



# Adafruit's Raspberry Pi Lesson 4. GPIO Setup

Created by Simon Monk



<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup>

Last updated on 2024-06-03 01:13:50 PM EDT

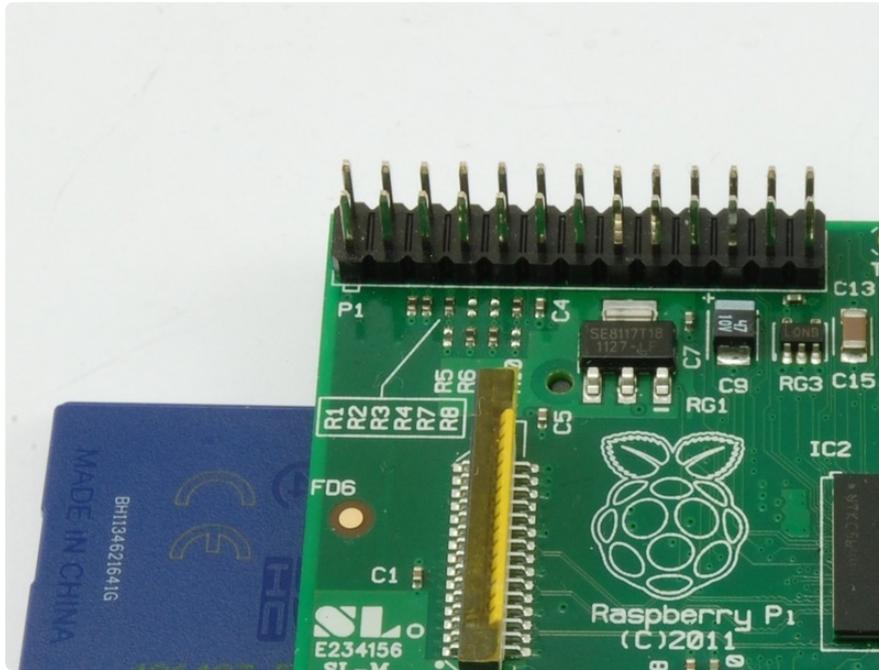
# Table of Contents

Overview	3
The GPIO Connector	4
Adafruit Pi Code	5
Configuring I2C	6
<ul style="list-style-type: none"><li>• Installing Kernel Support (with Raspi-Config)</li><li>• Testing I2C</li></ul>	
Configuring SPI	10
<ul style="list-style-type: none"><li>• Installing Kernel Support (with Raspi-Config)</li></ul>	
Test and Configure	12

---

# Overview

One of the great things about the Raspberry Pi is that it has a GPIO connector to which you can attach external hardware.



The GPIO connector actually has a number of different types of connection on them. There are:

- True GPIO (General Purpose Input Output) pins that you can use to turn LEDs on and off etc.
- I2C interface pins that allow you to connect hardware modules with just two control pins
- SPI interface with SPI devices, a similar concept to I2C but a different standard
- Serial Rx and Tx pins for communication with serial peripherals

In this tutorial, you are not actually building anything, but you will learn how to configure your Raspberry Pi and install useful libraries ready to start attaching some external electronics to it.

This tutorial is written for Raspbian based distributions.

# The GPIO Connector

The diagram below show GPIO pinouts used on different models of the Raspberry Pi. The earlier revisions of the Raspberry Pi were 26-pin based while the newer models are 40-pin.

## GPIO Numbers

Raspberry Pi B Rev 1 P1 GPIO Header			Raspberry Pi A/B Rev 2 P1 GPIO Header			Raspberry Pi B+ B+ J8 GPIO Header		
	Pin No.			Pin No.			Pin No.	
3.3V	1	2	5V	1	2	5V	1	2
GPIO0	3	4	5V	3	4	5V	3	4
GPIO1	5	6	GND	5	6	GND	5	6
GPIO4	7	8	GPIO14	7	8	GPIO14	7	8
GND	9	10	GPIO15	9	10	GPIO15	9	10
GPIO17	11	12	GPIO18	11	12	GPIO18	11	12
GPIO21	13	14	GND	13	14	GND	13	14
GPIO22	15	16	GPIO23	15	16	GPIO23	15	16
3.3V	17	18	GPIO24	17	18	GPIO24	17	18
GPIO10	19	20	GND	19	20	GND	19	20
GPIO9	21	22	GPIO25	21	22	GPIO25	21	22
GPIO11	23	24	GPIO8	23	24	GPIO8	23	24
GND	25	26	GPIO7	25	26	GPIO7	25	26
						DNC	27	28
						GND	29	30
						GPIO12	31	32
						GND	33	34
						GPIO16	35	36
						GPIO20	37	38
						GPIO21	39	40

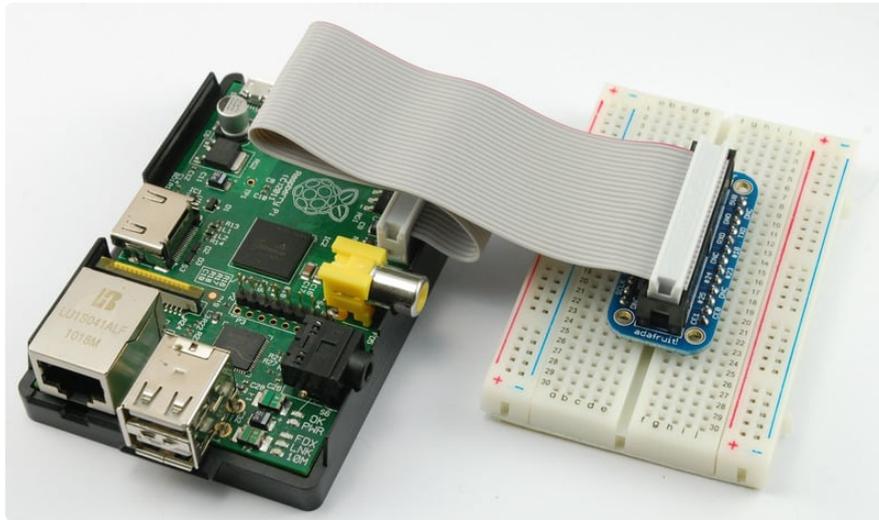
Key	
Power +	UART
GND	SPI
I <sup>2</sup> C	GPIO

As well as supplying power (GND, 3.3V and 5V) all the GPIO pins can be used as either digital inputs or outputs. The pins labelled SCL and SDA can be used for I2C. The pins labelled MOSI, MISO and SCKL can be used to connect to high speed SPI devices.

All the pins have 3.3V logic levels and are not 5V-safe so the output levels are 0-3.3V and the inputs should not be higher than 3.3V. If you want to connect a 5V output to a Pi input, [use a level shifter \(https://adafruit.com/product/1754\)](https://adafruit.com/product/1754)

A popular way to actually make the connections to the Raspberry Pi is to use a [Pi Cobbler Plus \(http://adafruit.com/product/2029\)](http://adafruit.com/product/2029) or a [Pi T-Cobbler Breakout \(http://adafruit.com/product/1754\)](http://adafruit.com/product/1754) for older 26-pin versions.

This uses a ribbon cable to connect the GPIO connector to solderless breadboard, where you can add your own components.



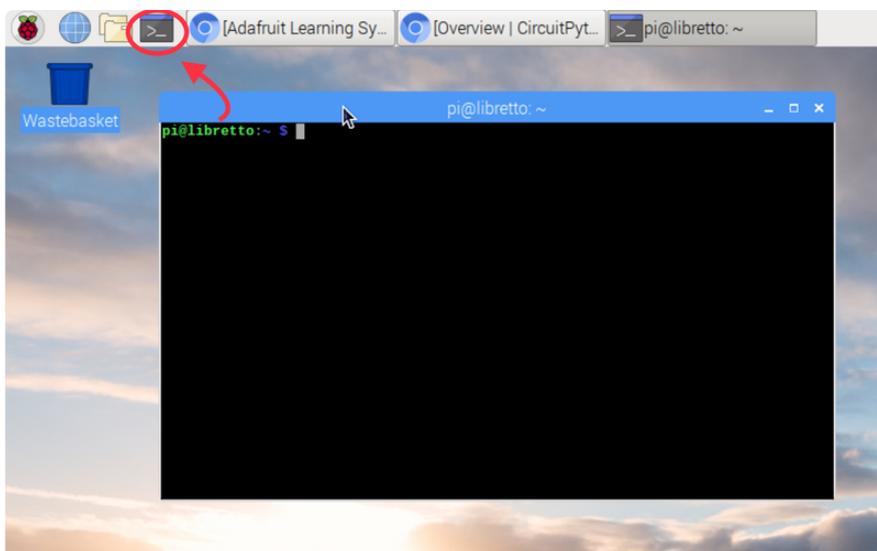
Make extra extra double-check sure that the PIN 1 indicator is in the corner of the Pi. If you have a gray cable its probably a red stripe, for black cables, a white stripe. That pin must not be next to the TV connector. Turn around or twist the cable until it is right

## Adafruit Pi Code

To make life easy for those wishing to experiment with attaching electronics to their Pi, Adafruit have produced an extensive and extremely useful collection of code. This includes simple [CircuitPython Libraries \(https://adafru.it/EQa\)](https://adafru.it/EQa) for a large number of modules, including displays, sensors, actuators and more.

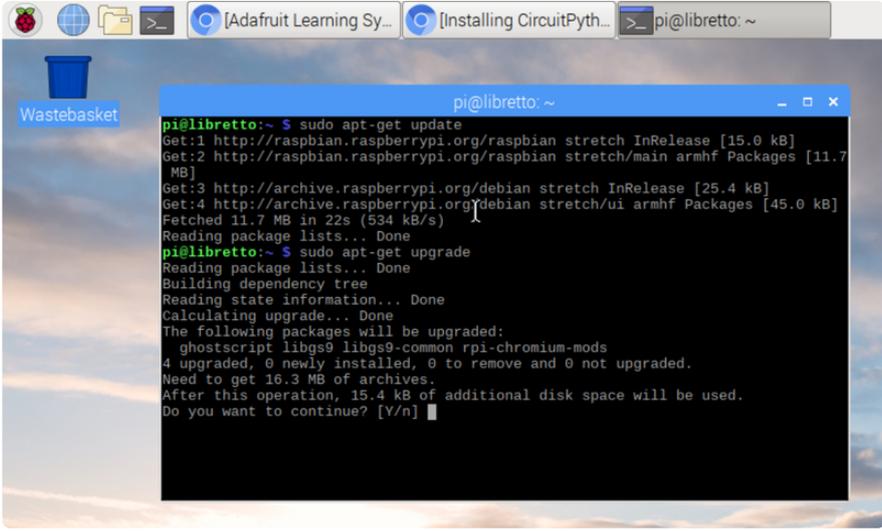
To fetch this code, you need to use some software called 'git' .

You will find the icon for a Terminal on your desktop.



Before we go any further, issue the following commands in a Terminal. This will ensure your packages are up to date. It does not matter which directory you are in.

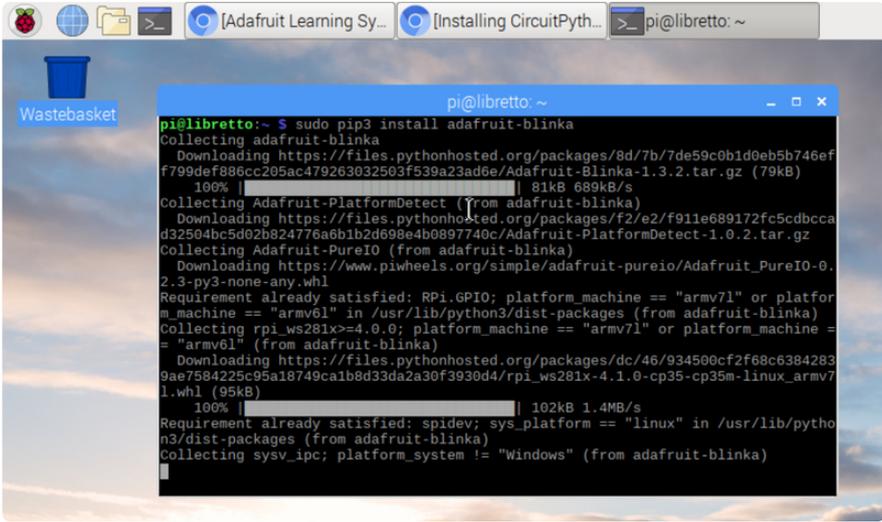
```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get dist-upgrade -y
```



```
pi@libretto:~$ sudo apt-get update
Get:1 http://raspbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]
Get:2 http://raspbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7 MB]
Get:3 http://archive.raspberrypi.org/debian stretch InRelease [25.4 kB]
Get:4 http://archive.raspberrypi.org/debian stretch/ui armhf Packages [45.0 kB]
Fetched 11.7 MB in 22s (534 kB/s)
Reading package lists... Done
pi@libretto:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  ghostscript libgs9 libgs9-common rpi-chromium-mods
4 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 16.3 MB of archives.
After this operation, 15.4 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Run the following command to install the `adafruit_blinka` CircuitPython Libraries.

```
sudo apt install python3-pip
pip3 install adafruit-blinka
```



```
pi@libretto:~$ sudo pip3 install adafruit-blinka
Collecting adafruit-blinka
  Downloading https://files.pythonhosted.org/packages/8d/7b/7de59c0b1d0eb5b746ef
f799def886cc205ac479263032503f539a23ad6e/Adafruit-Blinka-1.3.2.tar.gz (79kB)
  100% |#####| 81kB 689kB/s
Collecting Adafruit-PlatformDetect (from adafruit-blinka)
  Downloading https://files.pythonhosted.org/packages/f2/e2/f911e689172fc5cdbcba
d32504bc5d02b824776a6b1b2d698e4b0897740c/Adafruit-PlatformDetect-1.0.2.tar.gz
Collecting Adafruit-PureIO (from adafruit-blinka)
  Downloading https://www.piwheels.org/simple/adafruit-pureio/Adafruit_PureIO-0.
2.3-py3-none-any.whl
Requirement already satisfied: RPi.GPIO; platform_machine == "armv7l" or platfor
m_machine == "armv6l" in /usr/lib/python3/dist-packages (from adafruit-blinka)
Collecting rpi_ws281x>=4.0.0; platform_machine == "armv7l" or platform_machine =
"armv6l" (from adafruit-blinka)
  Downloading https://files.pythonhosted.org/packages/dc/46/934500cf2f68c6384283
9ae7504225c95a18749ca1b8d33da2a30f3930d4/rpi_ws281x-4.1.0-cp35-cp35m-linux_armv7
l.whl (95kB)
  100% |#####| 102kB 1.4MB/s
Requirement already satisfied: spidev; sys_platform == "linux" in /usr/lib/pytho
n3/dist-packages (from adafruit-blinka)
Collecting sysv_ipc; platform_system != "Windows" (from adafruit-blinka)
```

## Configuring I2C

I2C is a very commonly used standard designed to allow one chip to talk to another. So, since the Raspberry Pi can talk I2C we can connect it to a variety of I2C capable chips and modules.

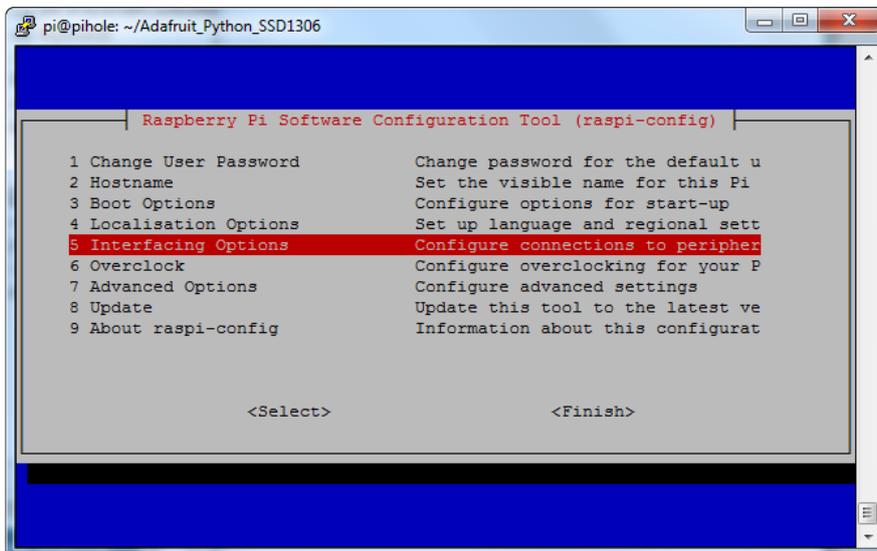
The I2C bus allows multiple devices to be connected to your Raspberry Pi, each with a unique address, that can often be set by changing jumper settings on the module. It is very useful to be able to see which devices are connected to your Pi as a way of making sure everything is working.

```
sudo apt-get install -y python3-smbus
sudo apt-get install -y i2c-tools
```

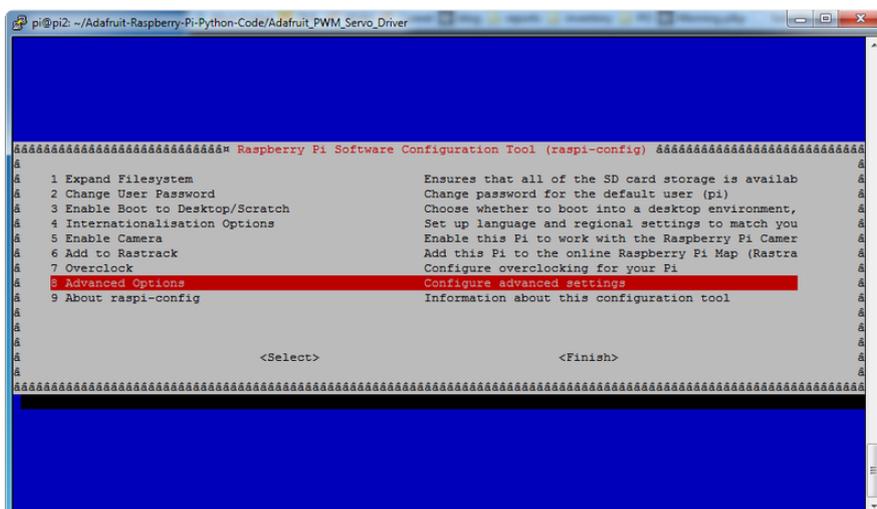
## Installing Kernel Support (with Raspi-Config)

Run `sudo raspi-config` and follow the prompts to install i2c support for the ARM core and linux kernel

Go to Interfacing Options



On older versions, look under **Advanced**

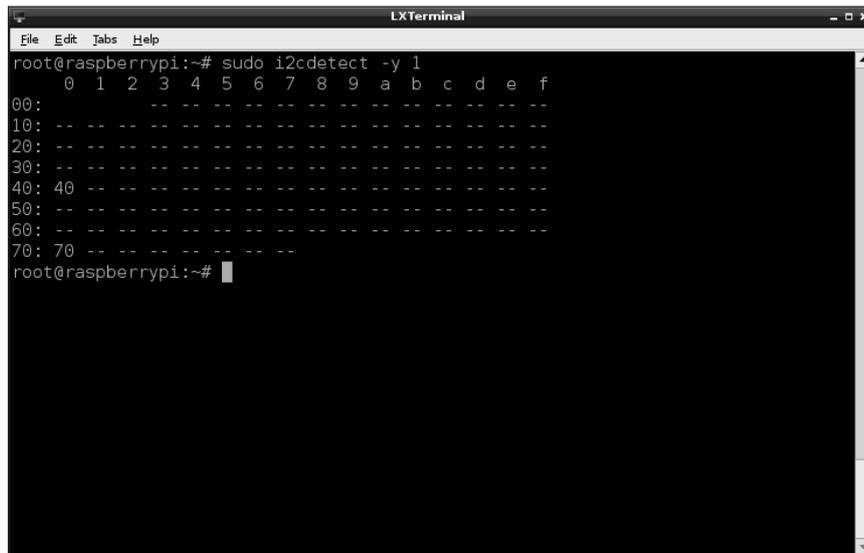




# Testing I2C

Now when you log in you can type the following command to see all the connected devices

```
sudo i2cdetect -y 1
```



```
root@raspberrypi:~# sudo 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 -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- -- -- -- -- -- -- -- --
root@raspberrypi:~#
```

This shows that two I2C addresses are in use – 0x40 and 0x70.

These values will be different for you depending on what is currently attached to the I2C pins of your Raspberry Pi

Note that if you are using one of the very first Raspberry Pis (a 256MB Raspberry Pi Model B) then you will need to change the command to:

```
sudo i2cdetect -y 0
```

The Raspberry Pi designers swapped over I2C ports between board releases. Just remember: 512M Pi's use i2c port 1, 256M ones use i2c port 0!

When you are finished in raspi-config reboot for the i2c modules to automatically load into the kernel.

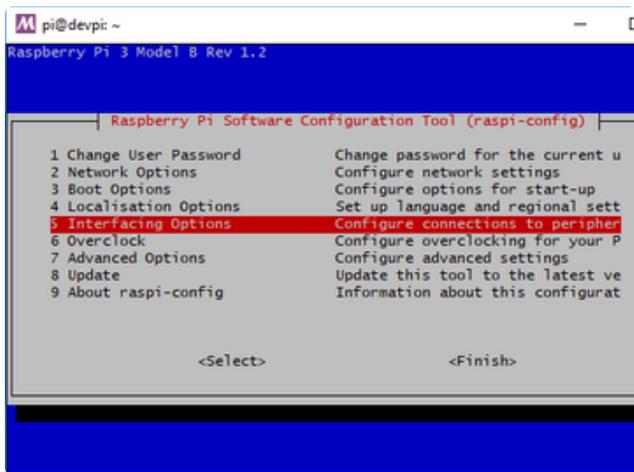
---

# Configuring SPI

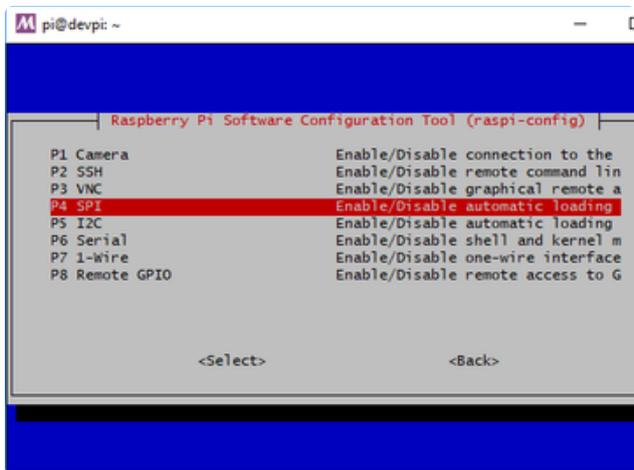
## Installing Kernel Support (with Raspi-Config)

Run `sudo raspi-config` and follow the prompts to install i2c support for the ARM core and linux kernel

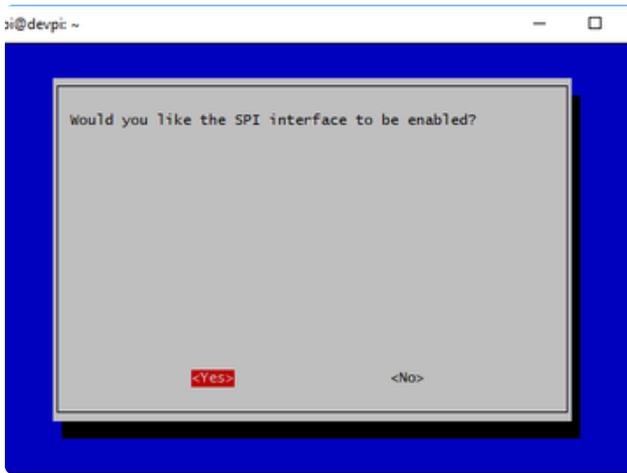
Go to Interfacing Options



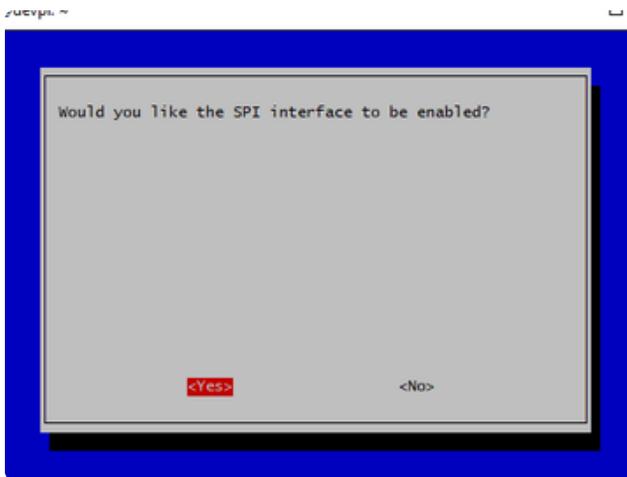
Go to Interfacing Options



Then select SPI



When asked if you want to **enable** select **YES**



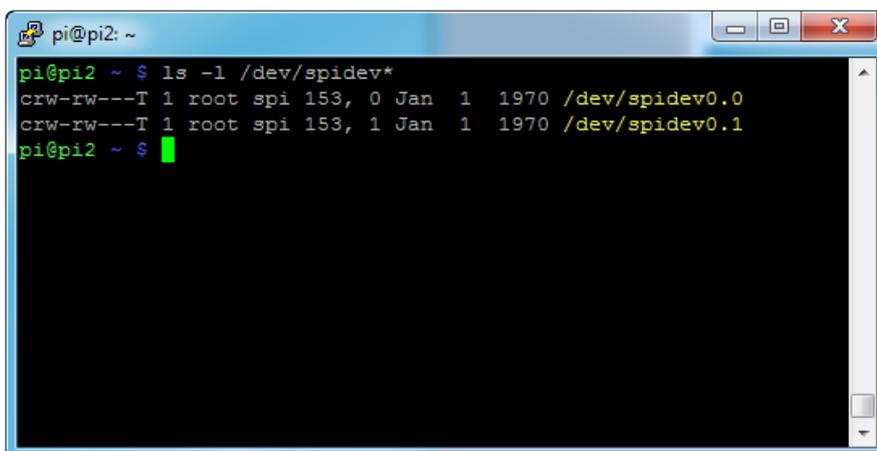
That's it!

Now reboot your Pi to make the SPI interface appear

Next time you log in you can check that you can see the devices with

`ls -l /dev/spidev*`

you should see two 'devices' one for each SPI bus



The `/dev/spidev0.x` will not show up when using the PiTFT screens as it is using the SPI interface.

---

## Test and Configure

The best way to test what you have done is to pick one of the projects (perhaps a fairly simple one to start with) and give it a try.

Here are some of the tutorials I would recommend as a first project:

- <http://learn.adafruit.com/raspberry-pi-e-mail-notifier-using-leds> (<https://adafru.it/aJ5>)
- <http://learn.adafruit.com/playing-sounds-and-using-buttons-with-raspberry-pi> (<https://adafru.it/aTD>)
- <http://learn.adafruit.com/matrix-7-segment-led-backpack-with-the-raspberry-pi> (<https://adafru.it/aPj>)
- <https://learn.adafruit.com/adafruits-raspberry-pi-lesson-12-sensing-movement/overview> (<https://adafru.it/sdr>)
- <https://learn.adafruit.com/adafruits-raspberry-pi-lesson-13-power-control/hardware> (<https://adafru.it/EQQ>)

[Click Here for the Next Lesson](https://adafru.it/aUA)

<https://adafru.it/aUA>