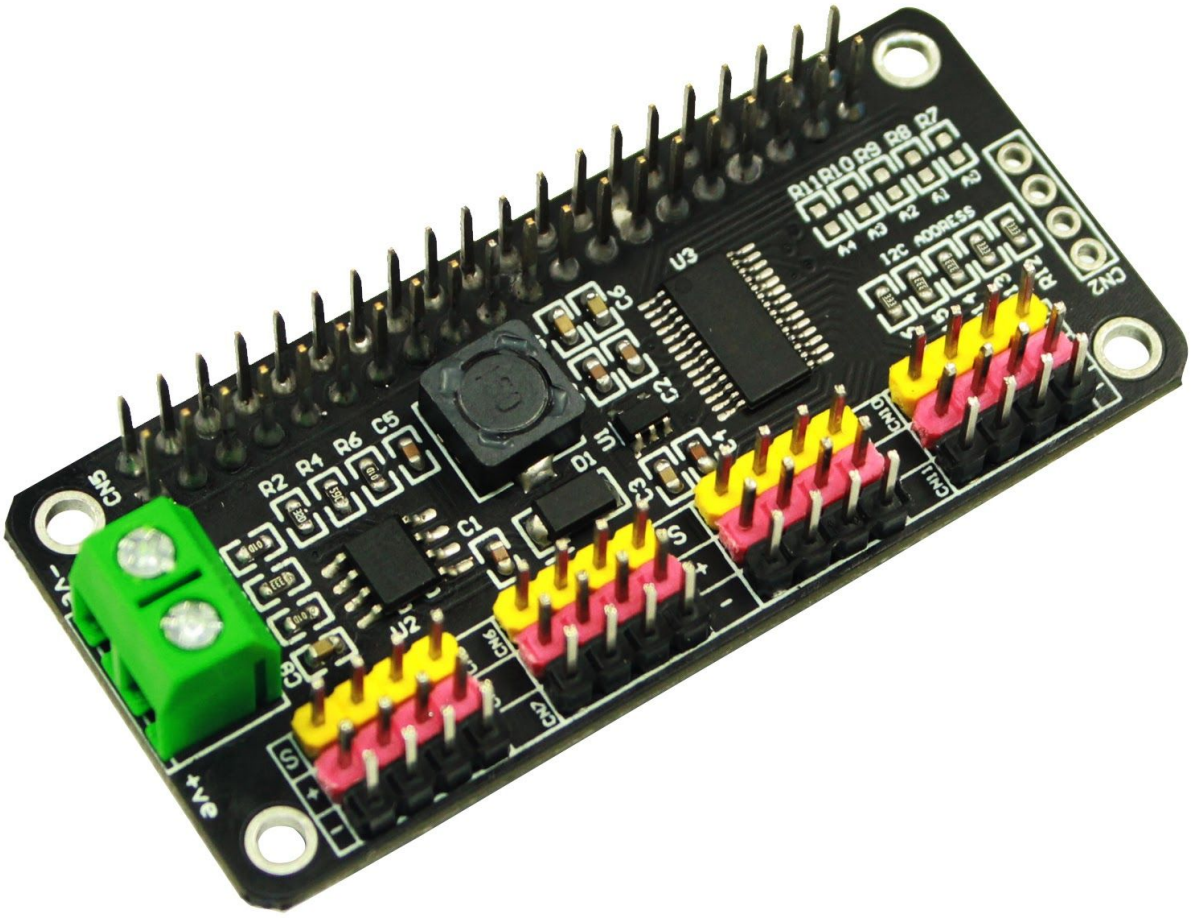
# SERVO DRIVER - HAT
## for Raspberry Pi

## User Manual



# CONTENT

# INTRODUCTION

The servo driver HAT is a PWM based expansion board designed for Raspberry Pi. The PCA9685 chip expands up to 16 channels and supports 12-bits resolution for each channel. It uses the I2C protocol for communication purpose.
The PCA9685 that provides 16 more pins that can be used as PWM output

## What's on the Board:

**Voltage Regulator** - It regulates the onboard supply to the desired voltage.

- The board has a 5V regulator having 3A output current that can be powered from the battery through VIN terminal.
- There is one more regulator that converts the 5V to 3.3V

**PCA9685** - It is a PWM driver that works on the I2C protocol. It provides the 16 channel for controlling purpose.

**Headers and Connectors** - to connect the power supply, Pi and also the motors connectors and headers are provided.

## Specifications:

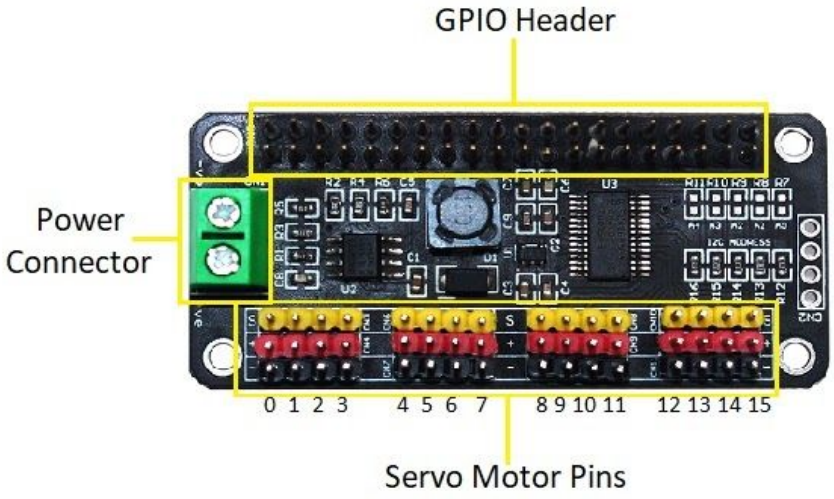
- Input voltage - 6V to 12V
- Servo driving voltage - 5V
- Logic voltage - 3.3 V
- Communication Protocol - I2C
- Dimensions - 65mm x 30mm

## Applications:

- Robotics

# HARDWARE DESCRIPTION

Servo driver is an easily stackable Raspberry Pi Hat with dimensions of 65mm x 30mm. The servo driver gives access to all the GPIO pins as well as the ports to connect the servo motors.




**GPIO Header -** To give the user access to all the GPIO pins of the Raspberry Pi a 40 pin header is mounted on it.

Two 5V and two 3V3 are there along with the ground(0V) pins. Leftover is the general-purpose pins. Some of the GPIO pins have special functions mentioned below:
PWM -

- Software PWM available on all pins
- Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19

For serial communications -
SPI:

- SPI0: MOSI (GPIO10); MISO (GPIO9); SCLK (GPIO11); CE0 (GPIO8), CE1 (GPIO7)

- SPI1: MOSI (GPIO20); MISO (GPIO19); SCLK (GPIO21); CE0 (GPIO18); CE1 (GPIO17); CE2 (GPIO16)

I2C:

- Data: (GPIO2); Clock (GPIO3)
- EEPROM Data: (GPIO0); EEPROM Clock (GPIO1)

UART:

- TX (GPIO14); RX (GPIO15)

**Servo Pins -** 3 pin header with a group of 4 is mounted on the board. 16 channels are provided to connect 16 servo motors starting from channel 0 to channel 15 as shown in the figure.

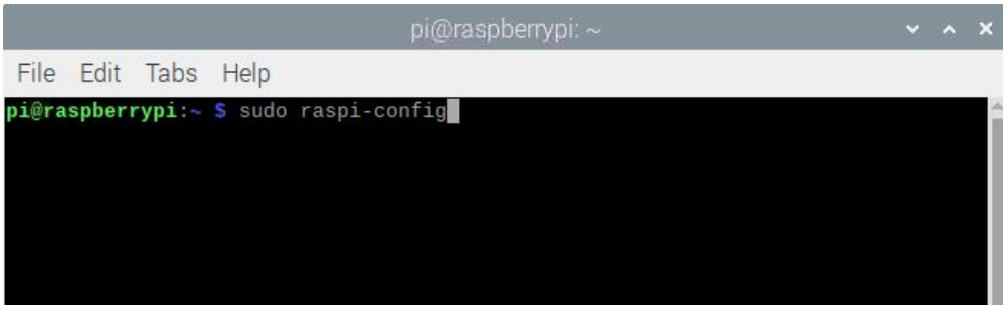
**Power Connector -** Input power supply connector is provided to give supply to the HAT as well as the Raspberry Pi board. No external power supply is needed to power up the Pi. Input voltage required is 6V to 12V.

# Enabling I2C:

There are two ways by which you can enable I2C in your device.

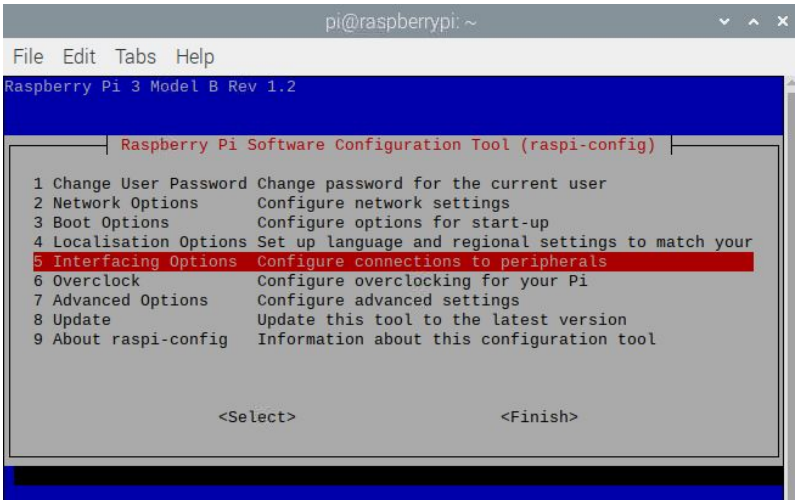
**First method - Using Pi's terminal**

➢ On the terminal write " sudo raspi-config ".

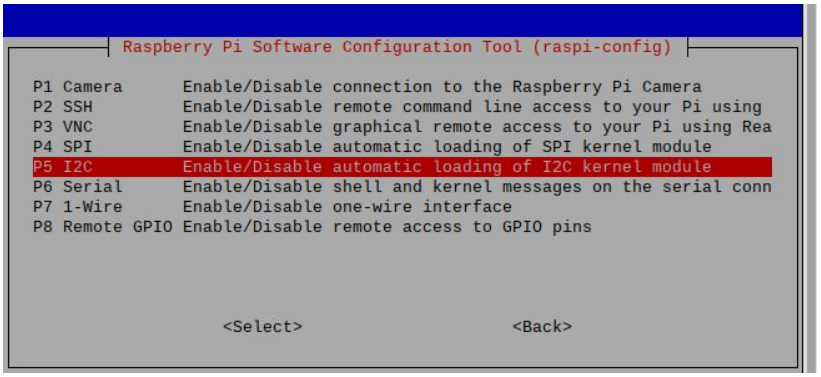



A popup window will open with configuration settings.

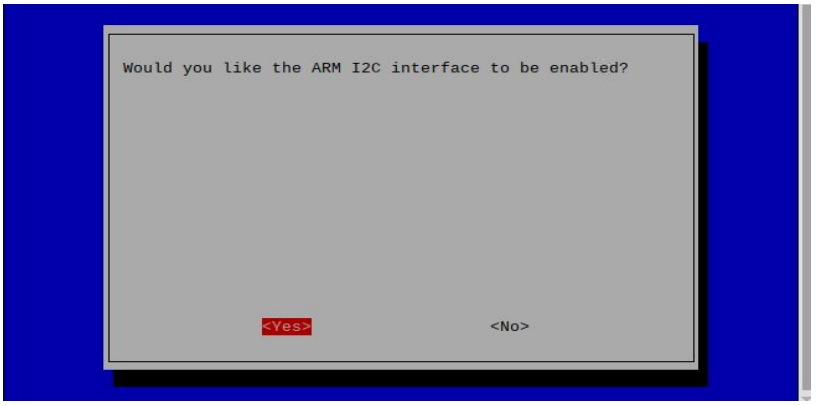
➢ Go to *"Interfacing options"*.




➢ Select "I2C" to enable it.



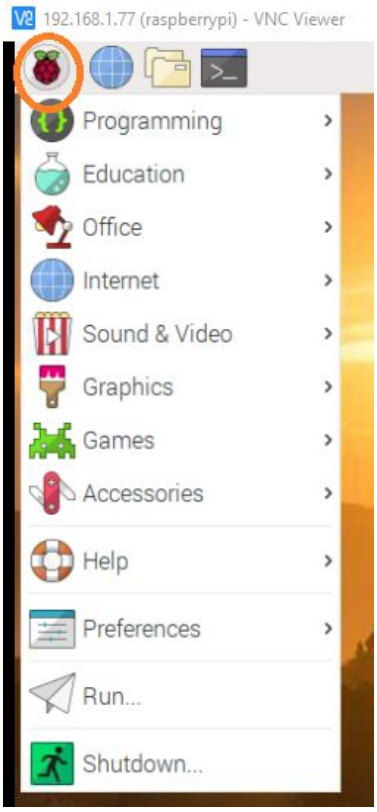
➢ A confirmation message will pop up. Select “Yes” to enable it.



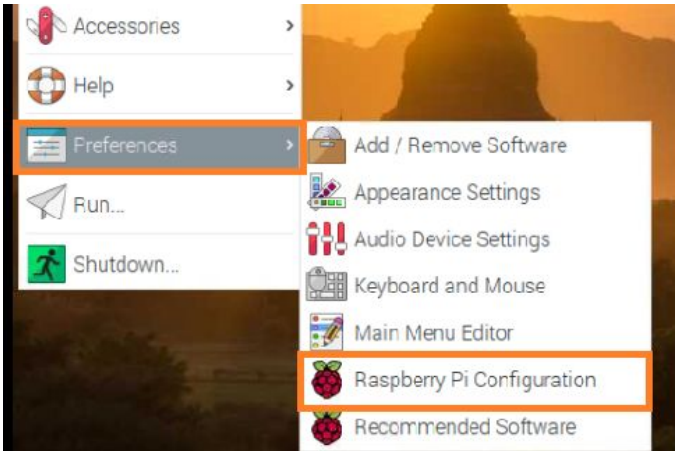

➢ I2C is now enabled on the device.

**Second method - Using GUI**

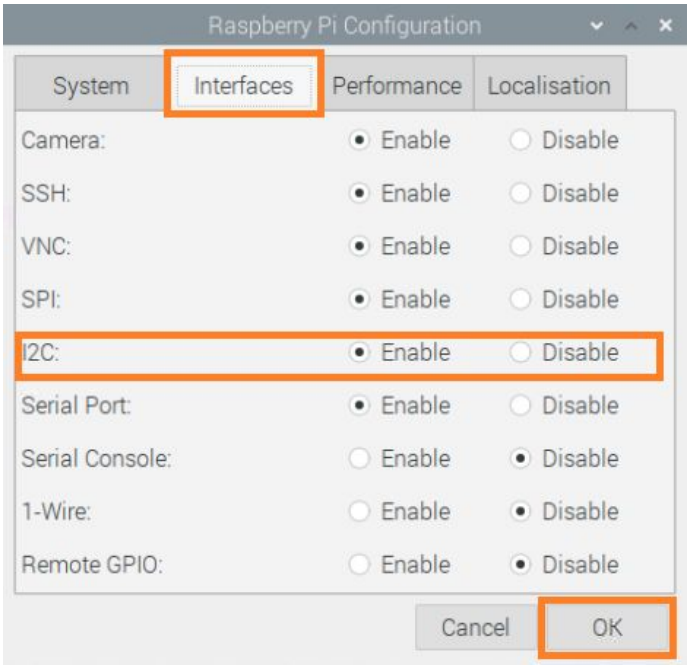
➢ Go to the Raspberry logo on the top left corner of the screen. Select Preferences from the dropdown menu.




➢ Select Raspberry Pi configuration.



➢ Go to the “Interface” and enable the I2C from there.



# HOW TO USE

## Stacking

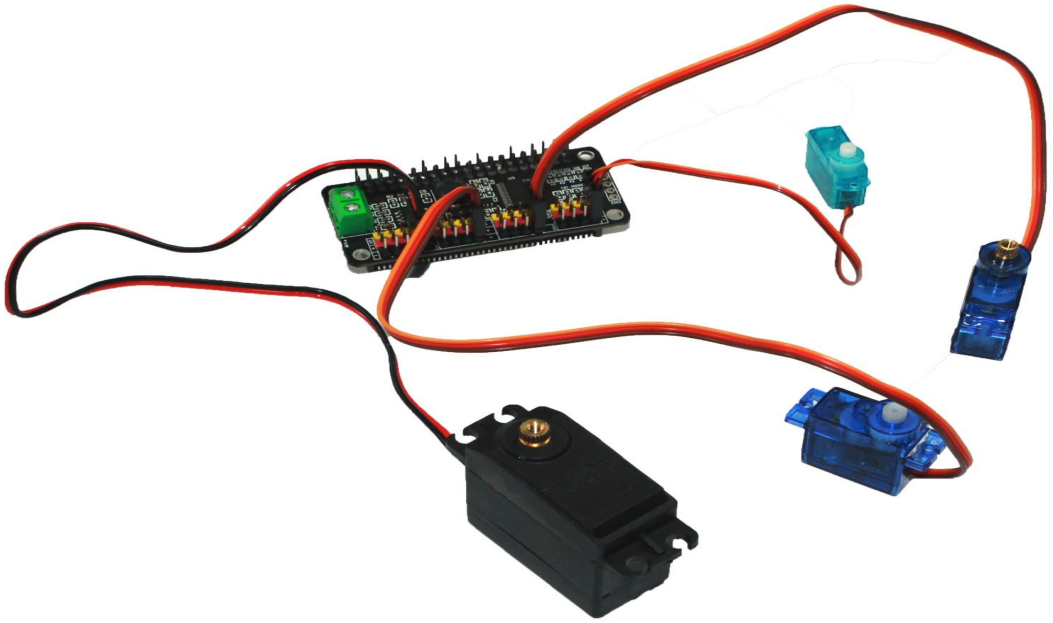
Stack the driver on to the Raspberry Pi.  It is stackable over all Raspberry Pi 40 pins models. After stacking you have access to header pins on the top of the board. Align the HAT over the Raspberry pi such that both male and female headers are over each other.

## Power supply

Connect the board with the power supply, it will automatically power up the Raspberry Pi. No external power supply is required to power up the Pi.

## Connecting Servo Motor

Connect the servo motors on the HAT and program it accordingly.



Colour description of the servo header and the servo motor is mentioned below.

| Servo Header | Functions |
|---|---|
| Yellow | Signal |
| Red | Vcc |
| Black | GND |

| Servo wires | Functions |
|---|---|
| Orange | Signal |
| Red | Vcc |
| Brown | GND |

## Check for the device

After connecting everything check for the device whether it is connected to the I2C bus. To check - got to the terminal and run a command and you will see the address of the device on the terminal as shown below.

In the terminal:

***"i2cdetect -y 1"***

```
pi@raspberrypi:~ $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: 40 -- -- -- -- -- -- -- UU -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- --
pi@raspberrypi:~ $
```

**Note:** If no address is found then, in that case, make sure that the I2C is enabled in your device. And if still the same issue is found then it might be that there is some soldering or hardware issue.

# PROGRAMMING SERVO DRIVER

## Installing Packages

Importing SMBus library.

Command - *sudo pip3 install smbus*

## Cloning directory

GitHub link - https://github.com/sbcshop/Raspberry-Pi-Servo-Driver-

The related example files can also be cloned to your Raspberry Pi, use git clone command. In terminal type-

***https://github.com/sbcshop/Raspberry-Pi-Servo-Driver-.git***

Once cloning is successful you can use the example codes and the library file servo_controller.py

## Running the program

Go to the directory and open the servo_contoller.py file with python editor and run the program.