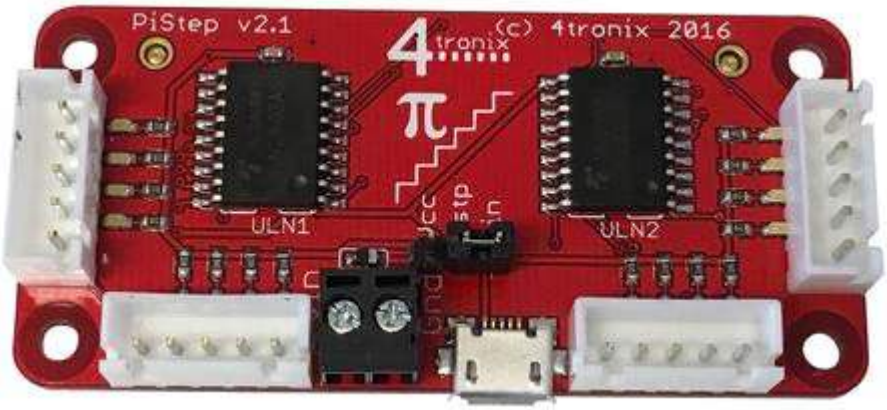
PiStep2 Quad Stepper Motor Control Board for Raspberry Pi

4TR PISTEP2Q



This neat little board plugs directly into the Raspberry Pi GPIO header and provides 2 or 4 connectors for small stepper motors

- Fully Assembled - No Soldering Required
- Raspberry Pi Zero Form Factor - works with all versions of Raspberry Pi with 40pin GPIO connector
- Stepper motors and Raspberry Pi not included

Various Power Options:

1. Powered from the Raspberry Pi 5V
2. From the 2-pin Terminal (whatever voltage is required for the motors)
3. Micro-USB - 5V only

Pinout is simple:

- Physical pins 11, 12, 13, 15 for Motor A (GPIO 17, 18, 27, 22)
- Physical pins 16, 18, 22, 7 for Motor B (GPIO 23, 24, 25, 04)
- Physical pins 33, 32, 31, 29 for Motor C (GPIO 13,12,6,5)
- Physical Pins 38, 37, 36, 35 for Motor D (GPIO 20,26,16,19)

Each pin has an associated white LED so you can see the stepper signals going through

Power Supply Alternatives

- **Jumper VCC-VSTP** (default). Power from the motors is taken from the Raspberry Pi 5V line
    - Micro-USB into Raspberry Pi. 5V for the Pi and the stepper motors goes through a poly-fuse which can trip if 2 motors are used simultaneously
    - Micro-USB into the PiStep board. 5V for both the Pi and the stepper motors is provided directly from the 5V USB input so no problems with 2 motors at once
- **Jumper VSTP-VIN**. Power for the motors is provided from the 2-pin screw terminal, so can be any voltage that the steppers can handle. Ensure you use the correct polarity! We recommend to keep it below 12V. You will find that the stepper motors can go up to 9V and will be able to step faster, the higher the voltage that is applied, but there will be some deterioration of the life of the stepper motor at a higher voltage.

Python Programming

Please see the excellent example here for some pointers. You will need to change the pin numbers as above and also change the speed so it steps at a visible rate. You may also want to remove the print statements to speed it up. The lines in Red below are changed from the original to operate Motor A.

```
# Use BCM GPIO references
# instead of physical pin numbers
GPIO.setmode(GPIO.BCM)

# Define GPIO signals to use
# Pins 18,22,24,26
# GPIO24,GPIO25,GPIO8,GPIO7
StepPins = [17,18,27,22]

# Set all pins as output
for pin in StepPins:
  print "Setup pins"
```

```
  GPIO.setup(pin,GPIO.OUT)
  GPIO.output(pin, False)

# Define some settings
StepCounter = 0
WaitTime = 0.01
```

ScratchGPIO Programming

These pins are identical to those required by ScratchGPIO and therefore can easily be driven using simple Scratch commands: https://cymplecy.wordpress.com/scratchgpio/

- Set motor type for Scratch to be Stepper motor

- Set the position of the stepper motor A

- Set the speed of the stepper motor A