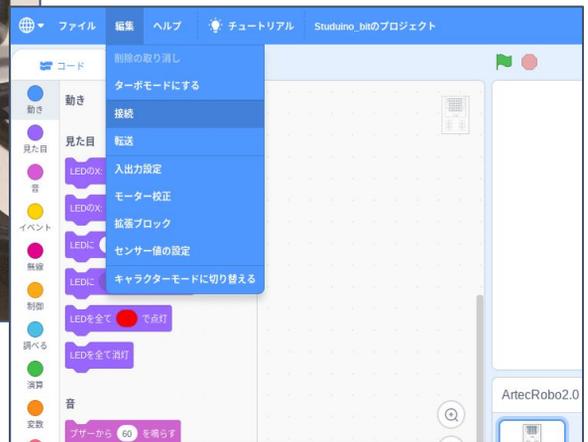


Studuino:bit「サンプル プログラミング集」 (第1.0版)



大分県立芸術文化短期大学
情報コミュニケーション学科 非常勤講師
望月陽一郎
mochizuki@oita-pic.ac.jp

作品のクレジットを表示すること・営利目的での利用をしないこと・元の作品を改変しないこと
(作者の許可を得ている場合は除きます)

目次

- 001 センサーの値を見る
- 002 センサーの値を表示する
- 003 ボタンでセンサーの値を表示する
- 004 プログラムを本体のみで実行する
- 005 明るさをグラフにする①/2
- 006 明るさをグラフにする②/2
- 007 リアルタイムグラフ
- 008 暗くなるとLED が点く
- 009 傾けると点が移動する
- 010 端まで来るとブザーが鳴る
- 011 動かすとブザーが鳴る
- 012 磁石をさがす

Studuino:bit (USB 接続の場合)

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

オンライン版 (WEBアプリ)

Windows / Mac / ChromeOS

Bluetooth接続版

USB接続版

※インターネット接続環境およびChromeブラウザ (Ver89以降) が必要です
※全ての端末で動作保証しているわけではありません。

どちらのモードで作成するか選択してください

作成したプログラムをPythonコードに変換したり、Studuino:bitに転送させたりすることができます。

スプライトを追加し、画面内でキャラクターを動かすプログラムができます。



Studuino:bit を
つなぐ

ファイル 編集 ヘルプ チュートリアル Studuino_bitのプロジェクト

コード

- 動き
- 見た目
- 音
- イベント
- 無線
- 制御
- 調べる
- 演算
- 変数

接続

接続すると出てくる
センサーの値

Studuino:bit	
ボタンA	1
ボタンB	1
光センサー	95
温度センサー	41.68
モーションセンサー	▲

ArtecRobo2.0

分類されたブロック

プログラムをつくる



【説明】

Studuino:bit プログラミングサイト

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

- ・インストール用のアプリダウンロード
- ・Web でプログラミング
ができます。

USB 接続→ロボットモード の場合、

- ①ブロックを組み合わせてプログラミング。
- ②USB 接続しているStuduino:bit で、
 - ・接続したままプログラムを実行。
 - ・転送してUSB 接続を解除し単独で実行。
できます。

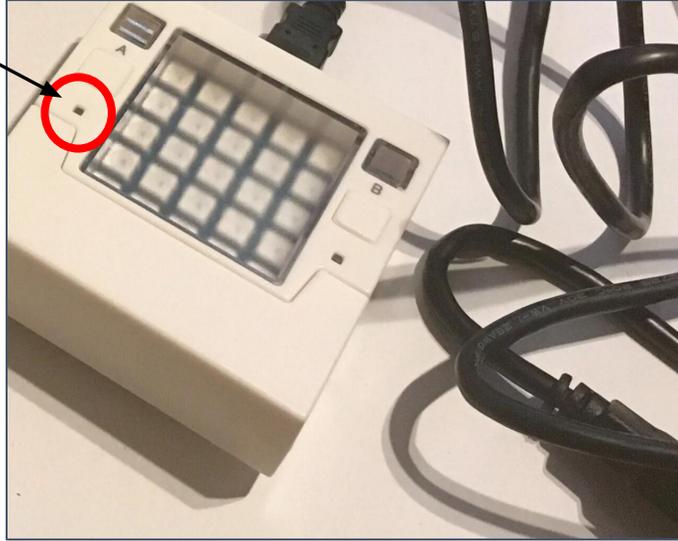
(プログラムをPC に保存することもできます。)

※ダウンロード保存しておいたプログラムを読み込んで修正することもできます。

プログラムを「転送」し、
電池ボックスを取り付けると、
単体でプログラムが動きます。

001 センサーの値を見る

電源ランプ
(USB 接続すると
緑色)

A screenshot of the Studuino IDE interface. The top menu bar includes 'ファイル', '編集', 'ヘルプ', 'チュートリアル', and 'Studuino_bitのプロジェクト'. The left sidebar shows various tool categories like '動き', '見た目', '音', etc. The main workspace displays a 'センサーボード' (Sensor Board) window for 'Studuino:bit'. This window lists various sensors and their current values. A green callout box with an arrow points to the 'モーションセンサー' (Motion Sensor) value, which is 0.05. The text in the callout box says 'センサーボードの値はリアルタイムで変わる' (The values on the sensor board change in real-time).

センサーボード	
Studuino:bit	
ボタンA	1
ボタンB	1
光センサー	95
温度センサー	33.09
モーションセンサー	0.05
加速度センサー X	-0.09
加速度センサー Y	0.98
加速度センサー Z	0
ジャイロセンサー X	0
ジャイロセンサー Y	2
ジャイロセンサー Z	-35
磁気センサー X	-36
磁気センサー Y	24
磁気センサー Z	

センサーボードの値は
リアルタイムで変わる

001 センサーの値を見る

【説明】

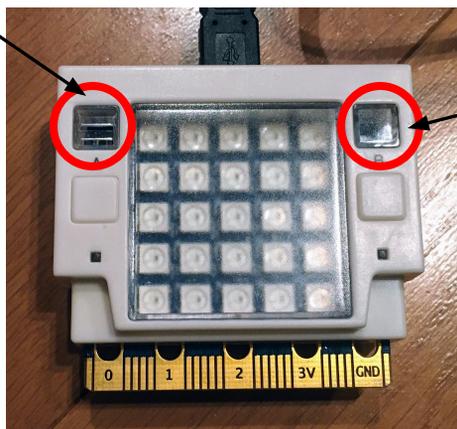
USB 接続すると、

- ・センサーボードが表示されます。
 - ・ボタン …押されている:0 押されていない:1
 - ・光センサー …暗い0~100明るい
(LED が明るく発光すると影響を受けることがあります。)
- ▼温度センサー …℃
(電源が入ると発熱による温度上昇で、気温を示しません。)
- ・モーションセンサー
 - ・加速度センサー …X, Y, Z 方向の動きを表示
(-2.00~2.00)
 - ・ジャイロセンサー …X, Y, Z 方向の傾きを表示
(静止状態(0,0,0)から動き出したか)
 - ・磁気センサー …X, Y, Z の向きで磁気の強さを表示
(どの方向から磁石が近づいているかなど)

温度センサー

光センサー

モーション
センサーは内蔵



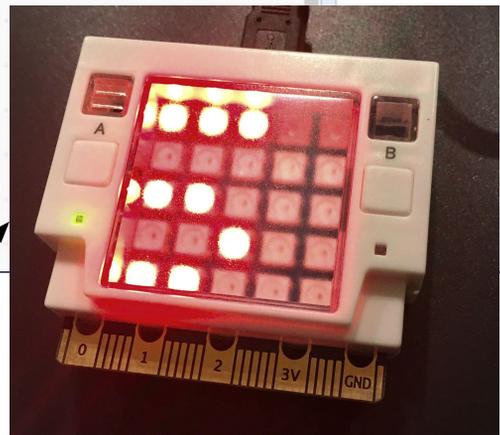
002 センサーの値を表示する

緑の旗をクリック
すると実行

The screenshot shows the Studuino IDE interface. The top menu bar includes 'ファイル', '編集', 'ヘルプ', 'チュートリアル', and 'Studuino_bitのプロジェクト'. Below the menu, there are tabs for 'コード' and 'Python'. A green flag icon is circled in red, with an arrow pointing to it from the text box above. The main workspace contains a Python script with the following code blocks:

```
が押されたとき  
LEDに 本体の光センサー を ● で表示させる
```

The left sidebar shows various categories of blocks: 動き, 見た目, 音, イベント, 無線, 制御, 調べる, 演算, and 変数. The '調べる' category is currently selected.



USB 接続した
Studuino:bit
のLED に表示される

002 センサーの値を表示する

【説明】

カテゴリ ブロック

- ・イベント → ・緑の旗が押されたとき
 - ・見た目 → ・LED にHello! を赤で表示する
 - ・調べる → ・本体の光センサー
- を組み立てます。

緑の旗が押されたとき

→ LED に「本体の光センサー(の値)」を赤で表示するというプログラムになります。

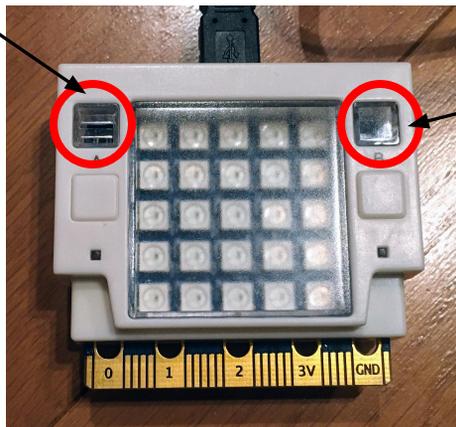
(変更できる部分)

- ・表示する色
 - ・表示するセンサーの値(光センサー・温度センサー・加速度センサー(X, Y, Z)・ジャイロセンサー(X, Y, Z)・磁気センサー(X, Y, Z))
- ※温度センサーの値は、本体の温度で、気温を示すわけではありません。

温度センサー

光センサー

モーション
センサーは内蔵



003 ボタンでセンサーの値を表示する



Studuino:bit の
ボタンを押すと、
LED に表示される

プログラムを「転送」し、
電池ボックスを取り付けると、
単体でプログラムが動きます。

003 ボタンでセンサーの値を表示する

【説明】

カテゴリ ブロック

- ・調べる → ・本体のボタンA が押されたとき
 - ・見た目 → ・LED にHello! を赤で表示する
 - ・調べる → ・本体の光センサー
- を組み立てます。

本体のボタンA が押されたとき

→ LED に「本体の光センサー(の値)」を赤で表示するというプログラムになります。

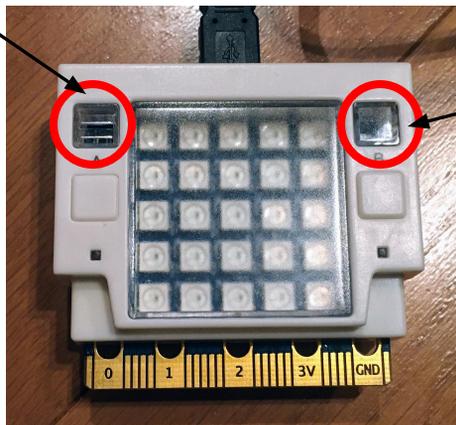
(変更できる部分)

- ・押すボタン(A, B)
 - ・表示する色
 - ・表示するセンサーの値(光センサー・温度センサー・加速度センサー(X, Y, Z)・ジャイロセンサー(X, Y, Z)・磁気センサー(X, Y, Z))
- ※温度センサーの値は、本体の温度で、気温を示すわけではありません。

温度センサー

光センサー

モーション
センサーは内蔵

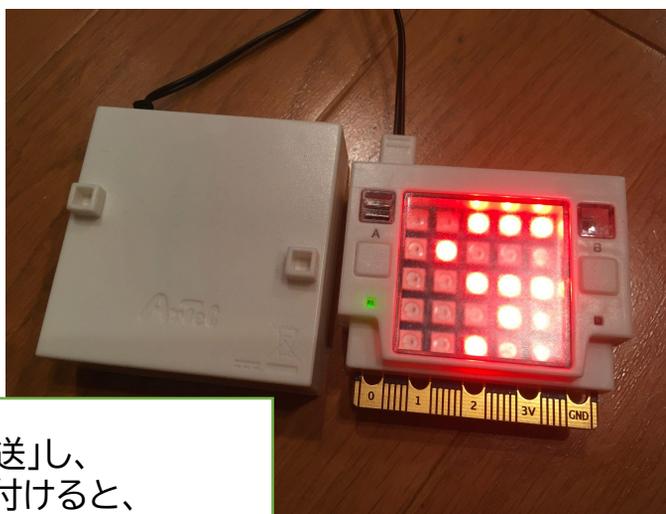


004 プログラムを本体のみで実行する

Studuino:bit に
プログラム転送

The screenshot shows the Studuino:bit software interface. A menu is open, and the '転送' (Transfer) option is circled in red. An arrow points from this menu to a dialog box titled '転送先の選択' (Transfer destination selection). In this dialog, the 'ボタンで値を表示' (Show value with button) option is circled in red, and an arrow points to it from a text box below. The dialog lists 10 sample programs (sample1 to sample10) with '転送' (Transfer) buttons next to each.

プログラムの名前を
入力して転送



プログラムを「転送」し、
電池ボックスを取り付けると、
単体でプログラムが動きます。

004 プログラムを本体のみで実行する

【説明】

※転送するほうが「実行が速くなる」プログラムがあります。

転送するプログラム保存先

・0～9 …上書きされます。

転送が成功すると「青」が点灯して、失敗すると「白」が点灯します。

※本体に保存されているプログラムから実行するものを選択

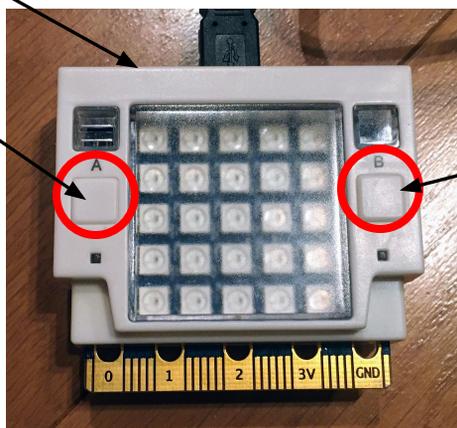
- ・ボタンA を押したまま「リセット」ボタンを押す。
- ・3秒後、電源ランプがついたらボタンA をはなす。

↓

- ・「0」が表示されます。
- …ボタンA を押すと、0→1→2…9→0と変わります。
- …ボタンB を押すとそのプログラムが選択されます。

リセットボタンは
ここ

ボタンA



ボタンB

005 明るさをグラフにする①/2



Studuino:bit の
ボタンA を押すと、
明るいときLED が点き
暗いとき点かない



プログラムを「転送」し、
電池ボックスを取り付けると、
単体でプログラムが動きます。

005 明るさをグラフにする①/2

【説明】

カテゴリ ブロック

- ・調べる → ・本体のボタンA が押されたとき
 - ・制御 → ・もし「 」なら…でなければ
 - ・演算 → ・「 」>50
 - ・調べる → ・本体の光センサー
 - ・見た目 → ・LED に「 」を表示させる
- を組み立てます。

本体のボタンA が押されたとき

- もし「本体の光センサー(の値)>95」なら
- LED に「すべて白」を表示させる
- でなければ
- LED に「すべて黒」を表示させる

というプログラムになります。

グラフ化の第①段階です。グラフを表示するブロックがないので、条件を分けながらつくっていきます。

- ・明るさセンサーは0~100の値ですが、指でセンサーを押さえてもそんなに下がらなかつたため、「95」でサンプルをつくっています。センサーボードの値を見ながら変えてみるとよいですね。



明るさセンサー

006 明るさをグラフにする②/2

The screenshot shows the Studuino:bit IDE interface. The main workspace contains a program with the following logic:

- When button A is pressed, check if the light sensor value is greater than 95. If yes, turn on 9 LEDs.
- Else, check if the light sensor value is greater than 85. If yes, turn on 8 LEDs.
- Else, check if the light sensor value is greater than 65. If yes, turn on 6 LEDs.
- Else, check if the light sensor value is greater than 25. If yes, turn on 2 LEDs.
- Otherwise, turn off all LEDs.

On the right, the 'センサーボード' (Sensor Board) window displays the following data:

センサーボード	
Studuino:bit	
ボタンA	1
ボタンB	1
光センサー	100
温度センサー	42.04
モーションセンサー	



Studuino:bit の
ボタンA を押すと、
明るさによって
LED のつく段数が
変わる

プログラムを「転送」し、
電池ボックスを取り付けると、
単体でプログラムが動きます。

006 明るさをグラフにする②/2

【説明】

カテゴリ ブロック

- ・調べる → ・本体のボタンA が押されたとき
- ・制御 → ・もし「 」なら…でなければ
- ・演算 → ・「 」 $>$ 50
- ・調べる → ・本体の光センサー
- ・見た目 → ・LED に「 」を表示させる

を組み立てます。005と使っているブロックは変わりません。

本体のボタンA が押されたとき

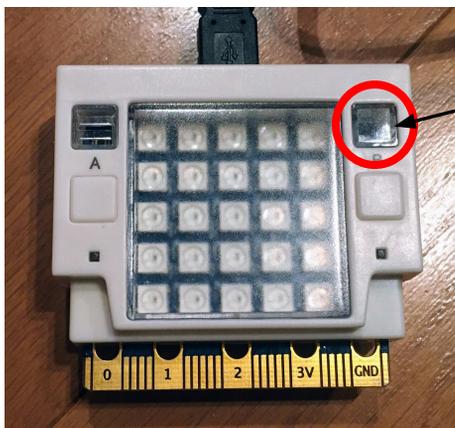
- もし「本体の光センサー(の値) $>$ 95」なら
- LED に「すべて白」を表示させる
- …ここに新しい条件を組み込んでいきます。
- でなければ
- LED に「すべて黒」を表示させる

というプログラムになります。

グラフ化の第②段階です。

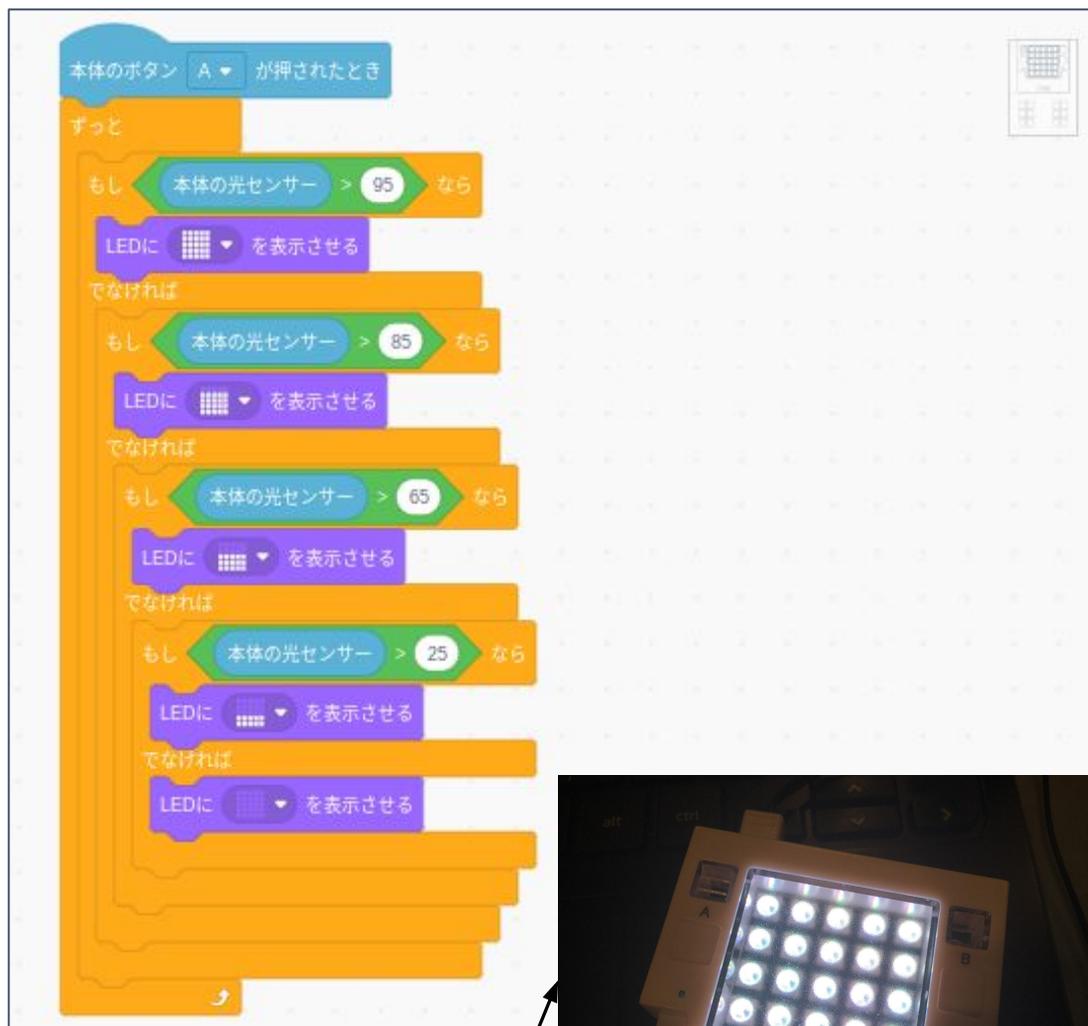
- …85より暗ければ、LED は4段まで
- …65より暗ければ、LED は3段まで点灯というように条件が絞られます。

※暗い場所に持って行ってボタンを押してみましょう。



明るさセンサー

007 リアルタイムグラフ



Studuino:bit の
ボタンA を押すと、
リアルタイムで
LED のつく段数が変わる



「接続」で実行するとき
LED の点滅に注意！

プログラムを「転送」し、
電池ボックスを取り付けると、
単体でプログラムが動きます。

007 リアルタイムグラフ

【説明】

カテゴリ ブロック

- ・調べる → ・本体のボタンA が押されたとき
- ・制御 → ・ずっと
- ・制御 → ・もし「 」なら…でなければ
- ・演算 → ・「 」>50
- ・調べる → ・本体の光センサー
- ・見た目 → ・LED に「 」を表示させる

を組み立てます。006との違いは、「ずっと」です。

本体のボタンA が押されたとき

ずっと

→ もし「本体の光センサー(の値)>95」なら

→ LED に「すべて白」を表示させる

…

→ でなければ

→ LED に「すべて黒」を表示させる

というプログラムになります。

「ずっと」にかこまれた部分が繰り返されるのでリアルタイムで変化します。

「接続」で実行するとき
LED が点滅することが
あるので注意！



明るさセンサー

プログラムを「転送」し、
電池ボックスで動かすほうが
実行速度が速く点滅が
なくなります。

008 暗くなるとLED が点く

The screenshot shows the Arduino IDE interface with a Python script. The script is as follows:

```
本体のボタン A が押されたとき  
ずっと  
  本体の光センサー > 96 まで繰り返す  
  LEDに [LED Matrix] を表示させる  
LEDを全て消灯
```

The left sidebar shows the '見た目' (Appearance) category selected, with various blocks for controlling LEDs and displaying text.

暗い場所に持っていくと
LED が点く

「接続」で実行するとき
LED の点滅に注意！

プログラムを「転送」し、
電池ボックスを取り付けると、
単体でプログラムが動きます。

008 暗くなるとLED が点く

【説明】

カテゴリ ブロック

- ・調べる → ・本体のボタンA が押されたとき
 - ・制御 → ・ずっと
 - ・制御 → ・「 」まで繰り返す
 - ・演算 → ・「 」>96
 - ・調べる → ・本体の光センサー
 - ・見た目 → ・LED に「 」を表示させる
 - ・見た目 → ・LED をすべて消灯
- を組み立てます。

本体のボタンA が押されたとき

ずっと

→ 「本体の光センサー(の値)>96」まで繰り返す。

→ LED に「すべて白」を表示させる

→ LED をすべて消灯

というプログラムになります。

これでなぜ暗くなると点灯するのかというと、センサーの値が96になるまでは「LED を点灯」が繰り返される からです。明るくなった時点で「消灯」という逆の発想ですね。

「接続」で実行するとき
LED が点滅することが
あるので注意！



明るさセンサー

プログラムを「転送」し、
電池ボックスで動かすほうが
実行速度が速く点滅が
なくなります。

009 傾けると点が移動する

センサーボード

Studuino:bit	
ボタンA	1
ボタンB	1
光センサー	100
温度センサー	37.51
モーションセンサー	▼
加速度センサー X	0.86
加速度センサー Y	-0.13
加速度センサー Z	0.56
ジャイロセンサー X	0
ジャイロセンサー Y	0
ジャイロセンサー Z	2

Scratch code blocks for the program:

- When button A is pressed: Turn on LED at X: 2, Y: 2.
- Forever loop:
 - Wait 0.5 seconds.
 - Turn off all LEDs.
 - Turn on LED at X: (Body Acceleration Sensor X) * 2 + 2, Y: 2. Round the X value to the nearest integer.

傾けると
LED が左右に移動



プログラムを「転送」し、
電池ボックスを取り付けると、
単体でプログラムが動きます。

009 傾けると点が移動する

【説明】

カテゴリ ブロック

- ・調べる → ・本体のボタンA が押されたとき
- ・見た目 → ・LED のX:「0」Y:「0」を赤で点灯
- ・制御 → ・ずっと ・制御 → ・0.5秒待つ
- ・見た目 → ・LED をすべて消灯
- ・調べる → ・本体の加速度センサー X
- ・演算 → ・「 \times 」 ・演算 → ・「 $+$ 」 ・演算 → ・「 \lceil 」を四捨五入を組み立てます。

本体のボタンA が押されたとき

→ ED のX:「2」Y:「2」を赤で点灯 …… 最初は真ん中に点灯
ずっと

→ 0.5秒待つ …… 点滅を少なくするため

→ LED をすべて消灯

→ LED のX:「※」Y:「2」を赤で点灯 …… ※の部分が変わることになります。

というプログラムになります。

※本体の加速度センサーの値を「センサーボード」で見ると、-1~1くらいの変化だったので、
(本体の加速度センサー X(の値) $\times 2 + 2$)を四捨五入して、0,1,2,3,4にあてはめました。

センサーの値	-1	-0.8	-0.6	-0.4	-0.2	0	0.2	0.4	0.6	0.8	1
$\times 2$	-2	-1.6	-1.2	-0.8	-0.4	0	0.4	0.8	1.2	1.6	2
$+ 2$	2	2	2	2	2	2	2	2	2	2	2
計算で出た値	0	0.4	1	1.2	1.6	2	2.4	2.8	3.2	3.6	4
四捨五入した値	0	0	1	1	2	2	2	3	3	4	4



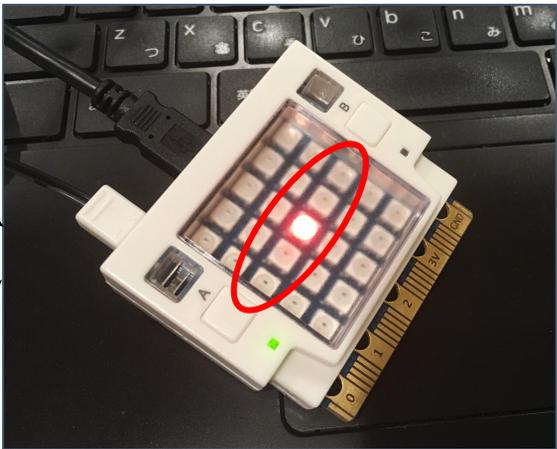
010 端まで来るとブザーが鳴る

The screenshot shows the Studuino IDE interface with a Python script. The script is as follows:

```
本体のボタン A が押されたとき
LEDのX: 2 Y: 2 を 赤 で点灯
ずっと
  0.5 秒待つ
  LEDを全て消灯
  X を 本体の加速度センサー X * 2 + 2 を四捨五入 にする
  LEDのX: X Y: 2 を 赤 で点灯
  もし X = 0 または X = 4 なら
    ブザーから 60 を鳴らす
  でなければ
    ブザーを止める
```

傾けると
LED が左右に移動

LED が端まで来ると
ブザーが鳴る



プログラムを「転送」し、
電池ボックスを取り付けると、
単体でプログラムが動きます。

010 端まで来るとブザーが鳴る

【説明】

カテゴリ ブロック

- ・調べる → ・本体のボタンA が押されたとき
- ・見た目 → ・LED のX:「0」Y:「0」を赤で点灯
- ・制御 → ・ずっと ・制御 → ・0.5秒待つ
- ・見た目 → ・LED をすべて消灯
- ・変数 → ・(変数)を「 」とする
- ・調べる → ・本体の加速度センサー X
- ・演算 → ・「 」*「 」 ・演算 → ・「 」+「 」 ・演算 → ・「 」を四捨五入
- ・制御 → ・もし「 」なら…でなければ
- ・音 → ・ブザーから「 」を鳴らす … 60はC4(ド)62はD4(レ)64はE4(ミ)など
- ・音 → ・ブザーを止める

を組み立てます。

009をつくりかえたものになります。違うのは、

- ・端(0または4)に来たとき、ブザーを鳴らす。
- ・それ以外(1, 2, 3,)のときにブザーを止める。

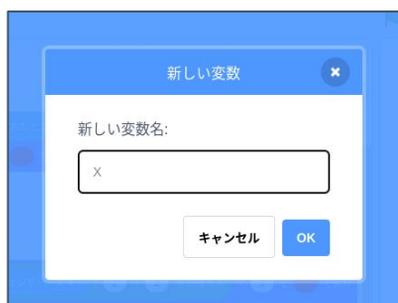
です。

条件(…または…)に、長い計算式を何度も入れたくないので、「変数」をつくっています。

一度、つくった変数「x」は、計算した答えの値である、とすると、そのあとは、

- ・x が…

と使うことができます。



011 動かすとブザーが鳴る

本体のボタン A が押されたとき

ずっと

X を 本体のジャイロセンサー X にする

もし X > 5 または X < -5 なら

ブザーから 60 を鳴らす

LEDに を表示させる

でなければ

ブザーを止める

LEDを全て消灯

センサーボード	
Studuino:bit	
ボタンA	1
ボタンB	1
光センサー	100
温度センサー	38.72
モーションセンサー	
加速度センサー X	0.08
加速度センサー Y	0.03
加速度センサー Z	1.00
ジャイロセンサー X	0
ジャイロセンサー Y	0
ジャイロセンサー Z	2
磁気センサー X	-17
磁気センサー Y	-36
磁気センサー Z	-13

止まっている状態から
少し動かすと、
ブザーが鳴りLED が点く



プログラムを「転送」し、
電池ボックスを取り付けると、
単体でプログラムが動きます。

011 動かすとブザーが鳴る

【説明】

カテゴリ ブロック

- ・調べる → ・本体のボタンA が押されたとき
- ・制御 → ・ずっと
- ・変数 → ・(変数)を「 」とする
- ・調べる → ・本体のジャイロセンサー X
- ・制御 → ・もし「 」なら・・・でなければ
- ・音 → ・ブザーから「 」を鳴らす …… 60はC4(ド)62はD4(レ)64はE4(ミ)など
- ・見た目 → ・LED に「赤」を表示させる
- ・音 → ・ブザーを止める
- ・見た目 → ・LED をすべて消灯

を組み立てます。

ジャイロセンサーX (の値)を何度も使うので、「変数」をつかっています。

ジャイロセンサーの値は一度「止まった」段階で「0」になります。

つまり一度動いて止まるとその状態がそのつど0になるのです。そこから+5 あるいは -5 以上動いたらブザーを鳴らし、LED が全点灯します。

が、止まったらそこが0になるので、ブザーは止まり、LED も消えます。



「止まっている状態」から
少し動かしてみます。



012 磁石をさがす



The image shows a Scratch script and a sensor board. The script starts with a 'when button A is pressed' event, followed by a 'loop forever' block. Inside the loop, the Z-axis value of the magnetic sensor is divided by 1000 and its absolute value is taken. Two conditional checks follow: if Z > 2, the red LED is lit; if Z > 1, the red LED is lit and all other LEDs are turned off. The sensor board data on the right shows the Z-axis magnetic sensor value as 2891, which is circled in red.

センサーボード	
Studuino:bit	
ボタンA	1
ボタンB	1
光センサー	100
温度センサー	35.44
モーションセンサー	
加速度センサー X	-0.05
加速度センサー Y	-0.09
加速度センサー Z	1.00
ジャイロセンサー X	0
ジャイロセンサー Y	0
ジャイロセンサー Z	2
磁気センサー X	-1025
磁気センサー Y	26
磁気センサー Z	2891

磁気センサー Z の値で
調べる

磁石に上から近づけると
LED が点く



プログラムを「転送」し、
電池ボックスを取り付けると、
単体でプログラムが動きます。

012 磁石をさがす

【説明】

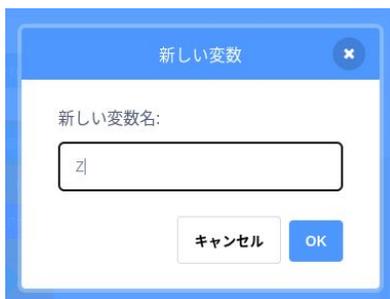
カテゴリ ブロック

- ・調べる → ・本体のボタンA が押されたとき
- ・制御 → ・ずっと
- ・変数 → ・(変数)を「 」とする
- ・調べる → ・本体の磁気センサー Z …磁石の向きにより、+ - がある
- ・演算 → ・「 」>50
- ・演算 → ・「 」/「 」
- ・演算 → ・「 」の切り下げ→絶対値
- ・制御 → ・もし「 」なら…でなければ
- ・見た目 → ・LED に「赤」を表示させる
- ・見た目 → ・LED をすべて消灯

を組み立てます。

磁気センサーZ (の値)を何度も使うので、「変数」をつけています。

磁気センサーの値は磁石の向きにより、-**** ~ +**** と変化する(センサーボードで見る)ので、1000で割ったあと絶対値(正の数にする)にすることで一桁の数にしています。



磁石に上から近づけると
LED が点く





Studuino:bit 「サンプル プログラミング集」 (第1.0版)

作品のクレジットを表示す
ること・営利目的での利用
をしないこと・元の作品を
改変しないこと
(作者の許可を得ている場
合は除きます)

2022/09/01 第1.0版

