ArtecRobo2.0 クラスライブラリ
リファレンス

2019/04/23　作成
2023/02/08　改訂

改訂履歴

改訂日付	改訂内容
2019/04/23	初版発行
2019/09/19	カラーセンサー・超音波センサーを追記
2023/02/08	お問い合わせ先の変更

目次

1. はじめに 1
2. ArtecRobo2.0 1
2.1. ロボット拡張ユニットの構成 1
2.2. MicroPython 2
3. 開発環境 2
4. ArtecRobo2.0 クラスライブラリ 2
4.1. DCMotor クラス 2
4.1.1. コンストラクタ 2
4.1.2. DC モーターの制御 3
4.2. Servomotor クラス 3
4.2.1. コンストラクタ 3
4.2.2. サーボモーターの制御 3
4.2.3. PWM の解放 4
4.3. Buzzer クラス 4
4.3.1. コンストラクタ 4
4.3.2. ブザーの制御 4
4.3.3. PWM の解放 4
4.4. LED クラス 5
4.4.1. コンストラクタ 5
4.4.2. LED の制御 5
4.5. IRPhotoReflector クラス 5
4.5.1. コンストラクタ 5
4.5.2. 赤外線フォトリフレクタの値の取得 5
4.6. LightSensor クラス 6
4.6.1. コンストラクタ 6
4.6.2. 光センサーの値の取得 6
4.7. SoundSensor クラス 6
4.7.1. コンストラクタ 6
4.7.2. 音センサーの値の取得 7
4.8. TouchSensor クラス 7
4.8.1. コンストラクタ 7

4.8.2. タッチセンサーの値の取得 7
4.9. Temperature クラス 8
4.9.1. コンストラクタ 8
4.9.2. 温度センサーの値の取得 8
4.10. Accelerometer クラス 8
4.10.1. コンストラクタ 8
4.10.2. 加速度センサーの値の取得 8
4.11. UltrasonicSensor クラス 9
4.11.1. コンストラクタ 9
4.11.2. 超音波センサーの値の取得 9
4.12. ColorSensor クラス 10
4.12.1. コンストラクタ 10
4.12.2. カラーセンサーの値の取得 10
5. 付録 10
5.1. 音について 10
5.2. カラーコード 11
5.3. クラス一覧 12
6. お問い合わせ先 15

1. はじめに

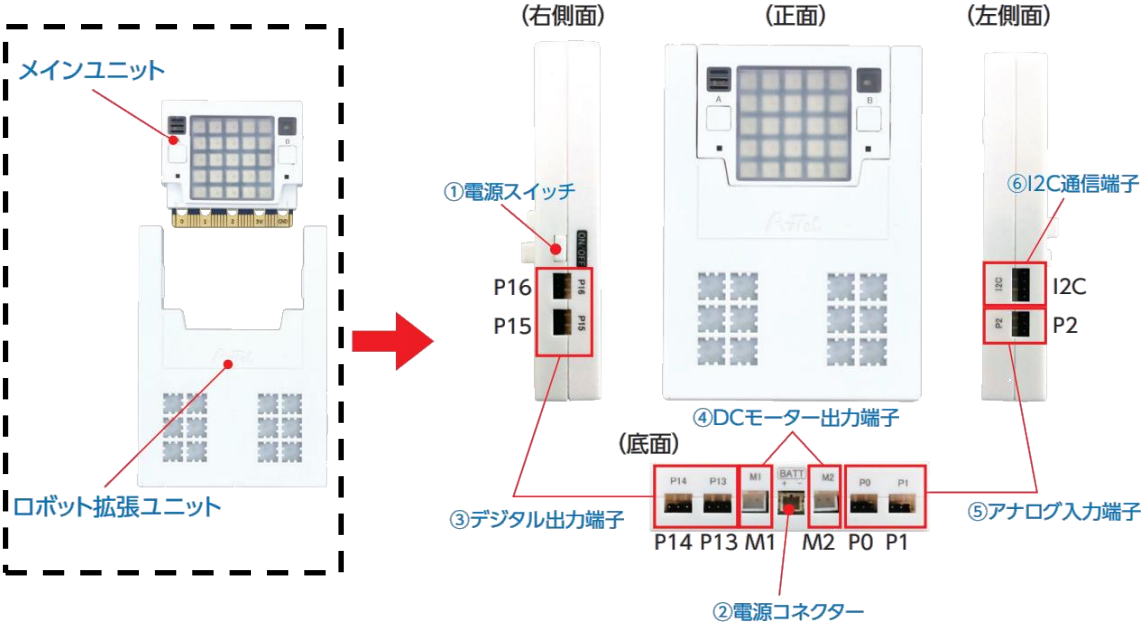
本書は、ArtecRobo2.0 を制御するクラスライブラリのリファレンスです。本書を読み進めるにあたり、Python の基本文法の理解が必要になります。

2. ArtecRobo2.0

ArtecRobo2.0 は、メインユニット Studuino:bit とロボット拡張ユニットで構成されます。使用できるプログラミング環境の一つに MicroPython があります。

2.1. ロボット拡張ユニットの構成

ロボット拡張ユニットの構成を下図に示します。



ロボット作成時に使用するモーターやセンサーなどのパーツは、ロボット拡張ユニットのデジタル出力端子、DC モーター出力端子、アナログ入力端子、I2C 通信端子のいずれかに接続します。端子とパーツの組み合わせを下記に記します。

端子	パーツ
デジタル出力端子	LED、ブザー、サーボモーター
DC モーター端子	DC モーター
アナログ入力端子	光センサー、音センサー、タッチセンサー、赤外線フォトリフレクタ
I2C 通信端子	加速度センサー

2.2. MicroPython

MicroPython は、マイコンを簡単に制御するために開発されたマイコン制御用の Python です。プログラムの構文は Python3.0 と同様ですが、メモリや CPU の性能に制限があるマイコン用に実装されており、MicroPython では使用できない Python の標準ライブラリもあります。また一方で、マイコンの GPIO を制御するライブラリなど MicroPython 特有のライブラリが標準で組み込まれています。

MicroPython は、様々なマイコンボードに移植されており、Studuino:bit ライブラリは、Studuino:bit 用に移植した MicroPython で動作します。

3. 開発環境

MicroPython の開発環境の一つとして、uPyCraft があります。uPyCraft の使用方法は別紙「Studuino:bit クラスライブラリ リファレンス」の 6.2 を参照してください。

4. ArtecRobo2.0 クラスライブラリ

ArtecRobo2.0 クラスライブラリは、下記の構成になっています。

パッケージ	モジュール	クラス	パーツ
pyatcrobo2	parts	DCMotor	DC モーター
		Servomotor	サーボモーター
		Buzzer	ブザー
		LED	LED
		LightSensor	光センサー
		SoundSensor	音センサー
		TouchSensor	タッチセンサー
		Temperature	温度センサー
		UltrasonicSensor	超音波センサー
		ColorSensor	カラーセンサー
		Accelerometer	加速度センサー

4.1. DCMotor クラス

DC モーターを制御するクラスです。

4.1.1. コンストラクタ

DC モーターを接続しているロボット拡張ユニットの端子（M1/M2）を文字列で指定して、

DC モーターを制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import DCMotor
m1 = DCMotor('M1')
```

4.1.2. DC モーターの制御

cw()メソッドで正転方向に、ccw()メソッドで逆転方向に DC モーターを回転します。stop()メソッドで DC モーターの回転を解放し、brake()メソッドで DC モーターの回転を停止します。また、power(power)メソッドで、DC モーターの回転速度を制御します。power 引数に 0～255 の整数で DC モーターの回転速度を設定します。

```
from pyatcrobo2.parts import DCMotor
import time
m1 = DCMotor('M1')
m2 = DCMotor('M2')
m1.power(100)
m2.power(100)
m1.cw()
m2.cw()
time.sleep_ms(1000)
m1.stop()
m2.stop()
```

ロボット拡張ユニットの M1、M2 端子に接続した DC モーターを速さ 100 で正転し、1 秒後に停止します。

4.2. Servomotor クラス

サーボモーターを制御するクラスです。

4.2.1. コンストラクタ

サーボモーターを接続しているロボット拡張ユニットの端子（P13/P14/P15/P16）を文字列で指定して、サーボモーターを制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import Servomotor
sv13 = Servomotor('P13')
```

4.2.2. サーボモーターの制御

set_angle(degree)メソッドで、サーボモーターの角度を設定します。

```
from pyatcrobo2.parts import Servomotor
import time
sv13 = Servomotor('P13')
sv13.set_angle(0)
time.sleep_ms(1000)
sv13.set_angle(180)
```

ロボット拡張ユニットの P13 端子に接続したサーボモーターが 0 度に移動して、1 秒後に 180 度に移動します。

4.2.3. PWM の解放

サーボモーターの角度の設定には PWM を使用しています。Studuino:bit では、同時に使用できる PWM は４つまでです。PWM を利用するパーツが 5 つ以上で、作成したサーボモーターオブジェクトを使用しない場合、release()メソッドをコールしてサーボモーターオブジェクトに割り当てられた PWM を解放してください。

4.3. Buzzer クラス

ブザーを制御するクラスです。

4.3.1. コンストラクタ

ブザーを接続しているロボット拡張ユニットの端子（P13/P14/P15/P16）を文字列で指定して、ブザーを制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import Buzzer
bz13 = Buzzer('P13')
```

4.3.2. ブザーの制御

on(sound, *, volume=None, duration=None)メソッドで、ブザーから音を出します。sound 引数に、MIDI ノートナンバーまたは音名または周波数を設定します。指定可能なノートナンバーと音名は、5.1 を参照してください。volume 引数に音量を 0～100 で設定します。duration 引数に出力する音の長さをミリ秒単位で設定します。duration 引数を設定しない場合、off()メソッドで音を止めます。

```
from pyatcrobo2.parts import Buzzer
bz13 = Buzzer('P13')
bz13.on('C4', duration=1000)
```

ロボット拡張ユニットの P13 端子に接続したブザーから'C4'の音が 1 秒間出ます。

4.3.3. PWM の解放

音の出力には PWM を使用しています。Studuino:bit では、同時に使用できる PWM は４つ

までです。PWM を利用するパーツが 5 つ以上で、作成したブザーオブジェクトを使用しない場合、release()メソッドをコールしてブザーオブジェクトに割り当てられた PWM を解放してください。

4.4. LED クラス

LED を制御するクラスです。

4.4.1. コンストラクタ

LED を接続しているロボット拡張ユニットの端子（P13/P14/P15/P16）を文字列で指定して、LED を制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import LED
led13 = LED('P13')
```

4.4.2. LED の制御

on()メソッドで、LED を点灯、off()メソッド LED を消灯します。

```
from pyatcrobo2.parts import LED
import time
led13 = LED('P13')
while True:
    led13.on()
    time.sleep_ms(500)
    led13.off()
    time.sleep_ms(500)
```

ロボット拡張ユニットの P13 端子に接続した LED が、500 ミリ秒間隔で点灯、消灯を繰り返します。

4.5. IRPhotoReflector クラス

赤外線フォトリフレクタを制御するクラスです。

4.5.1. コンストラクタ

赤外線フォトリフレクタを接続しているロボット拡張ユニットの端子（P0/P1/P2）を文字列で指定して、赤外線フォトリフレクタを制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import IRPhotoReflector
sensor0 = IRPhotoReflector('P0')
```

4.5.2. 赤外線フォトリフレクタの値の取得

get_value()メソッドで、赤外線フォトリフレクタの値をアナログ値で取得します。センサー

の値は 0 から 4095 の整数です。

```
from pyatcrobo2.parts import IRPhotoReflector
import time
sensor0= IRPhotoReflector ('P0')
while True:
    print(sensor0.get_value())
    time.sleep_ms(500)
```

ロボット拡張ユニットの P0 端子に接続した赤外線フォトリフレクタの値の値が 500 ミリ秒ごとに標準出力に表示されます。

4.6. LightSensor クラス

光センサーを制御するクラスです。

4.6.1. コンストラクタ

光センサーを接続しているロボット拡張ユニットの端子（P0/P1/P2）を文字列で指定して、光センサーを制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import LightSensor
sensor0 = LightSensor('P0')
```

4.6.2. 光センサーの値の取得

get_value()メソッドで、光センサーの値をアナログ値で取得します。センサーの値は 0 から 4095 の整数です。

```
from pyatcrobo2.parts import LightSensor
import time
sensor0 = LightSensor('P0')
while True:
    print(sensor0.get_value())
    time.sleep_ms(500)
```

ロボット拡張ユニットの P0 端子に接続した光センサーの値が 500 ミリ秒ごとに標準出力に表示されます。

4.7. SoundSensor クラス

音センサーを制御するクラスです。

4.7.1. コンストラクタ

音センサーを接続しているロボット拡張ユニットの端子（P0/P1/P2）を文字列で指定して、音センサーを制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import SoundSensor
sensor0 = SoundSensor('P0')
```

4.7.2. 音センサーの値の取得

get_value()メソッドで、音センサーの値をアナログ値で取得します。センサーの値は 0 から 4095 の整数です。

```
from pyatcrobo2.parts import SoundSensor
import time
sensor0 = SoundSensor ('P0')
while True:
    print(sensor0.get_value())
    time.sleep_ms(500)
```

ロボット拡張ユニットの P0 端子に接続した音センサーの値が 500 ミリ秒ごとに標準出力に表示されます。

4.8. TouchSensor クラス

タッチセンサーを制御するクラスです。

4.8.1. コンストラクタ

タッチセンサーを接続しているロボット拡張ユニットの端子（P0/P1/P2）を文字列で指定して、タッチセンサーを制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import TouchSensor
sensor0 = TouchSensor('P0')
```

4.8.2. タッチセンサーの値の取得

get_value()メソッドで、タッチセンサーの値をデジタル値で取得します。センサーの値は 0 または 1 です。また、is_pressed()メソッドでタッチセンサーが押されているかどうかを True/False で取得します。

```
from pyatcrobo2.parts import TouchSensor
import time
sensor0 = TouchSensor ('P0')
while True:
    print(sensor0.get_value())
    print(sensor0.is_pressed())
    time.sleep_ms(500)
```

ロボット拡張ユニットの P0 端子に接続したタッチセンサーが押されているかどうかを 500 ミリ秒ごとに標準出力に表示します。タッチセンサーが押されている場合は、0 と True が

表示されます。押されていない場合は、1 と False が表示されます。

4.9. Temperature クラス

温度センサーを制御するクラスです。

4.9.1. コンストラクタ

温度センサーを接続しているロボット拡張ユニットの端子（P0/P1/P2）を文字列で指定して、温度センサーを制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import Temperature
sensor0 = Temperature('P0')
```

4.9.2. 温度センサーの値の取得

get_value()メソッドで、温度センサーの値をアナログ値で取得します。センサーの値は 0 から 4095 の整数です。また、get_celsius()メソッドで、温度センサーの値を温度で取得します。

```
from pyatcrobo2.parts import Temperature
import time
sensor0 = Temperature('P0')
while True:
    print(sensor0.get_value())
    print(sensor0.get_celsius ())
    time.sleep_ms(500)
```

ロボット拡張ユニットの P0 端子に接続した温度センサーのアナログ値と温度が 500 ミリ秒ごとに標準出力に表示されます。

4.10. Accelerometer クラス

加速度センサーを制御するクラスです。

4.10.1. コンストラクタ

加速度センサーを接続しているロボット拡張ユニットの端子（I2C）を文字列で指定して、加速度センサーを制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import Accelerometer
sensor0 = Accelerometer('I2C')
```

フルスケール（最大測定限界）は、重力加速度 2G で、単位は、G です。

4.10.2. 加速度センサーの値の取得

get_values()メソッドで、加速度センサーの値を取得します。センサーの値はタプル(x, y, z)

で取得します。また、get_x()メソッド、get_y()メソッド、get_z()メソッドで x 軸方向、y 軸方向、z 軸方向の加速度を別々に取得することができます。

```
from pyatcrobo2.parts import Accelerometer
import time
sensor0 = Accelerometer ('I2C')
while True:
    print(sensor0.get_values())
    time.sleep_ms(500)
```

ロボット拡張ユニットの I2C 端子に接続した加速度センサーの値が 500 ミリ秒ごとに標準出力に表示されます。

4.11. UltrasonicSensor クラス

超音波センサーを制御するクラスです。

4.11.1. コンストラクタ

超音波センサーを接続しているロボット拡張ユニットの端子（P0/P1）を文字列で指定して、超音波センサーを制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import UltrasonicSensor
sensor0 = UltrasonicSensor ('P0')
```

4.11.2. 超音波センサーの値の取得

get_distance()メソッドで、計測対象物との距離を取得します。距離は float 型で単位は cm です。また、get_pulse_time()メソッドで、超音波を発射してから受信するまでの時間をマイクロ秒単位で取得します。距離が計測できない場合、RuntimeErrorre 例外を発生します。

```
from pyatcrobo2.parts import UltrasonicSensor
import time
sensor0 = UltrasonicSensor('P0')
while True:
    try:
        print(sensor0.get_distance())
    except RuntimeError as e:
        print(e)

    time.sleep_ms(100)
```

ロボット拡張ユニットの P0 端子に接続した超音波センサーの値が 100 ミリ秒ごとに標準出力に表示されます。

4.12. ColorSensor クラス

カラーセンサーを制御するクラスです。

4.12.1. コンストラクタ

カラーセンサーを接続しているロボット拡張ユニットの端子（I2C）を文字列で指定して、カラーセンサーを制御するオブジェクトを作成します。

```
from pyatcrobo2.parts import ColorSensor
sensor0 = ColorSensor ('I2C')
```

4.12.2. カラーセンサーの値の取得

get_values()メソッドで、計測対象物の色情報を取得します。色情報はタプル型で（赤成分、緑成分、青成分、輝度）で各成分の度合いになります。また、get_colorcode()メソッドで、対象物の色を取得します。

```
from pyatcrobo2.parts import ColorSensor
import time
color = ColorSensor('I2C')
while True:
    try:
        print(color.get_values())
        print(color.get_colorcode())
    except RuntimeError as e:
        print(e)

    time.sleep_ms(100)
```

ロボット拡張ユニットの I2C 端子に接続したカラーセンサーから、対象物の色情報とカラーコードが 100 ミリ秒ごとに標準出力に表示されます。

5. 付録

5.1. 音について

Buzzer クラスで使用できる MIDI ノートナンバーと音名です。

MIDI	音名	MIDI	音名	MIDI	音名	MIDI	音名	MIDI	音名	MIDI	音名	MIDI	音名
48	C3	60	C4	72	C5	84	C6	96	C7	108	C8	120	C9
49	CS3	61	CS4	73	CS5	85	CS6	97	CS7	109	CS8	121	CS9
50	D3	62	D4	74	D5	86	D6	98	D7	110	D8	122	D9
51	DS3	63	DS4	75	DS5	87	DS6	99	DS7	111	DS8	123	DS9
52	E3	64	E4	76	E5	88	E6	100	E7	112	E8	124	E9
53	F3	65	F4	77	F5	89	F6	101	F7	113	F8	125	F9
54	FS3	66	FS4	78	FS5	90	FS6	102	FS7	114	FS8	126	FS9

55	G3	67	G4	79	G5	91	G6	103	G7	115	G8	127	G9
56	GS3	68	GS4	80	GS5	92	GS6	104	GS7	116	GS8		
57	A3	69	A4	81	A5	93	A6	105	A7	117	A8		
58	AS3	70	AS4	82	AS5	94	AS6	106	AS7	118	AS8		
59	B3	71	B4	83	B5	95	B6	107	B7	119	B8		

5.2. カラーコード

定数	色	値
COLOR_RED	赤	1
COLOR_GREEN	緑	2
COLOR_BLUE	青	3
COLOR_WHITE	白	4
COLOR_YELLOW	黄	5
COLOR_ORANGE	オレンジ	6
COLOR_PURPLE	紫	7
COLOR_UNDEF	不明	0

5.3. クラス一覧

ArtecRobo2 クラスライブラリの一覧.を下記に記します。

機能	クラス/モジュール	メソッド/属性	説明
DC モーター	DCMotor	__init__(pin)	pin 引数に M1, M2 ピンを文字列で指定して DC モーターのインスタンスを生成します。 dcm = DCMotor('M1')
		cw()	DC モーターを正転方向に回転します。
		ccw()	DC モーターを逆転方向に回転します。
		stop()	DC モーターの回転を解放します。
		brake()	DC モーターの回転を停止します。
		power(power)	power 引数にパワー(0～255)を指定して DC モーターの回転の速さを制御します。
サーボモーター	Servomotor	__init__(pin)	pin 引数に P13, P14, P15, P16 ピンを文字列で指定してサーボモーターのインスタンスを生成します。 sv = ServoMotor('P13')
		set_angle(degree)	degree 引数に角度(0～180)を指定してサーボモーターの角度を制御します。 sv.set_angle(90)
		release()	サーボモーターオブジェクトに割り当てられた PWM を解放します。
ブザー	Buzzer	__init__(pin)	pin 引数に P13, P14, P15, P16 ピンを文字列で指定してブザーのインスタンスを生成します。 bzr = Buzzer('P13')
		on(sound, *, volume=-1, duration=-1)	sound 引数に音名('C3'～'G9'の文字列)、または MIDI ノートナンバー('48'～'127'の文字列)、または周波数(整数)を、volume 引数に出力する音の大きさ(1～100)、duration 引数に音の出力時間(ms)を指定して、ブザーを制御します。duration 引数を指定しない場合は、off メソッドがコールされるまで音を出力します。volume が指定されない場合は、音の高さに合わせた大きさになります。 bzr.on('50', duration=1000)　　# MIDI ノートナンバーの 50 番の高さの音が 1 秒間出力されます。 bzr.on('C4', volume=1, duration=1000)　　# 音名の C4 の高さの音が音量 1 で 1 秒間出力されます。 bzr.on(440)
		off()	ブザーから出力している音を停止します。 bzr.off()
		release()	ブザーオブジェクトに割り当てられた PWM を解放します。
LED	LED	__init__(pin)	pin 引数に P13, P14, P15, P16 ピンを指定して LED のインスタンスを生成します。 led = LED('P13')
		on()	LED を点灯します。 led.on()
		off()	LED を消灯します。 led.on()

赤外線 フォトリフレクタ	IRPhotoReflector	__init__(pin)	pin 引数に P0, P1, P2 ピンを文字列で指定して赤外線フォトリフレクタのインスタンスを生成します。 irp = IRPhotoReflector('P0')
		get_value()	センサーの値(0～4095)を取得します。 irp.get_value()
光センサー	LightSensor	__init__(pin)	pin 引数に P0, P1, P2 ピンを文字列で指定して光センサーのインスタンスを生成します。 ls = LightSensor('P0')
		get_value()	センサーの値(0～4095)を取得します。 ls.get_value()
温度センサー	Temperature	__init__(pin)	pin 引数に P0, P1, P2 ピンを文字列で指定して温度センサーのインスタンスを生成します。 ts = Temperature('P0')
		get_value()	センサーの値(0～4095)を取得します。 ts.get_value()
		get_celsius()	センサーの値を温度（セ氏）で取得します。 ts.get_celsius()
音センサー	SoundSensor	__init__(pin)	pin 引数に P0, P1, P2 ピンを文字列で指定して音センサーのインスタンスを生成します。 ss = SoundSensor('P0')
		get_value()	センサーの値(0～4095)を取得します。 ss.get_value()
タッチセンサー	TouchSensor	__init__(pin)	pin 引数に P0, P1, P2 ピンを文字列で指定してタッチセンサーのインスタンスを生成します。 ss = TouchSensor('P0')
		get_value()	センサーの値(0 or 1)を取得します。 ss.get_value()
		is_pressed()	センサーの状態を取得します。 タッチセンサーが押されている場合は True を返します。
超音波センサー	UltrasonicSensor	get_pulse_time ()	超音波が対象物に当たって返る時間をマイクロ秒で返します。
		get_distance()	対象物との距離を浮動小数点で返します。単位は、cm です。
カラーセンサー	ColorSensor	get_values	対象物の色情報をタプルで（赤成分、緑成分、青成分、輝度）返します。各値は度合いになります。
		get_colorcode	対象物の色をカラーコードで返します。5.2 参照
加速度センサー	Accelerometer	__init__(pin)	pin 引数に I2C ピンを文字列で指定して加速度センサーのインスタンスを生成します。 acc = Accelerometer('I2C')

		get_values()	x, y, z 軸の加速度を計測します。最大計測範囲は、±2G で単位は G です。 acc.get_values()
		get_x	= get_values()[0]
		get_y	= get_values()[1]
		get_z	= get_values()[2]

6. お問い合わせ先

株式会社 アーテック お客様相談窓口

お電話によるお問い合わせ 072-990-5656

お問い合わせフォーム https://www.artec-kk.co.jp/contact/