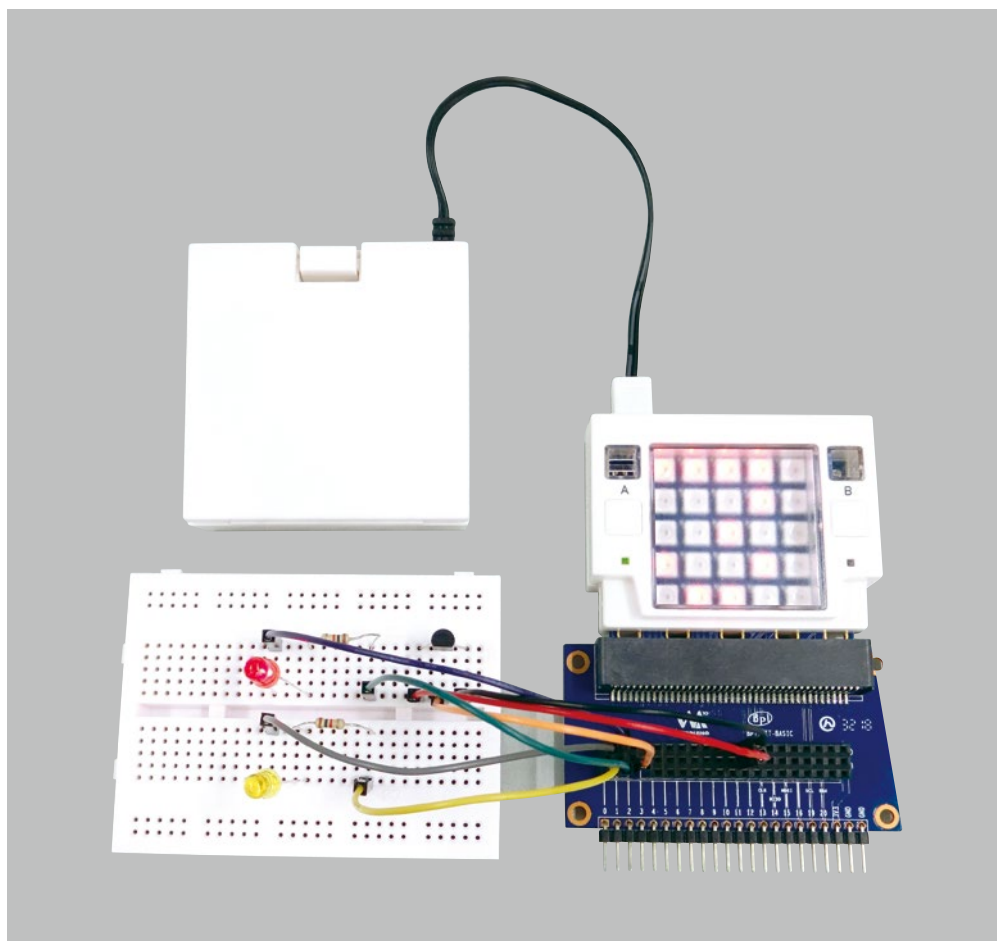


高校情報 I 計測・制御とプログラミング

教材監修：元 東京学芸大学 教育学部特任教授 天良和男先生

JavaScript 版

教員用



目次

準備 Windows をご使用の場合

1. ソフトウェアのダウンロード	3
2. ファームウェアの書き込み	3
3. Studuino:bit ライブラリ	5
4. プログラムを転送して実行する	5

準備 Mac をご使用の場合

1. ソフトウェアのダウンロード	7
2. ファームウェアの書き込み	7
3. Studuino:bit ライブラリ	9
4. プログラムを転送して実行する	9

演習

演習① 温度センサからのアナログ入力プログラム	11
演習② 温度センサからのアナログ入力と LED へのデジタル出力プログラム	14
演習③ 温度センサからのアナログ入力と 2 つの LED へのデジタル出力プログラム	17
参考:温度とアナログ入力から取得したデジタル値の関係	20

付録

付録 A. Studuino:bit ソフトウェアでのファームウェア更新手順	21
付録 B. ブレッドボードの使い方	22
付録 C. 変換エッジコネクタの使い方	22
付録 D. デジタル入出力アナログ入力オブジェクトの機能一覧	23
付録 E. basic オブジェクトの機能一覧	23
付録 F. Studuino:bit ソフトウェアを用いたプログラムの作成	24
参考:スプライトのy座標とアナログ入力から取得したデジタル値の関係	30

本テキストに掲載しているプログラムは下記ページからダウンロードできます。

<https://www.artec-kk.co.jp/dl/98072/>

準備 Windows をご使用の場合

1. ソフトウェアのダウンロード

JavaScriptでStuduino:bitのプログラムを作成するために必要なプログラム開発環境をダウンロードします。Studuino:bitでは、組み込みJavaScript実行環境のQuickJSを使用します。QuickJSは、JavaScriptの標準規格ES2020で定義された機能をサポートしています。

①下記ArtecRobo2.0のJavaScriptページからWindowsに対応したソフトウェアをダウンロードしてください。

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

②ダウンロードした「quickjssb_win.zip」を展開してください。

※PCの設定によっては、拡張子(.zip)が表示されない場合があります。

Zip ファイルのアイコンを右クリックし、表示されるメニューから「すべて展開」を選択します。

「圧縮(ZIP 形式)フォルダーの展開」ダイアログが開きますので、ダイアログの指示に従って展開してください。すでに解凍ソフトウェアが別途インストールされている場合は、「解凍」を選択してください。

2. ファームウェアの書き込み

Studuino:bitをJavaScriptで制御するためにStuduino:bitのファームウェアを書き換えます。

本項目を実施するとStuduino:bitにJavaScript専用ファームウェアが書き込まれた状態になりますので、JavaScript以外(ブロックプログラミング、Python)で利用される場合は「Studuino:bitソフトウェア」から「ファームウェア更新」を行ってください。Studuino:bitソフトウェアでのファームウェア更新の手順は付録Aを参照してください。

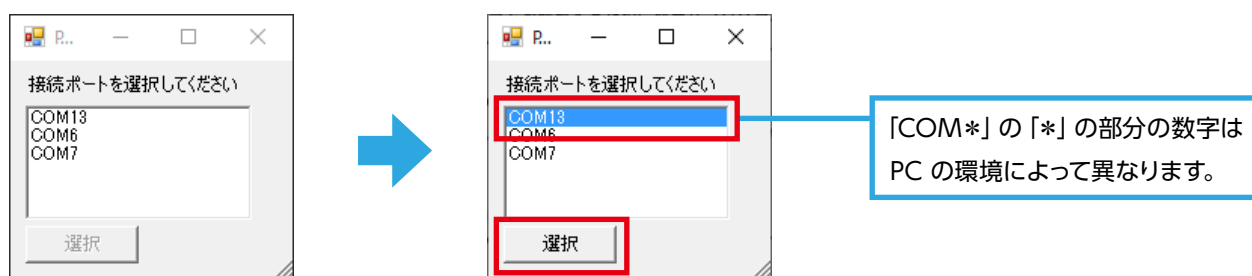
①Studuino:bitとPCを接続します。

②「ファームウェアの書き込み.bat」を実行します。

下記のようなウィンドウが立ち上がった場合は、Studuino:bitを接続しているポートをクリックして、「選択」をクリックしてください。

※PCの設定によっては、拡張子(.bat)が表示されない場合があります。

※下記のウィンドウは初回または接続ポートを変えたときのみ表示されます。



③下記のようなウィンドウが立ち上がりファームウェアの書き込みが開始されます。

```
C:\WINDOWS\system32\cmd.exe
Writing firmware.bin ...
esptool.py v3.1
Serial port COM3
Connecting.....
Chip is ESP32-D0WD (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: c8:2b:96:9b:28:70
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 1500000
Changed.
Configuring flash size...
Flash will be erased from 0x00000000 to 0x0020ffff...
Compressed 2162688 bytes to 434147...
Writing at 0x00035984... (25 %)
```

なお、batファイルをダブルクリックしたときにMicrosoft Defender SmartScreenが開き、「WindowsによってPCが保護されました」と表示される場合は「詳細情報」→「実行」の順にクリックすると、上記ウィンドウが立ち上がります。

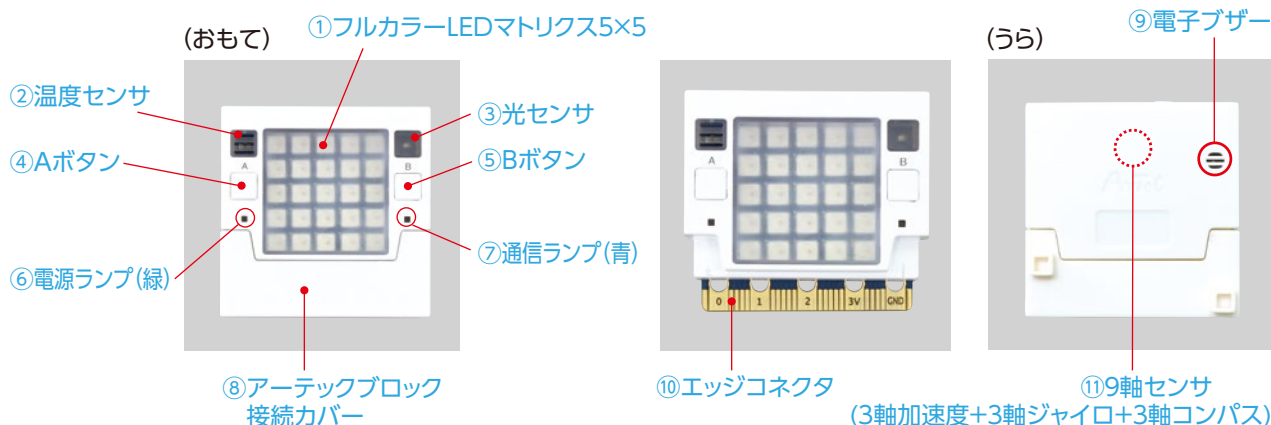


④下記のメッセージが表示されたらファームウェアの書き込み完了です。
ウィンドウを閉じてください。

```
Leaving...
Hard resetting via RTS pin...
ファームウェアの書き込みが完了しました。
Stduino:bitのリセットボタンを押してください。
何かキーを押すとウィンドウが閉じます。
```

3.Studuino:bit ライブラリ

Studuino:bitを制御する、オブジェクトや関数をまとめたモジュールとしてStuduino:bitライブラリを提供しています。Studuino:bitは下図のハードウェアを持っています。



2022年5月現在、JavaScriptでは、フルカラーLEDマトリクス5x5とエッジコネクタP0～P2が使用できます。

4. プログラムを転送して実行する

テキストエディタで編集したJavaScriptスクリプトをStuduino:bitで実行する手順を下記に記します。

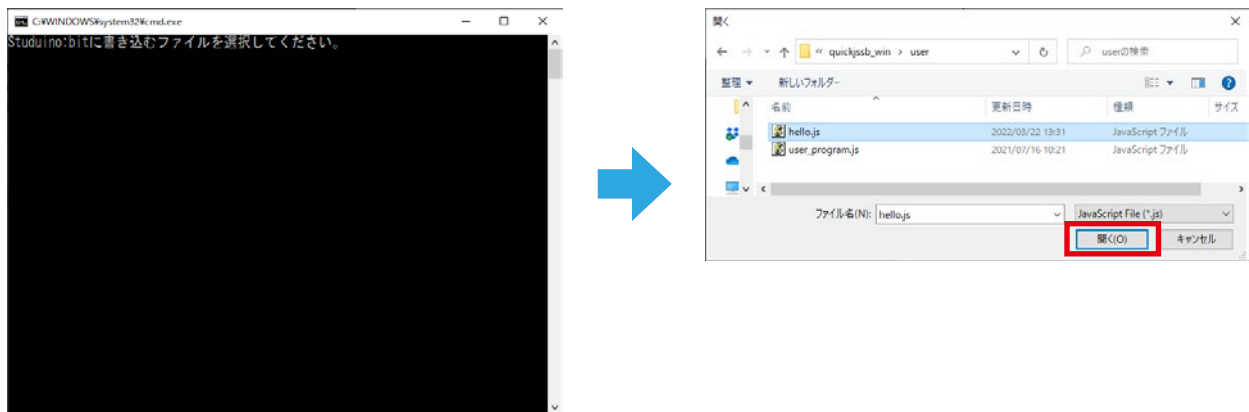
- ①Studuino:bitとPCを接続します。
- ②下記をメモ帳などで作成して「hello.js」というファイル名で文字コードUTF-8形式で保存してください。
「1.ソフトウェアのダウンロード」で展開したquickjssb_winフォルダ下のuserフォルダ下に保存すると次の③でファイル選択時にアクセスが簡単です。

```
basic.showString('Hello')
```



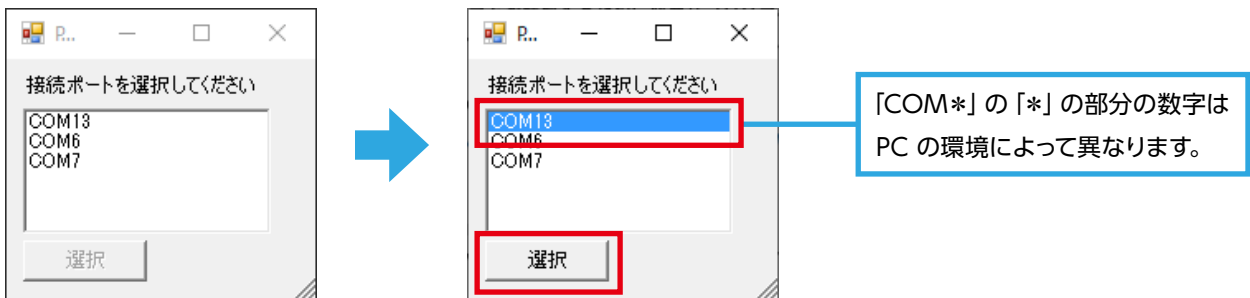
③ [作ったプログラムの書き込み.bat] を実行して、②で保存したファイルを選択してください。

※PCの設定によっては、拡張子 (.bat) が表示されない場合があります。



④下記のようなウィンドウが立ち上がった場合は、Studuino:bitを接続しているポートをクリックして、「選択」をクリックしてください。

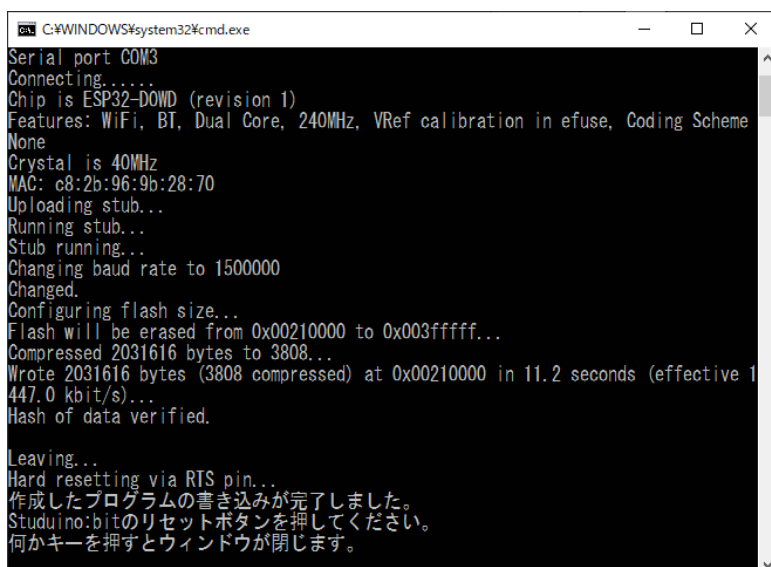
※下記のウィンドウは初回または接続ポートを変えたときのみ表示されます。



⑤下記のメッセージが表示されたら、プログラムの書き込み完了です。

Studuino:bitメインユニットの画面に文字列”Hello”がスクロール表示されることを確認してください。表示終了後は、Studuino:bitメインユニットのリセットボタンを押すことで再度プログラムが実行されます。

プログラムは転送されているため、USBケーブルを取り外し、電池ボックスを接続してスタンドアロンで動作を確認できます。



1. ソフトウェアのダウンロード

JavaScriptでStduino:bitのプログラムを作成するために必要なプログラム開発環境をダウンロードします。Stduino:bitでは、組み込みJavaScript実行環境のQuickJSを使用します。QuickJSは、JavaScriptの標準規格ES2020で定義された機能をサポートしています。

①下記ArtecRobo2.0のJavaScriptページからMacに対応したソフトウェアをダウンロードしてください。

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

②ダウンロードした「quickjsb_mac.zip」をダブルクリックして解凍します。

※PCの設定によっては、拡張子 (.zip) が表示されない場合があります。

2. ファームウェアの書き込み

Stduino:bitをJavaScriptで制御するためにStduino:bitのファームウェアを書き換えます。

本項目を実施するとStduino:bitにJavaScript専用ファームウェアが書き込まれた状態になりますので、JavaScript以外（ブロックプログラミング、Python）で利用される場合は「Stduino:bitソフトウェア」から「ファームウェア更新」を行ってください。Stduino:bitソフトウェアでのファームウェア更新の手順は付録Aを参照してください。

① Stduino:bitとPCを接続します。

②「ファームウェアの書き込み.command」を実行します。

初回のみ、ファイル上で右クリック(「controlキー」+クリック)→「開く」で実行してください。この際、管理者アカウントの入力が必要になります。2回目以降はダブルクリックで実行されます。

下記のようなウィンドウが表示される場合は「開く」をクリックしてください。

※PCの設定によっては、拡張子 (.command) が表示されない場合があります。



③下記のようなウィンドウが立ち上がりファームウェアの書き込みが開始されます。

```
kaihatsu — ファームウェアの書き込み.command — esptool + ファームウェアの書き込...
Last login: Tue Mar 22 09:10:02 on console
kaihatsu-no-Air:~ kaihatsu$ /Users/kaihatsu/Downloads/quickjssb_mac/ファームウェアの書き込み.command ; exit;

--- Studuino:bitを初期化します ---
esptool.py v3.0-dev
Serial port /dev/tty.usbmodem1421
Connecting.....
Chip is ESP32-D0WD (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: fc:f5:c4:6e:78:2c
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 10.5s
Hard resetting via RTS pin...

--- Studuino:bitにファームウェアを書き込みます ---
esptool.py v3.0-dev
Serial port /dev/tty.usbmodem1421
Connecting...█
```

PCにStuduino:bitが複数接続されている場合、接続ポートの一覧が表示されますので、数字を入力してポートを選択してください。

```
kaihatsu — ファームウェアの書き込み.command — ファームウェアの書き込み.comma...
Last login: Tue Mar 22 09:28:06 on ttys000
/Users/kaihatsu/Downloads/quickjssb_mac/ファームウェアの書き込み.command ; exit;
kaihatsu-no-Air:~ kaihatsu$ /Users/kaihatsu/Downloads/quickjssb_mac/ファームウェアの書き込み.command ; exit;

複数の接続が見つかりました
1. /dev/tty.usbmodem1411
2. /dev/tty.usbmodem1421
数字を入力してください
█
```

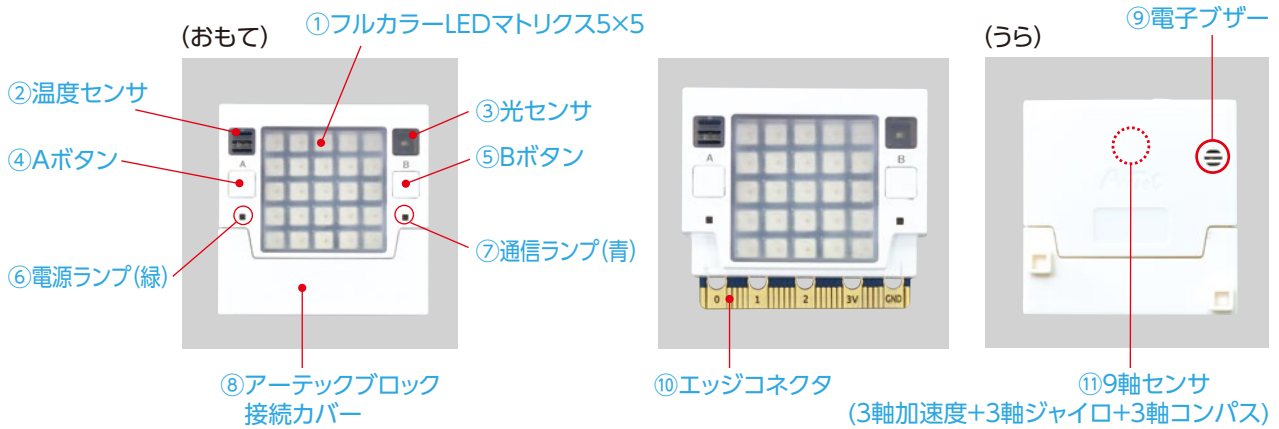
④下記のメッセージが表示されたらファームウェアの書き込み完了です。

ウィンドウを閉じてください。

```
ファームウェアの書き込みが完了しました。
Studuino:bitのリセットボタンを押してください。
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
Deleting expired sessions...none found.
```

3.Studuino:bit ライブラリ

Studuino:bitを制御する、オブジェクトや関数をまとめたモジュールとしてStuduino:bitライブラリを提供しています。Studuino:bitは下図のハードウェアを持っています。



2022年3月現在、JavaScriptでは、フルカラーLEDマトリクス5x5とエッジコネクタP0～P2が使用できます。

4. プログラムを転送して実行する

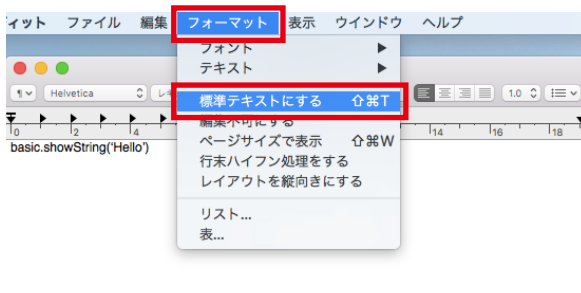
テキストエディットで編集したJavaScriptスクリプトをStuduino:bitで実行する手順を下記に記します。

①Studuino:bitとPCを接続します。

②下記をテキストエディットなどで作成します。

「フォーマット」→「標準テキストにする」を選択後、「hello.js」というファイル名で文字コードUTF-8形式で保存してください。

```
basic.showString("Hello")
```



③ [作ったプログラムの書き込み.command] を実行します。

初回のみ、ファイル上で右クリック(「controlキー」+クリック)→「開く」で実行してください。この際、管理者アカウントの入力が必要になります。2回目以降はダブルクリックで実行されます。

下記のようなウィンドウが表示される場合は「開く」をクリックしてください。

※PCの設定によっては、拡張子 (.command) が表示されない場合があります。



PCにStuduino:bitが複数接続されている場合、接続ポートの一覧が表示されますので、数字を入力してポートを選択してください。



④②で保存したファイルパスを入力します。ファイルをドラッグ&ドロップすることでパスが入力されます。



⑤ 下記のメッセージが表示されたら、プログラムの書き込み完了です。

Studuino:bitメインユニットの画面に文字列”Hello”がスクロール表示されることを確認してください。表示終了後は、Studuino:bitメインユニットのリセットボタンを押すことで再度プログラムが実行されます。

プログラムは転送されているため、USBケーブルを取り外し、電池ボックスを接続してスタンドアロンで動作を確認できます。

```
作成したプログラムの書き込みが完了しました。
Studuino:bitのリセットボタンを押してください。
logout
Saving session...
...copying shared history...
...saving history...truncating history files...
...completed.
```

[プロセスが完了しました]

演習① 温度センサからのアナログ入力プログラム

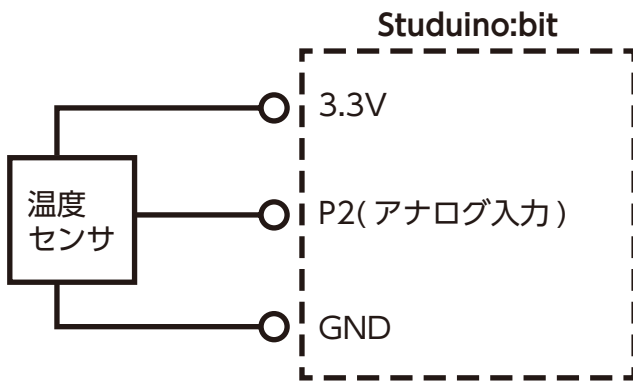
温度センサで周囲の温度を計測し、Studuino:bitに温度を表示するプログラムを作成します。

Studuino:bit本体内部には温度センサが内蔵されていますが、ここでは外部に接続した温度センサを使って計測する方法について説明します。

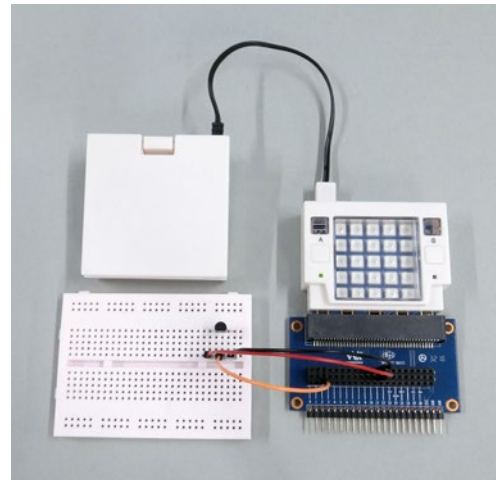
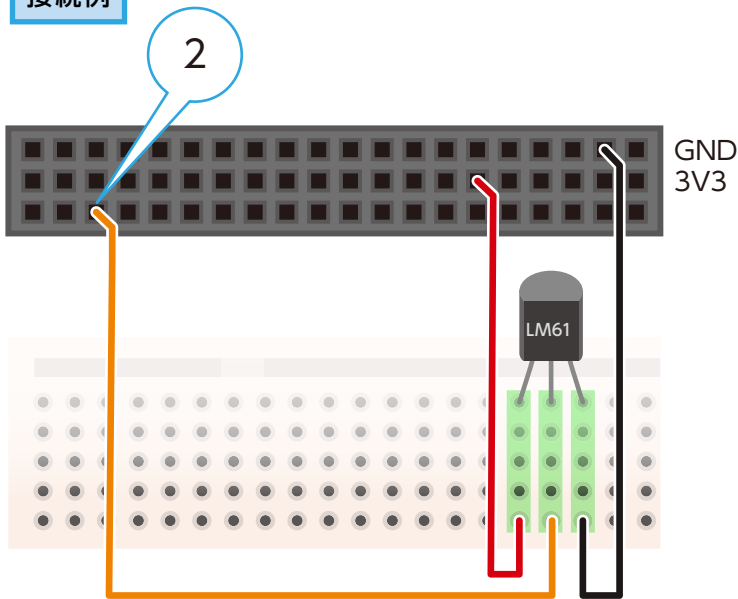
①温度センサとStuduino:bitを接続します。

ブレッドボードの使い方詳細は付録Bを参照してください。

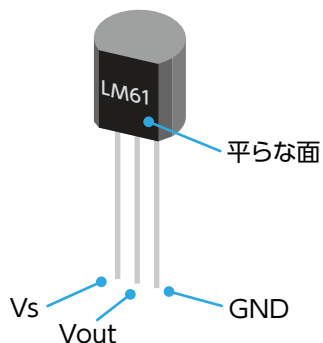
変換エッジコネクタの使い方詳細は付録Cを参照してください。



接続例



温度センサ



温度を計測するセンサです。

センサによって形状や出力形式は異なります。

今回使用する温度センサはLM61で、センサの出力はアナログ電圧です。

測定可能温度	- 30 ~ +100°C (LM61CIZ) - 25 ~ +85°C (LM61BIZ)
電源電圧	2.7 ~ 10V
出力電圧	10mV/°C
温度 0°Cの時の出力電圧	600mV

②Studuino:bitとPCを接続します。

③温度センサのアナログ入力から取得したデジタル値を、Studuino:bitのLEDディスプレイに表示するプログラムを記します。

プログラムを転送して実行し、手やヘアードライヤーなどを使って温度センサを温めるとLEDディスプレイに表示される値が変化することを確認してください。

値が変化しない場合はブレッドボードの接続が間違っていないか確認してください。

デジタル入出力アナログ入力オブジェクトの機能の詳細は付録Dを参照してください。

basicオブジェクトの機能詳細は付録Eを参照してください。

```
let val = 0;
basic.forever(function(){
  val = pins.analogReadPin(AnalogPin.P2);
  basic.showNumber(val);
})
```

ポートP2の値を読み取り、0 (0V) から1023 (3.3V) までの間の整数を返す。

LEDディスプレイに表示

④アナログ入力から取得したデジタル値を温度に変換し、Studuino:bitのLEDディスプレイに表示するプログラムを記します。

```
let val = 0;
basic.forever(function(){
  val = Math.floor(330 * pins.analogReadPin(AnalogPin.P2) / 1023 - 60);
  basic.showNumber(val);
  basic.showString('C');
})
```

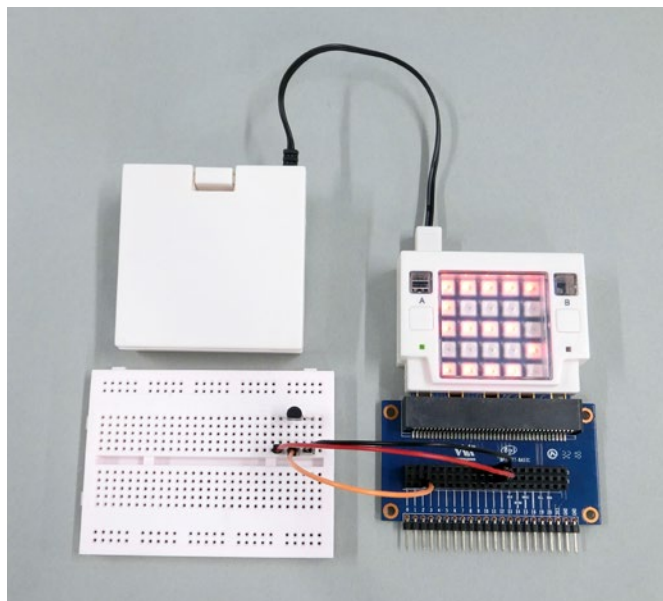
enshu1.js

センサ値を温度に変換 (P.20の式③参照)

QuickJSは、JavaScriptの標準規格ES2020で定義された構文をサポートしているため、変数の宣言 [let] を [var] に、無限ループ [basic.forever(function(){})] を [while(true){}] に、 [}] を [}] に変更しても同様の動作を確認できます。

```
var val = 0;
while(true){
  val = Math.floor(330 * pins.analogReadPin(AnalogPin.P2) / 1023 - 60);
  basic.showNumber(val);
  basic.showString('C');
}
```

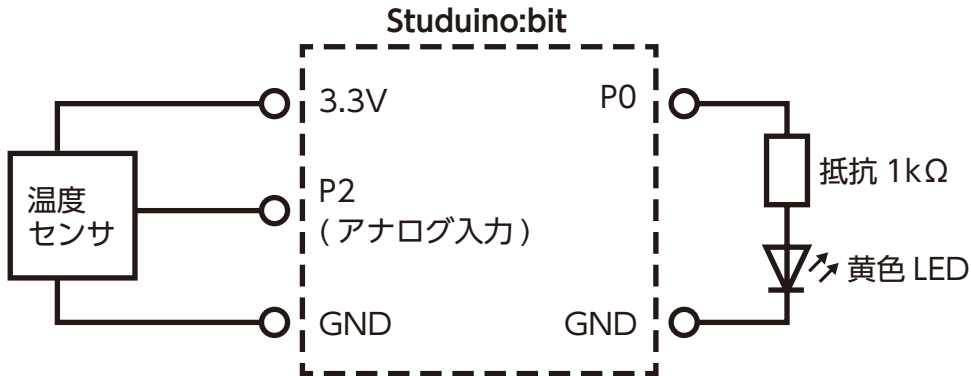
⑤プログラムを転送して実行し、Stduino:bitのLEDディスプレイに温度(例:25C)がスクロール表示されることを確認してください。



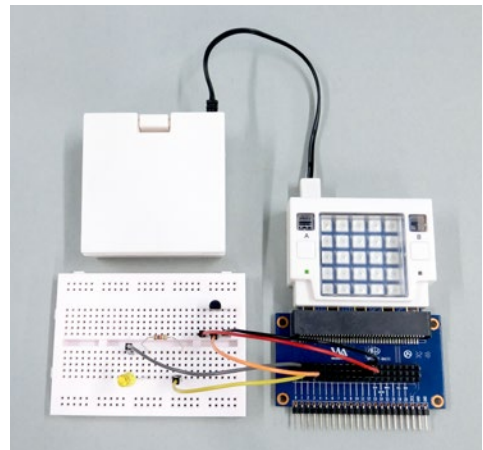
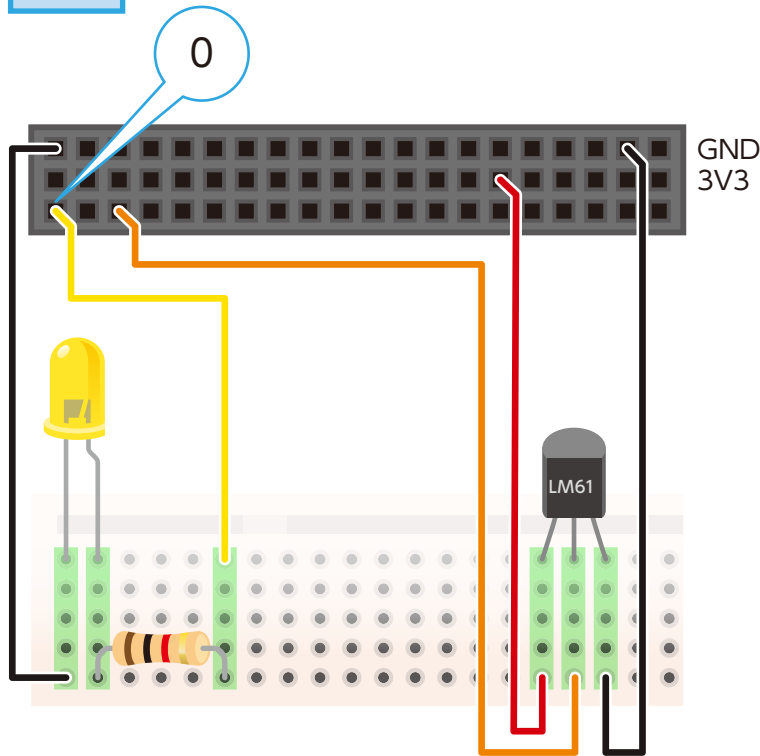
演習② 温度センサからのアナログ入力とLEDへのデジタル出力プログラム

計測した温度によってLEDの点灯と消灯を制御するプログラムを作成します。

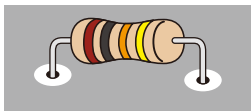
①演習①のブレッドボードに黄色LEDと抵抗を追加します。



接続例



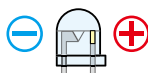
抵抗器



電流の流れを制限する部品です。抵抗の値は表面に記された4本の色帯で識別します。抵抗の値が大きくなるほど電流の流れる大きさは小さくなります。

LED

横から見たLED



短い側がマイナス

上から見たLED



平らな面がある側がマイナス

LED (Light Emitting Diode) はダイオードの一種です。電圧を与えることで発光する部品です。極性があるため、プラスとマイナスを間違えると発光しません。

②Studuino:bitとPCを接続します。

③黄色LEDを点灯するプログラムを記します。

下記プログラムを転送して実行し、黄色LEDが点灯することを確認してください。

LEDが点灯しない場合はブレッドボードの接続が間違っていないか確認してください。

```
pins.digitalWritePin(DigitalPin.P0, 1);
```

ポート0に接続された黄色LEDを点灯する

④計測した温度によってLEDの点灯と消灯を制御するプログラムを記します。

enshu2.js

```
let val = 0;
basic.forever(function () {
  val = Math.floor(330 * pins.analogReadPin(AnalogPin.P2) / 1023 - 60);
  basic.showNumber(val);
  basic.showString('C');
  if (val >= 30) {
    pins.digitalWritePin(DigitalPin.P0, 1);
  } else {
    pins.digitalWritePin(DigitalPin.P0, 0);
  }
})
```

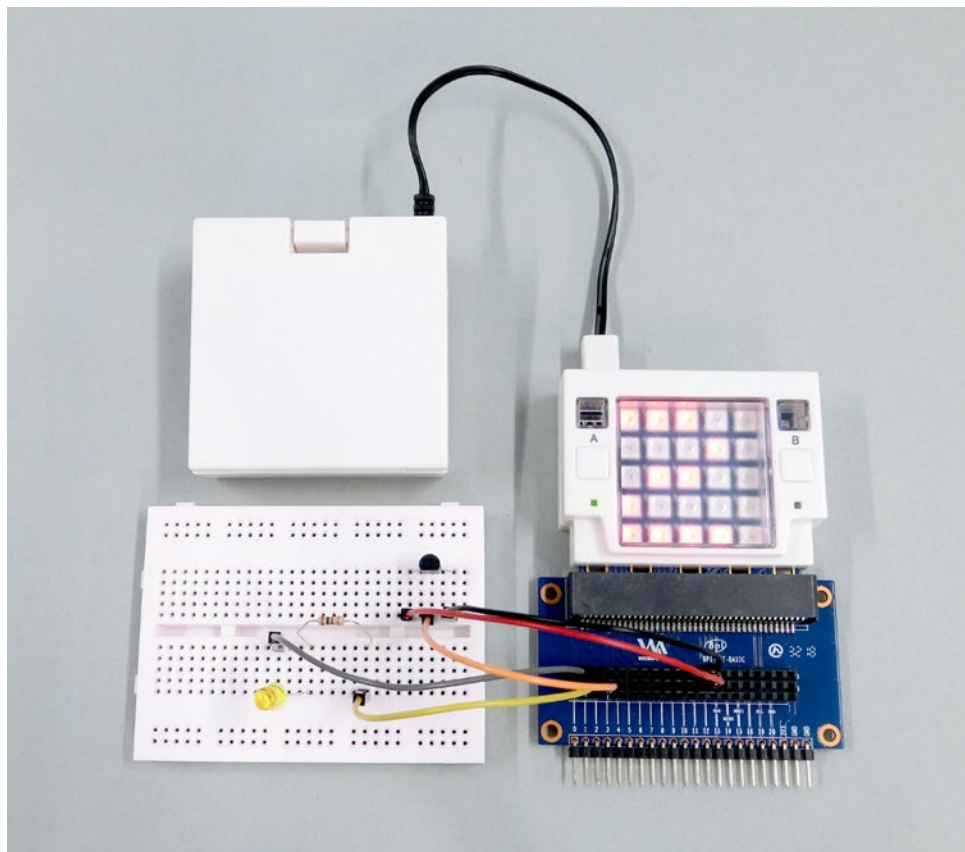
30℃以上のときLEDを
点灯する

30℃未満のときLED
を消灯する

QuickJSは、JavaScriptの標準規格ES2020で定義された構文をサポートしているため、変数の宣言 [let] を [var] に、無限ループ [basic.forever(function){}] を [while(true){}] に、 [}] を [}] に変更しても同様の動作を確認できます。

```
var val = 0;
while(true){
  val = Math.floor(330 * pins.analogReadPin(AnalogPin.P2) / 1023 - 60);
  basic.showNumber(val);
  basic.showString('C');
  if (val >= 30) {
    pins.digitalWritePin(DigitalPin.P0, 1);
  } else {
    pins.digitalWritePin(DigitalPin.P0, 0);
  }
}
```

⑤プログラムを転送して実行し、Stduino:bitのLEDディスプレイに温度がスクロール表示され、温度が30℃以上のときは黄色LEDが点灯、30℃未満のときは消灯することを確認してください。



演習③ 温度センサからのアナログ入力と2つのLEDへのデジタル出力プログラム

計測した温度によって2つのLEDの点灯と消灯を制御するプログラムを作成します。

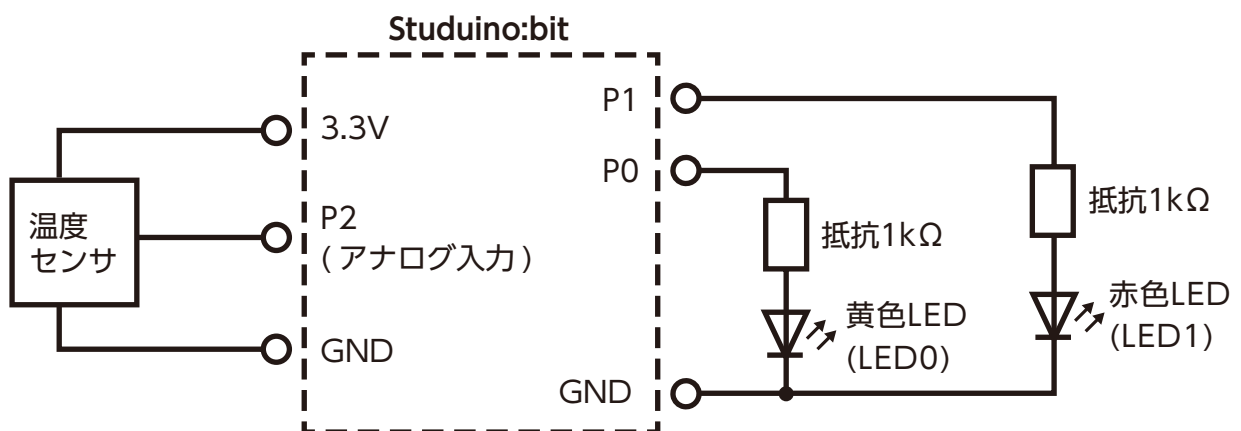
条件	LED 0	LED 1
室温 < 30℃	消灯	消灯
30℃ ≤ 室温 < 35℃	点灯	消灯
室温 ≥ 35℃	消灯	点灯

作例動画はコチラ

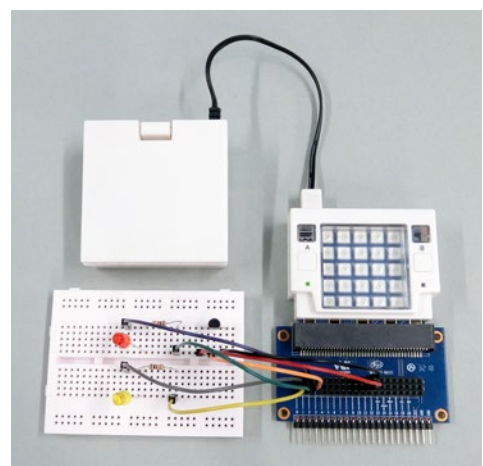
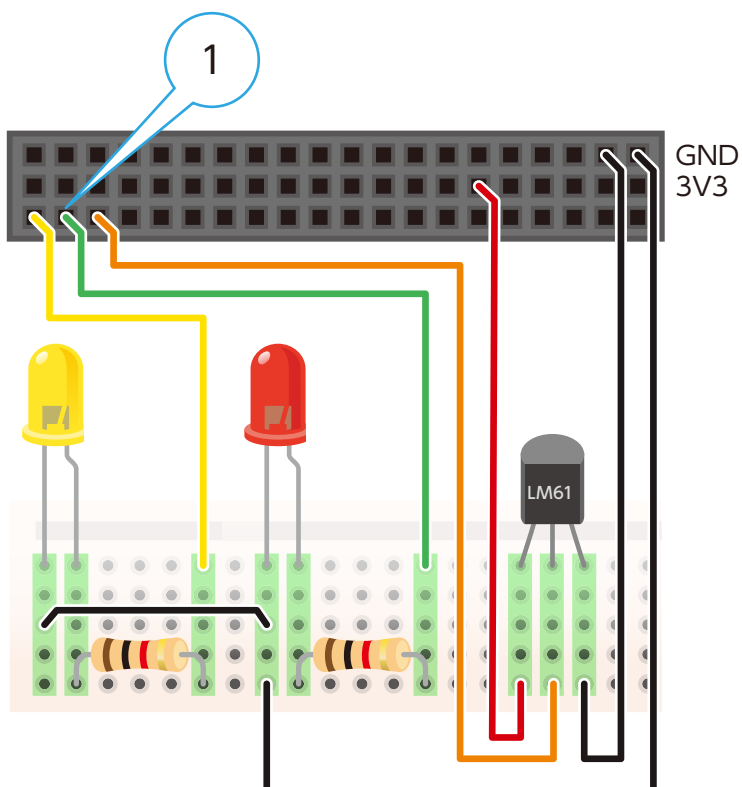


<https://youtu.be/XBnDEUUzKbU>

①演習②のブレッドボードに赤色LEDと抵抗を追加します。



接続例



②Studuino:bitとPCを接続します。

③赤色LEDを点灯するプログラムを記します。

下記プログラムを転送して実行し、赤色LEDが点灯することを確認してください。

LEDが点灯しない場合はブレッドボードの接続が間違っていないか確認してください。

```
pins.digitalWritePin(DigitalPin.P1, 1);
```

ポート1に接続された赤色LEDを点灯する

④計測した温度によってLEDの点灯と消灯を制御するプログラムを記します。

enshu3.js

```
let val = 0;
basic.forever(function () {
  val = Math.floor(330 * pins.analogReadPin(AnalogPin.P2) / 1023 - 60);
  basic.showNumber(val);
  basic.showString('C');
  if (val < 30) {
    pins.digitalWritePin(DigitalPin.P0, 0);
    pins.digitalWritePin(DigitalPin.P1, 0);
  } else if (30 <= val && val < 35) {
    pins.digitalWritePin(DigitalPin.P0, 1);
    pins.digitalWritePin(DigitalPin.P1, 0);
  } else {
    pins.digitalWritePin(DigitalPin.P0, 0);
    pins.digitalWritePin(DigitalPin.P1, 1);
  }
})
```

30°C未満のとき
黄色LED、赤色LEDを消灯する

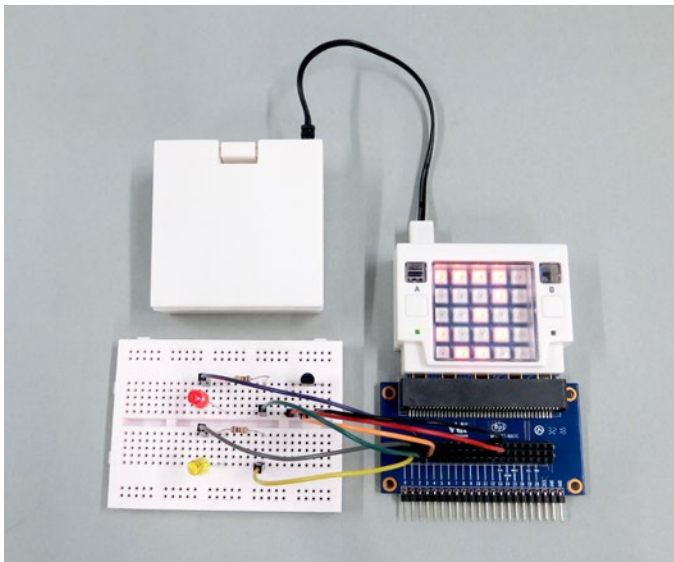
30°C以上35°C未満のとき
黄色LEDを点灯、赤色LEDを消灯する

35°C以上のとき
黄色LEDを消灯、赤色LEDを点灯する

QuickJSは、JavaScriptの標準規格ES2020で定義された構文をサポートしているため、変数の宣言「let」を「var」に、無限ループ「basic.forever(function(){})」を「while(true){}」に、「{）」を「{)」に変更しても同様の動作を確認できます。

```
var val = 0;
while(true){
  val = Math.floor(330 * pins.analogReadPin(AnalogPin.P2) / 1023 - 60);
  basic.showNumber(val);
  basic.showString('C');
  if (val < 30) {
    pins.digitalWritePin(DigitalPin.P0, 0);
    pins.digitalWritePin(DigitalPin.P1, 0);
  } else if (30 <= val && val < 35) {
    pins.digitalWritePin(DigitalPin.P0, 1);
    pins.digitalWritePin(DigitalPin.P1, 0);
  } else {
    pins.digitalWritePin(DigitalPin.P0, 0);
    pins.digitalWritePin(DigitalPin.P1, 1);
  }
}
```

⑤プログラムを転送して実行し、Studuino:bitのLEDディスプレイに温度がスクロール表示され、温度が30℃以上のときは黄色LEDが点灯し赤色LEDが消灯、35℃以上のときは黄色LEDが消灯し赤色LEDが点灯、30℃未満のときは2つのLEDが消灯することを確認してください。



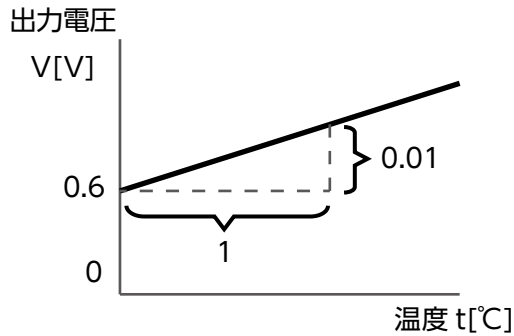
Scratch3.0に準拠しているStuduino:bitソフトウェア（ブロックプログラミング）で上記動作を確認する方法は付録Fを参照してください。

参考:温度とアナログ入力から取得したデジタル値の関係

温度センサの出力電圧Vと温度tの関係は式①のようになります。

$$V=0.01t+0.6$$

……式①

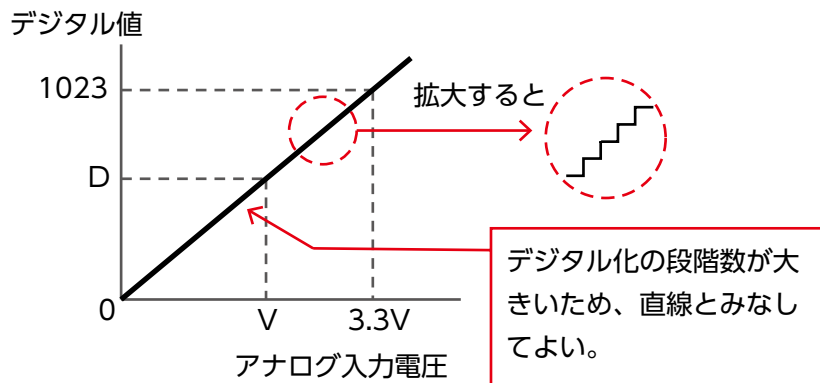


温度センサ LM61 の特性

ADコンバータのアナログ入力電圧Vとアナログ入力から取得したデジタル値Dの関係式は式②のようになります。

$$V = \frac{3.3D}{1023}$$

……式②



ADコンバータの特性

温度センサの出力電圧はADコンバータのアナログ入力電圧と等しいため、温度tとアナログ入力から取得したデジタル値Dの関係は式③と求まります。

$$V = \frac{330D}{1023} - 60$$

……式③

付録 A. Studuino:bitソフトウェアでのファームウェア更新手順

JavaScript専用ファームウェアを書き込まれたあとに、ブロックプログラミング、Pythonで利用される場合は「Studuino:bitソフトウェア」から「ファームウェア更新」を行ってください。

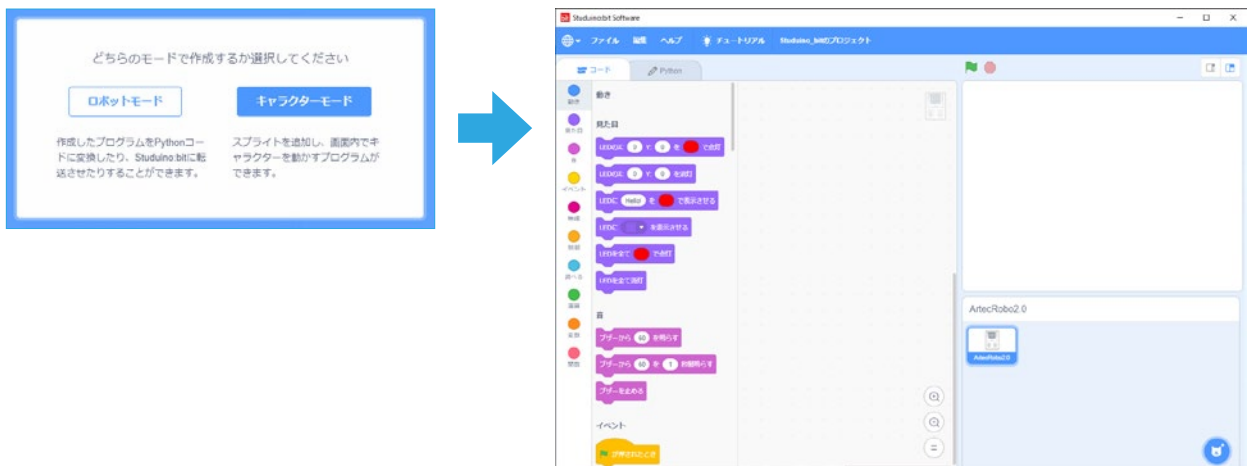
①下記URLの「インストール版ソフトウェアダウンロード」から対応OSのStuduino:bit ソフトウェアをダウンロードしてください。

ダウンロード・インストールする方法は、同ページ内にある「セットアップマニュアル」を参照してください。

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

②Studuino:bitとPCを接続します。

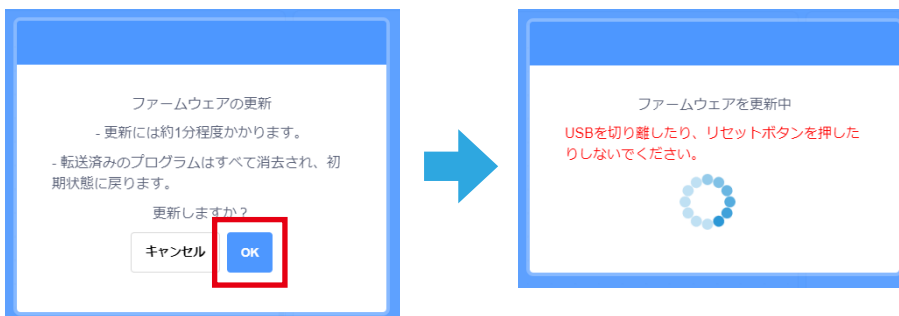
③「Studuino:bitソフトウェア」を起動し、「ロボットモード」または「キャラクターモード」を選択してください。



④上部の「ヘルプ」メニューから「ファームウェア更新」を選択します。



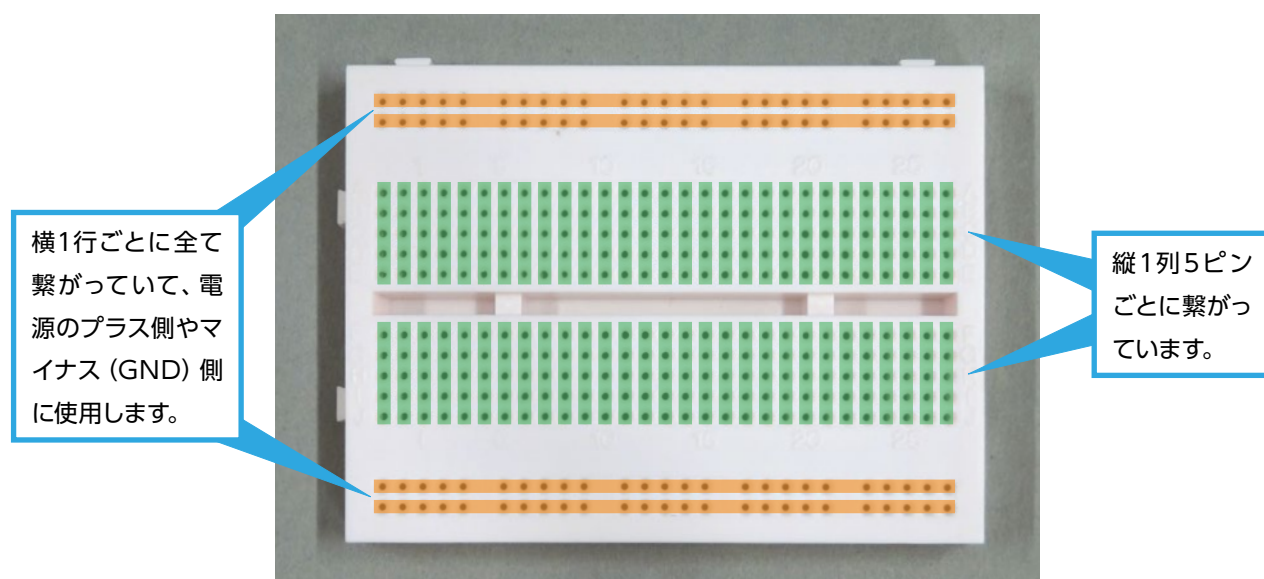
⑤Studuino:bitとUSB接続した状態で「OK」をクリックするとファームウェアの更新が始まります。



⑥更新が完了したら「OK」をクリックして、Studuino:bit本体のリセットボタンを押してください。

付録 B. ブレッドボードの使い方

ブレッドボードは内部が繋がっており、部品を差し込むだけで電子回路を組み立てることができます。



付録 C. 変換エッジコネクタの使い方

Studuino:bit下部のエッジコネクタ主要端子全てにアクセスすることを可能にする、変換基板です。ジャンプワイヤを介して、回路の実装などが簡単に行えます。



注意事項

エッジコネクタは金属がむき出しになっている部分があります。そこに金属製のものが接触してショートしないように注意してください。

付録 D. デジタル入出力アナログ入力オブジェクトの機能一覧

pinsオブジェクトは端子モジュールP0/P1/P2に対応しています。

DigitalPin.P0/DigitalPin.P1はデジタル入出力アナログ入力に対応しています。

AnalogPin.P2はアナログ入力のみに対応しているため、digitalWritePin(DigitalPin.P2,value)には対応していません。デジタル入力用のdigitalReadPinメソッドは使用できます。

メソッド	説明
digitalWritePin(DigitalPin, value)	value 引数が 1 の場合は High に、0 の場合は Low にデジタル信号を設定します。
digitalReadPin(DigitalPin)	デジタル信号を取得します。High の場合は 1 を、Low の場合は 0 を返します。
analogWritePin(AnalogPin, value)	PWM 信号を端子に出力します、value 引数は 0(0%) ~ 1023(100%) です。
analogReadPin(AnalogPin)	端子の電圧を読み取り、0(0V) から 1023(3.3V) までの間の整数値を返します。

付録 E. basicオブジェクトの機能一覧

メソッド	説明
showNumber(value,interval)	value(数字) を LED ディスプレイに表示します。1 桁以上ある場合は水平方向にスクロールさせます。 interval(数字) にはスクロールの速さをミリ秒で設定します。指定しない場合は 100 が設定されます。
showString(value,interval)	value(文字) を LED ディスプレイに表示します。画面よりも大きい場合は水平方向にスクロールさせます。 interval(数字) にはスクロールの速さをミリ秒で設定します。指定しない場合は 100 が設定されます。
clearScreen()	全ての LED の明るさを 0(オフ) に設定します。
forever()	プログラムの一部を繰り返し実行します。
pause(value)	プログラムを指定したミリ秒数一時停止します。

付録 F. Studuino:bit ソフトウェアを用いたプログラムの作成

Scratch3.0に準拠しているStuduino:bitソフトウェア（ブロックプログラミング）を用いて演習③のプログラムを作成します。

本付録に記載のプログラムや、演習①～②のStuduino:bitソフトウェアを用いたプログラム、関数やグラフを用いない場合のプログラムは下記URLよりダウンロードできます。

<https://www.artec-kk.co.jp/dl/98072/>

1.Studuino:bitソフトウェアでプログラムを作成する

①下記URLの「インストール版ソフトウェアダウンロード」から対応OSのStuduino:bitソフトウェアをダウンロードしてください。

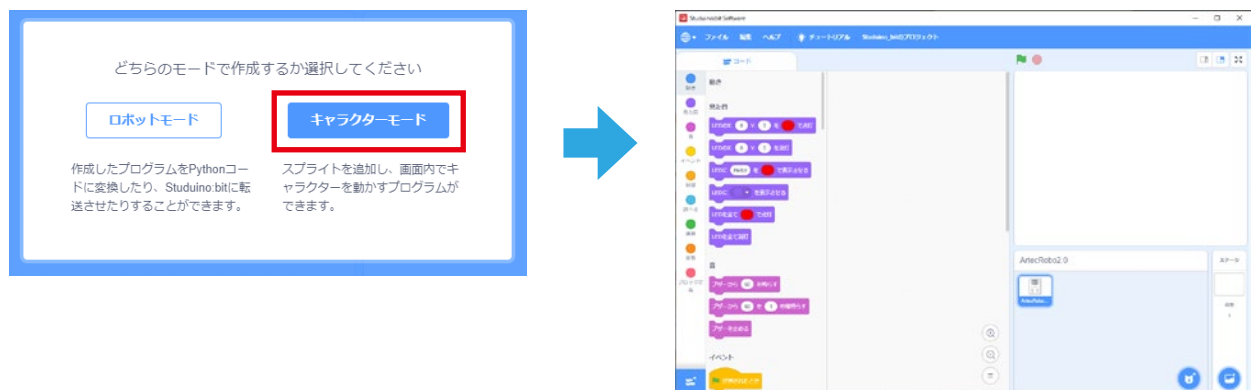
ダウンロード・インストールする方法は、同ページ内にある「セットアップマニュアル」を参照してください。

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

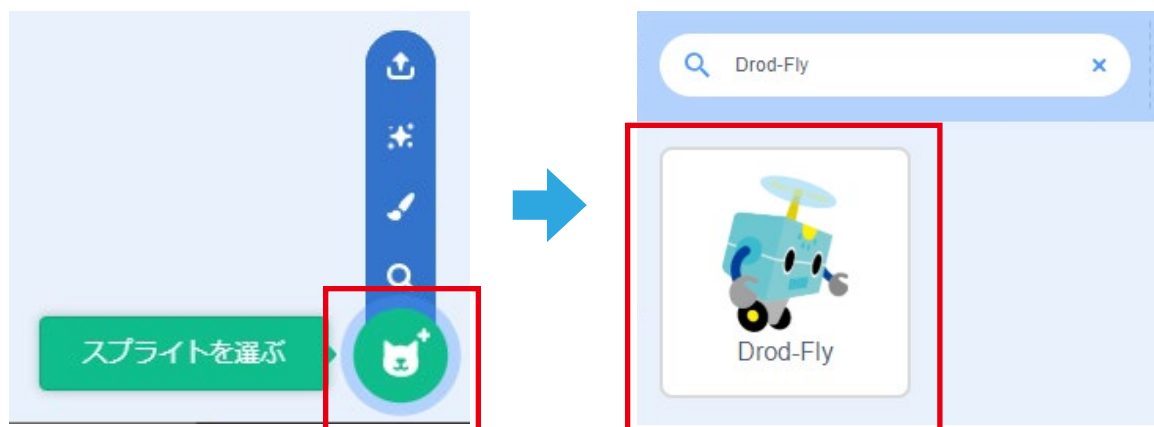
②Studuino:bitとPC接続します。

JavaScript専用ファームウェアが書き込まれている場合は、付録Aを参考にStuduino:bitのファームウェア更新を行ってください。

③ [Studuino:bit ソフトウェア] を起動し、「キャラクターモード」を選択してください。



④右下の「スプライトを選ぶ」から「Drod-Fly」を検索し、追加します。



◆ スプライトの切り替え

プログラムはスプライトごとに作成します。スプライトエリアの各スプライトをクリックすることで、それぞれのプログラム作成画面に切り替わります。



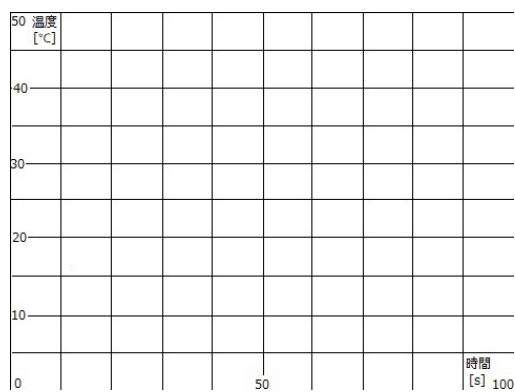
⑤ 使用する背景をダウンロードします。

下記URLよりStuduino:bitソフトウェア版のサンプルプログラムをダウンロードをして、展開（解凍）してください。

<https://www.artec-kk.co.jp/dl/98072/>

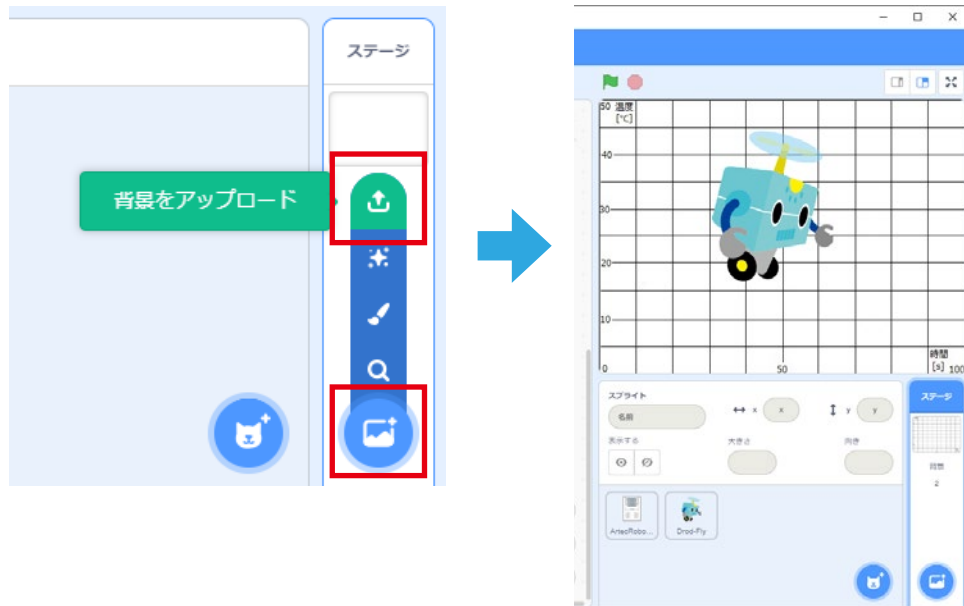
ダウンロードしたフォルダに含まれる画像ファイル「方眼紙.png」を背景に使用します。

※PCの設定によっては、拡張子「.png」が表示されない場合があります。

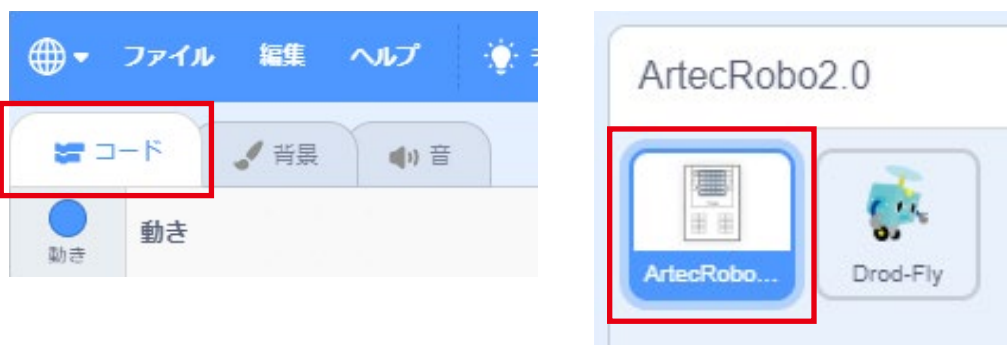


方眼紙.png

⑥右下の「背景を選ぶ」から、「背景をアップロード」を選択し⑤の「方眼紙.png」をアップロードします。



⑦左上のタブ「コード」を選択してスクリプトエリアを表示し「ArtecRobo2.0」の spriteを選択してプログラムを作成します。



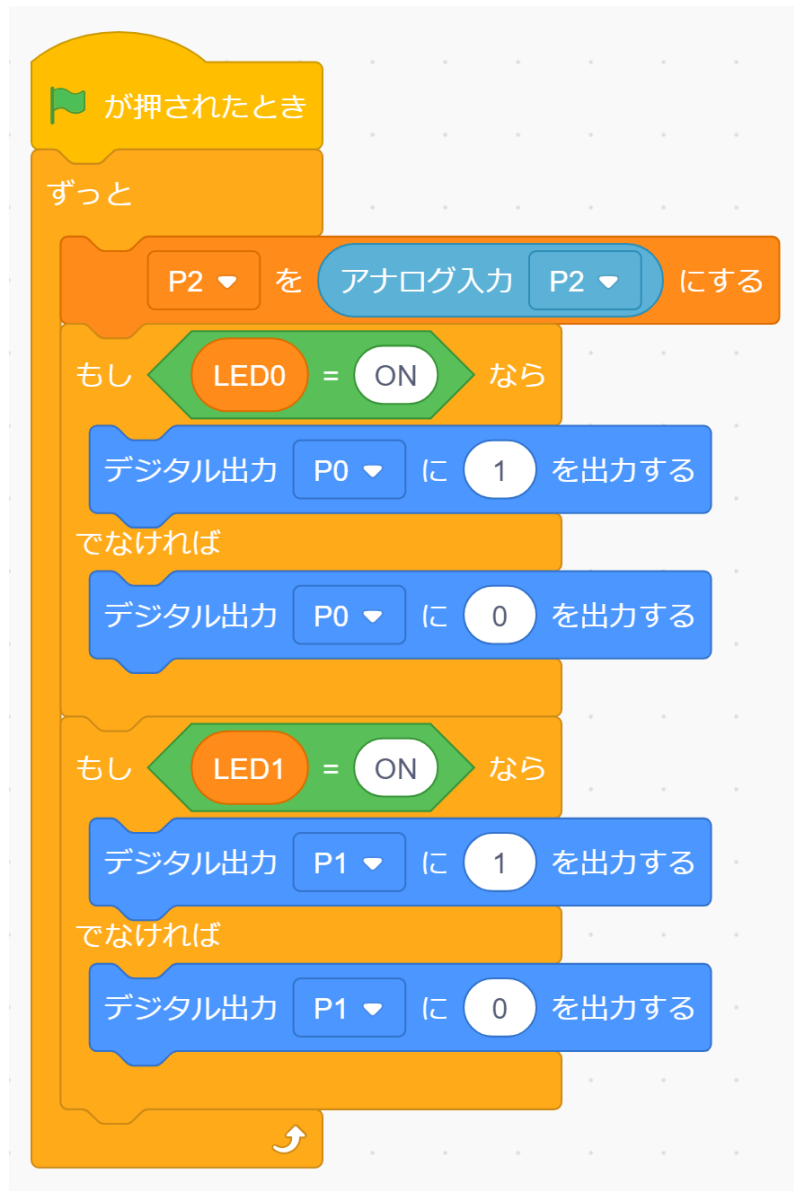
⑧「ArtecRobo2.0」のspriteを選択した状態で、汎用入出力ブロックを追加します。

上部の「編集」メニューから「拡張ブロック」、「汎用入出力」を選択してください。

「動き」ブロックグループに「デジタル出力」と「アナログ出力」ブロックが追加され、「調べる」ブロックグループに「デジタル入力」と「アナログ入力」ブロックが追加されます。



⑨ 「ArtecRobo2.0」 (=Studuino:bit) のスプライトを選択した状態で下記のプログラムを作成します。



⑩ [Drod-Fly] のスプライトを選択し、下記のプログラムを作成します。

The main script starts with setting size to 20% and x-coordinate to -240. It then erases everything and moves the pen down. A loop labeled 'ずっと' (forever) begins with a calculation: $\text{温度} = 330 * P2 / 1023 - 60$ (rounded to the nearest integer). It then says '温度 と °C と LED0 = と LED0 と LED1 = と LED1'.

Inside the loop, there are three conditional branches based on temperature:

- もし 温度 < 30 なら** (If temperature < 30):
 - Define '30未満' (30未満)
 - Set color effect to 0.
 - Set LED0 to OFF.
 - Set LED1 to OFF.
- もし 温度 > 30 または 温度 = 30 かつ 温度 < 35 なら** (If temperature > 30 or (temperature = 30 and temperature < 35)):
 - Define '30以上35未満' (30以上35未満)
 - Set color effect to 20.
 - Set LED0 to ON.
 - Set LED1 to OFF.
- もし 温度 > 35 なら** (If temperature > 35):
 - Define '35以上' (35以上)
 - Set color effect to 100.
 - Set LED0 to OFF.
 - Set LED1 to ON.

The loop ends with a 'グラフ描画' (Graph Drawing) block and a '0.165 秒待つ' (wait 0.165 seconds) block.

定義 30未満

- 色 の効果を 0 にする
- LED0 を OFF にする
- LED1 を OFF にする

定義 30以上35未満

- 色 の効果を 20 にする
- LED0 を ON にする
- LED1 を OFF にする

定義 35以上

- 色 の効果を 100 にする
- LED0 を OFF にする
- LED1 を ON にする

スペース キーが押されたとき

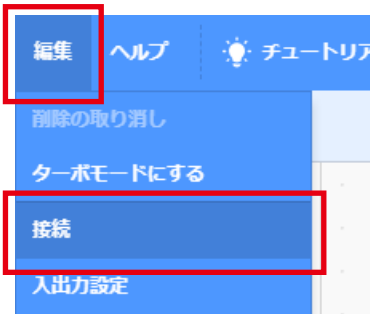
- 全部消す
- ペンを上げる
- x座標を 0、y座標を 0 にする
- 大きさを 100% にする
- すべてを止める

定義 グラフ描画

- x座標を 1 ずつ変える
- もし x座標 > 239 なら
 - x座標を -240 にする
 - 全部消す
- y座標を $2376 * P2 / 1023 - 612$ にする

この式の導出方法はP.30を参照してください。

⑪上部の「編集」メニューから「接続」を選択し、Studuino:bitと接続します。



⑫Studuino:bitと接続されると、センサーボードが表示されます。

汎用入出力の[P2]アナログ入力の値が変化することを確認してください。

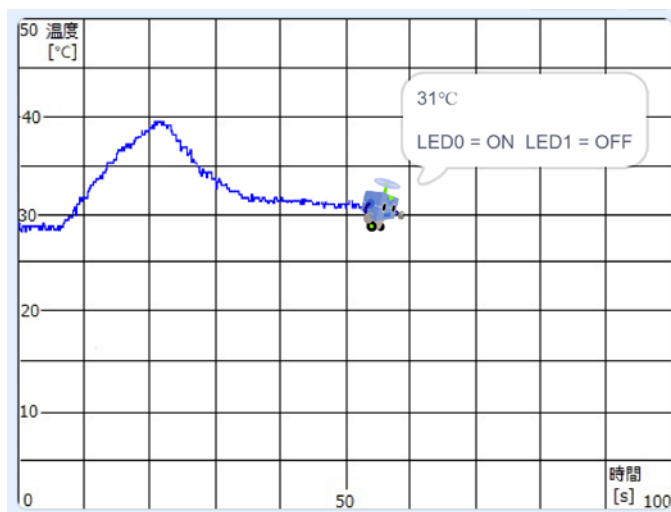
値が変化しない場合はブレッドボードの接続が間違っていないか確認してください。

センサーボード	
Studuino:bit	▼
ボタンA	1
ボタンB	1
光センサー	66
温度センサー	30.55
モーションセンサー	▲
汎用入出力	▼
[P2] アナログ入力	273

⑬ステージの上の「緑の旗」をクリックしてプログラムを実行します。



⑭プログラムを実行すると下記のように表示されます。



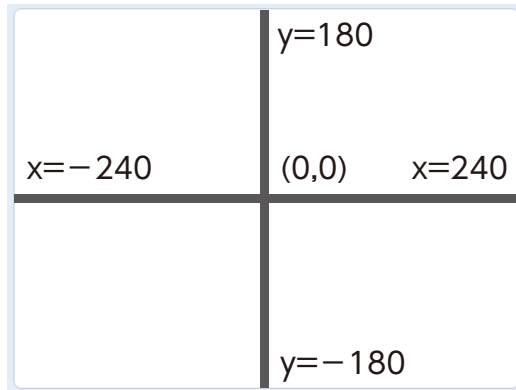
プログラムの停止

キーボードのスペースキーを押すと、プログラムが停止され、描画したグラフも消去されて初期状態に戻ります。

ステージ上の「赤丸」をクリックするとプログラムは停止されますが、描画したグラフは消去されません。

参考:スプライトのy座標とアナログ入力から取得したデジタル値の関係

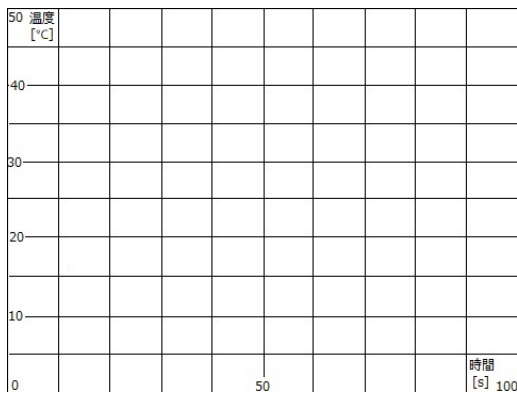
Studuino:bitソフトウェアのステージのx方向は-240~240、y方向は-180~180です。



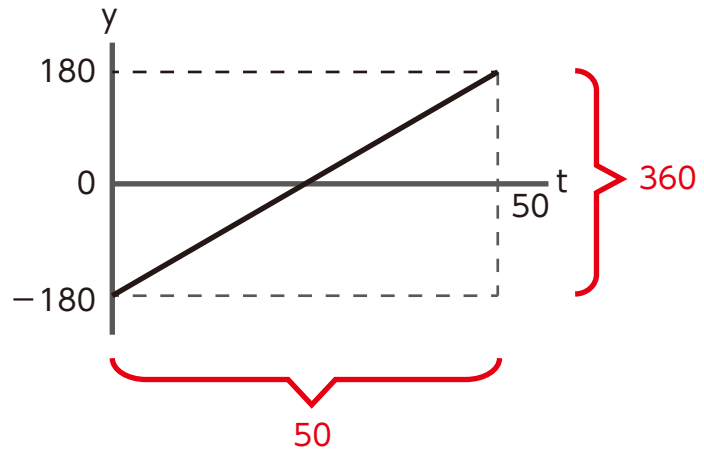
ステージ

Studuino:bitソフトウェアのステージの座標を背景の方眼紙の座標に変換します。

温度 t とし、スプライトのy座標を y とすると式④のようになります。



方眼紙



$$y = \frac{360}{50} t - 180 \quad \dots\dots\text{式④}$$

スプライトのy座標とアナログ入力から取得したデジタル値Dの関係は上記式④とP.20の式③より、式⑤と求められます。

$$V = \frac{330D}{1023} - 60 \quad \dots\dots\text{式③}$$

$$y = \frac{2376D}{1023} - 612 \quad \dots\dots\text{式⑤}$$

Artec Robo
アーテックロボ

計測・制御とプログラミング

JavaScript 版 教員用

株式会社 **アーテック** お客様相談窓口



◀Webからのお問い合わせはこちら
<https://www.artec-kk.co.jp/contact/>

お電話でのお問い合わせはこちら
TEL 072-990-5656