

[Account](#)[Cart](#)

Knowledge Base

The AB Electronics UK Knowledge Base provides support solutions, tutorials and troubleshooting guides.



[Home](#) / [Help and Support](#) / [Knowledge Base](#) / [Getting Started](#) / I2C Part 2 - Enabling I²C on the Raspberry Pi

I2C Part 2 - Enabling I²C on the Raspberry Pi

How to enable I²C on the Raspberry Pi

Created: 02/10/2015 | Last Updated: 27/05/2020

This tutorial is the second in a four-part series on the communication protocol I²C; explaining what it is, how it works and how you can use it on your Raspberry Pi.

For the other tutorials in this series click on the links below.

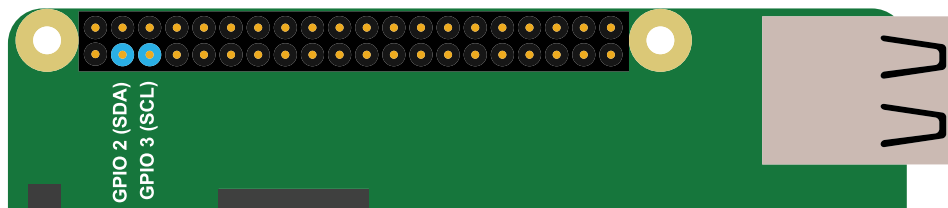
- [Part 1 : Introducing I²C](#)
- **Part 2: Enabling I²C on the Raspberry Pi**
- [Part 3: I²C tools in Linux](#)
- [Part 4: Programming I²C with Python](#)

Introduction

This tutorial explains how to install and configure I²C on the Raspbian Buster and later distributions of Linux for the Raspberry Pi from <http://www.raspberrypi.org/downloads>.

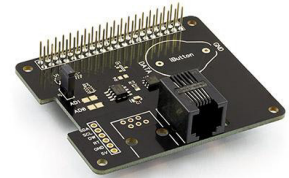
I²C, pronounced I-squared-C or I-2-C, is a popular serial bus used by a wide range of manufacturers for low speed serial communication.

The I²C port on the Raspberry Pi uses the pins GPIO2 (SDA) and GPIO3 (SCL) as shown below.



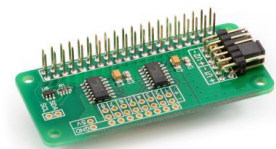
Enabling I²C using the Raspbian Desktop

Related Products



1 Wire Pi Plus

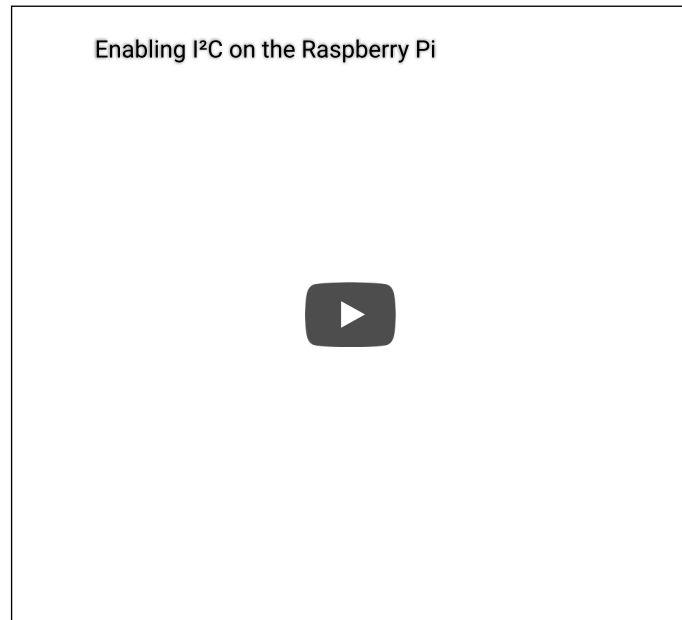
£11.99 Inc VAT: £14.39



ADC Differential Pi

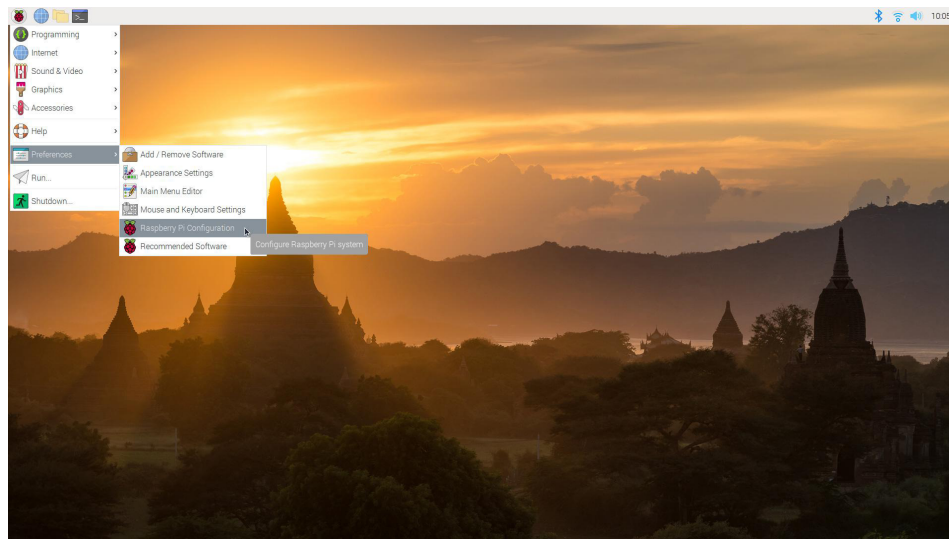
£14.99 Inc VAT: £17.99

Watch the video or follow the steps below to enable the I²C port from within the Raspbian Desktop environment.



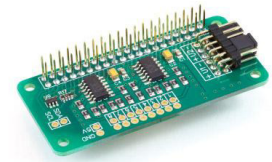
Step 1:

Begin by turning on your Raspberry Pi and allow Linux to boot to the desktop. Once booted click on the start menu in the top left corner. Click the preferences icon and select Raspberry Pi Configuration.



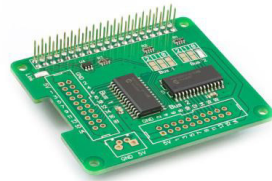
Step 2:

After a few seconds, the Raspberry Pi Configuration program will open. Click on the Interfaces tab and find the I²C option. Click on the Enable button next to I²C and then click the OK button.



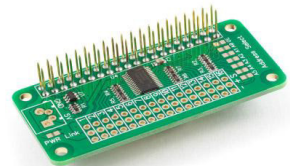
ADC Pi

£15.99 Inc VAT: £19.19



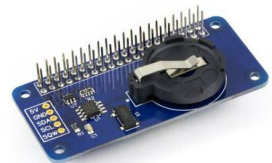
IO Pi Plus

£12.49 Inc VAT: £14.99



Servo PWM Pi

£8.49 Inc VAT: £10.19



RTC Pi

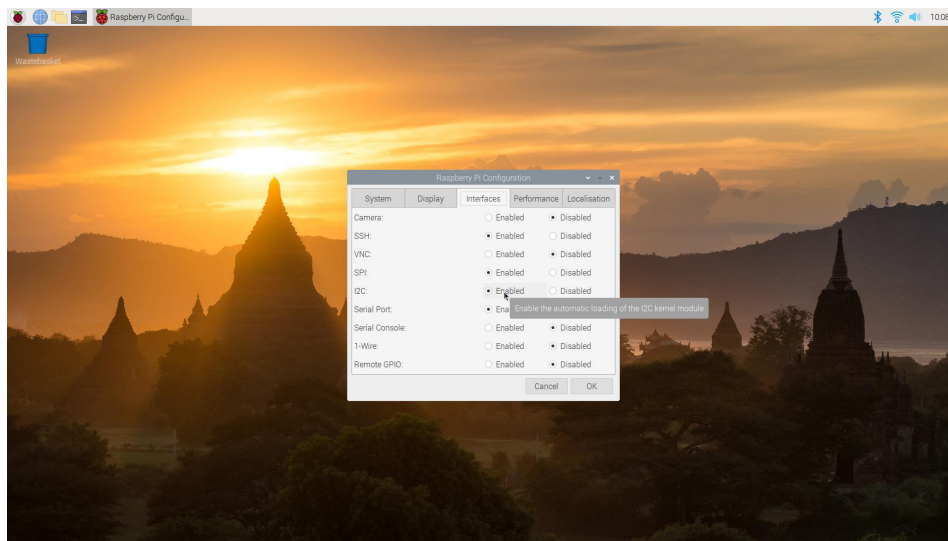
£8.25 Inc VAT: £9.90

Related Articles

[PCB Header Assembly Jig](#)

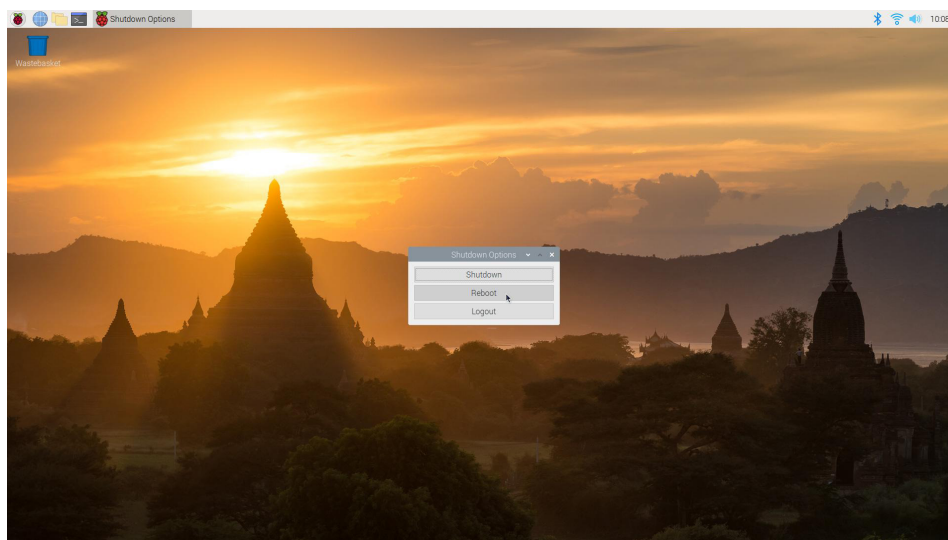
[Raspbian Jessie or Stretch](#)

[Static IP Setup](#)



Step 3:

Before you can use the I²C bus you will need to reboot your Raspberry Pi so click again on the start menu icon. Click shutdown at the bottom of the menu and in the dialog box that appears click reboot.



Enabling I²C in the Raspbian Linux Terminal

Step 1: Download the latest Raspbian linux image and burn to your SD Card following the instructions on http://elinux.org/RPi_Easy_SD_Card_Setup

If you are using Raspbian Linux 3.18 or later you need to go into the raspberry pi config utility and enable I²C.

```
sudo raspi-config
```

Select **5 Interfacing Options** and then **P5 I2C**. A prompt will appear asking **Would you like the ARM I2C interface to be enabled?**, select Yes, exit the utility and reboot your raspberry pi.

```
sudo reboot
```

When your Raspberry Pi has rebooted you will now have I²C and SMBus Python 2 or 3 support.

Installing I2C Tools and Python Libraries

[Samba Setup on Raspberry Pi](#)

[I2C Part 1 - Introducing I2C](#)

[I2C Part 2 - Enabling I²C on the Raspberry Pi](#)

[I2C Part 3 - I²C tools in Linux](#)

[I2C Part 4 - Programming I²C with Python](#)

[SPI and Raspbian Linux on a Raspberry Pi](#)

[Using Pythonpath with the Python Libraries](#)

[I2C, SMBus and Raspbian Stretch Linux](#)

[I2C, SMBus and Raspbian Wheezy Linux](#)

If you are using the Raspbian Lite version of Raspbian Linux you may need to install the i2ctools package as well as several python libraries. You can do this using the following commands.

```
sudo apt-get update
sudo apt-get install python-smbus python3-smbus python-dev python3-dev
i2c-tools
```

You can test if i2ctools is working by listing all of the I²C devices connected to your Raspberry Pi.

```
sudo i2cdetect -y 1
```

i2cdetect will display a grid of numbers with the addresses of any I²C devices shown within the grid. In the following example two I²C devices are shown on addresses 0x68 and 0x69.

```
      0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60:  -- -- -- -- -- -- -- 68 69 -- -- -- -- -- --
70:  -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

Setting the I²C Bus Speed

The I²C bus supports several bus speeds, typically 100KHz, 400KHz, 1MHz, 3.4MHz and 5.0MHz. You can set the bus speed by editing /boot/config.txt. Open the file with nano using the command:

```
sudo nano /boot/config.txt
```

Add the following text to the bottom of the file; The number is the frequency of the I²C bus in hertz:

```
dtoverlay=i2c-baudrate=400000
```

Save your changes, exit the nano editor and reboot

```
sudo reboot
```

Notes for the Raspberry Pi 3B, 3B+ and Zero W

The I²C controller on the Raspberry Pi 3B, 3B+ and Zero W has its clock linked to the VPU core so as the VPU frequency changes depending on processor load so does the clock rate of the I²C bus. This can cause problems with any devices that expect a constant clock rate during communication.

To solve this issue the VPU core frequency must be set to a fixed frequency by editing /boot/config.txt. Open the file with nano using the command:

```
sudo nano /boot/config.txt
```

Add the following text to the bottom of the file.

```
core_freq=250
```

Save your changes, exit the nano editor and reboot

```
sudo reboot
```

Choosing the correct I²C bus speed

I²C devices are designed with a maximum speed for the I²C bus. This speed varies from device to device and can normally be found in the manufacturer's datasheet for that device. Devices will run at speeds below the maximum but not faster so when choosing the correct bus speed for your Raspberry Pi you will need to find the maximum speed for each device and then set the I²C bus to be the speed of the slowest device.

Other factors can also limit the speed of the I²C bus including the number of devices connected to the bus, cable length and cable capacitance so you may find that if I²C communication is not working correctly you will have to set the clock frequency to a slower speed.

The maximum recommended speed for the expansion boards made by AB Electronics UK is listed in the table below.

Expansion Board	Maximum Clock Frequency
1 Wire Pi Plus	400 KHz (400000)
1 Wire Pi Zero	400 KHz (400000)
ADC Pi	1 MHz (1000000)
ADC Differential Pi	1 MHz (1000000)
Expander Pi	100 KHz (100000)
IO Pi Plus	1 MHz (1000000)
IO Pi Zero	1 MHz (1000000)
RTC Pi	100 KHz (100000)
Servo PWM Pi	1 MHz (1000000)

I²C Device Addresses

Before the Raspberry Pi can communicate with an I²C device it needs to know the address assigned to that device. I²C devices typically use a 7-bit address which is sent at the start of each communication from the master to the slave device. 7 bits gives a total of 128 possible device addresses.

Some devices will have their address pre-set by the manufacturer while others will have a small range of addresses that can be selected by the user. Each device on the bus must have a unique address, if two devices share the same address then there can be a conflict that stops the devices from functioning correctly.

To find a list of the devices connected to the I²C bus on the Raspberry Pi you can use the following command:

```
sudo i2cdetect -y 1
```

On the original Raspberry Pi with the 26 pin GPIO header you may need to change the “-y 1” to “-y 0” for the command to work.

I2cdetect returns a grid of possible addresses with each device displayed as shown in the screenshot below.

```
pi@raspberrypidev:~ $ sudo i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: 20 21 -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: 40 -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: 70 -- -- -- -- -- -- -- -- -- -- -- -- -- --
pi@raspberrypidev:~ $
```

If no devices are detected check that you have followed all the steps at the start of this tutorial for enabling the I²C bus and the devices connected to the Raspberry Pi are powered and do not share the same I²C address.

[Part 1: Introducing I²C](#)

[Part 3: I²C tools in Linux](#)

Was this article helpful? [Yes](#) / [No](#)