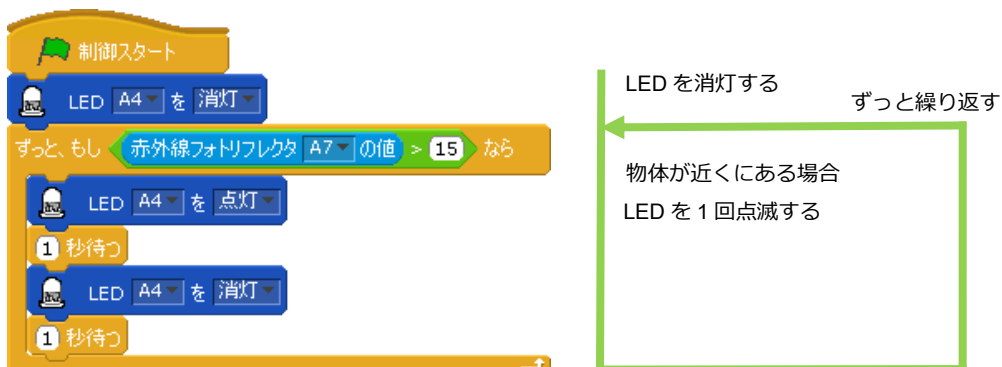
- ⑨ LEDの初期状態を消灯に設定するために、「動き」パレットから「LED A4 を点灯」ブロックを「制御スタート」ブロックにくっつけ、消灯に設定します。




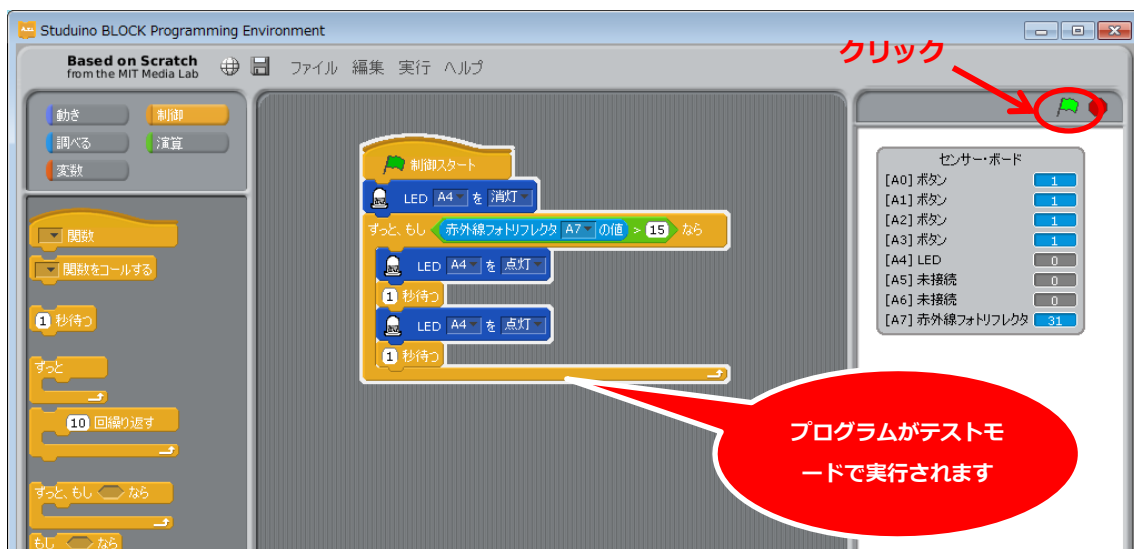
- ⑩ ⑧で作成したブロックを「LED A4 を消灯」ブロックの下にくっつけます。



以上で、物体が近づくとLEDが点滅するプログラムができました。



- ⑪ 物体を近づけて LED が点滅することを確認めます。Studuino 基板と PC が USB コードで接続されていることを確認し、メニューバーの「実行」から「テストモード開始」を選択してください。
- ⑫ テストモードに移行したら、をクリックしてください。



**⚠ テストモード中は、Studuino 基板と PC を接続する USB コードを抜かないでください**  
センサー・ボードを見ながら赤外線フォトフレクタに物体を近づけるなどして、15 よりも値が大きくなった時に LED が点滅することを確認してください。

メニューバーの「実行」から「プログラム作成・転送」を選択し、Studuino 基板にプログラムを転送することで、USB コードを外し、電池ボックスでプログラムを実行することができます。

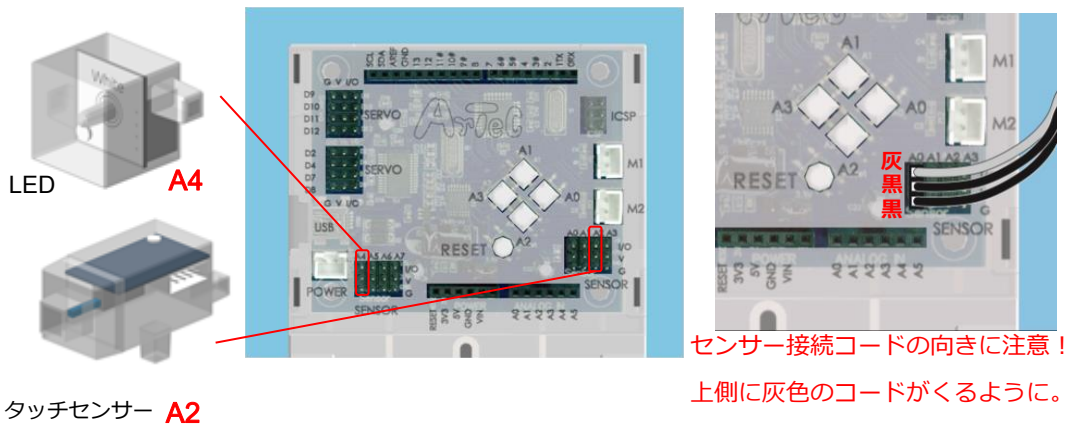


## 4.4. タッチセンサー

タッチセンサーは、ボタンが押されているかどうかの状態を判定します。物体との衝突などの判定に使用します。

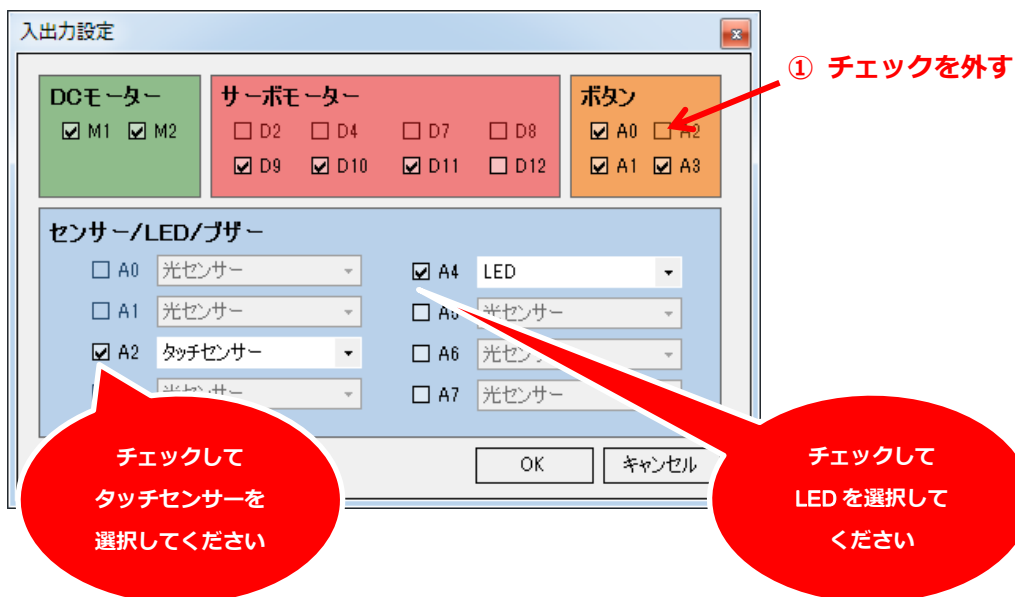
### 4.4.1. Studuino 基板とタッチセンサーの接続

Studuino 基板のセンサー/LED/ブザー用コネクターの A4 に LED、A2 にタッチセンサー接続します。



### 4.4.2. 入出力ポート情報の設定

ブロックプログラミング環境に Studuino 基板のポート情報を設定します。ブロックプログラミング環境のメニューバーの「編集」から「入出力設定...」を選択して、入出力設定ダイアログボックスを開きます。入出力設定ダイアログボックスの「ボタン」エリアの A2 のチェックを外し、「センサー/LED/ブザー」エリアの A2 をチェックしてタッチセンサーを選択して下さい。また、「センサー/LED/ブザー」エリアの A4 に LED を選択して下さい。



#### 4.4.3. タッチセンサーの動作確認

タッチセンサーの値をチェックします。Studuino 基板と PC を USB コードで接続し、メニューバーの「実行」から「テストモード開始」を選択してください。テストモードが開始されるとセンサー・ボードが表示され、「[A2] タッチセンサー」が表示されます。

タッチセンサーは、押されていない状態の時は 1 で、押されている状態の時は 0 になります。タッチセンサーを押して、センサー・ボードのタッチセンサーの値が変化することを確認してください。



確認ができたなら、メニューバーの「実行」から「テストモード終了」を選択し、テストモードを終了してください。

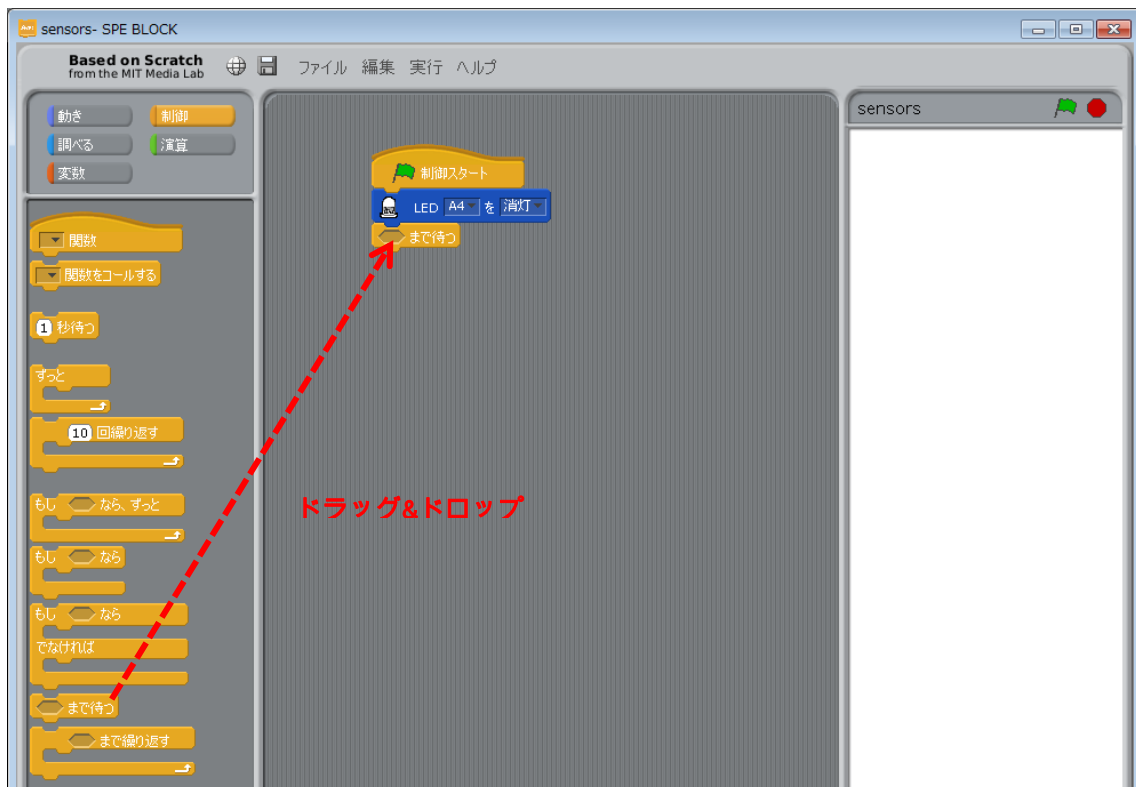
#### 4.4.4. プログラミング

タッチセンサーを使ってタッチセンサーが押されたら LED を点灯するプログラムを作成します。本プログラムで「まで待つ」ブロックを使ったプログラムを学習します。

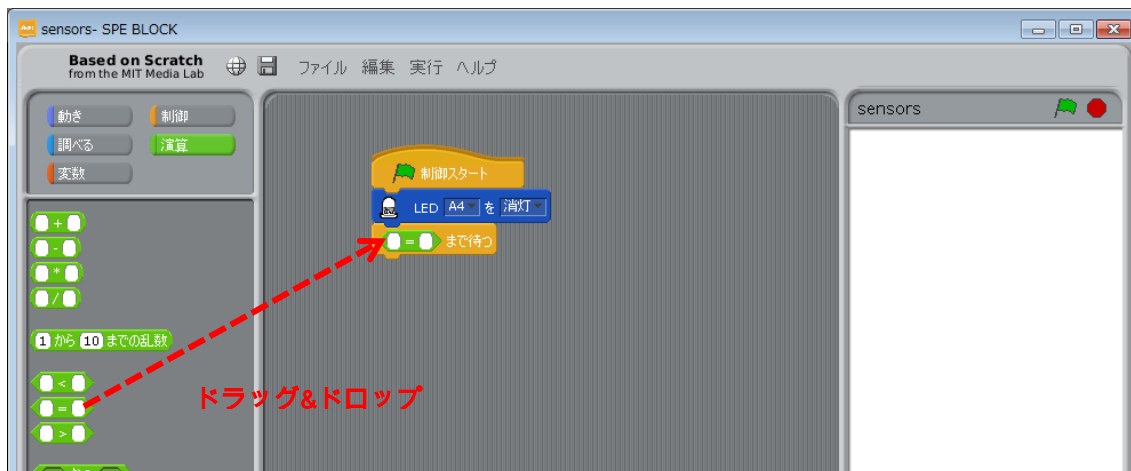
- ① 「動き」パレットから「LED A4 を点灯」ブロックを「制御スタート」ブロックにくっつけて、消灯に設定します。



- ② 「制御」パレットから「まで待つ」ブロックを「LED A4 を点灯」ブロックにくっつけます。



- ③ 「演算」パレットから「=」ブロックをくっつけます。



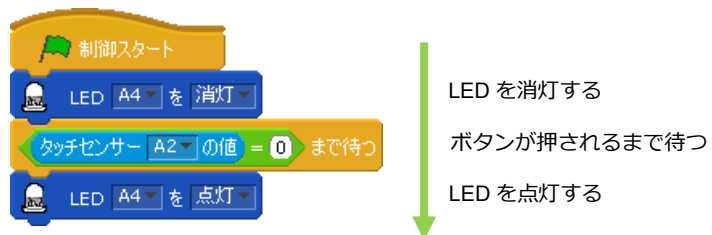
- ④ 「調べる」パレットから「タッチセンサー A2 の値」ブロックを「=」ブロックの左辺にくっつけ、右辺に 0 を設定します。



- ⑤ 「動き」パレットから「LED A4 を 点灯」ブロックをくっつけます。



以上で、タッチセンサーのボタンが押されると LED が点灯するプログラムができました。



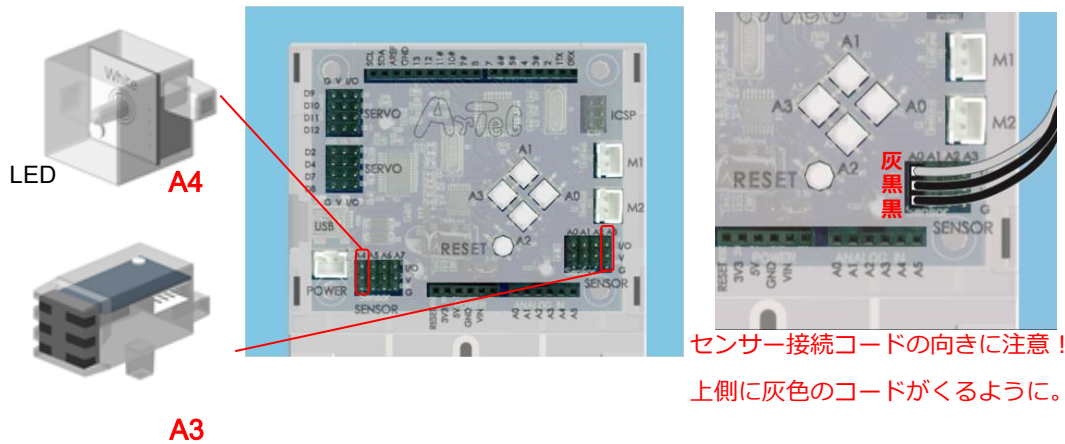
Studuino 基板と PC が USB コードで接続されていることを確認し、メニューバーの「実行」から「プログラム作成・転送」を選択してください。タッチセンサーを押すと LED が点灯することを確認してください。

## 4.5. 音センサー

音センサーは、音の大きさを計測するセンサーです。

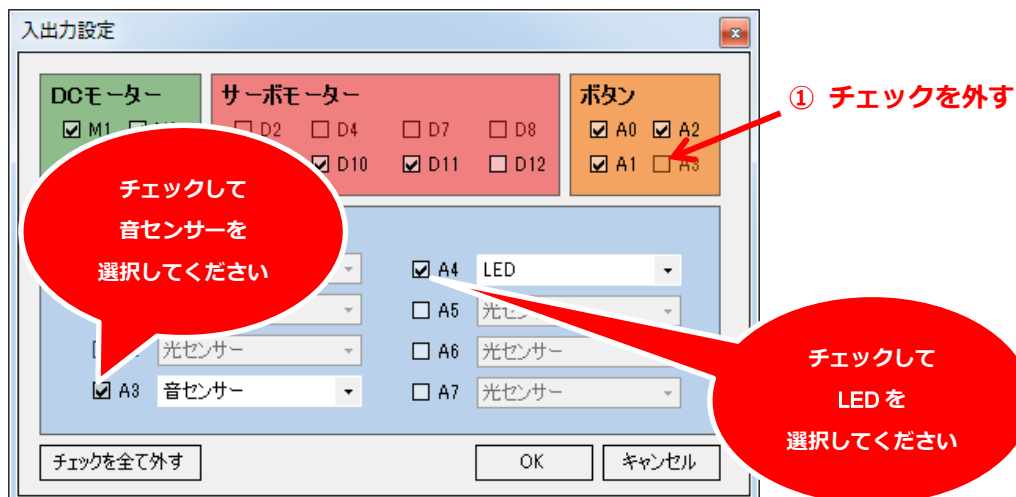
### 4.5.1. Studuino 基板と音センサーの接続

Studuino 基板のセンサー/LED/ブザー用コネクターの A4 に LED、A3 に音センサーを接続します。



### 4.5.2. 入出力ポート情報の設定

ブロックプログラミング環境に Studuino 基板のポート情報を設定します。ブロックプログラミング環境のメニューバーの「編集」から「入出力設定...」を選択して、入出力設定ダイアログボックスを開きます。入出力設定ダイアログボックスの「ボタン」エリアの A3 のチェックを外し、「センサー/LED/ブザー」エリアの A3 をチェックして音センサーを選択して下さい。また、「センサー/LED/ブザー」エリアの A4 に LED を選択して下さい。



### 4.5.3. 音センサーの動作確認

音センサーの値をチェックします。Studuino 基板と PC を USB コードで接続し、メニューバーの「実行」から「テストモード開始」を選択してください。テストモードが開始されるとセンサー・ボードが表示され、「[A3] 音センサー」が表示されます。


音センサーの値は、音の大きさを 0~50 で表し、無音状態の時は 0 で、音が大きくなるにつれて 50 に近づきます。Studuino 基板に接続した音センサーの近くで音を出したり、音センサーに息を吹きかけたりして、センサー・ボードの音センサーの値が変化することを確認してください。

| センサー・ボード   |    |
|------------|----|
| [A0] ボタン   | 1  |
| [A1] ボタン   | 1  |
| [A2] ボタン   | 1  |
| [A3] 音センサー | 29 |
| [A4] LED   | 0  |
| [A5] 未接続   | 0  |
| [A6] 未接続   | 0  |
| [A7] 未接続   | 0  |

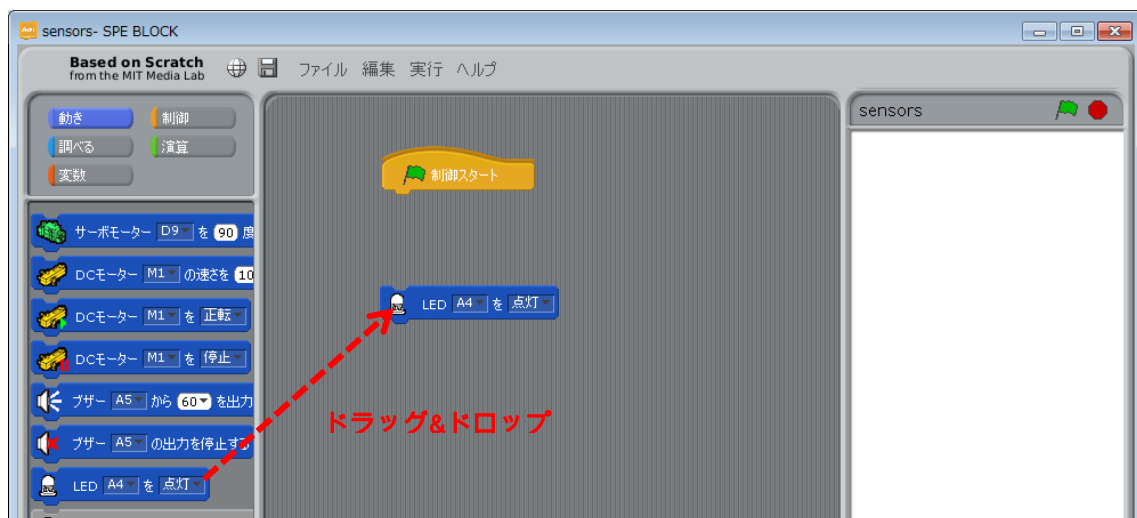
音の大きさや  
吹きかける息の強さを  
変えることで値が変化します

確認ができれば、メニューバーの「実行」から「テストモード終了」を選択し、テストモードを終了してください。

### 4.5.4. プログラミング

音センサーを使って息を吹きかけたら、点滅していた LED が消灯するプログラムを作成します。本プログラムで  ブロックを使ったプログラムを学習します。

① 「動き」パレットから  ブロックをドラッグします。



② 「制御」パレットから「1秒待つ」ブロックをくっつけます。



③ 「LED A4 を点灯」ブロック上で右クリックし、「複製」を選択します。



④ 複製した「LED A4 を点灯」ブロックの「点灯」を「消灯」に変更します。

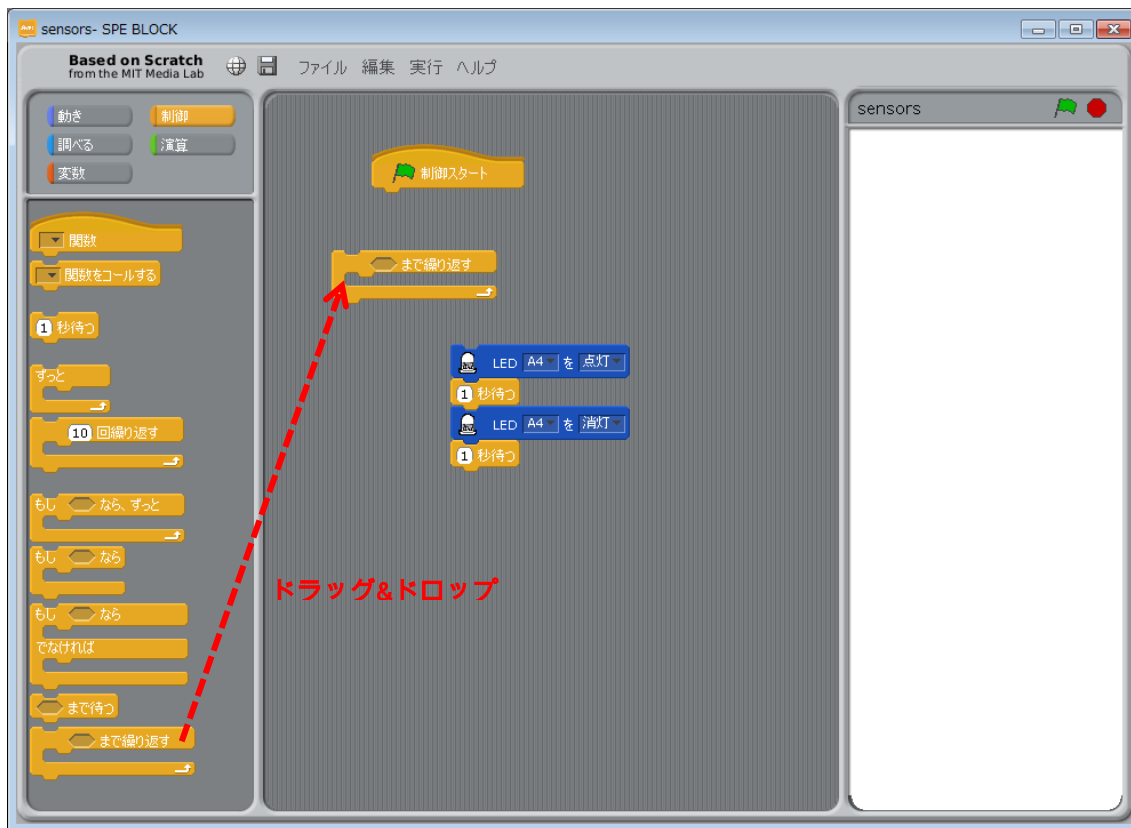




⑤ 複製したブロックを複製元のブロックとくっつけます。




⑥ 「制御」パレットから  ブロックをドラッグします。



- ⑦ 「演算」パレットから  ブロックをくっつけます。



- ⑧  ブロックの左辺に、「調べる」パレットから  ブロックをくっつけ、右辺に 30 を設定します。



- ⑨ ⑤で作成したブロックを  ブロックの開いているところに入れます。



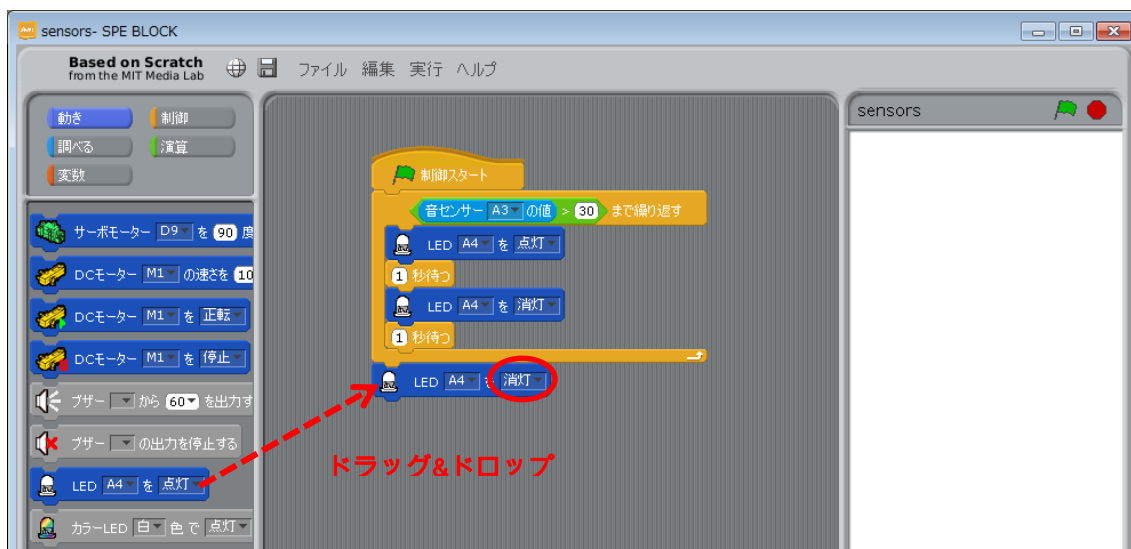
作成したブロックは、息が吹きかけられるまで(音センサーの値が 30 より大きくなるまで)、LED を点滅します。



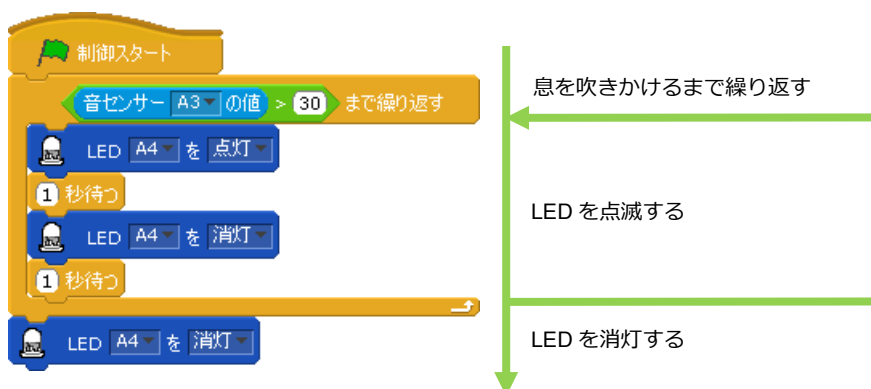
⑩ ⑨で作成した作成したブロックを **制御スタート** ブロックにくっつけます。




⑪ 「動き」パレットから **LED A4 を点灯** ブロックをくっつけ、消灯に設定します。

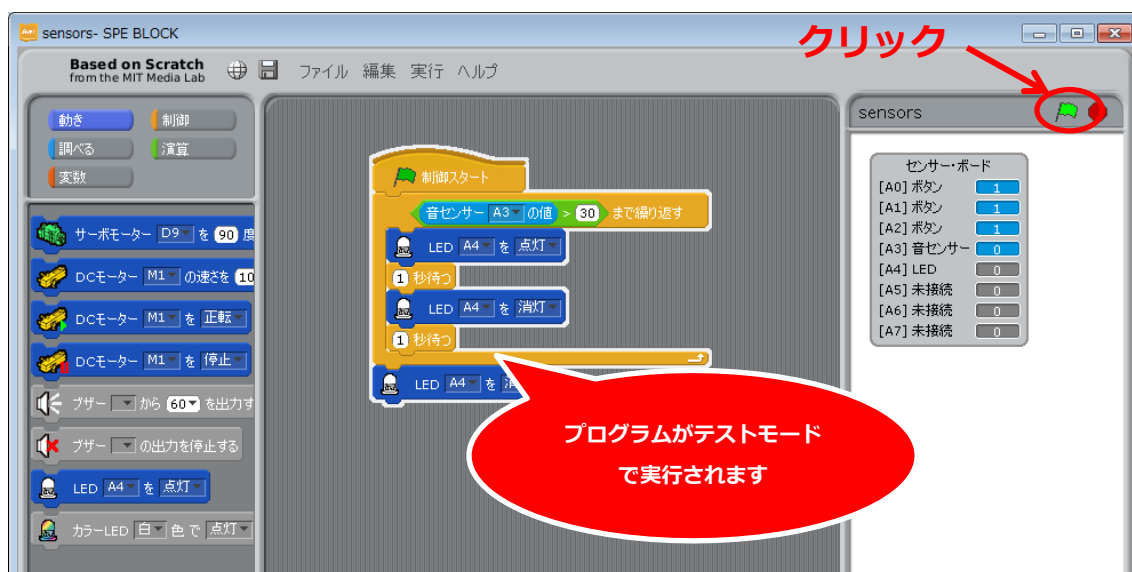


以上で、音センサーに息を吹きかけたら、点滅していた LED が消灯するプログラムができました。



⑫ 息を吹きかけることで LED が消灯することを確認めます。Studuino 基板と PC が USB コードで接続されていることを確認し、メニューバーの「実行」から「テストモード開始」を選択してください。

⑬ テストモードに移行したら、 をクリックしてください。



**⚠ テストモード中は、Studuino 基板と PC を接続する USB コードを抜かないでください**  
 センサー・ボードを見ながら音センサーに長めに息を吹きかけ、30 よりも値が大きくなった時に LED が消灯することを確認してください。

メニューバーの「実行」から「プログラム作成・転送」を選択し、Studuino 基板にプログラムを転送することで、USB コードを外し、電池ボックスでプログラムを実行することができます。