Preface

About HiPi.io

We created HiPi.io as a project at BuyaPi.ca/PiShop.us because we love empowering people to create and modify technology in ways that are important to them. We find such satisfaction in inventing, coding, and hacking, and want to share these activities with anyone who's interested. Our products support a large spectrum of opportunities, from a child's first robot interaction to professional development tools.

As one of largest Raspberry Pi distributors, we work with everyone from local schools to Fortune 500 companies. To ensure that our products are affordable and within the reach of as many people as possible, we run a lean operation. We only sell online to avoid costly retail locations and provide basic sales support. We're a small and dedicated team, working hard to bring you the items you need!

About the HiPi.io Sensor Kit v4.0

This Sensor Kit is suitable for all Raspberry Pi boards with the 40-pin GPIO header including models B+/2B/3B/3B+/4B and the Zero family (some soldering may be required). It includes dozens of different modules for you to learn. We provide simple lessons that correspond to each sensor to help you quickly get started with your own projects!

In this manual, we will provide lessons with schematic diagrams and illustrations to show you how the circuits work and how to use the circuits with the provided code examples in C/wiringPi and Python3.

The HiPi.io Sensor Kit v4.0 product page is located at https://www.hipi.io/sk4 and the code can be found in our code repository at https://www.hipi.io/sk4/repo. The latest pre-configured Raspberry Pi OS image is located at https://HiPi.io/sk4/downloads.

Latest manual version

You can find the latest version of this manual in PDF format at https://www.hipi.io/sk4/doc. You can also find updates to the manual (Errata) there.

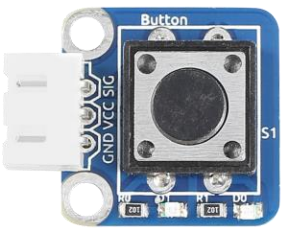
Free Support

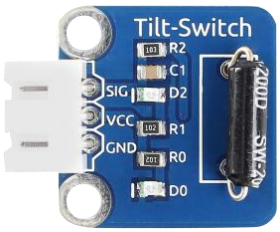
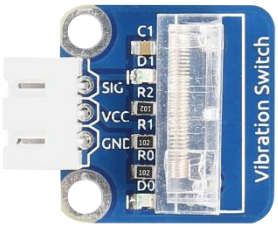
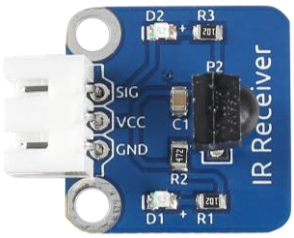
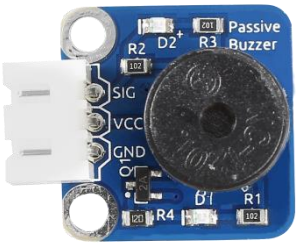
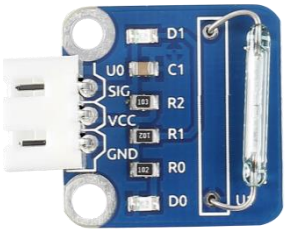
If you have any **TECHNICAL questions**, look for your problem in the **FAQ** section on our website and the **ISSUES** section of our code repository at https://www.hipi.io/sk4/repo. If you don't find an answer to your problem, open an issue in our code repository or send us an email!

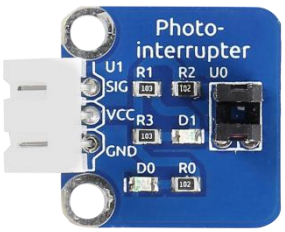
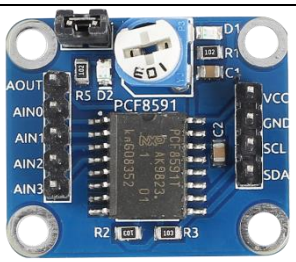
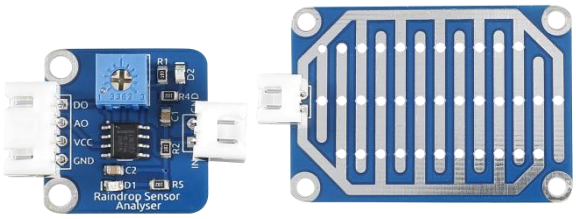
For **NON-TECHNICAL questions** like order and shipment issues, please **send an email to support@hipi.io**.

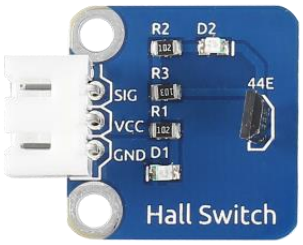
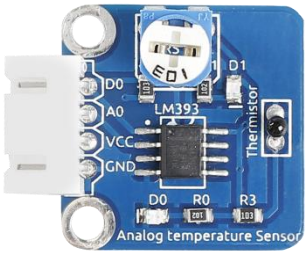
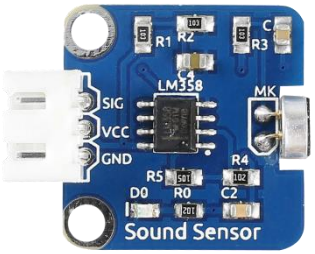
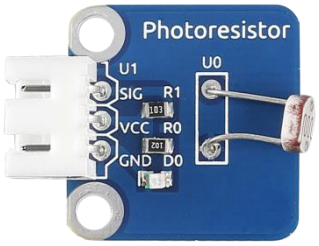
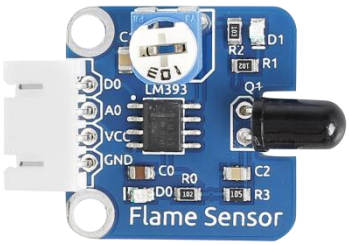
Table of Contents

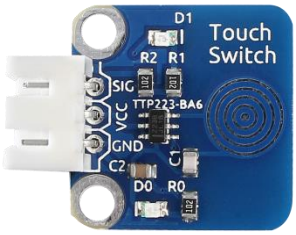
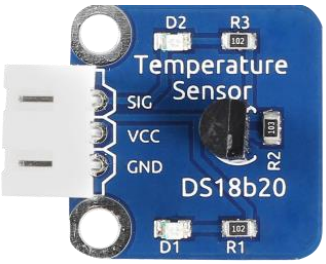
Component List

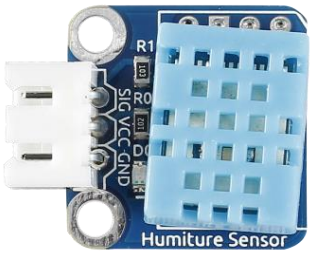
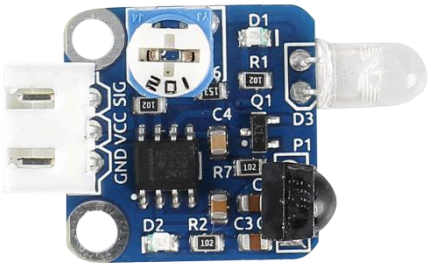
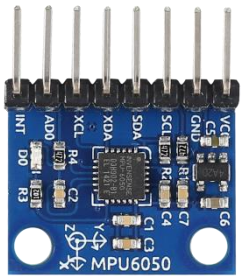
No.	Name	Qty	Component
1	Dual-Color LED	1	
2	RGB LED	1	
3	Auto Flash LED	1	
4	Relay Module	1	
5	Laser Emitter	1	
6	Button	1	

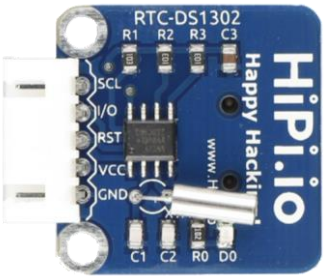
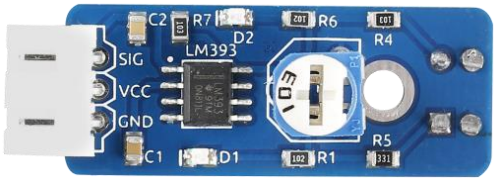
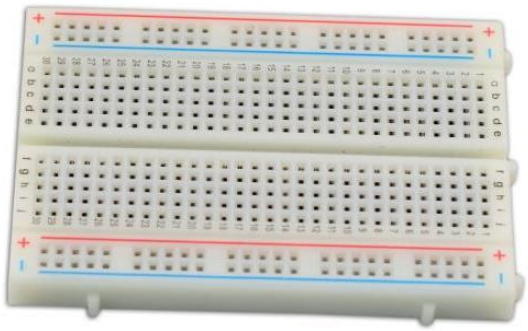
7	Tilt-Switch	1	
8	Vibration Switch	1	
9	Infrared Receiver	1	
10	Active Buzzer	1	
11	Passive Buzzer	1	
12	Reed Switch	1	

13	Photo-interrupter	1	
14	AD/DA Converter PCF8591	1	
15	Raindrop Sensor	1	
16	Joystick PS2	1	
17	Potentiometer	1	
18	Analog Hall Sensor	1	

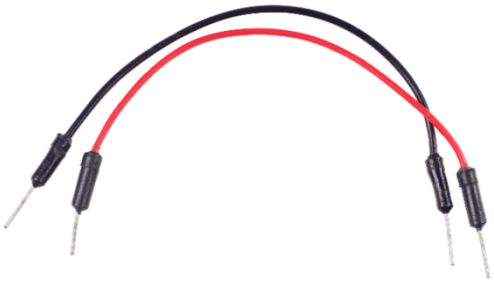
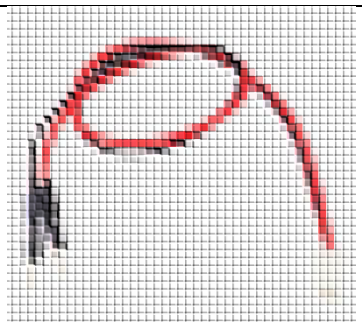
19	Hall Switch Sensor	1	
20	Analog Temperature Sensor	1	
21	Thermistor	1	
22	Sound Sensor	1	
23	Photoresistor	1	
24	Flame Sensor	1	

25	Gas Sensor	1	
26	Remote Control	1	
27	Touch Switch	1	
28	Ultrasonic Sensor	1	
29	Temperature Sensor DS18B20	1	
30	Rotary Encoder	1	

31	Humidity/Temperature "Humiture" Sensor	1	
32	IR Obstacle Module	1	
33	I2C LCD1602 Module	1	
34	Barometer-BMP180	1	
35	MPU6050 Module	1	

36	RTC-DS1302 Module	1	
37	Tracking Sensor	1	
38	Breadboard	1	
39	T-Cobbler	1	
40	40-pin Ribbon Cable for T-Cobbler	1	

41	2-Pin Non-Reversible Cable (Connector to male)	2	
42	3-Pin Non-Reversible Cable (Connector to male)	5	
43	4-Pin Non-Reversible Cable (Connector to male)	5	
44	5-Pin Non-Reversible Cable (Connector to male)	5	
45	Jumper wires (Male to Female)	20	

46	Jumper wires (Male to Male)	10	
47	2-Pin Cable, (Connector to Connector)	1	

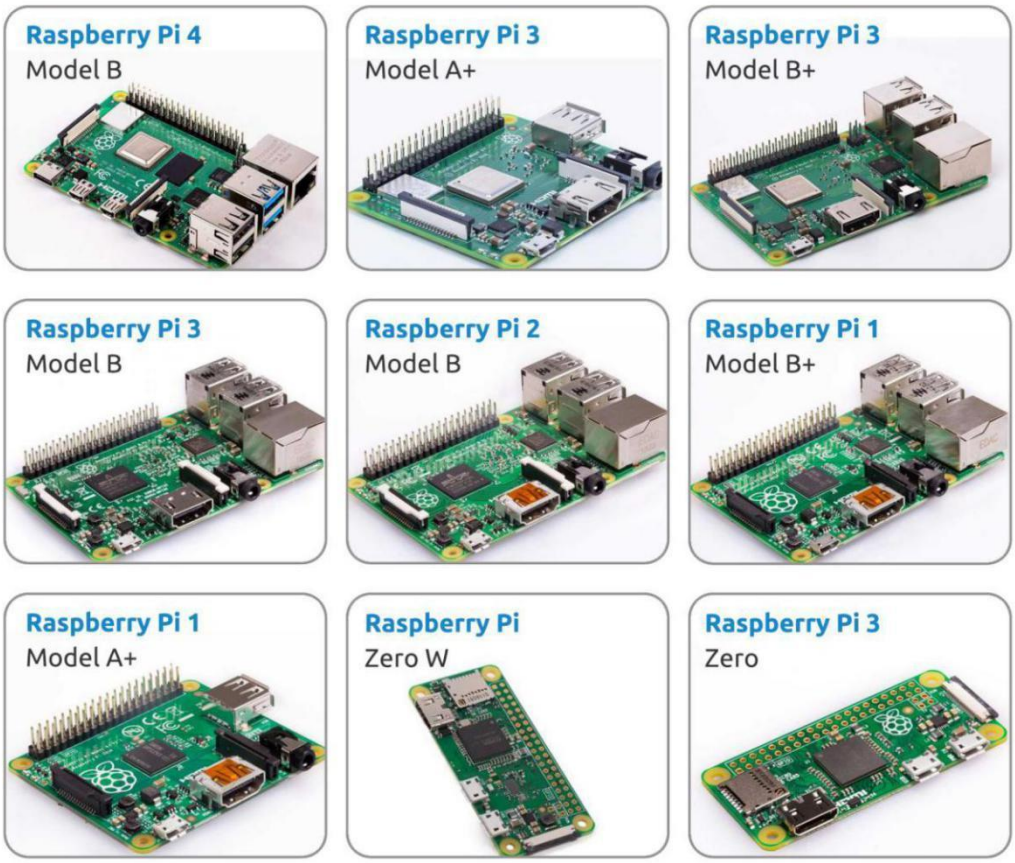
Tools and Materials

Required Components

Raspberry Pi

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.

Our kit is compatible with the following Raspberry Pi Boards:



Power Supply

Not all power supplies will work with the Raspberry Pi (phone chargers for example often don't provide a stable enough voltage). Use a power supply that has proven compatibility with Raspberry Pi boards.

Raspberry Pi 4

To connect to a power socket, the Raspberry Pi 4 uses a USB-C port (the same found on some mobile phones and computers). You will need a power supply which provides at least 3.0 amps.

Raspberry Pi Zero up to 3B+

To connect to a power socket, the Raspberry Pi Zero up to Raspberry Pi 3B+ has a micro USB port (the same found on many mobile phones). You will need a power supply which provides at least 2.5 amps.

Micro SD Card

Your Raspberry Pi needs an SD card to store all its files and the Raspberry Pi OS. You will need a reliable Class 10 micro SD Card that has been tested for use with the Raspberry Pi (8GB capacity minimum).

Optional Components

Screen

Your Raspberry Pi will need a display to show its desktop environment. Any computer monitor or TV will do. If your display comes with built-in speakers your Raspberry Pi will play sound through them.

Mouse & Keyboard

When using your Raspberry Pi with a display, you will need an input device like a USB or Bluetooth keyboard and mouse.

Please note that Raspberry Pi boards in the Zero family use the USB Micro-B standard. A USB-OTG style adapter cable may be required. .

HDMI

The Raspberry Pi has HDMI output which is compatible with most modern TVs and computer monitors. If your screen has only DVI or VGA ports, you will need an appropriate adapter or cable.

Please note that Raspberry Pi boards in the Zero and 4B families require Mini and Micro HDMI cable standards respectively. It is always recommended to choose a cable or adapter that has been tested to work well with Raspberry Pi boards.

Case

A case offers protection from short circuits, and many cases provide methods to mount the case and Pi to a wall, under a table, or on the back of your TV.

Speakers or Headphones

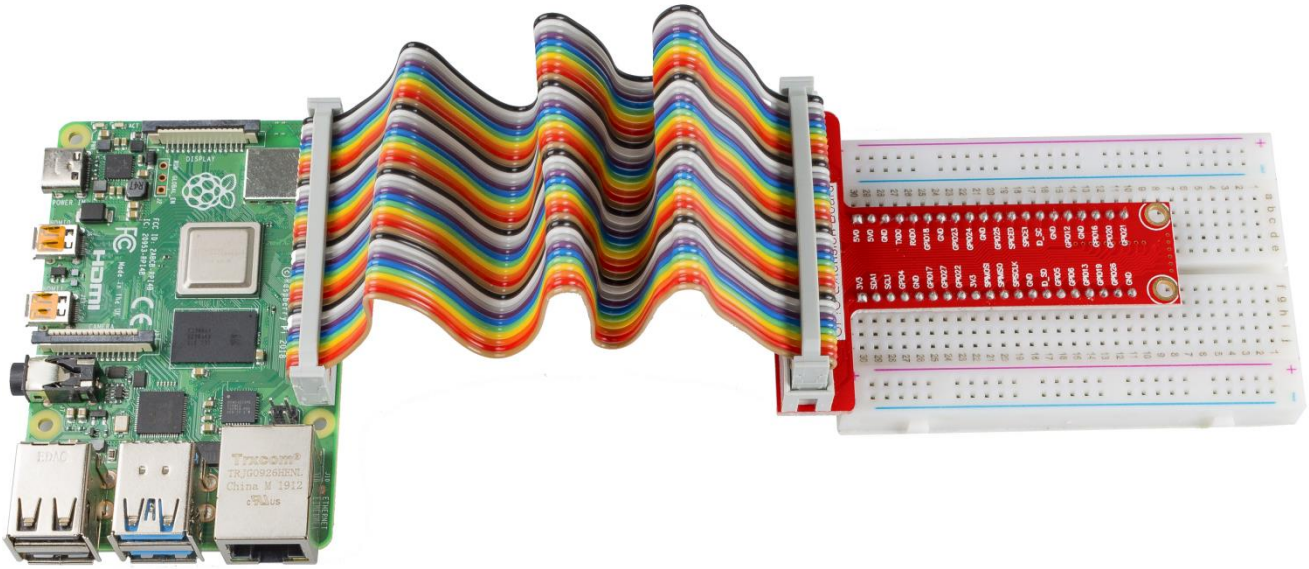
The Raspberry Pi is equipped with 3.5mm audio ports that can be used with most speakers and headphones as an alternative to sound over HDMI. You can switch the preferred sound output device from the sound icon in the top menu of the GUI desktop.

GPIO Extension Board

Connecting to the Raspberry Pi

Before starting out, you'll need to get familiar with the Raspberry Pi's GPIO pins.

These pins can be easily accessed using a breadboard with our GPIO Extension board. Working with the Raspberry Pi in this way can help prevent damage to the board by limiting how often you will be directly in contact with its exposed parts.



Pin Numbers

The Raspberry Pi GPIO pins have three naming conventions; wiringPi, BCM, and Board. The 40-pin GPIO Extension board generally uses the BCM convention, except for certain pins including the I2C and SPI port. The following table will clarify the equivalents between the three conventions.

Note:

1) In the C programming language, the WiringPi convention is used primarily but BCM can be selected by using `wiringPiSetupGpio (void) ;` at the beginning of your program.

2) In Python, both Board and BCM naming is used. The function `GPIO.setmode()` is used to declare them.

Name	WiringPi	Board	BCM		Board	WiringPi	Name
GPIO Extension Board							
3.3V	3V3	1	3V3	5.0V	2	5.0V	5V
SDA	8	3	SDA	5.0V	4	5.0V	5V
SCL	9	5	SCL	GND	6	GND	0V
GPIO7	7	7	GPIO4	TXD	8	15	TXD
0V	GND	9	GND	RXD	10	16	RXD
GPIO0	0	11	GPIO17	GPIO18	12	1	GPIO1
GPIO2	2	13	GPIO27	GND	14	GND	0V
GPIO3	3	15	GPIO22	GPIO23	16	4	GPIO4
3.3V	3.3V	17	3.3V	GPIO24	18	5	GPIO5
MOSI	12	19	MOSI	GND	20	GND	0V
MISO	13	21	MISO	GPIO25	22	6	GPIO6
SCLK	14	23	SCLK	CE0	24	10	CEO
0V	GND	25	GND	CE1	26	11	CE1
IN_SDA	30	27	EED	EEC	28	31	ID_SCL
GPIO21	21	29	GPIO5	GND	30	GND	0V
GPIO22	22	31	GPIO6	GPIO12	32	26	GPIO26
GPIO23	23	33	GPIO13	GND	34	GND	0V
GPIO24	24	35	GPIO19	GPIO16	36	27	GPIO27
GPIO25	25	37	GPIO26	GPIO20	38	28	GPIO28
0V	GND	39	GND	GPIO21	40	29	GPIO29

Quick Start

Get the pre-configured image

You can download the latest pre-configured Raspberry OS image (with all libraries installed) here: https://HiPi.io/sk4/downloads

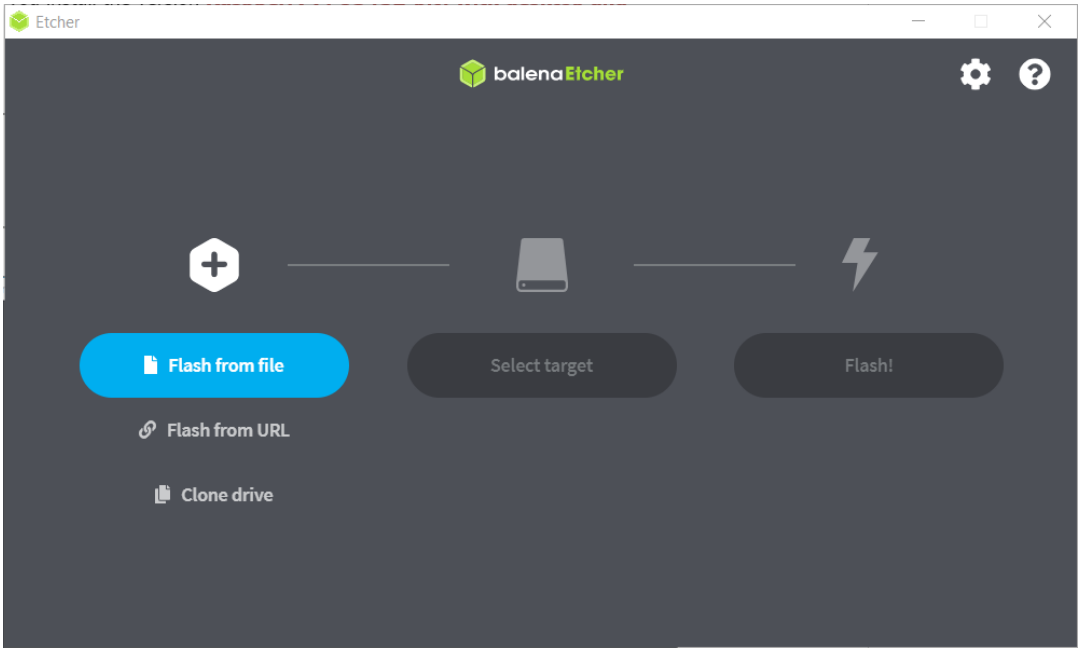
To write this image to your micro SD card, please follow the **Balena Etcher** section

Using Balena Etcher

Step 1: Download the latest version of **Balena Etcher** from https://www.balena.io/etcher/ and install it.

Step 2: Choose the image file you downloaded.

Step 3: Plug the USB card reader into the computer, then start Etcher.



Step 3.1: Select the image file you downloaded in Step 2.

Step 3.2: Select the SD card you want to write to. Everything will be erased!!!

Step 3.3: Click Flash. Etcher will write the image and verify it.

If you are not planning to use the preconfigured image and want to prepare your Raspberry OS image from scratch, please follow the guide in the **Appendix: Pi preparation** at the end of this manual.

Connect to your Pi

Get the IP Address

After the Raspberry Pi is connected to WI-FI, we need to get its IP address to connect to it. There are many ways to find the IP address, and two of them are listed below:

Checking via the router

If you have permission to log in the router (such as a home network), you can check the addresses assigned to Raspberry Pi on the admin interface of the router.

The default hostname of the Raspberry Pi OS is **raspberrypi**, search for it and note the IP address.

Network Segment Scanning

You can also use network scanning to look up the IP address of Raspberry Pi. Software like Advanced IP scanner can be used for this purpose.

Scan the IP range used by your network and the network name of all connected devices will be displayed. The default hostname of the Raspberry Pi OS is **raspberrypi**, search for it and note the IP address.

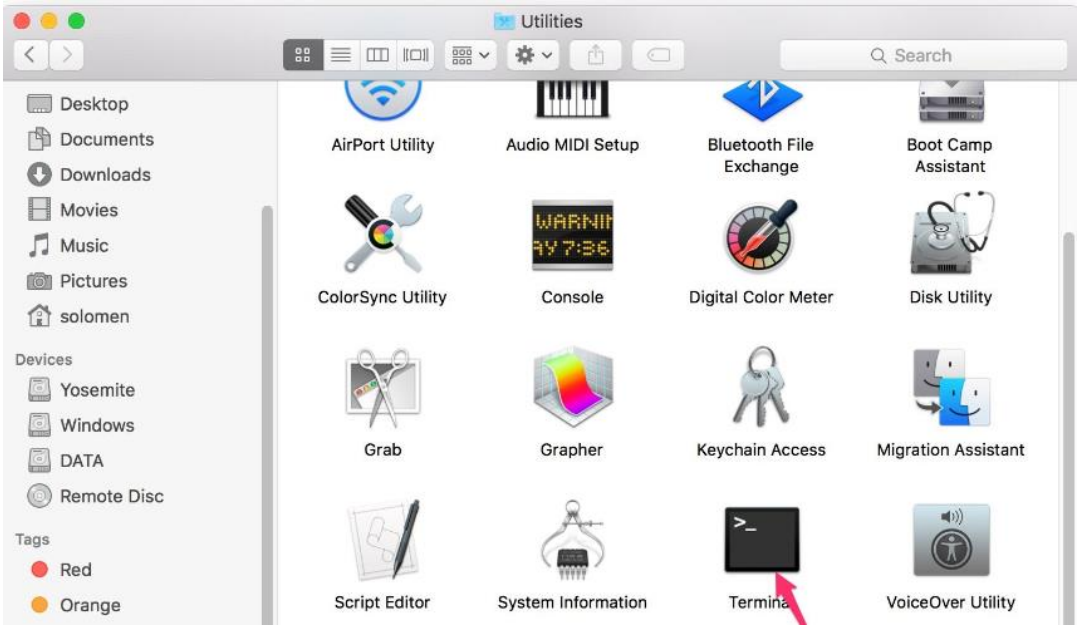
Using SSH for Remote Control

We can access the Bash Shell of Raspberry Pi by connecting with SSH. Bash is the default shell of Linux (Raspberry Pi OS is based on Linux). The Shell is a program that allows the user to send commands to Unix/Linux and see the result. It helps you get work done!

For Linux or/Mac OS X Users

Step 1

Go to **Applications**->**Utilities**, find the **Terminal**, and open it.



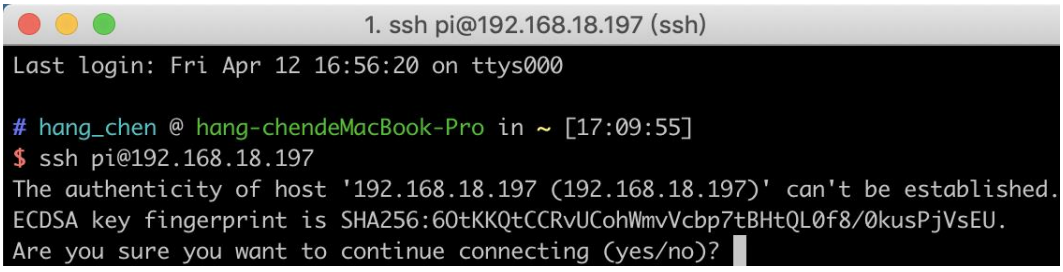
Step 2

Type in "**`ssh pi@ip_address`**" where "pi" is the username and "ip_address" is the IP address you noted earlier. For example:

```
ssh pi@192.168.18.197
```

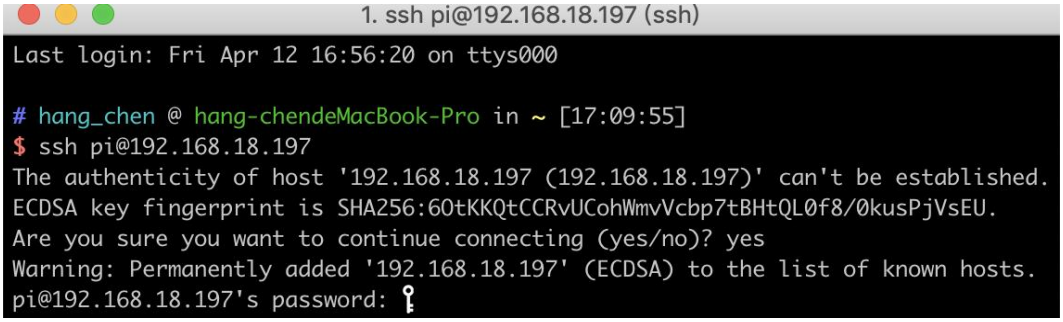
Step 3

Input "yes".



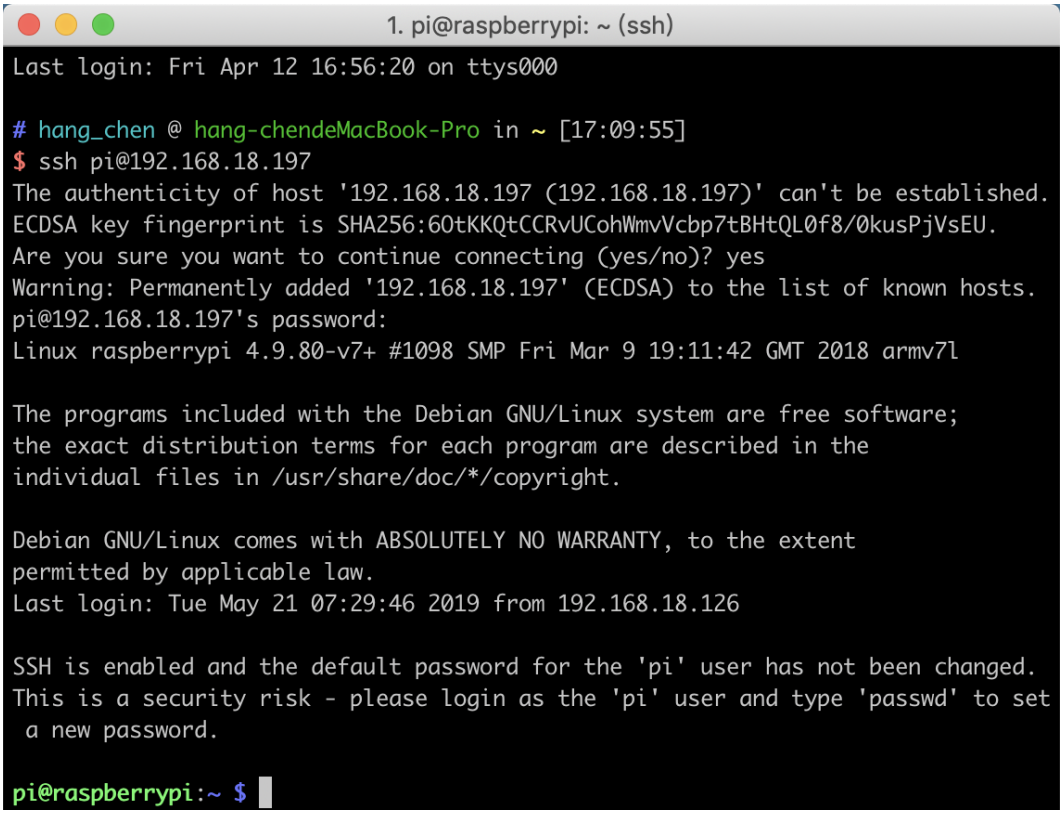
Step 4

Type in password ("**raspberry**" is the default for user "pi").



Step 5

Raspberry Pi is now connected and you are ready for the next steps!



Note:	When you type the password, the characters will not be visible, this is normal.

For Windows Users

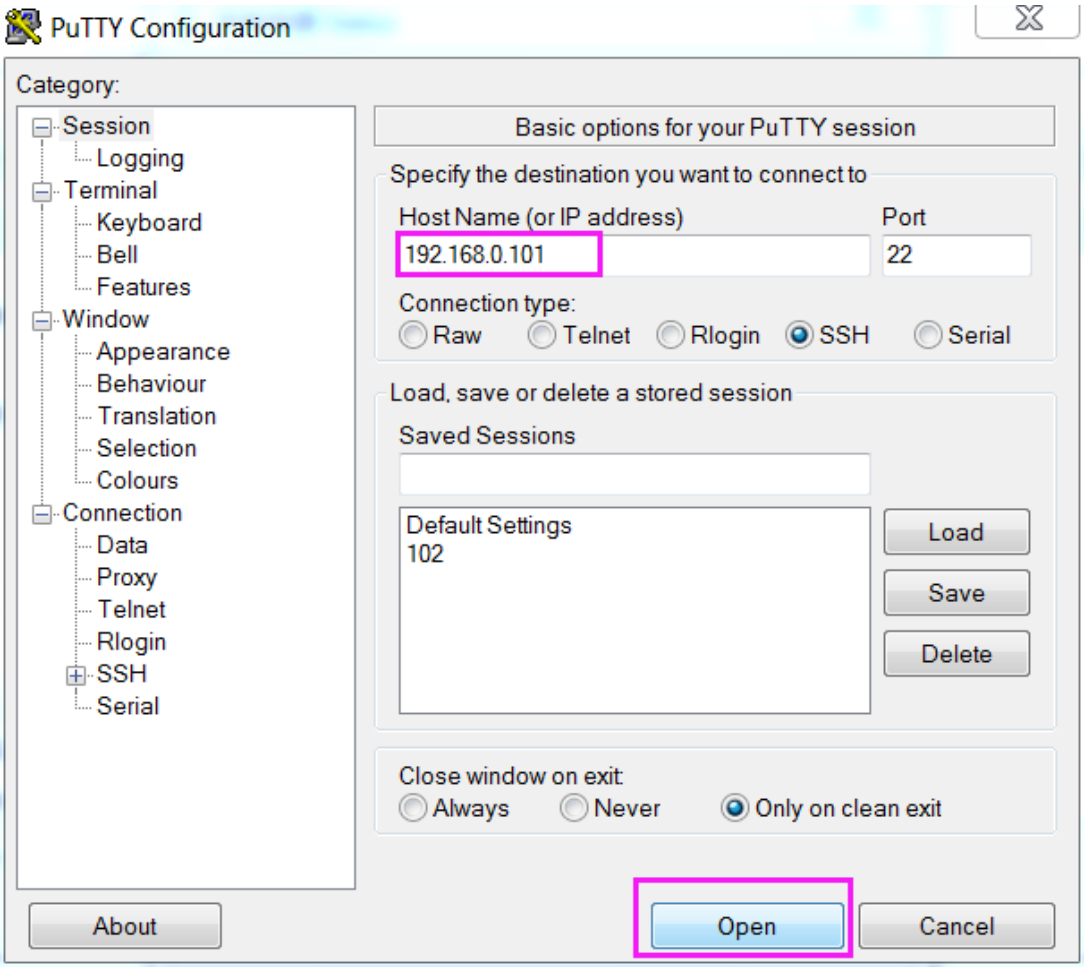
If you're a Windows user, you can use SSH with the application of some software. Here, we recommend PuTTY.

Step 1

Download PuTTY.exe from https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html. We recommend the 32-bit version.

Step 2

Open PuTTY and click **Session** on the left tree-alike structure. Enter the IP address you noted earlier in the text box under **Host Name (or IP address)** and 22 under **Port** (by default it is 22).

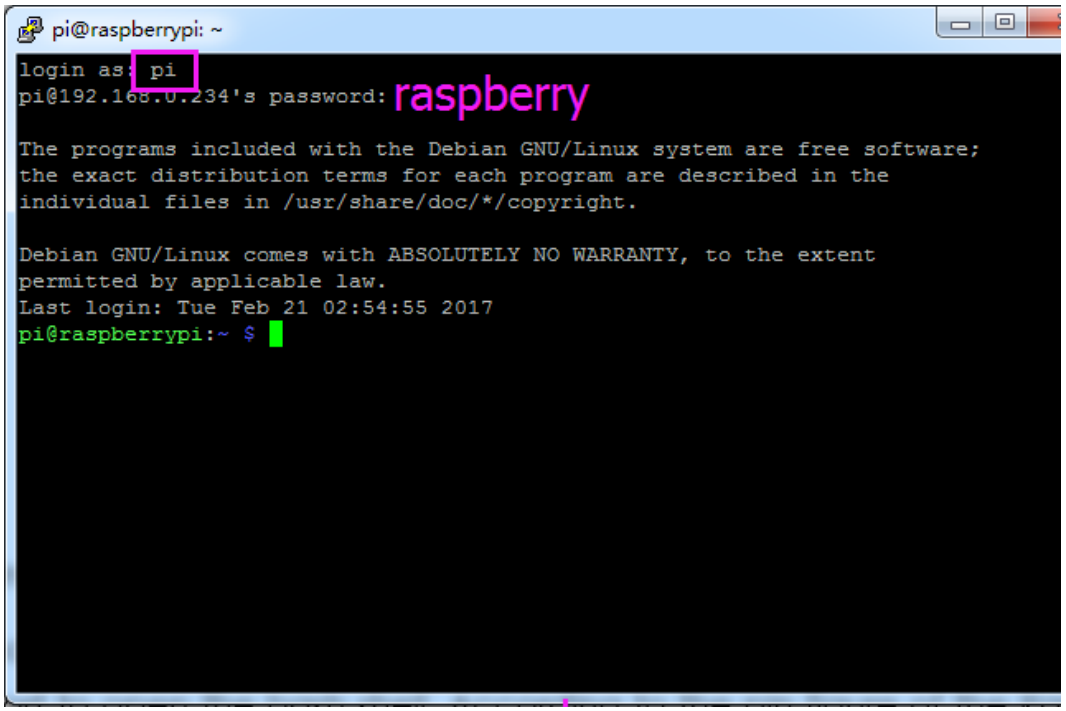


Step 3

Click **Open**. Note that when you first connect to the Raspberry Pi with PuTTY, will present a security prompt. To accept, click **Yes**.

Step 4

When PuTTY connects, the prompt "**login as:**" appears. Type in the user name ("**pi**" is the default) and the password ("**raspberry**" is the default for the user "pi").



Note:	When you type the password, the characters will not be visible, this is normal.

Step 5

Raspberry Pi is now connected and you are ready for the next steps!

Using the Remote Desktop

If you don't like to use the terminal window to control your Raspberry Pi, you can also use the remote desktop function. It is easier to use and managing files in the Raspberry Pi becomes as easy as on a Windows PC or Mac. There are two ways to control the desktop of the Raspberry Pi remotely: **VNC** and **XRDP**.

VNC

Here is how to enable the remote desktop access through VNC:

Enable VNC service

The VNC service is installed in the system, but by default, VNC is disabled. You need to enable it.

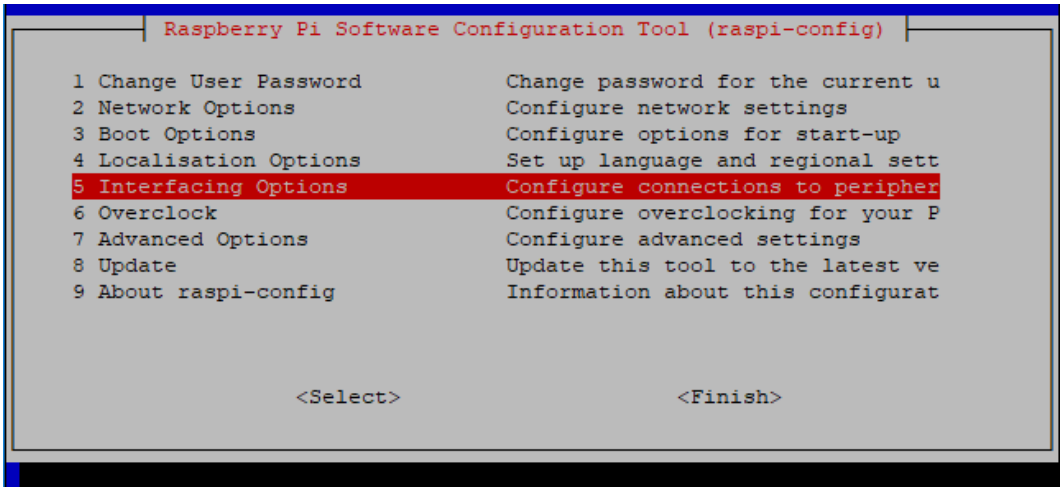
Step 1

Type the following command:

```
sudo raspi-config
```

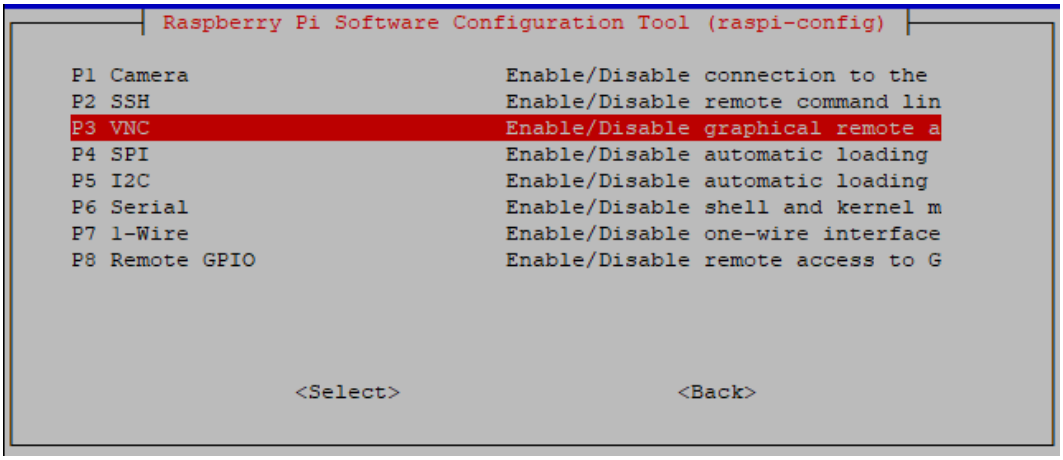
Step 2

On the configuration interface, select "**Interfacing Options**" by the forward and backward keys.



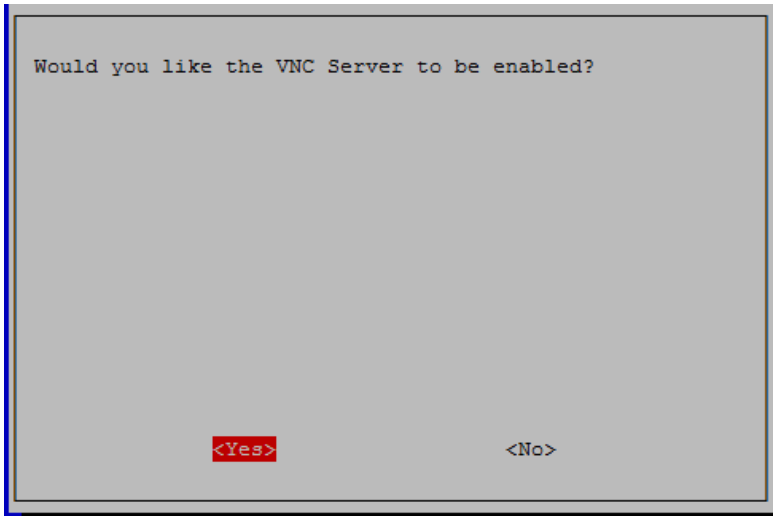
Step 3

Select **VNC**.



Step 4

Select **Yes -> OK -> Finish** to exit the configuration.



Login to VNC

Step 1

Now, it's time to install the VNC Viewer on your computer. From the link below, download the version appropriate to your computer:

https://www.realvnc.com/en/raspberrypi/

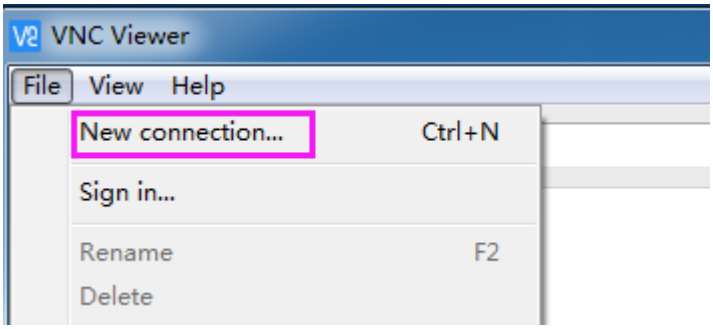
https://www.realvnc.com/en/connect/download/viewer/

Step 2

Once the download is finished, execute the installation file and follow the instructions.

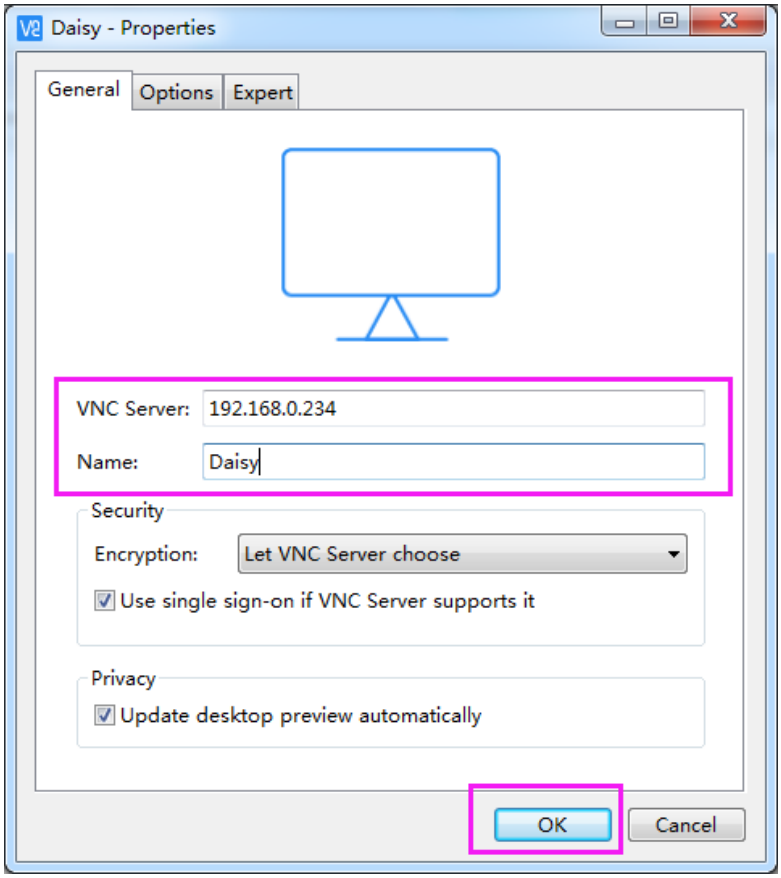
Step 3

Open VNC Viewer, then select "**New connection**".



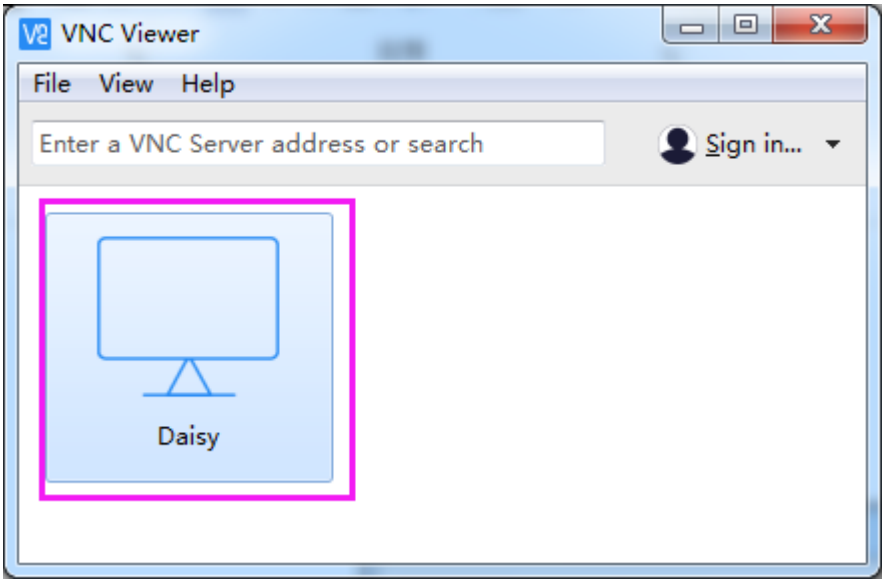
Step 3

In the first box, type the IP address of Raspberry Pi you noted earlier and type a name for the connection in **name**. It can be almost any name you want.



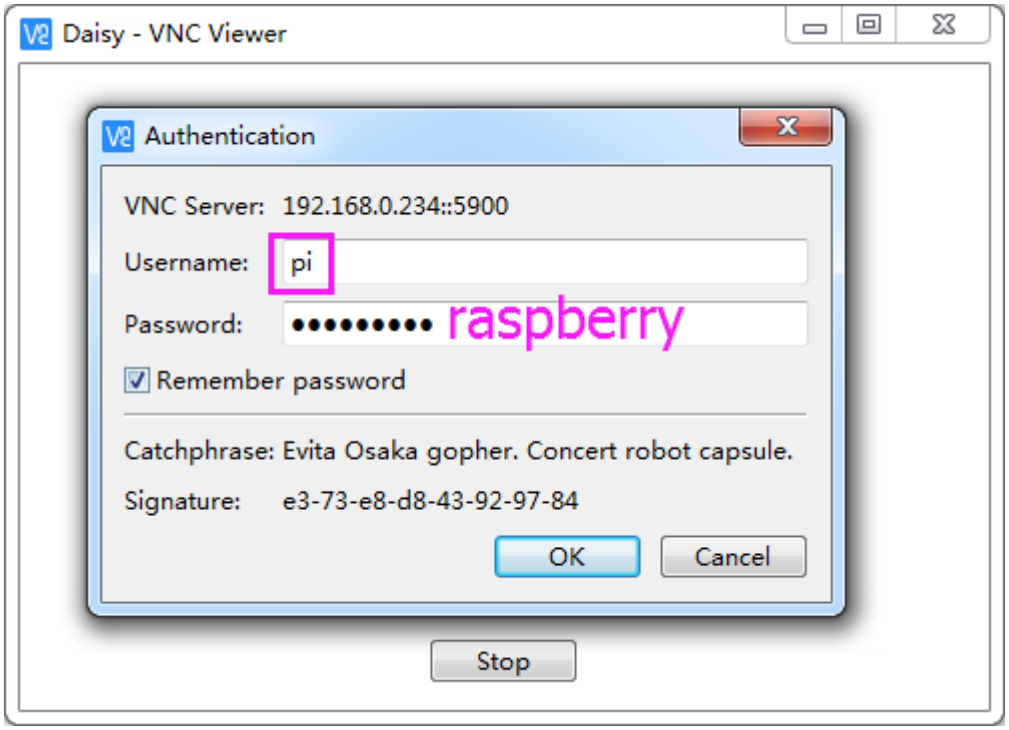
Step 4

Double click the **connection** just created:



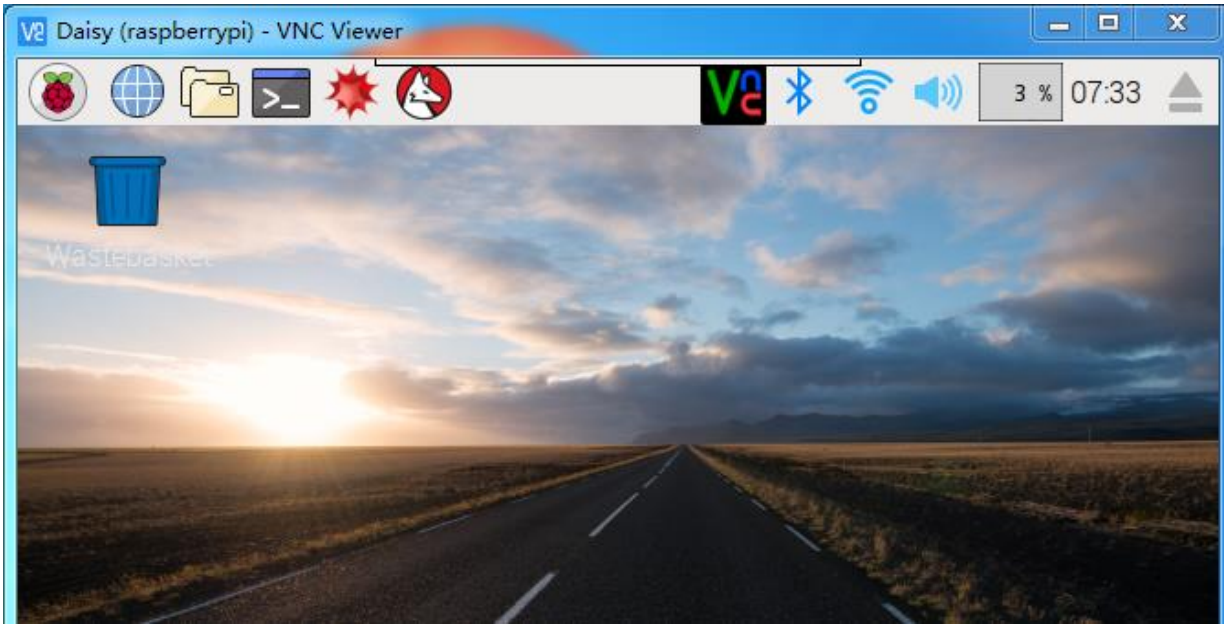
Step 5

Enter Username (**pi**) and Password (**raspberry** by default) and click OK.



Step 6

Now you can see the desktop of the Raspberry Pi:



Step 7

Raspberry Pi is now connected and you are ready for the next steps!

XRDP

XRDP provides a graphical interface to connect to remote machines using the Microsoft Remote Desktop Protocol.

Install XRDP

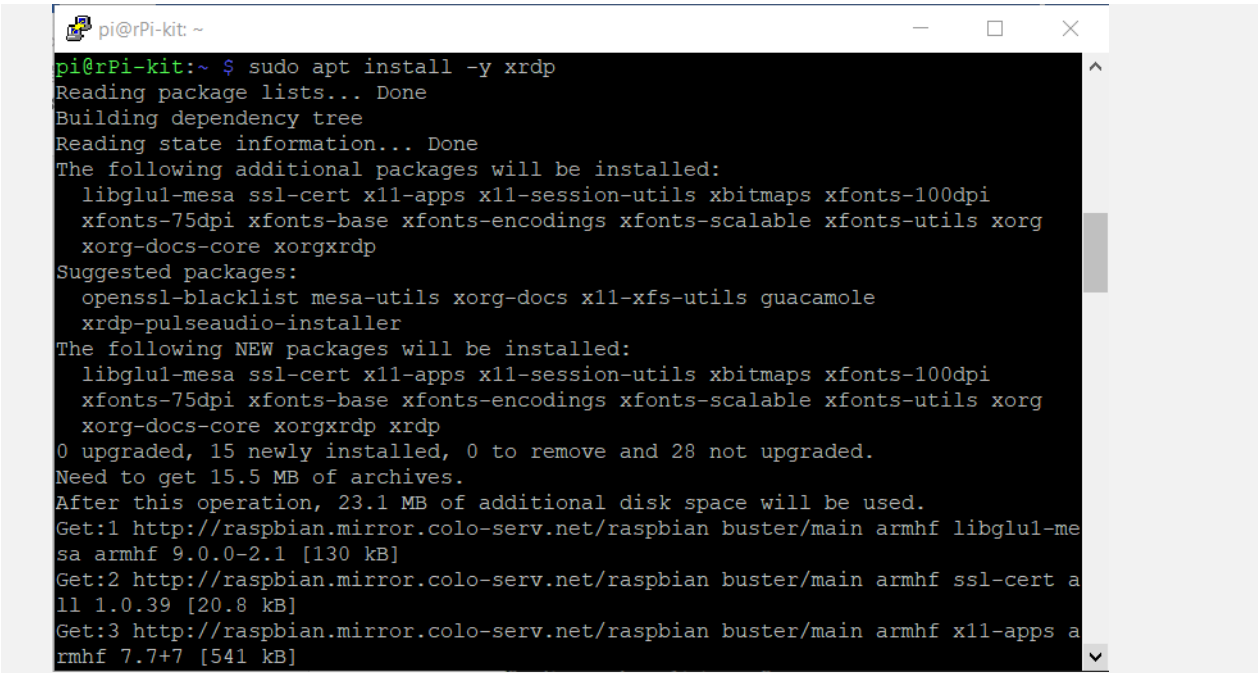
Step 1

Login to Raspberry Pi by using SSH.

Step 2

Type the following commands to install XRDP:

```
sudo apt update
sudo apt install -y xrdp
```



Step 3

After the installation is completed, you are ready to use Windows remote desktop applications to login to your Raspberry Pi!

Login to XRDP

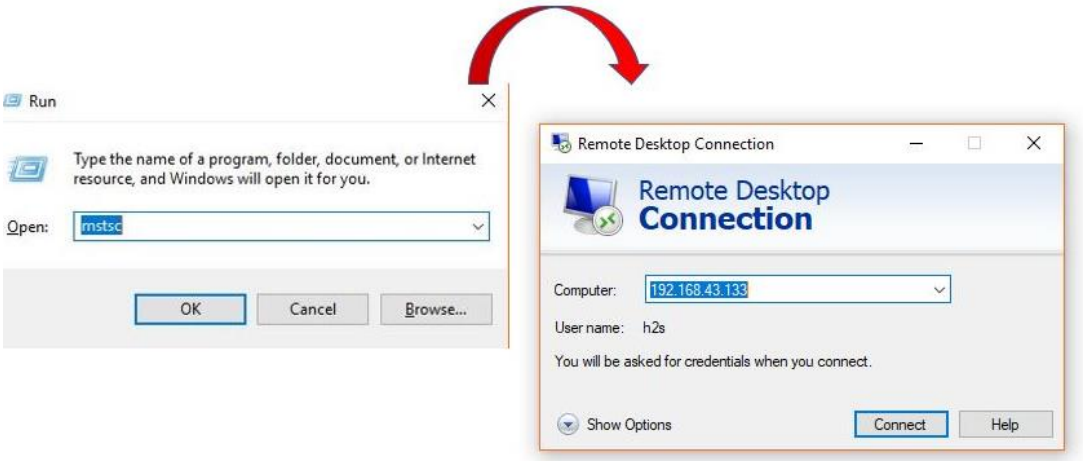
Step 1

- If you are a Windows user, you can use the Remote Desktop feature that comes with Windows.
- If you are a Mac user, you can download and use Microsoft Remote Desktop from the APP Store. There is not much difference between the two.

In the steps below, we will use Windows remote desktop.

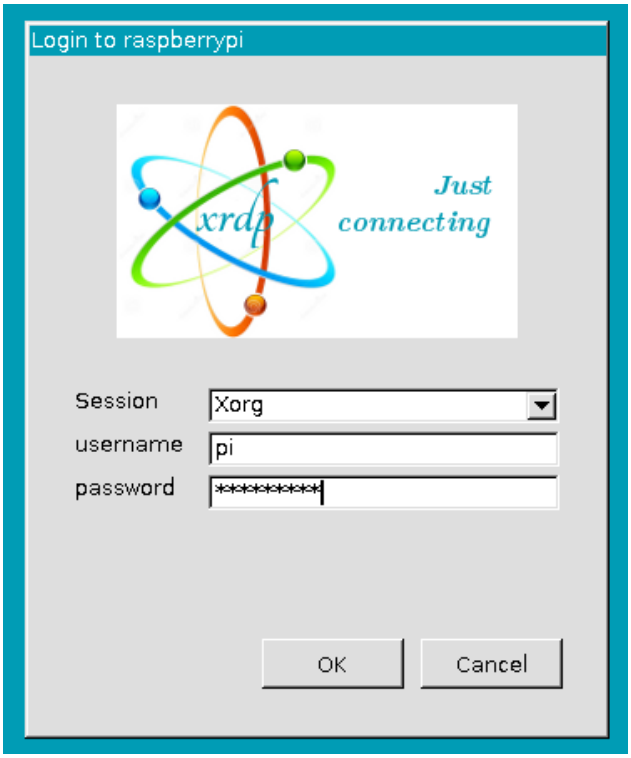
Step 2

Press the (WIN+R) keys to open the Run dialog box and type in "mstsc" to open Remote Desktop Connection. Type the IP address of your Raspberry Pi, then click on "Connect". If a warning screen appears, click yes to accept.



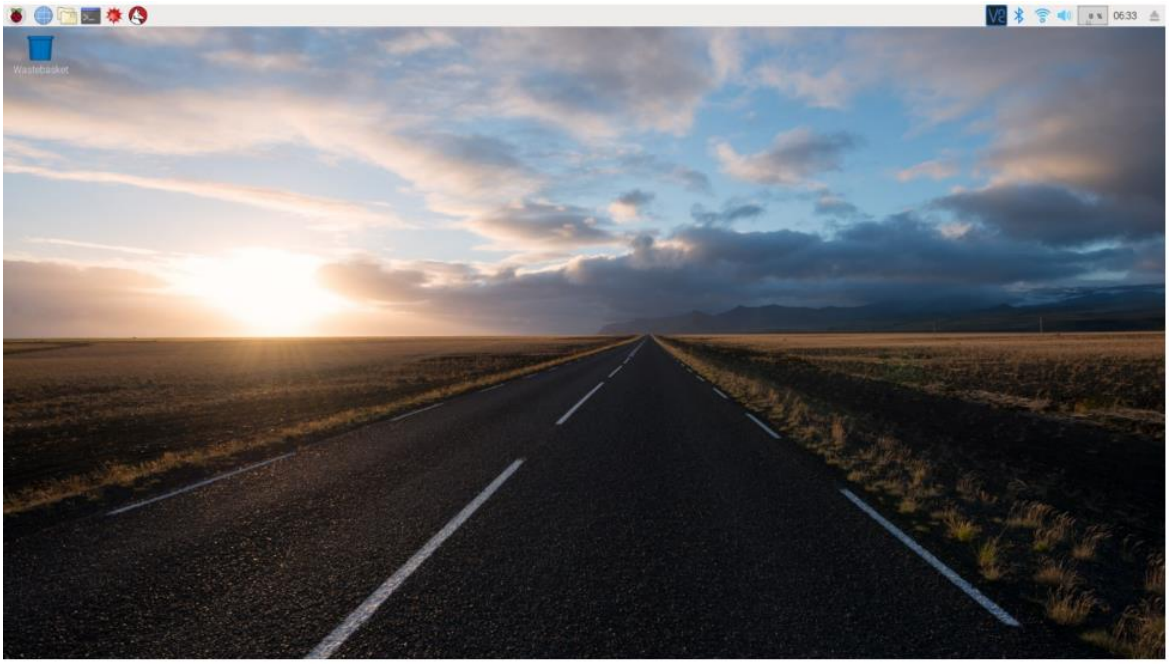
Step 3

The **xrdp** login screen will appear. Enter the user name and password of your Raspberry Pi and click OK. Remember, by default the user name of Raspberry Pi is **"pi"** and the password is **"raspberry"**.



Step 4

You successfully logged in to your Raspberry Pi by using remote desktop!



Step 5

Raspberry Pi is now connected and you are ready for the next steps!

Libraries

The two important libraries that are used in programming with the Raspberry Pi are wiringPi and RPI.GPIO. They are installed and ready to use on the Raspberry Pi OS operating system by default

RPi.GPIO

In Python, the GPIO pins are programmed with the API provided by RPi.GPIO. For examples and documents, visit http://sourceforge.net/p/raspberry-gpio-python/wiki/Home/

To see which version of python your Raspberry Pi is using, and test whether RPI.GPIO is installed, type the following commands in the terminal:

```
python
```

```
pi@raspberrypi:~ $ python
Python 2.7.9 (default, Mar  8 2015, 00:52:26)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

```
import RPi.GPIO
```

```
pi@raspberrypi:~ $ python
Python 2.7.9 (default, Mar  8 2015, 00:52:26)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import RPi.GPIO
>>>
```

If no error appears, RPi.GPIO is installed. Type in the following to quit:

```
exit()
```

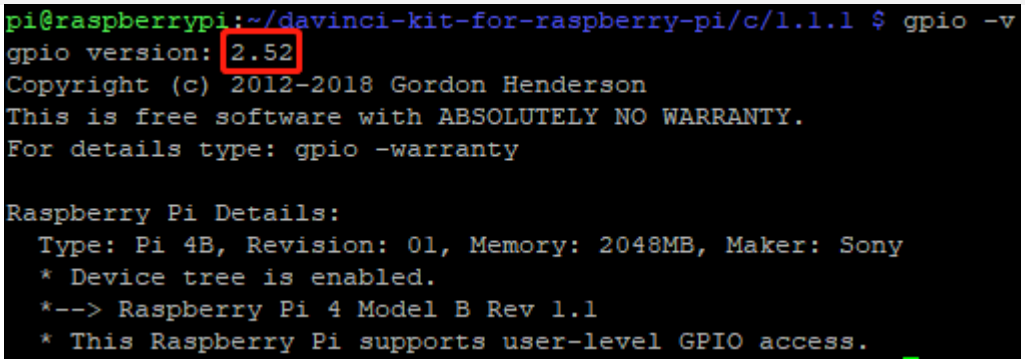
```
>>> exit()
pi@raspberrypi:~ $
```

WiringPi

wiringPi is a GPIO library for the C language that works with the Raspberry Pi (GNU Lv3 compliant). Arduino users will be able to become familiar with wiringPi more easily as it uses similar functions.

Commands in wiringPi are available to control the Raspberry Pi's GPIO pins and allow interfacing with many devices. To test whether the library is installed, type the following commands in the terminal:

```
gpio -v
```



Note: If you are using Raspberry Pi 4B, but the GPIO version is **2.50**, it will cause no response after the C language code is running, that is, GPIO pins do not work.

To fix this, you need to manually update to version **2.52** by following these steps:

In terminal, type:

```
cd /tmp
wget https://project-downloads.drogon.net/wiringpi-latest.deb
sudo dpkg -i wiringpi-latest.deb
```

Check with:

```
gpio -v
```

and make sure it's version 2.52.

A message should appear indicating the gpio version and Copyright notes.

To read the status of the Raspberry Pi's pins, use:

```
gpio readall
```

```
pi@raspberrypi:~ $ gpio readall
 +-----+-----+---------+------+---+---Pi 3---+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
 |   2 |   8 |   SDA.1 | ALT0 | 1 |  3 || 4  |   |      | 5V      |     |     |
 |   3 |   9 |   SCL.1 | ALT0 | 1 |  5 || 6  |   |      | 0v      |     |     |
 |   4 |   7 | GPIO. 7 |   IN | 0 |  7 || 8  | 1 | IN   | TxD     | 15  | 14  |
 |     |     |      0v |      |   |  9 || 10 | 1 | IN   | RxD     | 16  | 15  |
 |  17 |   0 | GPIO. 0 |   IN | 0 | 11 || 12 | 0 | IN   | GPIO. 1 | 1   | 18  |
 |  27 |   2 | GPIO. 2 |   IN | 0 | 13 || 14 |   |      | 0v      |     |     |
 |  22 |   3 | GPIO. 3 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO. 4 | 4   | 23  |
 |     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 | 5   | 24  |
 |  10 |  12 |    MOSI | ALT0 | 1 | 19 || 20 |   |      | 0v      |     |     |
 |   9 |  13 |    MISO | ALT0 | 1 | 21 || 22 | 0 | IN   | GPIO. 6 | 6   | 25  |
 |  11 |  14 |    SCLK | ALT0 | 0 | 23 || 24 | 1 | OUT  | CE0     | 10  | 8   |
 |     |     |      0v |      |   | 25 || 26 | 1 | OUT  | CE1     | 11  | 7   |
 |   0 |  30 |   SDA.0 |   IN | 1 | 27 || 28 | 1 | OUT  | SCL.0   | 31  | 1   |
 |   5 |  21 | GPIO.21 |   IN | 0 | 29 || 30 |   |      | 0v      |     |     |
 |   6 |  22 | GPIO.22 |   IN | 0 | 31 || 32 | 0 | IN   | GPIO.26 | 26  | 12  |
 |  13 |  23 | GPIO.23 |   IN | 1 | 33 || 34 |   |      | 0v      |     |     |
 |  19 |  24 | GPIO.24 |   IN | 0 | 35 || 36 | 0 | IN   | GPIO.27 | 27  | 16  |
 |  26 |  25 | GPIO.25 |   IN | 0 | 37 || 38 | 0 | IN   | GPIO.28 | 28  | 20  |
 |     |     |      0v |      |   | 39 || 40 | 0 | IN   | GPIO.29 | 29  | 21  |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+---Pi 3---+---+------+---------+-----+-----+
```

For more details about wiringPi, you can refer to: http://wiringpi.com/download-and-install/

Download the Code

First, change directory to your home folder:

```
cd ~
```

Note:	cd, short for change directory, is to change to the intended directory from the current path. Here, we go to your home directory (usually /home/pi/) by using the shortcut "~".

Next, we need to download the examples we will need from the HiPi.io code repository. There are two ways:

Git method

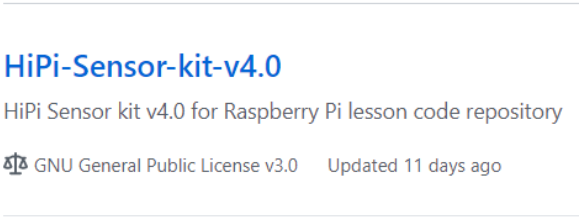
Clone the repository from GitHub (C code and python code) by typing the following command:

```
git clone https://github.com/hipi-io/HiPi-Sensor-kit-v4.0
```

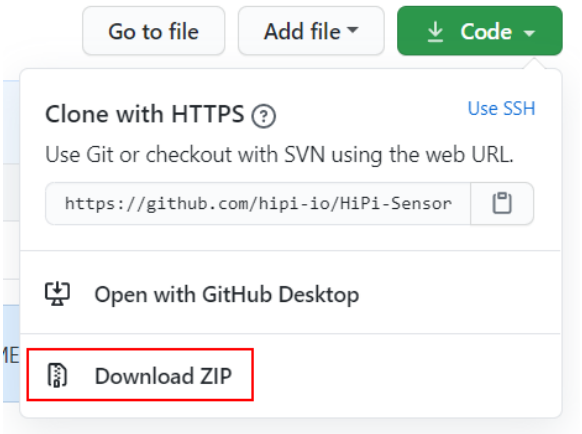
The benefit of using **git** is that you can download the latest code in your folder easily.

Zip file method

You can also type https://www.hipi.io/sk4/repo in the address bar of your web browser and find the code repository for the Sensor Kit on the destination page.



Click on the repository. On the page directed, click **Clone or download** on the right side.



After downloading, type the following commands:

```
   cd ~
   mv ~/Downloads/HiPi-Sensor-kit-v4.0-HiPi.io.zip ~/HiPi-Sensor-kit-
v4.0.zip
   unzip HiPi-Sensor-kit-v4.0.zip
```

The files are now in the folder **~/HiPi-Sensor-kit-v4.0,** ready for you to experiment!

Lesson 1 Dual-Color LED

Introduction

A dual-color light emitting diode (LED) is capable of emitting two different colors of light, typically red and green. Housed in a 3mm or 5mm epoxy package, the module has 3 leads; common cathode or common anode are available. This dual-color LED features two LED terminals (or pins), arranged in the circuit in anti-parallel and connected by a cathode/anode. A positive voltage can be directed to one of the LED terminals, which will cause light of the corresponding color to be emitted. In a dual-color LED, only one of the pins can receive voltage at a time, so this type of LED is frequently used as indicator lights in a variety of devices, including televisions, digital camera, and remote controls.

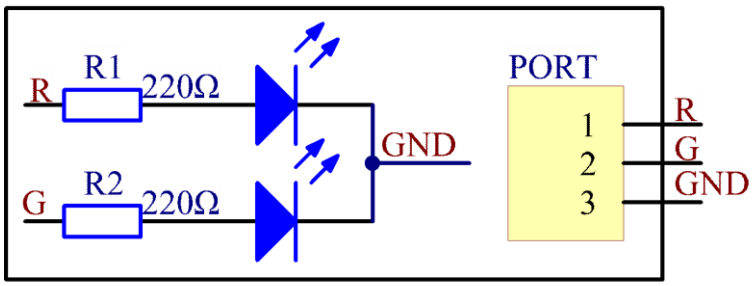
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Dual-color LED module
- 1 * 3-Pin Non-Reversible Cable
- Jumper Wires

Purpose

The module is connected to the Raspberry Pi's GPIO pins and programmed to change the LED's from red to green and then mixed colors (using PWM).

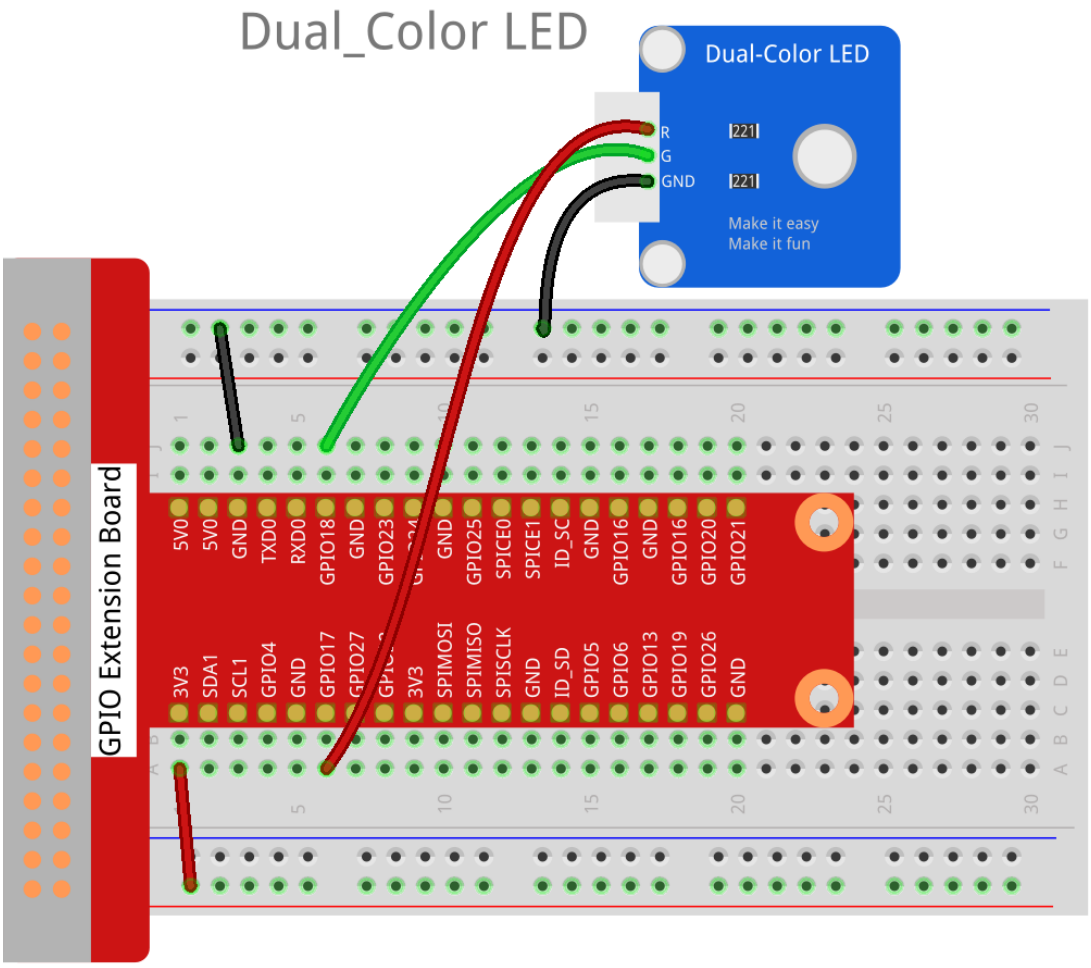
The schematic diagram of the module is as shown below:



Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Dual-Color LED Module
GPIO0	GPIO17	R
GND	GND	GND
GPIO1	GPIO18	G



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/01_dule_color_led/
```

Step 3: Compile.

```
gcc dule_color_led.c -lwiringPi -lpthread
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

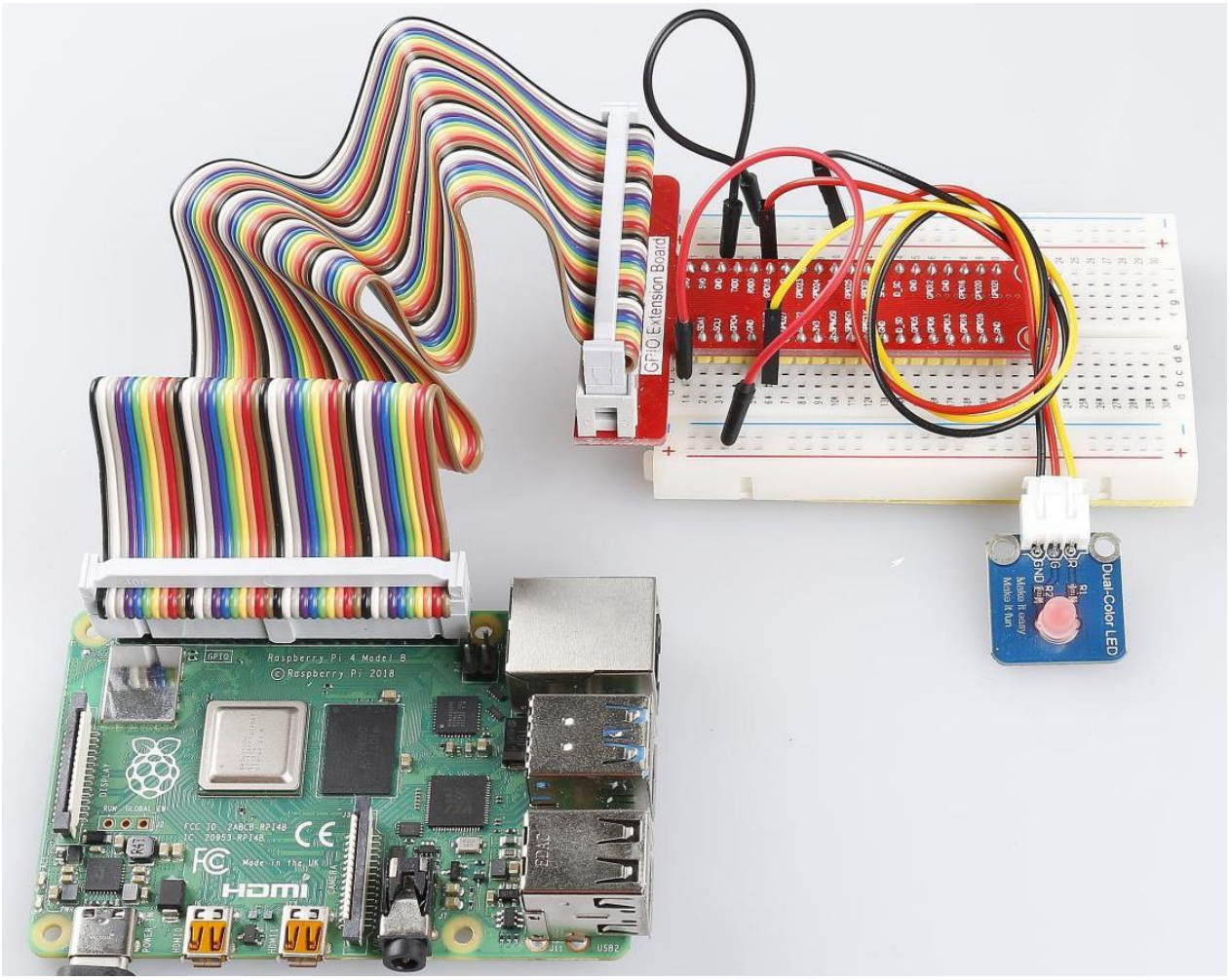
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 01_dule_color_led.py
```

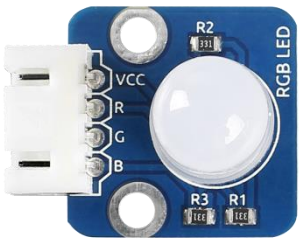
You can see the dual-color LED render green, red, and mixed colors.



Lesson 2 RGB LED Module

Introduction

RGB LED modules are comprised of three LEDs (red, green, and blue) bundled together into a transparent or translucent plastic shell with four pin lead out. The three colors of LED can be mixed to emit light of various colors by controlling the circuit.



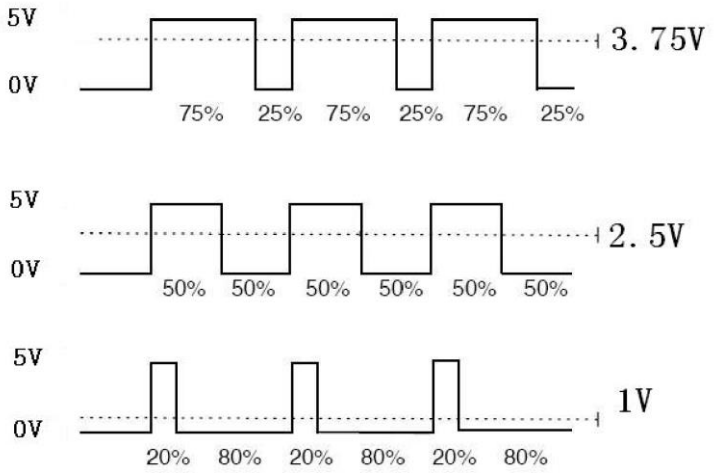
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * RGB LED module
- 1 * 4-Pin Non-Reversible Cable
- Jumper Wires

Purpose

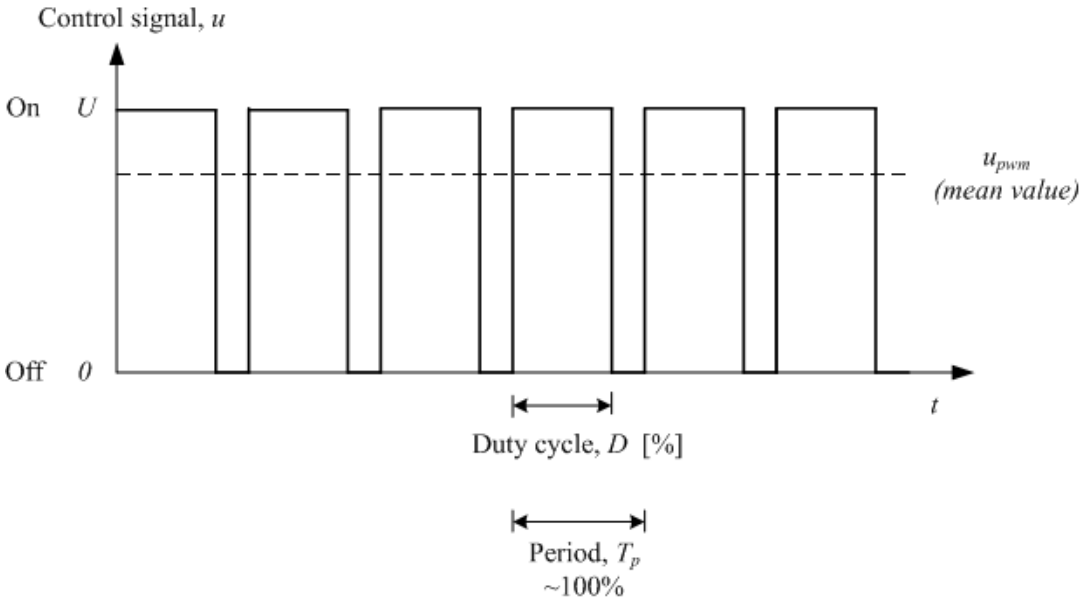
In this experiment, we will use PWM technology to control the brightness of RGB.

Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.



We can see from the top oscillogram that the amplitude of DC voltage output is 5V. However, the actual voltage output is only 3.75V through PWM, for the high level only takes up 75% of the total voltage within a period.

Here are the three basic parameters of PWM:

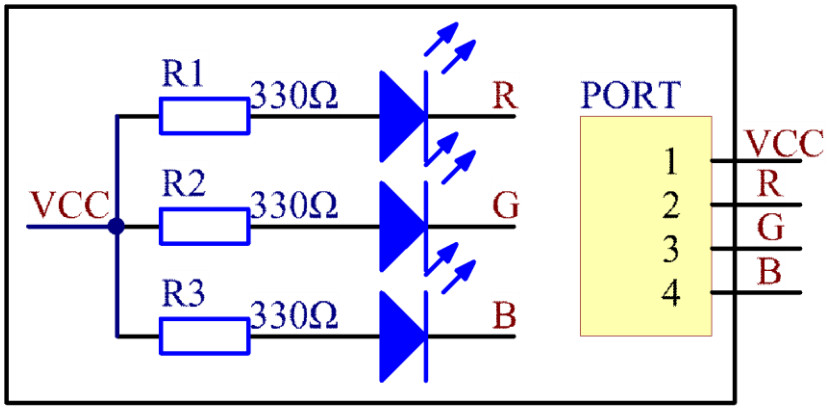


1. The term **duty cycle** describes the proportion of "on" time to the regular interval or "period" of time
2. **Period** describes the reciprocal of pulses in one second.
3. The voltage **amplitude** here is 0V-5V.

Here we input any value between 0 and 255 to the three pins of the RGB LED to make it display different colors.

RGB LEDs can be categorized into common anode LED and common cathode LED. In this experiment, we use a common cathode RGB LED.

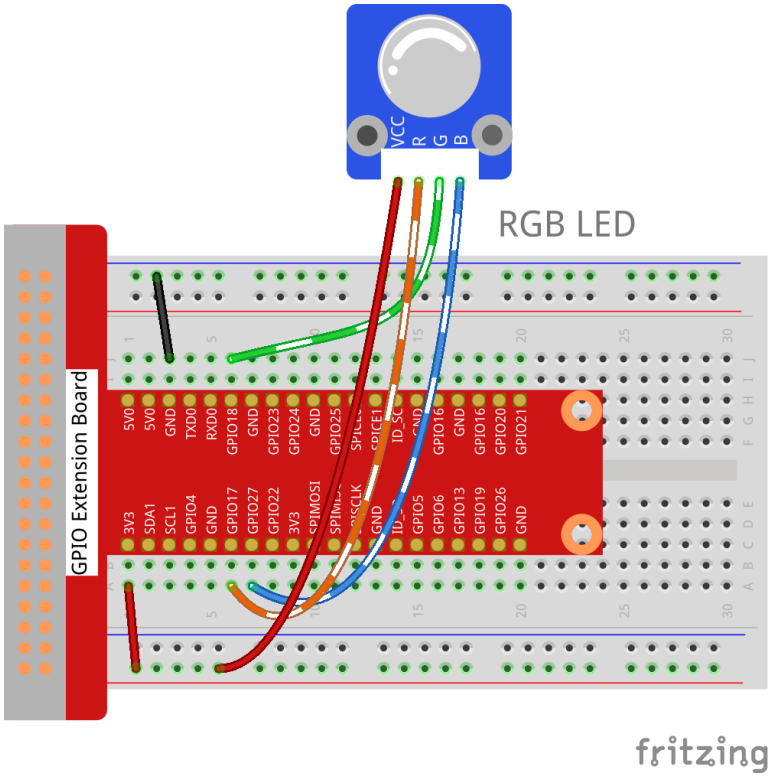
The schematic diagram of the module is as shown below:



Procedure

Step 1: Build the circuit according to the following method.

Raspberry Pi	GPIO Extension Board	RGB LED Module
3.3V	3V3	VCC
GPIO0	GPIO17	R
GPIO1	GPIO18	G
GPIO2	GPIO27	B



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/02_rgb_led/
```

Step 3: Compile.

```
gcc rgb_led.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

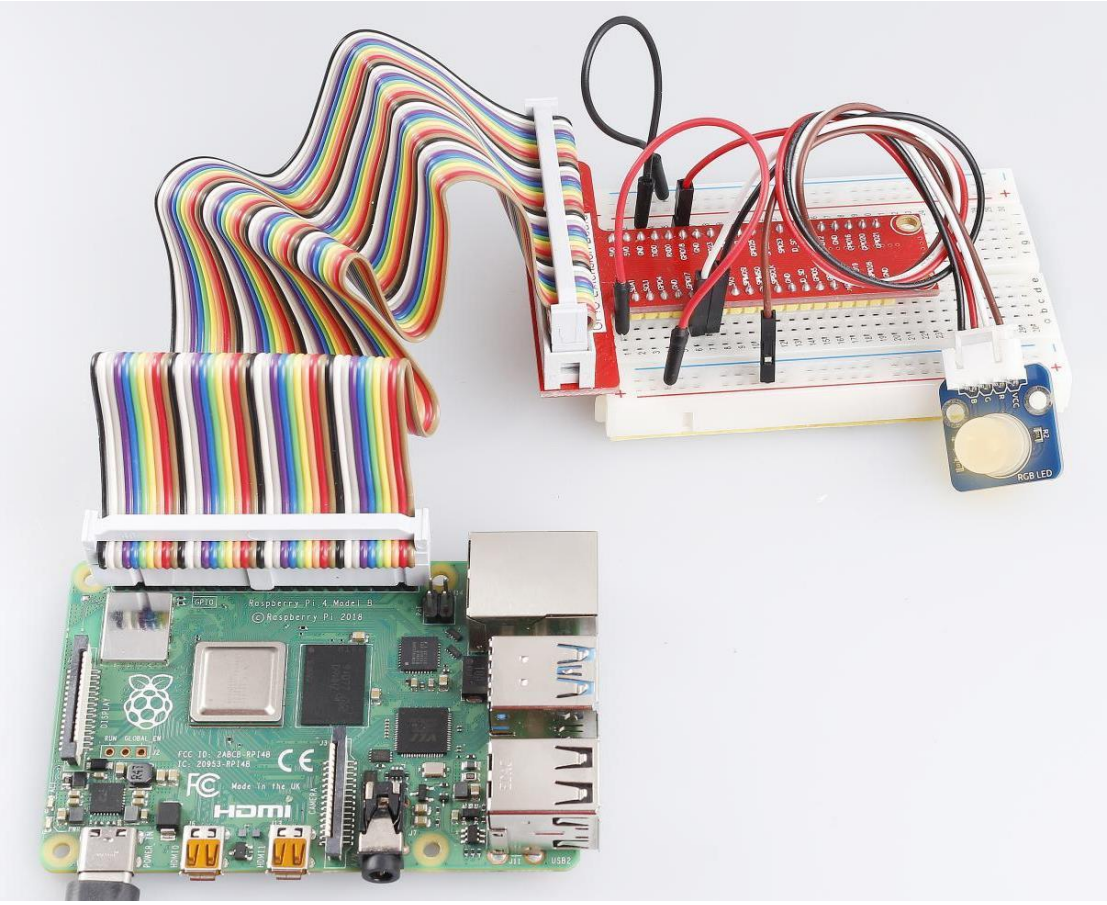
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 02_rgb_led.py
```

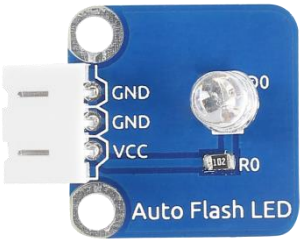
You will see the RGB LED light up, and display different colors in turn.



Lesson 3 7-Color Changing LED

Introduction

The Auto-flash LED module will automatically change between the 7 built-in colors. This module can be used to create interesting effects with light.



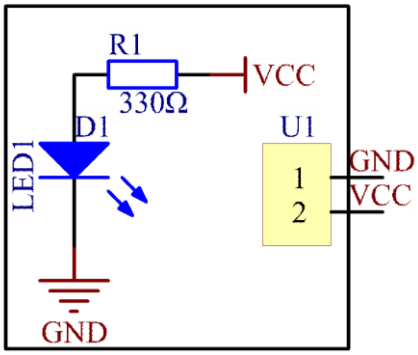
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Auto-flash LED module
- 1 * 3-Pin Non-Reversible Cable

Purpose

This module will automatically change between the 7 built-in colors on power on.

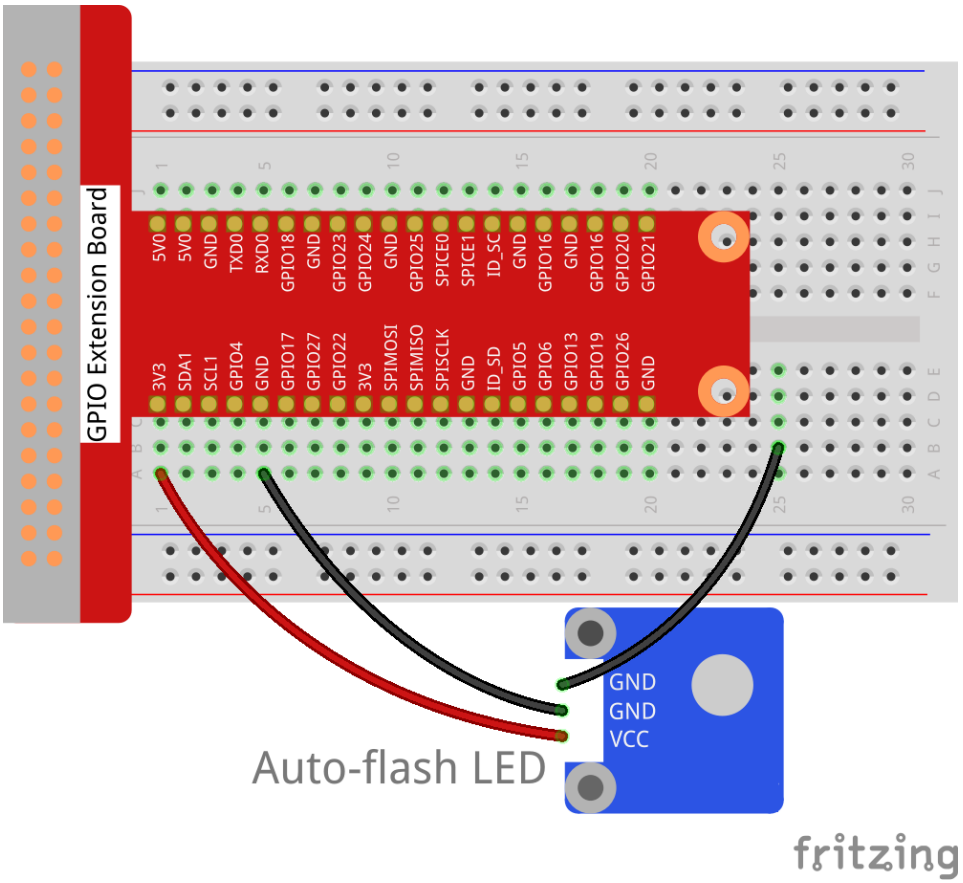
The schematic diagram of the module is as shown below:



Procedure

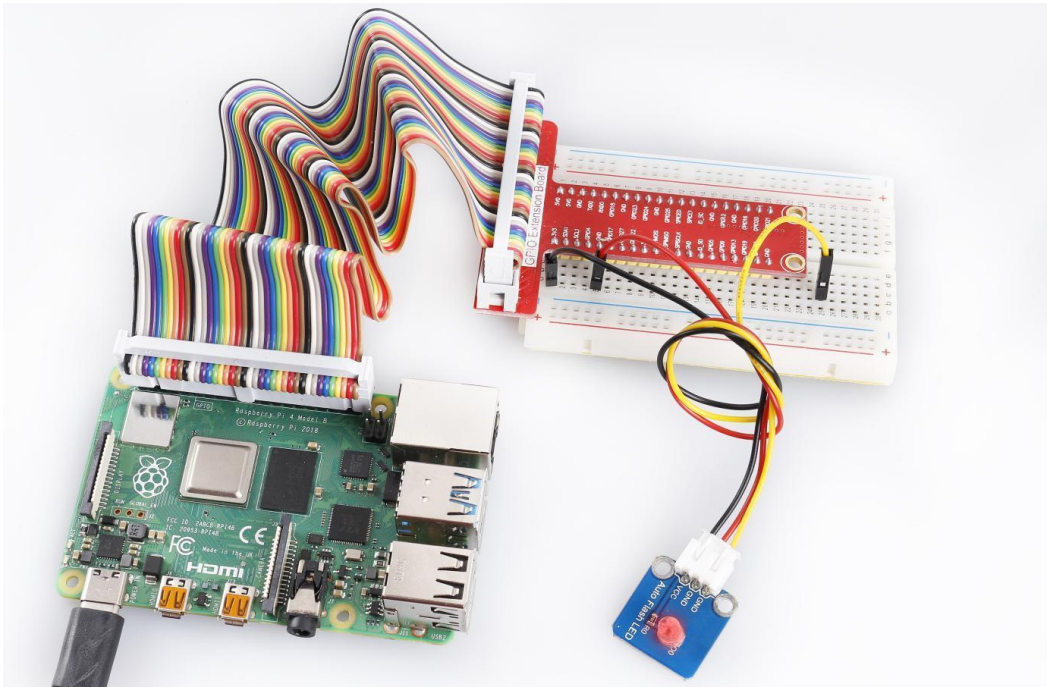
Build the circuit.

Raspberry Pi	GPIO Extension Board	Auto-flash LED Module
GND	GND	GND
3.3V	3V3	VCC



Note:

There are two "GND" pins on the module. You only need to connect one of them.
You will see the module flashing between the seven colors.


GPIO Extension Board
Raspberry Pi 4 Model B
©Raspberry Pi 2018
HDMI
ALRD FRAIN LED
GND
VCC
SIG

Lesson 4 Relay Module

Introduction

The Relay Module is used to provide a connection between two or more devices in response to an input signal. It is suitable for driving high powered electrical equipment, such as light bulbs, electric fans, and air conditioners. The relay module can be used with a Raspberry Pi to control high voltages with a low voltage.

Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- Jumper Wires
- 1 * Relay module
- 1 * Dual-color LED module
- 2 * 3-Pin Non-Reversible Cable

Purpose

Relay

– There are 5 parts in every relay:

1. **Electromagnet** – Consists of an iron core in a coil of wires. When electricity passes through it, it becomes magnetized. It can be activated with either a direct current (DC) or alternating current (AC) depending on its construction.

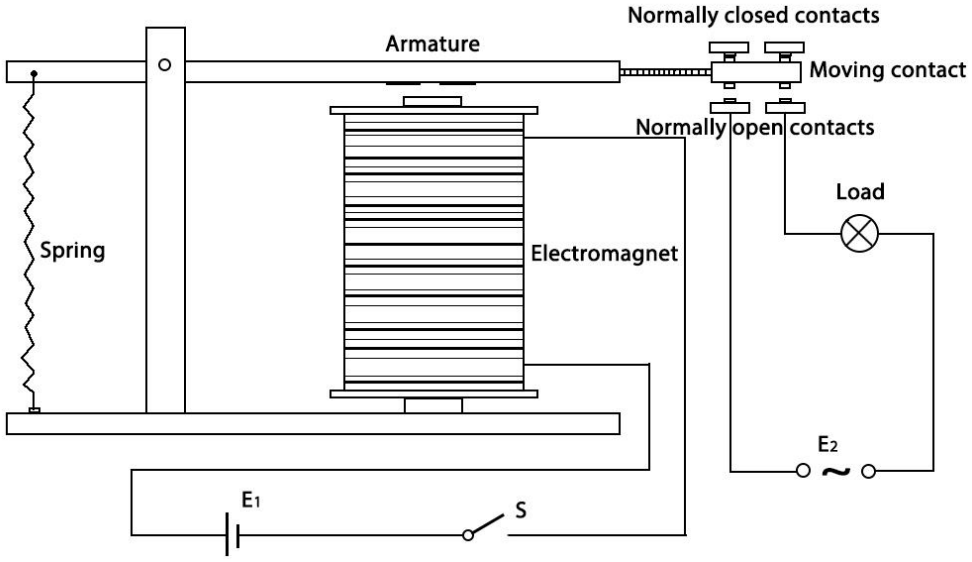
2. **Armature** – The movable magnetic strip is known as an armature. When a current flows through the electromagnet, it produces a magnetic field which moves the armature.

3. **Spring** – When no currents flow through the coil on the electromagnet, the spring pulls the armature away from the electromagnet.

4. Set of electrical **contacts** – There are two contact points and a common:

- Normally open - disconnected when it is inactive, and connected to the common contact when the relay is activated.
- Normally closed – connected to the common contact when it is inactive, and not connected when the relay is activated.

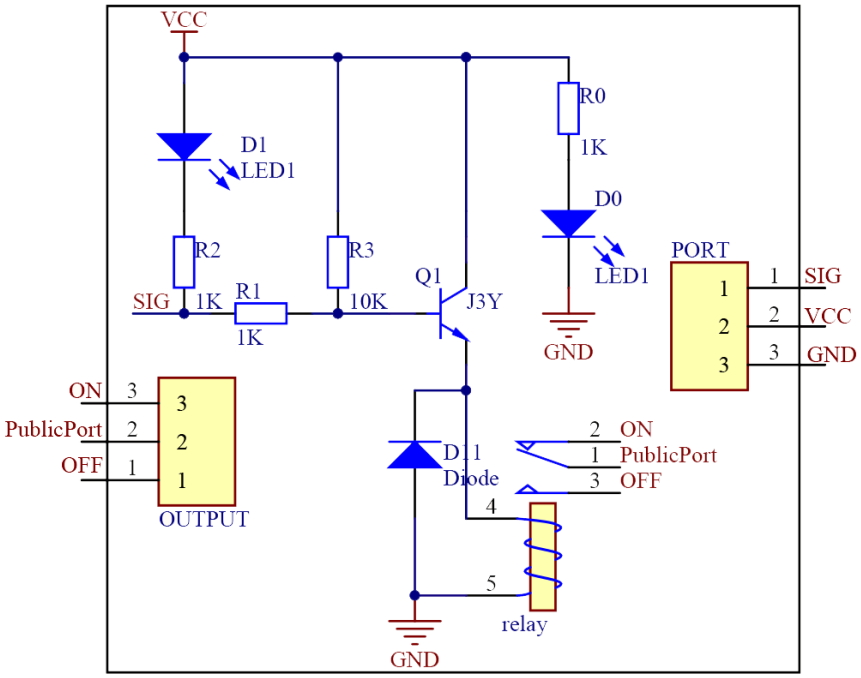
5. Molded frame – Relays are covered with plastic for protection.



The SIG pin of this module is connected to the GPIO pins. When the pin outputs a high level (3.3V) current, the transistor will conduct due to the saturation. The normally open contact of the relay will be closed, while the normally closed contact of the relay will be broken. When the current output is low (0V), the transistor will be cut off, and the relay returns to its initial state.

The schematic diagram of the module is as shown below:

- Common is shown as PublicPort
- N/C is shown as ON
- N/O is shown as OFF

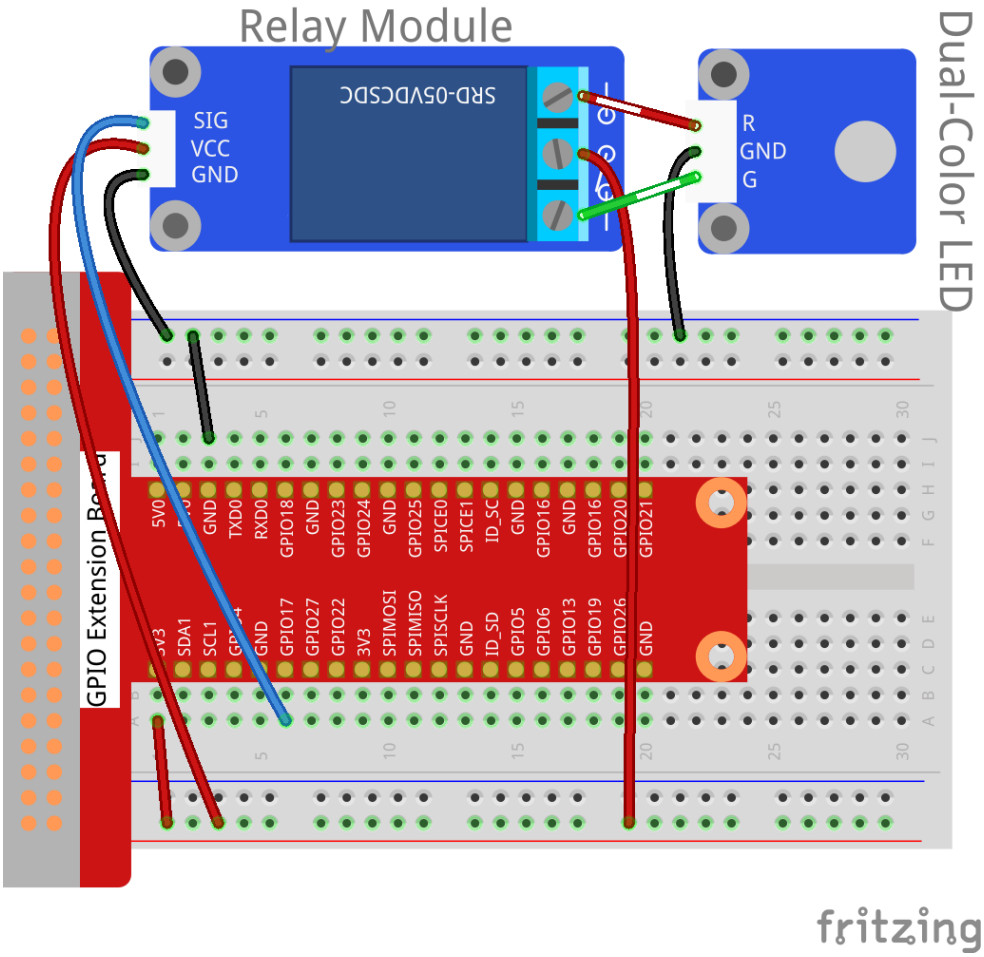


Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Relay Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND
3.3V	3V3	COM

Dual-color LED Module	GPIO Extension Board	Relay Module
R	*	Normal Open
GND	GND	*
G	*	Normal Close



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/04_relay/
```

Step 3: Compile.

```
gcc relay.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

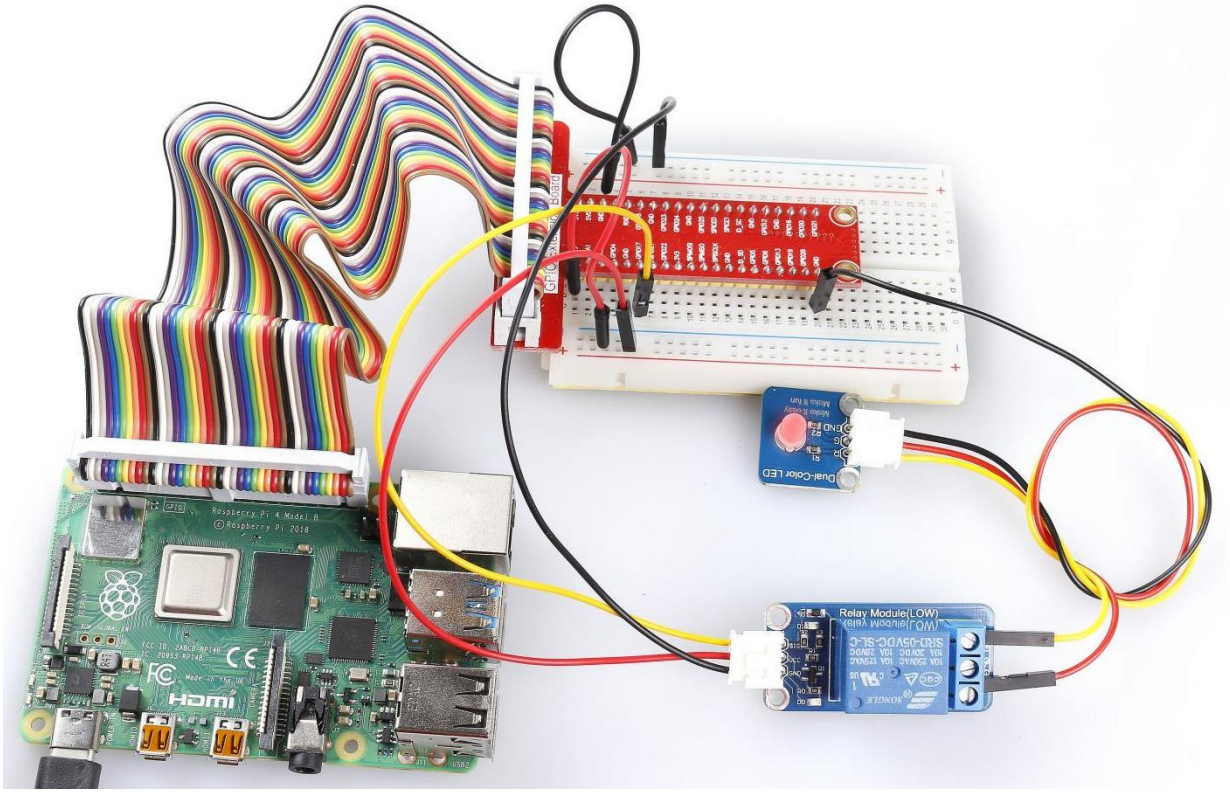
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 04_relay.py
```

When the contacts move, you may hear a ticking sound. A high voltage device (like a 220V light bulb) can be attached to the output port of the relay. The relay will act as an automated switch.


Dual-Color LED
Relay Module(LOW)

Lesson 5 Laser Emitter Module

Introduction

This module will emit laser light, which is widely used in medical treatment, military, and many other fields due to its good directivity and energy concentration.

Required Components

- 1 * Raspberry Pi

- 1 * Breadboard

- 1 * Laser Emitter module

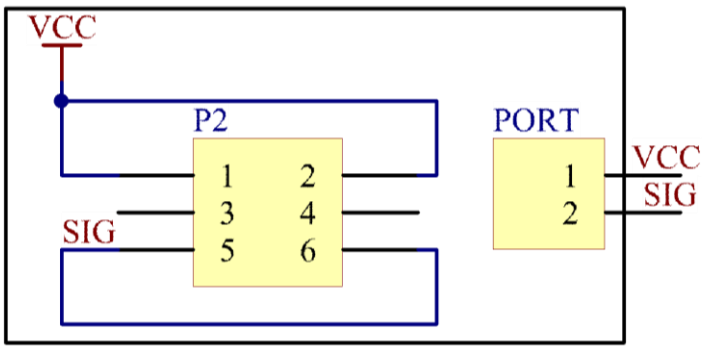
- 1 * 2-Pin Non-Reversible Cable

Purpose

A laser is a device that emits light through a process of optical amplification based on the stimulated emission of electromagnetic radiation. Lasers differ from other sources of light because they emit light coherently.

Spatial coherence allows a laser to be focused to a tight spot, enabling applications like laser cutting and lithography, and a laser beam to stay narrow over long distances (collimation), enabling applications like laser pointers. Lasers can also have high temporal coherence which allows them to have a very narrow spectrum, i.e., they only emit light of a single color. And its temporal coherence can be used to produce pulses of light—as short as a femtosecond.

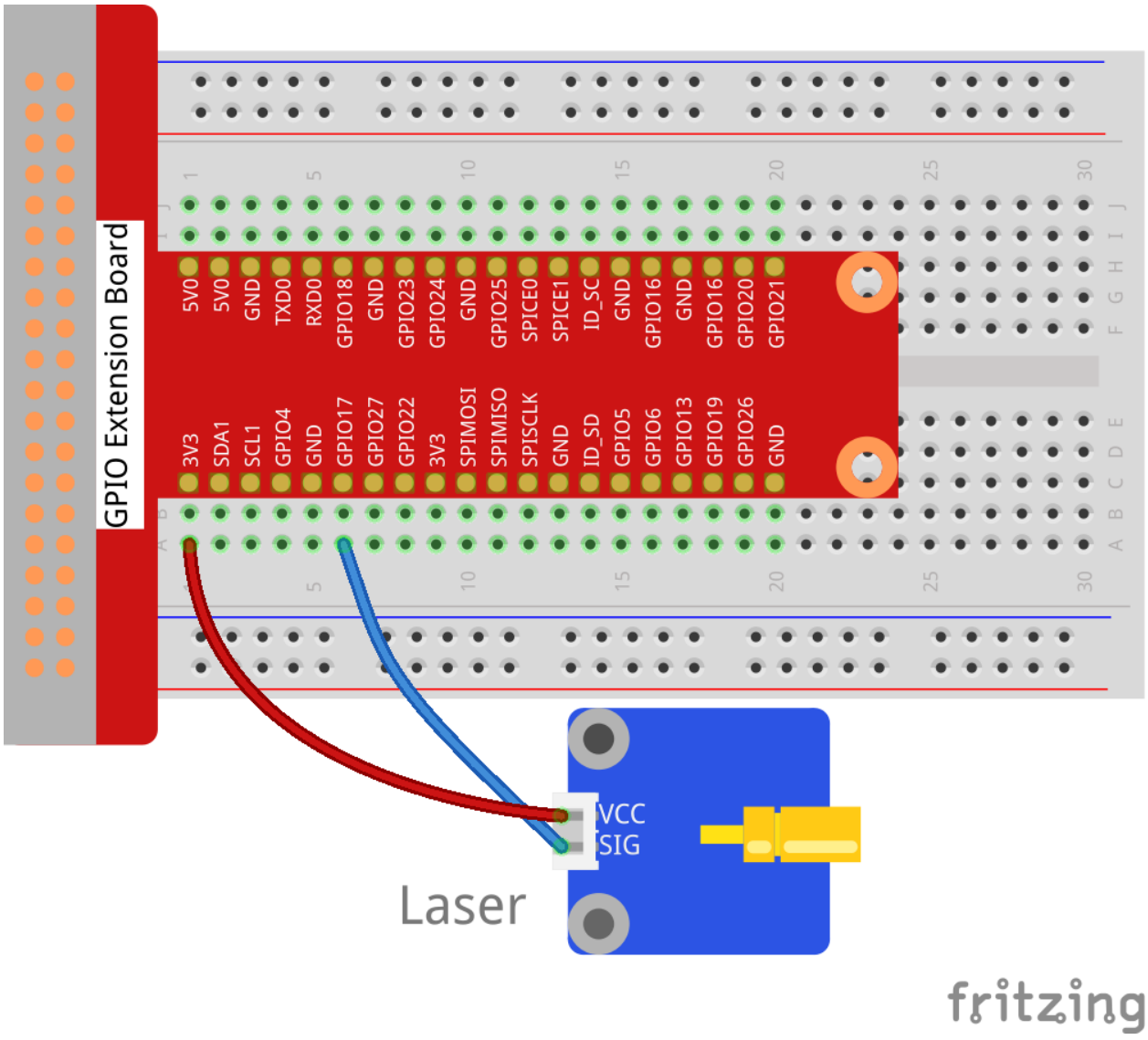
The schematic diagram of the module is as shown below:



Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Laser Emitter Module
3.3V	3V3	VCC
GPIO0	GPIO17	SIG



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/05_laser/
```

Step 3: Compile.

```
gcc laser.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

Step 2: Change directory.

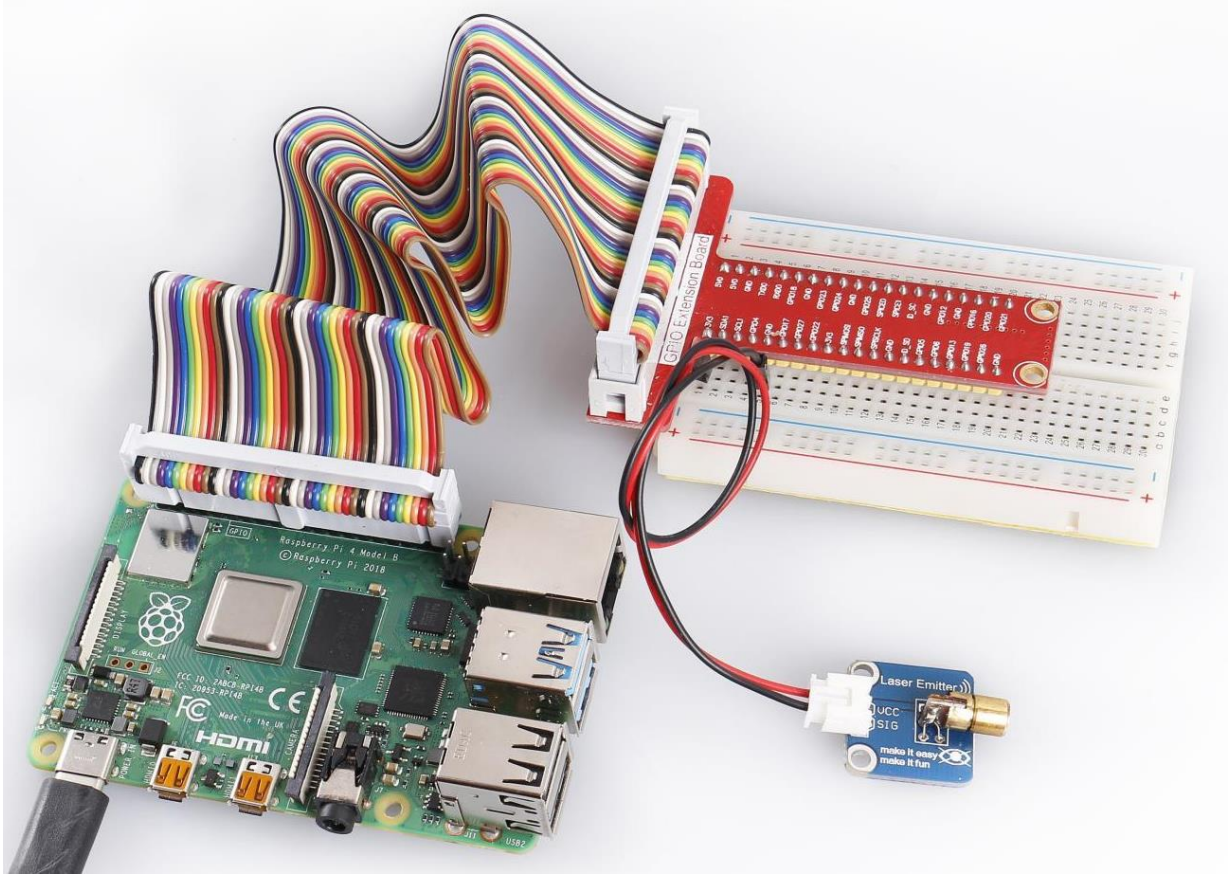
```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 05_laser.py
```

The module will now send out Morse code signals.

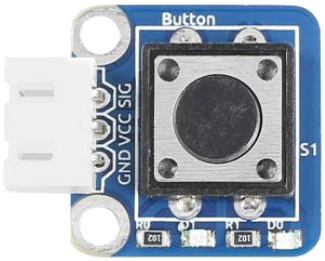
Warning: DO NOT look directly at the laser head. It can cause great harm to your eyes. You can point the laser beam to the table and see the light spot on the table.



Lesson 6 Button Module

Introduction

In this lesson, we will use a button module to control a dual-color LED module.



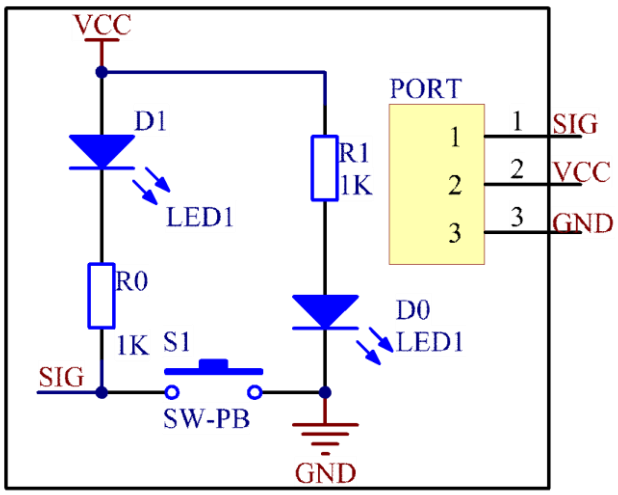
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- Jumper Wires
- 1 * Button module
- 1 * Dual-color LED module
- 2 * 3-Pin Non-Reversible Cable

Purpose

The Button Module acts as a normally open input device for the Raspberry Pi. When pressed, the GPIO connected to the button will change to low level (0V). Corresponding code can be used to track this GPIO change and carry out an action as a result. In this case, a string will be printed on the screen and the LED will change color.

The schematic diagram of the module is as shown below:

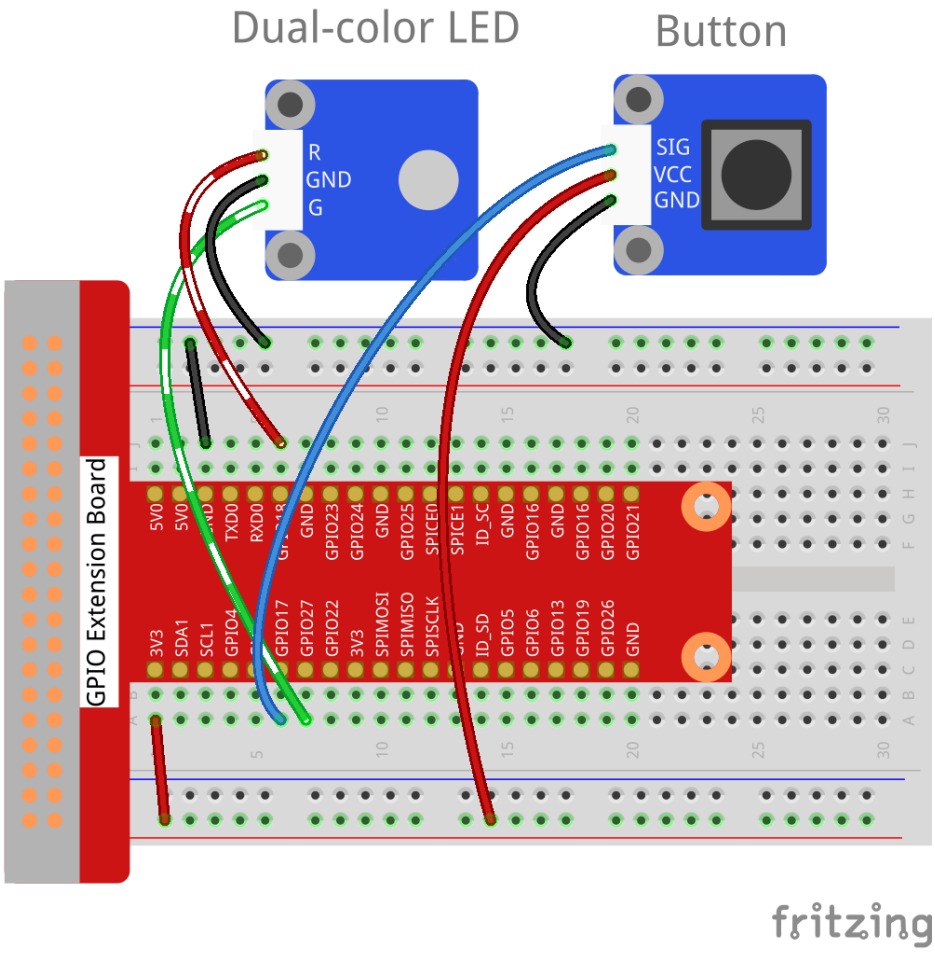


Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Button Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND

Raspberry Pi	GPIO Extension Board	Dual-Color LED Module
GPIO1	GPIO18	R
GND	GND	GND
GPIO2	GPIO27	G



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/06_button/
```

Step 3: Compile.

```
gcc button.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

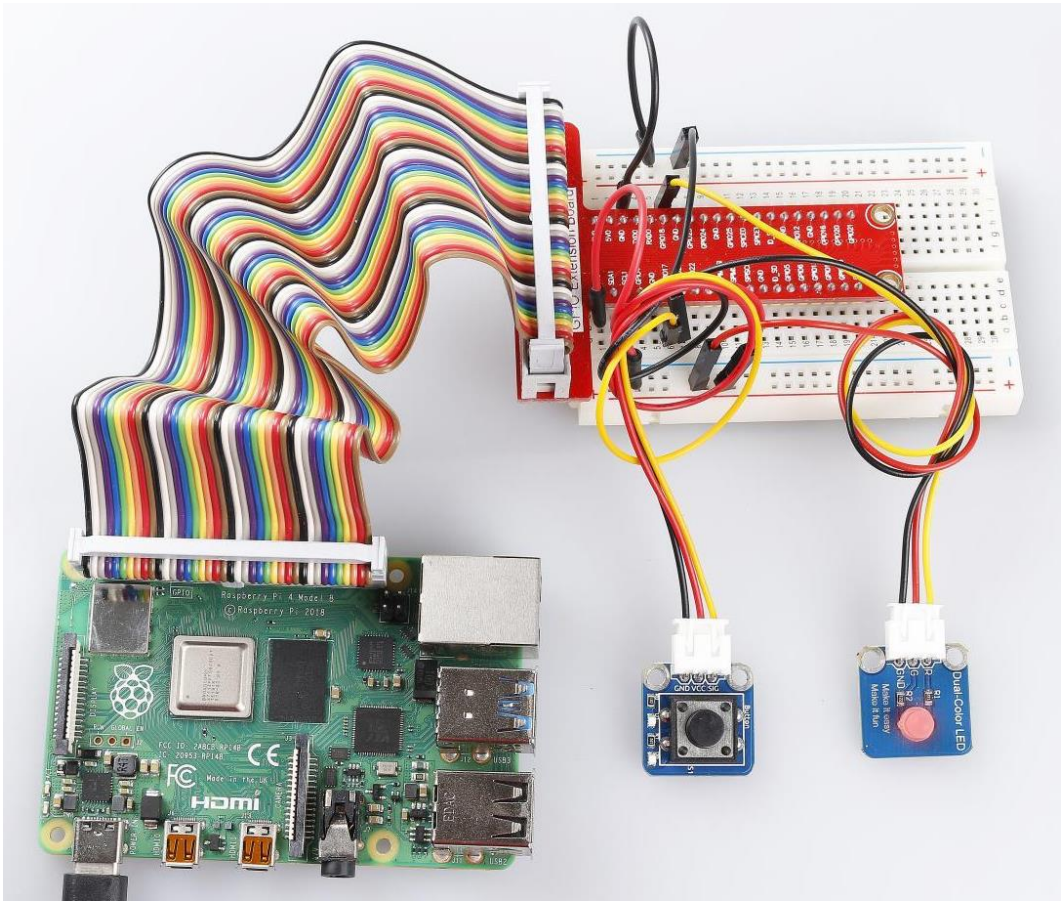
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 06_button.py
```

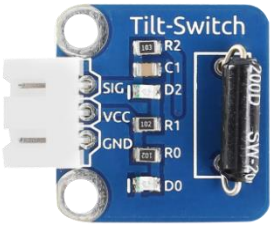
The LED on the module will emit green light. If you press the button, "Button pressed" will be printed on the screen and the LED will emit red light. If you release the button, "Button released" will be printed on the screen and the LED will flash green again.



Lesson 7 Tilt-Switch Module

Introduction

The Tilt-Switch Module in this kit uses an internal metal ball to detect inclinations of a small angle.



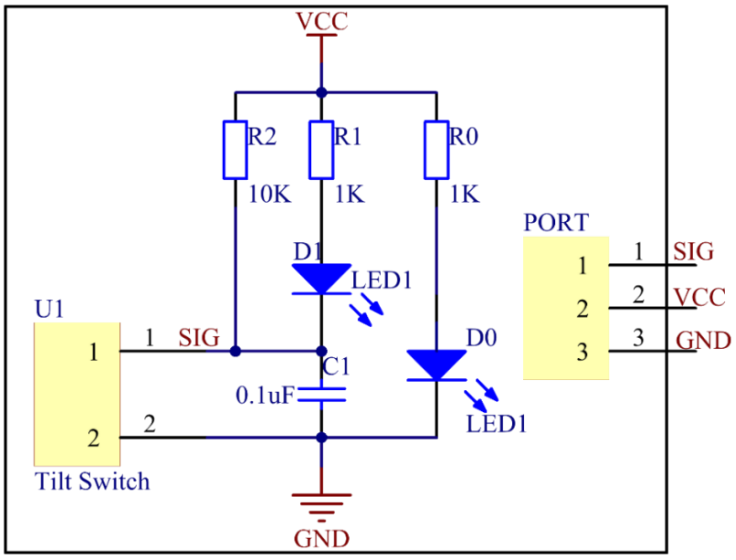
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Dual-color LED module
- 1 * Tilt-switch module
- 2 * 3-Pin Non-Reversible Cable

Purpose

The circuit is triggered by the Tilt-Switch when the ball is moved by a small inclination. When an external force causes the internal ball to shake, the tilt switch will conduct causing the LED to emit red light instead of the default green.

The schematic diagram of the module is as shown below:

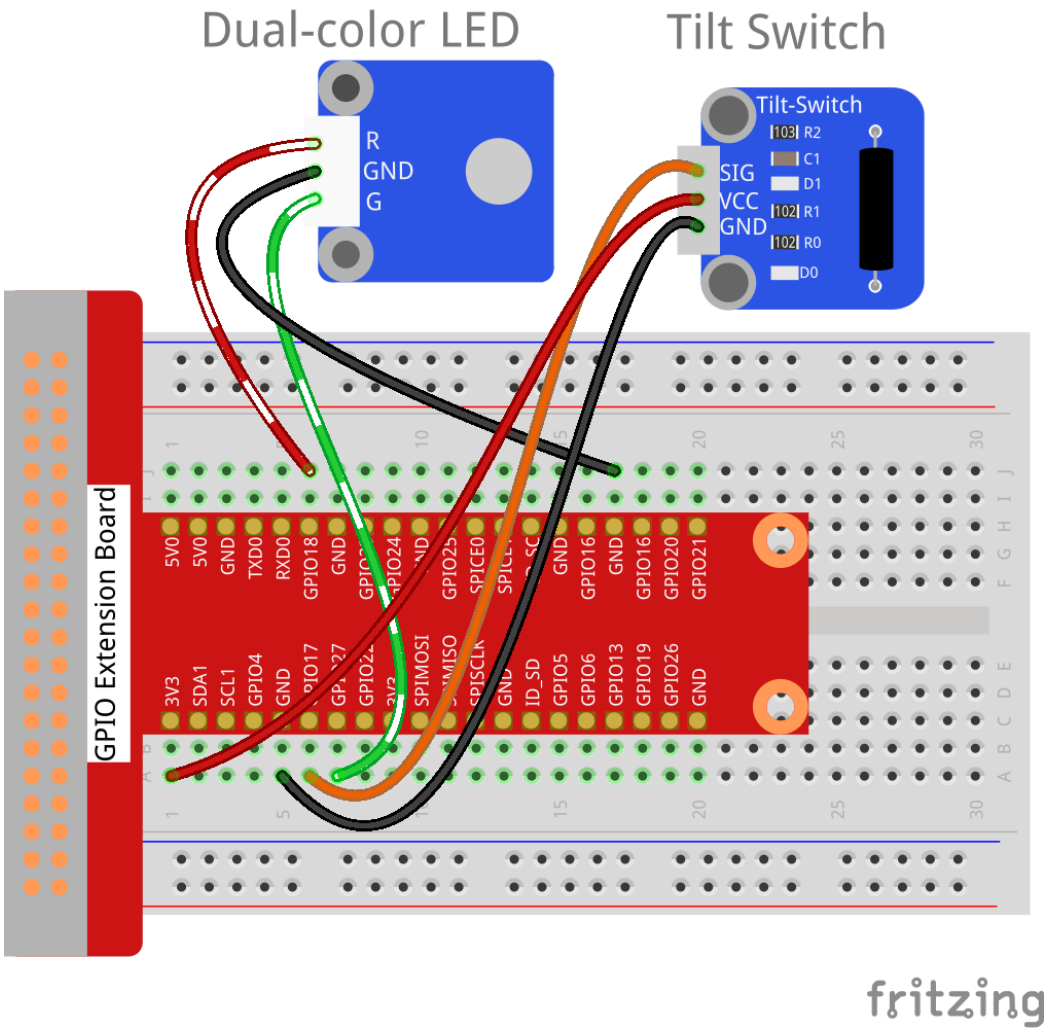


Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Tilt Switch Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND

Raspberry Pi	GPIO Extension Board	Dual-Color LED Module
GPIO1	GPIO18	R
GND	GND	GND
GPIO2	GPIO27	G



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/07_tilt_switch/
```

Step 3: Compile.

```
gcc tilt_switch.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

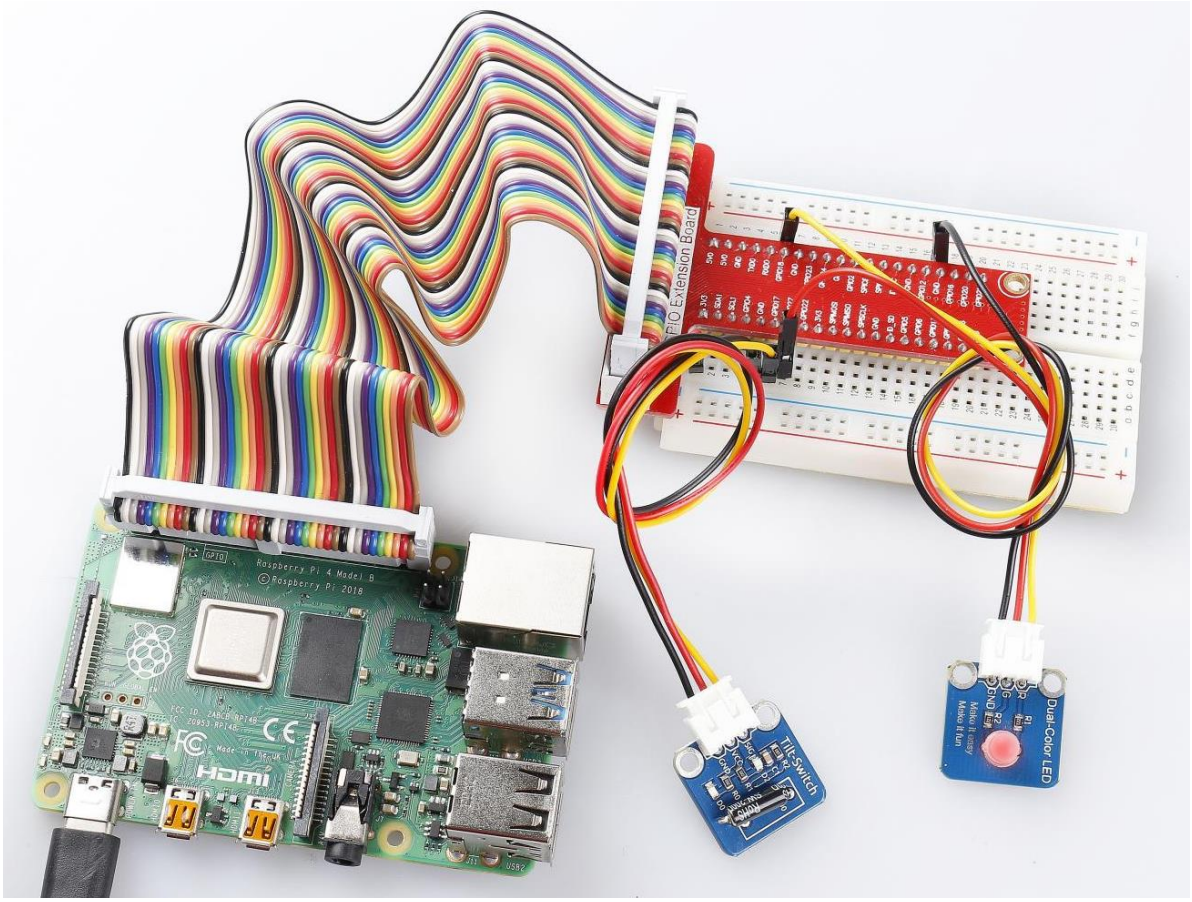
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 07_tilt_switch.py
```

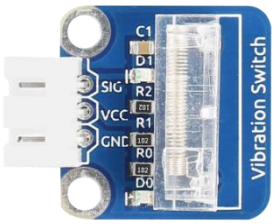
Place the tilt switch module horizontally, and the LED will flash green. If you tilt it, "Tilt!" will be printed on the screen and the LED will change to red. Place it horizontally again, and the LED will flash green again.



Lesson 8 Vibration Switch

Introduction

A vibration switch, also called spring switch or shock sensor, is an electronic switch which induces shock force and transfers the result to a circuit device thus triggering it to work. It contains the following parts: conductive vibration spring, switch body, trigger pin, and packaging agent.



Required Components

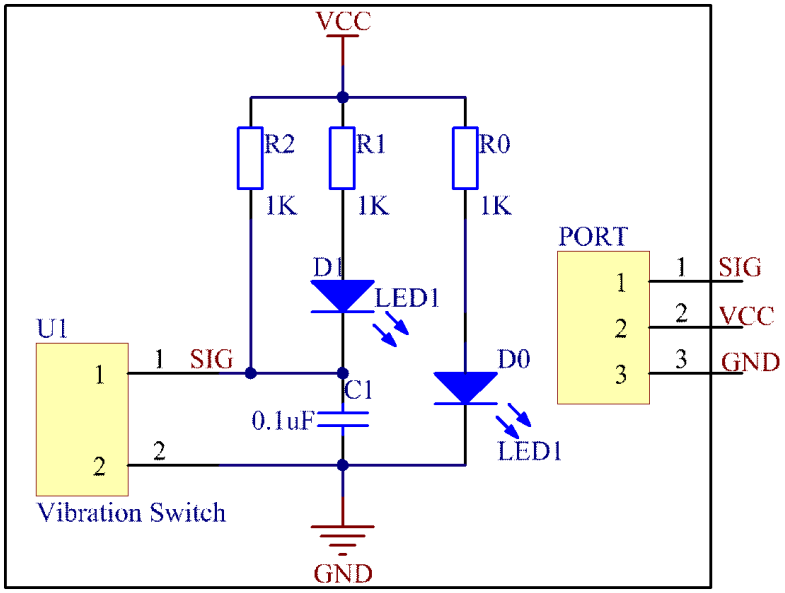
- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Dual-color LED module
- 1 * Vibration switch module
- 2 * 3-Pin Non-Reversible Cable

Purpose

In a vibration switch module, the conductive vibration spring and trigger pin are precisely placed in the switch and fixed by adhesive. Normally, the spring and the trigger pin are separated. Once the sensor detects shock, the spring will vibrate and contact with the trigger pin, thus conducting and generating trigger signals.

In this experiment, a dual-color LED module is connected to the Raspberry Pi to indicate activity. When the vibration sensor is tapped or knocked, the LED will flash either red or green.

The schematic diagram:

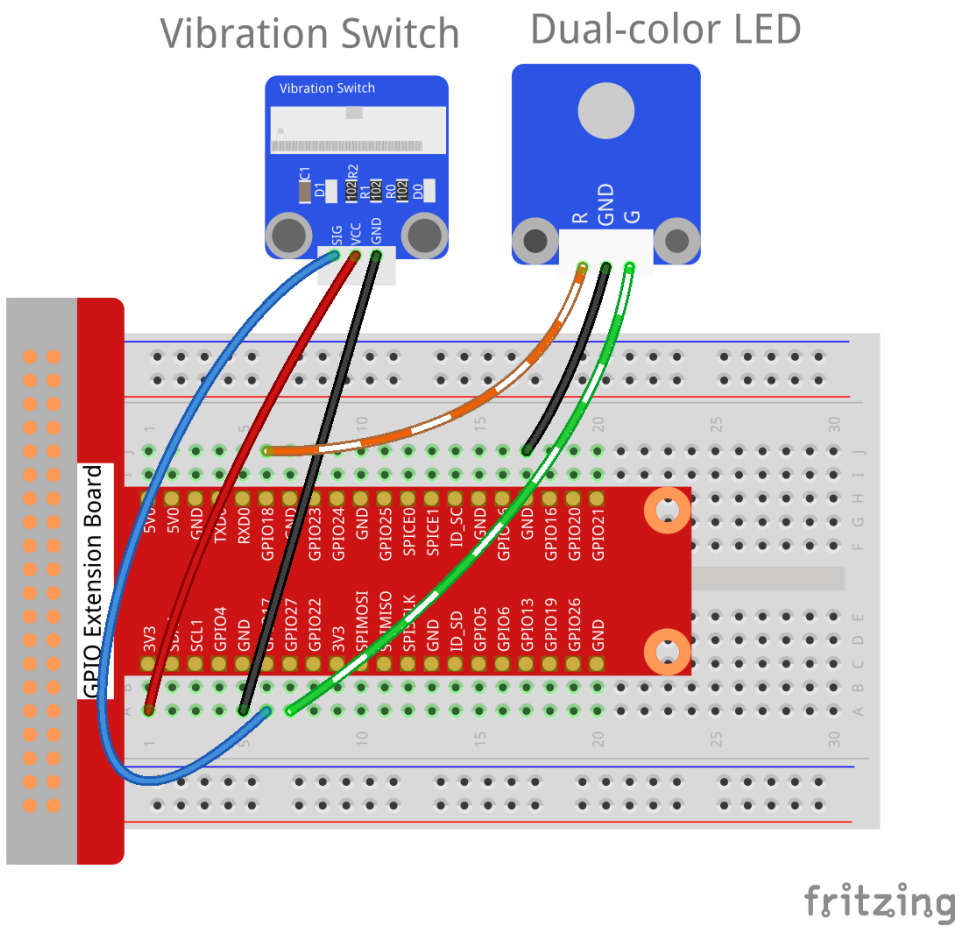


Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Vibration Switch Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND

Raspberry Pi	GPIO Extension Board	Dual-Color LED Module
GPIO1	GPIO18	R
GND	GND	GND
GPIO2	GPIO27	G



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/08_vibration_switch/
```

Step 3: Compile.

```
gcc vibration_switch.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

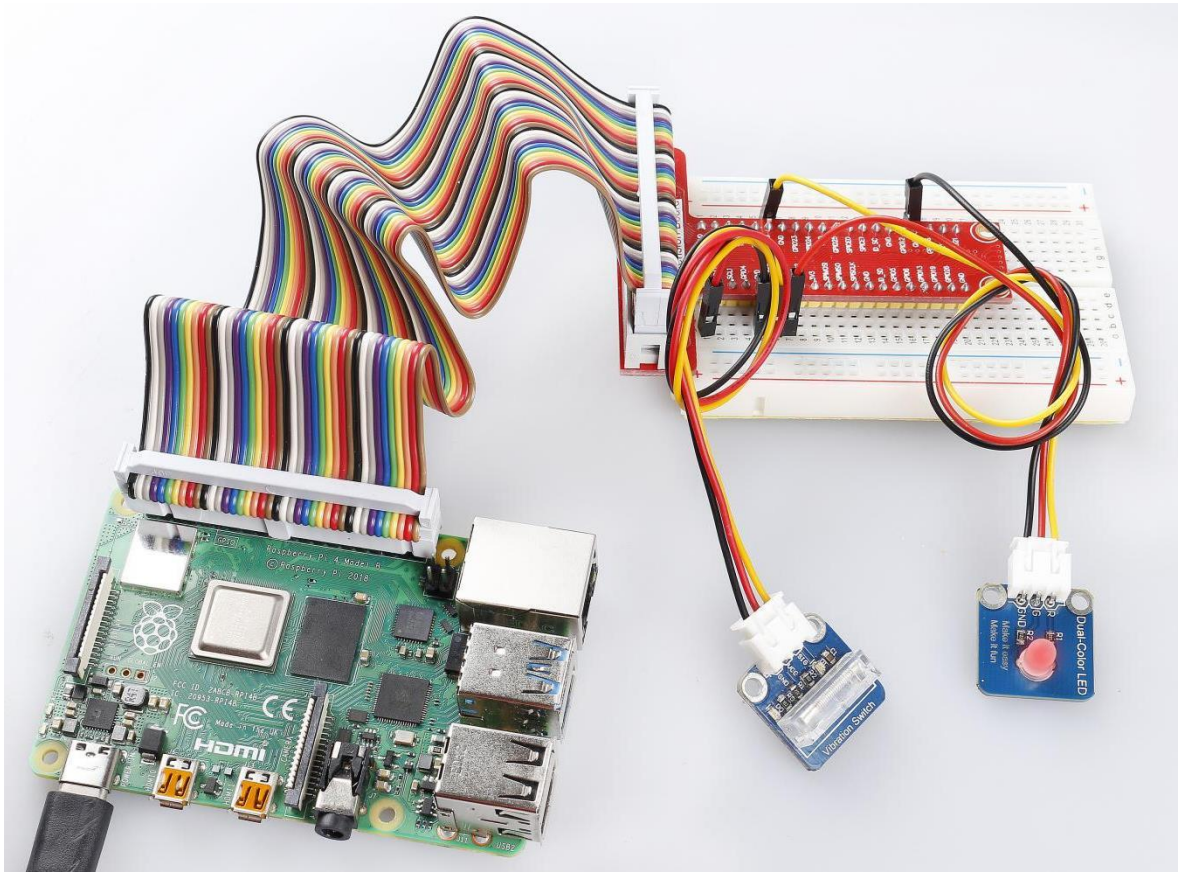
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 08_vibration_switch.py
```

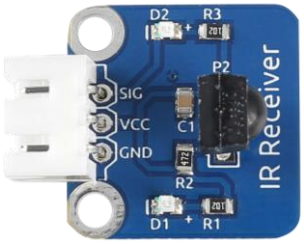
Tap or knock the sensor to see the LED flash red or green. Each tap or knock will make it alternate between the two colors.


Dual-Color LED
Vibration Switch

Lesson 9 IR Receiver Module

Introduction

An infrared-receiver is a component that can independently receive infrared rays and output signals at the TTL level. this module is suitable for all kinds of projects with infrared remote control and transmission.



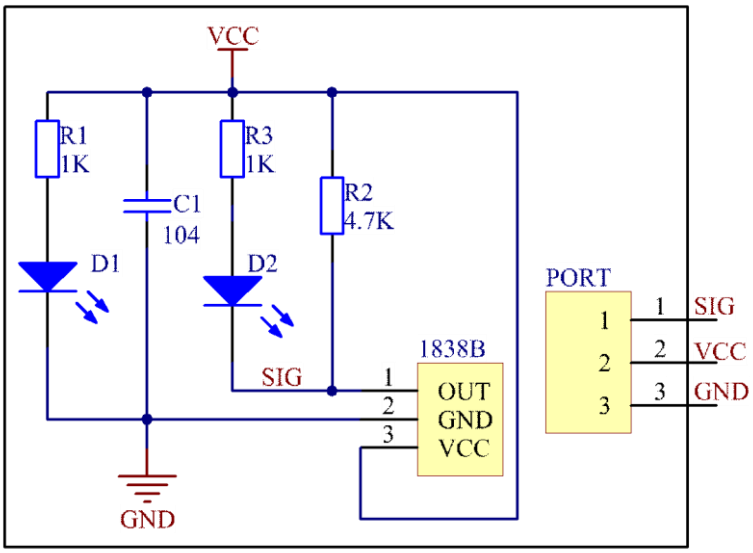
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * IR receiver module
- 1 * IR Remote Controller
- 1 * 3-Pin Non-Reversible Cable

Purpose

In this experiment, the IR remote control sends signals to the IR receiver when buttons are pressed. The Raspberry Pi will keep a counter to track each time the IR module receives a signal.

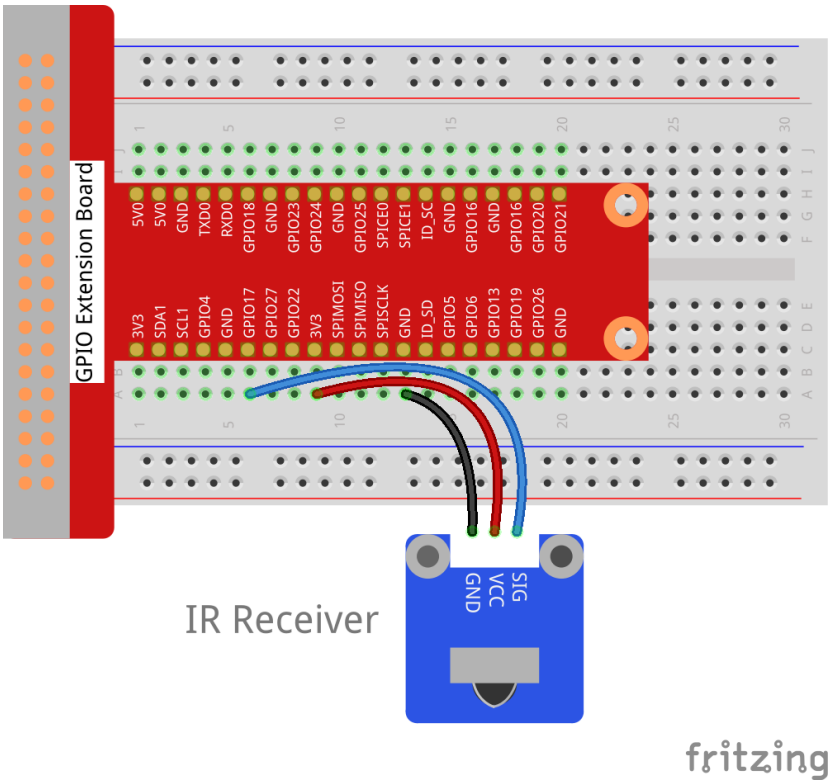
The schematic diagram of the module is as shown below:



Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	IR Receiver Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/09_ir_receiver/
```

Step 3: Compile.

```
gcc ir_receiver.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

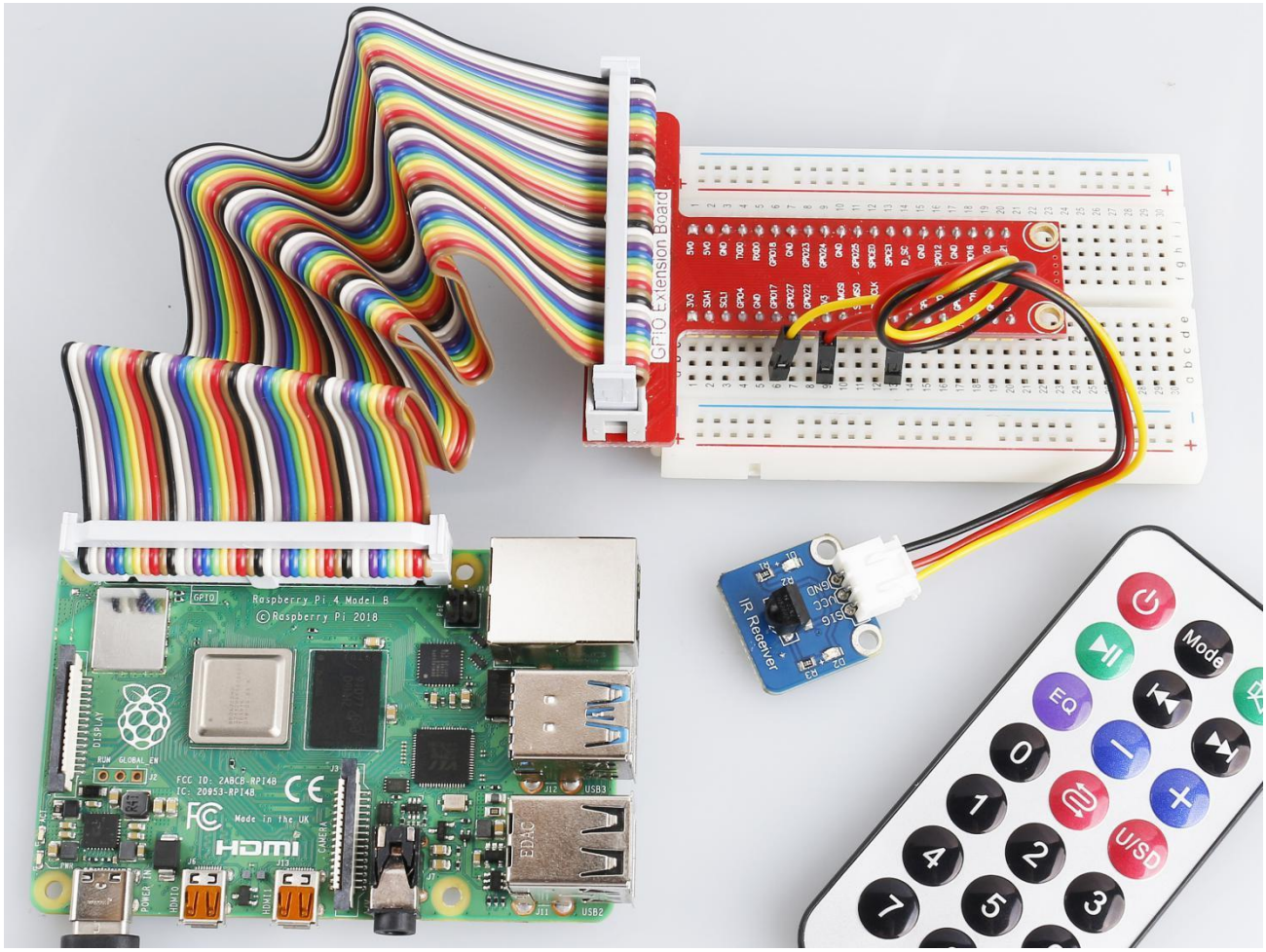
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 09_ir_receiver.py
```

Press any key on the remote. The LED module will blink and the counter will print to the screen.


GPIO Extension Board
Raspberry Pi 4 Model B
©Raspberry Pi 2018
IR Receiver
GND
VCC
SIG
Mode
EQ
U/SD

Lesson 10 Buzzer Module

Introduction

Buzzers can be categorized as either active or passive (see the example pictures below).

Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Passive buzzer module
- 1 * Active buzzer module
- 1 * 3-Pin Non-Reversible Cable

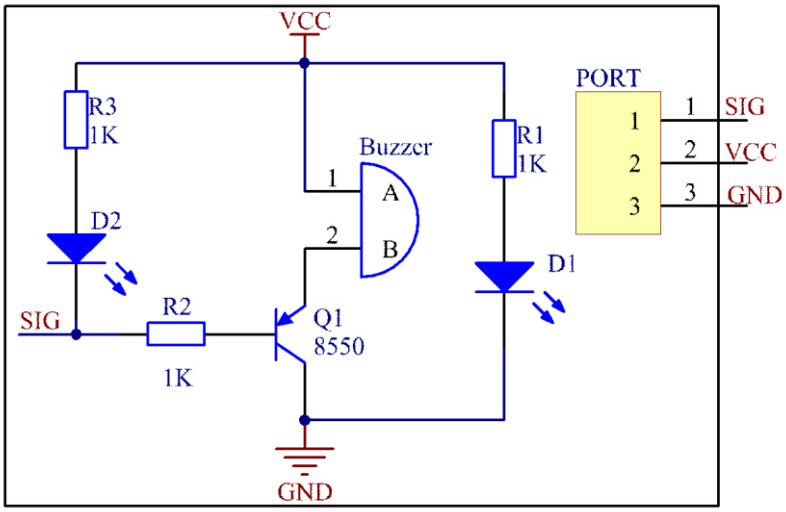
Purpose

To identify the type of buzzer, place them face down to expose the pins. The one with the green circuit is the passive buzzer, while the one with the black fill is the active buzzer.

	
Active buzzer	Passive buzzer

The difference between the two types of buzzers is that the active one has a built-in oscillating source which will make sounds when electrified. The passive buzzer on the other hand will only beep when square waves between 2k and 5k are used to drive it.

The schematic diagram of the module is as shown below:



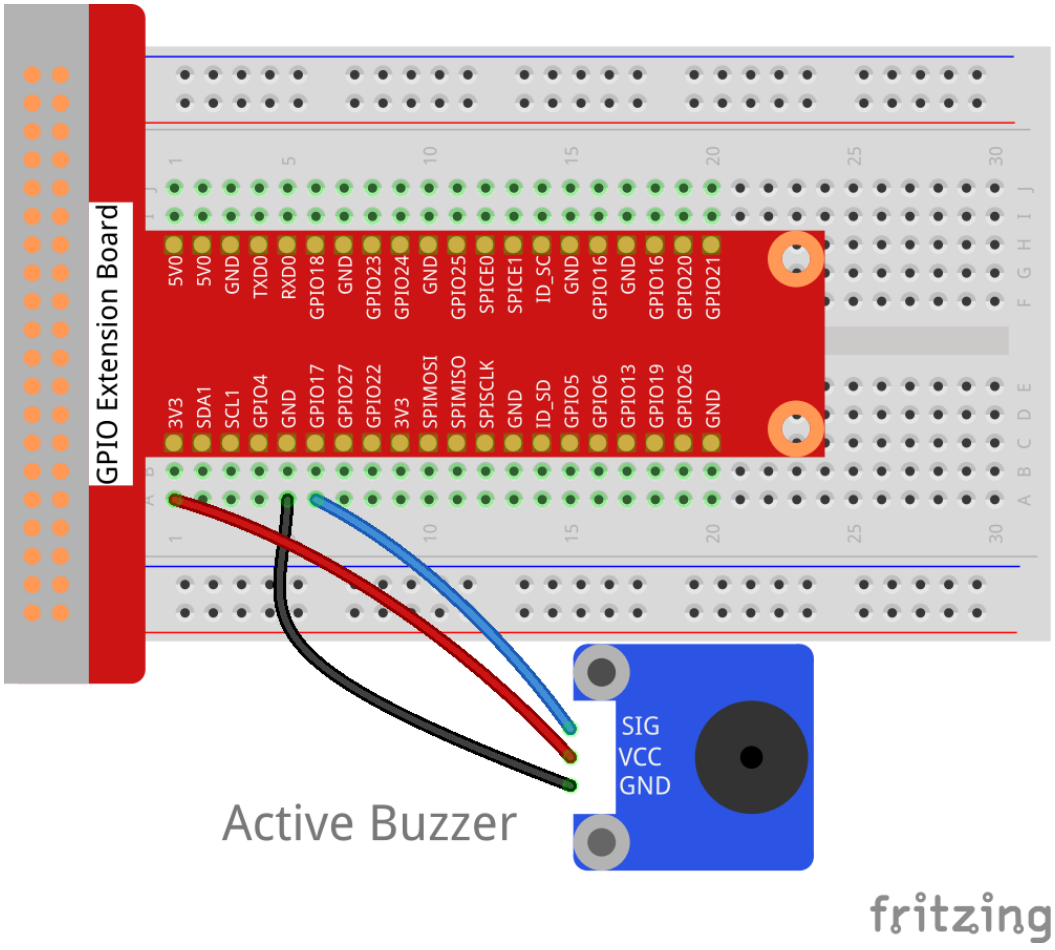
Procedure

Active Buzzer

Note:

The active buzzer has built-in oscillating source, so it will beep as long as it is wired up, but it can only beep with fixed frequency.

Step 1: Build the circuit.



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/10_active_buzzer/
```

Step 3: Compile.

```
gcc active_buzzer.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

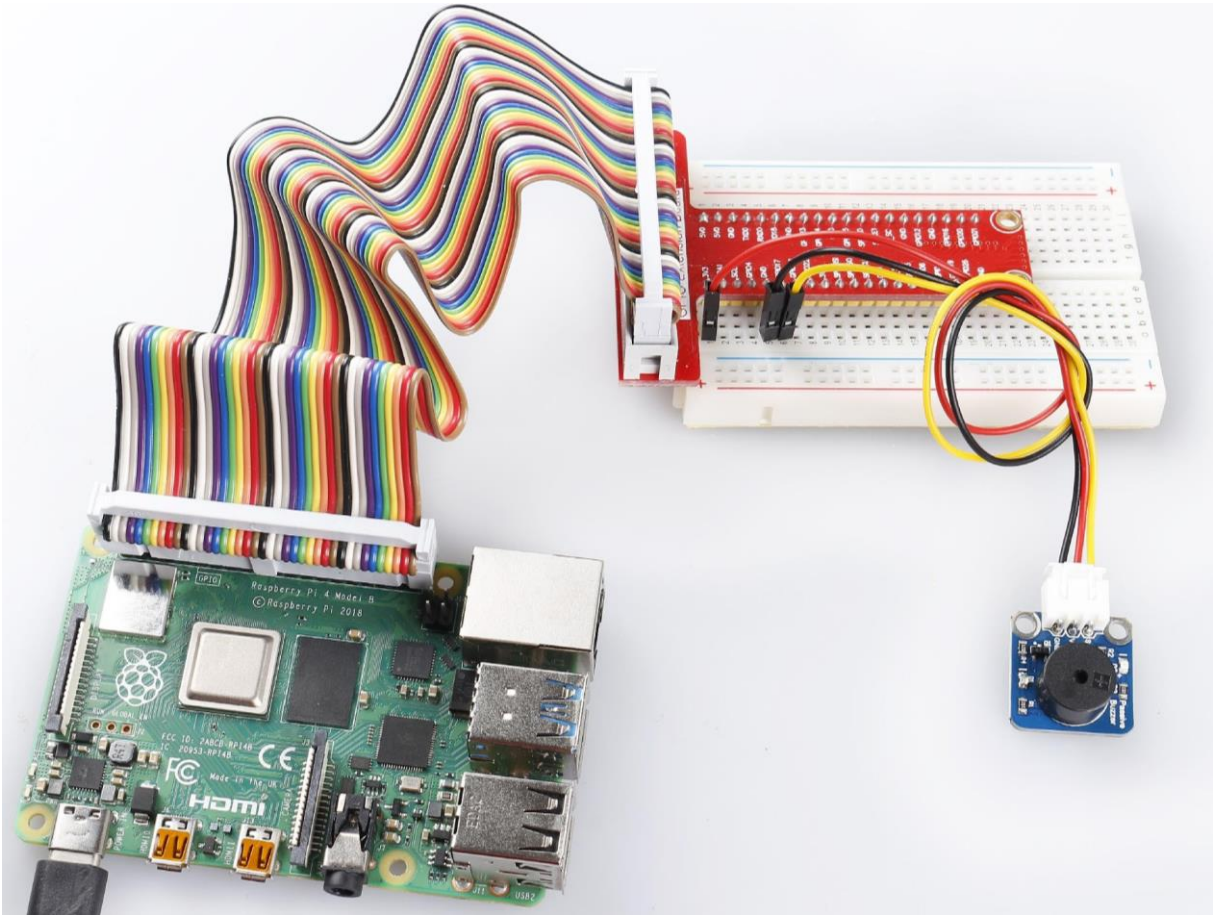
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 10_active_buzzer.py
```

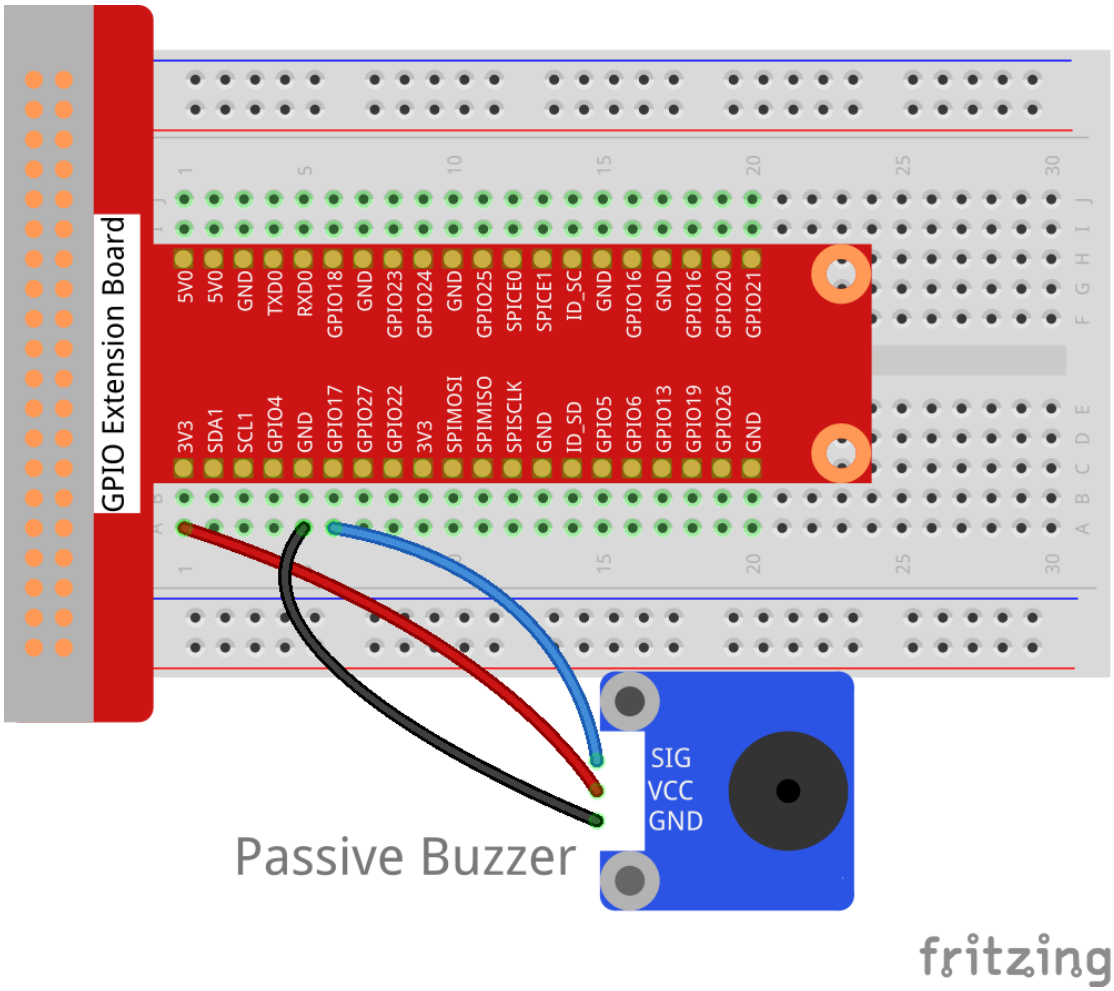
Now you can hear the active buzzer beeping.



Passive Buzzer

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Passive Buzzer Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/10_passive_buzzer/
```

Step 3: Compile.

```
gcc passive_buzzer.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

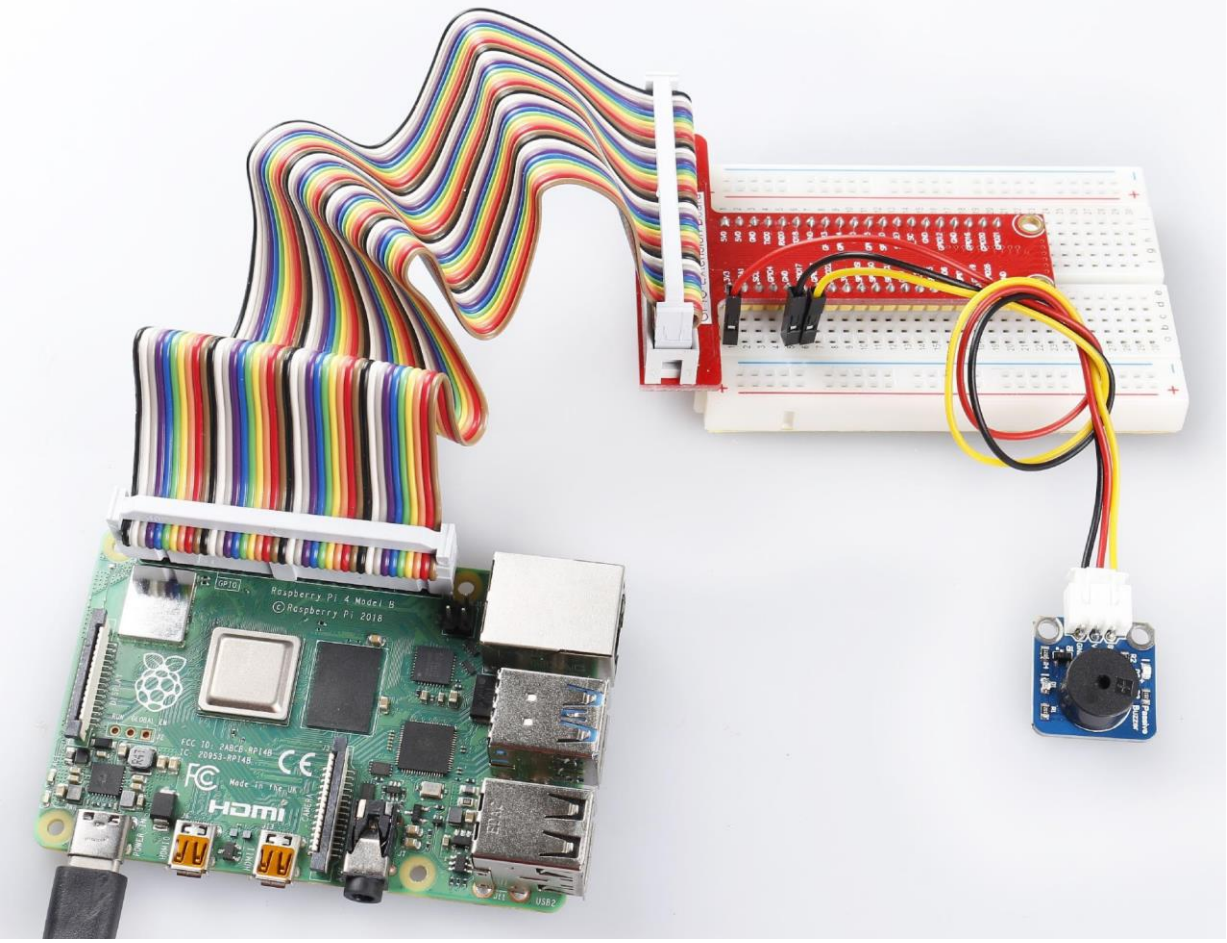
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 10_passive_buzzer.py
```

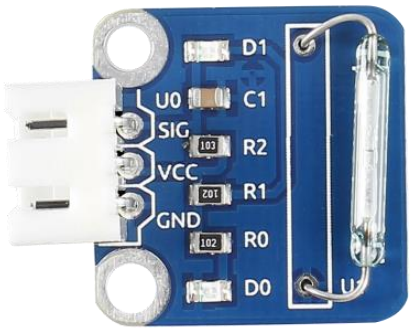
Now you can hear the passive buzzer playing music.



Lesson 11 Reed Switch

Introduction

Reed switches are used to detect the existence of magnetic fields.



Required Components

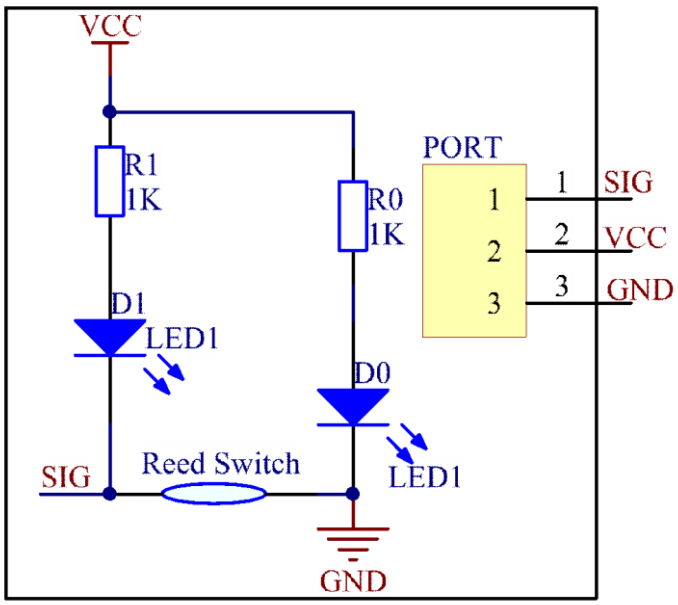
- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Reed switch module
- 1 * Dual-color LED module
- 2 * 3-Pin Non-Reversible Cable
- 1 * Magnet (Not Included)

Purpose

A reed switch is a type of line switch that triggers as a result of magnetic signals. The "switch" here means dry reed pipe, which is a type of passive electronic component with the advantages of a simple structure, small size, and convenient control. The shell of the reed switch is commonly a sealed glass pipe filled with inert gases in which two iron elastic reed electroplates are equipped. Normally, the two reeds are separated within the tube, however when a magnetic substance approaches they are drawn together. As a result, the reeds will pull together and complete the circuit.

Once the external magnetic force disappears, the two reeds will separate and return to normal. This technology is used in a variety of communication devices.

The schematic diagram of the module is as shown below:

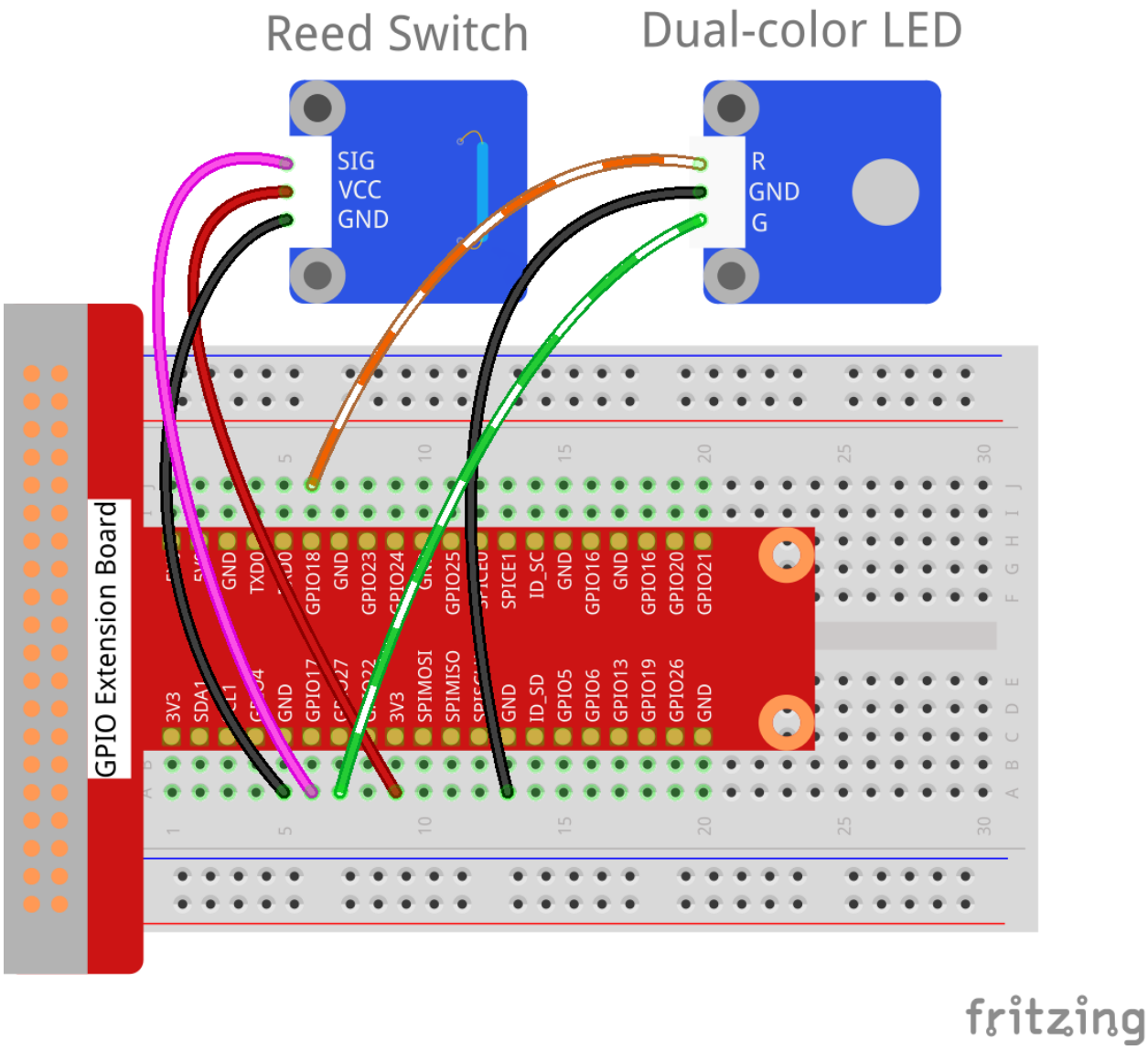


Procedure

Step 1: Build the circuit

Raspberry Pi	GPIO Extension Board	Reed Switch Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND

Raspberry Pi	GPIO Extension Board	Dual-color LED Module
GPIO1	GPIO18	R
GND	GND	GND
GPIO2	GPIO27	G



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/11_reed_switch/
```

Step 3: Compile.

```
gcc reed_switch.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

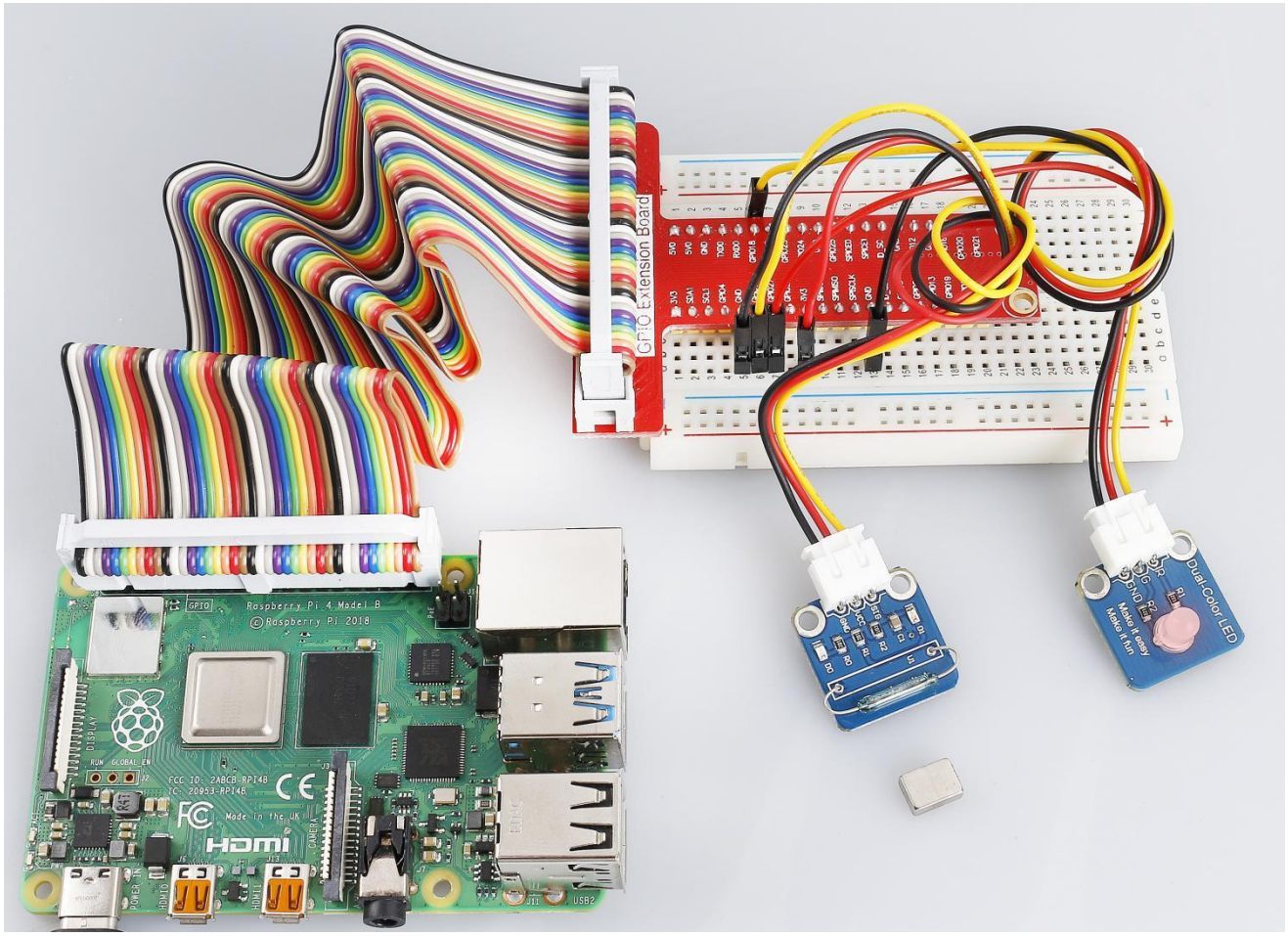
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 11_reed_switch.py
```

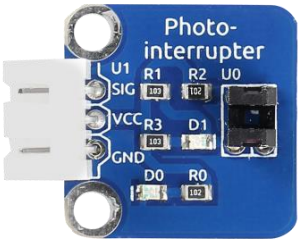
The LED will flash green until a magnet is placed near the reed switch. "Detected Magnetic Material!" will print to the screen, and the LED will change to red. Once the magnet is removed, the LED will return to green.


GPIO Extension Board
Raspberry Pi 4 Model B
©Raspberry Pi 2018
Made in the UK
HDMI
Dual Color LED
Make it easy
Make it fun

Lesson 12 Photo-interrupter

Introduction

A photo-interrupter is a sensor with a light-emitting component and a light-receiving component placed face-to-face. When an object passes through the sensor, the light is interrupted. This sensor is widely used in speed measurement technology.



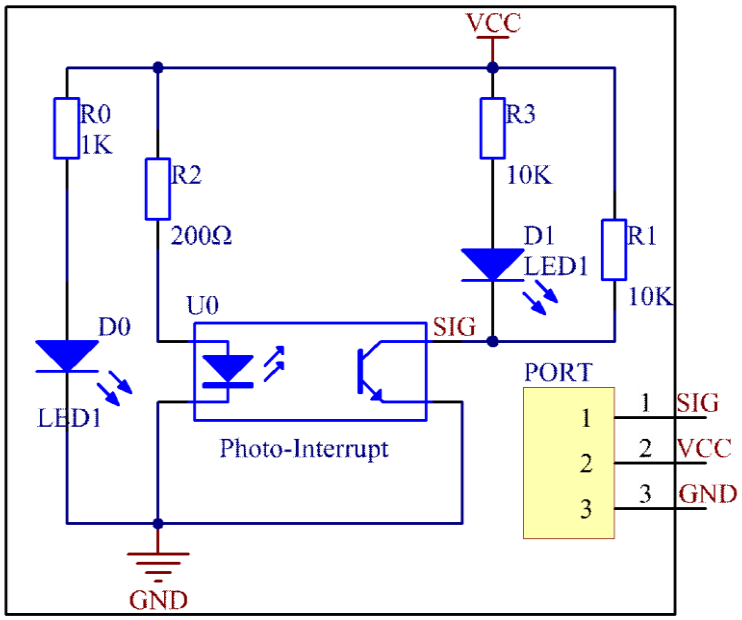
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Dual-color LED module
- 1 * Photo-interrupter module
- 2 * 3-Pin Non-Reversible Cable

Purpose

A photo-interrupter consists of two parts: transmitter and receiver. The transmitter (e.g., an LED or a laser) emits light which then reaches the receiver. If the light between them is interrupted even for a moment, the receiver will detect an incident and change the output level accordingly.

The schematic diagram of the module is as shown below:

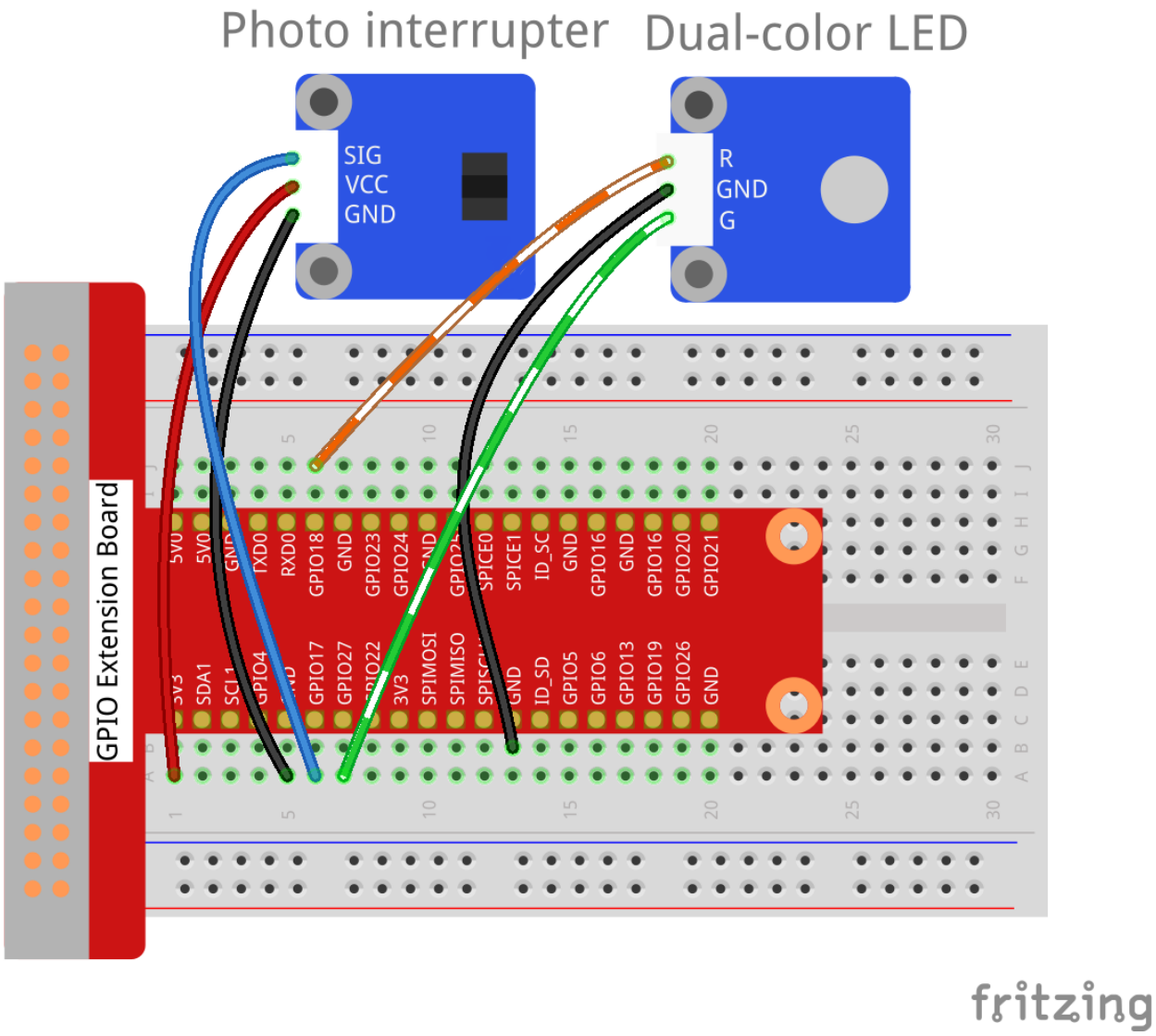


Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Photo-interrupter Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND

Raspberry Pi	GPIO Extension Board	Dual-color LED Module
GPIO1	GPIO18	R
GND	GND	GND
GPIO2	GPIO27	G



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/12_photo_interrupter/
```

Step 3: Compile.

```
gcc photo_interrupter.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

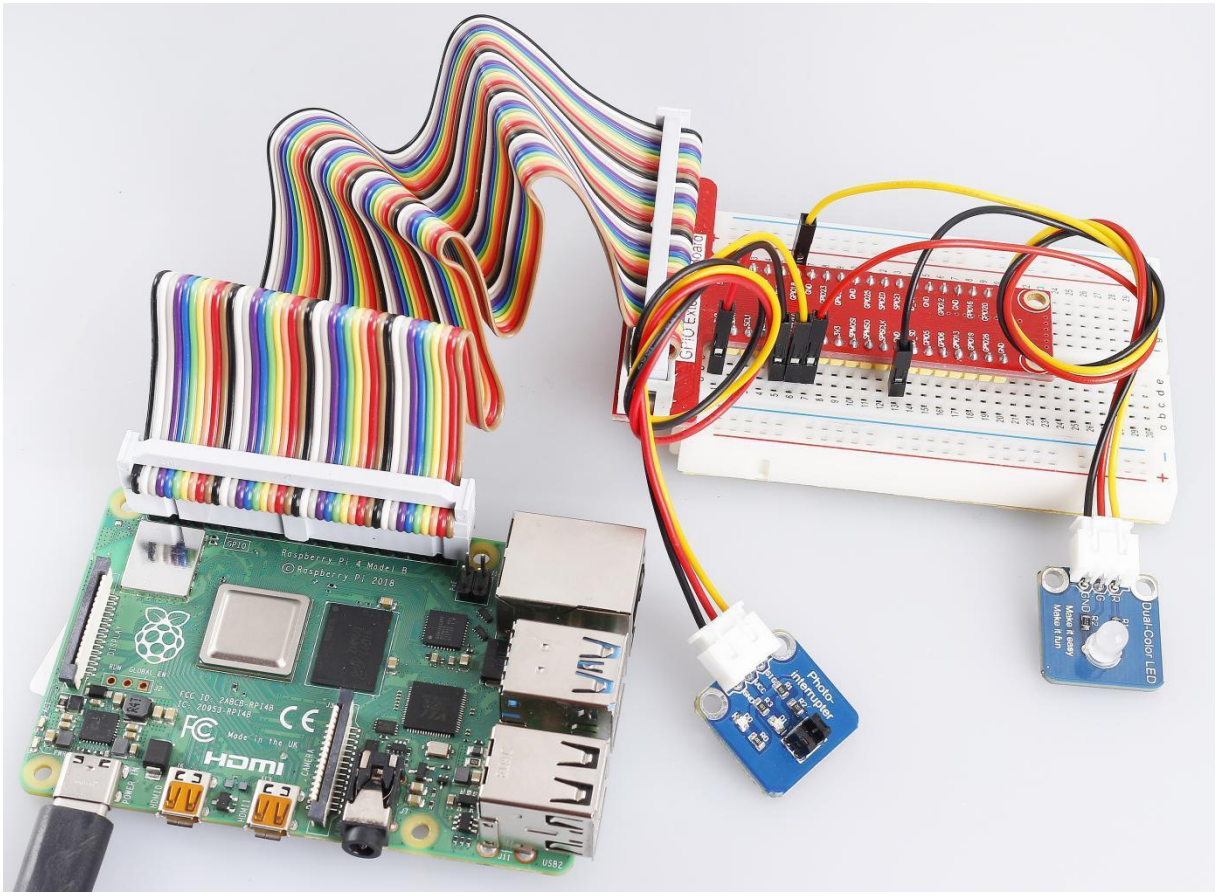
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 12_photo_interrupter.py
```

The LED will light up green. Use a piece of paper in the gap between the photo interrupter. "Light was blocked" will be printed to the screen and the LED will flash red until the paper is removed.

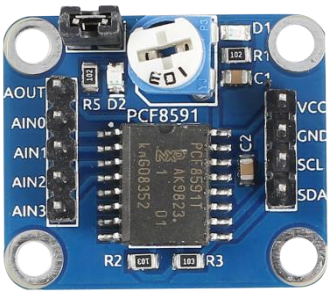

Raspberry Pi 4 Model B
Photo-interrupter
Dual-Color LED

Lesson 13 PCF8591

Introduction

The PCF8591 is a single-chip, single-supply low-power 8-bit CMOS data acquisition device with four analog inputs, one analog output and a serial I2C-bus interface. Three address pins A0, A1 and A2 are used for programming the hardware address, allowing the use of up to eight devices connected to the I2C-bus without additional hardware. Address, control and data to and from the device are transferred serially via the two-line bidirectional I2C-bus.

The functions of the device include analog input multiplexing, on-chip track and hold function, 8-bit analog-to-digital conversion and an 8-bit digital-to-analog conversion. The maximum conversion rate is given by the maximum speed of the I2C-bus.



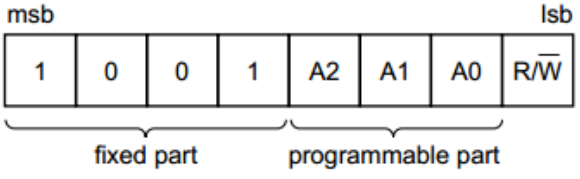
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * PCF8591 module
- 1 * Dual-Color LED module
- 1 * 3-Pin Non-Reversible Cable
- Jumper Wires

Purpose

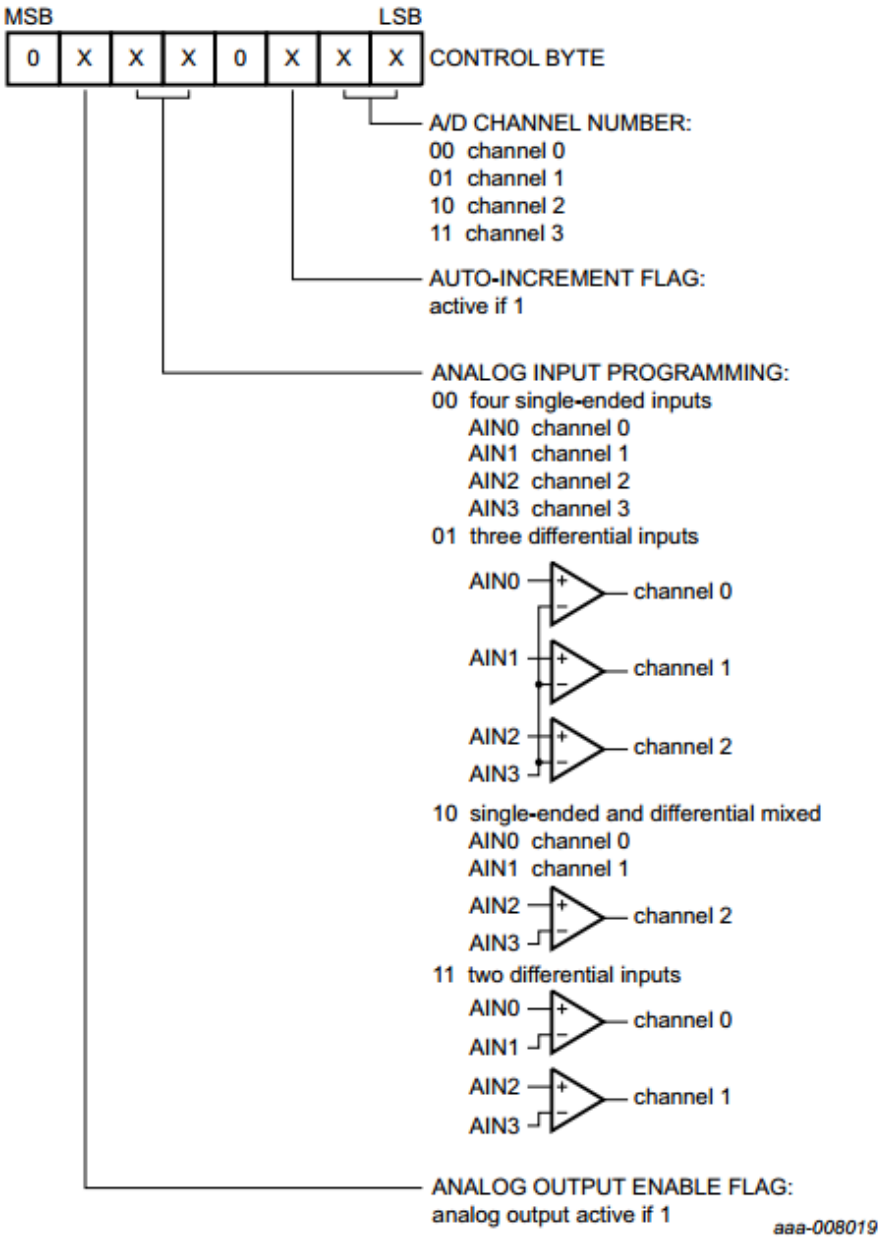
Addressing:

Each PCF8591 device in an I2C-bus system is activated by sending a valid address to the device. The address consists of a fixed part and a programmable part. The programmable part must be set according to the address pins A0, A1 and A2. The address always has to be sent as the first byte after the start condition in the I2C-bus protocol. The last bit of the address byte is the read/write-bit which sets the direction of the following data transfer (see below).



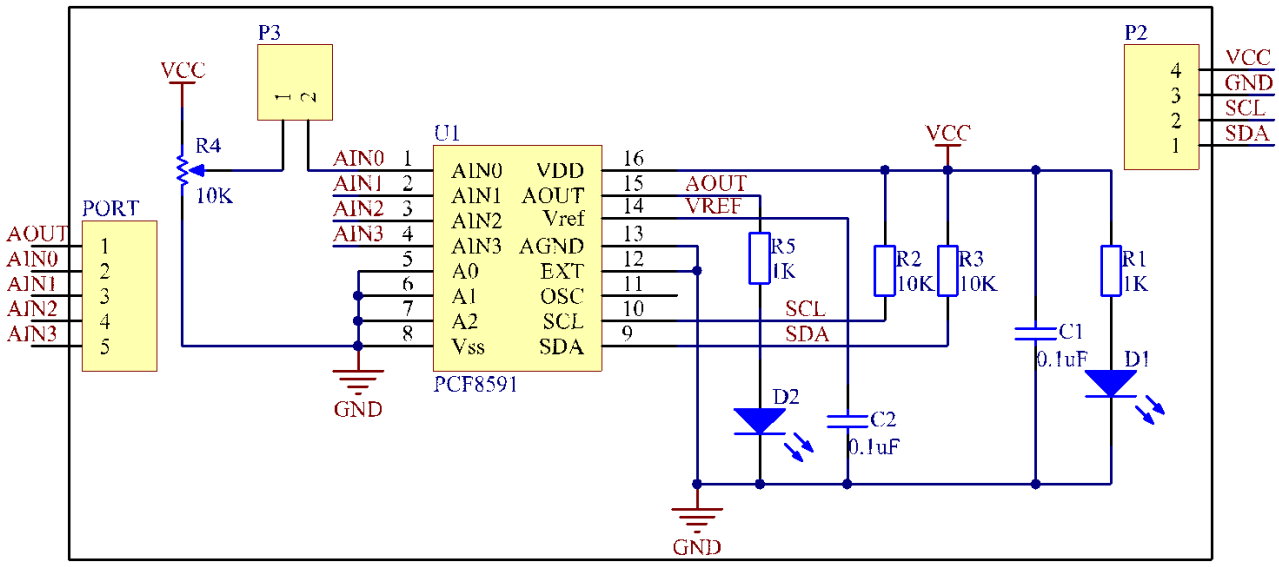
Control byte:

The second byte sent to a PCF8591 device will be stored in its control register and is required to control the device function. The upper nibble of the control register is used for enabling the analog output, and for programming the analog inputs as single-ended or differential inputs. The lower nibble selects one of the analog input channels defined by the upper nibble (see Fig.5). If the auto-increment flag is set, the channel number is incremented automatically after each A/D conversion. See the figure below.



In this experiment, the AIN0 (Analog Input 0) port is used to receive analog signals from the potentiometer module, and AOUT (Analog Output) is used to output analog signals to the dual-color LED module so as to change the luminance of the LED.

The schematic diagram:



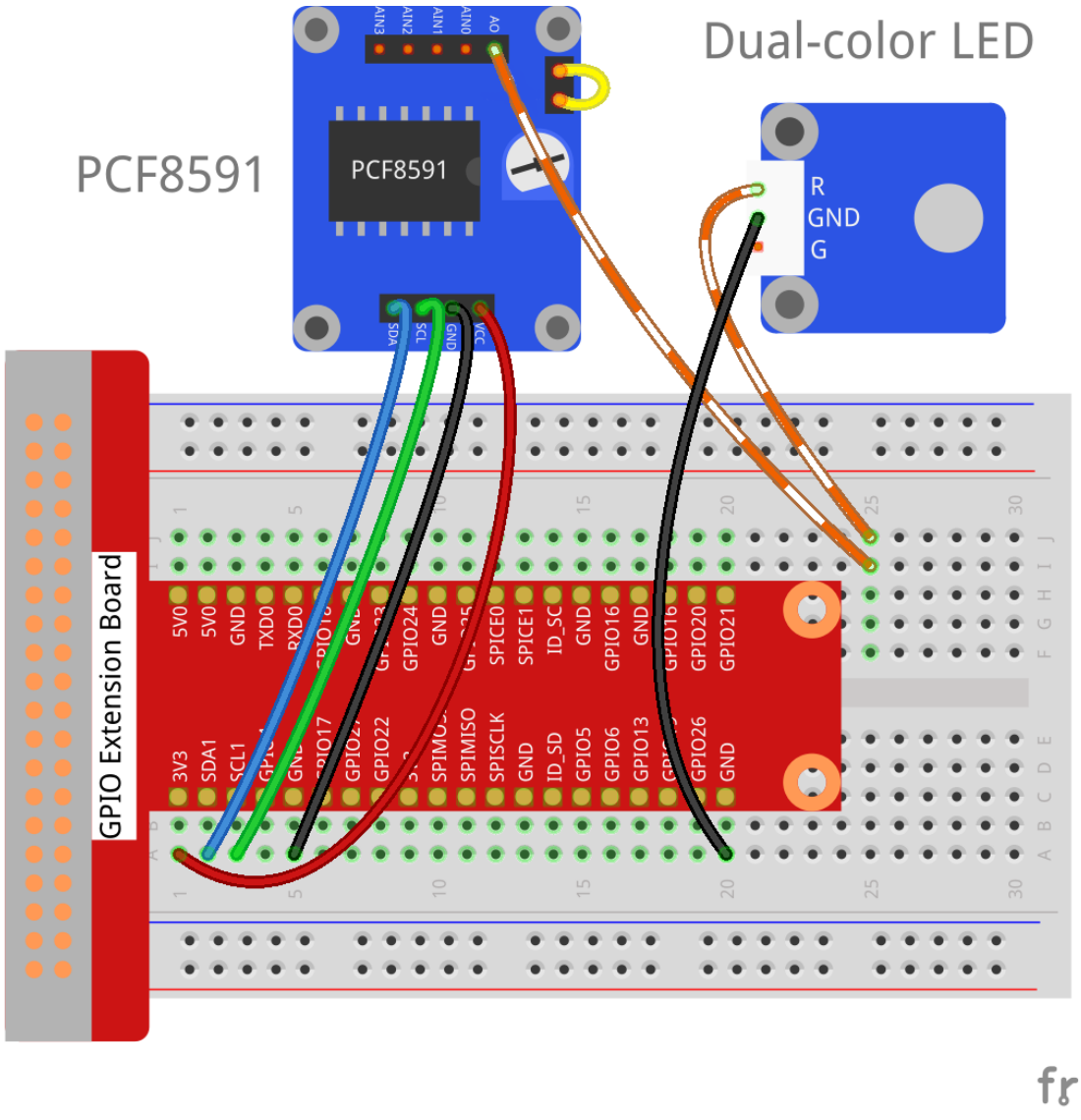
Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	PCF8591 Module
SDA	SDA1	SDA
SCL	SCL1	SCL
3.3V	3V3	VCC
GND	GND	GND

Dual-Color Module	GPIO Extension Board	PCF8591 Module
R	*	AOUT
GND	GND	GND
G	*	*

Note:	Connect the two pins next to the potentiometer of the PCF8591 module with the jumper cap attached.



Step 2: Setup I2C (see **Appendix**. If you have set I2C, skip this step.)

For C Users:

Step 3: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/13_pcf8591/
```

Step 4: Compile.

```
gcc pcf8591.c -lwiringPi
```

Step 5: Run.

```
sudo ./a.out
```

For Python Users:

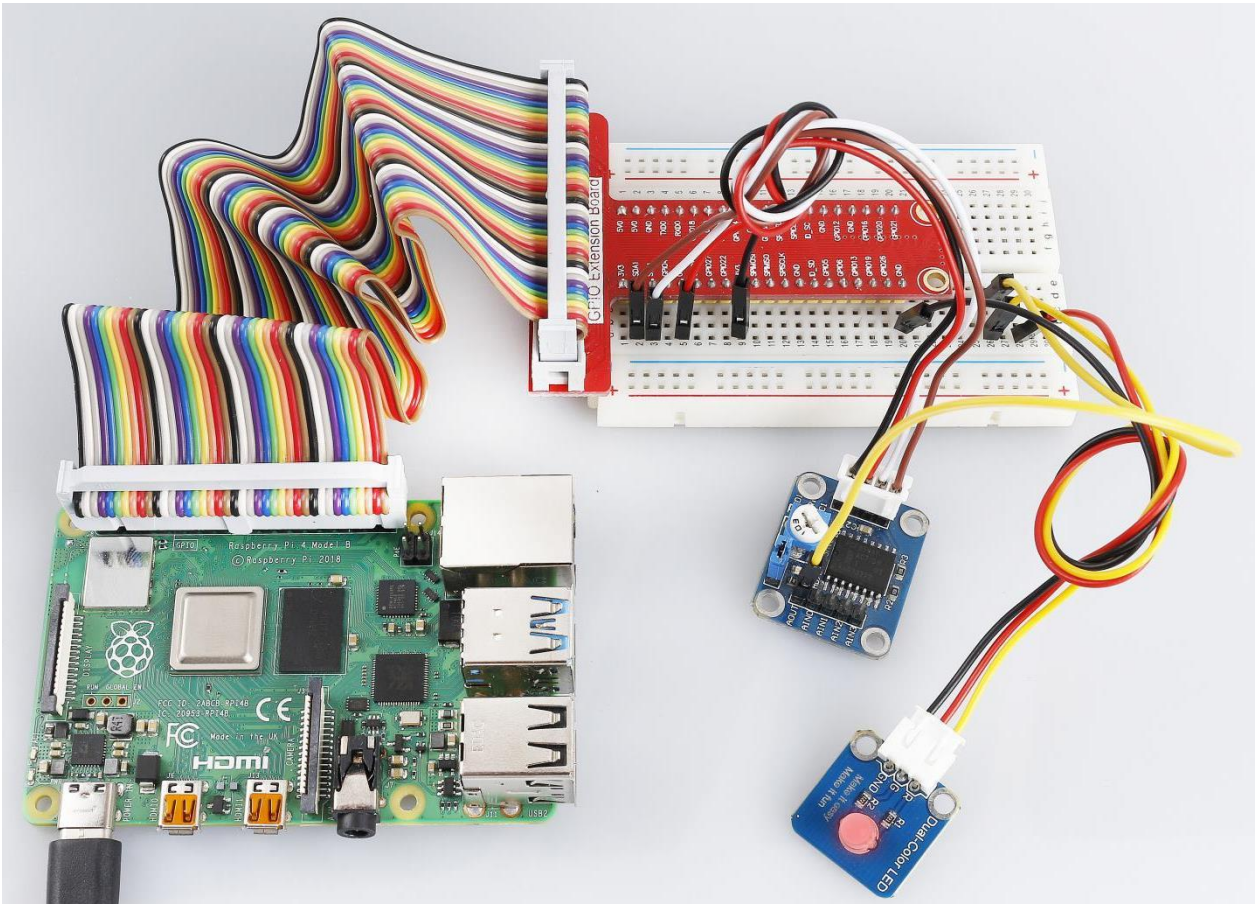
Step 3: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 4: Run.

```
sudo python3 13_pcf8591.py
```

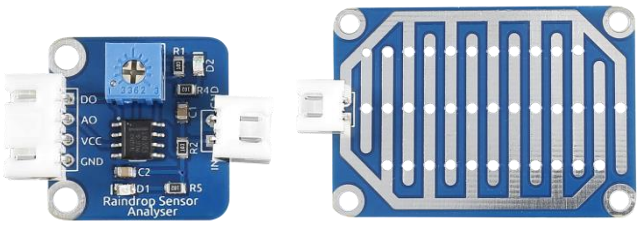
Now, turn the knob of the potentiometer on PCF8591, and you can see the luminance of the LED change and a value between 0 and 255 printed on the screen.



Lesson 14 Rain Detection Module

Introduction

The rain detection module can be used in the open air to detect raindrops and send a signal to the Raspberry Pi.



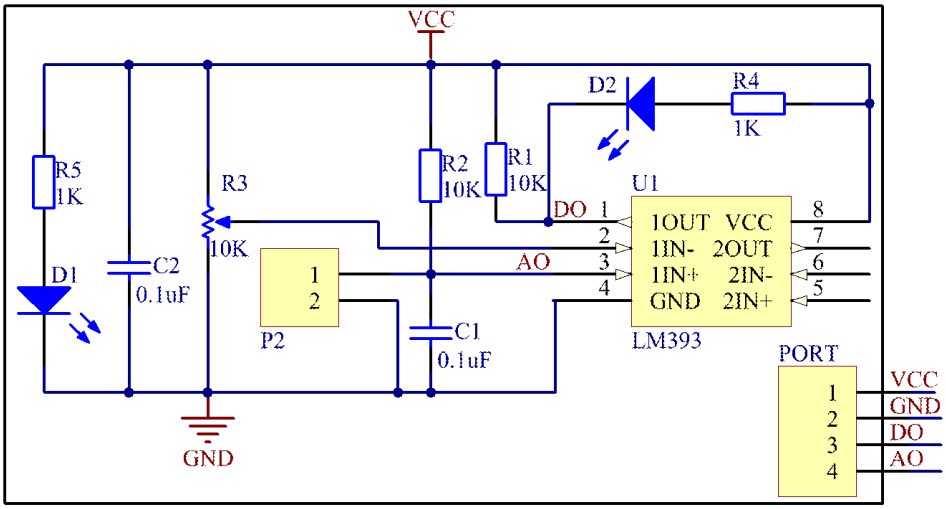
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Rain Detection module
- 1 * PCF8591
- 1 * LM393
- 1 * 2-Pin ribbon cable
- 1 * 4-Pin Non-Reversible Cable
- Jumper Wires

Purpose

The rain detection module is composed of metal wires which run parallel but do not make contact. When a raindrop lands on the board, the metal wires will conduct and this voltage between them will trigger the board to send a signal to the Raspberry Pi.

The schematic diagram of the module is as shown below:



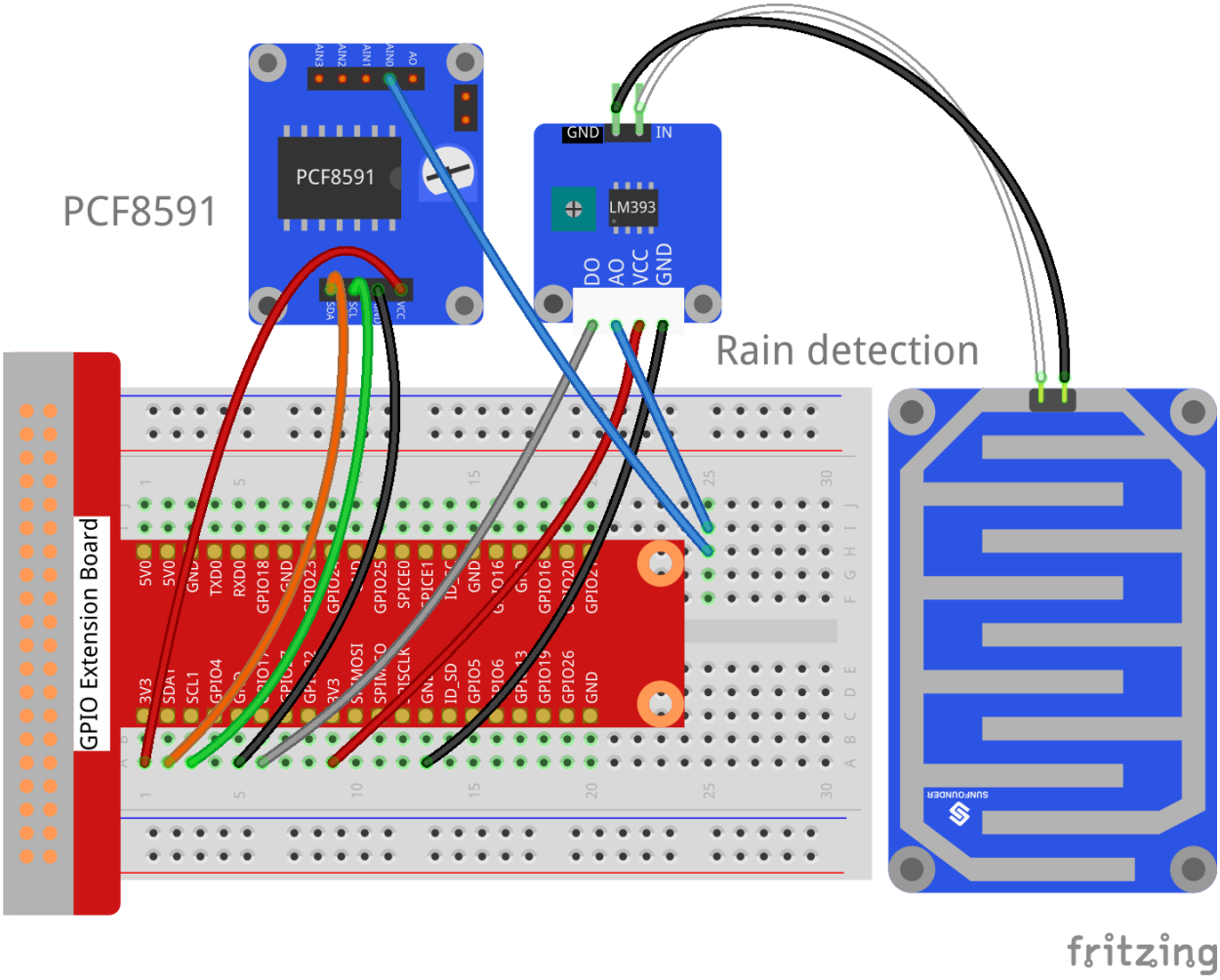
Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	PCF8591 Module
SDA	SDA1	SDA
SCL	SCL1	SCL
3.3V	3V3	VCC
GND	GND	GND

LM393	GPIO Extension Board	PCF8591 Module
DO	GPIO17	*
AO	*	AIN0
VCC	3V3	VCC
GND	GND	GND

Note:	The two pins on the rain detection board are exactly the same. You can connect them to pin IN and GND on LM393.



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/14_rain_detector/
```

Step 3: Compile.

```
gcc rain_detector.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

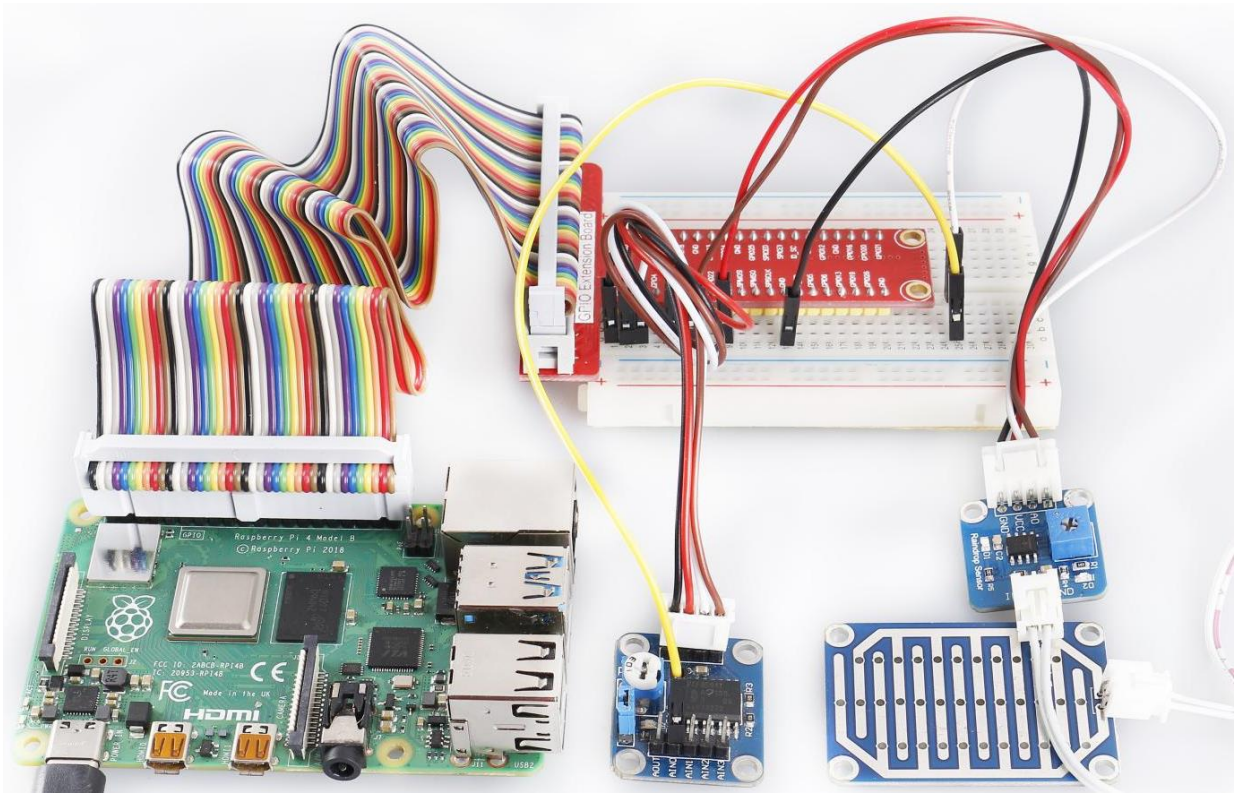
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 14_rain_detector.py
```

Now drop some water onto the rain detection board until "**raining**" displayed on the screen. You can adjust the potentiometer on LM393 to detect the threshold of rainfall.


GPIO Extension Board
Raspberry Pi 4 Model B
©Raspberry Pi 2018
HDMI
VCC
GND
DO
AO

Lesson 15 Joystick PS2

Introduction

This joystick module takes five types of input: up, down, left, right, and press-down.

Required Components

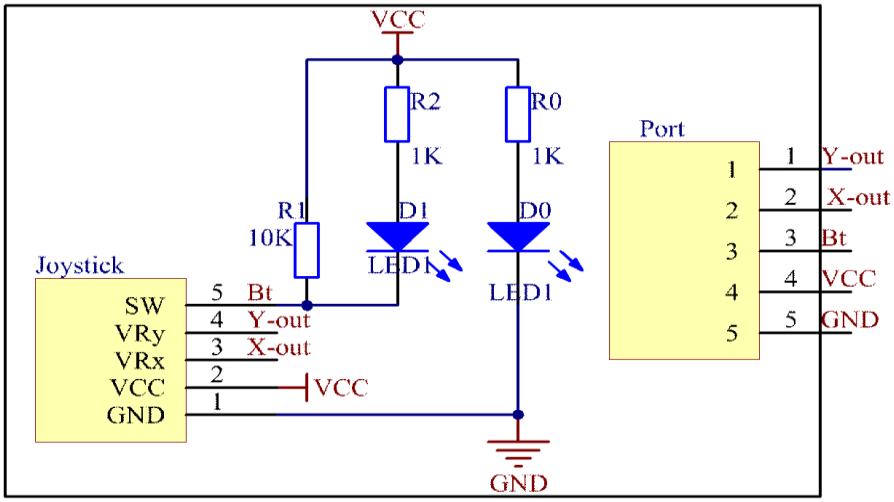
- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * PCF8591
- 1 * Joystick PS2 module
- 1 * 5-Pin Non-Reversible Cable
- Jumper Wires

Purpose

This module has two analog outputs (corresponding to X and Y coordinates) and one digital output representing whether it is pressed on the Z axis.

In this experiment, the X and Y pins are connected to the analog input ports of the A/D converter to convert the signals into digital ones. The Raspberry is programmed to detect the movements of the Joystick.

The schematic diagram of the module is as shown below:

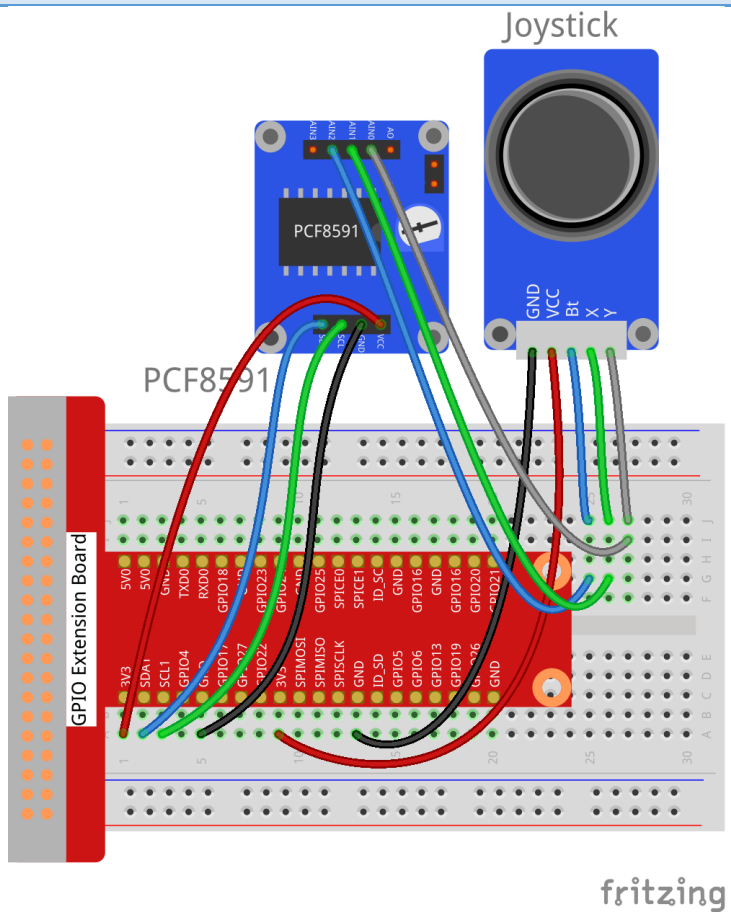


Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	PCF8591 Module
SDA	SDA1	SDA
SCL	SCL1	SCL
3.3V	3V3	VCC
GND	GND	GND

Joystick PS2	GPIO Extension Board	PCF8591 Module
Y	*	AIN0
X	*	AIN1
Bt	*	AIN2
VCC	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/15_joystick_PS2/
```

Step 3: Compile.

```
gcc joystick_PS2.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

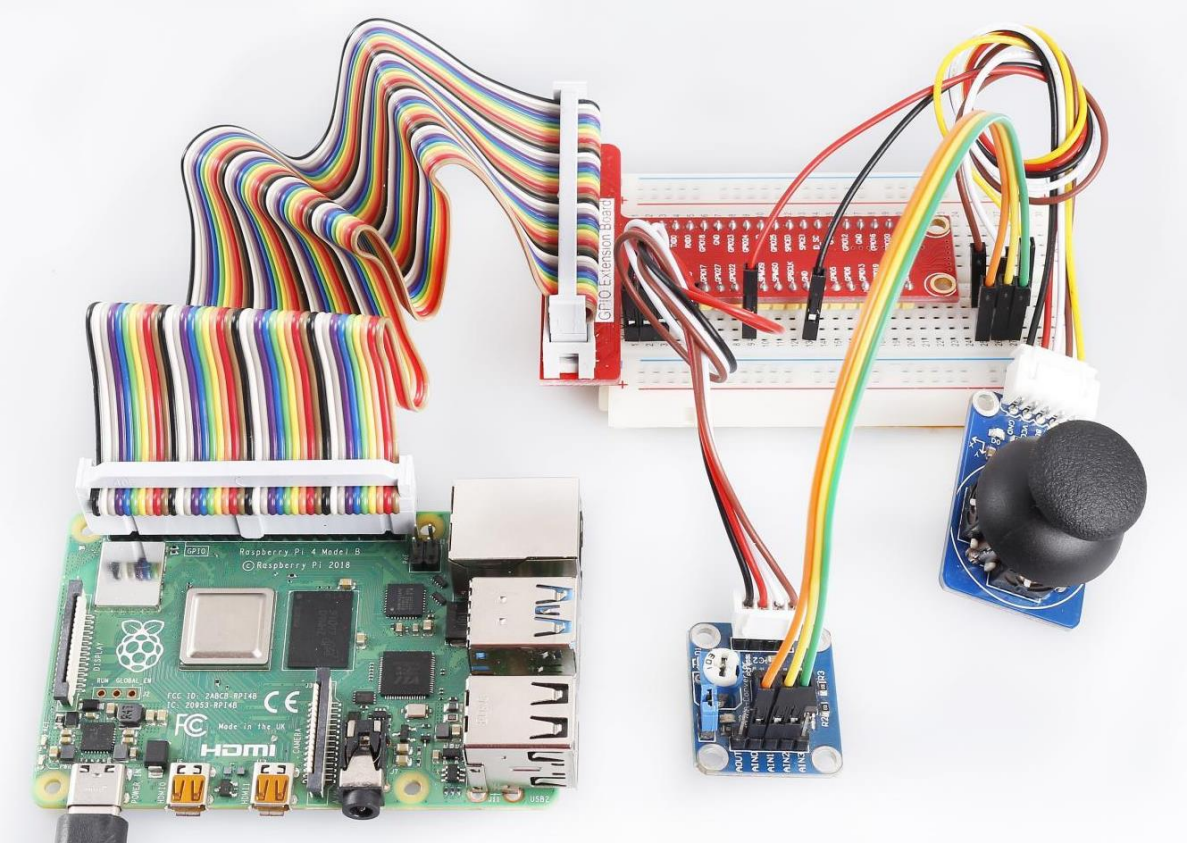
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 15_joystick_PS2.py
```

Text will now be printed to the screen to show which direction the joystick is being moved in and the words "Button Pressed" will print when the joystick is pressed down.



Lesson 16 Potentiometer Module

Introduction

A potentiometer is a device which is used to vary the resistance in an electrical circuit without interrupting the circuit.

Required Components

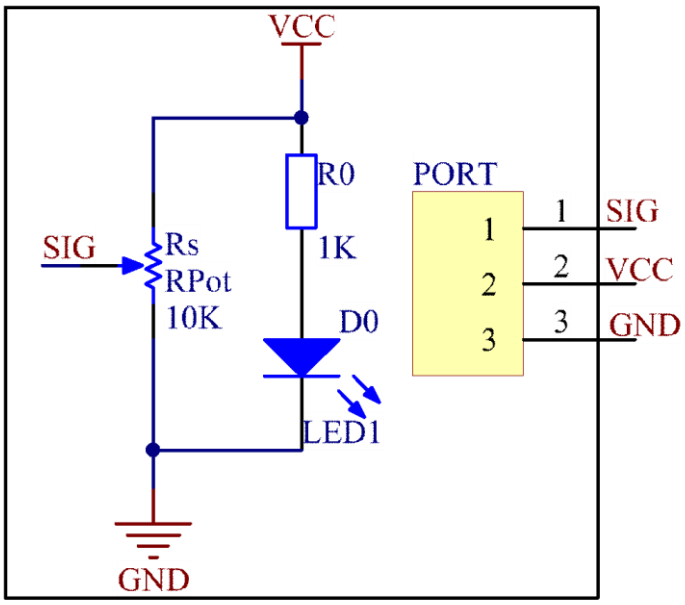
- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Potentiometer module
- 1 * Dual-Color LED module
- 2 * 3-Pin Non-Reversible Cable
- Jumper Wires

Purpose

This potentiometer is an analog electrical component. What's the difference between an analog and digital potentiometer? Simply put, a digital potentiometer refers to two states like on/off, high/low levels, i.e. either 0 or 1, while an analog one supports signals like a number from 1 to 1000. The signal value changes over time instead of keeping an exact number. Analog signals can include light intensity, humidity, temperature, and so on.

In this experiment, the PCF8591 module is used to read the analog value of the potentiometer and output it digitally as a value which will then be shown by the LED. Connect pin SIG of the potentiometer to pin AIN0 of PCF8591 and pin R or Pin G on the Dual-Color LED module to pin AOUT of PCF8591.

The schematic diagram of the module is as shown below:



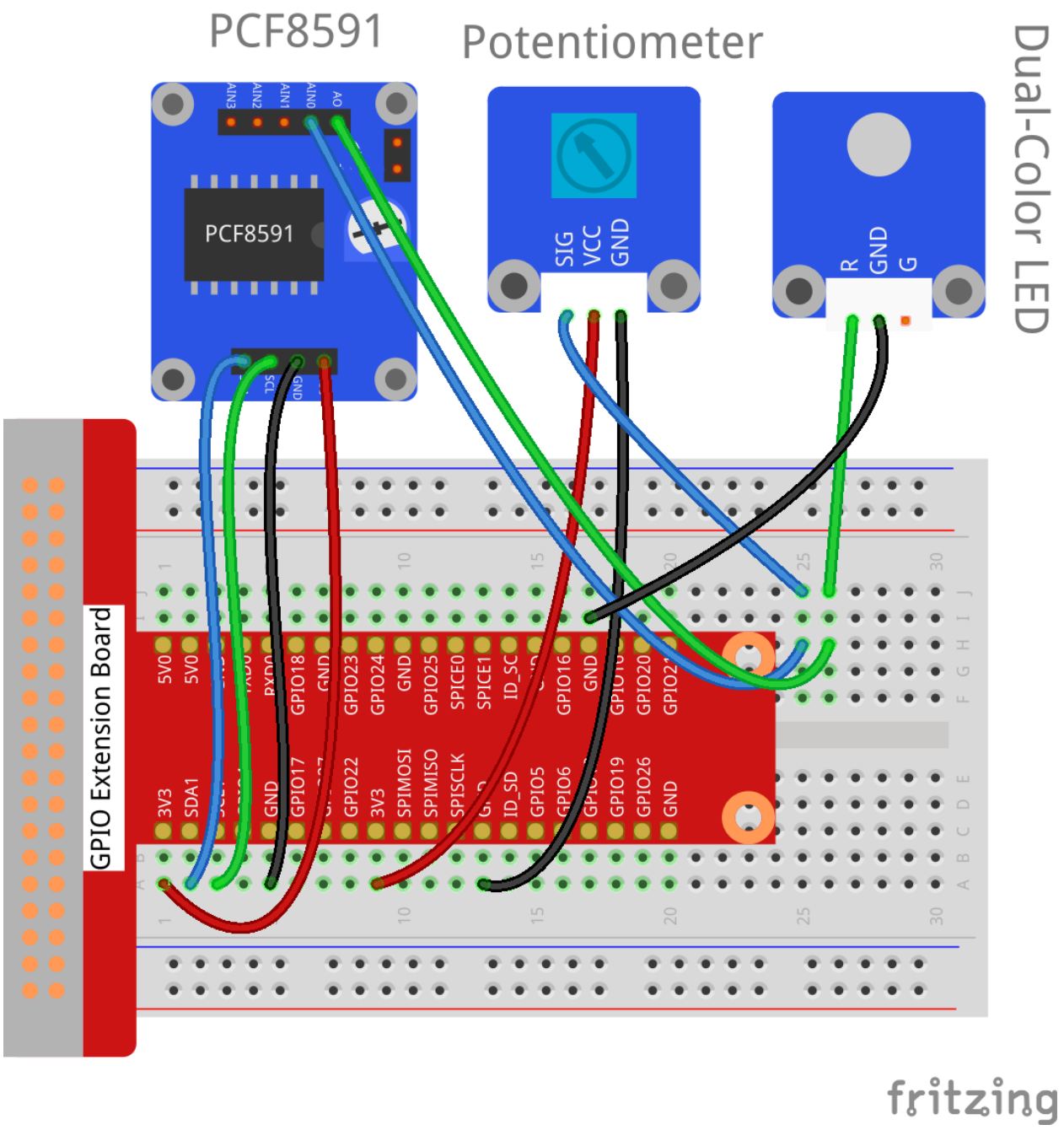
Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	PCF8591 Module
SDA	SDA1	SDA
SCL	SCL1	SCL
3.3V	3V3	VCC
GND	GND	GND

Potentiometer	GPIO Extension Board	PCF8591 Module
SIG	*	AIN0
VCC	3V3	VCC
GND	GND	GND

Dual-Color Module	GPIO Extension Board	PCF8591 Module
R	*	AOUT
GND	GND	GND
G	*	*



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/16_potentiometer/
```

Step 3: Compile.

```
gcc potentiometer.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

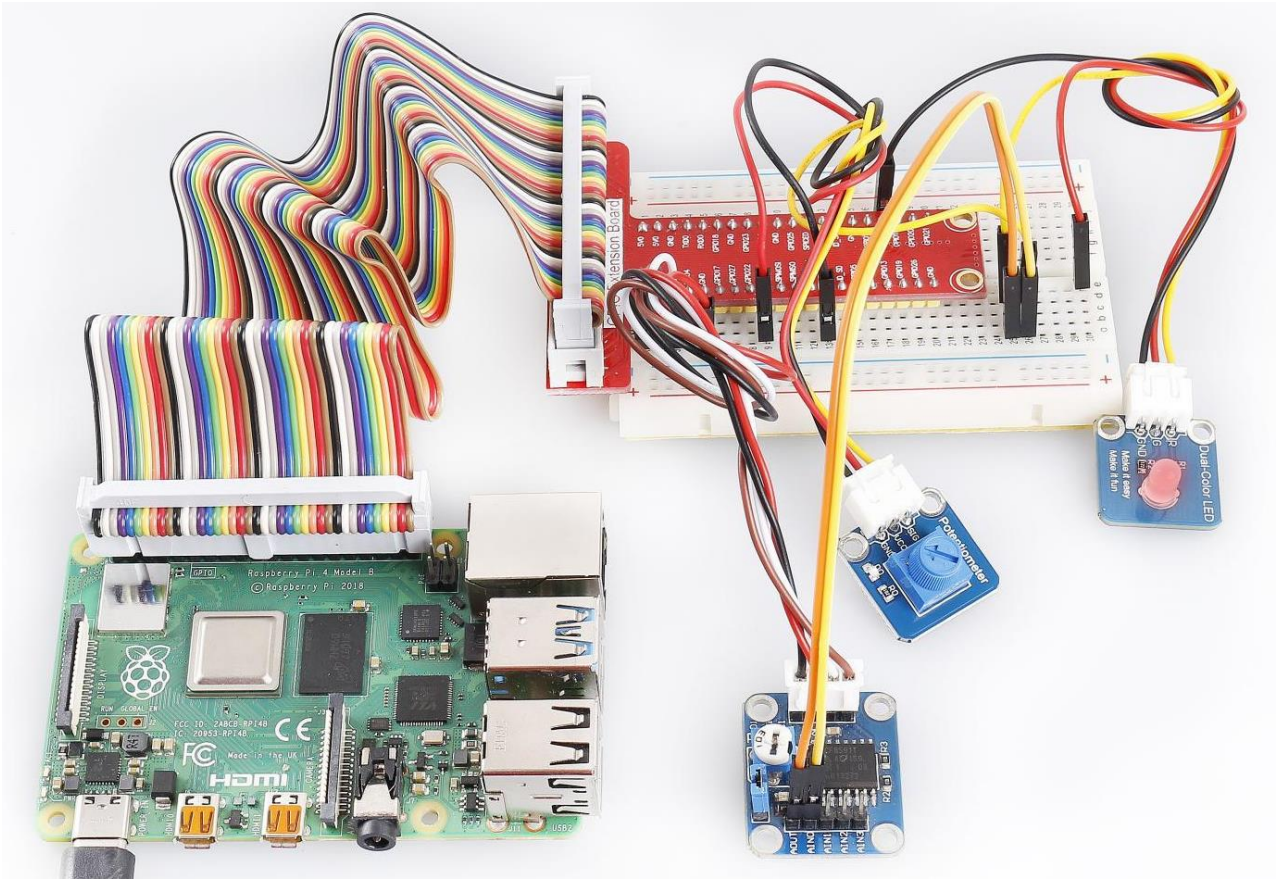
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 16_potentiometer.py
```

Turn the knob of the potentiometer, and you can see the value printed on the screen change from 0 (minimum) to 255 (maximum).



Lesson 17 Hall Sensor

Introduction

A Hall Sensor (named for the Hall Effect) varies its output voltage in response to a magnetic field. They are used for proximity switching, positioning, speed detection, and current sensing applications.

Hall sensors can be categorized into linear (analog) sensors and switch (digital) sensors. A switch sensor outputs a digital signal and consists of a voltage regulator, Hall element, differential amplifier, Schmitt trigger, and output terminal. The linear sensor outputs an analog signal and consists of a Hall element, linear amplifier, and emitter follower. If a comparator is added to a linear sensor, it becomes able to output both analog and digital signals.

Required Components

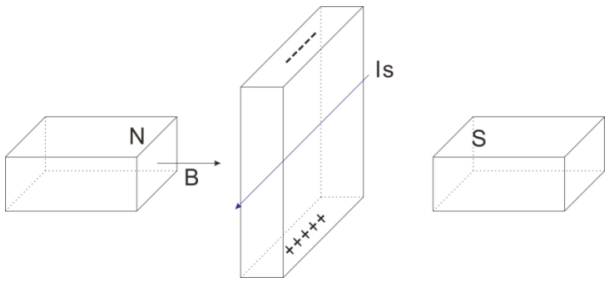
- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Analog Hall Switch module
- 1 * Dual-color LED module
- 1 * Switch hall module
- 1 * PCF8591
- 2 * 3-Pin Non-Reversible Cable
- 1 * 4-Pin Non-Reversible Cable
- Jumper Wires
- A Magnet (Not Included)

Purpose

Hall Effect

The Hall Effect is a type of electromagnetic effect discovered by Edwin Hall in 1879 while researching metal conductivity mechanisms. The effect is seen when a conductor is passed through a uniform magnetic field. The natural electron drift of the charge carrier causes the magnetic field to apply a Lorentz force (the force exerted on a charged particle in an

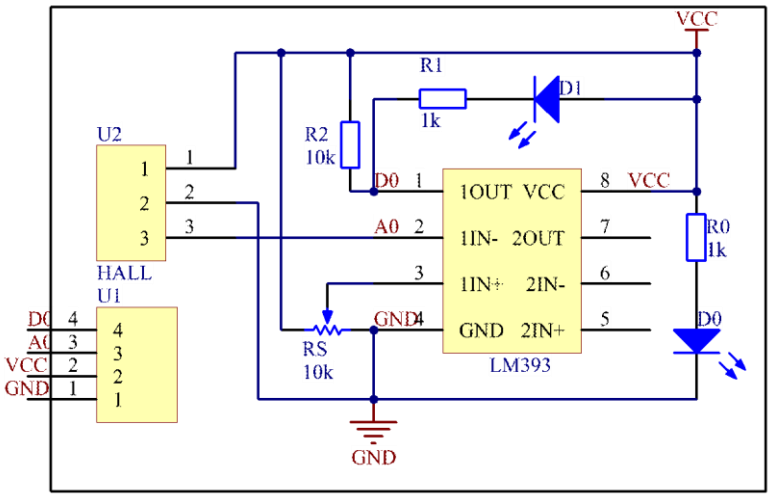
electromagnetic field). The result is a charge separation, with a buildup of either positive or negative charges on the bottom or top of the plate.



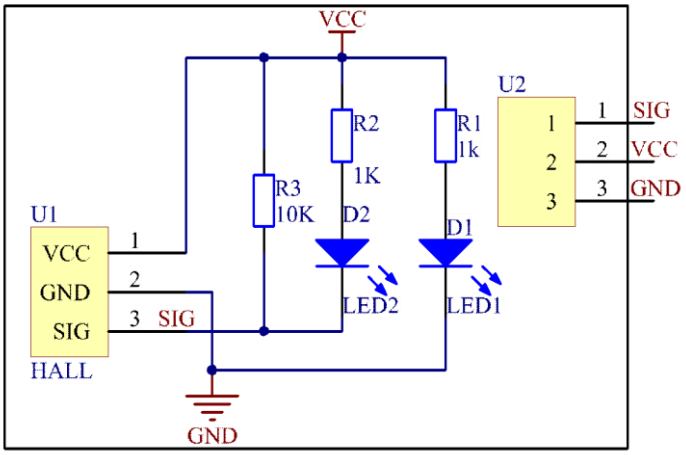
Hall sensor

Electricity carried through a conductor will produce a magnetic field that varies with current, and a Hall sensor can be used to measure the current without interrupting the circuit. Typically, the sensor is integrated with a wound core or permanent magnet that surrounds the conductor to be measured.

The schematic diagram of the analog Hall sensor module:



The schematic diagram of the Switch hall module:



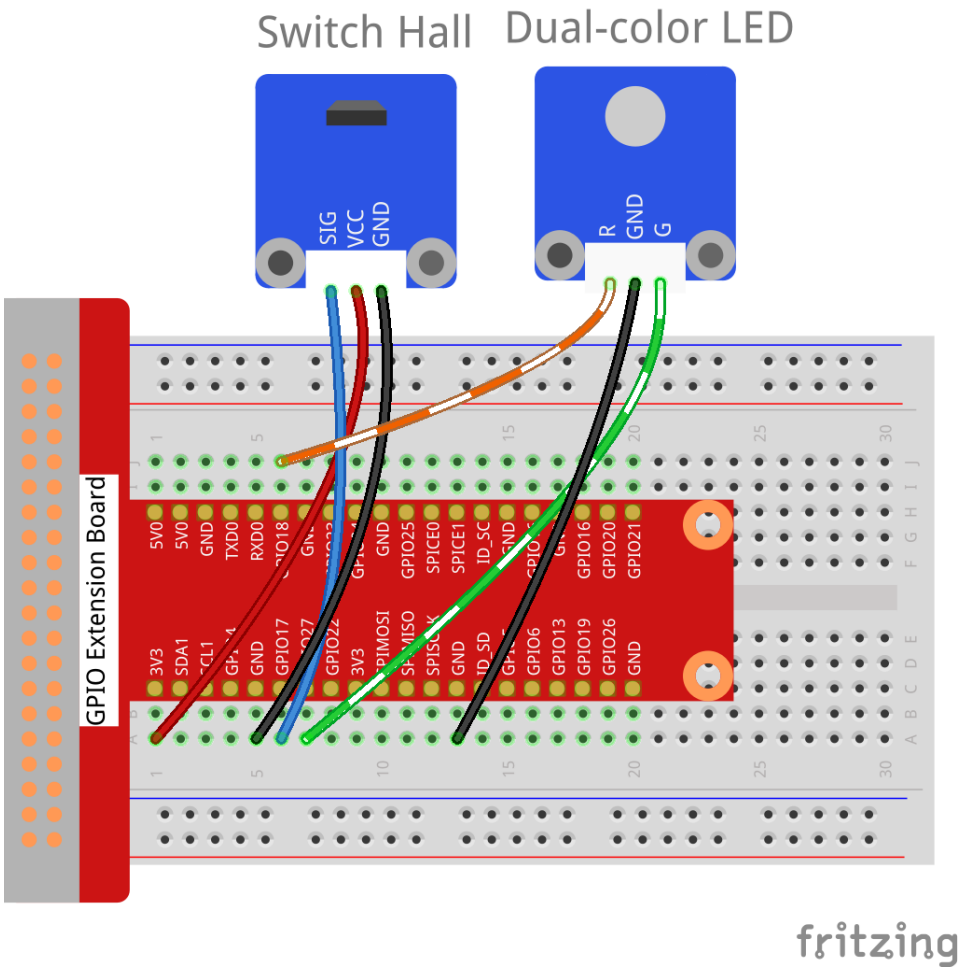
Procedure

For the Switch Hall Sensor:

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Switch Hall Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND

Raspberry Pi	GPIO Extension Board	Dual-color LED Module
GPIO1	GPIO18	R
GND	GND	GND
GPIO2	GPIO27	G



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/17_switch_hall/
```

Step 3: Compile.

```
gcc switch_hall.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

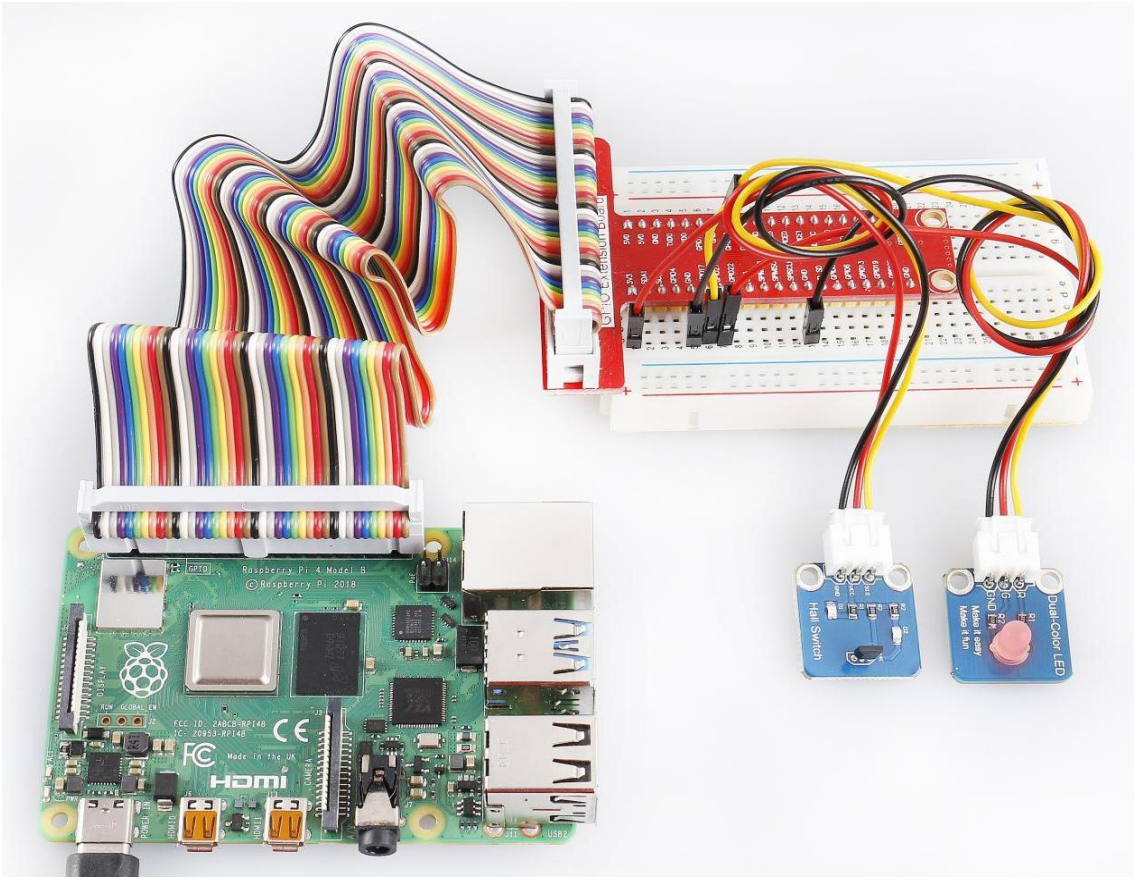
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 17_switch_hall.py
```

Place a magnet close to the Hall sensor. The string "Detected magnetic materials" will print to the screen and the LED will light up.

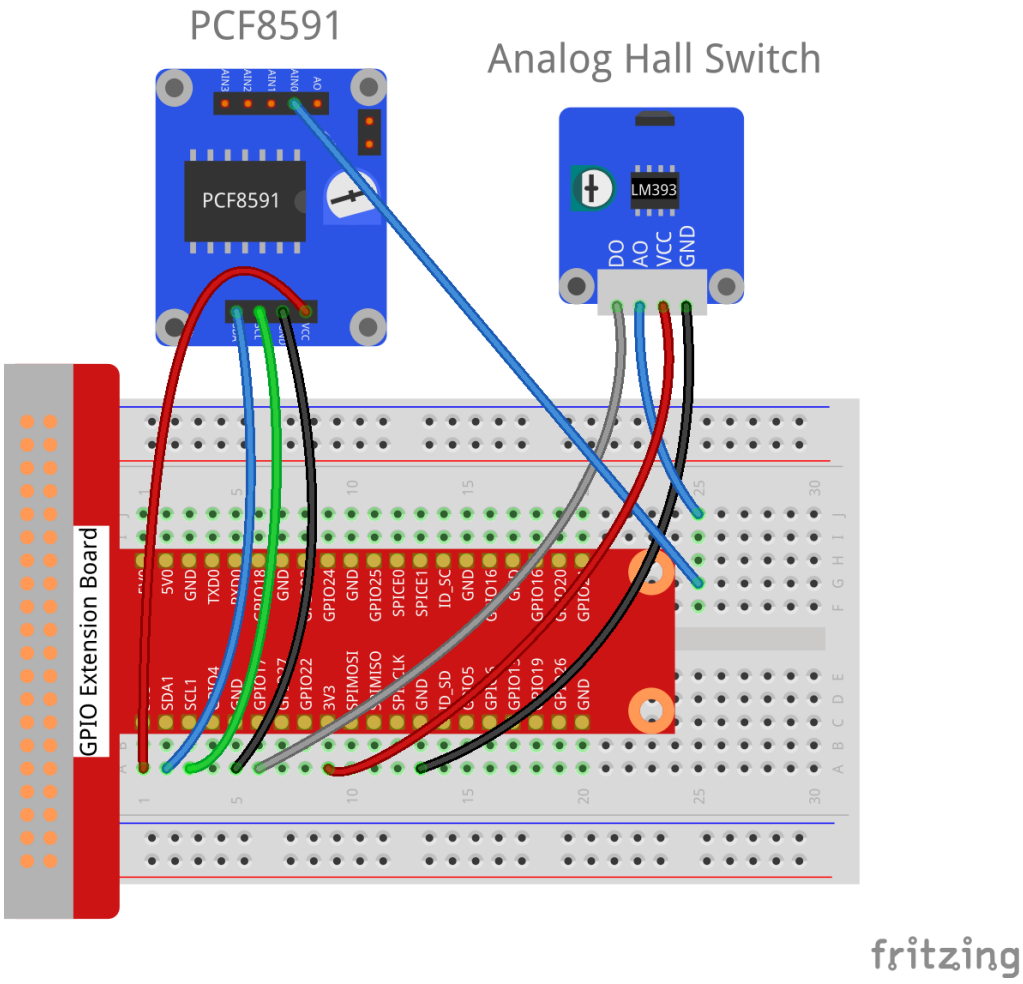


For the Analog Hall Switch:

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	PCF8591 module
SDA	SDA1	SDA
SCL	SCL1	SCL
3.3V	3V3	VCC
GND	GND	GND

Analog Hall Switch	GPIO Extension Board	PCF8591 module
DO	GPIO17	*
AO	*	AIN0
VCC	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/17_analog_hall_switch/
```

Step 3: Compile.

```
gcc analog_hall_switch.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

Step 2: Change directory.

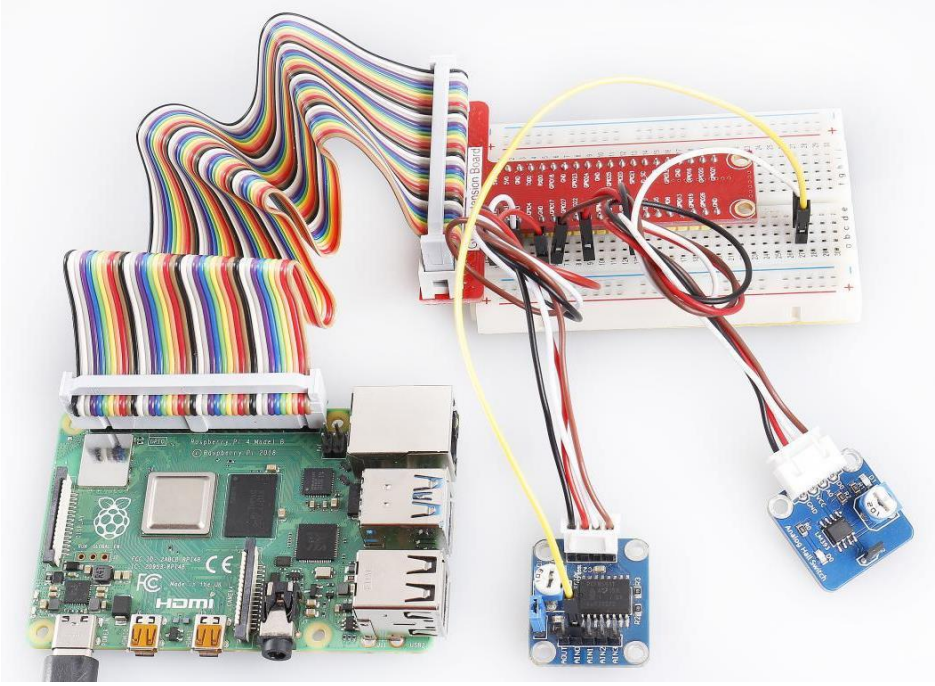
```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 17_analog_hall_switch.py
```

A message will display to the screen indicating the intensity of the magnetic field. When a magnet is used, the direction facing the sensor (North or South) will print to the screen.

Note:	Pin D0 of the Analog Hall Sensor will output "0" only when the south pole of the magnet approaches it, otherwise it will output "1".

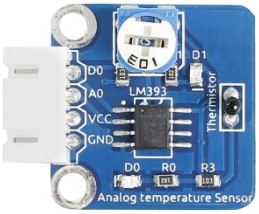


Lesson 18 Temperature Sensor

Introduction

A temperature sensor is a component that takes a temperature and outputs it as a signal. They can be divided into two types: thermal resistors (thermistors) or thermocouplers. A thermistor is made of semiconductor materials and usually uses a negative temperature coefficient (NTC) in which the resistance decreases as the temperature rises. Thermistors are considered the more sensitive sensor of the two types.

Two types of thermistor sensor modules are included with this kit:



Analog temperature sensor

Thermistor

Required Components

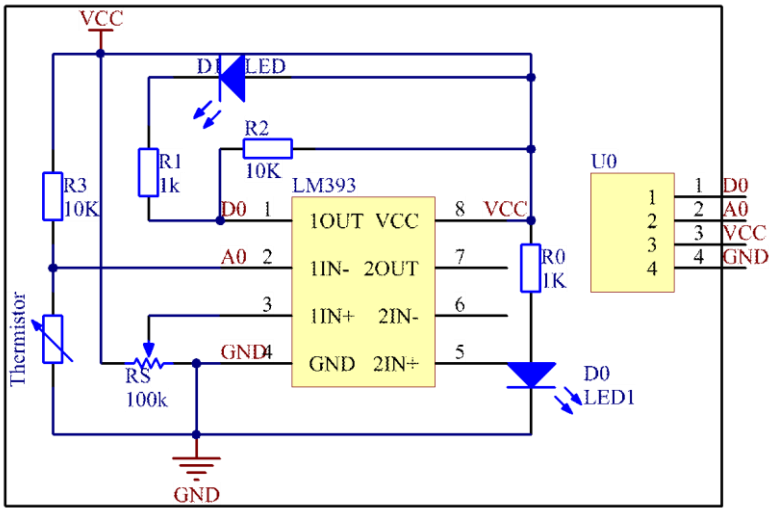
- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Analog-temperature Sensor module
- 1 * Thermistor module
- 1 * PCF8591
- 1 * 3-Pin Non-Reversible Cable
- 1 * 4-Pin Non-Reversible Cable
- Jumper Wires

Purpose

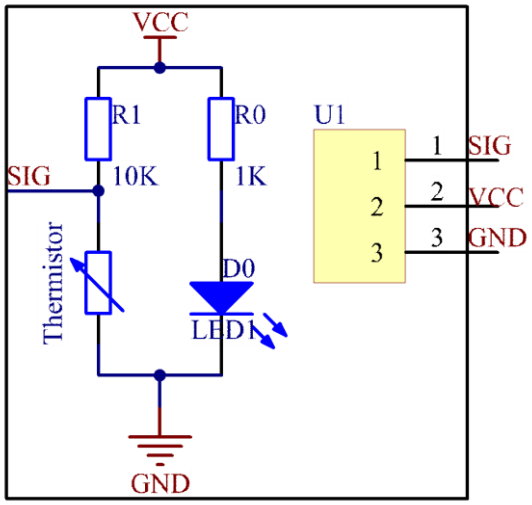
These thermistor modules can detect real-time temperature changes. When the ambient temperature increases, the resistance of the thermistor module will decrease.

In this experiment, we use an analog-digital converter PCF8591 to convert analog signals into digital ones.

The schematic diagram for analog temperature sensor:



The schematic diagram for the thermistor module:



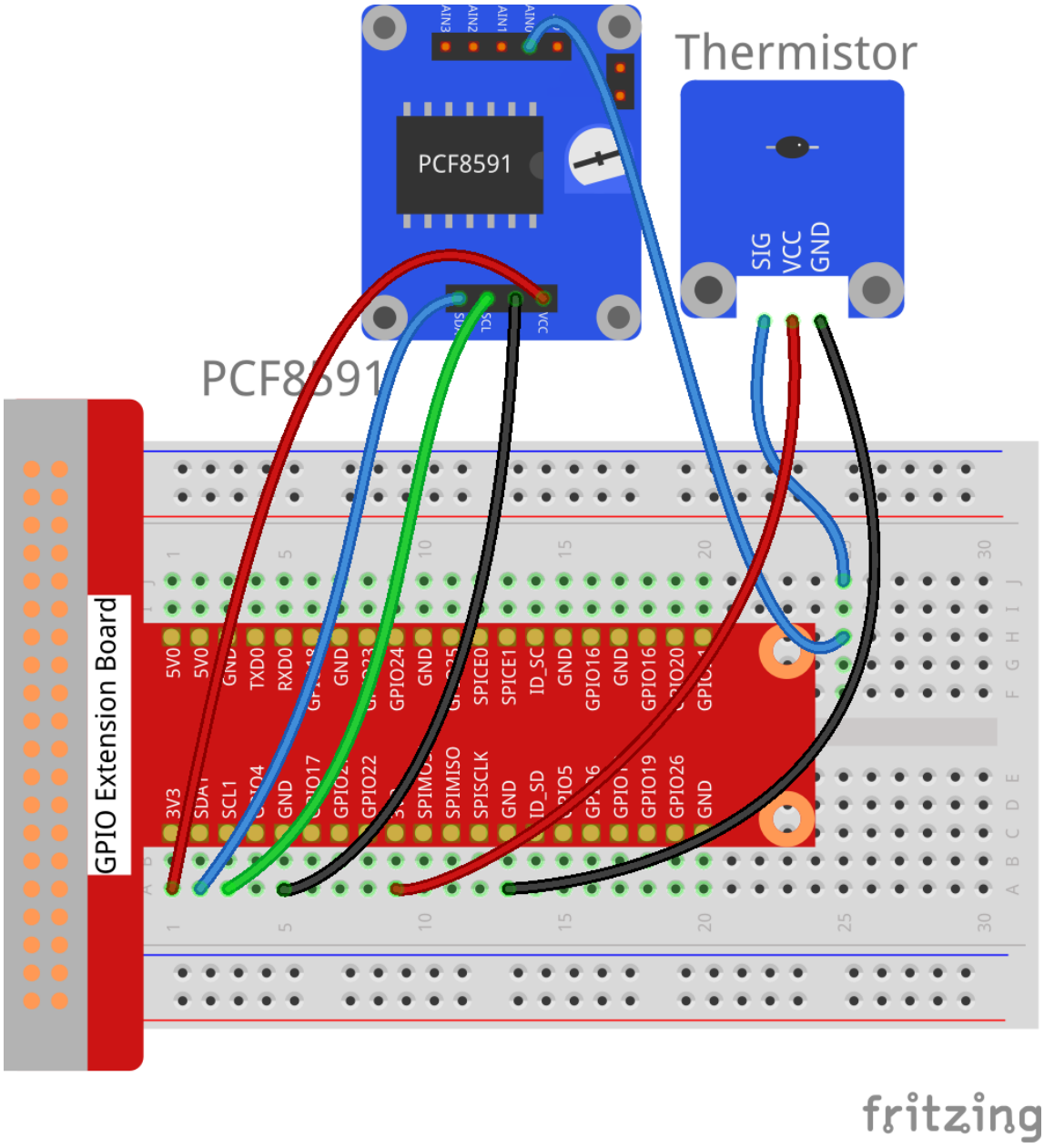
Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	PCF8591 Module
SDA	SDA1	SDA
SCL	SCL1	SCL
3.3V	3V3	VCC
GND	GND	GND

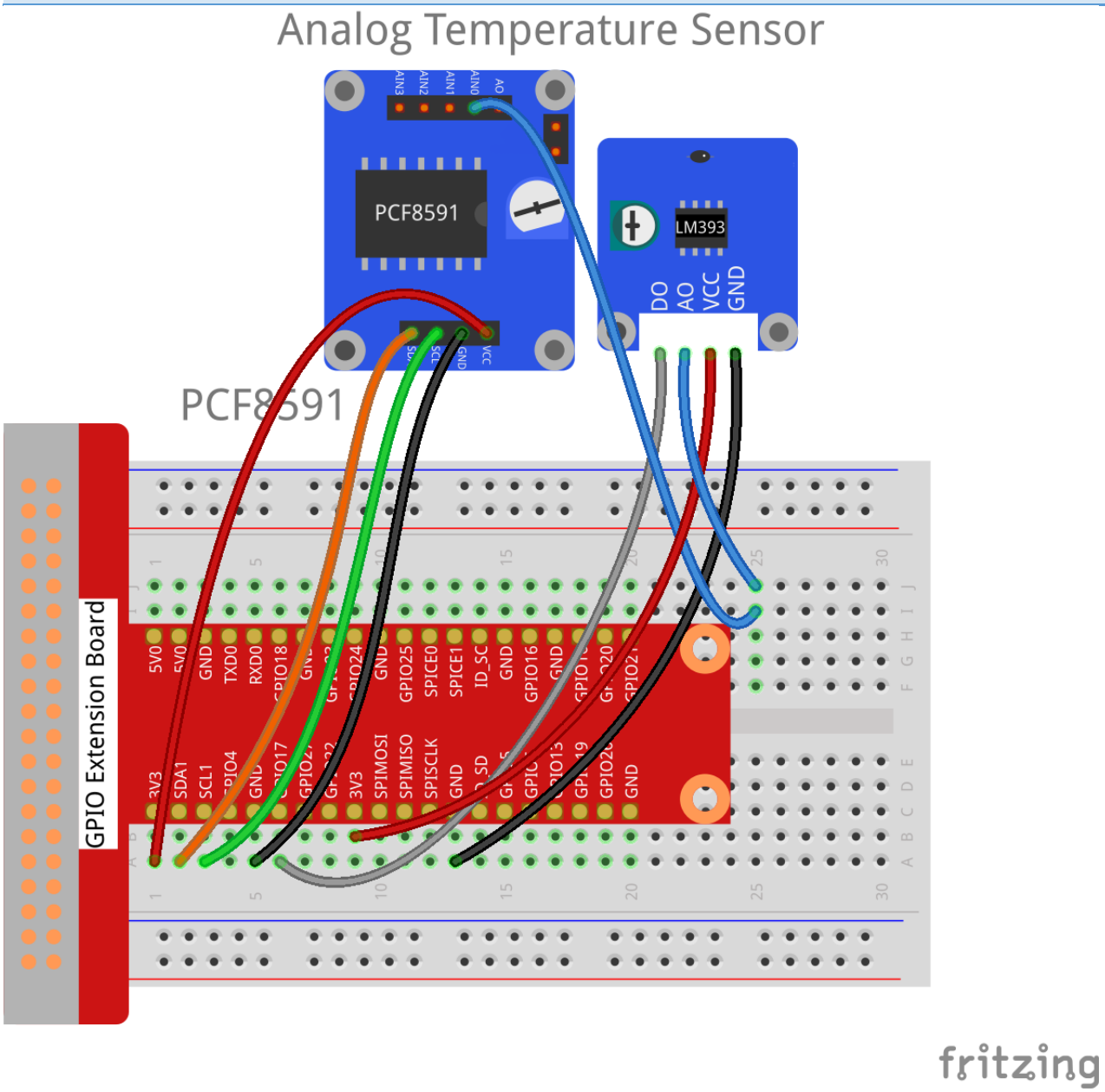
For thermistor module:

Thermistor Module	GPIO Extension Board	PCF8591 Module
SIG	*	AIN0
VCC	3V3	VCC
GND	GND	GND



For analog temperature sensor module:

Analog Temperature Module	GPIO Extension Board	PCF8591 Module
DO	GPIO17	*
AO	*	AIN0
VCC	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/18_thermistor/
```

Step 3: Compile.

```
gcc thermistor.c -lwiringPi -lm
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 18_thermistor.py
```

Now touch the thermistor and you can see the value of current temperature printed on the screen change accordingly.

Temperature alarm setting:

If you use the **Analog Temperature Sensor** module, uncomment the line under **1**:

For C language:

```
55          // For a threshold, uncomment one of the code for
56          // which module you use. DONOT UNCOMMENT BOTH!
57          //----------------------------------------
58          // 1. For Analog Temperature module(with DO)
59          tmp = digitalRead(DO);
60
61          // 2. For Thermister module(with sig pin)
62          // if (temp > 33) tmp = 0;
63          // else if (temp < 31) tmp = 1;
```

For Python

```
41          ##################################################
42          # 1. For Analog Temperature module(with DO)
43          tmp = GPIO.input(DO);
44
45          # 2. For Thermister module(with sig pin)
46          #if temp > 33:
47          #    tmp = 0;
48          #elif temp < 31:
49          #    tmp = 1;
50          ##################################################
```

If you use the **Thermistor module**, uncomment the lines under **2**:

For C language:

```
55          // For a threshold, uncomment one of the code for
56          // which module you use. DONOT UNCOMMENT BOTH!
57          //-----------------------------------------
58          // 1. For Analog Temperature module(with DO)
59          // tmp = digitalRead(DO);
60
61          // 2. For Thermister module(with sig pin)
62          if (temp > 33) tmp = 0;
63          else if (temp < 31) tmp = 1;
64          //-----------------------------------------
```

For Python

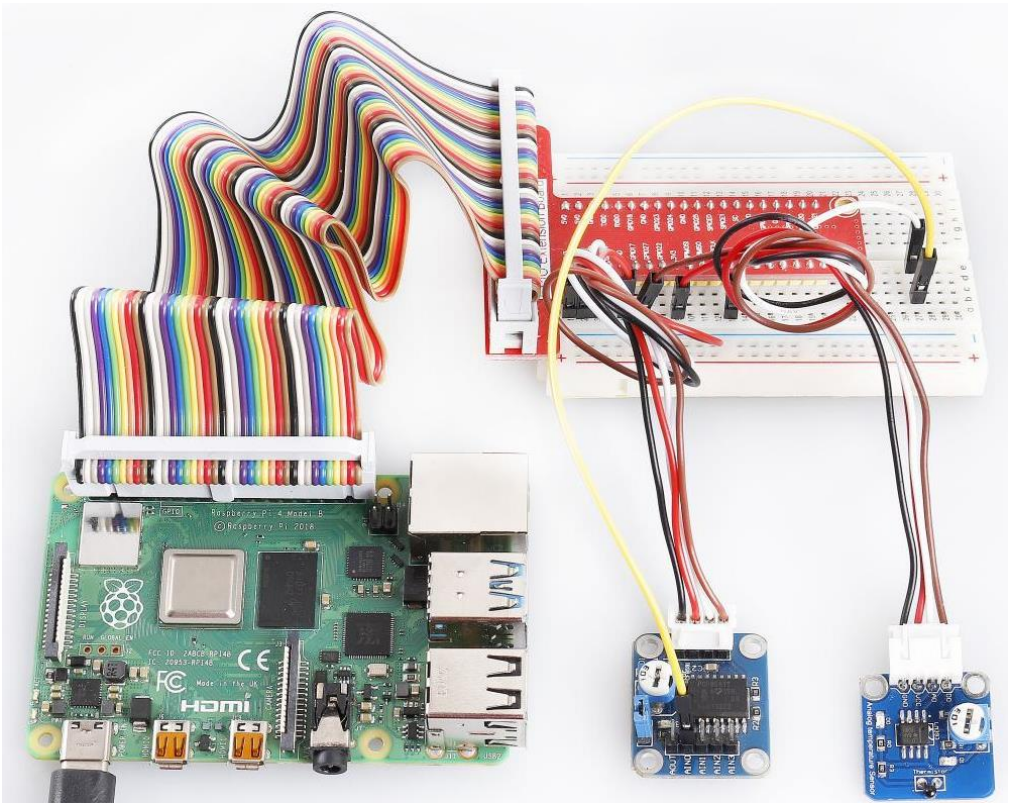
```
41          ##################################################
42          # 1. For Analog Temperature module(with DO)
43          #tmp = GPIO.input(DO);
44          #
45          # 2. For Thermister module(with sig pin)
46          if temp > 33:
47              tmp = 0;
48          elif temp < 31:
49              tmp = 1;
50          ##################################################
```

After editing the code, repeat step 2, 3, and 4 (or step 2, 3 for Python users) to run your modified program.

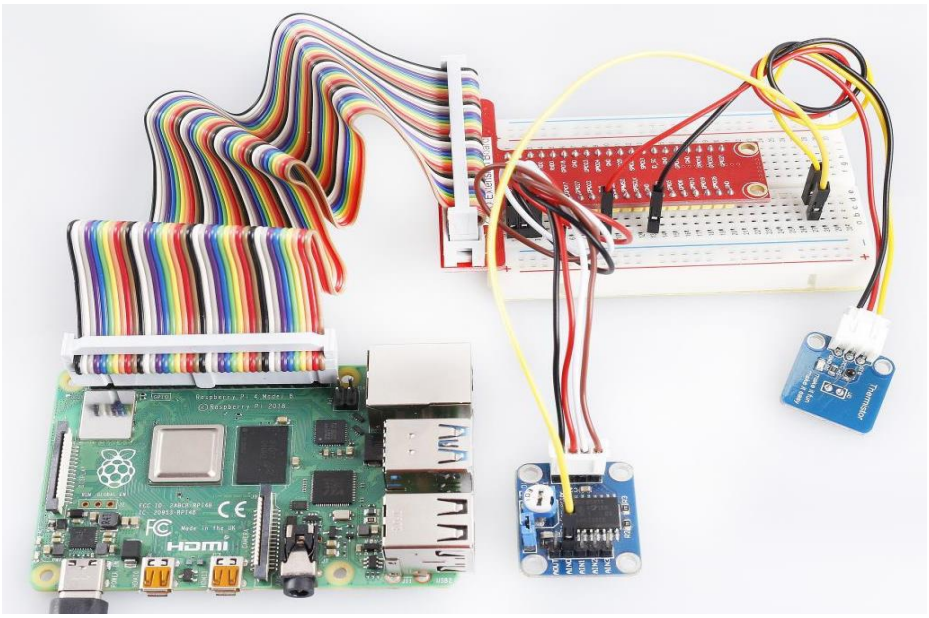
You can still see temperature value printed on the screen constantly. If you pinch the thermistor for a while, its temperature will rise slowly. "**Too Hot!**" will be printed on the screen. Release your fingers, and let it stay in the open air for a while, or blow on the module. When the temperature drops down slowly, "**Better**" will be printed.

Note:	The analog temperature sensor adjusts alarm temperature by the potentiometer on the module. The thermistor uses the alarm temperature indicated by the program.

A photo of the analog temperature sensor:



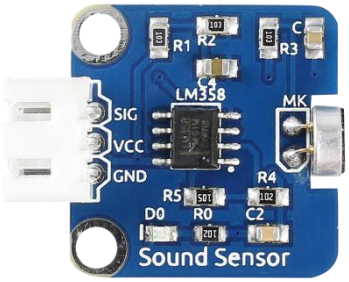
A photo of the thermistor module:



Lesson 19 Sound Sensor

Introduction

The Sound Sensor module picks up sound waves in an ambient environment and converts them into an electrical signal.



Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * PCF8591
- 1 * Sound sensor module
- 1 * 3-Pin Non-Reversible Cable
- Jumper Wires

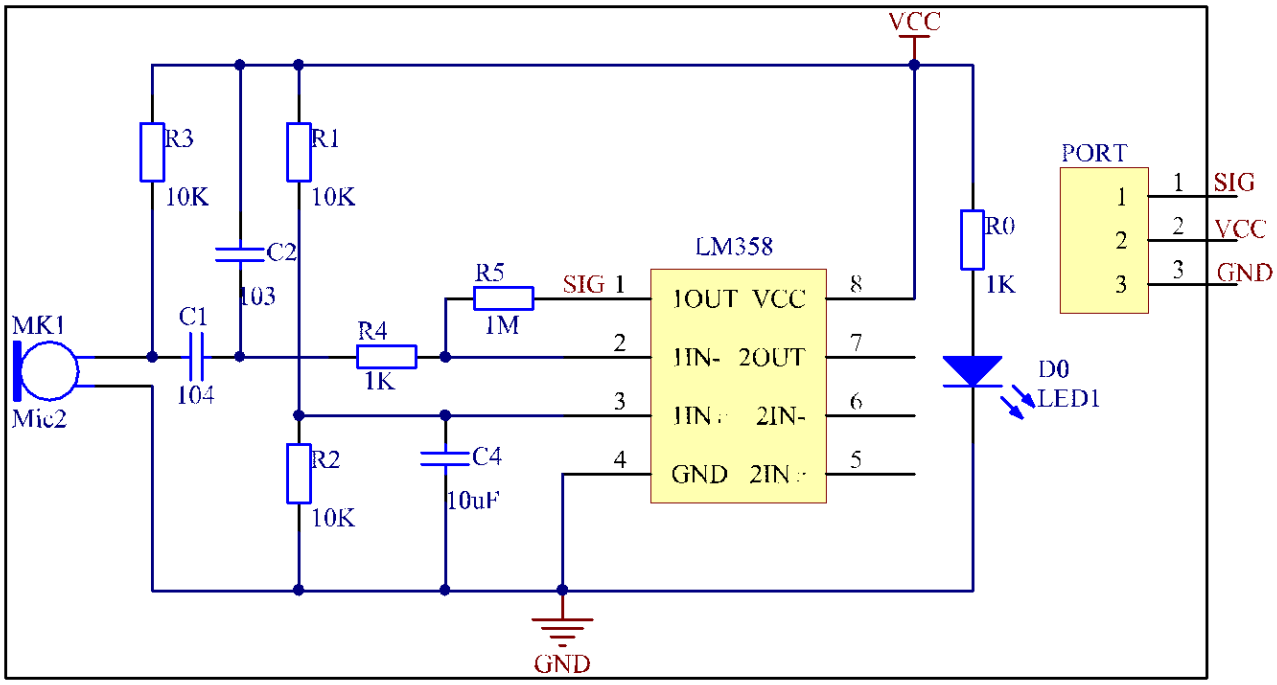
Purpose

The microphone on the sensor module can convert audio signals into electrical signals (analog quality), which are then converted into digital quality using the PCF8591.

LM358 is a dual-channel operational amplifier. It contains two independent, high gain, and internally compensated amplifiers, but we will only use one of them in this experiment. The microphone transforms sound signals into electrical signals and then sends out the signals to pin 2 of LM358 and outputs them to pin 1 (that's pin SIG of the module) via the external circuit. Then, PCF8591 is used to read the analog values.

PCF8591 is an 8-bit resolution, 4-channel A/D， 1-channel D/A conversion chip. We connect the output terminal (SIG) to AIN0 of PCF8591 to detect the strength of the audio signal in real-time.

The schematic diagram of the module is as shown below:

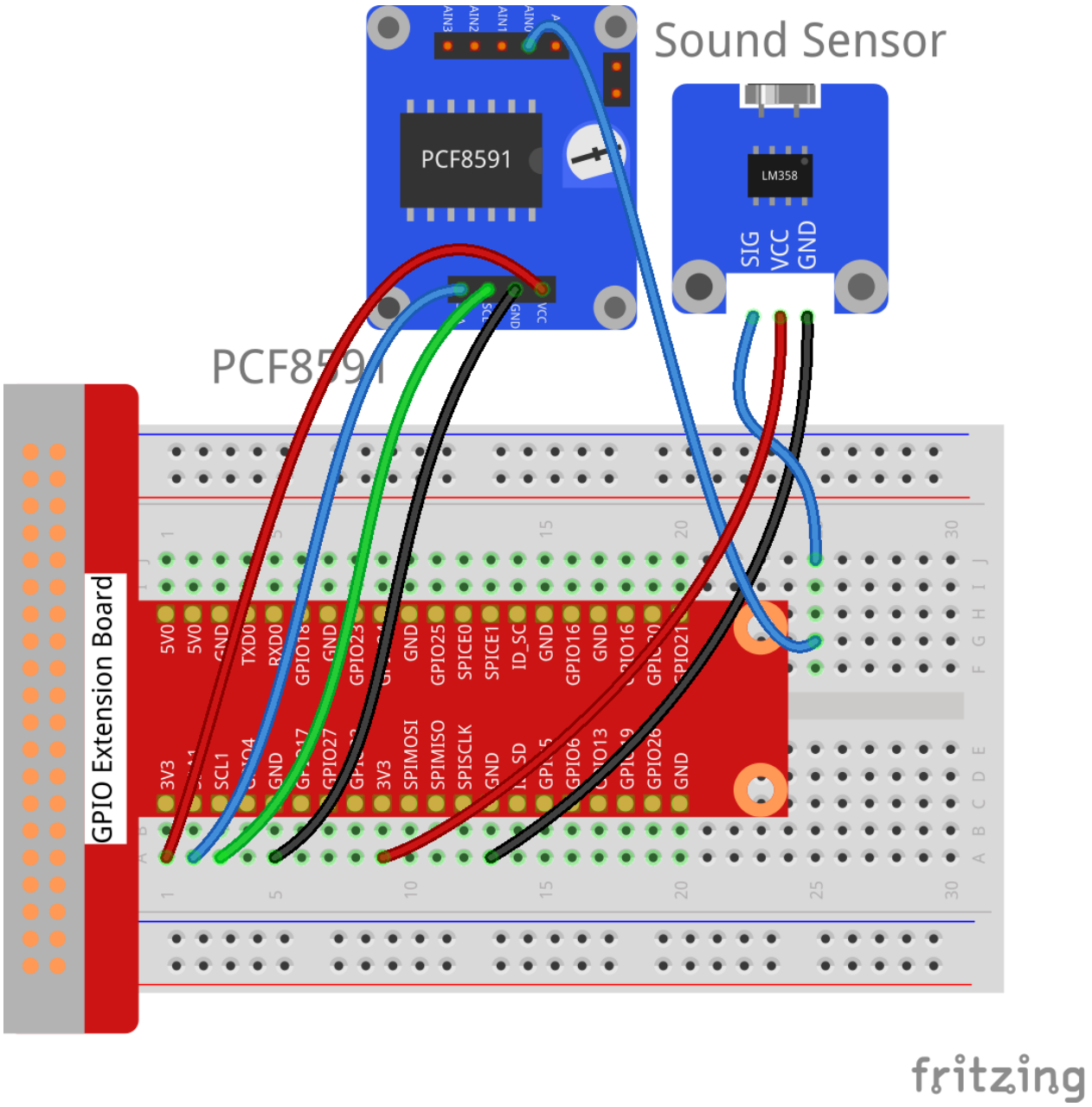


Procedure

Step 1: Build the circuit according to the following method.

Raspberry Pi	GPIO Extension Board	PCF8591 Module
SDA	SDA1	SDA
SCL	SCL1	SCL
3.3V	3V3	VCC
GND	GND	GND

Sound Sensor Module	GPIO Extension Board	PCF8591 Module
SIG	*	AIN0
VCC	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/19_sound_sensor/
```

Step 3: Compile.

```
gcc sound_sensor.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

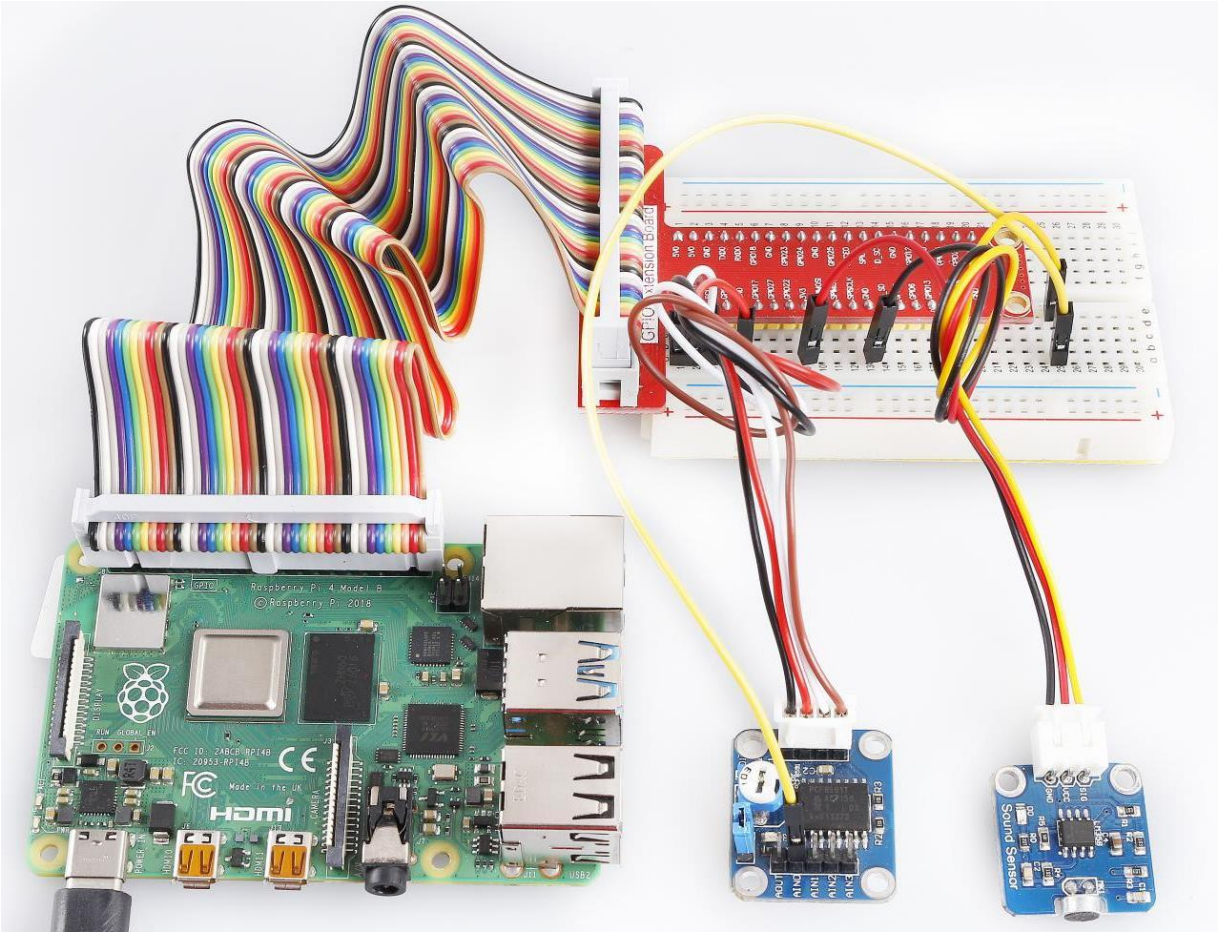
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 19_sound_sensor.py
```

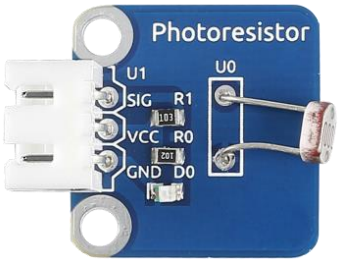
Now, speak close to or blow to the microphone, and you can see "Voice In!! ***" printed on the screen.



Lesson 20 Photoresistor Module

Introduction

The photoresistor module is a light-controlled variable resistor. The resistance lowers as the light increases in intensity.



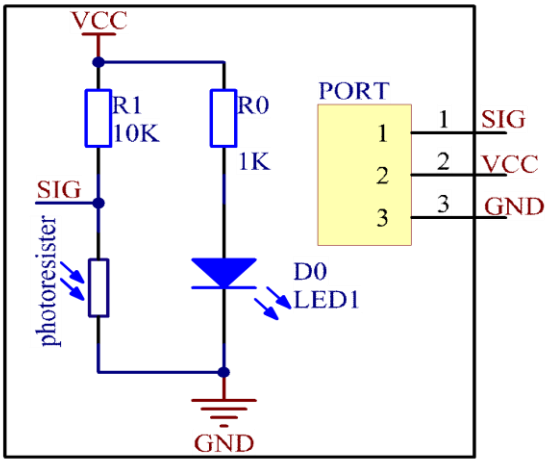
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * PCF8591
- 1 * Photoresistor module
- 1 * 3-Pin Non-Reversible Cable
- Jumper Wires

Purpose

As the light intensity increases, the resistance of the photoresistor will decrease, changing the output voltage. Analog signals collected by the photoresistor are then converted to digital signals using the PCF8591 converter.

The schematic diagram is as follows:

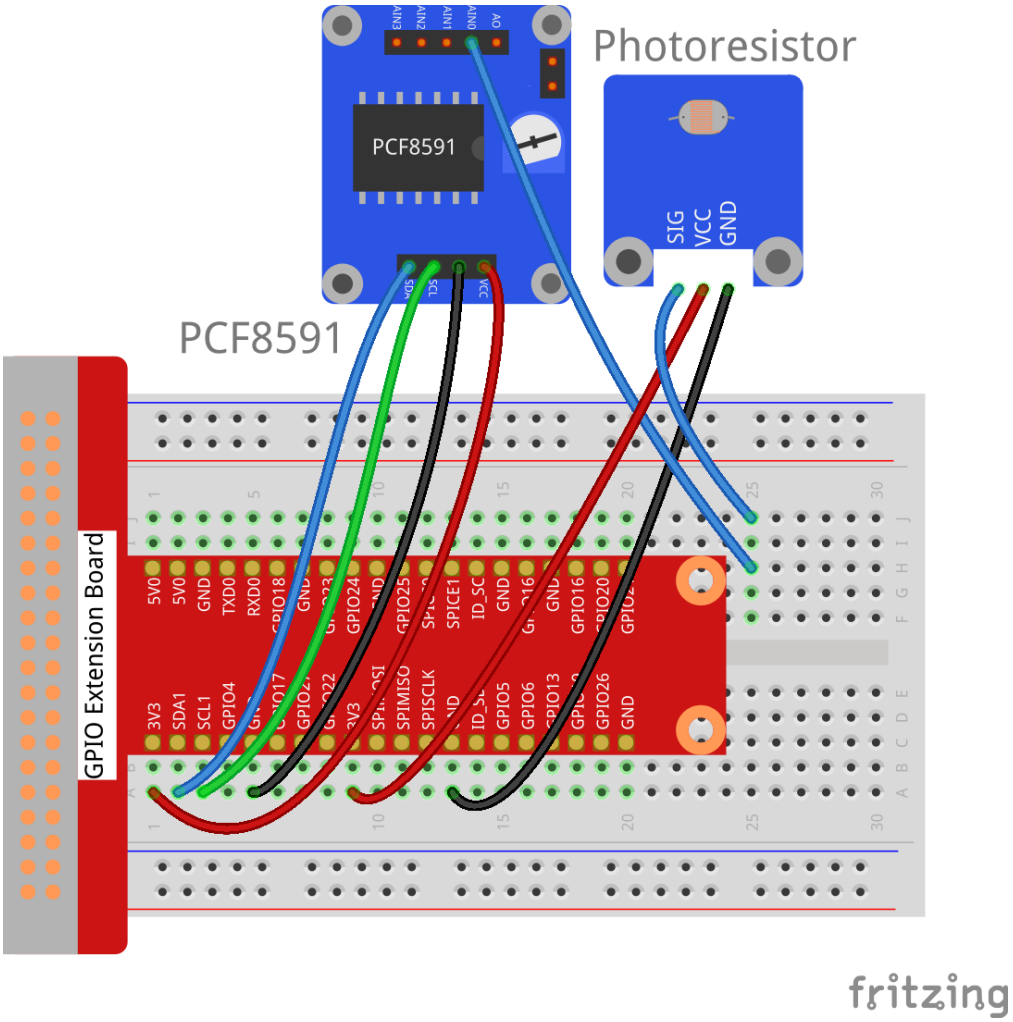


Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	PCF8591 Module
SDA	SDA1	SDA
SCL	SCL1	SCL
3.3V	3V3	VCC
GND	GND	GND

Photoresistor	GPIO Extension Board	PCF8591 Module
SIG	*	AIN0
VCC	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/20_photoresistor/
```

Step 3: Compile.

```
gcc photoresistor.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

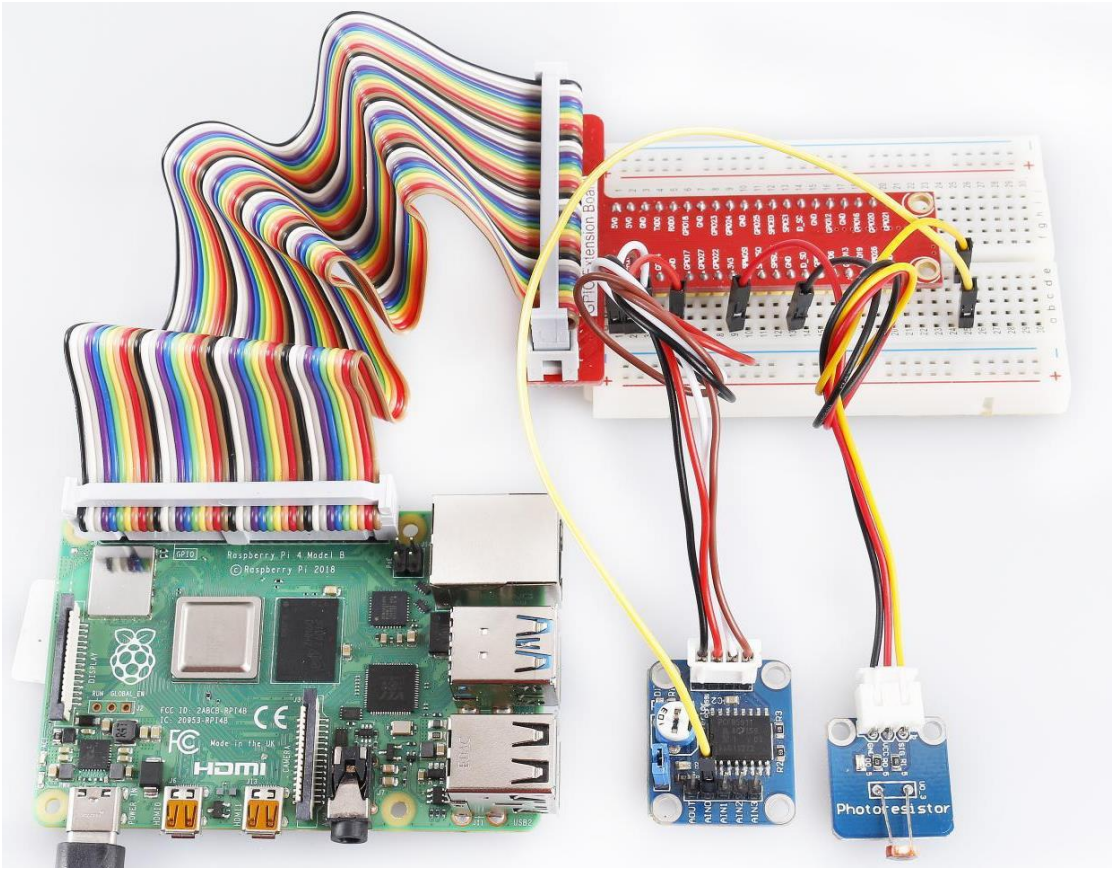
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

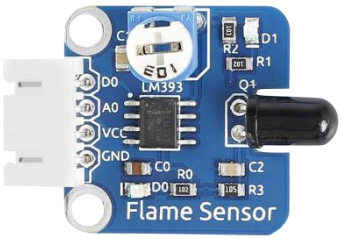
```
sudo python3 20_photoresistor.py
```

Experiment with different light intensities (by covering the module for example). The value printed to the screen will change accordingly.



Lesson 21 Flame Sensor

Introduction



The flame sensor module works to detect flames by capturing infrared rays with specific wavelengths.

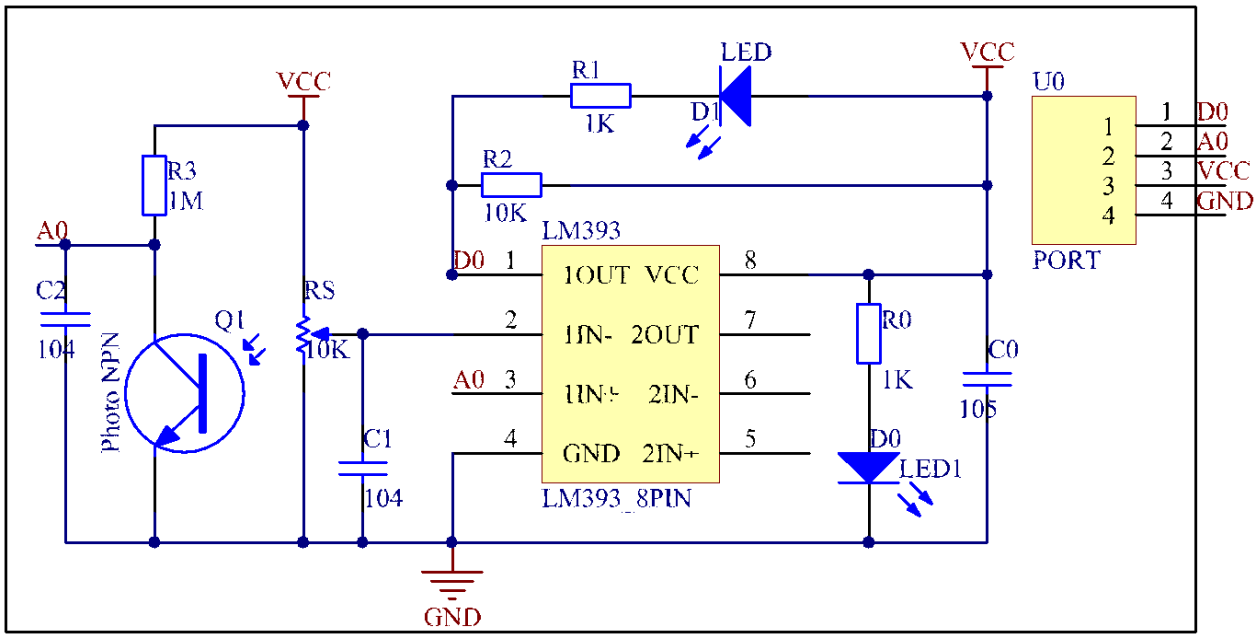
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Flame sensor module
- 1 * PCF8591
- 1 * 4-Pin Non-Reversible Cable
- Jumper Wires
- A Lighter (Not Included)

Purpose

There are several types of flame sensors and in this experiment a far-infrared flame sensor is used. It can detect infrared rays with wavelengths ranging from 700nm to 1000nm. The module converts the detected infrared light into current changes, expressed as an analog signal. The PCF8591 is then used to convert to a digital signal. In this experiment, the pin D0 of the flame sensor module is connected to the Raspberry Pi to detect whether a flame exists.

The schematic diagram is as follows:

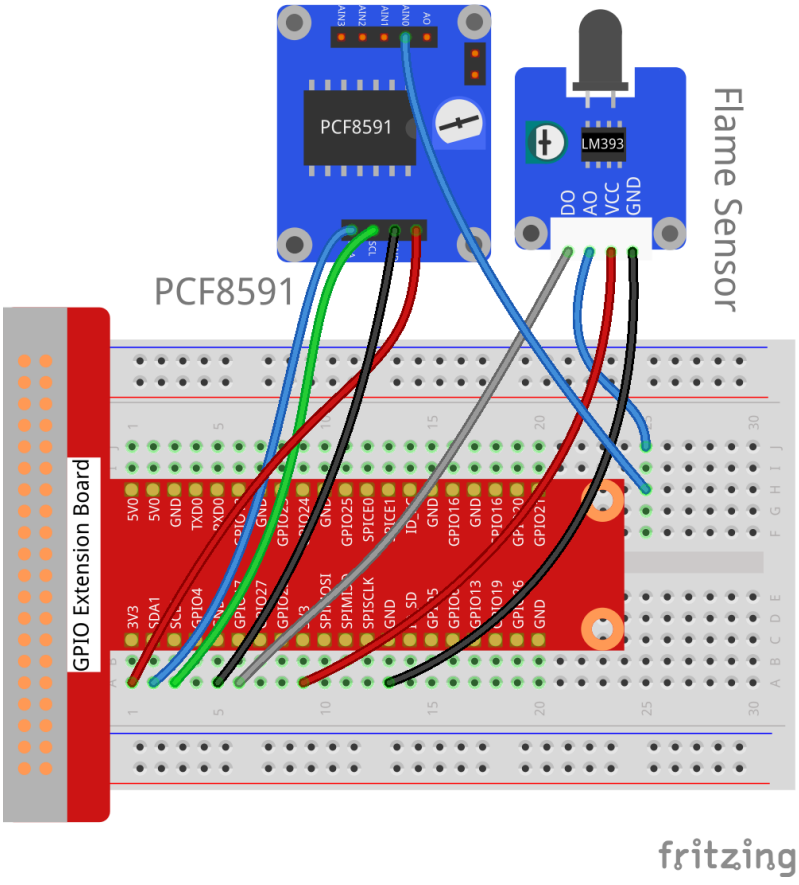


Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	PCF8591 Module
SDA	SDA1	SDA
SCL	SCL1	SCL
3.3V	3V3	VCC
GND	GND	GND

Flame Sensor	GPIO Extension Board	PCF8591 Module
DO	GPIO17	*
AO	*	AIN0
VCC	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/21_flame_sensor/
```

Step 3: Compile.

```
gcc flame_sensor.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

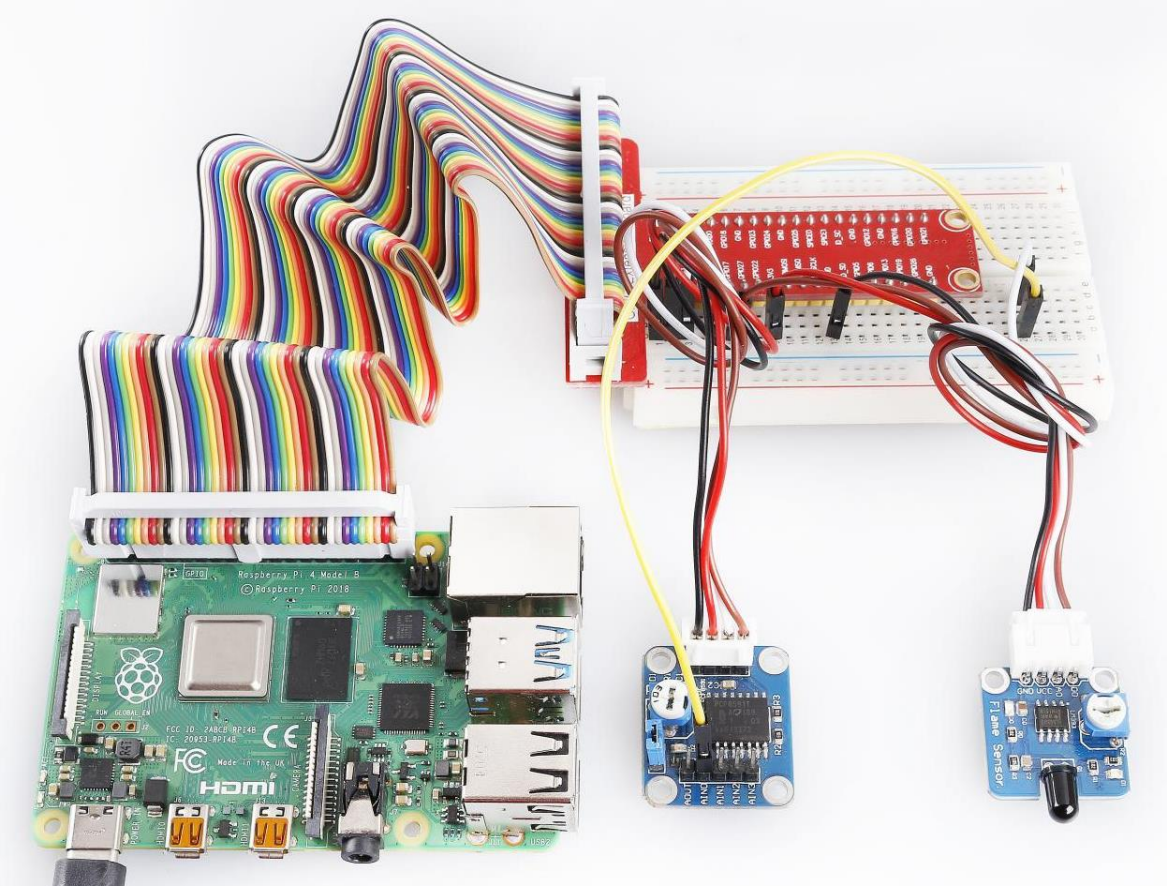
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 21_flame_sensor.py
```

Use a lighter or match near the sensor, within a range of 80cm, and "Fire!" will be printed to the screen. When the flame source is removed, the word "Safe~" will display instead.



Lesson 22 Gas Sensor

Introduction

The Gas Sensor MQ-2 is used to detect flammable gas and smoke concentrations in the air. They are used in household, industrial, and automobile gas detecting equipment.

Required Components

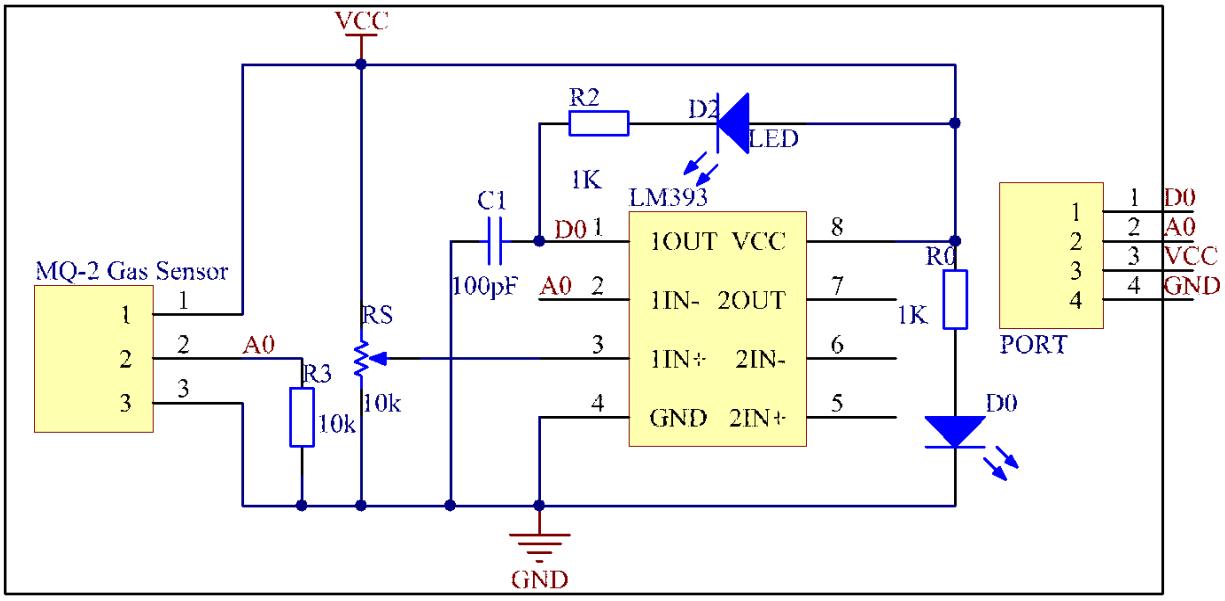
- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Active Buzzer module
- 1 * PCF8591
- 1 * Gas sensor module
- 1 * 3-Pin Non-Reversible Cable
- 1 * 4-Pin Non-Reversible Cable
- Jumper Wires

Purpose

The MQ-2 gas sensor uses ion type and N-type tin oxide semiconductors inside a gas sensitive material. Tin oxide will absorb oxygen in the air and form oxygen anion absorption to decrease electron density in the semiconductor and lower its resistance. When in contact with smoke, the surface conductivity changes to become more conductive the higher the concentration.

In this experiment, if harmful gases reach a certain concentration, the buzzer will beep in warning.

The schematic diagram of the module is as shown below:



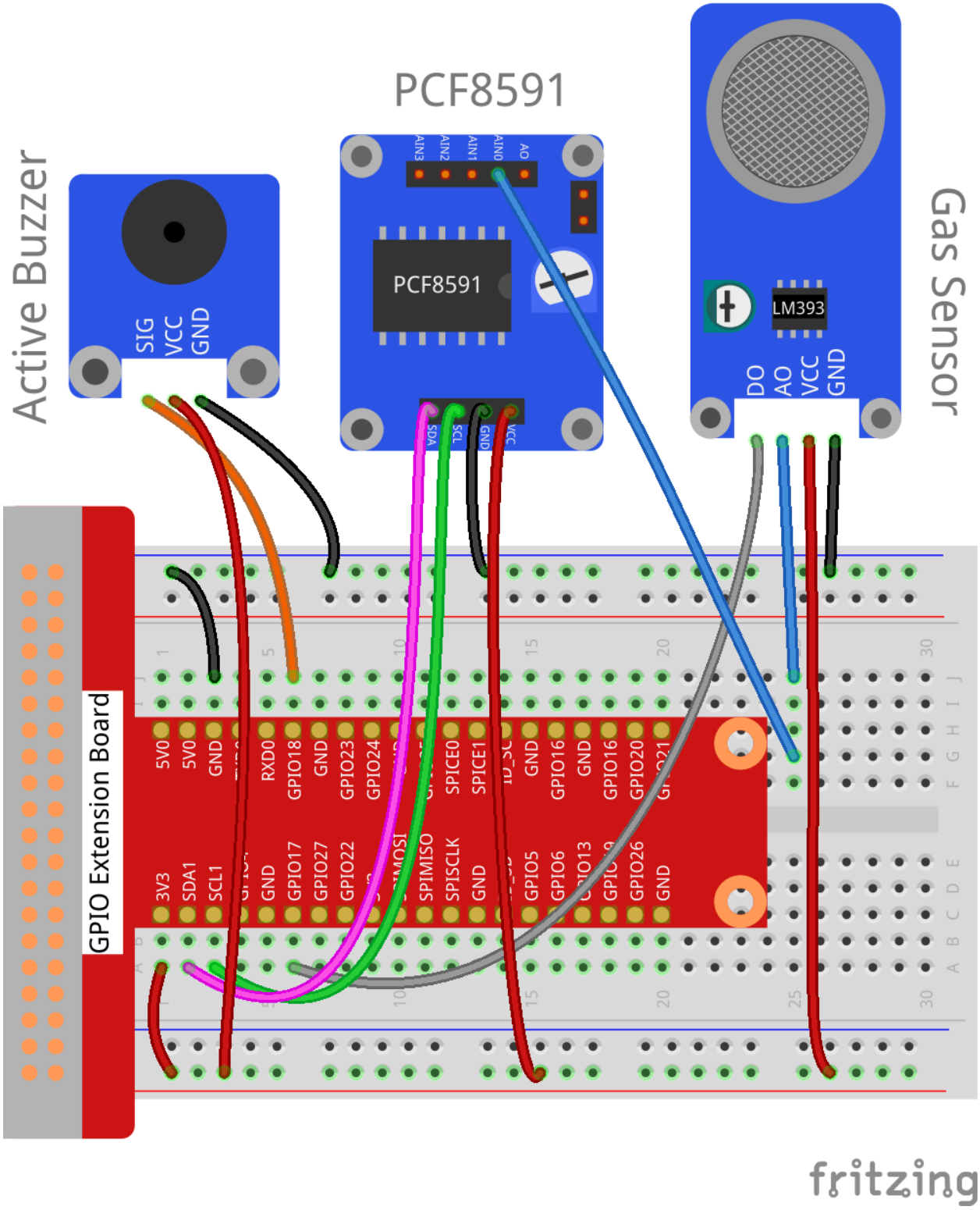
Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	PCF8591 Module
SDA	SDA1	SDA
SCL	SCL1	SCL
3.3V	3V3	VCC
GND	GND	GND

Gas Sensor Module	GPIO Extension Board	PCF8591 Module
DO	GPIO17	*
AO	*	AIN0
VCC	3V3	*
GND	GND	GND

Raspberry Pi	GPIO Extension Board	Active Buzzer Module
GPIO1	GPIO18	SIG
3.3V	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/22_gas_sensor/
```

Step 3: Compile.

```
gcc gas_sensor.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

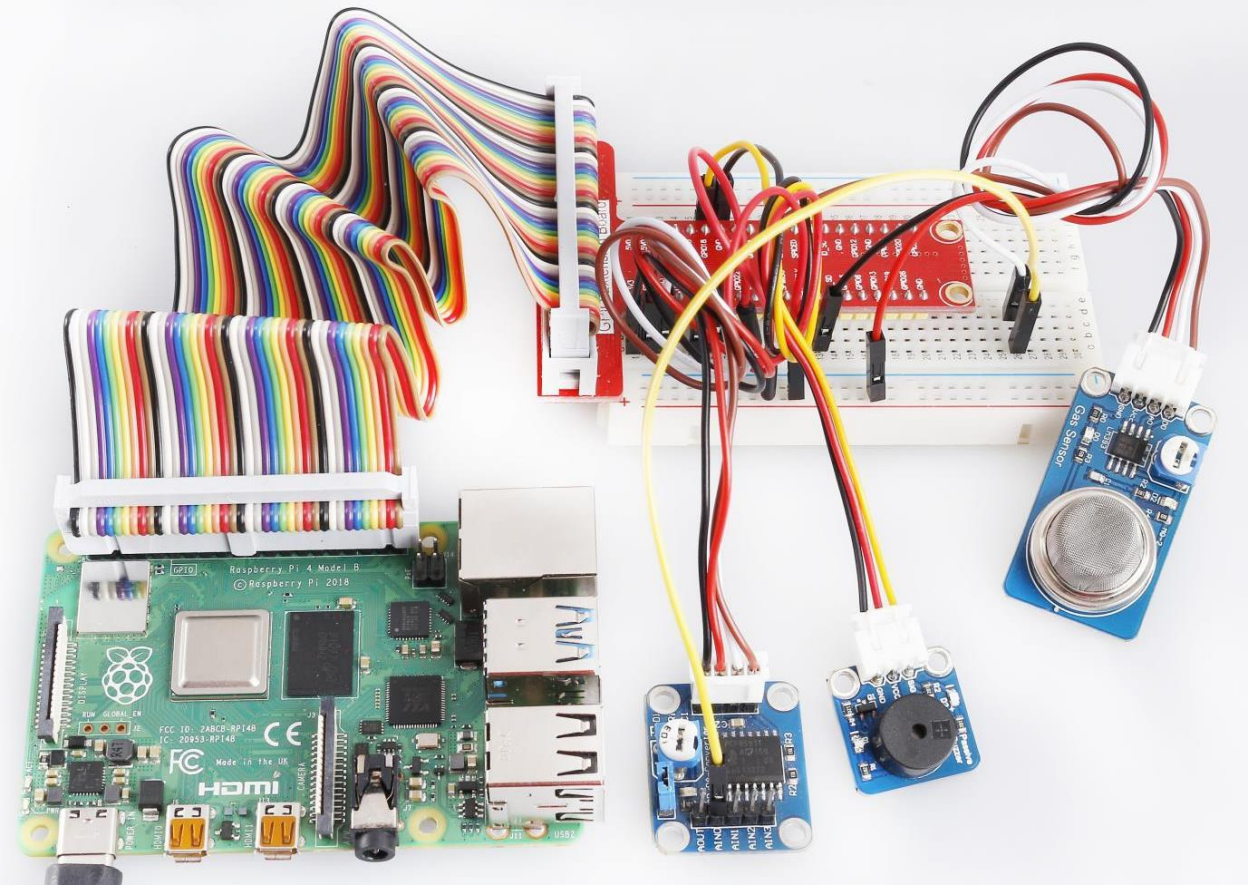
```
sudo python3 22_gas_sensor.py
```

Place a lighter close to the MQ-2 gas sensor, and press the switch to release gasses. A value between 0 and 255 will be displayed on the screen. If harmful gases reach a certain concentration, the buzzer will beep, and "Danger Gas!" will be printed on the screen.

You can also turn the shaft of the potentiometer on the module to raise or reduce the concentration threshold.

The MQ-2 gas sensor needs to heat up for a few moments. Wait until the value printed on the screen remains steady before beginning.

Note:	It is normal for the gas sensor to generate heat. The higher the temperature of the sensor, the greater its sensitivity.


Raspberry Pi 4 Model B
©Raspberry Pi 2018
Gas Sensor

Lesson 23 IR Remote Control

Introduction

Each button on the IR remote control has a string of specific encoding. When a button is pressed, the IR transmitter on the remote will send out the corresponding IR encoding signals. The IR receiver will decode the signals and use them to identify which button was pressed.

Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * IR Receiver
- 1 * RGB LED module
- 1 * IR Remote Control
- 1 * 3-Pin Non-Reversible Cable
- 1 * 4-Pin Non-Reversible Cable

Purpose

In this experiment, the lirc library is used to read the infrared signals returned by the buttons on the remote control and translate them into button values. The libraries liblircclient-dev (C) and pylirc (Python) are used to simplify the process.

In this experiement, use the chart below to see how the nine remote buttons will map to colors on the RGB LED module. Each row represents one color, and each column a brightness.

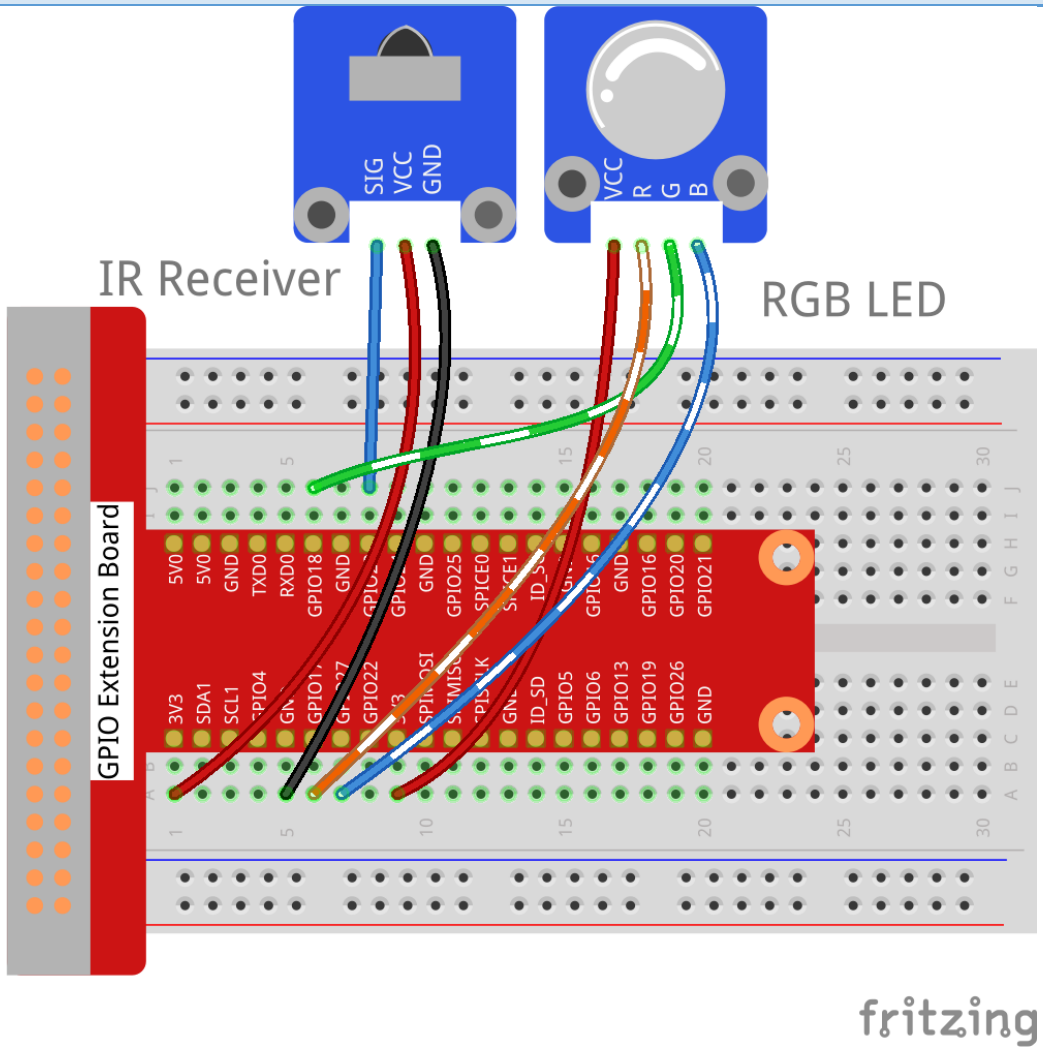
	OFF	Dark	Bright
Red			
Green			
Blue			

Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	IR Receiver Module
GPIO4	GPIO23	SIG
3.3V	3V3	VCC
GND	GND	GND

Raspberry Pi	GPIO Extension Board	RGB LED Module
3.3V	3V3	VCC
GPIO0	GPIO17	R
GPIO1	GPIO18	G
GPIO2	GPIO27	B



Step 2: Upgrade.

```
   sudo su -c "grep '^deb ' /etc/apt/sources.list | sed 's/^deb/deb-
src/g' > /etc/apt/sources.list.d/deb-src.list"
   sudo apt update
   sudo apt install devscripts
```

Step 3: Installing the library and the patch for gpio-ir:

For Raspbian **Stretch**:

```
   sudo apt install dh-exec doxygen expect libasound2-dev libftdi1-dev
libsystemd-dev libudev-dev libusb-1.0-0-dev libusb-dev man2html-base
portaudio19-dev socat xsltproc python3-yaml dh-python libx11-dev python3-
dev python3-setuptools
   mkdir build
   cd build
   apt source lirc
   wget https://raw.githubusercontent.com/neuralassembly/raspi/master/lirc-
gpio-ir-0.10.patch
   patch -p0 -i lirc-gpio-ir-0.10.patch
   cd lirc-0.10.1
   debuild -uc -us -b
   cd ..
   sudo apt install ./liblirc0_0.10.1-
5.2_armhf.deb ./liblircclient0_0.10.1-5.2_armhf.deb ./lirc_0.10.1-
5.2_armhf.deb
```

For Raspberry Pi OS **Buster**:

```
   sudo apt build-dep lirc
   mkdir build
   cd build
   apt source lirc
   wget https://raw.githubusercontent.com/neuralassembly/raspi/master/lirc-
gpio-ir.patch
   patch -p0 -i lirc-gpio-ir.patch
   cd lirc-0.9.4c
   debuild -uc -us -b
   cd ..
   sudo apt install ./liblirc0_0.9.4c-9_armhf.deb ./liblirc-client0_0.9.4c-
9_armhf.deb ./lirc_0.9.4c-9_armhf.deb
```

If you encounter problems during the installation process, please try a few more times. The final install command will fail. Please configure the files shown below first, i.e., /boot/config.txt and /etc/lirc/lirc_options.conf. Afterwards, please try the final install command again.

Step 4: Set up lirc.
Open your */boot/config.txt* file:

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

Add this to the file:

```
# Uncomment this to enable the lirc-rpi module
#dtoverlay=lirc-rpi
dtoverlay=gpio-ir,gpio_pin=23
dtoverlay=gpio-ir-tx,gpio_pin=22
```

Press Ctrl +O and Ctrl +X, save and exit .

Step 5: When using Raspberry Pi OS Buster, the following command is needed:

```
sudo mv /etc/lirc/lirc_options.conf.dist /etc/lirc/lirc_options.conf
sudo mv /etc/lirc/lircd.conf.dist /etc/lirc/lircd.conf
```

Step 6: edit /etc/lirc/lirc_options.conf.

Open the /etc/lirc/lirc_options.conf

```
sudo nano /etc/lirc/lirc_options.conf
```

Modify the file as below:

```
driver = default
device = /dev/lirc1
```

Step 7: Run install command again.

```
sudo apt install ./liblirc0_0.10.1-5.2_armhf.deb ./liblircclient0_0.10.1-5.2_armhf.deb ./lirc_0.10.1-5.2_armhf.deb
```

Step 8: Copy the configuration file to/home/pi and /etc/lirc:

```
cd /home/pi/HiPi-Sensor-kit-v4.0
cp lircd.conf /home/pi
sudo cp lircd.conf /etc/lirc/
```

Step 9: Reboot the Raspberry Pi after the change.

```
sudo reboot
```

Step 10: Test the IR receiver.
Check if lirc module is loaded:

```
ls /dev/li*
```

You should see this:

```
/dev/lirc0      /dev/lirc1
```

Step 11: Run the command to start outputting raw data from the IR receiver:

```
irw
```

When you press a button on the remote, you can see the button name printed on the screen.

```
pi@raspberrypi:~ $ irw
0000000000000001 00 KEY_CHANNELDOWN ./lircd.conf
0000000000000003 00 KEY_CHANNELUP ./lircd.conf
0000000000000002 00 KEY_CHANNEL ./lircd.conf
0000000000000004 00 KEY_PREVIOUS ./lircd.conf
0000000000000005 00 KEY_NEXT ./lircd.conf
0000000000000006 00 KEY_PLAYPAUSE ./lircd.conf
0000000000000008 00 KEY_VOLUMEDOWN ./lircd.conf
0000000000000007 00 KEY_VOLUMEUP ./lircd.conf
0000000000000009 00 KEY_EQUAL ./lircd.conf
0000000000000015 00 BTN_1 ./lircd.conf
0000000000000014 00 BTN_0 ./lircd.conf
000000000000000a 00 KEY_NUMERIC_0 ./lircd.conf
000000000000000b 00 KEY_NUMERIC_1 ./lircd.conf
```

If it does not appear, somewhere may be incorrectly configured. Check again that you've connected everything and haven't crossed any wires.

For C Users:

Step 5: Download LIRC client library:

```
sudo apt install liblircclient-dev
```

Step 6: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/23_ircontrol/
```

Step 7: Copy the *lircrc* file to */etc/lirc/lirc/*:

```
sudo cp lircrc /etc/lirc/
```

Step 8: Compile.

```
gcc ircontrol.c -lwiringPi -llirc_client
```

Step 9: Run.

```
sudo ./a.out
```

For Python Users:

Step 5: Download and install pylirc:

Pylirc is LIRC Python wrapper and it's required to access LIRC from Python programs. To install Pylirc you should complete the following steps.

Install Pylirc dependencies:

```
sudo apt install python3-dev
sudo apt install liblircclient-dev
```

Install Pylirc:

```
   wget
https://files.pythonhosted.org/packages/a9/e1/a19ed9cac5353ec07294be7b1aefc8f
89985987b356e916e2c39b5b03d9a/pylirc2-0.1.tar.gz
   tar xvf pylirc2-0.1.tar.gz
   cd pylirc2-0.1
```

Step 6: Replace file pylircmodule.c:

```
   rm pylircmodule.c
   wget https://raw.githubusercontent.com/project-
owner/Peppy.doc/master/files/pylircmodule.c
```

Step 7: Install Pylirc (assuming that Python 3.7 is in use):

```
   sudo python3 setup.py install
   sudo mv /usr/local/lib/python3.7/dist-packages/pylircmodule.cpython-37m-
arm-linux-gnueabihf.so /usr/local/lib/python3.7/dist-packages/pylirc.cpython-
37m-arm-linux-gnueabihf.so
```

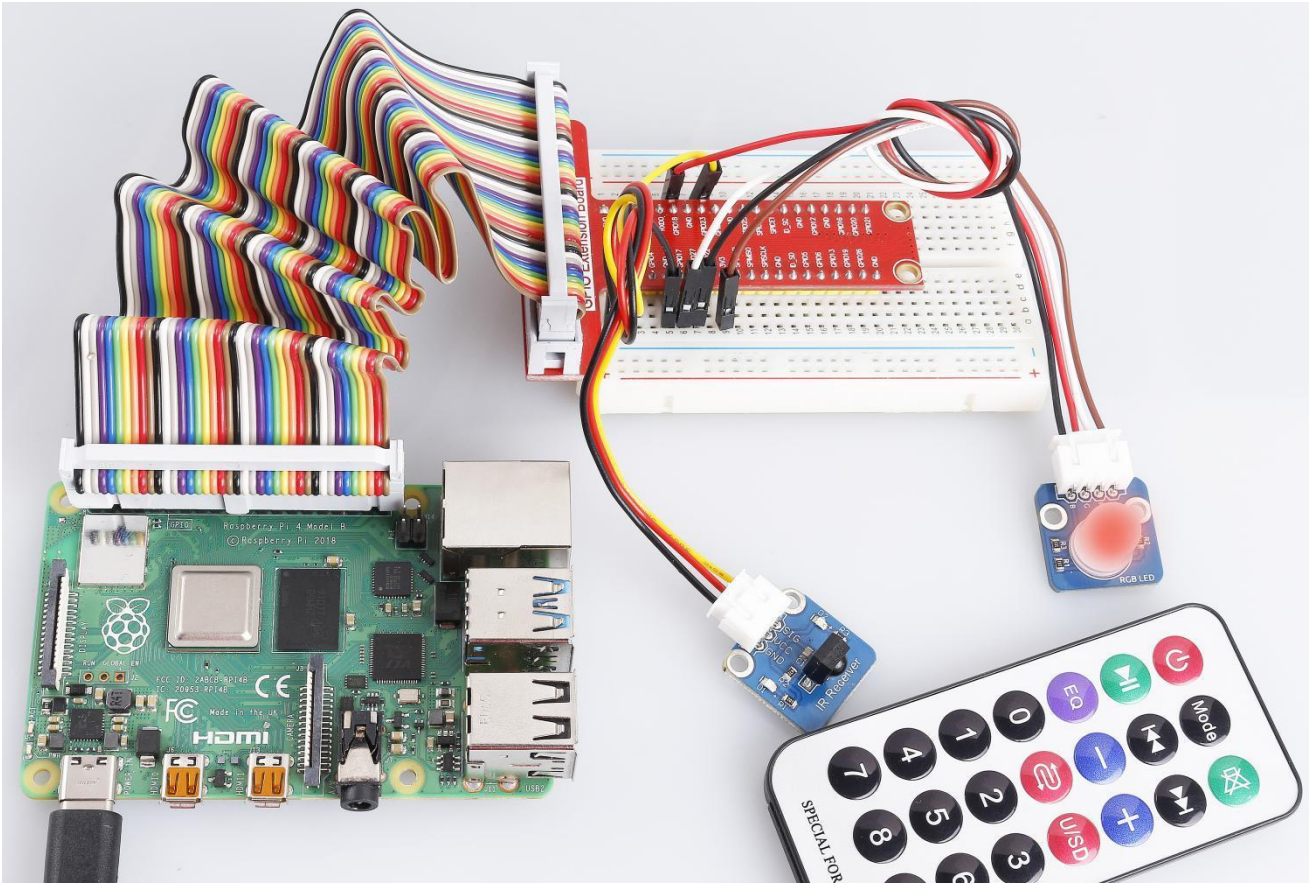
Step 8: Change directory:

```
   cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 9: Run.

```
   sudo python3 23_ircontrol.py
```

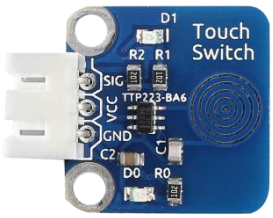
Each of the nine buttons represent a different combination of color and brightness. See the chart at the beginning of the instructions for reference.



Lesson 24 Touch Switch

Introduction

The touch sensor module operates with the conductivity of the human body. When touched, the metal on the base electrode of the transistor will activate, changing the level on pin SIG



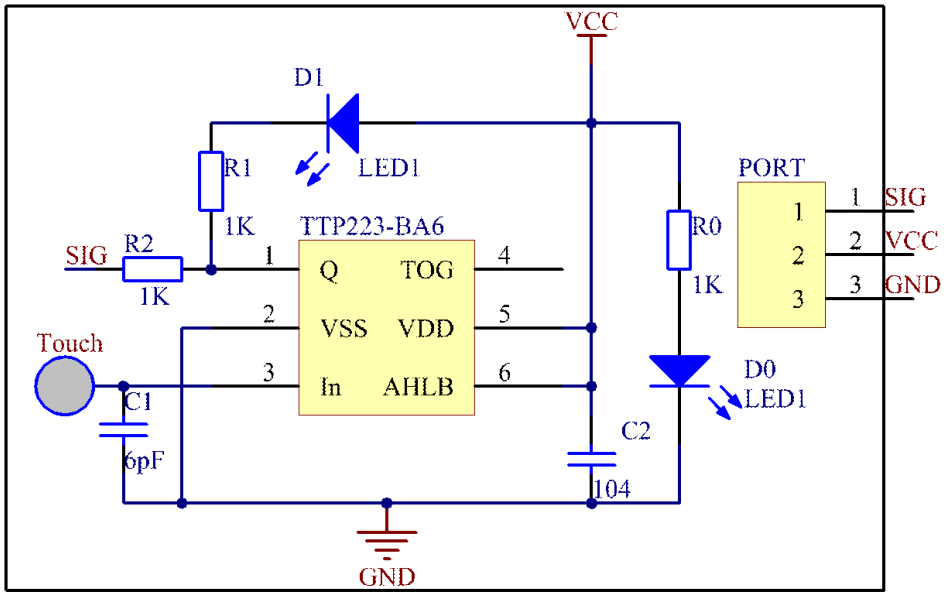
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Touch sensor module
- 1 * Dual-Color LED module
- 2 * 3-Pin Non-Reversible Cable

Purpose

When the module is touched, the conductivity of the human body will amplify the electromagnetic wave signals through the transistor. They are then processed by the module's comparator and outputted as a steady signal.

The schematic diagram of the module is as shown below:

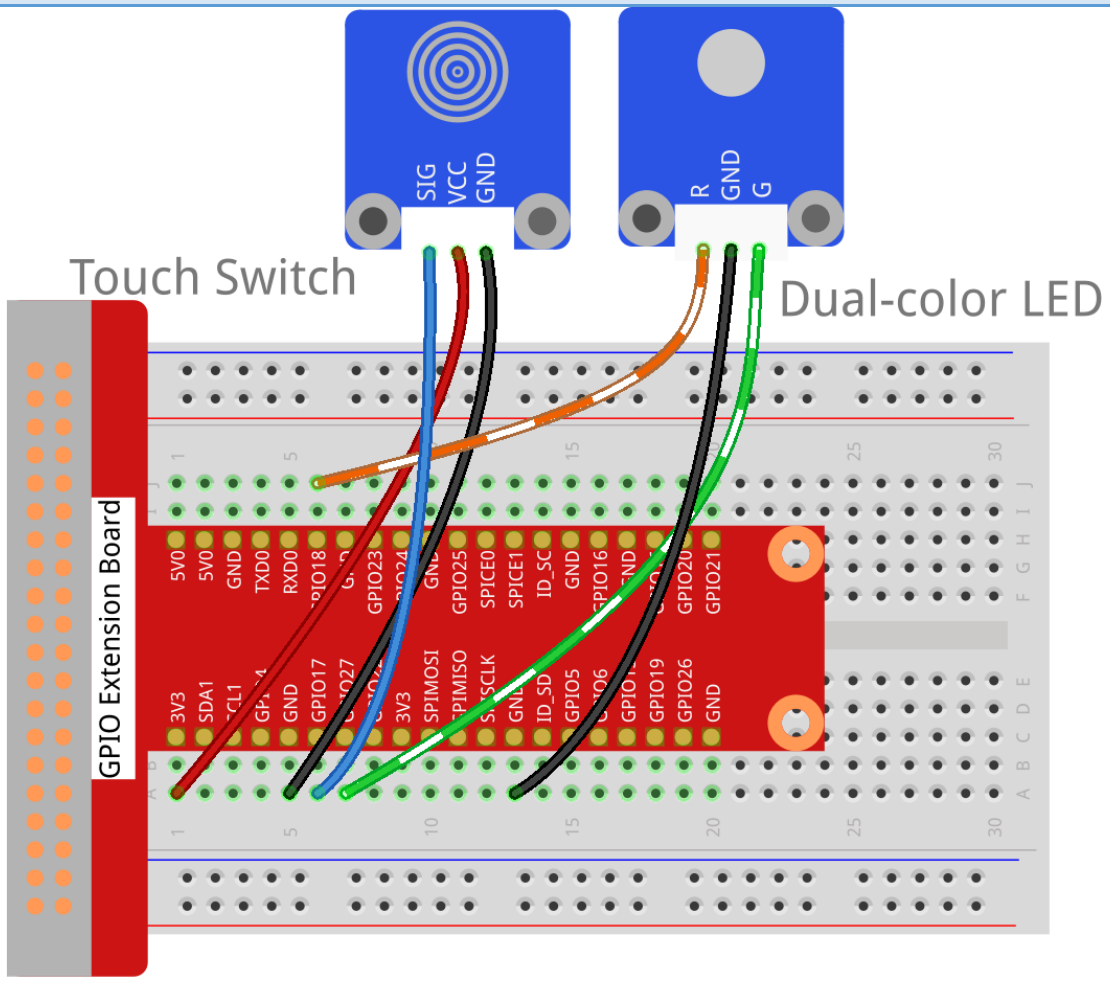


Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Touch Sensor Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND

Raspberry Pi	GPIO Extension Board	Dual-Color LED Module
GPIO1	GPIO18	R
GND	GND	GND
GPIO2	GPIO27	G



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/24_touch_switch/
```

Step 3: Compile.

```
gcc touch_switch.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

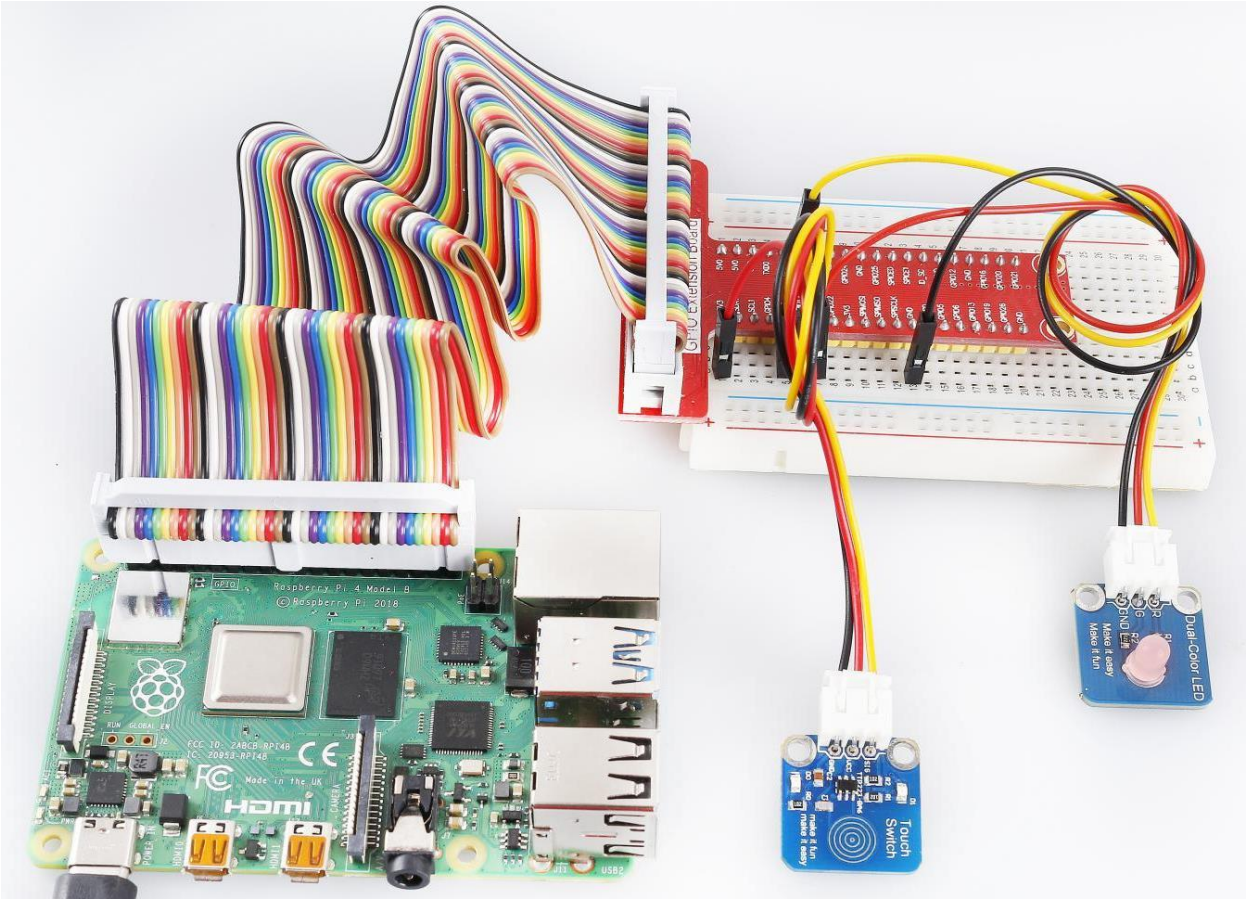
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 24_touch_switch.py
```

When the metal disk is touched, the LEDs will change color and "ON" and"OFF" will be printed to the screen.



Lesson 25 Ultrasonic Ranging Module

Introduction

The ultrasonic sensor uses sound to accurately detect objects and measure distances. It sends out ultrasonic waves and converts them into electronic signals.

Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Ultrasonic ranging module
- 1 * 4-Pin Non-Reversible Cable

Purpose

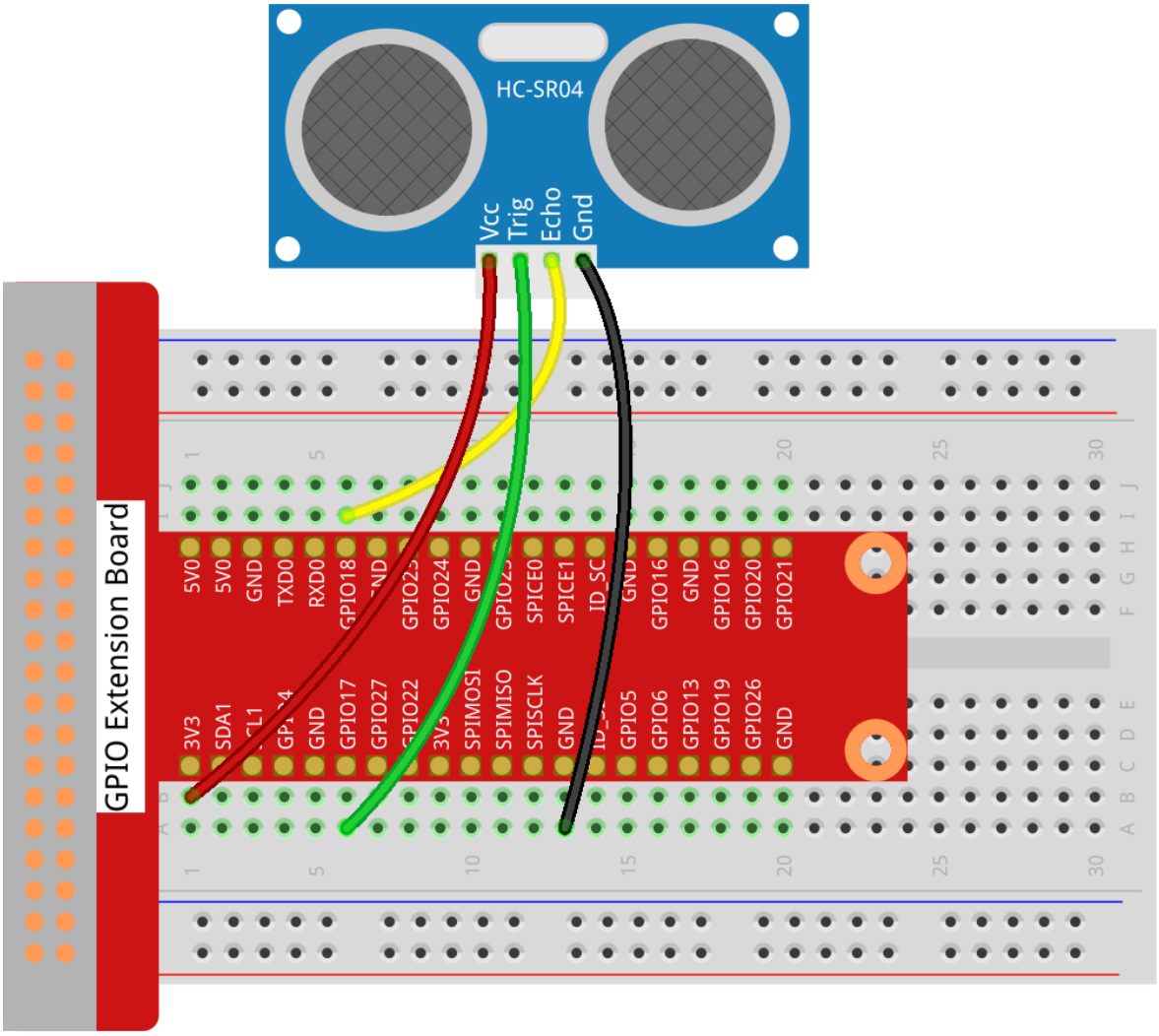
This sensor works by sending out a sound wave and calculating the time it takes for it to return to the ultrasonic sensor. By doing this, it can tell how far away objects are relative to the sensor.

Test distance = (high level time * velocity of sound (340M/S)) / 2 (in meters)

Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Ultrasonic Ranging Module
3.3V	3V3	VCC
GPIO0	GPIO17	Trig
GPIO1	GPIO18	Echo
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/25_ultrasonic_ranging/
```

Step 3: Compile.

```
gcc ultrasonic_ranging.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

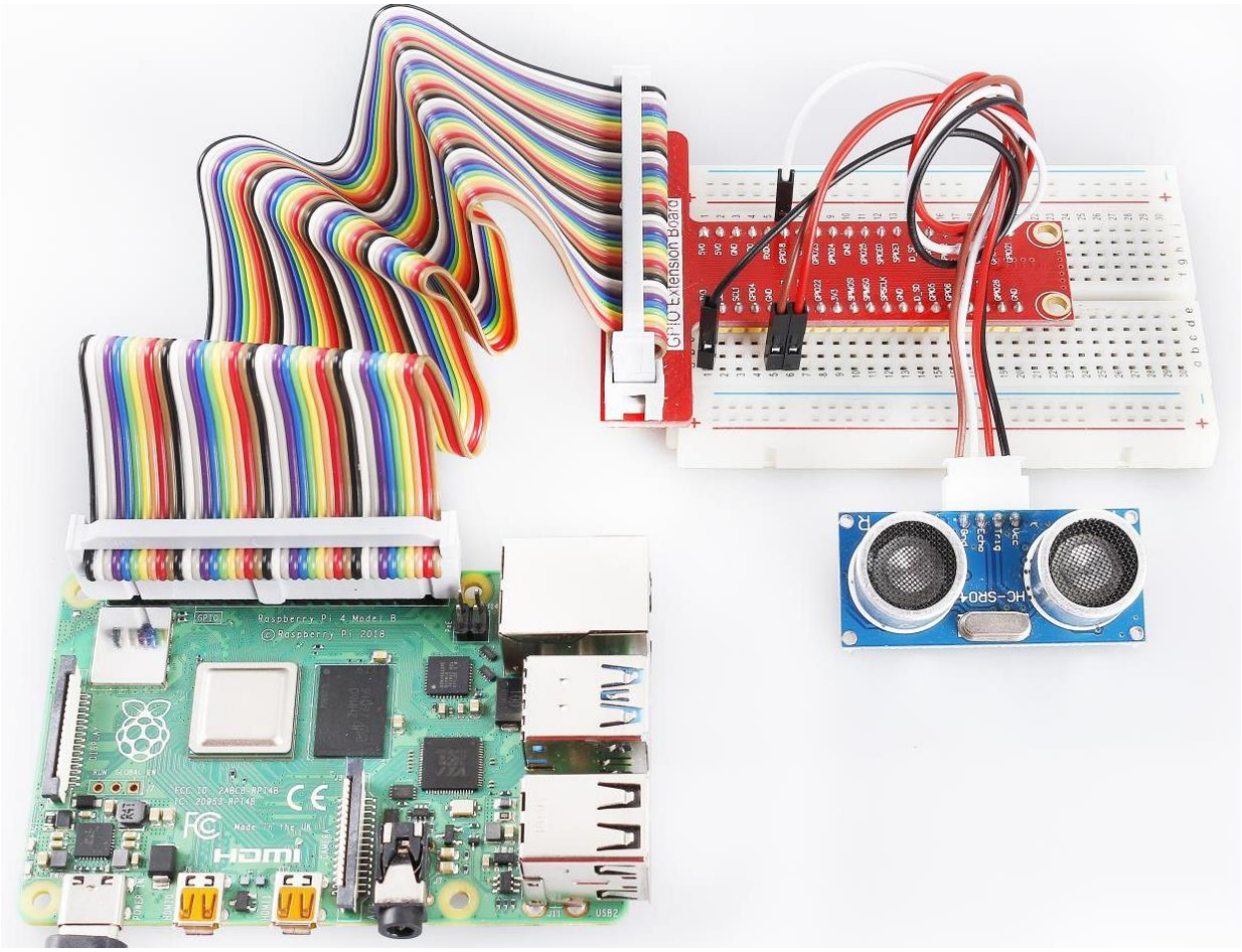
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 25_ultrasonic_ranging.py
```

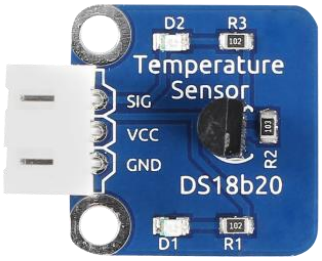
To find the distance between an obstacle (like your hand) and the sensor, slowly move it within the sensor's range and observe the distance printed on the screen.


GPIO Extension Board
Raspberry Pi 4 Model B
©Raspberry Pi 2018
HC-SR04
Trig
Echo
Vcc
Gnd
HDMI

Lesson 26 DS18B20 Temperature Sensor

Introduction

The DS18B20 is a small, low-cost, high precision digital temperature sensor with strong anti-interference capabilities. The sensor is easy to wire (using 1-wire bus) and can directly output temperature data.



Required Components

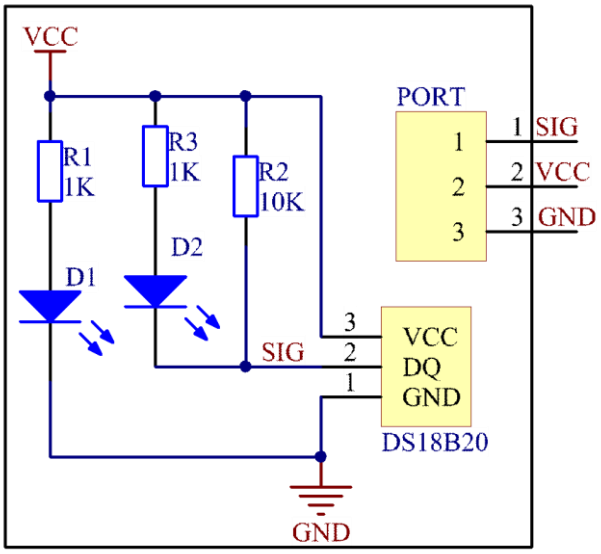
- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * DS18B20 Temperature Sensor module
- 1 * 3-Pin Non-Reversible Cable

Purpose

With its unique single-wire interface, DS18B20 requires only one pin for two-way communication with a microprocessor. It supports multi-point temperature networking; eight sensors can be connected at a time.

When using the DS18B20, a 10K Ohm resistor must be used with the middle pin DQ to pull up the level.

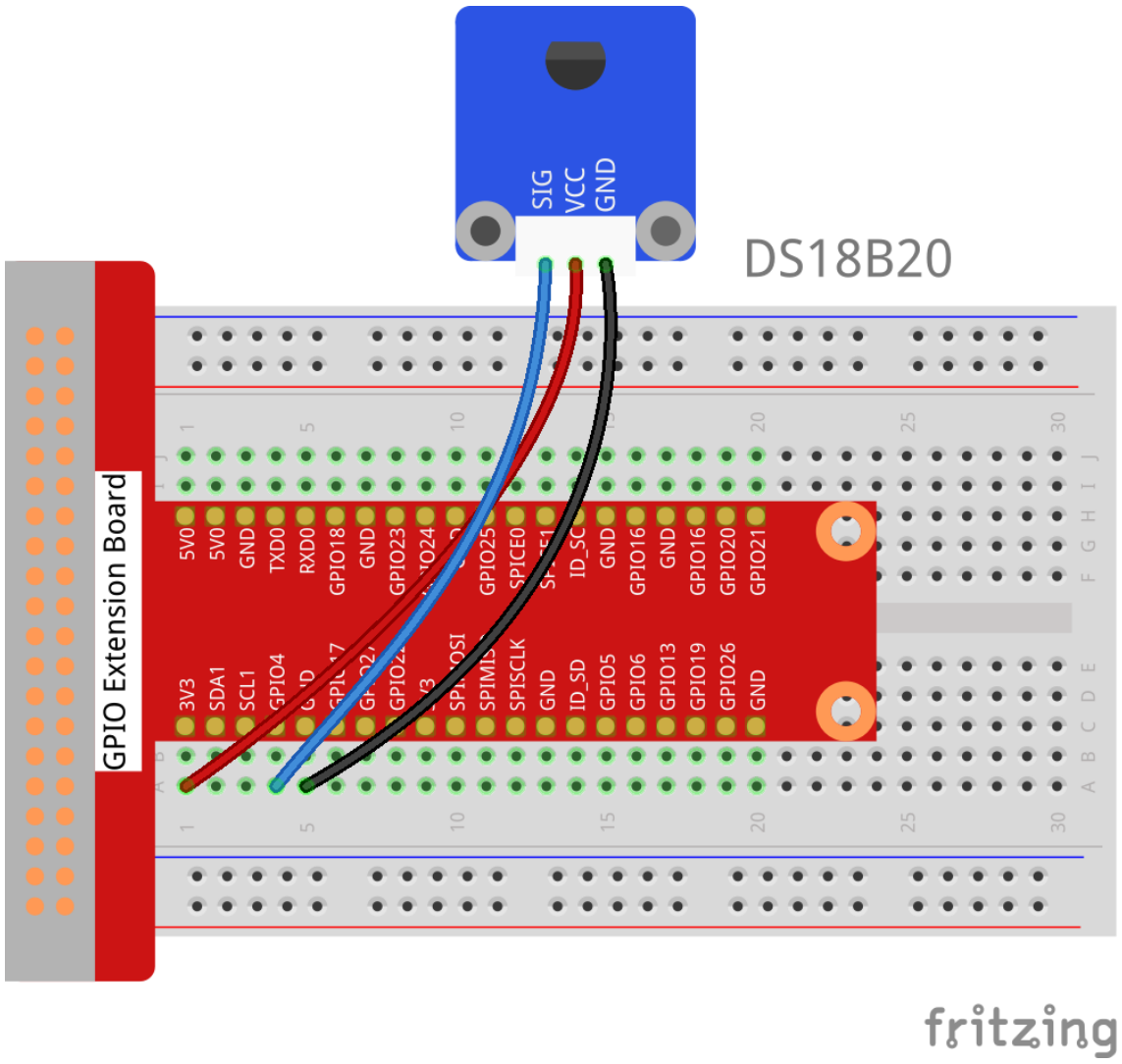
The schematic diagram of the module is as shown below:



Procedure

Step 1: Build the circuit according to the following method.

Raspberry Pi	GPIO Extension Board	DS18B20 Temperature Sensor
GPIO7	GPIO4	SIG
3.3V	3V3	VCC
GND	GND	GND



Step 2: Upgrade your kernel.

```
sudo apt update
sudo apt upgrade
```

Step 3: You can edit that file with nano.

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

Then scroll to the bottom and type.

```
dtoverlay=w1-gpio
```

Then reboot with

```
sudo reboot
```

Mount the device drivers and confirm whether the device is working.

```
sudo modprobe w1-gpio
sudo modprobe w1-therm
cd /sys/bus/w1/devices/
ls
```

The result should be as follows:

```
root@rasberrypi:/sys/bus/w1/devices# ls
28-00000495db35 w1_bus_master1
```

The serial number of the DS18B20 may vary depending on the client. In this example, "28-00000495db35" is the serial number.

Step 4: Check the current temperature.

```
cd 28-00000495db35
ls
```

The result should be as follows:

```
root@rasberrypi:/sys/bus/w1/devices/28-00000495db35# ls
driver  id  name    power   subsystem   uevent  w1_slave
cat w1_slave
```

The result should be as follows:

```
root@raspberrypi:/sys/bus/w1_slave/28-00000495db35# cat w1_slave
a3 01 4b 46 7f ff 0d 10 ce : crc=ce YES
a3 01 4b 46 7f ff 0d 10 ce t=26187
```

The second line's "t" value corresponds to the temperature. To convert to Celsius, divide by 1000.

For C Users:

Step 2: Change directory and edit.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/26_ds18b20/
nano ds18b20.c
```

Find the following line, replace "28-00000495db35" with your sensor address. Save and exit.

```
char* addr = "/sys/bus/w1/devices/28-00000495db35/w1_slave";
```

Step 6: Compile.

```
gcc ds18b20.c -lwiringPi
```

Step 7: Run.

```
sudo ./a.out
```

For Python Users:

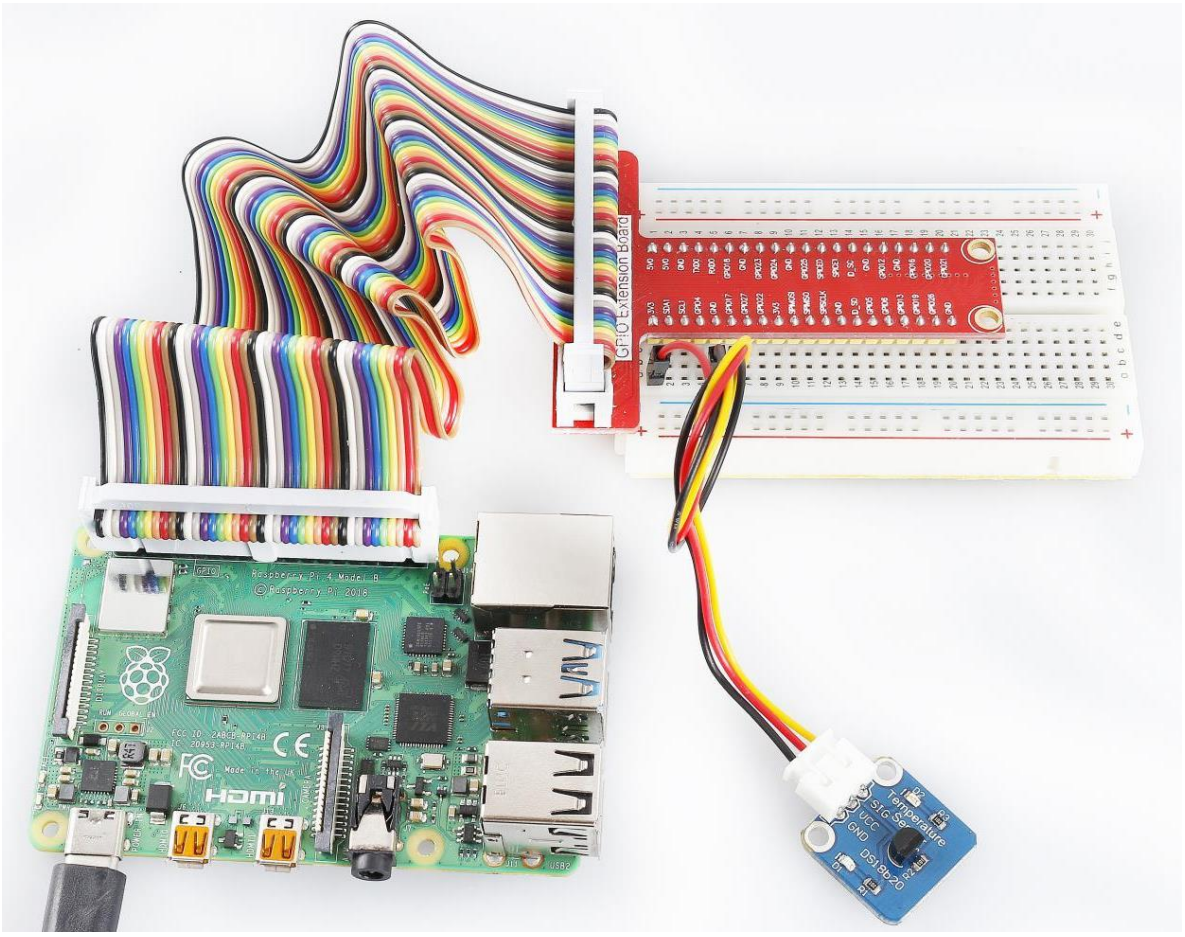
Step 5: Change directory and edit.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
nano 26_ds18b20.py
```

Step 6: Run.

```
sudo python3 26_ds18b20.py
```

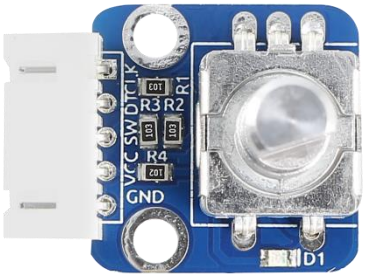
Now, you can see the current temperature value displayed on the screen.


GPIO Extension Board
Temperature
SIG
VCC
GND
DS18b20

Lesson 27 Rotary Encoder Module

Introduction

A rotary encoder is an electro-mechanical device that converts the angular position or motion of a shaft or axle to analog or digital code. Rotary encoders are usually placed at the side which is perpendicular to the shaft. They act as sensors for detecting angle, speed, length, position, and acceleration in the field of automation.



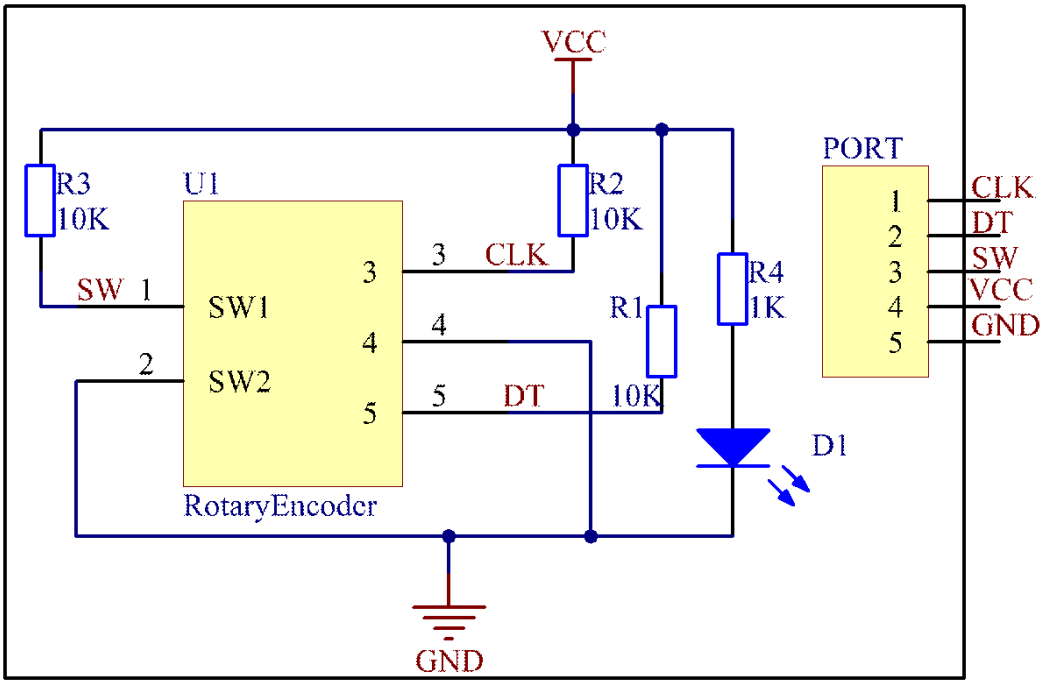
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Rotary Encoder module
- 1 * 5-Pin Non-Reversible Cable

Purpose

Most rotary encoders have 5 pins with three functions; turning left or right and pressing down. Pin 1 and 2 are switch wiring terminals used to press, Pin 4 is generally connected to ground, and Pin 3 and 5 are connected to a pull-up resistor to the microprocessor. In this experiment they are connected to GPIO0 and GPIO1 on the Raspberry Pi. When the module is rotated left and right, there will be pulse inputs in pins 1 and 3.

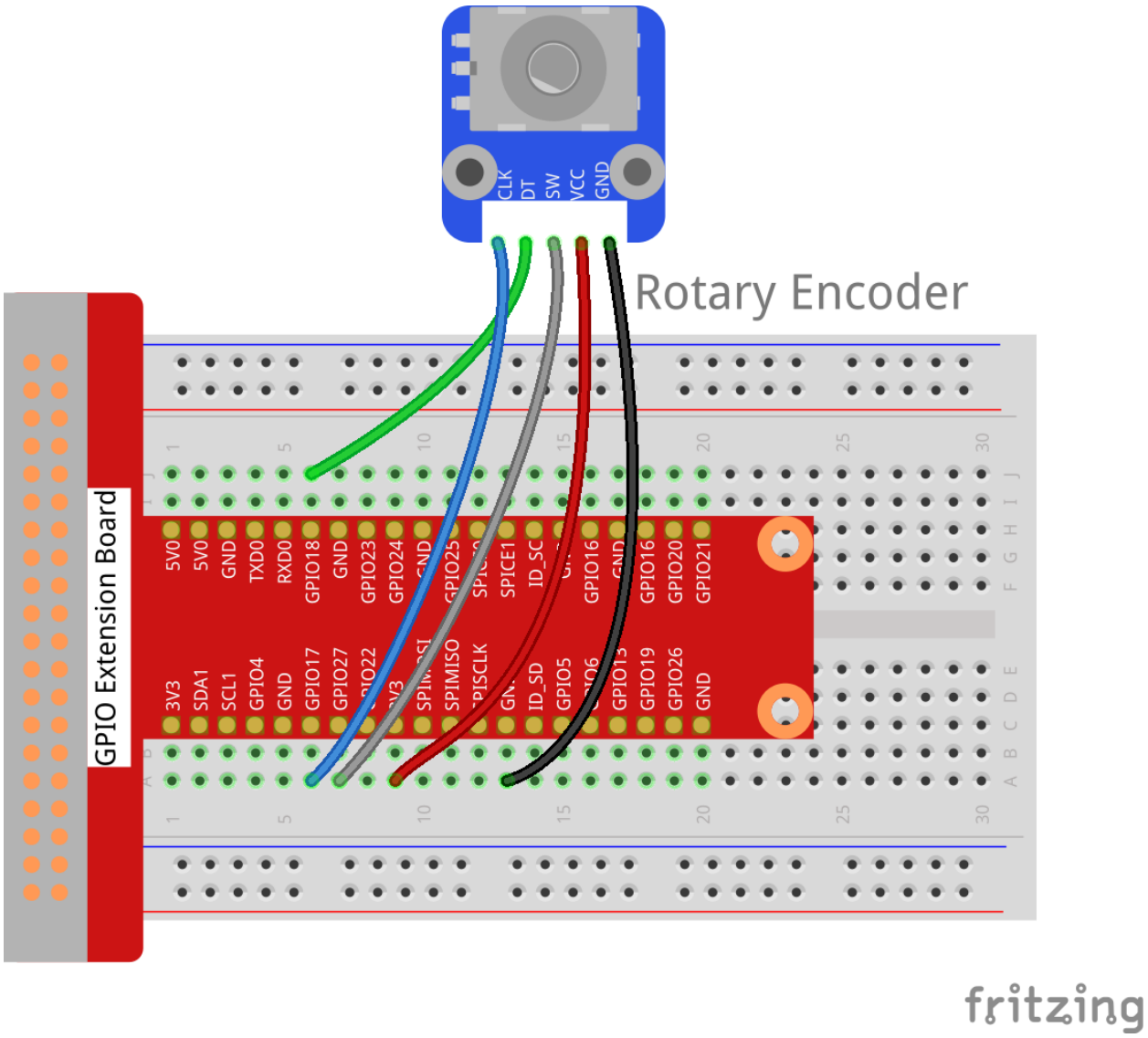
If the output of 1 and 2 are both high, the switch rotates clockwise. When both are low, it will rotate counterclockwise. As a result, during SCM programming you can set the output of 1 to high and know based on the rotation direction the state of output 2.



Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Rotary Encoder Module
GPIO0	GPIO17	CLK
GPIO1	GPIO18	DT
GPIO2	GPIO27	SW
3.3V	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/27_rotary_encoder/
```

Step 3: Compile.

```
gcc rotary_encoder.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

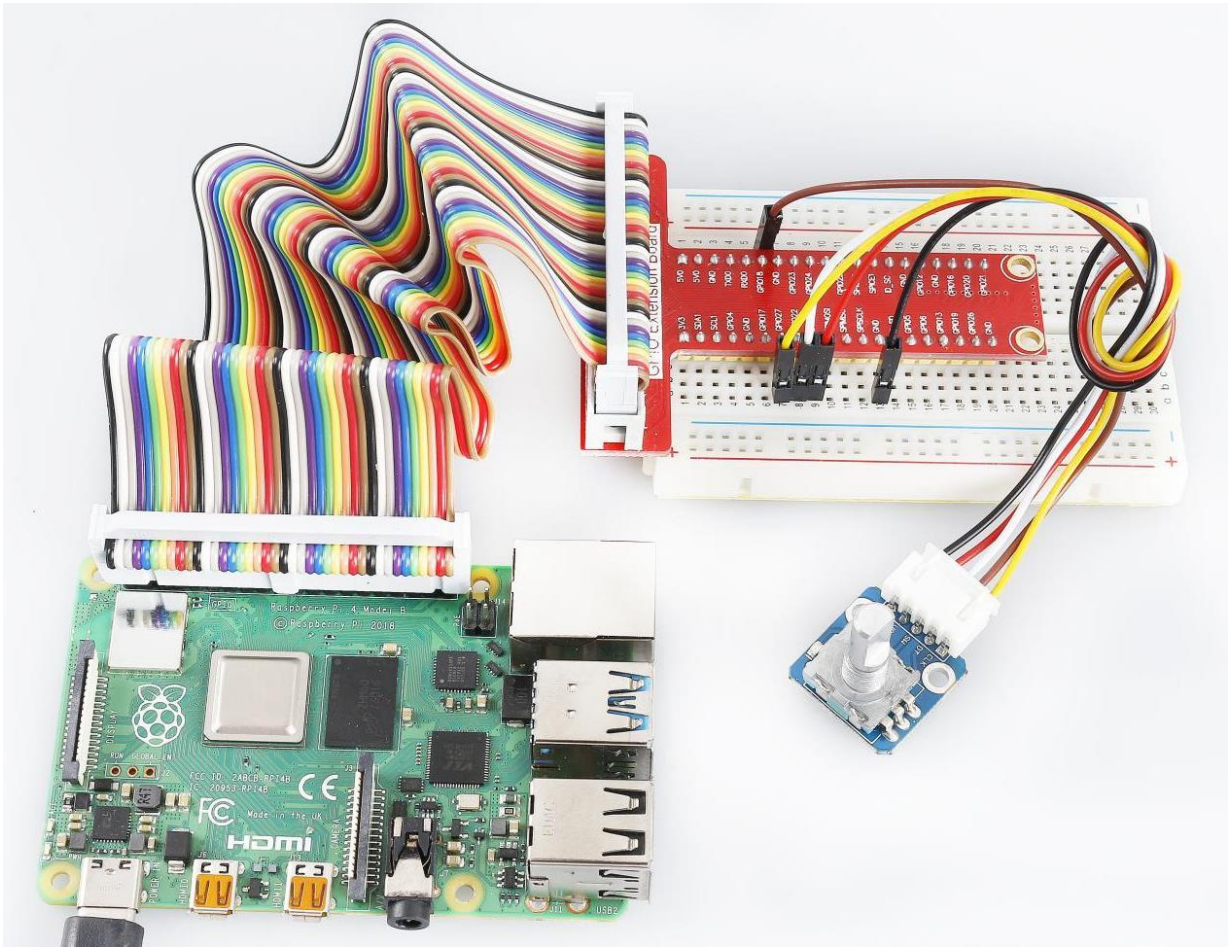
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 27_rotary_encoder.py
```

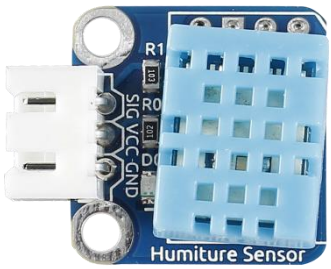
Now rotate the shaft of the rotary encoder, and the value printed on the screen will change. Rotate the rotary encoder clockwise, the value will increase; Rotate it counterclockwise, the value will decrease; Press the rotary encoder, the value will be reset to 0.



Lesson 28 Humiture Sensor

Introduction

The DHT11 digital temperature and humidity sensor is a composite sensor that gives a calibrated digital signal as output. This technology provides high reliability and excellent long-term stability.

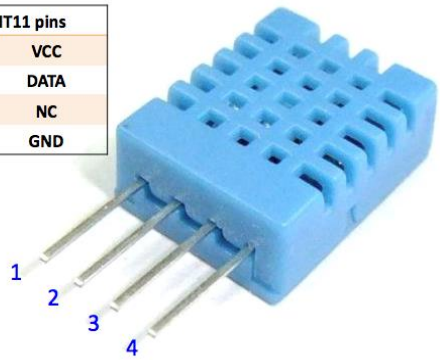


Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Temperature-Humidity (Humiture) module
- 1 * 3-Pin Non-Reversible Cable

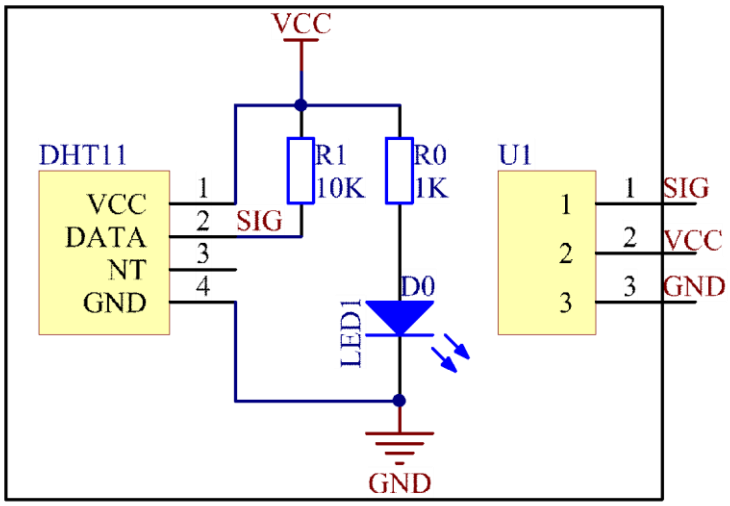
Purpose

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



The module has three pins available: VCC, GND, and DATA. The communication process begins with the DATA line sending a start signal to the DHT11, which returns an answer signal. The host begins to receive the 40-bit humiture data (8-bit humidity integer + 8-bit humidity decimal + 8-bit temperature integer + 8-bit temperature decimal + 8-bit checksum).

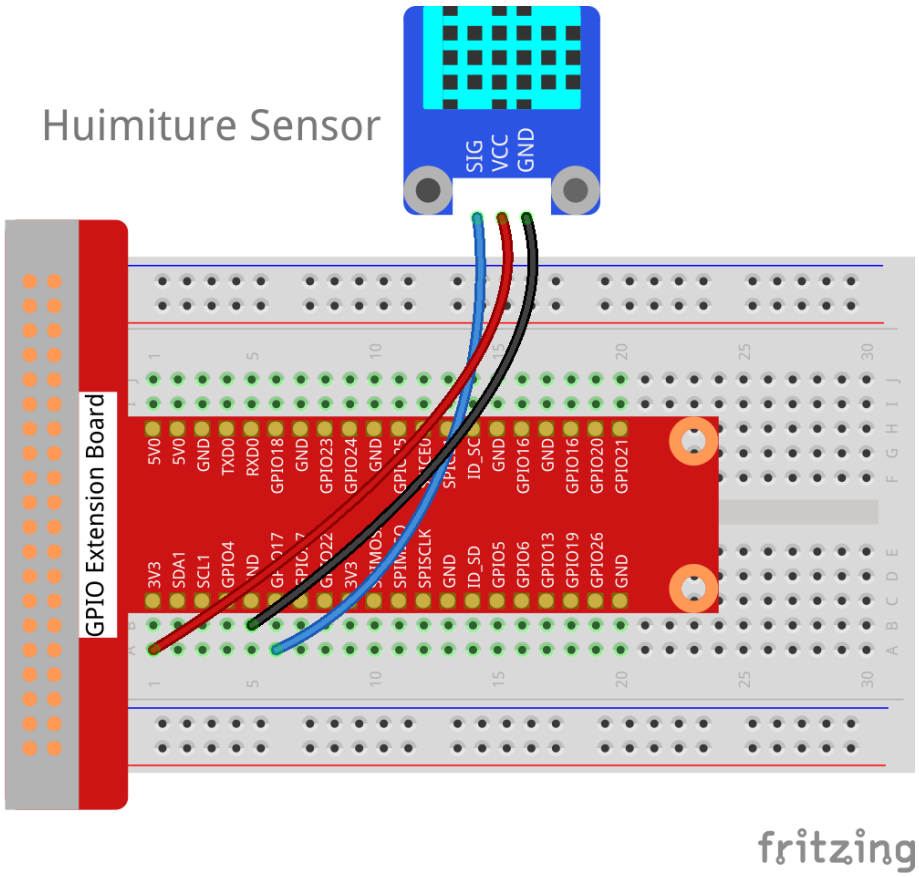
The datasheet for the module is as follows:



Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Humiture Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/28_humiture/
```

Step 3: Compile.

```
gcc humiture.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

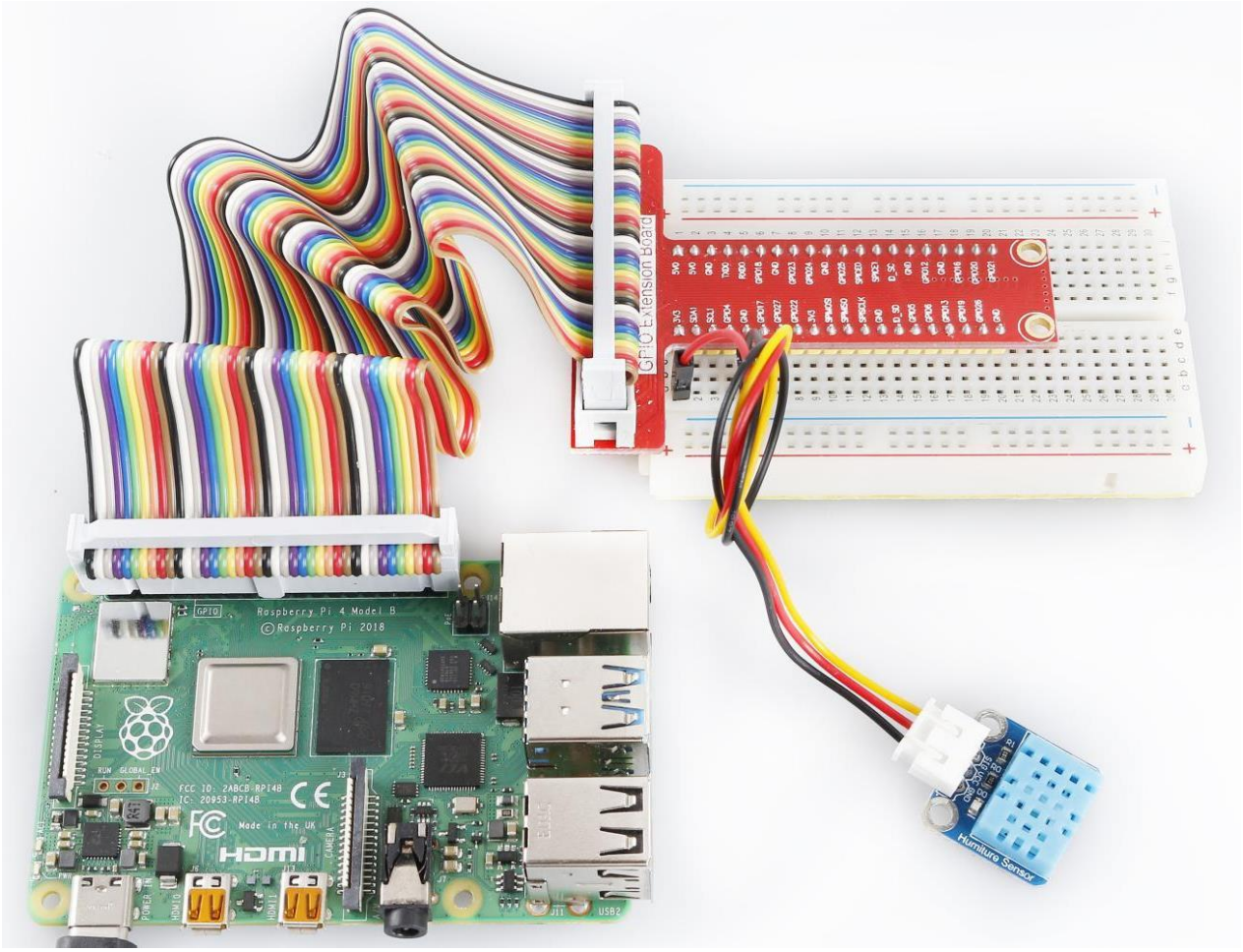
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 28_humiture.py
```

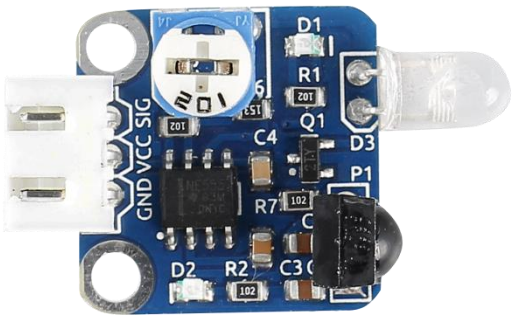
The humidity and temperature values will now print to the screen.


GPIO Extension Board
Raspberry Pi 4 Model B
©Raspberry Pi 2018
Made in the UK
HDMI
Humiture Sensor

Lesson 29 IR Obstacle Avoidance Module

Introduction

The IR obstacle avoidance module uses infrared light to detect obstacles in front of the sensor.



Required Components

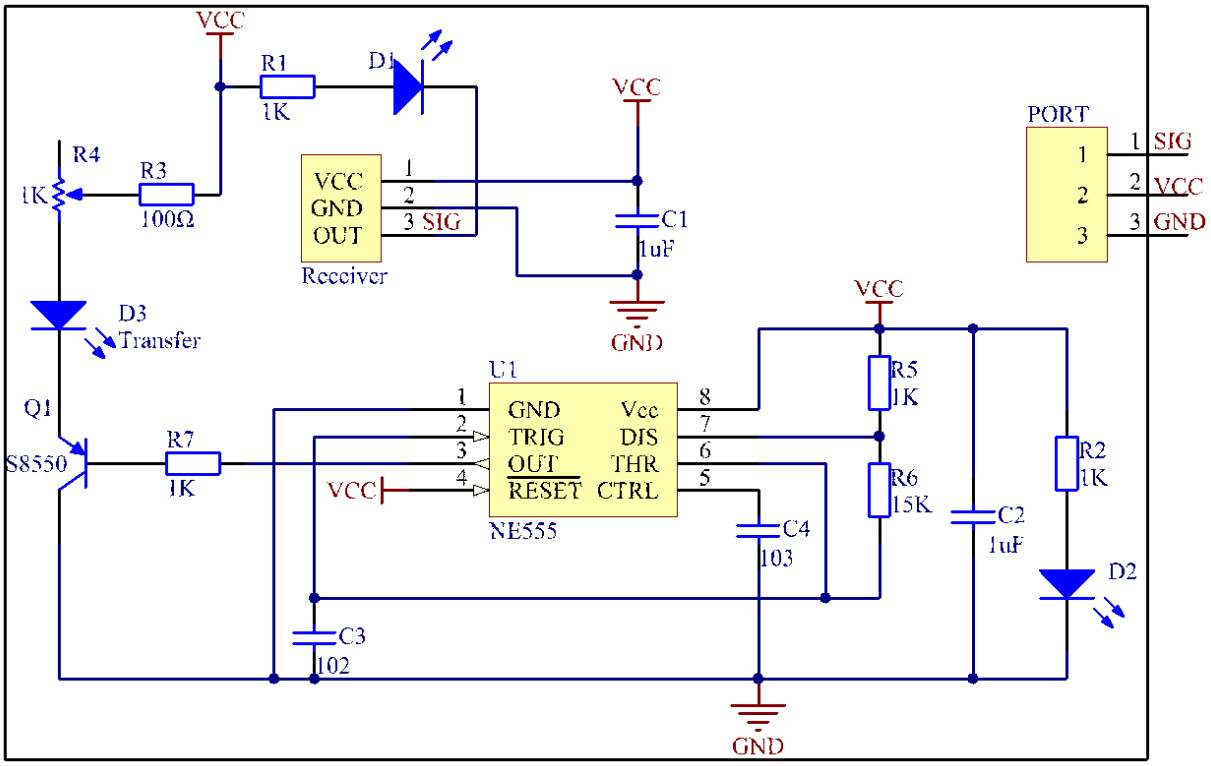
- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * IR Obstacle module
- 1 * 3-Pin Non-Reversible Cable

Purpose

An obstacle avoidance sensor consists of an infrared-transmitter, infrared-receiver, and a potentiometer. It works by sending out IR light, and if none is reflected back into the receiver (from bouncing off a surrounding object), the sensor interprets the way as being all-clear. An obstacle is detected when IR light reflects back to the module.

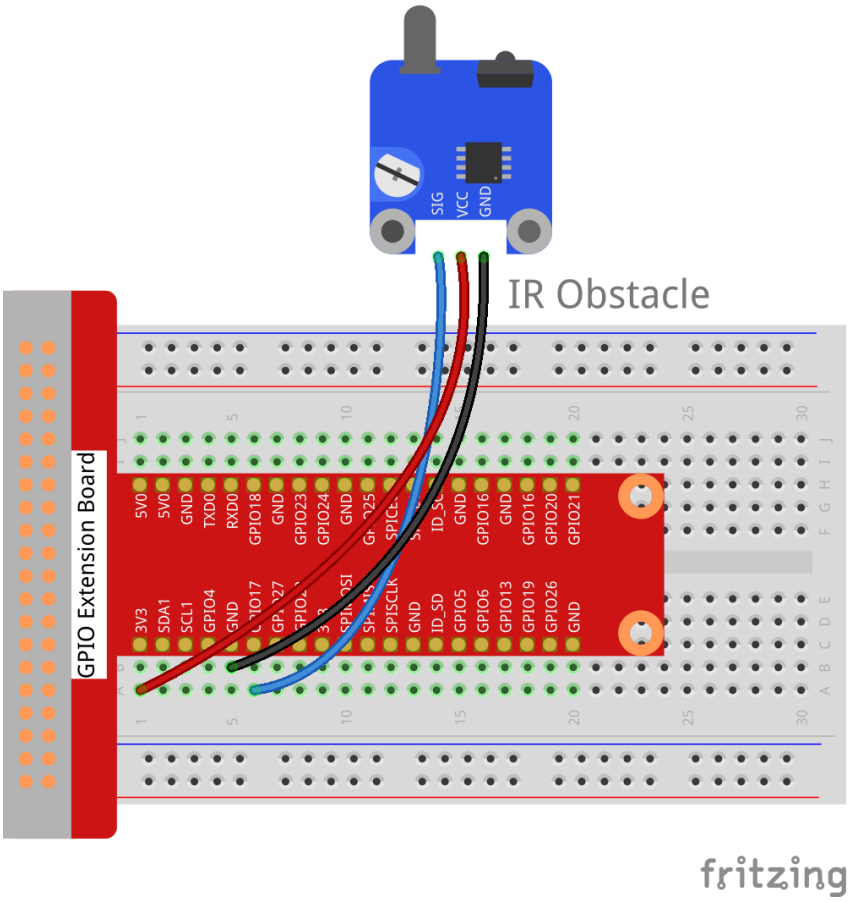
Note: The detection distance of the infrared can be adjusted using the potentiometer.

The schematic diagram of the module is as shown below:



Procedure

Step 1: Build the circuit.



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/30_ir_obstacle/
```

Step 3: Compile.

```
gcc ir_obstacle.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

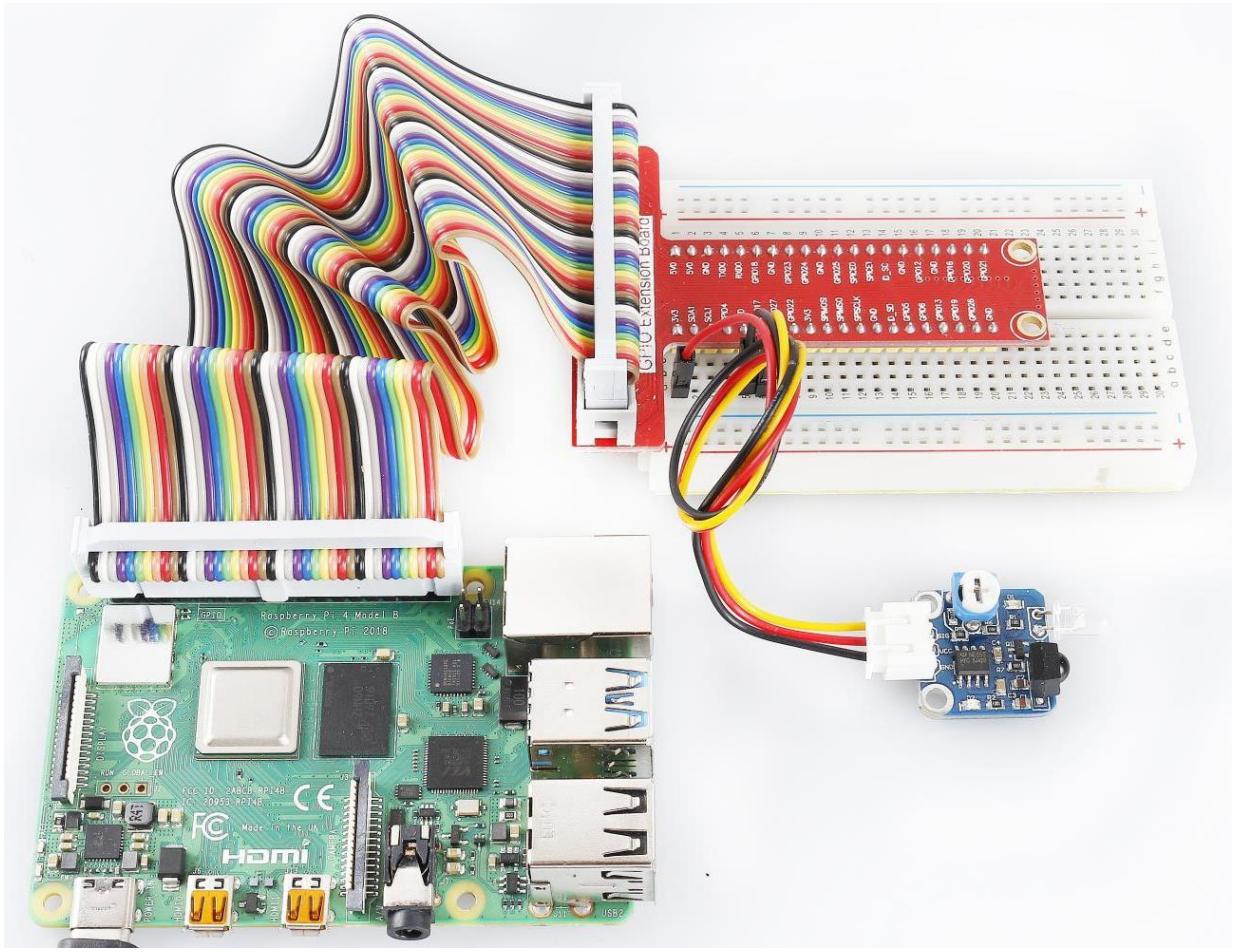
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 30_ir_obstacle.py
```

When an obstacle is in front of the module, the words "Detected Barrier!" will print to the screen.


GPIO Extension Board
Raspberry Pi 4 Model B
© Raspberry Pi 2018
HDMI

Lesson 30 I2C LCD1602

Introduction

The LCD1602 module is a liquid crystal display that can display 32 (16*2) characters at a time. It has 16 pins, 7 of which can be used simultaneously. A PCF8574 I2C chip can be used to expand the I/O ports to reduce the number of GPIO pins being occupied.

Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * I2C LCD1602
- Jumper Wires

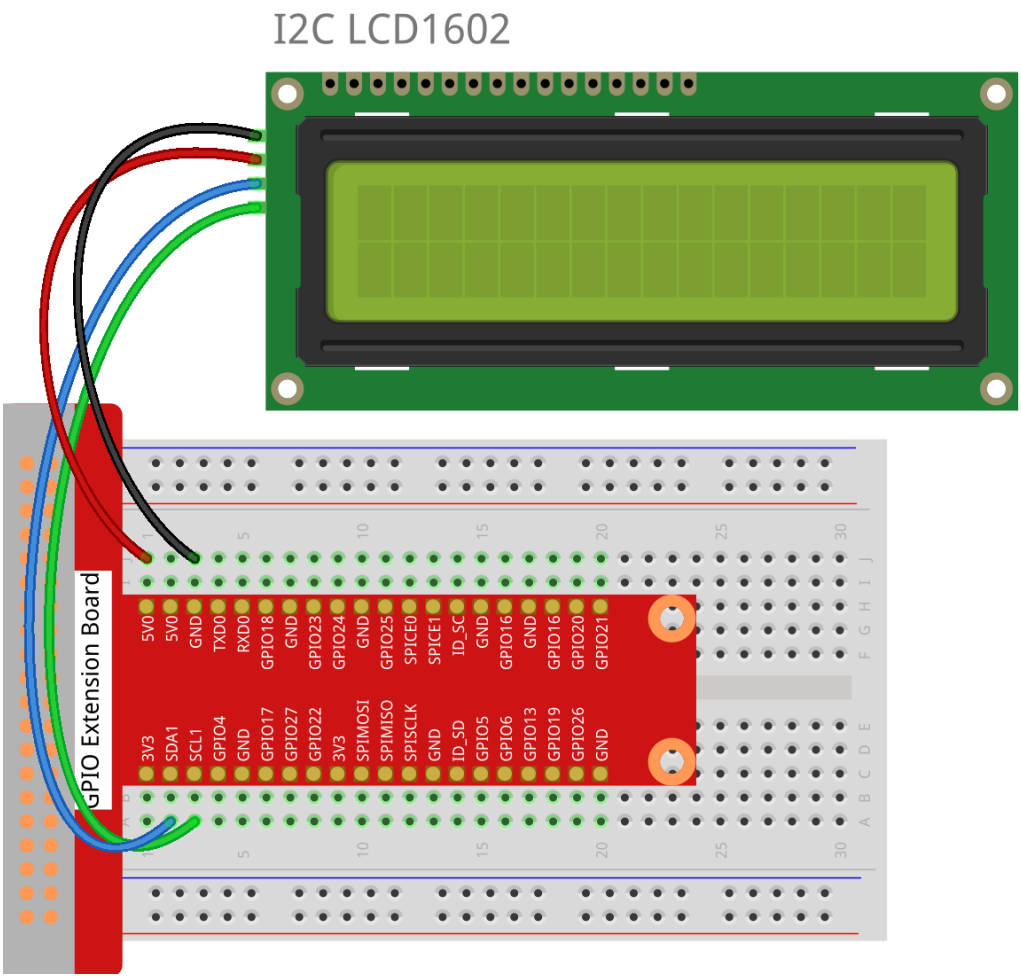
Purpose

In this experiment, I2C is used to configure the LCD and display characters. The I2C slave address of the module is 0x27.

Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	I2C LCD1602 Module
SCL	SCL1	SCL
SDA	SDA1	SDA
5V	5V0	VCC
GND	GND	GND



Step 2: Setup I2C (see Appendix. If you have set I2C, skip this step.)

For C Users:

Step 3: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/30_i2c_lcd1602/
```

Step 4: Compile.

```
gcc i2c_lcd1602.c -lwiringPi
```

Step 5: Run.

```
sudo ./a.out
```

For Python Users:

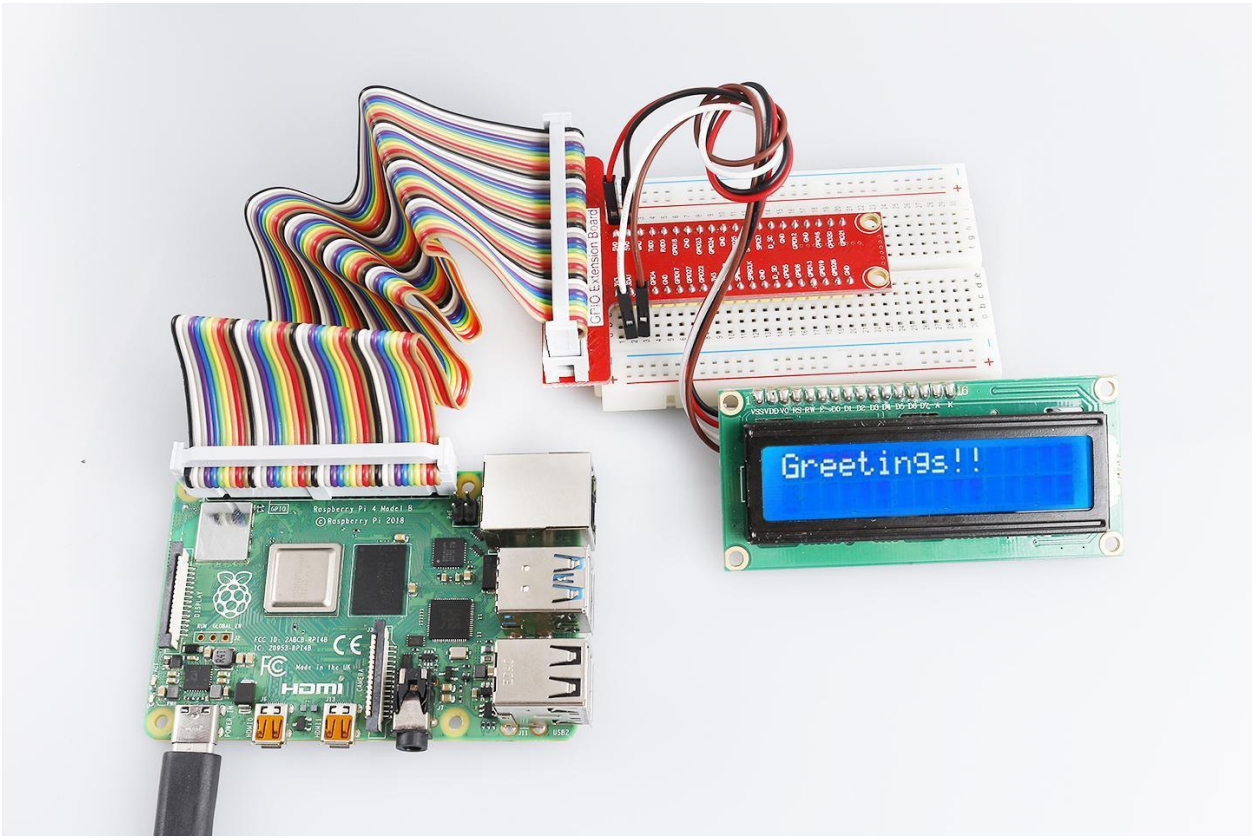
Step 3: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 4: Run.

```
sudo python3 30_i2c_lcd1602.py
```

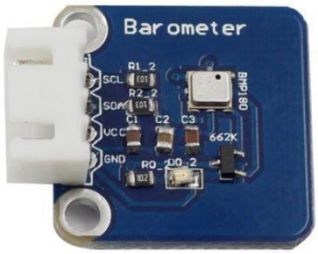
Now you can see "Greetings!!" displayed on the LCD.


GPIO Extension Board
Greetings!!

Lesson 31 Barometer-BMP180 Module

Introduction

The BMP180 barometer is a modern high-performance barometric pressure sensor. They are used in advanced mobile devices like smart phones and tablets. Compliant with the older BMP085, it boasts improvements like a smaller size and more digital interfaces.



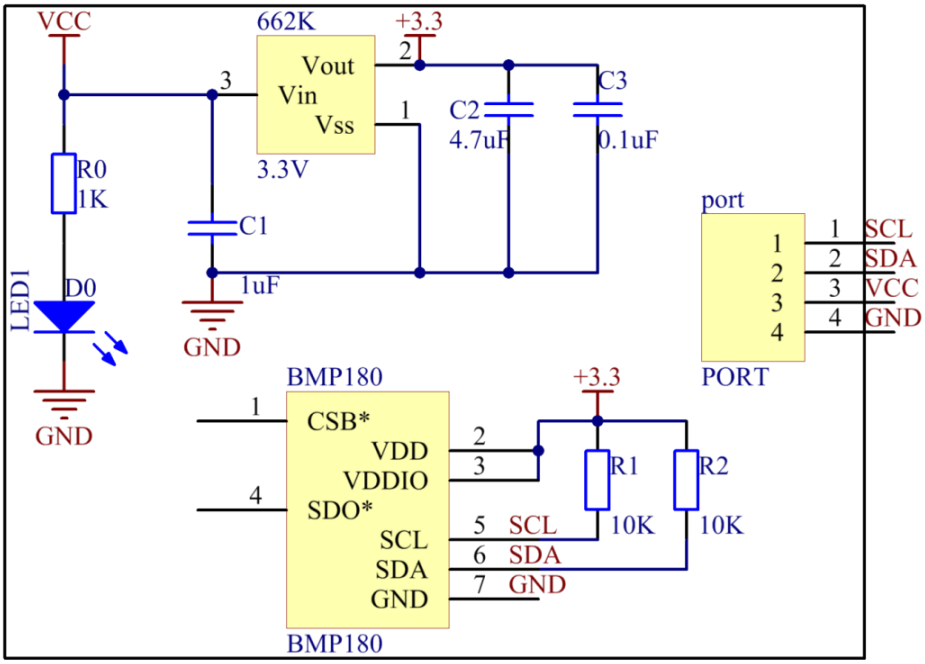
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Barometer module
- 1 * 4-Pin Non-Reversible Cable

Purpose

In this experiment, the module will be used to measure air pressure and temperature.

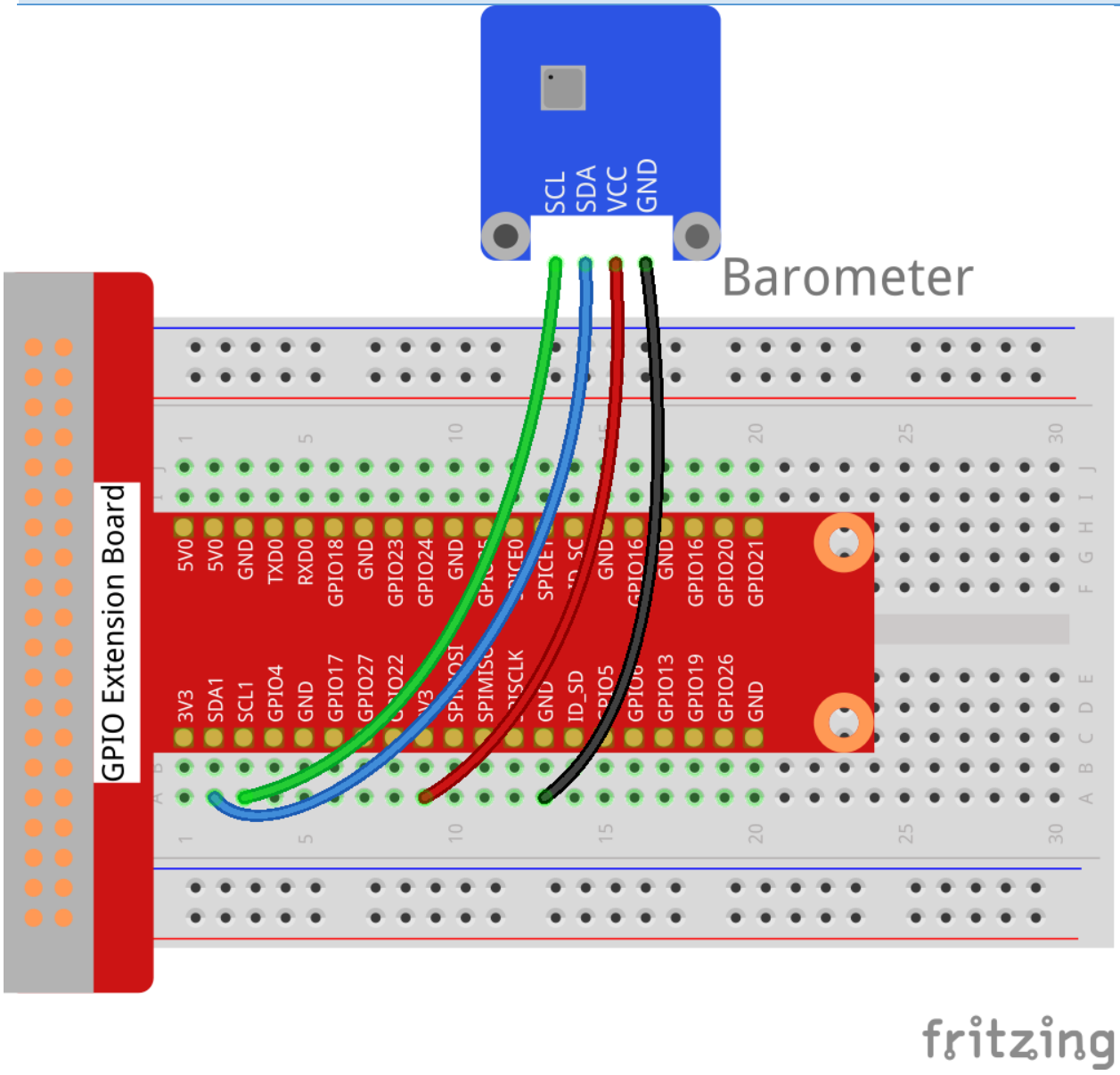
The schematic diagram of the module is as follows:



Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Barometer
SCL	SCL1	SCL
SDA	SDA1	SDA
3.3V	3V3	VCC
GND	GND	GND



Step 2: Setup I2C (see Appendix . If you have set I2C, skip this step.)

For C Users:

Step 3: Download libi2c-dev.

```
sudo apt install libi2c-dev
```

Step 4: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/31_barometer/
```

Step 5: Compile.

```
gcc barometer.c bmp180.c -lm -lwiringPi -lwiringPiDev
```

Step 6: Run.

```
sudo ./a.out
```

For Python Users:

Step 3: Install smbus for I2C.

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

Step 4: We'll need to install some utilities for the Raspberry Pi to communicate over I2C.

```
git clone https://github.com/adafruit/Adafruit_Python_BMP.git
cd Adafruit_Python_BMP
sudo python3 setup.py install
```

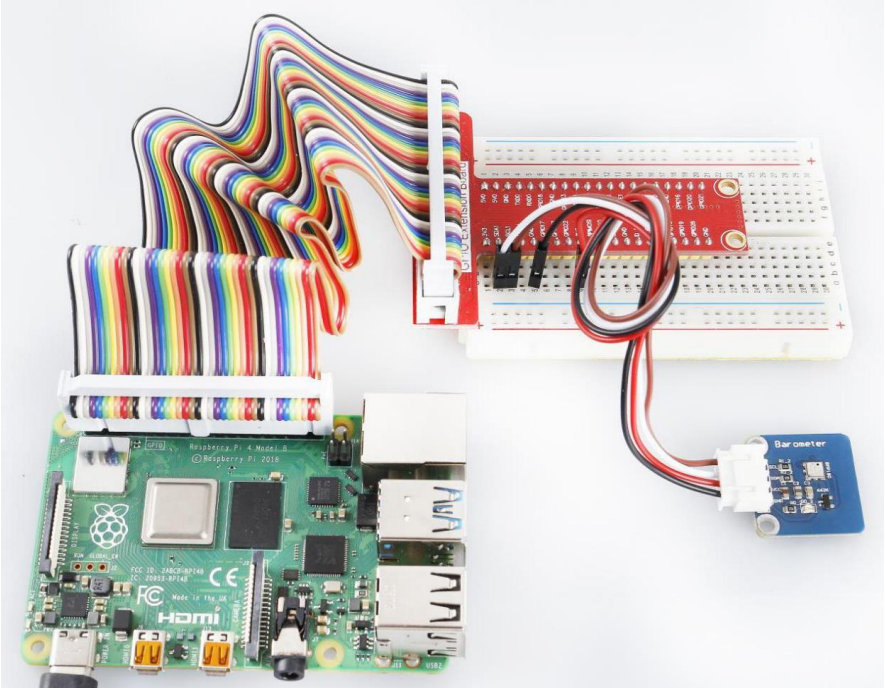
Step 5: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 6: Run.

```
sudo python3 31_barometer.py
```

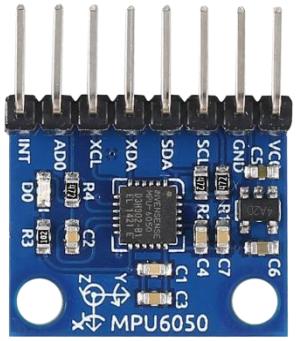
The temperature and pressure values will now be displayed on the screen.



Lesson 32 MPU6050 Gyro Acceleration Sensor

Introduction

The MPU-6050 is the world's first and only 6-axis motion tracking devices designed for the low power, low cost, and high performance requirements of smartphones, tablets and wearable sensors.



Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * MPU-6050 module
- Jumper Wires

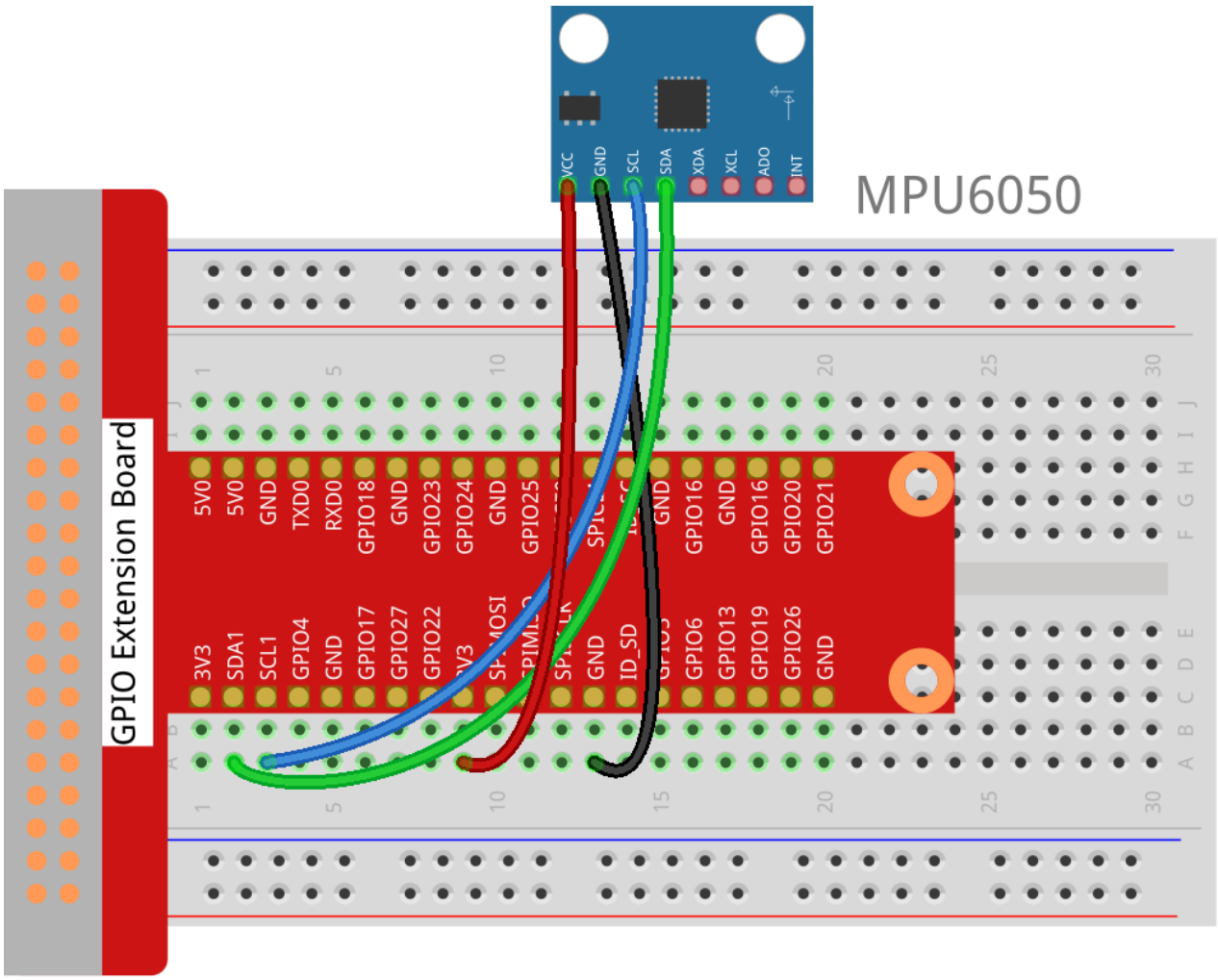
Purpose

In this experiment, I2C is used to obtain the value of the three-axis acceleration sensor and three-axis gyroscope, which are then displayed to the screen.

Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	MPU-6050 Module
SCL	SCL1	SCL
SDA	SDA1	SDA
3.3V	3V3	VCC
GND	GND	GND



Step 2: Setup I2C (see Appendix. If you have set I2C, skip this step.)

For C Users:

Step 3: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/32_mpu6050/
```

Step 4: Compile.

```
gcc 32_mpu6050.c -lwiringPi -lm
```

Step 5: Run.

```
sudo ./a.out
```

For Python Users:

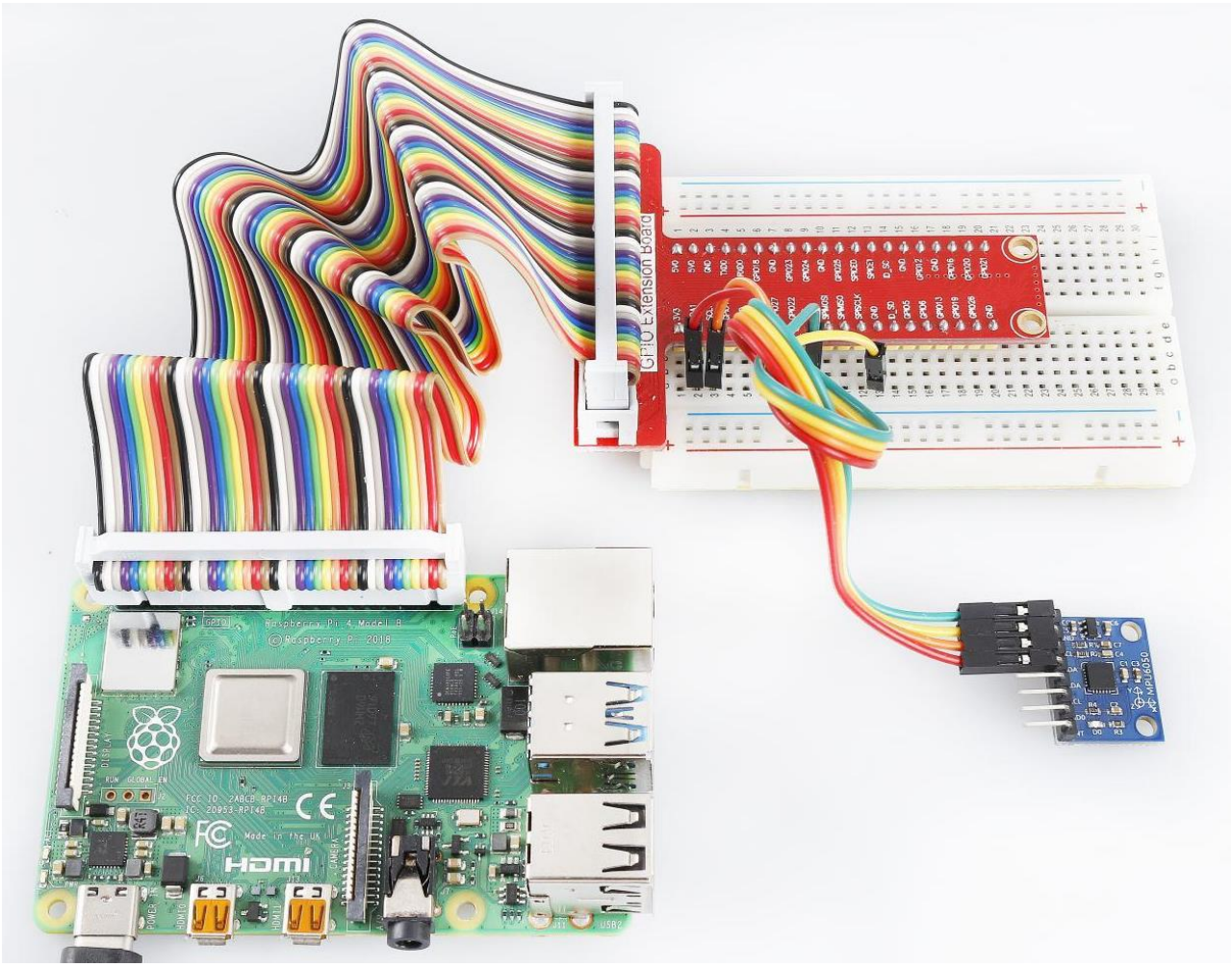
Step 3: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 4: Run.

```
sudo python3 32_mpu6050.py
```

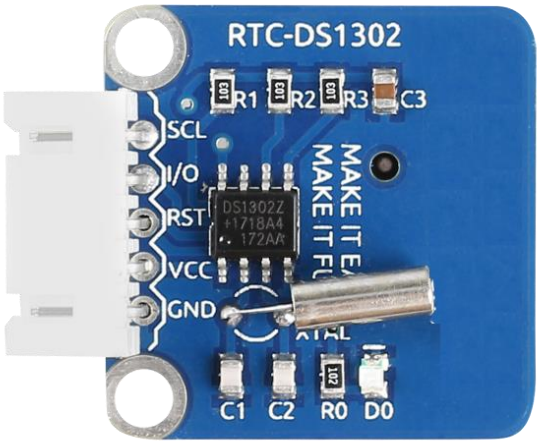
Values from the acceleration sensor, gyroscope, and XY-axis rotation will now print to the screen.


GPIO Extension Board
Raspberry Pi 4 Model B
Raspberry Pi 2018
Made in the UK
HDMI
MPU6050

Lesson 33 - RTC DS1302

Introduction

The DS1302 features a built-in low-power Real-Time Clock/calendar and 31-bytes of static RAM, and connects via a simple serial interface. The real-time clock/calendar circuit provides information about second, minute, hour, day, week, month, and year, automatically adjusting the number of days per month and days in a leap year. A choice between 24-hour and 12-hour clock display is available.



Required Components

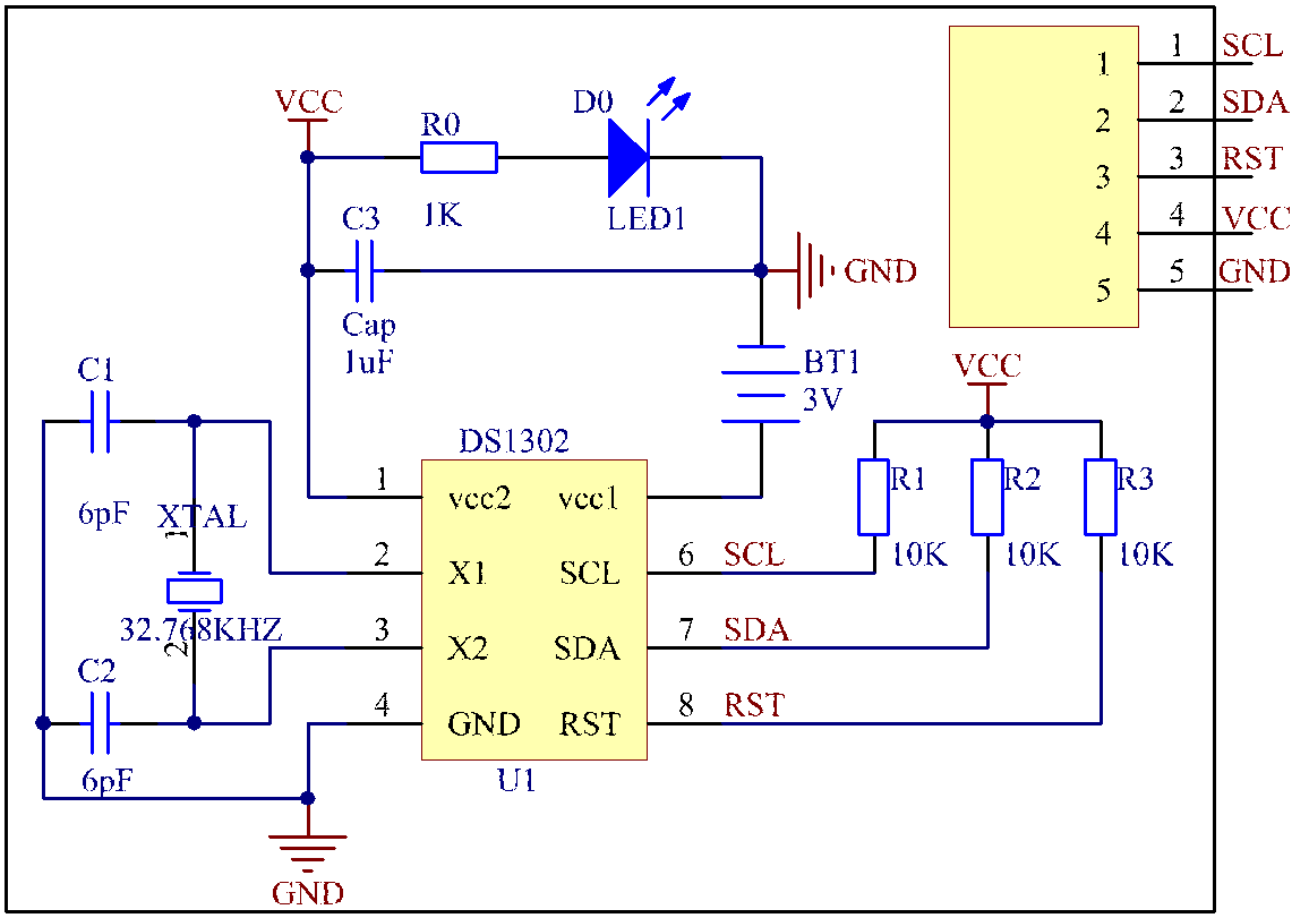
- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * DS1302 RTC module
- 1 * 5-Pin Non-Reversible Cable

Purpose

Interfacing the DS1302 with a microprocessor is simplified by using synchronous serial communication. Only three I/O wires are required to communicate with the clock/RAM: RST, serial data (SDA) and serial clock (SCL). SDA can transfer to and from the clock/RAM one byte at a time or in a burst of up to 31 bytes.

Once the time is set manually in the DS1302, the accurate time and date can be read from it by the program.

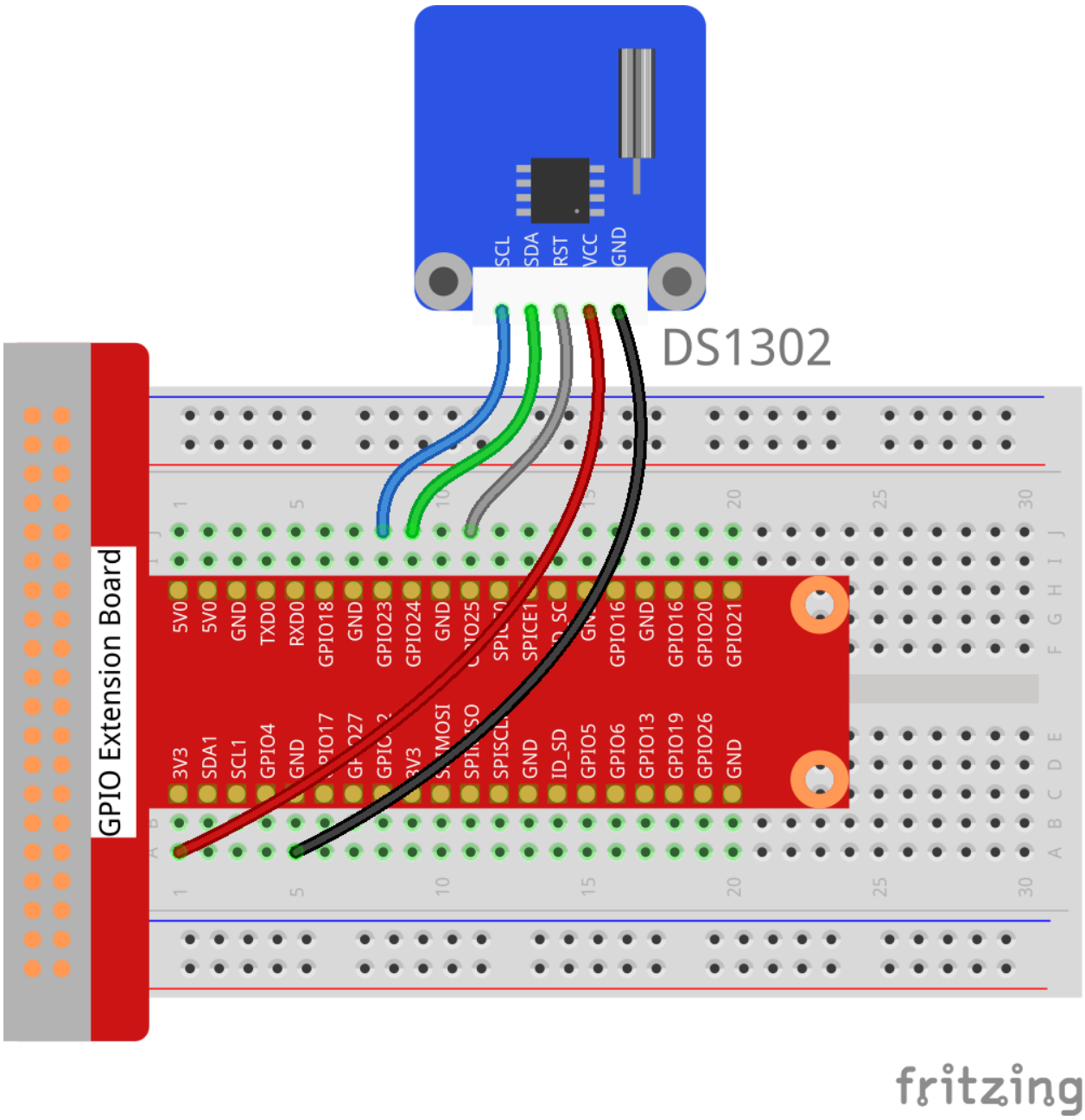
The schematic diagram of the module is as shown below:



Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	RTC DS1302 Module
GPIO4	GPIO23	SCL
GPIO5	GPIO24	I/O or SDA
GPIO6	GPIO25	RST
3.3V	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/33_ds1302/
```

Step 3: Compile:

```
gcc rtc_ds1302.c -lwiringPi -lwiringPiDev
```

Step 4: Set up time by:

```
sudo ./a.out -sdsc
```

Set year, month, date as YYYYMMDD

Set hour, minute, second as HHMMSS (24-hour clock)

Set weekday (0 as Sunday)

Step 5: Run:

```
sudo ./a.out
```

For Python Users:

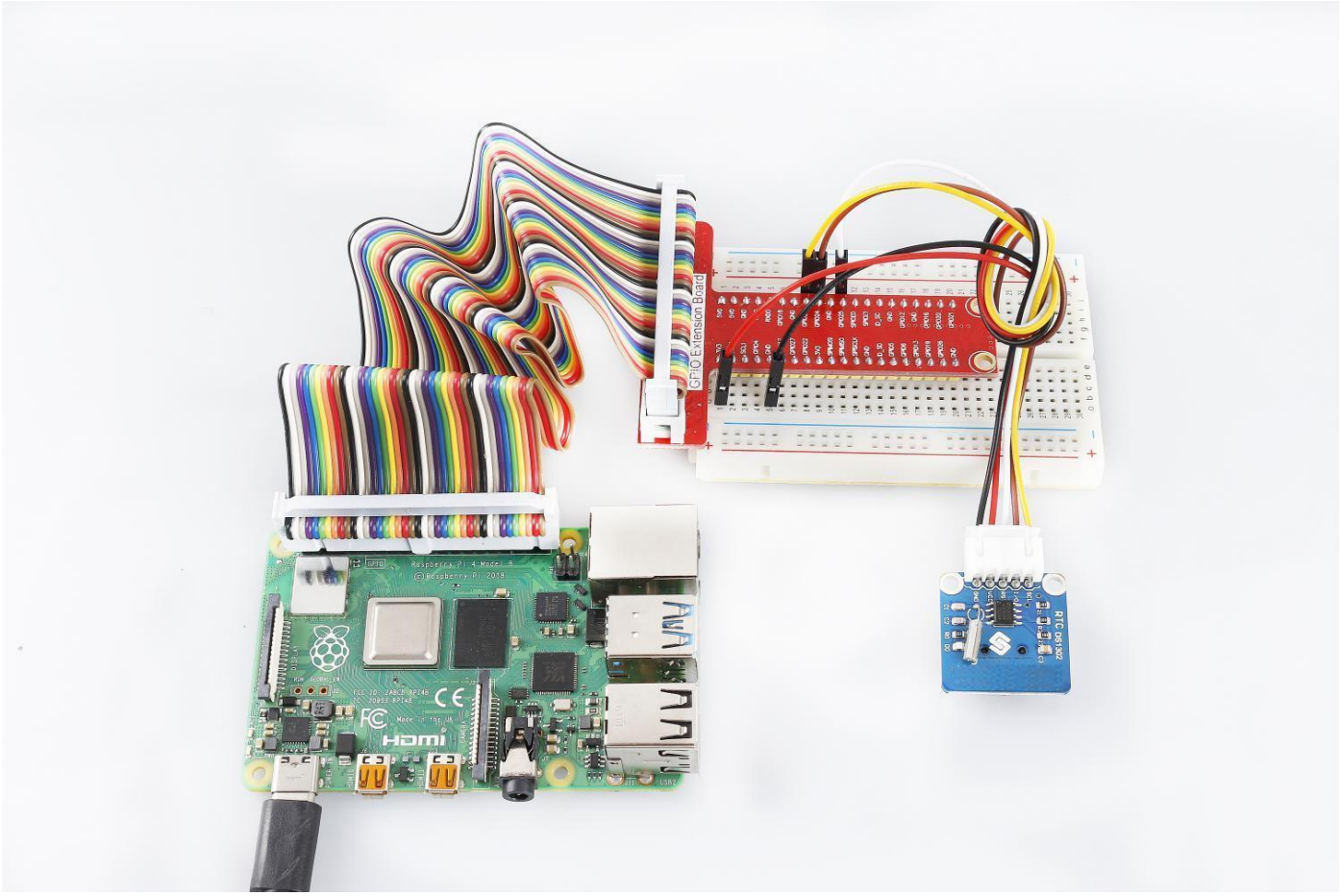
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 33_ds1302.py
```

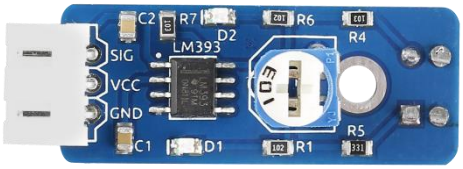
Now you can see the time on the screen.



Lesson 34 - Tracking Sensor

Introduction

This infrared tracking module uses a TRT5000 sensor and a blue-colored infrared LED. This LED, when powered, emits infrared light invisible to the human eye. The receiver's photo-transistor conductivity changes depending on the amount of infrared light received.



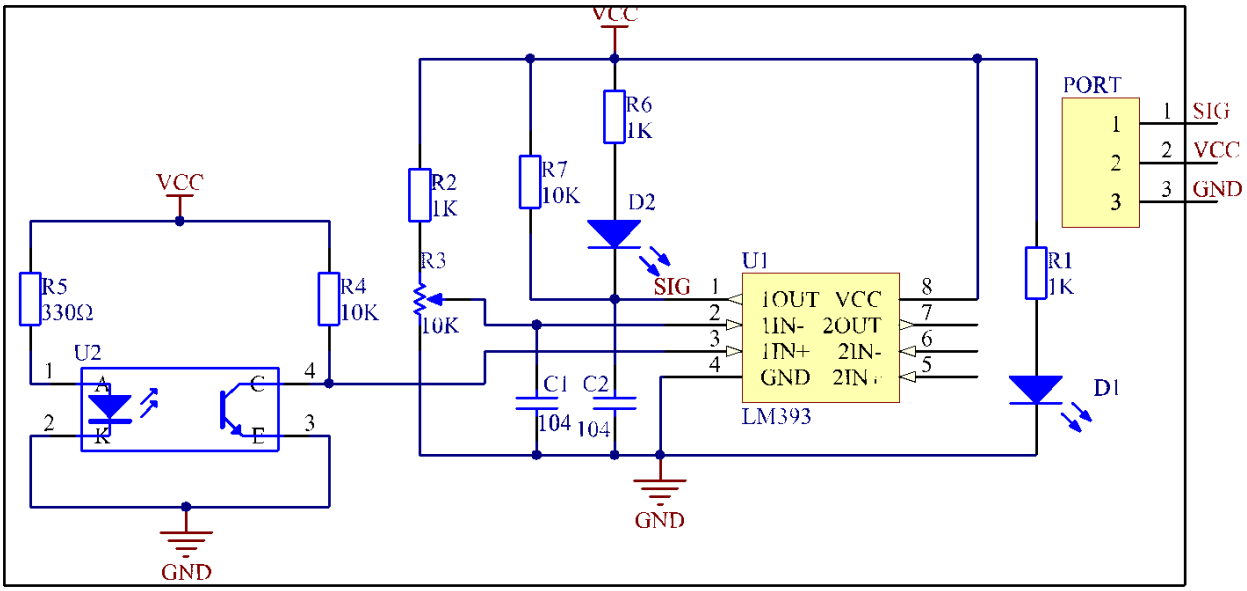
Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Tracking sensor module
- 1 * 3-Pin Non-Reversible Cable

Purpose

When the infrared transmitter emits its rays against a white surface, they will be reflected and caught by the IR receiver causing the pin SIG to output at a low level. Against a black surface, the infrared rays will be absorbed reflecting nothing back to the receiver, causing pin SIG to output at a high level.

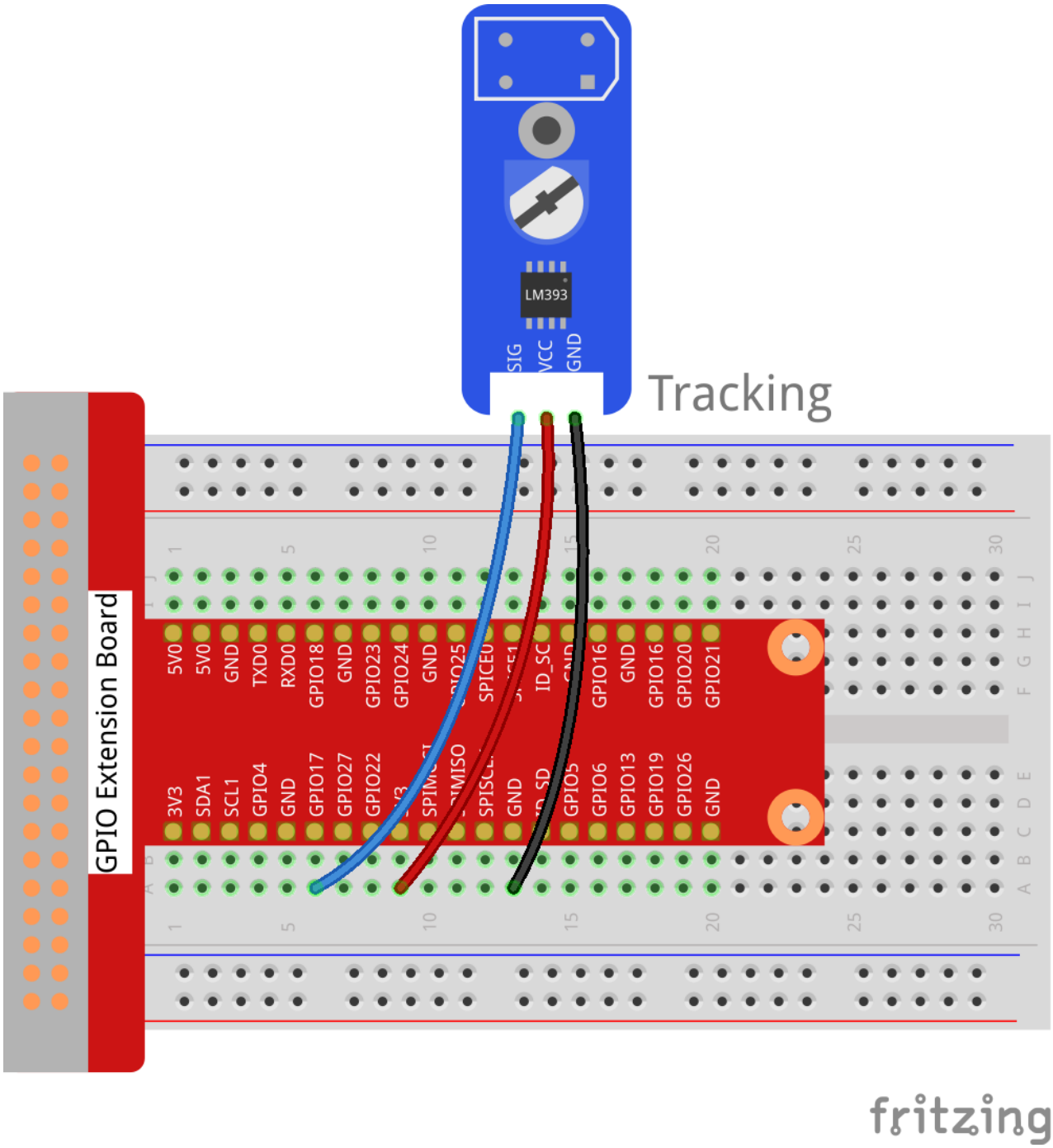
The schematic diagram of the module is as shown below:



Procedure

Step 1: Build the circuit.

Raspberry Pi	GPIO Extension Board	Tracking Sensor Module
GPIO0	GPIO17	SIG
3.3V	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/34_tracking/
```

Step 3: Compile.

```
gcc tracking.c -lwiringPi
```

Step 4: Run.

```
sudo ./a.out
```

For Python Users:

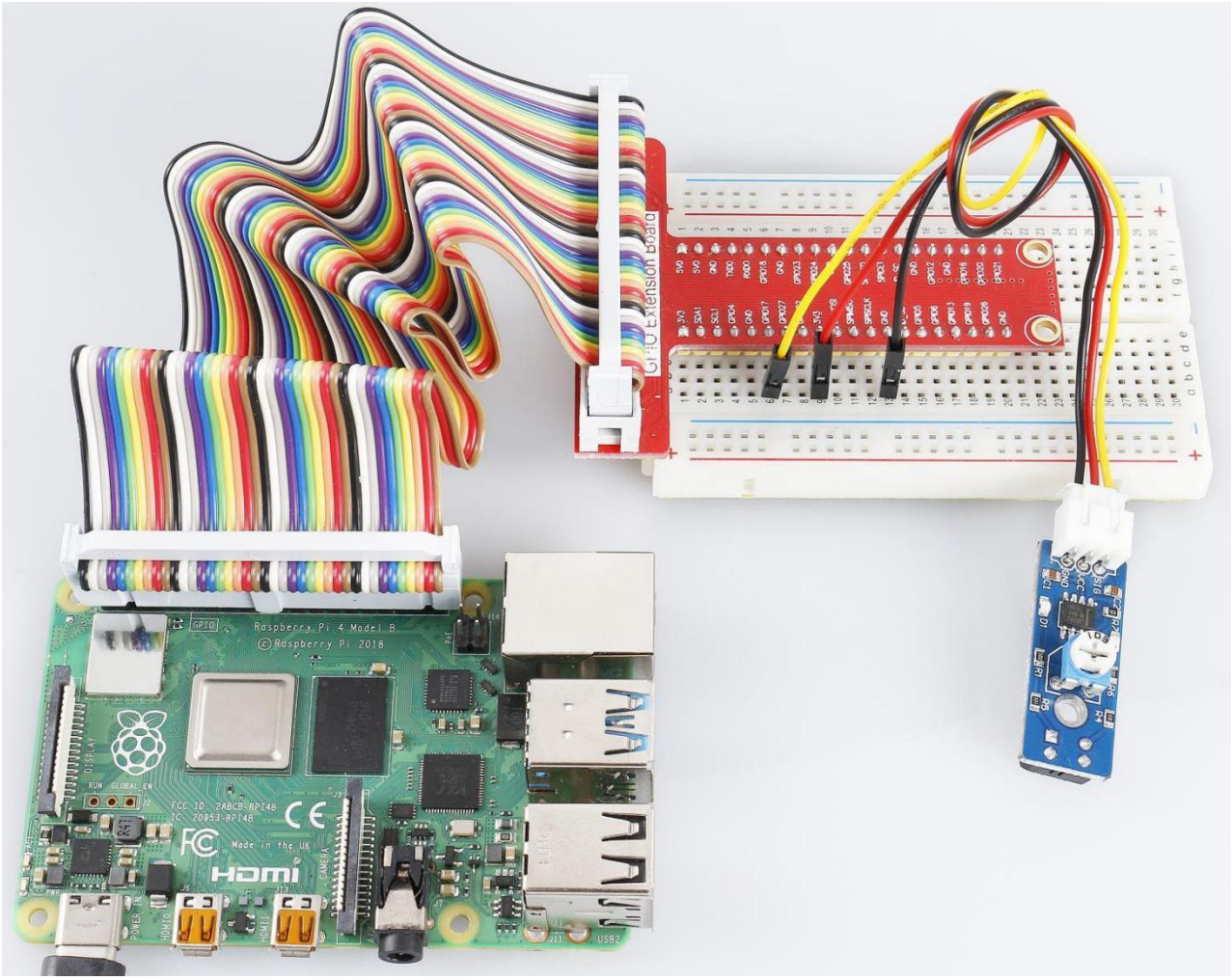
Step 2: Change directory.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 34_tracking.py
```

When the tracking sensor encounters a black line, text will be printed to the screen.



Lesson 35 - Intelligent Temperature Measurement System

Introduction

This experiment will use several modules to build an intelligent temperature measurement system.

Required Components

- 1 * Raspberry Pi
- 1 * Breadboard
- 1 * Active Buzzer
- 1 * RGB LED Module
- 1 * DS18B20 Temperature Sensor
- 1 * PCF8591
- 1 * Joystick PS2
- Jumper wires

Purpose

Building from lesson 26, this temperature system will now take input commands from the joystick module to control the limit value. The upper limit will be controlled by the Left/Right axis, and the lower limit by the Up/Down axis. Pressing the joystick will log the system out.

Procedure

Step 1: Build the circuit.

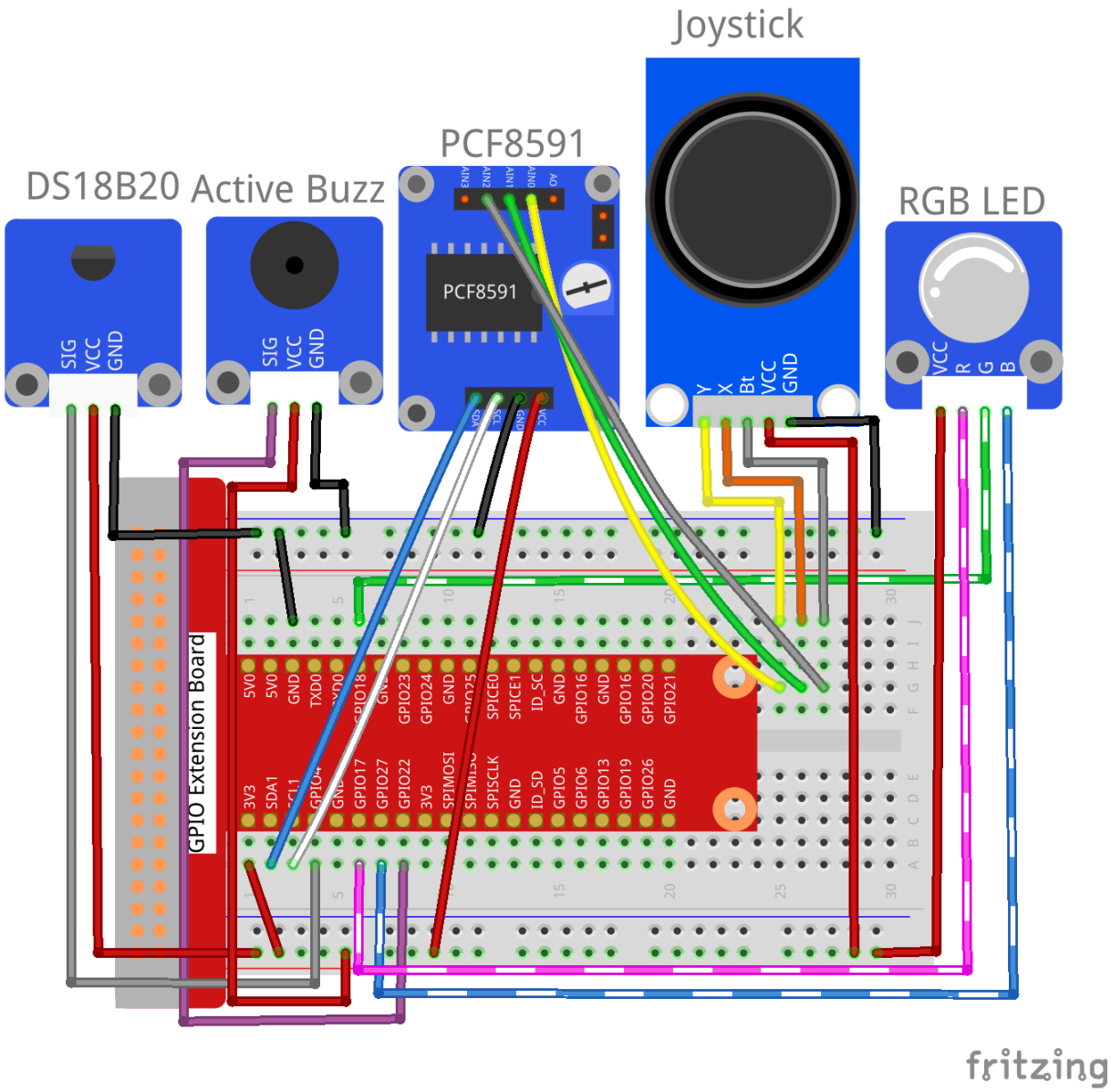
Raspberry Pi	GPIO Extension Board	DS18B20 Module
GPIO7	GPIO4	SIG
3.3V	3V3	VCC
GND	GND	GND

Raspberry Pi	GPIO Extension Board	PCF8591 Module
SDA	SDA1	SDA
SCL	SCL1	SCL
3.3V	3V3	VCC
GND	GND	GND

Joystick PS2	GPIO Extension Board	PCF8591 Module
Y	*	AIN0
X	*	AIN1
Bt	*	AIN2
VCC	3V3	*
GND	GND	*

Raspberry Pi	GPIO Extension Board	RGB LED Module
GPIO0	GPIO17	R
GPIO1	GPIO18	G
GPIO2	GPIO27	B
3.3V	3V3	VCC

Raspberry Pi	GPIO Extension Board	Active Buzzer Module
GPIO3	GPIO22	SIG
3.3V	3V3	VCC
GND	GND	GND



For C Users:

Step 2: Check the address of your sensor.

```
ls /sys/bus/w1/devices/
```

It may be like this:

```
28-031467805fff  w1_bus_master1
```

Copy or write down 28-XXXXXXX. It is the address of your sensor.

Step 3: Change directory and edit.

```
cd /home/pi/HiPi-Sensor-kit-v4.0/C/35_expand02/
nano temp_monitor.c
```

Find the function *float tempRead(void)*, and the line "fd = open(XXXXXX)". Replace "28-031467805ff" with your sensor address.

```
   float tempRead(void)
   {
   float temp;
   int i, j;
   int fd;
   int ret;

   char buf[BUFSIZE];
   char tempBuf[5];

   fd = open("/sys/bus/w1/devices/28-031467805fff/w1_slave", O_RDONLY);
#This is on one line!

   if(-1 == fd){
   perror("open device file error");
   return 1;
   }
```

Save and exit.

Step 4: Compile.

```
gcc temp_monitor.c -lwiringPi
```

Step 5: Run.

```
sudo ./a.out
```

For Python Users:

Step 2: Change directory.

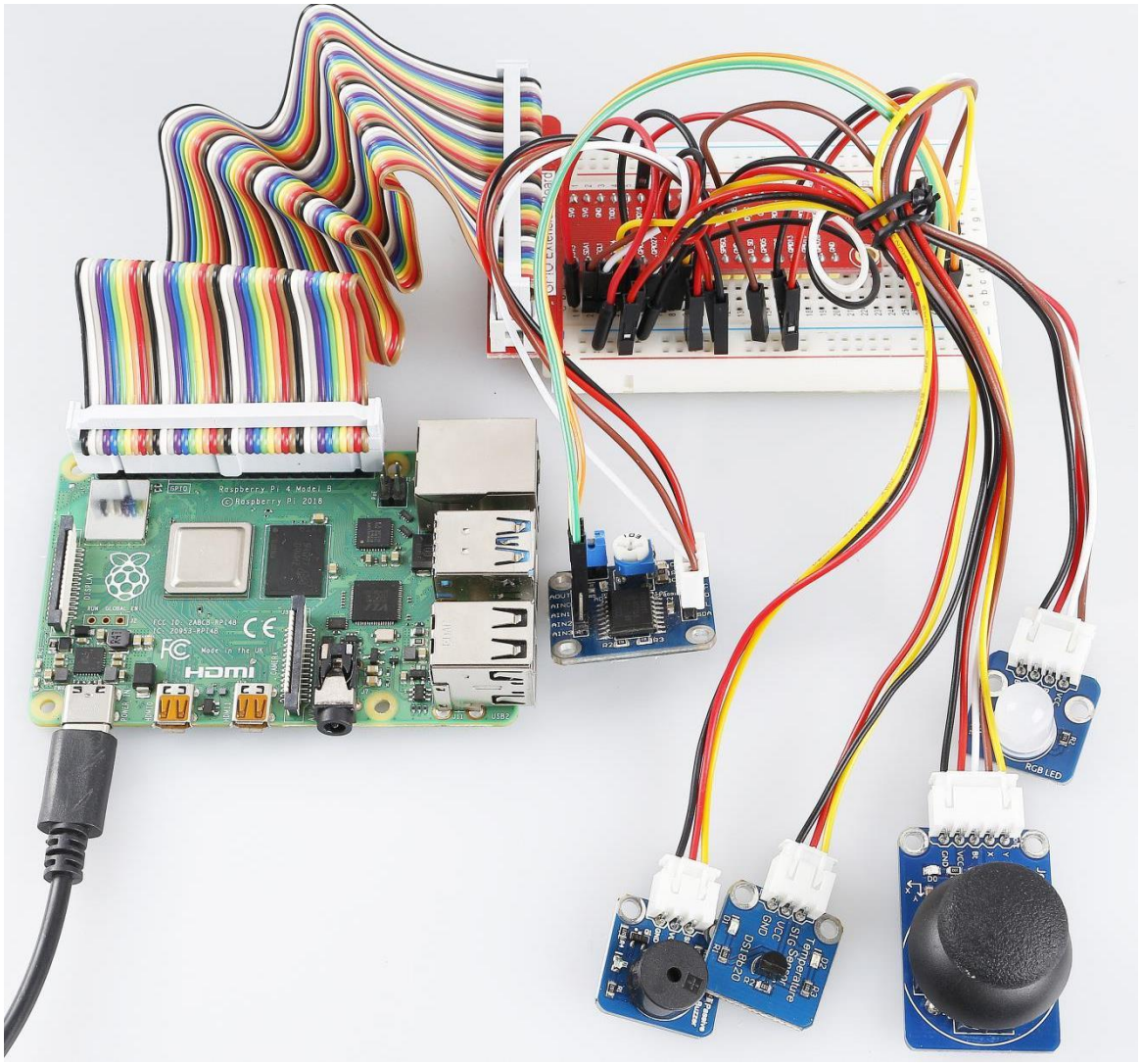
```
cd /home/pi/HiPi-Sensor-kit-v4.0/Python/
```

Step 3: Run.

```
sudo python3 35_temp_monitor.py
```

Move the joystick left and right to set the upper limit, up and down to set the lower limit.

When the ambient temperature reaches the limit of either value, the buzzer will beep.


Raspberry Pi 4 Model B
© Raspberry Pi 2018
HDMI
RGB LED
Temperature
Sig
VCC
GND
DS18B20

Appendix: Pi Preparation

In this chapter, we will learn how to prepare and use Raspberry Pi.

The Raspberry Pi can be used with a monitor and keyboard or without (which is known as headless). The headless install might require some Wi-Fi setup in a configuration file and is a bit more advanced. For this reason, we recommend that beginners start with a keyboard and monitor as described in this chapter.

If You Have a Screen

If you have a screen, you can use the NOOBS (New Out-Of-Box System) to install the Raspberry Pi OS.

Required Components

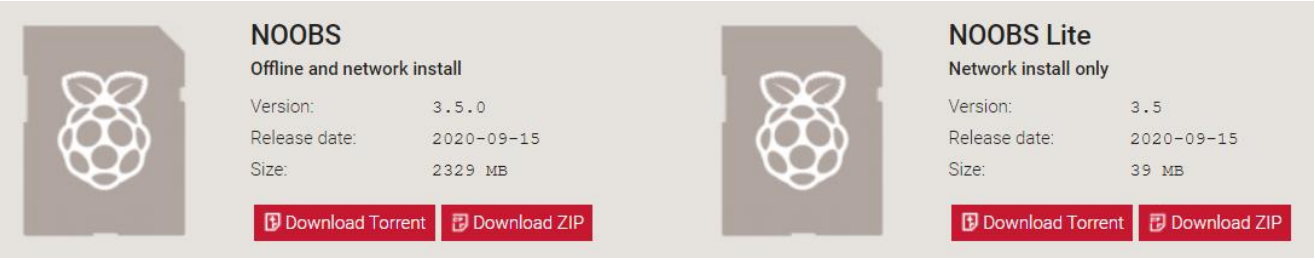
Any Raspberry Pi	1 * Power Adapter
1 * Monitor	1 * Monitor Power Adapter
1 * HDMI cable	1 * Micro-SD card
1 * Mouse	1 * Keyboard
1 * Personal Computer	

Installing NOOBS

Step 1

On the NOOBS download page, you can choose NOOBS or NOOBS LITE - the only difference is that there is a built-in offline Raspberry Pi OS installer in NOOBS, while the NOOBS LITE can only be operated with an internet connection. We suggest you use NOOBS.

Here is the download address of NOOBS: https://www.raspberrypi.org/downloads/noobs/



Step 2

Get the **SD Memory Card Formatter** software from the link below and install it. Plug in the micro-SD card reader and format the micro-SD card with the SD Card Formatter. All files will be erased, so backup important files first!

Download from here: https://www.sdcard.org/downloads/formatter/index.html

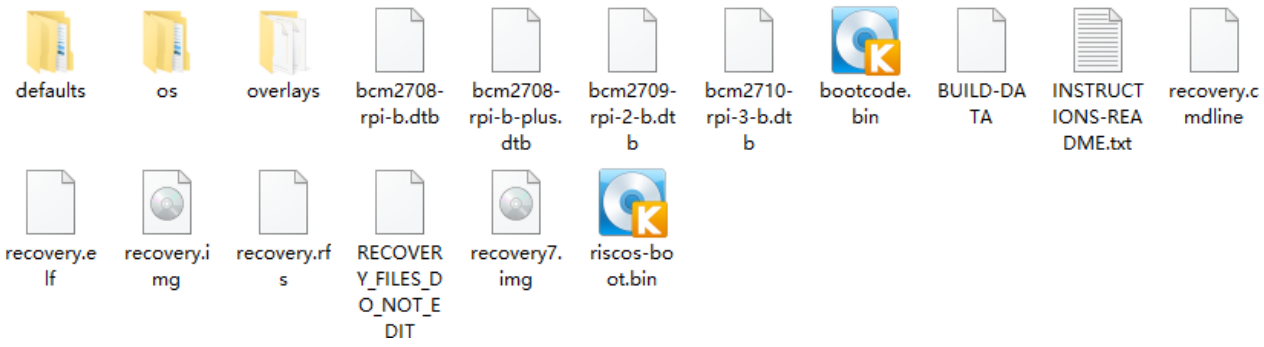
Step 3

Next, you will need to extract the files from the NOOBS zip archive you downloaded from the Raspberry Pi website.

1. Find the downloaded archive — by default, it should be in your Downloads folder.

2. Double-click on it to extract the files, and keep the resulting Explorer/Finder window open.

Finally Select all the files in the NOOBS folder and copy them to the SD card.



Step 4

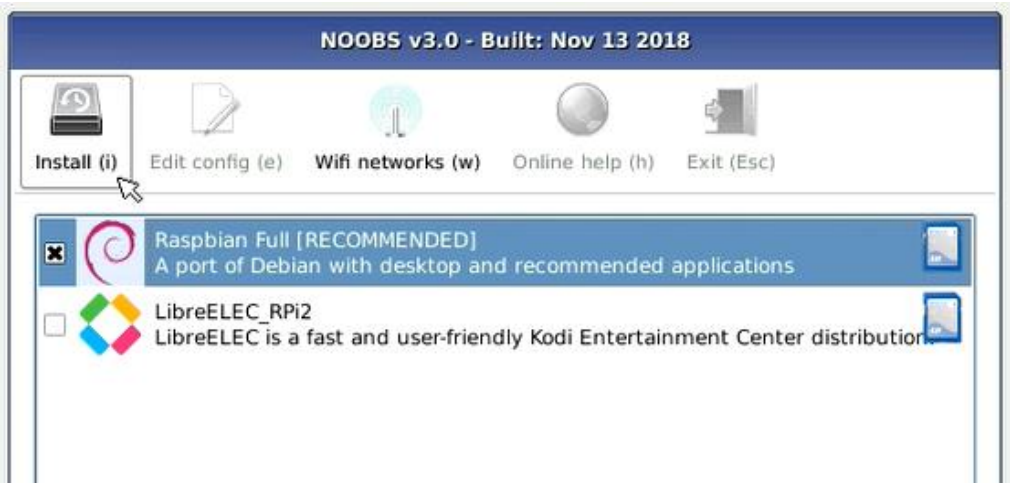
All the files transferred, the SD card pops up.

Step 5

Insert the SD card into the Raspberry Pi. In addition, connect the screen, and mouse to it. Finally power up the Raspberry Pi with a power adapter.

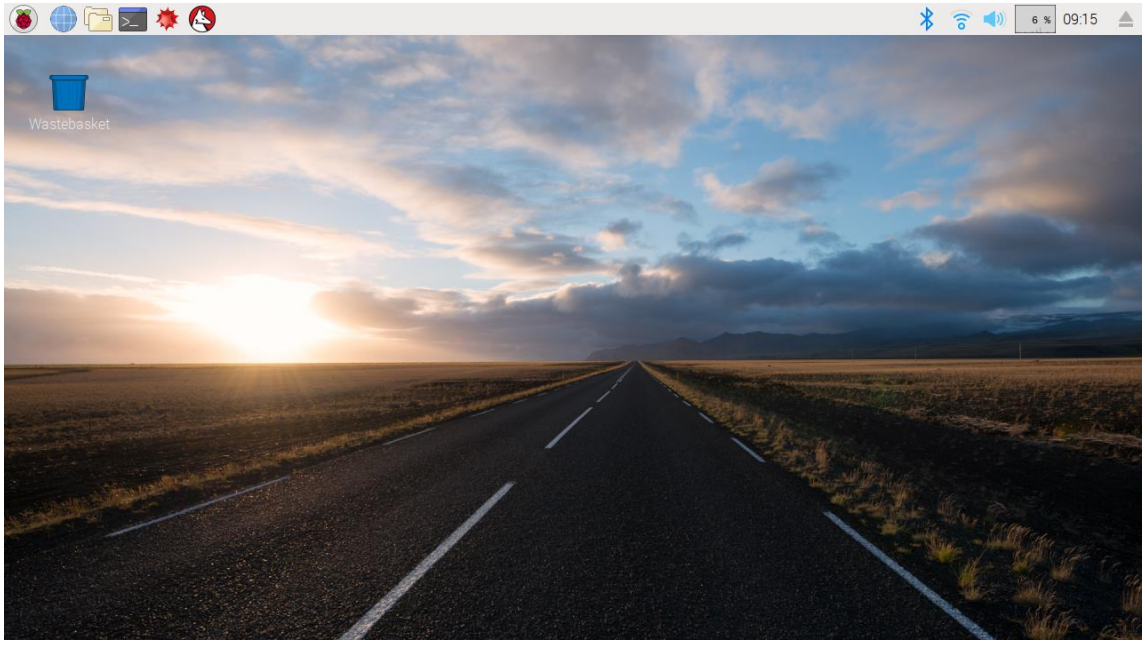
Step 6

It will go to the NOOBS interface after starting up. If you use NOOBS LITE, you need to select Wi-Fi networks (w) first. Tick the checkbox of the Raspberry Pi OS and click Install in the top left corner. The NOOBS will help to conduct the installation automatically. This process will take a few minutes.



Step 7

When the installation is done, the system will restart automatically and the desktop of the system will appear.



Step 8

If you run Raspberry Pi for the first time, the application of "Welcome to Raspberry Pi" pops up and guides you to perform the initial setup.

Step 9

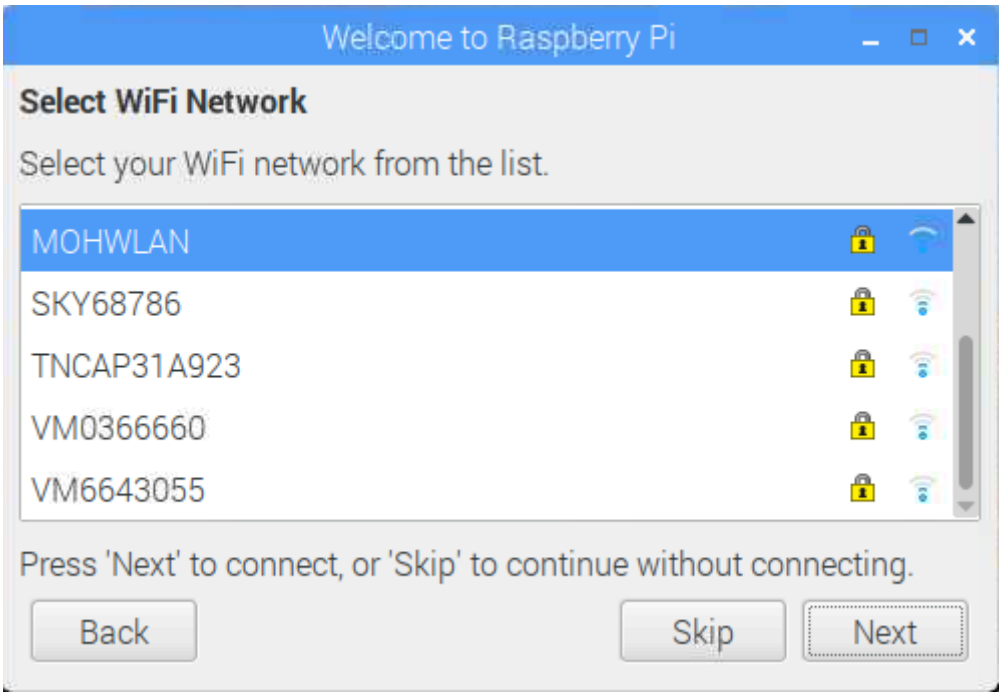
Set country/region, language and time zone, and then click "next" again.

Step 10

Input the new password of Raspberry Pi and click "Next".

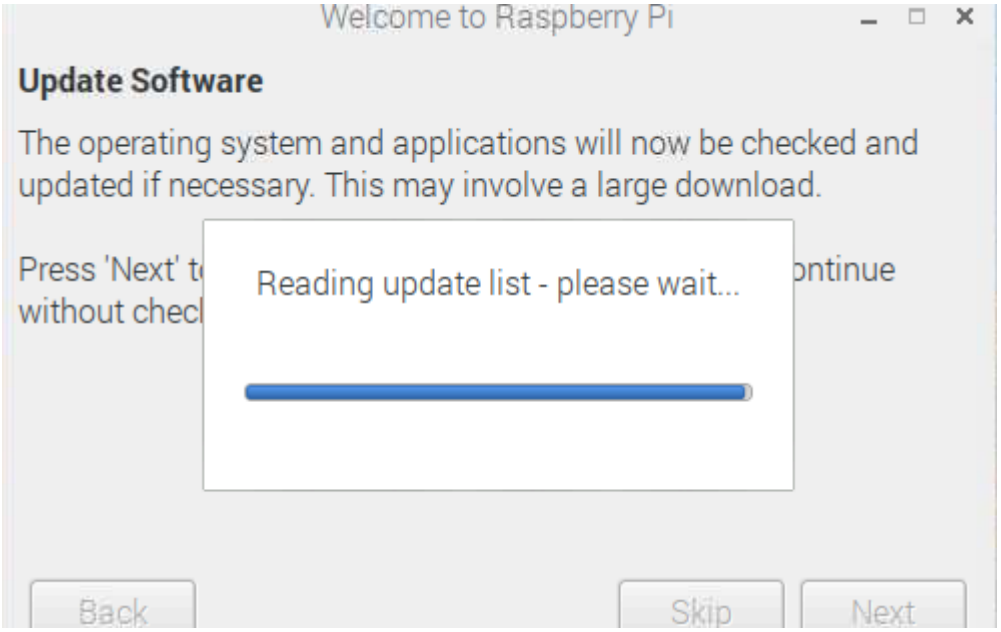
Step 11

Connect the Raspberry Pi to WI-FI and click "Next".



Step 12

Retrieve the available updates.

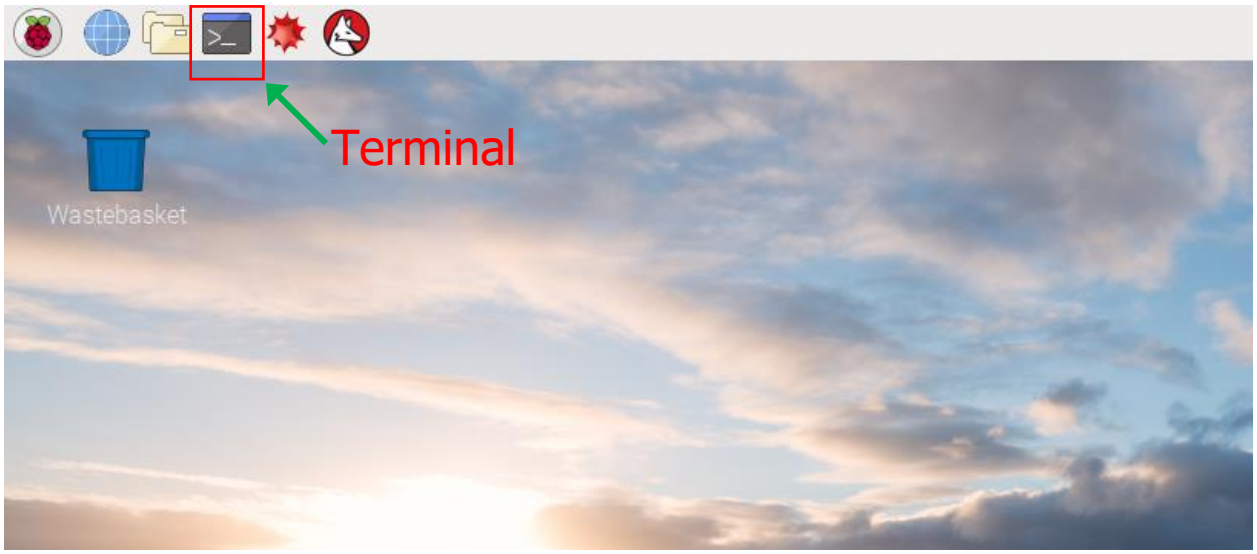


Step 13

Click "Done" to complete the Settings.

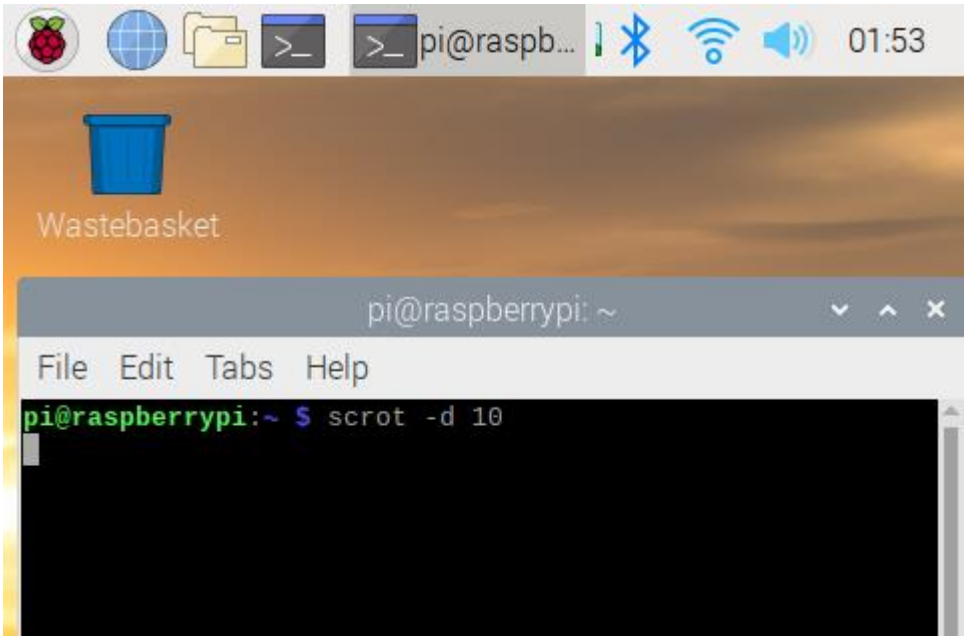
Step 14

Click the Terminal icon on the top left corner.



Step 15

Then you can input the commands on the Terminal.



Note:	You can check the complete tutorial of NOOBS on the official website of the Raspberry Pi: **https://www.raspberrypi.org/help/noobs-setup/**.

If You Have No Screen

If you don't have a screen, you can write the Raspberry Pi OS system to the Micro SD card and you can control the Raspberry Pi remotely on a PC by modifying the configuration file of the network settings in the Micro SD card.

Required Components

Any Raspberry Pi	1 * Power Adapter
1 * Micro SD card	1 * Personal computer

Installing Raspberry Pi OS

There are 2 ways to install the system:

1. Using **Raspberry Pi Imager** with an internet connection.
2. Using **Balena Etcher** with a downloaded image file.

Raspberry Pi Imager is the method recommended by the Raspberry Pi official website for beginners. With it, you can directly write the Raspberry Pi OS into micro-SD. However, each time you write to a micro-SD card, Raspberry Pi OS has to be downloaded again and this takes more time. It is also possible to use an already downloaded image file (See the alternate step 3).

With **Balena Etcher**, you need to download the Raspberry Pi OS image first, then use the tool to write it to your SD card, which can be confusing. But once you successfully finish the flashing at the first time, it only takes about 10 minutes to flash again.

Using Raspberry Pi Imager

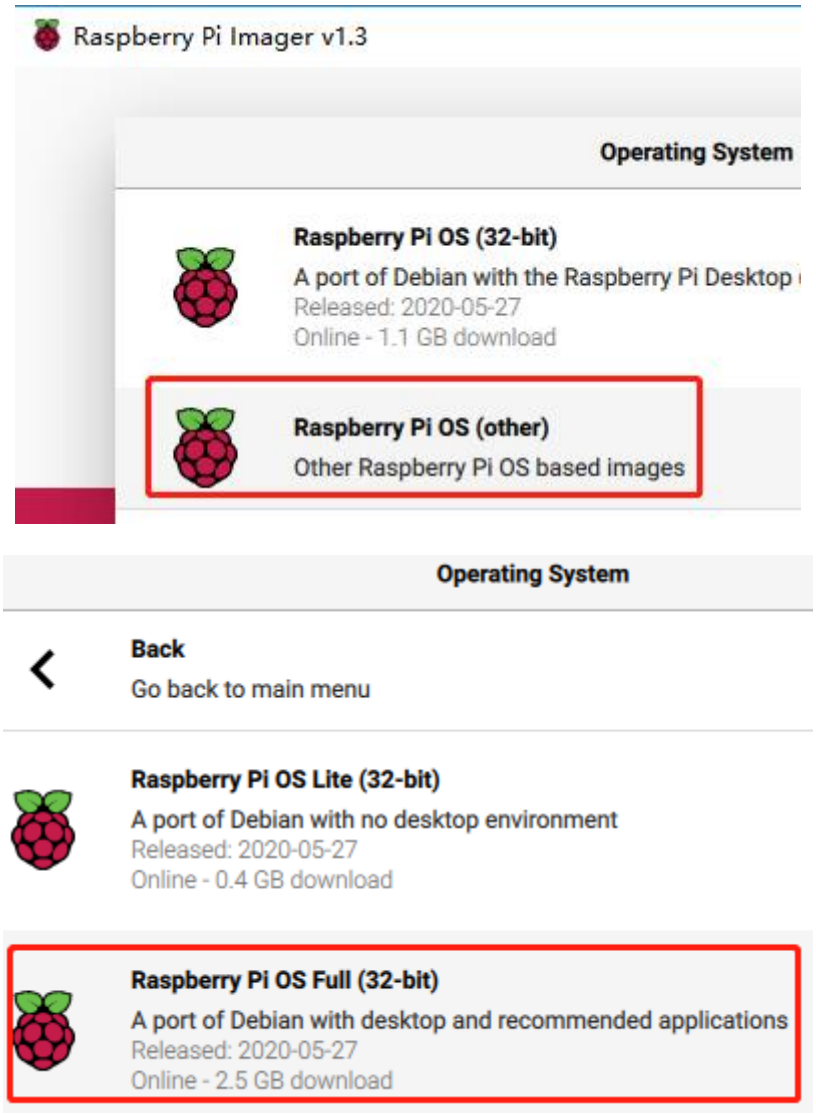
The Raspberry Pi Foundation has developed a graphical SD card writing tool that works on Mac OS, Ubuntu 18.04 and Windows, and is the easiest option for most users as it will download the image and install it automatically to the SD card.

Step 1: Download the latest version of **Raspberry Pi Imager** from https://www.raspberrypi.org/downloads/ and install it.

Step 2: Connect an SD card reader with the SD card inside.

Step 3: Open Raspberry Pi Imager and choose **Raspberry Pi OS (other) -> Raspberry Pi OS Full (32-bit)**.

Alternate Step 3: Select "Custom" and browse to the location of a downloaded image.



Step 3.1: Choose the SD card you wish to write your image to.

Step 3.2: Review your selections and click 'WRITE' to begin writing data to the SD card.

Note:	If using the Raspberry Pi Imager on Windows 10 with Controlled Folder Access enabled, you will need to explicitly allow the Raspberry Pi Imager permission to write the SD card. If this is not done, Raspberry Pi Imager will fail with a "failed to write" error.

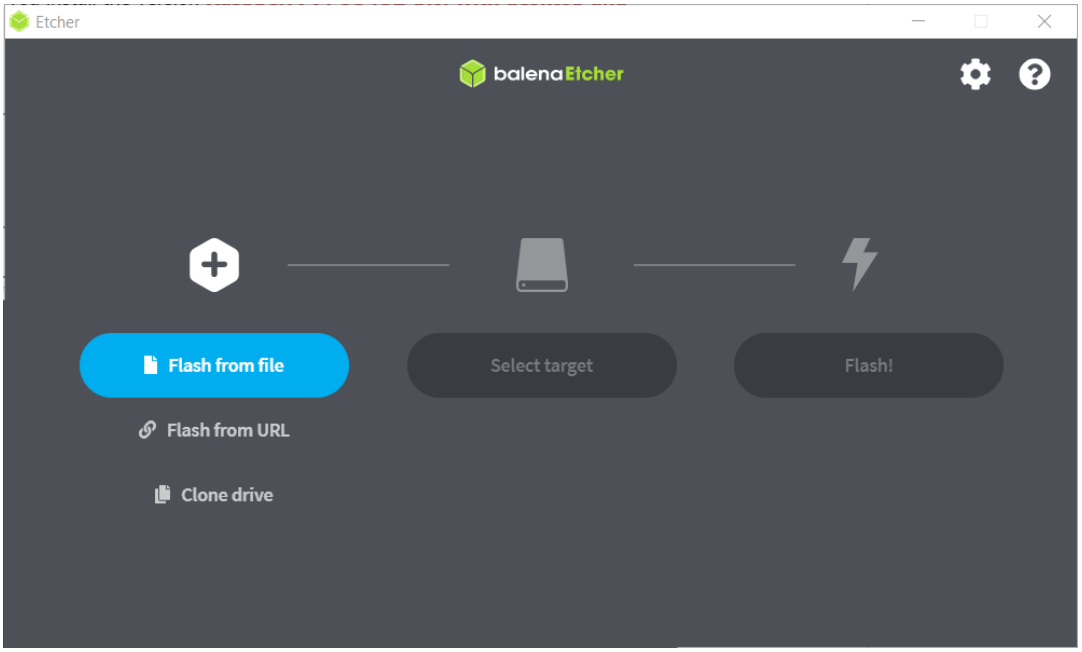
Using Balena Etcher

Step 1: Download the latest version of **Balena Etcher** from https://www.balena.io/etcher/ and install it.

Step 2: There are three different image files for Raspberry Pi OS (32-bit) available: **Lite**, **With Desktop**, and **Desktop and recommended software**. We recommend you install the version **Raspberry Pi OS (32-bit) with desktop and recommended software**.
The image can be downloaded from the official website here: https://www.raspberrypi.org/downloads/raspberry-pi-os/.

Note:	The Raspberry Pi OS with desktop image contained in the ZIP archive is over 4GB in size and uses the ZIP64 format. If you want to uncompress the archive, use a tool that supports ZIP64 like 7-Zip (Windows), The Unarchiver (Mac) and Unzip (Linux).

Step 3: Plug the USB card reader into the computer, then start Etcher.



Step 3.1: Select the image file you downloaded in Step 2.

Step 3.2: Select the SD card you want to write to. Everything will be erased!!!

Step 3.3: Click Flash. Etcher will write the image and verify it.

At this point, Raspberry Pi OS is installed. **Keep the USB card reader in your computer** if you want to configure Wi-Fi.

Connect the Raspberry Pi to the Internet

There are two methods to connect the Raspberry Pi to the network: the first one is using a **network cable**, the other way is using **WI-FI**. If you want to connect via **WI-FI, read the next section**. If you want to use a **network cable**, connect it to your Pi and **skip the next section**.

Since the Pi 3B and above, Raspberry Pi has built-in Wi-Fi. If you use an earlier version of Raspberry Pi, a USB WI-FI Adapter is needed. See https://elinux.org/RPi_USB_Wi-Fi_Adapters for more information.

If you want to connect to your network using WI-FI, you will need to create or modify the WI-FI configuration file wpa_supplicant.conf located on the SD card. A text editor that uses "Linux" line endings, not "DOS/Windows" line endings will be needed. One such editor for Windows is Notepad++.

The drive where the file needs to be should be named "boot".

Write the following in **wpa_supplicant.conf**, replacing the "<…>" with the proper values:

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=<COUNTRY>

network={
 ssid="<SSID>"
 psk="<PASSWORD>"
}
```

For proper Wi-Fi operation, **<COUNTRY>** needs to be set to the two-letter ISO/IEC alpha2 code for the country in which you are using your Raspberry Pi. This is **CA** for Canada and **US** for United-States. For other countries, please refer to the following link:

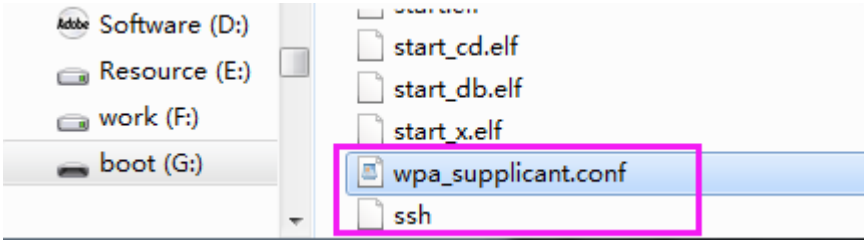
https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2#Officially_assigned_code_elements

You need to replace <**SSID**> with the name of your Wi-Fi network and <**PASSWORD**> with your Wi-Fi password.

Raspberry Pi OS will use this file to configure WI-FI when it runs next time.

Enable SSH

To remote control Raspberry Pi, you need to enable SSH. SSH uses a reliable, secure protocol for remote login sessions and other network services. Generally, SSH is disabled in new Raspberry Pi OS installations. To enable it, you need to create a file named SSH in the drive named "boot".

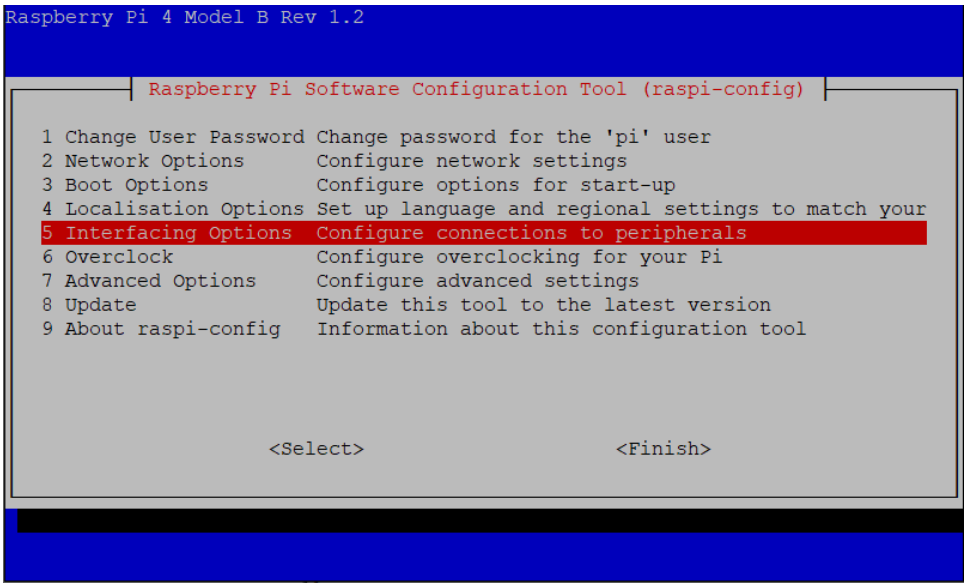


With the ssh file present, Raspberry Pi OS will enable SSH access after it starts.

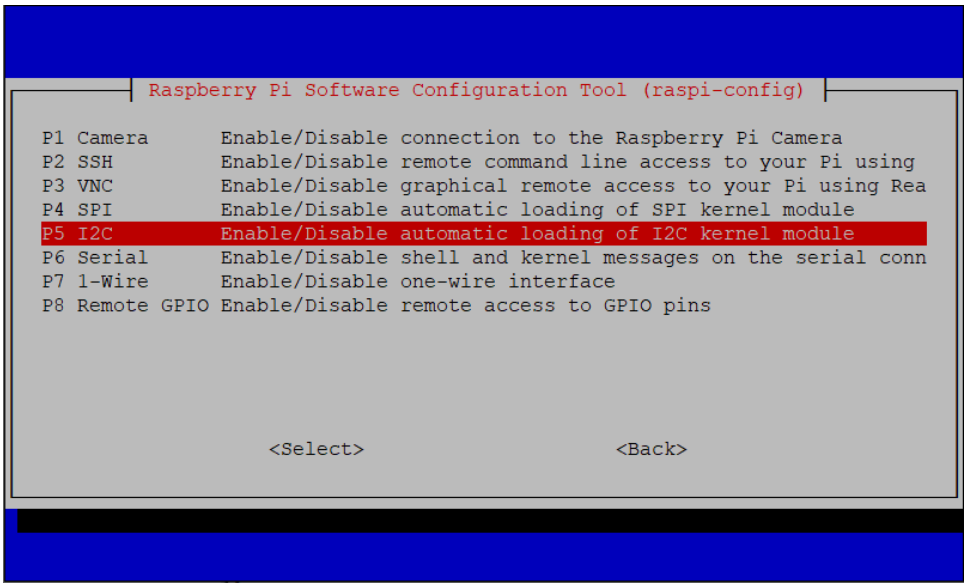
Appendix: I2C Configuration

Step 1: Enable the I2C port of your Raspberry Pi (If not already enabled, If enabled, go to the next step):

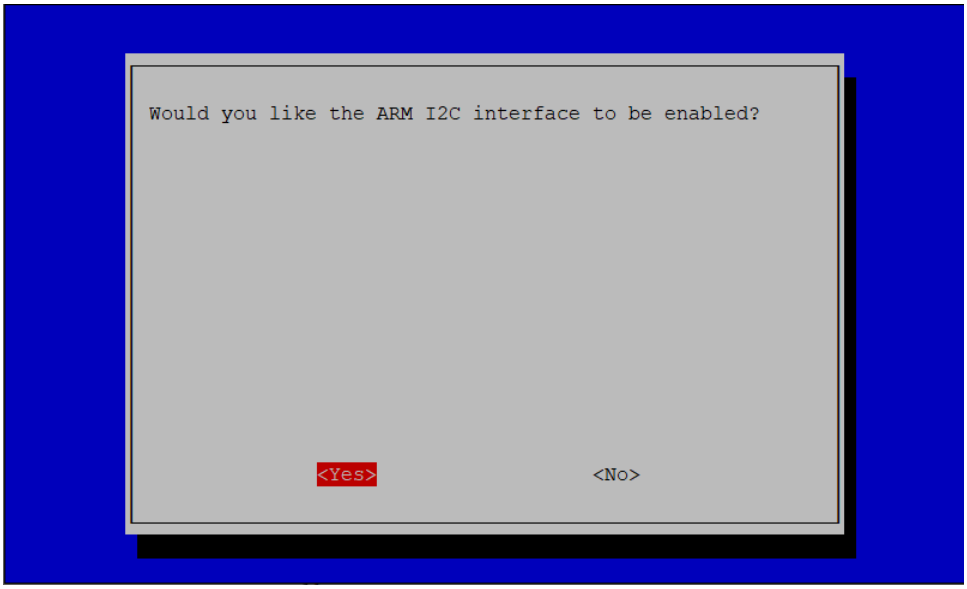
```
sudo raspi-config
```



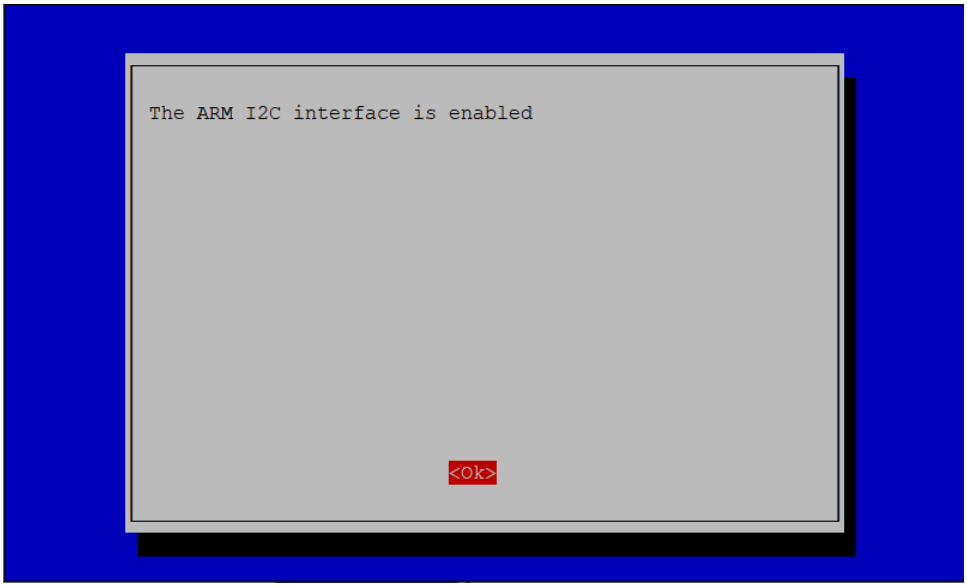
Select option **5 Interfacing options**



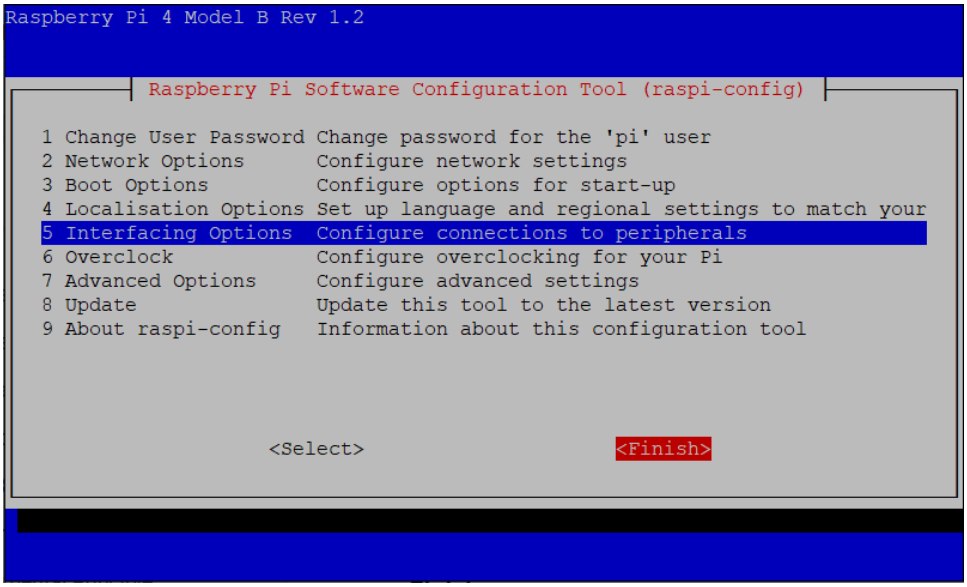
Select option **P5 I2C**


Would you like the ARM I2C interface to be enabled?
<Yes>
<No>

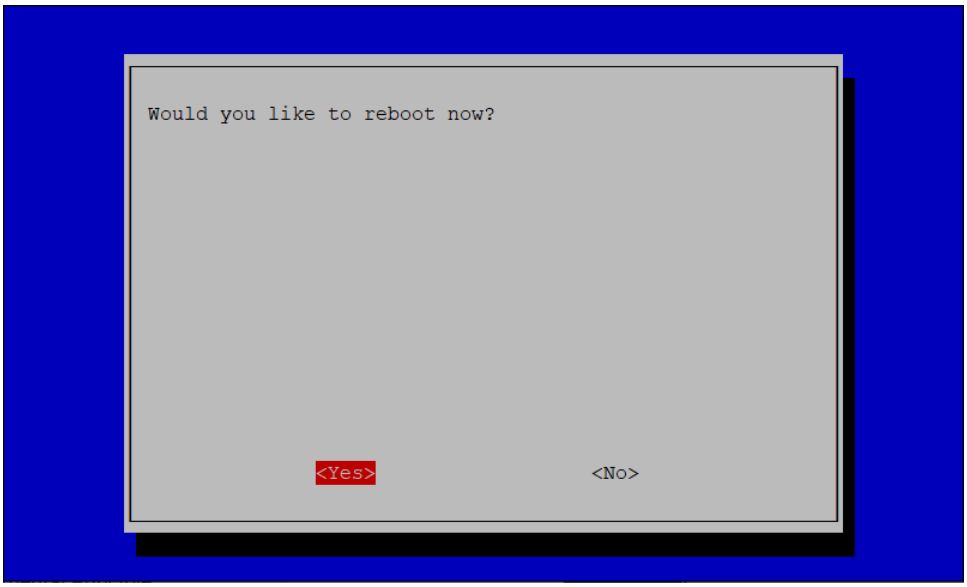
<Yes>


The ARM I2C interface is enabled
<Ok>

<Ok>



Select <Finish>



<Yes> (If you do not see this page, continue to the next step)

Step 2: Install i2c-tools.

```
sudo apt install i2c-tools
```

Step 3: Check the address of the I2C device:

i2cdetect -y 1 # For Raspberry Pi 2 and higher version

i2cdetect -y 0 # For Raspberry Pi 1

```
pi@raspberrypi ~ $ i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- 48 -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

If there's a I2C device connected, the results will be similar to the screenshot above. In this example, the address of the device is 0x48, 48 is printed.

Step 4:

For C language users

Install libi2c-dev.

```
sudo apt install libi2c-dev
```

For Python users:

Install smbus for I2C.

```
sudo apt install python3-smbus
```

Copyright

HiPi Industries Inc. (HiPi.io) Copyright Notice