

---

# Raspberry Pi Pico Motor Driver Expansion Board

## Description

Raspberry Pi Pico driver expansion board is suitable for 2WD or 4WD

## Features

- Standard Raspberry Pi Pico header, supports Raspberry Pi Pico series
- I2C controlled, supports 32x different I2C addresses by setting the 5 address jumpers
- Onboard PCA9685 chip, provides 12-bit hardware PWM to adjust motor speed
- Onboard TB6612FNG dual H-bridge motor driver, high efficiency, low heating
- Integrates 5V regulator, up to 3A output current, can be powered from battery through VIN terminal
- Breakout unused pins of Pico for easy extension

- 
- Comes with development resources and manual (Raspberry Pi Pico C/C++ and MicroPython examples)

## **Specifications**

Operating voltage: 6~12V (VIN terminals)

Logic level: 3.3V

PWM driver: PCA9685

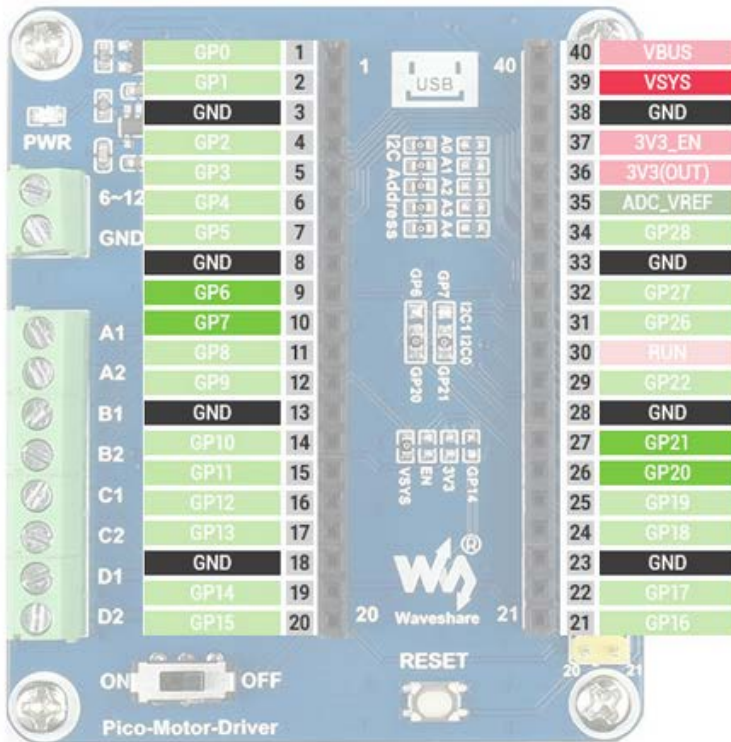
Motor driver: TB6612FNG

Control interface: I2C

Diameter: 3.0mm

Dimensions: 65 × 56mm

# Pinout Definition



VSYS	5V power supply
GND	Ground
GP6	SDA I2C SDA (alternative)
GP7	SCL I2C SCL (alternative)
GP20	SDA I2C SDA (default)
GP21	SCL I2C SCL (default)

Wire up hardware

Don't connect the Pico reversely

Observe an end with silk prints on the module and an end of USB port to determine connection direction.

You can also depend on signals of pins and pins of Pico to determine connection direction.

---

## Download via Raspberry Pi, open the Raspberry Pi terminal:

```
sudo apt-get install p7zip-full
```

```
cd ~
```

```
sudo wget
```

```
https://www.waveshare.net/w/upload/5/5c/Pico-Motor-Driver-code.7z
```

```
7z Pico_pico-Motor-Driver-code.7z -o./pico-Motor-Driver-code.7z
```

```
cd ~/pico-Motor-Driver-code
```

Click the example program to download directly

## Use C via Raspberry Pi

We use the Raspberry Pi. Because cmake has multiple platforms and can be moved, you can compile on the PC.

Compile under the C directory

```
cd ~/pico-Motor-Driver-code/c/
```

Create and enter build directory in the folder and add SDK.

```
../../pico-sdk is the directory of the SDK.
```

The example program has build, jsut enter it.

---

cd build

export PICO\_SDK\_PATH=../../pico-sdk

(Note: write the correct path of your own SDK)

Implement cmake and generate into Makefile files

cmake ..

Implement make and generate implement files, the first compile will take a while.

make -j9

After compiling, the uf2 file will be generated.

Press a key of the Pico board, connect the pico board to the Raspberry Pi via a USB cable and release the key.

Then Raspberry Pi will recognize a drive( RPI-RP2), copy the main.uf2 from the build folder to the drive(RPI-RP2).

cp main.uf2 /media/pi/RPI-RP2/

## **Python**

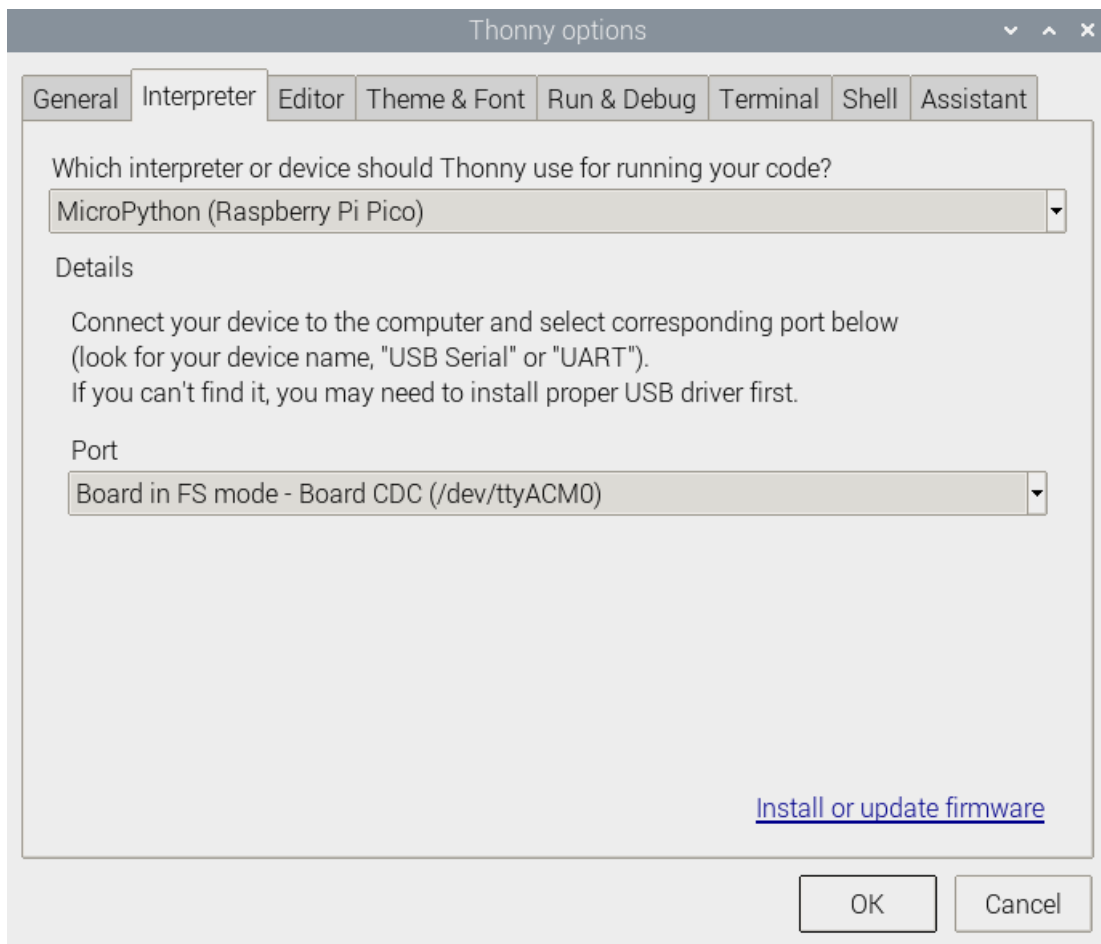
1. Update the firmware of Micropython, copy the pico\_micropython\_XXXXX.uf2 file to the pico.

---

2. Open Thonny IDE on the Raspberry Pi(click Raspberry Pi-> Programming -> Thonny Python IDE), you can check the version information: Help->About Thonny

Make sure this version contain Pico support package, and click Tools -> Options... -> Interpreter, then select MicroPython(Raspberry Pi Pico and the ttyACM0 port

As shown below;



If the Thonny version does not have the pico support package, enter the following command to update the Thonny IDE `sudo apt upgrade thonny`

---

Click File->Open...->python/pico-Motor-Driver-code/PCA9685.py to run the script

Motors A and B will rotate at full speed for 2S clockwise, and then rotate at half speed anticlockwise for 4S

## **python**

### **Windows environment**

Press and hold the BOOTSET button on the Pico board, connect the pico to the USB port of the computer through the Micro USB cable, and release the button after the computer recognizes a removable hard disk (RPI-RP2).

Download the pico\_micropython\_XXXXX.uf2, then copy it to the drive(RPI-RP2).

Open Thonny IDE(note: use the latest Thonny, otherwise, the Pico support package is not included. The latest version under Windows is v3.3.3

Click tool->setting->interpreter, then select the corresponding port of Pico

Click file->open->servo.py and click run

The following picture indicates that the program has run.

---

Shell ✕

```
MicroPython v1.13-290-g556ae7914 on 2021-01-21; Raspberry Pi Pico with RP2040  
Type "help()" for more information.  
>>> %Run -c $EDITOR_CONTENT
```

The experiment result is as same as the program C