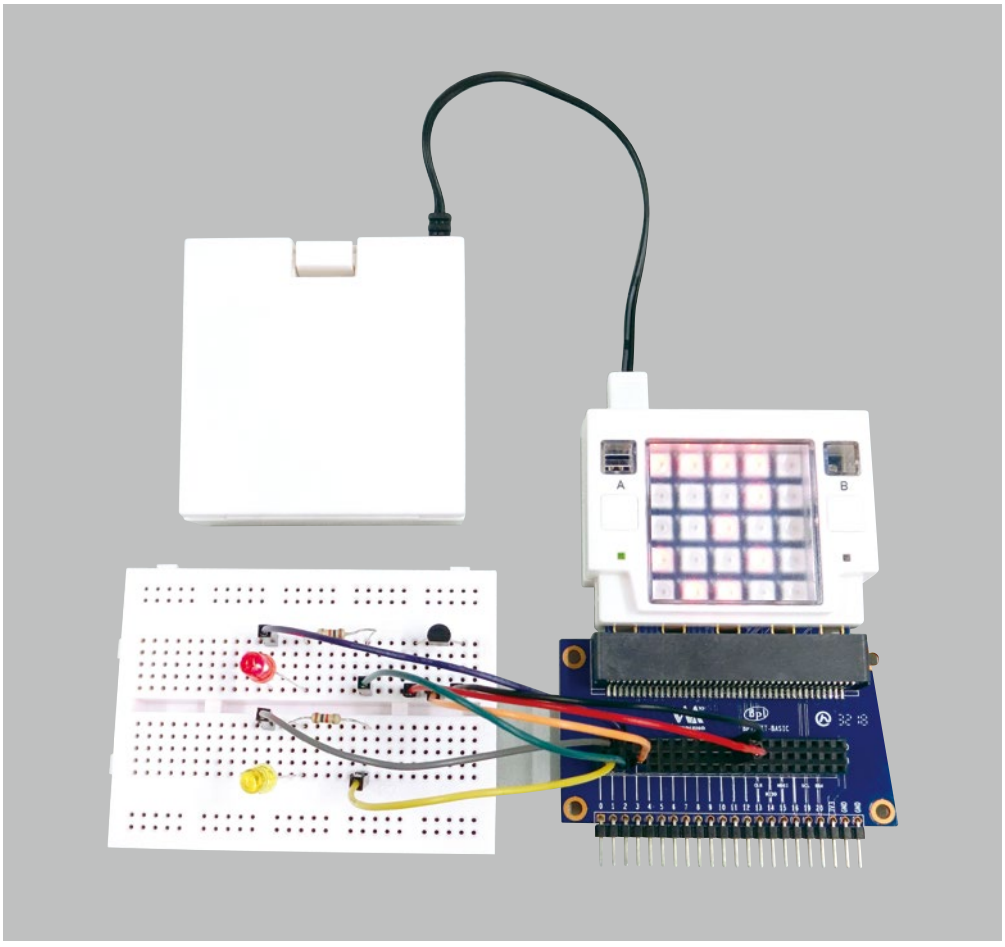
Studuino®:bit
スタディーノ ビット

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

Python 版

教員用



ArTeC®

目 次

準備

演習

付録

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

1. ソフトウェアのインストール

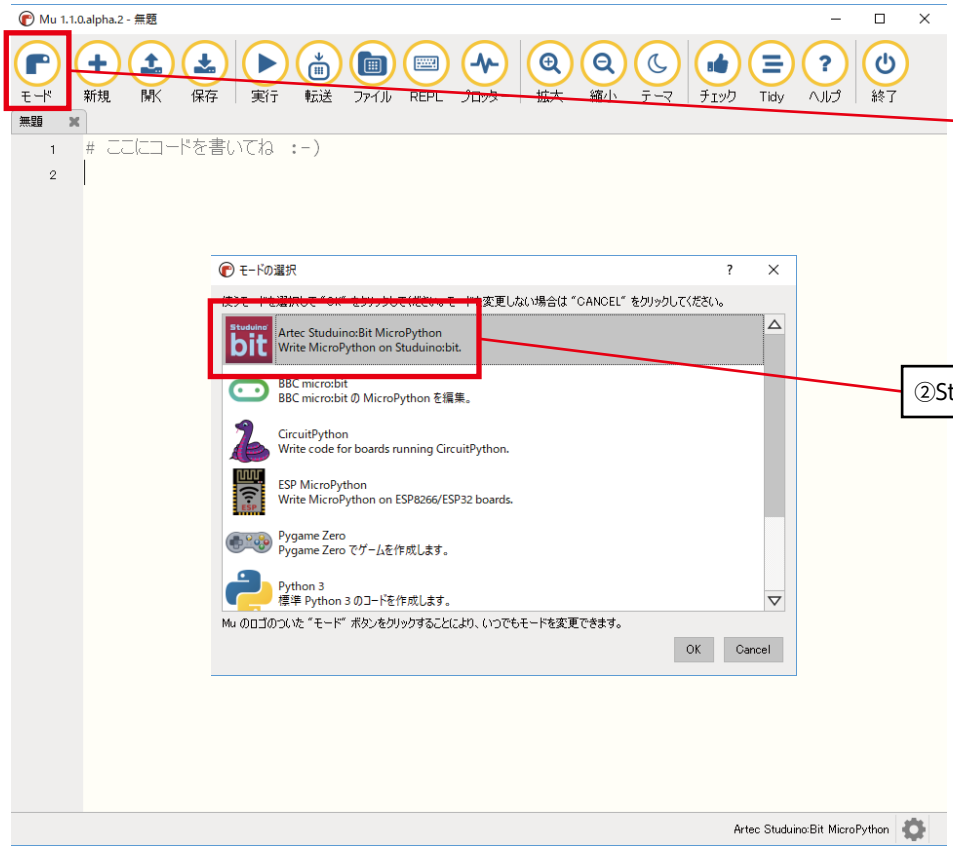
PythonでStuduino:bitのプログラムを作成するために必要なプログラム開発環境をインストールします。
ここではStuduino:bit用に機能拡張されたMuエディタを使用します。
下記ArtecRobo2.0のPythonページ内にあるドキュメントを参考に、Studuino:bit用Muエディタをダウンロード、インストールしてください。

https://www.artec-kk.co.jp/artecrobo2/ja/software/python.php

Studuino:bitなどのボード型コンピュータで動作するPythonは、普通のPythonではなく、MicroPythonです。MicroPythonはStuduino:bitなどのボード型コンピュータで実行できるように実装されたPython言語の処理系で、Python3.0との間で高い互換性を有します。MicroPythonを使ったプログラムはMuエディタなどを使って作成します。

2.REPL でプログラムを実行する

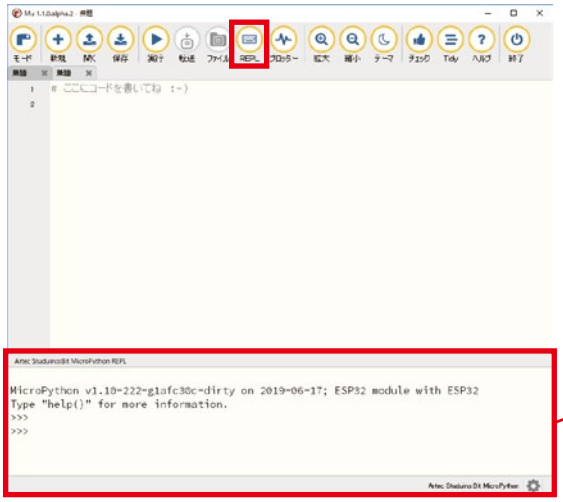
Python の対話型インタープリタ REPL (Read-Evaluate-Print Loop) を使用する手順を下記に記します。
①Studuino:bit と PC を接続します。
②Mu を起動して、下図のように「モード」ボタンをクリックして、「Artec Studuino:bit MicroPython」をダブルクリックしてください。



①モードをクリック

②Studuino:bit をダブルクリック

③「REPL」ボタンをクリックすると、Mu ウィンドウの下にウィンドウが表示され、入力待ちを示す「>>>」が表示されます。
REPL 表示中は「転送」ボタン、「ファイル」ボタンは灰色表示になり使用できません。
再度「REPL」ボタンをクリックすることでウィンドウが非表示になり、「転送」ボタン、「ファイル」ボタンを使用できるようになります。



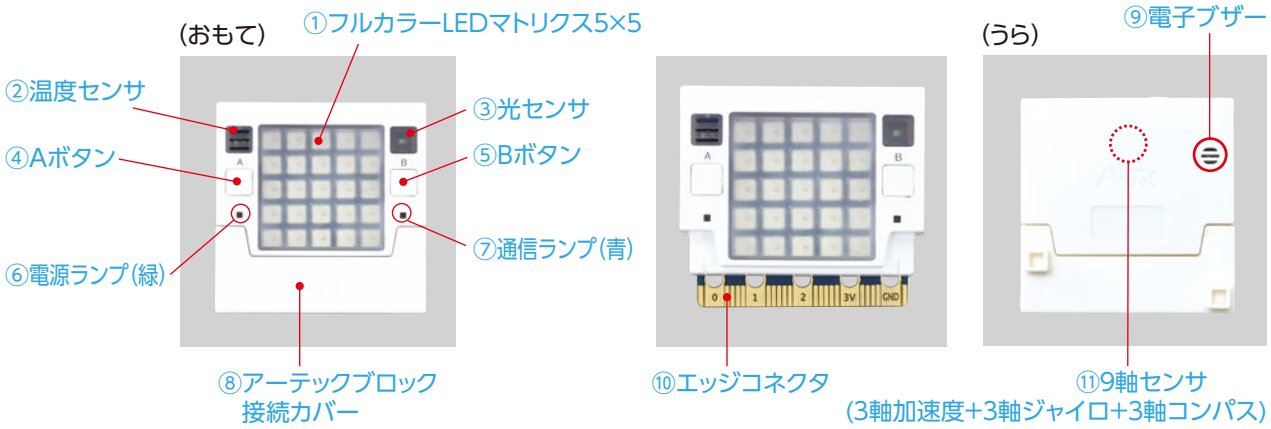
このウィンドウで REPL を使用します。

④REPL の自動エコーによって、ウィンドウの「>>>」の後にプログラムを入力すると、すぐに結果を見ることができます。REPL で下記を実行し、"Hello World"が表示されることを確認してください。

```
>>>print('Hello World')
```

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。



Studuino:bitライブラリを使用することで、Studuino:bitのハードウェアを簡単に制御できます。ライブラリには、Studuino:bitの各ハードウェアに対応したオブジェクトが定義されています

番号	ハードウェア	オブジェクト	機能概要
①	フルカラーLEDマトリクス５×５	display	各LEDのON/OFFやLEDを使ったアニメーションなど
②	温度センサ	temperature	温度センサの値の取得
③	光センサ	lightsensor	光センサの値の取得
④	Aボタン	button_a	AボタンのON/OFF判定など
⑤	Bボタン	button_b	BボタンのON/OFF判定など
⑨	電子ブザー	buzzer	電子ブザーで鳴らす音の高さをON/OFFなど
⑩	エッジコネクタ	p0～p16、p19、p20	エッジコネクタの入出力制御
		pin0～pin3	micor:bit 互換エッジコネクタの入出力制御 (※)
⑪	３軸加速度センサ	accelerometer	加速度センサの値の取得
	３軸ジャイロセンサ	gyro	ジャイロセンサの値の取得
	３軸地磁気センサ	compass	地磁気センサの値の取得

※:pin0とp0、pin1とp1、pin2とp2、pin3とp3の併用はできません。
※:pin0～pin3オブジェクトは、p0～p3オブジェクトと同じメソッドが使用できますが、read_analog()メソッドは0～1023の値を返します。

各オブジェクトは、pystubit.boardモジュールに定義されています。

REPLで下記を実行し、Studuino:bitメインユニットの画面に文字列”Hello, MicroPython”がスクロール表示されることを確認してください。

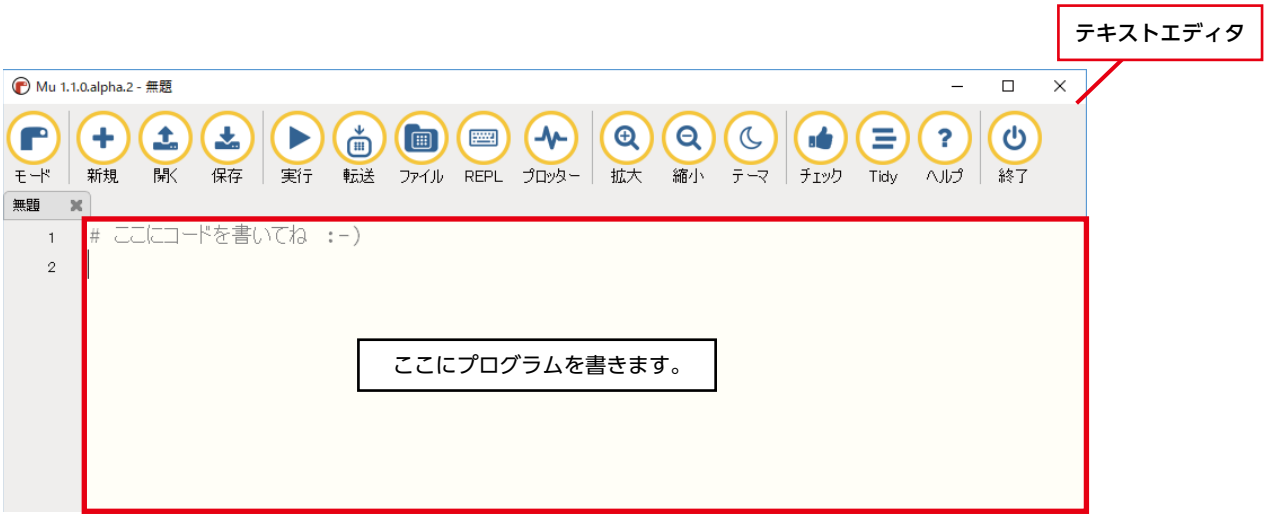
```
>>> from pystubit.board import display
>>> display.scroll('Hello, MicroPython')
```

4.「実行」ボタンでプログラムを実行する

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

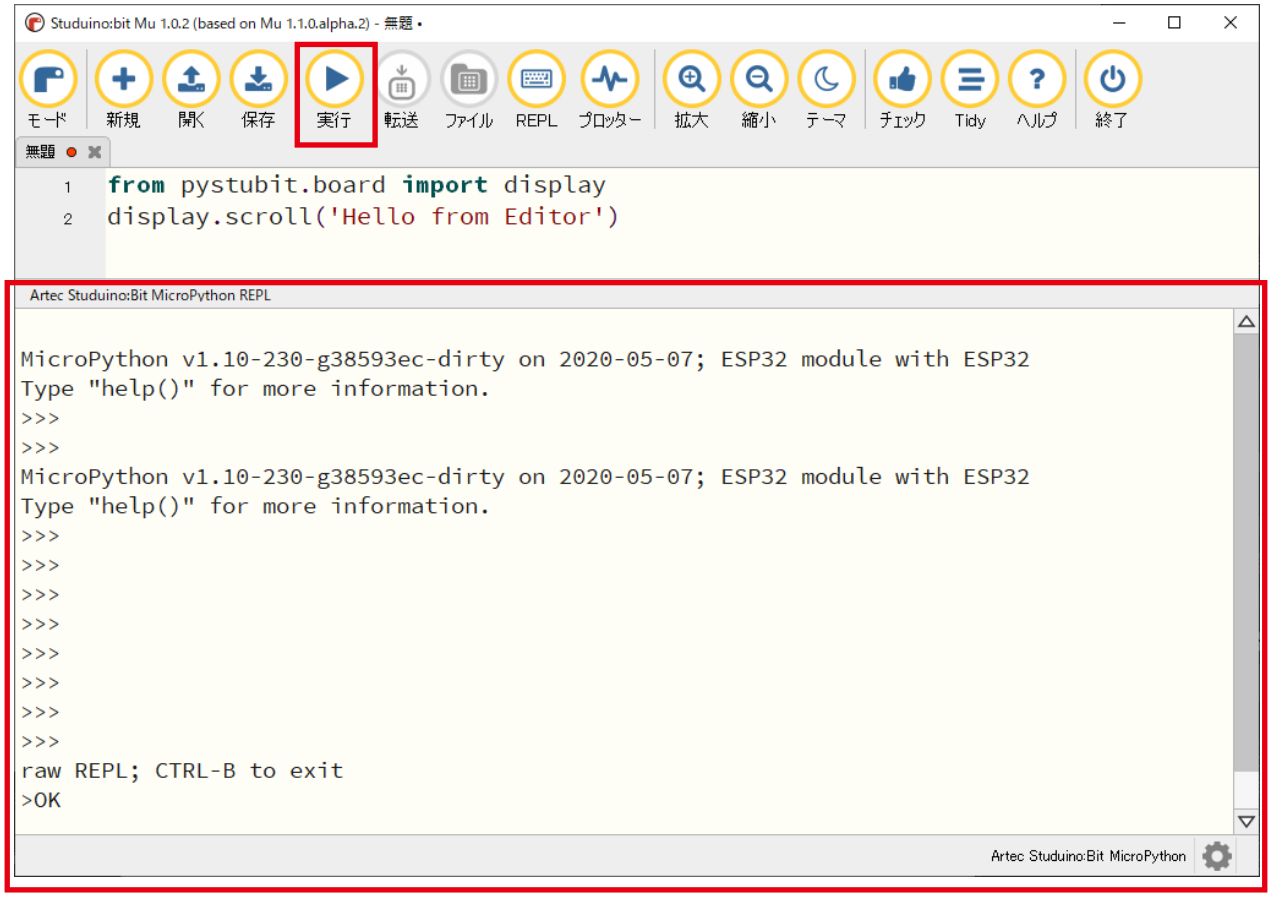
①下記をテキストエディタに入力します。

```
from pystubit.board import display
display.scroll('Hello from Editor')
```



②「実行」ボタンをクリックするとStuduino:bitでプログラムが実行されます。

Studuino:bitメインユニットの画面に文字列"Hello from Editor"がスクロール表示されることを確認してください。



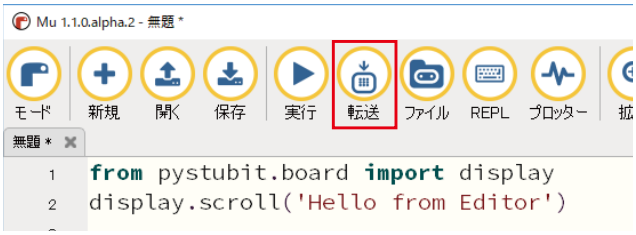
「実行」ボタンをクリックすると下側にウィンドウが表示され、REPLを使用できるようになります。

5.「転送」ボタンでプログラムを実行する

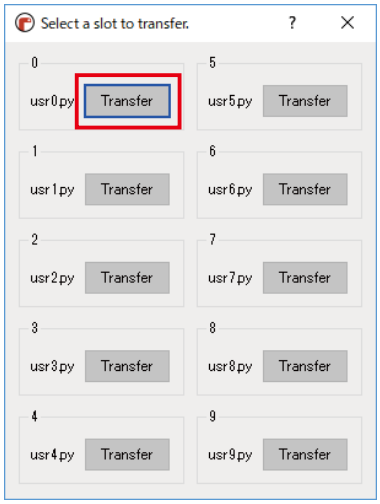
Muエディタで編集したPythonスクリプトをStuduino:bitに転送し、実行する手順を下記に記します。

① 4.①のプログラムをテキストエディタに入力し、「転送」ボタンをクリックします。

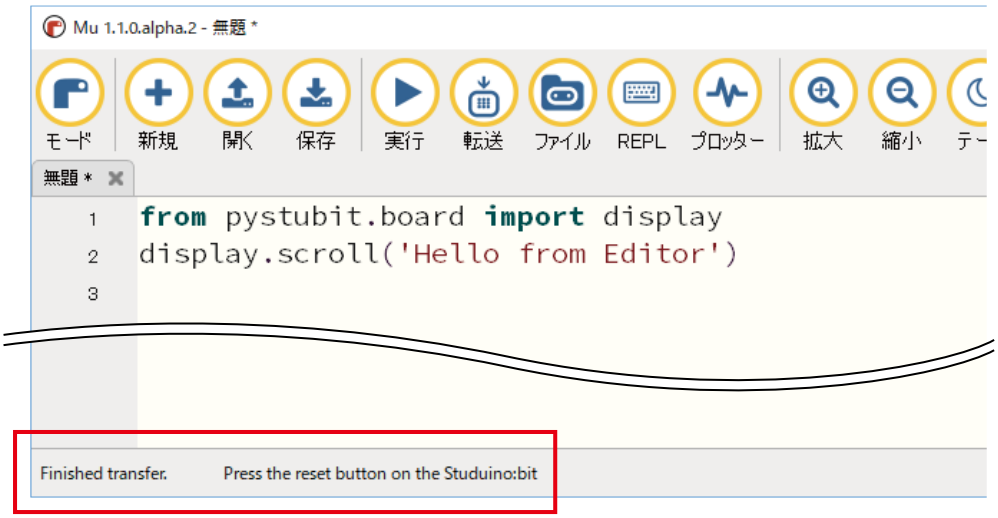
※「転送」ボタンが無効（灰色表示）になっている場合は「REPL」ボタンをクリックし、REPLウィンドウを閉じることで「転送」ボタンが有効になります。



②表示されるダイアログで、プログラムの転送先0～9のいずれかの「Transfer」ボタンをクリックします。



ステータスバーに「Finished transfer.」と表示されたら、転送完了です。



Studuino:bitメインユニットの画面に文字列"Hello from Editor"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ーンで動作を確認できます。

複数のプログラムを転送した場合の実行方法

複数のプログラムを 0 ～ 9 の番号を指定して転送することができます。

通常は最後に転送したプログラムが実行されますが、以下の方法で番号を指定して転送済みのプログラムを実行することができます。

①電源を接続する（通常起動）

(1) 電池ボックスもしくは USB ケーブルをメインユニットに接続します。

(2) 電源ランプ（緑）が点灯し、電源が自動的に ON になり起動します。

通常起動時は前に実行したプログラムが自動的に実行されます。

②プログラム選択モードへの移行

(1) A ボタンを押したまま、リセットボタンを押してください。
リセットボタンを押すと再起動のため一時的に電源ランプが消灯します。

(2) リセットボタンを離して 3 秒後に電源ランプが再点灯したら、
A ボタンを離します。

(3) 緑の LED で 0 が表示されるとプログラム選択モードへ移行しています。

③プログラムの選択・起動

(1) プログラム選択モード時に A ボタンを押すと、表示が 0・1・2・・・と切り替わります。9 まで切り替わると 0 に戻ります。

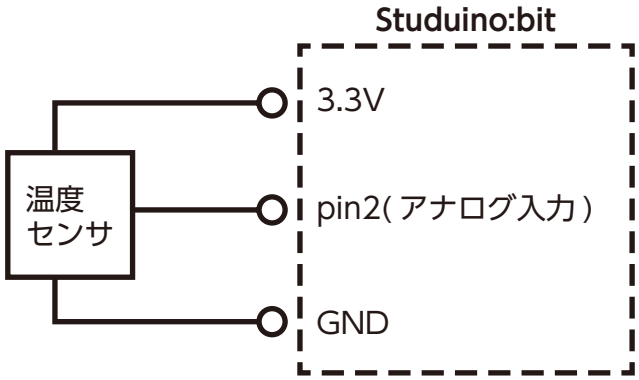
(2) 選択した番号で B ボタンを押すと、各番号に割り振られたプログラムが実行されます。

次回より通常起動時に (2) で選択したプログラムが自動で実行されます。

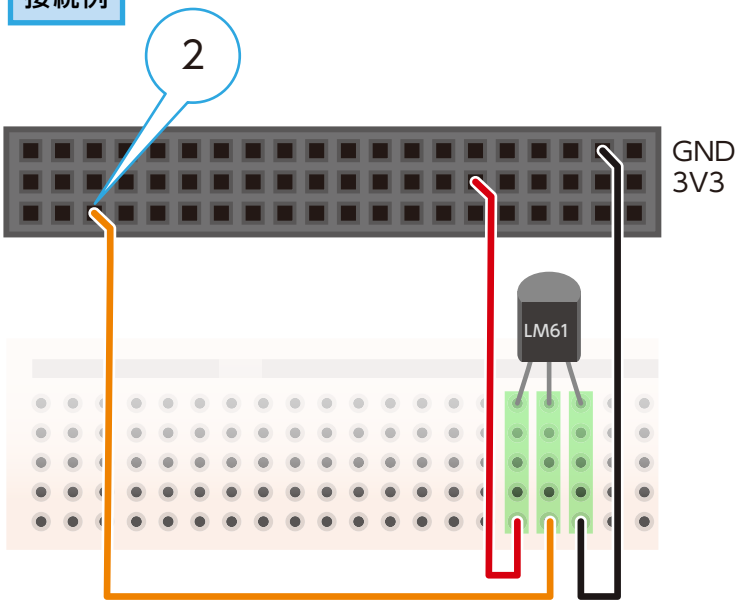
※工場出荷段階で動作確認用に各番号にはサンプルプログラムが書き込まれています。
※サンプルプログラムはソフトウェアを使用して新たなプログラムを書き込むと上書きされます。

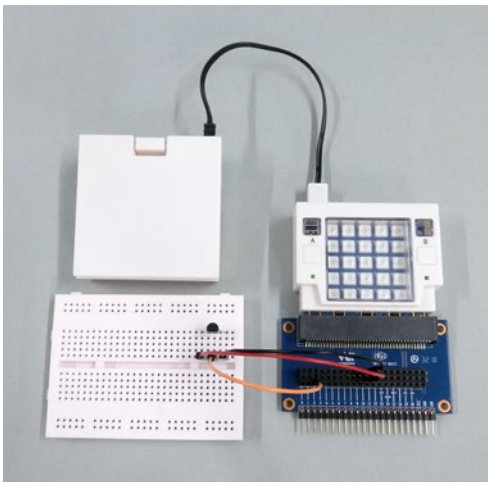
演習① 温度センサからのアナログ入力プログラム

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A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Bを参照してください。

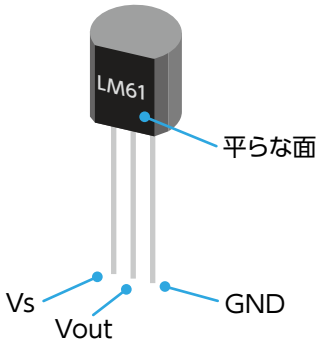


接続例





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℃ (LM61CIZ) －25 ～ +85℃ (LM61BIZ)
電源電圧	2.7 ～ 10V
出力電圧	10mV/℃
温度 0℃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、1秒ごとにREPLに表示するプログラムを記します。
下記をテキストエディタに入力し、「実行」ボタンを押すと、REPLに0～1023の間の整数が表示されます。
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Cを参照してください。

```
from pystubit.board import pin2
import time
while True:
    val = pin2.read_analog()
    print(val)
    time.sleep(1)
```

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。

```
6       time.sleep(1)
Artec Studuino:Bit MicroPython REPL
268
274
271
273
274
274
275
274
273
```

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。displayオブジェクトの機能詳細は付録Dを参照してください。

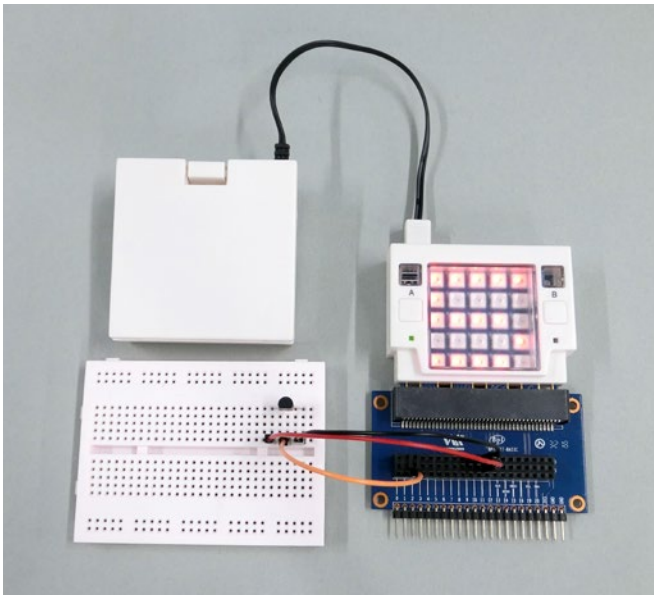
enshu1.py

```
from pystubit.board import pin2, display
while True:
    val =int(330*pin2.read_analog()/1023 - 60)
    display.scroll(str(val)+'C')
```

センサ値を温度に変換 (P.16の式③参照)

LEDディスプレイに表示

上記プログラムを実行してStuduino: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。



⑤Muエディタの機能を使い、温度をグラフ表示するプログラムを記します。

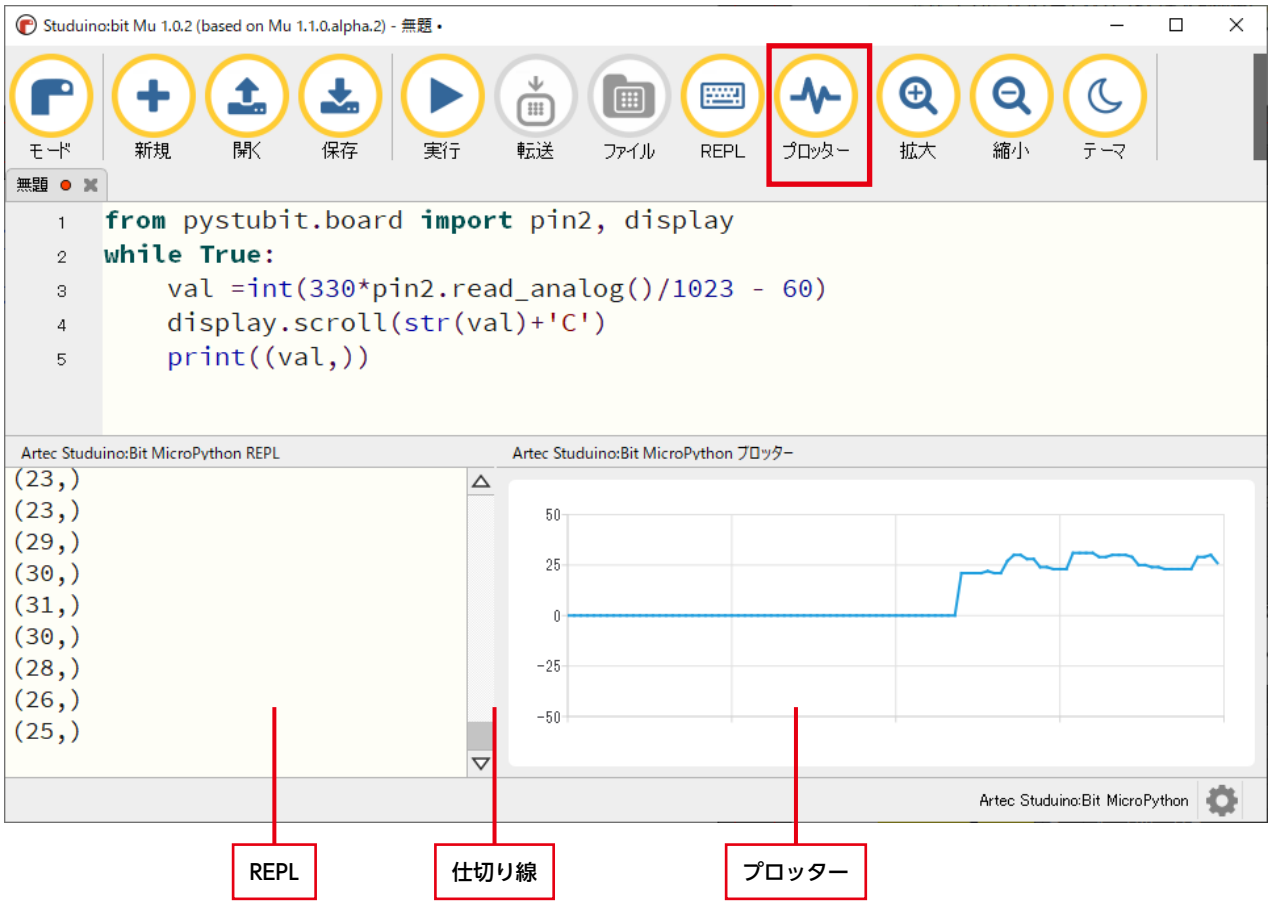
```
from pystubit.board import pin2, display
while True:
    val =int(330*pin2.read_analog()/1023 - 60)
    display.scroll(str(val)+'C')
    print((val,))
```

センサ値を温度に変換 (P.16の式③参照)

LEDディスプレイに表示

REPLとプロッターに表示

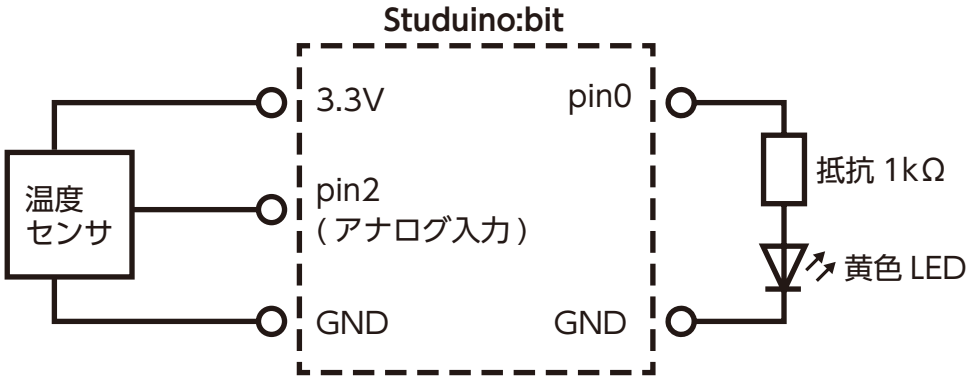
上記プログラムを実行すると、REPLに温度が表示されます。
メニューの「プロッター」ボタンをクリックすると、プロッターに温度がグラフ表示されます。
REPLとプロッターの間の仕切り線を左右にドラッグすると、それぞれの幅を変えることができます。
printの引数の「(val,)」は、タプル（一次元配列）で指定するため丸括弧 () で囲み、要素が1個のタプルの末尾にはカンマ「,」が必要です。「print((val,))」を「print(val)」に変更すると、REPLには表示されますが、プロッターには表示されません。プロッターに表示するデータはタプルで指定する必要があります。
また、「print((val1, val2, val3))」のように指定すると、複数データのグラフを表示することもできます。



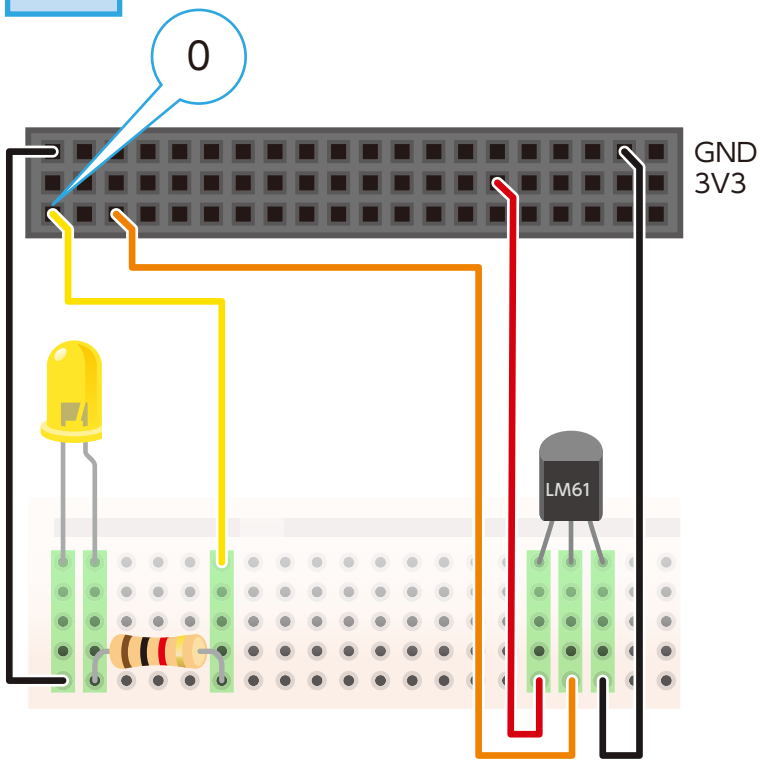
演習② 温度センサからのアナログ入力とLEDへのデジタル出力プログラム

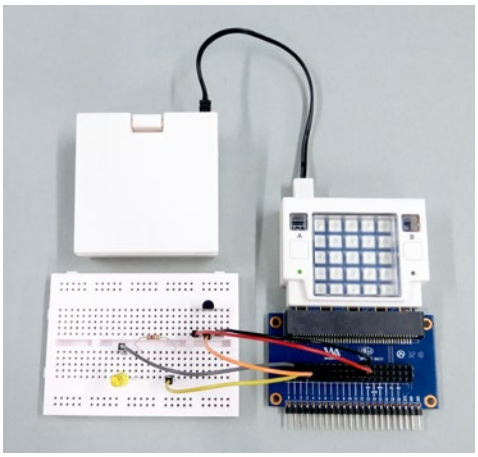
計測した温度によってLEDの点灯と消灯を制御するプログラムを作成します。

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。

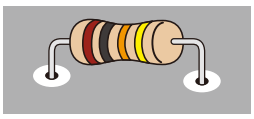


接続例



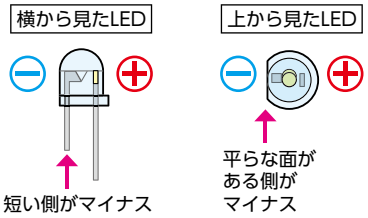


抵抗器



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



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。

```
from pystubit.board import pin0
pin0.write_digital(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。

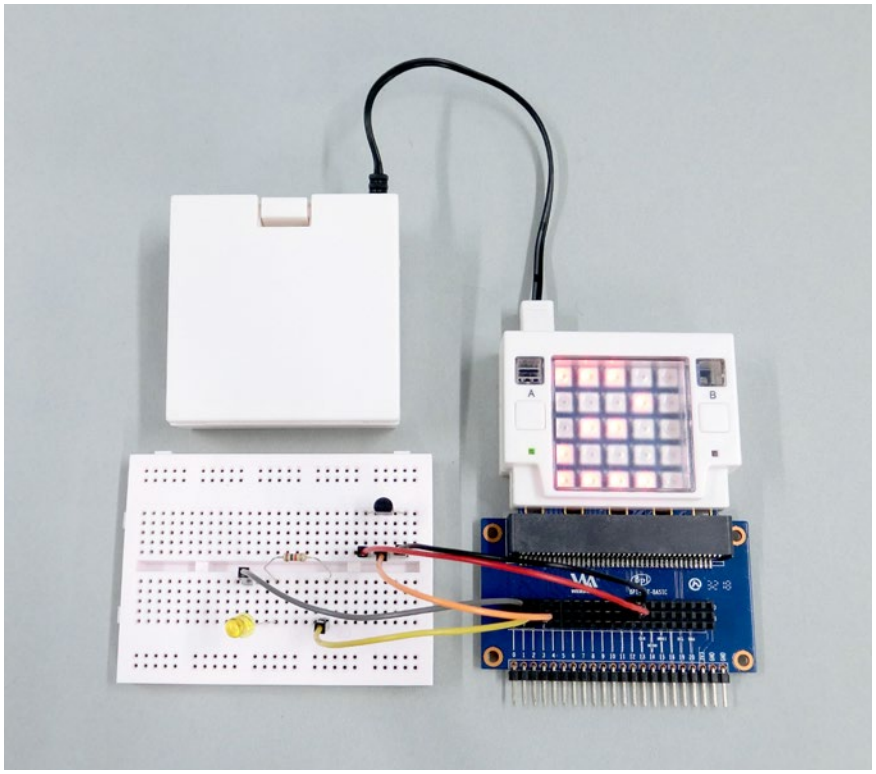
enshu2.py

```
from pystubit.board import pin0, pin2, display
while True:
    val =int(330*pin2.read_analog()/1023 - 60)
    display.scroll(str(val)+'C')
    if val >= 30:
        pin0.write_digital(1)
    else:
        pin0.write_digital(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

上記プログラムを実行して、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、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

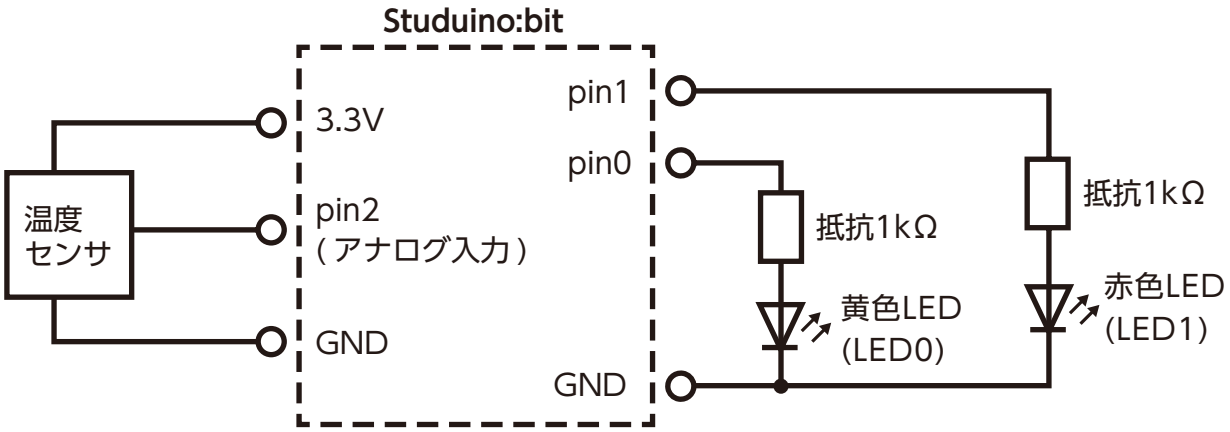
計測した温度によって２つのLEDの点灯と消灯を制御するプログラムを作成します。

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

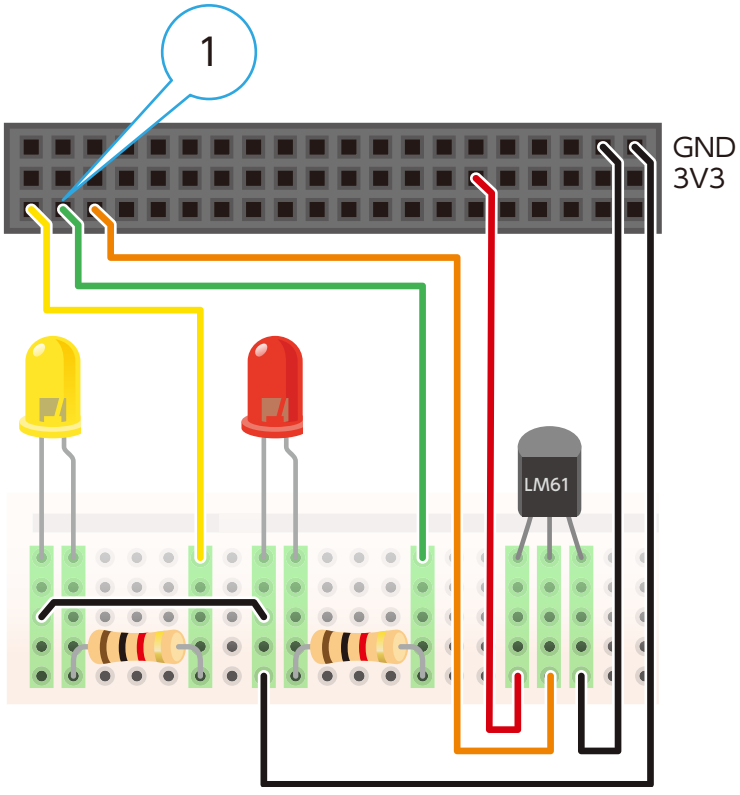
作例動画はコチラ

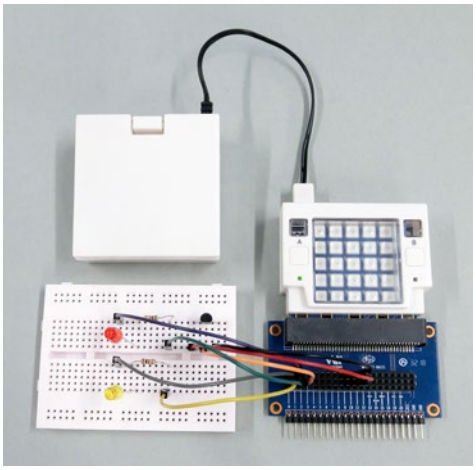
https://youtu.be/XBnDEUUzKbU

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。

```
from pystubit.board import pin1
pin1.write_digital(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.py

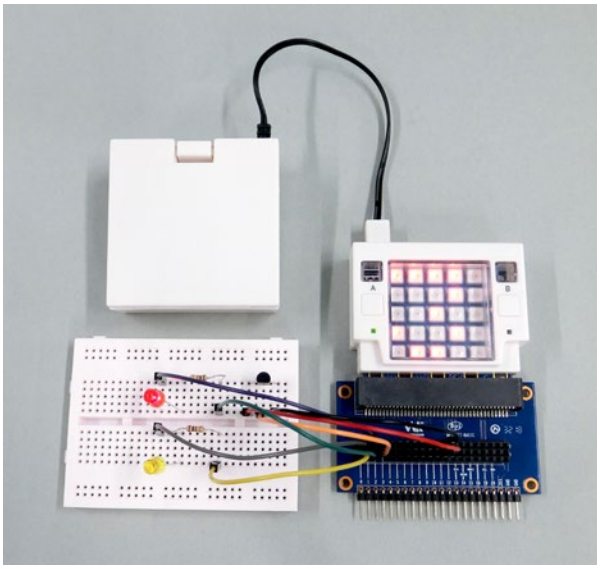
```
from pystubit.board import pin0, pin1, pin2, display
while True:
    val =int(330*pin2.read_analog()/1023 - 60)
    display.scroll(str(val)+'C')
    if val < 30:
        pin0.write_digital(0)
        pin1.write_digital(0)
    elif 30 <= val and val < 35:
        pin0.write_digital(1)
        pin1.write_digital(0)
    else:
        pin0.write_digital(0)
        pin1.write_digital(1)
```

30℃未満のとき
黄色LED、赤色LEDを消灯する

30℃以上35℃未満のとき
黄色LEDを点灯、赤色LEDを消灯する

35℃以上のとき
黄色LEDを消灯、赤色LEDを点灯する

上記プログラムを実行して、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。

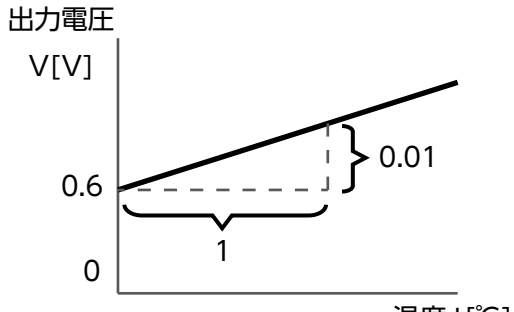


Scratch3.0に準拠しているStuduino:bitソフトウェア（ブロックプログラミング）で上記動作を確認する方法は付録Eを参照してください。

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。

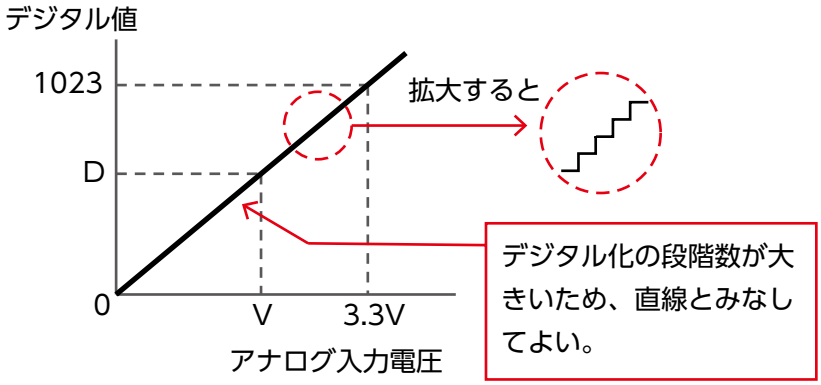
$$V=0.01t+0.6$$ ……式①



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



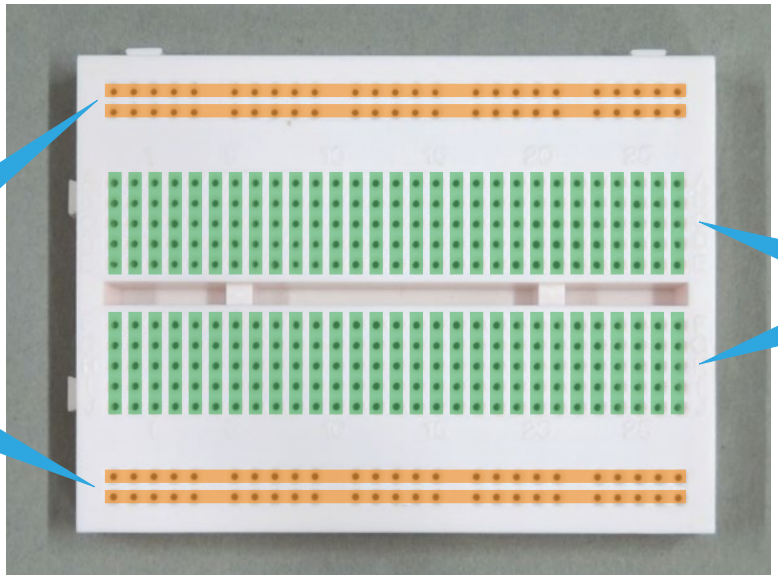
AD コンバータの特性

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. 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。

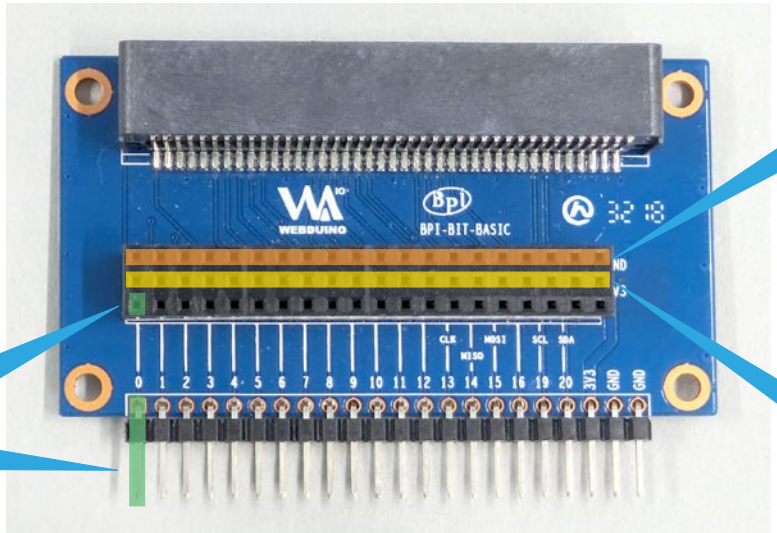


横1行ごとに全て繋がっていて、電源のプラス側やマイナス（GND）側に使用します。

縦1列5ピンごとに繋がっています。

付録 B. 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。
ジャンパワイヤを介して、回路の実装などが簡単に行えます。



横1行がGNDに繋がっています。

pin0に繋がっています。
真ん中に書かれている数字と対応します。

横1行が3.3V(3V3)に繋がっています。

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 C. デジタル入出力アナログ入力オブジェクトの機能一覧

端子モジュールはpin0/pin1/pi2/pin3です。
pin2/pin3はデジタル入出力に対応していないため、pin2.write_digital(value)やpin3.write_digital(value)を実行するとエラーになります。
デジタル入力用のread_digitalメソッドは使用できます。

メソッド	説明
write_digital(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。
read_digital()	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。
write_analog(value)	PWM 信号を端子に出力します、value 引数は 0(0%) ～ 1023(100%) です。
status()	PWM の使用状況を表示します。
read_analog()	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D.displayオブジェクトの機能一覧

メソッド	説明
get_pixel(x, y)	x 列 y 行の LED の色を返します。返り値は (R,G,B) です。
set_pixel(x, y, color)	x 列 y 行の LED の色を color で設定します。color は、(R,G,B), [R,G,B], #RGB で指定します。
clear()	すべての LED の明るさを 0 (オフ) に設定します。
show(iterable, delay=400, *, wait=True, loop=False, clear=False, color=None)	iterable(イメージ / 文字列 / 数字) を順番に表示します。
scroll(string, delay=150, *, wait=True, loop=False, color=None)	string(文字 / 数字) を水平方向にスクロールさせます。
on()	ディスプレイの電源を ON にします。
off()	ディスプレイの電源を OFF にします。(ディスプレイに関連づけられた GPIO 端子を他の目的に再利用できるようにします)。
is_on()	ディスプレイの電源が ON であれば「True」を、OFF であれば「False」を返します。

LEDの色を設定する下記のプロパティを持ちます。
BLACK (消灯)、WHITE (白)、RED (赤)、LIME (ライム)、BLUE (青)、YELLOW (黄)、CYAN (シアン)、MAGENTA (マゼンタ)、SILVER (銀)、GRAY (灰)、MAROON (茶色)、OLIVE (オリーブ)、GREEN (緑)、PURPLE (紫)、TEAL (青緑)、NAVY (紺)

付録 E. 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

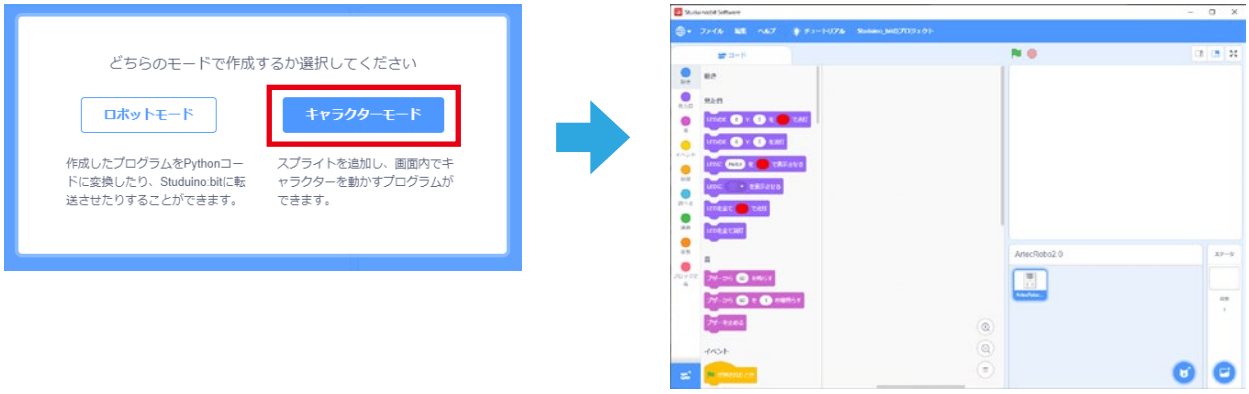
①下記URLの「インストール版ソフトウェアダウンロード」から対応OSのStuduino:bit　ソフトウェアをダウンロードしてください。

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。

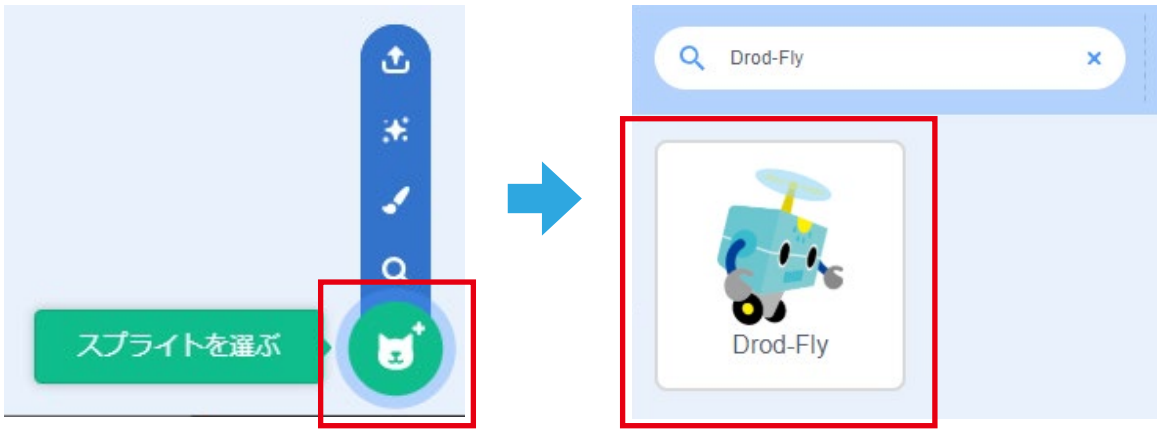
https://www.artec-kk.co.jp/artecrobo2/ja/software/

②Studuino:bitとPC接続します。

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。

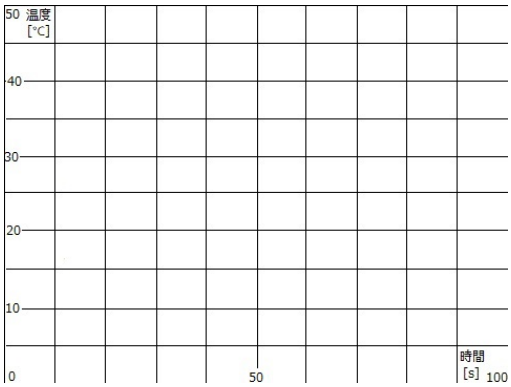
⑤使用する背景をダウンロードします。

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/

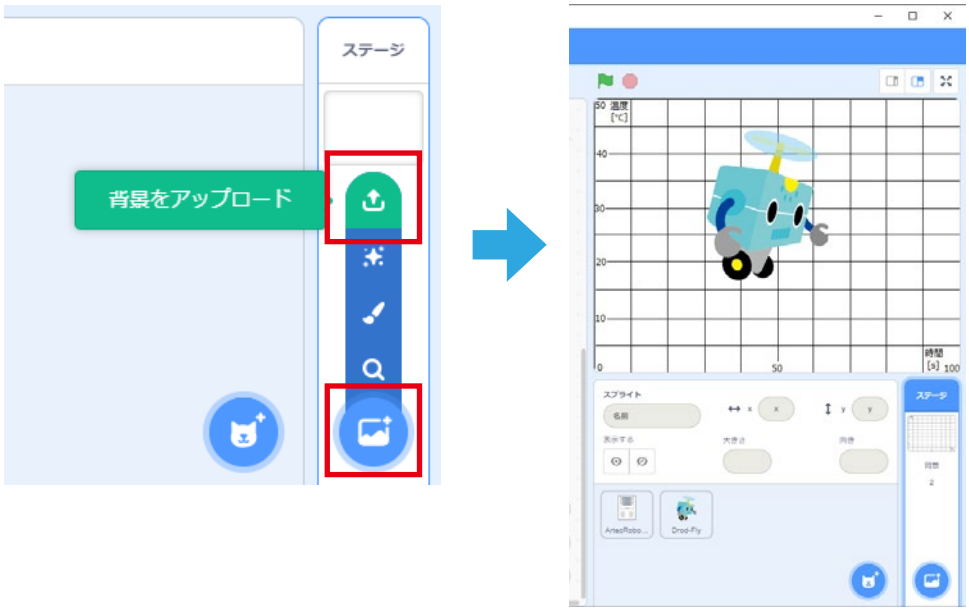
ダウンロードしたフォルダに含まれる画像ファイル「方眼紙.png」を背景に使用します。

※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。

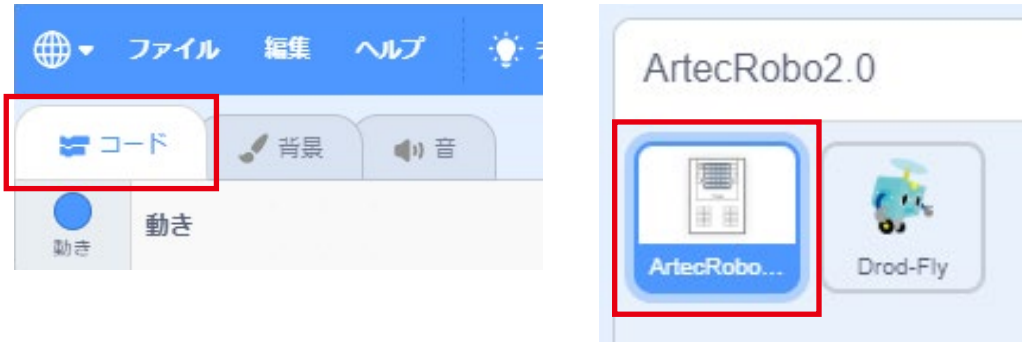


方眼紙.png

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。



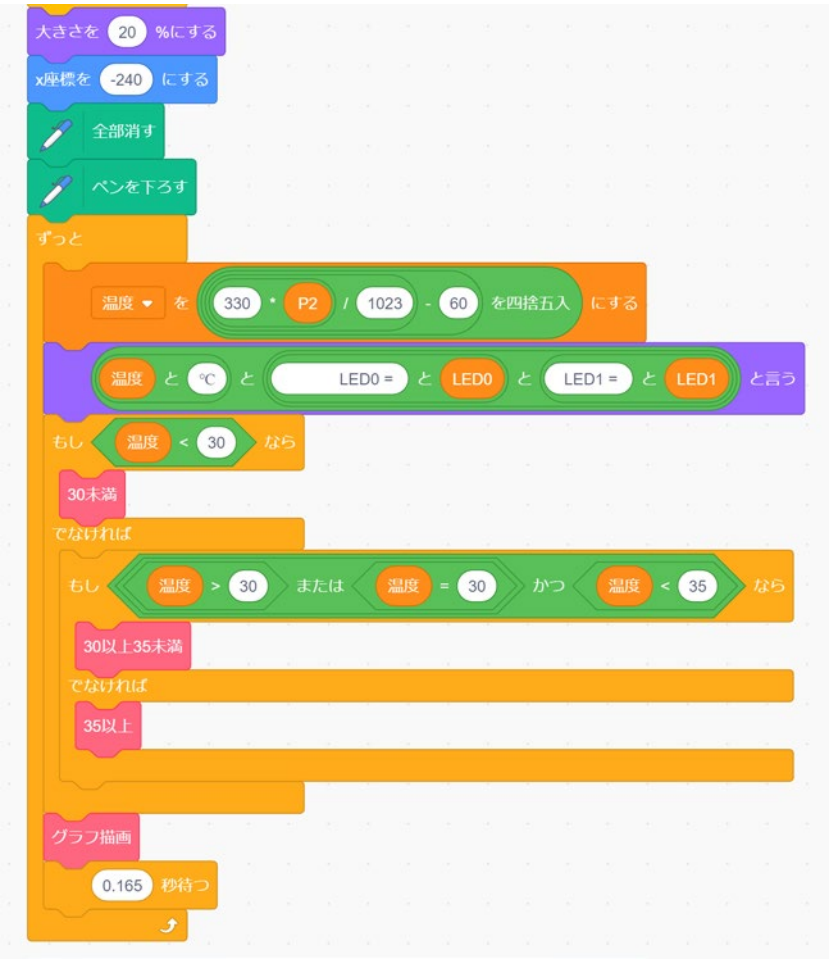
⑦左上のタブ「コード」を選択してスクリプトエリアを表示し「ArtecRobo2.0」のスプライトを選択してプログラムを作成します。



⑧「ArtecRobo2.0」のスプライトを選択した状態で、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。

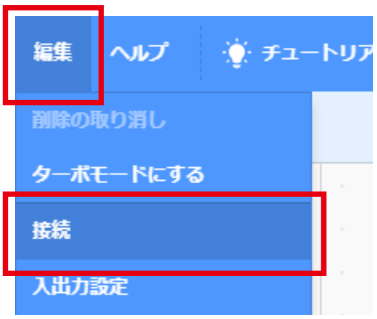
⑨「ArtecRobo2.0」（=Studuino:bit）のスプライトを選択した状態で下記のプログラムを作成します。

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。

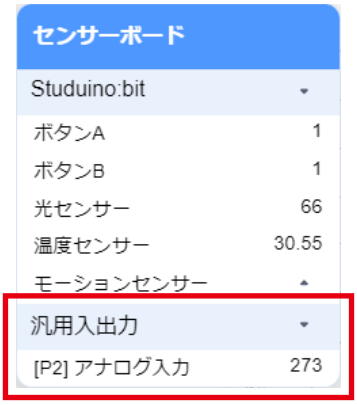


この式の導出方法はP.26を参照してください。

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。

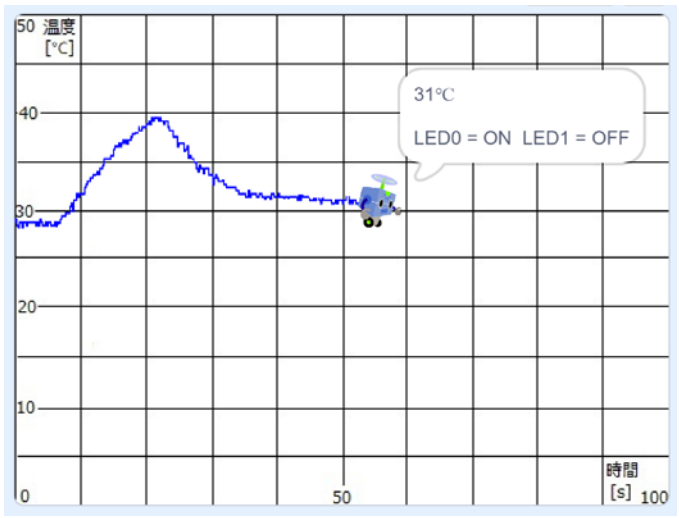


⑫Studuino:bitと接続されると、センサーボードが表示されます。
汎用入出力の[P2]アナログ入力の値が変化することを確認してください。
値が変化しない場合はブレッドボードの接続が間違っていないか確認してください。



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。

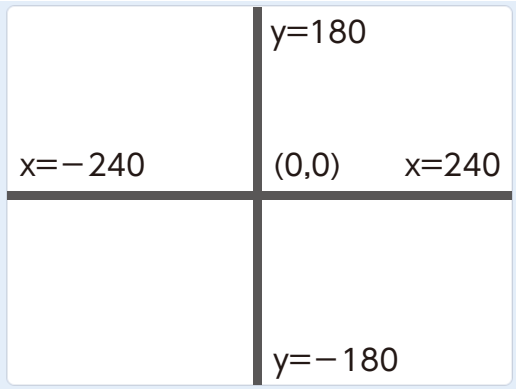
⑭プログラムを実行すると下記のように表示されます。



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。
ステージの上の「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。

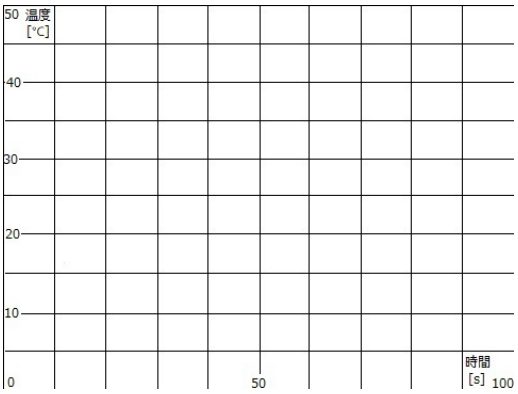
参考:スプライトのy座標とアナログ入力から取得したデジタル値の関係

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。

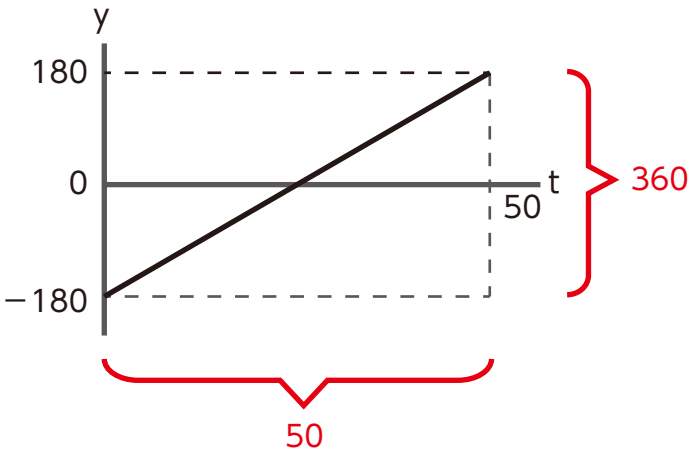


ステージ

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



$$y=\frac{360}{50}t-180$$ ……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.16の式③より、式⑤と求められます。

$$V=\frac{330D}{1023}-60$$ ……式③

$$y=\frac{2376D}{1023}-612$$ ……式⑤