# Table of Contents

Overview 3

The GPIO Connector 4

Adafruit Pi Code 5

Configuring I2C 6
- Installing Kernel Support (with Raspi-Config)
- Testing I2C

Configuring SPI 10
- Installing Kernel Support (with Raspi-Config)

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 shows 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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3V</td>
<td>1 2 SV</td>
<td>3.3V 1 2 SV</td>
<td>3.3V 1 2 SV</td>
</tr>
<tr>
<td>GPIO0</td>
<td>3 4 GPIO14</td>
<td>GPIO2 3 4 V</td>
<td>GPIO2 3 4 V</td>
</tr>
<tr>
<td>GPIO1</td>
<td>5 6 GND</td>
<td>GPIO3 5 6 GND</td>
<td>GPIO3 5 6 GND</td>
</tr>
<tr>
<td>GPIO2</td>
<td>7 8 GND</td>
<td>GPIO4 7 8 GND</td>
<td>GPIO4 7 8 GND</td>
</tr>
<tr>
<td>GPIO3</td>
<td>9 10 GPIO15</td>
<td>GPIO5 9 10 GND</td>
<td>GPIO5 9 10 GPIO15</td>
</tr>
<tr>
<td>GPIO4</td>
<td>11 12 GPIO18</td>
<td>GPIO6 11 12 GPIO18</td>
<td>GPIO6 11 12 GPIO18</td>
</tr>
<tr>
<td>GPIO5</td>
<td>13 14 GPIO23</td>
<td>GPIO7 13 14 GPIO23</td>
<td>GPIO7 13 14 GPIO23</td>
</tr>
<tr>
<td>GPIO6</td>
<td>15 16 GPIO22</td>
<td>GPIO8 15 16 GPIO22</td>
<td>GPIO8 15 16 GPIO22</td>
</tr>
<tr>
<td>GPIO7</td>
<td>17 18 GPIO24</td>
<td>GPIO9 17 18 GPIO24</td>
<td>GPIO9 17 18 GPIO24</td>
</tr>
<tr>
<td>GPIO8</td>
<td>19 20 GPIO10</td>
<td>GPIO10 19 20 GND</td>
<td>GPIO10 19 20 GND</td>
</tr>
<tr>
<td>GPIO9</td>
<td>21 22 GPIO25</td>
<td>GPIO11 21 22 GPIO25</td>
<td>GPIO11 21 22 GPIO25</td>
</tr>
<tr>
<td>GPIO10</td>
<td>23 24 GPIO11</td>
<td>GPIO12 23 24 GPIO12</td>
<td>GPIO12 23 24 GPIO12</td>
</tr>
<tr>
<td>GPIO11</td>
<td>25 26 GPIO6</td>
<td>GPIO13 25 26 GPIO7</td>
<td>GPIO13 25 26 GPIO7</td>
</tr>
<tr>
<td>GND</td>
<td></td>
<td>DNC 27 28 GPIO17</td>
<td>DNC 27 28 DNC</td>
</tr>
<tr>
<td>UART</td>
<td></td>
<td>GPIO18 29 30 GPIO19</td>
<td>GPIO18 29 30 GPIO19</td>
</tr>
<tr>
<td>SPI</td>
<td></td>
<td>GPIO20 31 32 GPIO21</td>
<td>GPIO20 31 32 GPIO21</td>
</tr>
<tr>
<td>PC</td>
<td></td>
<td>GPIO21 33 34 GPIO12</td>
<td>GPIO21 33 34 GPIO12</td>
</tr>
<tr>
<td>GPIO</td>
<td></td>
<td>GPIO16 35 36 GPIO17</td>
<td>GPIO16 35 36 GPIO17</td>
</tr>
</tbody>
</table>

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.

A popular way to actually make the connections to the Raspberry Pi is to use a Pi Cobbler Plus or a Pi T-Cobbler Breakout 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.
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 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.

Make 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.
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
```

Run the following command to install the adafruit_blinka CircuitPython Libraries.

```
pip3 install 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 python-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
then I2C

Enable!

sudo reboot
Testing I2C

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

```bash
sudo i2cdetect -y 1
```

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:

```bash
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

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
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://learn.adafruit.com/adafruits-raspberry-pi-lesson-12-sensing-movement/overview
- https://learn.adafruit.com/adafruits-raspberry-pi-lesson-13-power-control/hardware

Click Here for the Next Lesson