Studuino

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



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	じめに	. 1
1.1.	Studuino のセットアップ	. 1
1.2.	接続方法	. 1
2. LE	D	.4
2.1.	Studuino 基板と LED との接続	. 4
2.2.	入出カポート情報の設定	. 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. ブナ	ヂー	15
3.1.	Studuino 基板とブザーの接続	15
3.2.	入出カポート情報の設定	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	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	
4.4. 夕	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.	入出カポート情報の設定	53
4.4.3.	タッチセンサーの動作確認	
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.	入出カポート情報の設定	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

<u>http://www.artec-kk.co.jp/studuino</u>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/ブザー用コネクターに接続します。

※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。



※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 のコネクター使用時は、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ー(M1、M2)に接続します。

※DC モーターのコネクター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ーM1 とサーボモーター用コネクターD2,D4 は同時に使用できません。 ※DC モーター用コネクターM2 とサーボモーター用コネクター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ーに接続します。



※DC モーター用コネクターM1 とサーボモーター用コネクターD2,D4 は同時に使用できません。 ※DC モーター用コネクターM2 とサーボモーター用コネクター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ー(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ー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ー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、Studuino 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. Studuino 基板と LED との接続

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. 入出カポート情報の設定

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 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。

				① チ /	エックを入れる
入出力設定					
DCモーター ② M1 ② M2	サーボモーター □ D2 □ D4 ☑ D9 ☑ D10	□ D7 □ ☑ D11 □	1 D8 ☑ A0 1 012 ☑ A1	☑ A2 ☑ A3	
センサー/LED/	ブザー				② クリック
A0 光セン	/サー 👻	☑ A4 光	センサー		
A1 光セン	/サー 👻	□ A5 光1	センサー チセンサー		
A2 光セン	/サー 👻	□ A6 赤	センサー 外線フォトリフレクタ		📕 ③ LED を選択
□ A3 光セン	/サー 👻	🗆 A7 🎵	D f		
				_	
チェックを全て外す			OK 🔫	シセル	
			4	ОК ボ :	タンをクリック

以上でブロックプログラミング環境に Studuino 基板の入出カ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

LED Mail を 感灯 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

LED Marke 点灯 ブロックをスクリプトエリア内にドラッグし、 クにくっつけます。





※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。 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 になります。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
Studuinoと通信できません	Studuino 基板と PC が USB コードでしっかり
StuduinoとPCがUSBで接続されていることを確認して下さい。 OK	接続されていることを確認してください
プログラムを作成できません	▶ ● 制御スタート ブロックに接続されていない
按統之れにしないりロックが1471至します。 OK	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
Studuinoと通信できません	ンがある場合は、そのアプリケーションを終了
シリアルボートは、ほかのアプリケーションが使用中です。	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
 ブロックを使って、リピート処理を学習します。



② LED M-1 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。



 後製されたブロックを¹⁹⁰⁰ブロックの下にくっつけます。



 ④ 「制御」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、プログ ラム全体を囲う形でドロップします。



⑤ 2つ目の (10 Martine LED Martine LED Martine LED Martine LED Martine LED Jack Strategy June 2000 「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。

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

Studuino BLOCK Programming En	vironment	
Based on Scratch from the MIT Media Lab	■ ファイル 編集 実行 ヘルプ	
	 ▲ 利却スタート 3-2 ▲ 10 回線の返す ▲ 10 回線の返す ● LED A4 を 強打 ● LED A4 を 強打 ● LED A4 を 強打 ● ドラッグ&ドロップ 	



します。

Studuino BLOCK Programming Environment		
Based on Scratch from the MIT Media Lab 🕀 🔡 ファイル 編集 実行 ヘルプ		
動き 制御 調べる 演算 変数 第 開設 第 第 約 第 7 第 7 第 7 第 7 第 7 第 1 1 1 <	10回線の返す ブロックの中に ドロップします 10回線の返す ドロップします 10回線の返す 10回線の返す	





● ##X2-ト 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 ^{● 制御スタート}ブロックにくっつけます。

Studuino BLOCK Programmin	g Environment	- • •
Based on Scratch from the MIT Media Lab	∌ 🖥 ファイル 編集 実行 ヘルプ	
10回繰り返す		

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。



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 10 回点滅します。

10 回繰り返す 次に、 ⊇ブロックのリピート回数を変更することで点滅回数を変えます。 10 回繰り返す (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ー(POWER)に接続し、電源 を ON にし、LED が 1 秒間隔で指定回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。





リセットボタン

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。

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 ^{「「「」」 を ① 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、^{「「」」」を} □ にする ブロ

ックの値を 0.5 に設定します。



(ime)ブロックを
 ジャクレックにくっつけます。

📴 Studuino BLOCK Programming Environment	
Based on Scratch 🕀 启 ファイル 編集 実行 ヘルプ	

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。



上側に灰色のコードがくるように。

3.2. 入出カポート情報の設定

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 「サームを「から Gor を出力する」 ブロックと 「サームを「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。

「動き」パレットから
 「サー A5 から GOD を出力する
 ブロックを
 ブロックにく
 っつけます。

start- SPE BLOCK		
Based on Scratch from the MIT Media Lab	🕀 🖥 ファイル 編集 実行 ヘルプ	
	 ▲ 細胞スタート ● ほく ブザー A5 から 60 を出力する ● ほ ● ドラッグ&ドロップ ● サブ 	start 🇯

② 「 プサー 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¹⁰⁰⁰⁰ブロックを^{く ブザーA5¹⁰⁶ GA² ^{2出力する}ブロックの下にくっつけます。}

start- SPE BLOCK			
Based on Scratch from the MIT Media Lab	, 🕀 📙 ファイル 編集 実行 ヘルプ		
動き 制御 調べる 運算 変数 関数 1000000000000000000000000000000000000	 ▲ #御以タート ● 「サー A5」から 64● を出力する ● お待つ ドラッグ&ドロップ 	start	

④ 「動き」パレットの^{(すサーA5)の出力を停止する}ブロックを^{(1)秒(5)}ブロックの下にくっつけま

す。

start- SPE BLOCK		
Based on Scratch from the MIT Media Lab	ファイル 編集 実行 ヘルプ	
動き 制御 調べる 演算 変数 第二日 ひこモーター D9 を 90 度 ひこモーター M1 の速さを 10 ひこモーター M1 の速さを 10 ひこモーター M1 を 正転 ロン DCモーター ロン DCモーター ロン た 原虹 ブザー A5 から 30 デ き出力 ブザー A5 の出力を信息工 レビD で 点気工	ま	Start 🎘 •

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¹⁰⁶⁰ブロックを追加します。

Studuino BLOCK Programming Environment	
Based on Scratch From the MIT Media Lab	
② 1000 ブロックの待ち時間を 0.3 秒と 0.1 秒に設定します。	,
Studuino BLOCK Programming Environment	- • ×
Based on Scratch from the MIT Media Lab	
③	どされたブロ
Studuino BLOCK Programming Environment Based on Scratch from the MIT Media Lab Tアイル 編集 実行 ヘルプ	

 from the MIT Media Lab
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1)
 (1

④ ③の手順を繰り返しブロックの組を7つ作ります。最後の ります(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 び 25 AS MA 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ことができます。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

 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 ③ ③ Iiit_note
(3) づロックを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 へをクリックすると、 クリ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。

🤤 Studuino BLOCK Programm	ing Environment		
Based on Scratch from the MIT Media Lab	🕀 🖥 ファイル 編集 実行 ヘルプ		
動き	● 制御スタート ① を list_note - (ご追加する) 12 を list_note - (ご追加する) 15 を list_note - (ご追加する)	実行されます リストに値が 追加されます	ほうしょう (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)

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、「ist_note」の1番目に0を挿入するブロックを # ##スタート にく
 - っつけ、「2」番目に「30」を挿入すると設定します。



⑥ Ӓをクリックします。



をスクリプ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。





⑨ 「List_note」 0 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 が削除する
ブロックをスクリプ





以上のようにリストを利用すれば、値を一列で管理(追加・挿入・置換・削除)するとことが 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
の1番目	ブロックで指定された番号の値を取得します
list_note▼ の長さ	リストの長さ(要素の数)を取得します
list_note V (こ 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、リストからで プサー AS MA CAR を出力する 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。



① 2番目を list_note / から前除する 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。



	ブロックに	値をそれぞれ設定します。	
Studuino BLOCK Program	ming Environment		
Based on Scratch from the MIT Media Lab	🕀 🖥 ファイル 編集	ミ 実行 ヘルプ	
	除する 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	 制油スタート 60): list_note (ご良加する) 60): list_note (ご良加する) 67): list_note (ご良加する) 67): list_note (ご良加する) 69): list_note (ご良加する) 	▶ ● list_note (空) + 長さ:0
			II I
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 国数	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_coo	le」を入力しまで	す。	
Studuino BLOCK Program			
	ming Environment		
Based on Scratch from the MIT Media Lab	ming Environment	実行 ヘルプ	
Based on Scratch from the MIT Media Lab 動き 制御 調小る 演算 変数 2 ご 関数をコールする ご 砂待つ ジ 変数名設定グ グが表示され 6し もし なら、すっと もし なら	ming Environment ① ファイル 編集 ドラッグ&ド イアロ hます	実行 ヘルプ 創御スタート 1 クリソ ロッフ のを list_note (Ci的加する) 20 を list_note (Ci的加する) 20 を list_note (Ci的加する) 20 を list_note (Ci的加する) 20 ので (Cinnate (Cin	□ ■ ク ② 選択 (空) 単数名を入力 ④ クリック
Based on Scratch from the MIT Media Lab 動き 制御 調べる 演算 変数 運算 一 関数 運算 変数 1 砂待つ 1 砂待つ 1 砂待つ 1 砂待つ 1 ジガ表示さく 1 もし から、すっと もし から 以上で、リストに	ming Environment ① ファイル 編集 ドラッグ&ド ・ドラッグ&ド ・ ・ ドラッグ&ド ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	実行 ヘルブ	 〇〇〇〇 ク (2) 選択 (2) 選択 (2) 選択 (2) 選択 (2) 選択 (2) 選択 (3) (2) (3) (3) (3) (3) (3) (3) (3) (3) (3) (3
Based on Scratch from the MIT Media Lab 動き 制御 調小30 演算 変数 演算 変数 調数をコールする 動き 期御 関数をコールする 調査 り待つ が表示され もし から、すっと もし から、すっと もし から、マッと した。 関数	Image Environment Image Environment	実行 ヘルブ ま行 ヘルブ リロンタート 1 クリッ ロップ ロップ 第規 (3) を list_note (5) を加える (5) を list_note (5) を加える (5) の最初のメロディを登録する くっつけられたブロックは、	 ・ ・
Based on Scratch from the MIT Media Lab 前き 前き 第二 第二 <td>ming Environment ・ ファイル 編集 ・ ドラッグ&ド ・ ドラッグ&ド ・ ドラッグ&ド ・ 「きらきらぼし ・ ブロックに・ ・ ・ ・ で実行されます</td> <td>実行 ヘルブ 第行 ヘルブ 1 クリッ 1 クリッ 1 クリッ 1 の最初のメロディを登録する くっつけられたブロックは、 5 今回の場合、</td> <td> □ ■ ■ 2 選択 (2) 選択 (2) 選択 (2) 選択 (2) 選択 (2) 選択 (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)</td>	ming Environment ・ ファイル 編集 ・ ドラッグ&ド ・ ドラッグ&ド ・ ドラッグ&ド ・ 「きらきらぼし ・ ブロックに・ ・ ・ ・ で実行されます	実行 ヘルブ 第行 ヘルブ 1 クリッ 1 クリッ 1 クリッ 1 の最初のメロディを登録する くっつけられたブロックは、 5 今回の場合、	 □ ■ ■ 2 選択 (2) 選択 (2) 選択 (2) 選択 (2) 選択 (2) 選択 (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)

- ⑤ 「制御」パレットから「「^{関数をコールする}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 🤨 フサー 🔤 カル5 💷 を出力する ブロックと 👎 フサー 🔤 の出力を停止する ブロッ

クを下にくっつけます。



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



- ③ 「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。

🔁 Studuino BLOCK Programming Environment		
Studuino BLOCK Programming Environment Based on Stratch ① ① ⑦	Int_code BBX () & list_note (;)\$1075 () & list_note (;)\$1075	(空) () () () () () () () () () () () () ()

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。

📴 Studuino BLOCK Programming Environment	
Based on Scratch 🕀 🔚 ファイル 編集 実行 ヘルプ from the MIT Media Lab	
bits #1日 INC 2015年17 INE CODE ● (日本の10 ● 日本の10 ● 日本の1	▶ ● Iist_note (堂) + 長さ:0



20「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。
Studuino BLOCK Programming Environment		
Bensed on Scratch Bensed on Scratch ●	int_code Bist () & list_note (cishu)d5 () & list_note (cishu)d5	number 1 list_note (空) + 長さ:0

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/ブザー用コネクターの A4 に接続します。



上側に灰色のコードがくるように。

A3

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. 入出カポート情報の設定

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。



① クリック

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
Studuinoと通信できません	Studuino 基板と PC が USB コードで
StuduinoとPCがUSBで接続されていることを確認して下さい。	しっかり接続されていることを確認
	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
Studuinoと通信できません	リケーションを終了してください。
シリアルボートは、ほかのアプリケーションが使用中です。 シリアルボートは、ほかのアプリケーションが使用中です。	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。

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、Studuino 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
	テストモードを一度終了することで
Studuinoとの通信が切断されました	再度プログラミングを続けることが
プロジェクトを保存し、Studuinoをリセットして、ソフトウェアを再起動して下さい	できますが、通信状態が不定のため、
ОК	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。

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。



確認ができたら、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。

4.1.3. 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「「「」」フロックを使ったプログラムを学習します。
- 「動き」パレットから
 「動き」パレットから
 「動き」パレットから



「制御」パレットから¹⁹⁶⁷ブロックを下にくっつけます。



③ 「 「 「 「 」 「 」 「 」 「 」 「 」 「 」 を選択します。



4	複製した -	^{LED A4 を 感灯} ブロックの「点灯」を「消灯」に変更します。	
-	sensors- SPE BLOCK		- • •
	Based on Scratch from the MIT Media Lab	∌ 🖥 ファイル 編集 実行 ヘルプ	
ſ	動き 制御	sensors	/ A
H	調べる 演算		
	変数		
			- 1
			- 1
		LED A4 を 点灯	
	1 秒待つ		
	تم <mark>ك</mark>		- 1
		1 赵待つ	- 1

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。

🔤 sensors- SPE BLOCK		
Based on Scratch from the MIT Media Lab 🕀 🔚 ファイル	/ 編集 実行 ヘルプ	
(動き)単調(なる) (実数) (実数) (実数) (目数をコールする) (目数をコールする) (目数をコールする) (目数をコールする) (目の回線り)返す	sensors ま 加加スタート	
	10 回繰り返す	

「制御」 パレットから 6 10 回繰り返す -**)**` . – クを

────ブロックの開いているところに入れます	o
------------------------	---

🔤 sensors- SPE BLOCK		
Based on Scratch from the MIT Media Lab 🕀 居 ファイル 編	錬 実行 ヘルプ	
	・ ・ ・ ・ ・ </th <th></th>	

⑦ 回数を2回に設定します。



⑧ 「制御」パレットから ブロックをドラッグします。

Sensors- SPE BLOCK		- • ×
Based on Scratch from the MIT Media Lab 🕀 📄 ファイル 編集 実行 ヘルプ		
動き 割前 調べる 注意 変数 「「関数 「「関数 「「」」」」 「「関数 「」」」 「「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」」 「」」」 「」」」」 「」」」 「」」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」」 「」」	Sensors 2 回線9週2 2 正日 A4 を 点灯 1 投稿 2 正日 A4 を 満灯 2 世紀 1 投稿 1 投稿 1 投稿 1 投稿 1 投稿 1 投稿 1 しん 4 を 満灯 1 投稿 1 しん 4 を 満灯 1 しん 4 を 満び 1 しん 4 を 満ひ 1 しん 4 を 一を 満ひ 1 しん 4 を 一を 一	

⑨ 「演算」パレットから()」)ブロックをくっつけます。



「調べる」パレットから
 「調べる」パレットから
 「ロックを
 「コックの左辺にくっつけ、右辺
 に0を設定します。



① で作成したブロックを

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。



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。

ſ	🐸 Studuino BLOCK Programi	ming Enviro	onment			① クリック
	Based on Scratch from the MIT Media Lab	⊕ ⊟	ファイル	編集	実行(ルプ	
	動き 制御				プログラム作成・転送 テストモード開始	~② クリック
	[調べる] 演算					
	愛 数					

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になります。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
Studuinoと通信できません	Studuino 基板と PC が USB コードでしっかり
StuduinoとPCがUSBで接続されていることを確認して下さい。 OK	接続されていることを確認して下さい。
プログラムを作成できません	メボリンタート ブロックに接続されていない
技統されていないワロックが存在します。 OK	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。
	Studuino 基板と通信しているアプリケーショ
Studuinoと通信できません	ンがある場合は、そのアプリケーションを終了
シリアルボートは、ほかのアプリケーションが使用中です。	してください。通信しているアプリケーション
ンリアルホートを1まっている可用を1生めめるアフリケーンヨンを#ネ子してみてくたろい。 OK	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。

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. 入出カポート情報の設定

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。



確認ができたら、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。

4.2.4. 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 本プログラムで 本プ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、光センサー Men one ブロックをくっつけ、

右辺に 50 を設定します。

🔤 sensors- SPE BLOCK	
Based on Scratch 🕀 园 ファイル 編集 実行 ヘルプ	
動き 制御 副べる 注算 支放 第 制御スタート 光センサー▲60000 トラッグ&ドロップ タッチセンサー●の値 60000 音センサー●の値 50000 市外線反射センサー●の値 50000 市水線反射センサー●の値 50000 市水線反射センサー●の値 50000 市力 70000	sensors 🎘 🍋

- 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 AITを 点灯」}ブロックを いているところに入れます。下の段の^{■ LED AITを 点灯}ブロックの設定を消灯に設定します。

Sensors- SPE BLOCK	
Based on Scratch from the MIT Media Lab 🕀 🔚 ファイル 編集 実行 ヘルプ	
sensors	/ * •
サーボモーター D9 を 90 度	
CE-ター M1 の速さを 10 もし く 光センサー A6 の値 < 50 なら	
CE-ター M1 を 正転 EE A4 を 点灯	
CRUTIUS DCモーター M1-を停止-	
● ブザー ■ から 60▼ を出力す	
◆ ブザー ● の出力を停止する	

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
 ◎



⑥ ⑤で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。



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、 ぺをクリックしてください。

Based on Scratch from the MIT Media Lab ⊕ 日 ファイル 編集 実行 ヘルヴ
動き 制御 調べる 演算 変数 アッと
「 周線 「 周線 「 関数 「 図 目録の返す 「 ① 回線の返す 」

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、照射した赤外線の反射を受光することで、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/ブザー用コネクターの A4 に LED、A7 に赤外線フォトリフ レクタを接続します。



赤外線フォトリフレクタ

4.3.2. 入出カポート情報の設定

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。

上側に灰色のコードがくるように。



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。



確認ができたら、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

(3)

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 す。本プログラムで プロックを使ったプログラムを学習します。
- 「動き」パレットから
 「動き」パレットから
 「動き」パレットから



「制御」パレットから¹⁹⁶⁰ブロックを下にくっつけます。



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。



 複製した 	^{▶ ▲▲▲を「「」} ブロックの「点灯」を「消灯」に変更します。	
Sensors- SPE BLOCK		- • •
Based on Scratch from the MIT Media Lab	▋ ファイル 編集 実行 ヘルプ	

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。

🧧 sensors- SPE BLOCK		- • •
Based on Scratch from the MIT Media Lab	ファイル 編集 実行 ヘルプ]
	sensor:	5 🏓
 関数をコールする 1 秒待つ 	 □ LED A4 を点灯 □ 抄待つ □ LED A4 を消灯 	
すっと 	1 秒待 <u>)</u>	

⑥ 「制御」パレットから
 プロックをドラッグします。

😇 sensors- SPE BLOCK	- • •
Based on Scratch 🕀 🔚 ファイル 編集 実行 ヘルプ	
	sensors 🔎 🦱
「「関数をコールする」 すっと、もし こうなら	
60 0 k6	

⑦ 「演算」パレットから())ブロックをくっつけます。



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。

Sensors- SPE BLOCK	
Based on Scratch from the MIT Media Lab 🕀 🔚 ファイル 編集 実行 ヘルプ	

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



④ 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



⑩ ⑧で作成したブロックを 40 MAT を 満て ブロックの下にくっつけます。

sensors- SPE BLOCK	
Based on Scratch from the MIT Media Lab 🕀 🔚 ファイル 編集 実行 ヘルプ	
	sensors 🎮 🍋

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。

/ 制御スタート	IFDを消灯する	
🧕 LED A4 🛛 を 消灯 🖌		ずっと繰り返す
ずっと、もし (赤外線フォトリフレクタ A7 (の値) > 15) なら LED A4 を 点灯 1 秒待つ LED A4 を 消灯 1 秒待つ 1 秒待つ	物体が近くにある場合 LED を 1 回点滅する	

物体を近づけて LED が点滅することを確かめます。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。

Studuino BLOCK Programming Environment			
Based on Scratch from the MIT Media Lab	▋ ファイル 編集 実行 ヘルプ		
	▲ 期御スタート ● LED A4 を 消灯▼ すった、もし、赤外線フォトリフレクタ A7 の値 > 13 なら ● LED A4 を 点灯▼ 1 秒待つ ■ LED A4 を 点灯▼ 1 秒待つ ■ 10行つ	た/サー・ボード [A0] ボタン [A1] ボタン [A2] ボタン [A3] ボタン [A3] ボタン [A4] LED [A5] 未接続 [A6] 未接続 [A7] 赤外線フォトリフレクタ 3]	
10回線2023 すっと、もし なら もし なら		プログラムがテストモ ードで実行されます	

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リ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。



タッチセンサー A2

上側に灰色のコードがくるように。

4.4.2. 入出カポート情報の設定

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。本プログラムでをです。ブロックを使ったプログラムを学習します。

「動き」パレットから
 「動き」パレットから
 「ロックを
 「動き」パレットから
 「「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」」
 「」
 「」」
 「」」
 」」</

sensors- SPE BLOCK	
Based on Scratch 🕀 🔚 ファイル 編集 実行 ヘルプ	
Image: An international conditional conditerational conditional conditerational conditional con	sensors 🎘

② 「制御」パレットからをまで行うブロックを した A4 を 満て ブロックにくっつけます。

ansors- SPE BLOCK	
Based on Scratch 🕀 📄 ファイル 編集 実行 ヘルプ	
前き 「利御 同べる 「演算 変数 て数 したD A4~を 消灯 まで待う まで待う	sensors 🎘 ●
3 2 117 5 2 11 10 回線2) 127 6 1 0 126 3 7 2 ドラッグ& ドロップ	
60 た6 60 た6 でなければ まで待う まで続つ返す	

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(物チセンサーA2-0値)ブロックを
 ①=①ブロックの左辺にくっつけ、
 右辺に0を設定します。

sensors 🎘 🔴

⑤ 「動き」パレットから 🧧 💷 🖽 🚛 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。



A3

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。



確認ができたら、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。

4.5.4. 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
- ① 「動き」パレットから 🤷 💷 🖽 を 💯 ブロックをドラッグします。



「制御」パレットから¹⁰⁰⁰ブロックをくっつけます。







④ 複製した 41 を 点灯 ブロックの「点灯」を「消灯」に変更します。

🔤 sensors- SPE BLOCK		
Based on Scratch from the MIT Media Lab	■ ファイル 編集 実行 ヘルプ	
動き 制御	sensors	/ A
国数		
関数をコールする		
1 秒待つ		
đ-r		
	1 0 C C C C C C C C C C C C C C C C C C	
10 回繰り返す		

- • ×

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。

Sensors- SPE BLOCK	
Based on Scratch from the MIT Media Lab 🕀 🔚 ファイル 編集 実行 ヘルプ	
動き 初御 調べる 注算 変数 「開数をつルする ひ待つ ひ待つ む () 回編り返す	
 ⑥ 「制御」パレットから ご ブロックをドラッグします。 ▶ SERSORS- SPE BLOCK 	
Based on Scratch 田 同 ファイル 編集 実行 ヘルプ	
新き 新設 新設 第目の Sensors 第日の 第2 第目の 第目の 第目の 第目の 第日の 第日の 第日の 第日の 第日の 第日の 第日の 第日の 第日の 第日の 第日の 第日の 第日の 第日の 第日の <td< td=""><td></td></td<>	

⑦ 「演算」パレットから ●●● ブロックをくっつけます。

🧧 sensors- SPE BLOCK		- • ×
Based on Scratch from the MIT Media Lab 🕀 🖥 ファイ。	ル 編集 実行 ヘルプ	
動き 制御 開べる 演算 支数 ・・・ ・・ ・・ ・・ ・・	Sensors 第11前229-ト まで繰り返す 。	
 から 10 までの乱敗 > かつ または 	 □ LED A4 を 点灯 □ 投稿つ □ LED A4 を 消灯 □ LED A4 を 消灯 □ 投稿つ 	

⑧ ● ブロックの左辺に、「調べる」パレットから(音センサーA3 10億) ブロックをくっつけ、

右辺に 30 を設定します。

🤤 sensors- SPE BLOCK	- • •
Based on Scratch from the MIT Media Lab 🕀 启 ファイル 編集 実行 ヘルプ	
< 音センサー A3 の値 > 30 まで繰り返す	
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。

🧧 sensors- SPE BLOCK		- • ×
Based on Scratch from the MIT Media Lab	- ファイル 編集 実行 ヘルプ	
動き 制御 調べる 演算 変数 (注算) 光センサー の価 カッチセンサー の価 音センサー A3 加速度センサー の価 ボタン A0 の値	▲ 制御スタート ● 新御スタート ● 世ンナー A3 の値 > 30 まで繰り返す ● LED A4 を 道灯 ● LED A4 を 道灯 ● とD A4 を 道灯 ● ようつ	

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。

	音センサー A3 の値 > 30 まで繰り返す LED A4 を 点灯 ~ 1 秒待つ 1 秒待つ 1 秒待つ	
 ③で作成した作成した ⑤で作成した作成した Based on Scratch ⑤ 「「「「「」」」 ⑦ 「」」 ⑦ 「」 ① 「」	Eブロックを ブロックにくっつけま (ル 編集 実行 ヘルプ	す . ा ∎ ≍
首ピンサー A3 00億 (赤外線反射センサー ■の値) (加速度センサー ■の値) (ポタン A0 ■ の値)	1 秒待つ 2 LED A4 to 满灯 T 1 秒待つ	

① 「動き」パレットから 41 を 点灯 ブロックをくっつけ、消灯に設定します。

🤤 sensors- SPE BLOCK	
Based on Scratch 🕀 📄 ファイル 編集 実行 ヘルプ]
動き 制御 正式 演算 支払 「「「」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」」	sensors 🎘 🌢

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。



- ② 息を吹きかけることで LED が消灯することを確かめます。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。
- ③ 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。