



Rechercher des projets

[Aide](#)[Sponsors](#)[Se connecter](#)[S'inscrire](#)

piServoCtl 1.1.0



Dernière version

`pip install piServoCtl` Dernière version : 20
oct. 2020

This is a servo motor control library for Raspberry Pi.

Navigation

[Description
du projet](#) [Historique
des versions](#) [Téléchargement
des fichiers](#)

Liens du projet

[Homepage](#)

Statistiques

Consultez les
statistiques pour ce
projet via [Libraries.io](#),

Description du projet

piServoCtl

Overview

This is a servo motor control library for Raspberry Pi. In addition to normal servo motors, continuous rotation servo motors can also be controlled. It utilizes hardware PWM. Therefore, the servo motor can be moved smoothly. Since you are using pigpio, you need to install pigpio and start pigpiod. To start pigpiod, you need to press the `sudo pigpiod` command or run pigpiod automatically.

In addition, you can easily make an RC car by using two rotation servo motors. Use the Drive class to set the GPIO pin and servo motor. You can easily implement the steering function by using the steering method.


Installation


```
sudo pip3 install piServoCtl
```

ou bien en utilisant
[notre jeu de données public sur Google BigQuery](#)

Métadonnées

Licence : MIT License
 (MIT License)

Créé par : [naoto64](#) 

 Servo, Motor, PWM,
 Control, RaspberryPi

Maintenu par



Classifieurs

Development Status

- 5 - [Production/Stable](#)

License

- [OSI Approved :: MIT License](#)

Programming Language

- [Python](#)
- [Python :: 3](#)
- [Python :: 3.4](#)
- [Python :: 3.5](#)
- [Python :: 3.6](#)
- [Python :: 3.7](#)

Topic

- [Scientific/Engineering](#)
- [Software Development :: Libraries :: Python Modules](#)

Usage

First, start pigpiod (if it is not started). Then import the module. The module name is "piservo". Please note that it is different from the name when it was installed.

```
sudo pigpiod
```

Servo

The easiest way is to use `Servo(gpio)` (where `gpio` is the GPIO pin number). Use GPIO that supports hardware PWM.

You can control the servo motor with the `write` method.

Drive

The easiest way is to use `Drive(left_gpio, right_gpio)` (`left_gpio`, `right_gpio` enter the GPIO pin number). Use GPIO that supports hardware PWM. You can swap the `left_gpio` and `right_gpio` to flip the front and back of the car.

You can control the car with the `steering` method.

Demo

Servo

```
from piservo import Servo
import time

myservo = Servo(12)

myservo.write(180)
time.sleep(3)
myservo.write(0)
time.sleep(3)
myservo.stop()
```

Drive

```
from piservo import Drive
import time

mycar = Drive(12, 13)

mycar.steering(50, 45)
time.sleep(1)
mycar.steering(-50, 0)
time.sleep(1)
mycar.stop()
```

Method

Servo

```
Servo(gpio, min_value=0, max_value=180, min_pulse=0.5, frequency=50)
```

gpio: The gpio pin number to which the servo motor is connected.

min_value: Minimum angle of servo motor (speed if it is a rotation servo motor).

max_value: Maximum angle of servo motor (speed if it is a rotation servo motor).

min_pulse: Minimum control pulse width of servo motor (millisecond).

max_pulse: Maximum control pulse width of servo motor (millisecond).

frequency: PWM frequency of the servo motor.

Create an instance.

```
Servo.write(value)
```

value: Servo motor drive angle (speed if it is a rotation servo motor).

Drives the servo motor.

```
Servo.read()
```

Read the current value of the servo motor.

```
Servo.start()
```

Starts control of the servo motor.

```
Servo.stop()
```

Stops control of the servo motor.

Drive

```
Drive(left_gpio, right_gpio, min_value=-100, max_value=100,
```

left_gpio: GPIO pin number to which the left servo motor is connected.

right_gpio: GPIO pin number to which the right servo motor is connected.

min_value: Minimum speed of servo motor.

max_value: Maximum speed of servo motor.

min_pulse: Minimum control pulse width of servo motor (millisecond).

max_pulse: Maximum control pulse width of servo motor (millisecond).

frequency: PWM frequency of the servo motor.

Create an instance.

```
Drive.steering(speed=50, direction=0)
```

speed: Speed of movement.

direction: Direction of movement.

It moves in the specified direction at the specified speed.

```
Drive.stop()
```

Stop moving.

```
Drive.start()
```

Start control.

```
Drive.set_speed(speed)
```

speed: Speed of movement.

Moves in the previously specified direction and at the specified speed.

```
Drive.set_direction(direction)
```

direction: Direction of movement.

Moves in the specified direction at the previously specified speed.

```
Drive.get_speed()
```

Get the current speed.

```
Drive.get_direction()
```

Gets the current direction.

License

MIT

Détails du projet

Liens du projet

 [Homepage](#)


Statistiques

Consultez les statistiques pour ce projet via [Libraries.io](https://libraries.io), ou bien en utilisant [notre jeu de données public](#) sur Google BigQuery

Métadonnées

Licence : MIT License (MIT License)

Créé par : [naoto64](#) 

 Servo, Motor, PWM, Control, RaspberryPi

Maintenu par



Classifieurs

Development Status

- [5 - Production/Stable](#)

License

- [OSI Approved :: MIT License](#)

Programming Language

- [Python](#)
- [Python :: 3](#)
- [Python :: 3.4](#)
- [Python :: 3.5](#)
- [Python :: 3.6](#)
- [Python :: 3.7](#)

Topic

- [Scientific/Engineering](#)
- [Software Development :: Libraries :: Python Modules](#)

Historique des versions

[Notifications de version](#) | [Flux RSS](#) 

CETTE VERSION



1.1.0
20 oct. 2020

**1.0.0**

18 oct. 2020

Téléchargement des fichiers

Téléchargez le fichier pour votre plateforme. Si vous n'êtes pas sûr de savoir lequel choisir, apprenez-en plus sur l'[installation de paquets](#).

Distribution des sources

! Aucun fichier de distribution des sources n'est disponible pour cette version. Voir tutoriel [Générer une archive de distribution](#).

Distributions compilées

[piServoCtl-1.1.0-py3-none-any.whl](#) (4.6 kB [afficher les empreintes](#))

Uploaded 20 oct. 2020 **py3**

[piServoCtl-1.1.0-py2-none-any.whl](#) (4.6 kB [afficher les empreintes](#))

Uploaded 20 oct. 2020 **py2**

Aide

Installation de paquets
Publier de paquets
Guide d'utilisation
Nom du projet réservé
FAQ

À propos de PyPI

PyPI sur Twitter
Tableau de bord de l'infrastructure
Statistiques
Logos et marques
Nos sponsors

Contribuer à PyPI

Utilisation de PyPI

Bugs et commentaires	Code de conduite
Contribuer sur GitHub	Signaler un problème de sécurité
Traduire PyPI	Politique de confidentialité
Sponsor pypi	Conditions d'utilisation
Liste des développeurs et développeuses	Politique d'utilisation acceptable

Statut : Tous les systèmes sont opérationnels

Développé et maintenu par la communauté Python, pour la communauté Python.
Faites un don aujourd'hui !

"PyPI", "Python Package Index", and the Blocks logos are registered Trademarks of the Python Software Foundation .

© 2023 Python Software Foundation
[Plan du site](#)

[English](#) [español](#) [français](#) [日本語](#) [português \(Brasil\)](#) [українська](#) [Ελληνικά](#) [Deutsch](#) [中文 \(简体\)](#)
[中文 \(繁體\)](#) [русский](#) [עברית](#) [esperanto](#)