

# Raspberry Pi GPIO Library Introduction

Currently, there are two major GPIO libraries for Raspberry Pi, RPi.GPIO and wiringPi.

## ***RPi.GPIO:***

RPi.GPIO is a python module to control Raspberry Pi GPIO channels. For more information about RPi.GPIO, please visit:

<https://pypi.python.org/pypi/RPi.GPIO/>

For examples and documentation, please visit:

<http://sourceforge.net/p/raspberry-gpio-python/wiki/Home/>

The RPi.GPIO module is pre-installed in the official Raspbian operating system, you can use it directly.

## ***wiringPi:***

The wiringPi is a GPIO access library written in C for the BCM2835/6 SOC used in the Raspberry Pi. It's released under the GNU LGPLv3 license and is usable from C and C++ and many other languages with suitable wrappers. It's designed to be familiar to people who have used the Arduino "wiring" system.

For more information about wiringPi, please visit : <http://wiringpi.com/>

## ***Install wiringPi:***

Step 1 : Get the source code

```
$ git clone git://git.drogon.net/wiringPi
```

Step 2 : Compile and install

```
$ cd wiringPi
```

```
$ git pull origin
```

```
$ sudo ./build
```

Press Enter, the script "build" will automatically compile wiringPi source code and then install it to the Raspberry Pi.

Next, verify whether the wiringPi is installed successfully or not:

**wiringPi** includes a command-line utility `gpio` which can be used to program and setup the GPIO pins. You can use this to read and write the pins and even use it to control them from shell scripts.

You can verify whether the wiringPi is installed successfully or not by the following commands:

```
$ sudo gpio -v
```

```
pi@raspberrypi ~ $ sudo gpio -v
gpio version: 2.26
Copyright (c) 2012-2015 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
  Type: Model 2, Revision: 1.1, Memory: 1024MB, Maker: Sony
pi@raspberrypi ~ $
```

```
$ sudo gpio readall
```

```
pi@raspberrypi ~ $ sudo gpio readall
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi |   Name   | Mode | V | Physical | V | Mode |   Name   | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2   | 8   | SDA.1    | ALT0 | 1 | 3        | 4 |      | 5v       |     |     |
| 3   | 9   | SCL.1    | ALT0 | 1 | 5        | 6 |      | 0v       |     |     |
| 4   | 7   | GPIO. 7  | IN   | 1 | 7        | 8 | 1 ALT0 | TxD      | 15  | 14  |
|     |     | 0v       |      |   | 9        | 10| 1 ALT0 | 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 | 0 | 19       | 20|      | 0v       |     |     |
| 9   | 13  | MISO     | ALT0 | 0 | 21       | 22| 0 IN   | GPIO. 6  | 6   | 25  |
| 11  | 14  | SCLK     | ALT0 | 0 | 23       | 24| 1 ALT0 | CE0      | 10  | 8   |
|     |     | 0v       |      |   | 25       | 26| 1 ALT0 | CE1      | 11  | 7   |
| 0   | 30  | SDA.0    | IN   | 1 | 27       | 28| 1 IN   | SCL.0    | 31  | 1   |
| 5   | 21  | GPIO.21  | IN   | 1 | 29       | 30|      | 0v       |     |     |
| 6   | 22  | GPIO.22  | IN   | 1 | 31       | 32| 0 IN   | GPIO.26  | 26  | 12  |
| 13  | 23  | GPIO.23  | IN   | 0 | 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@raspberrypi ~ $
```

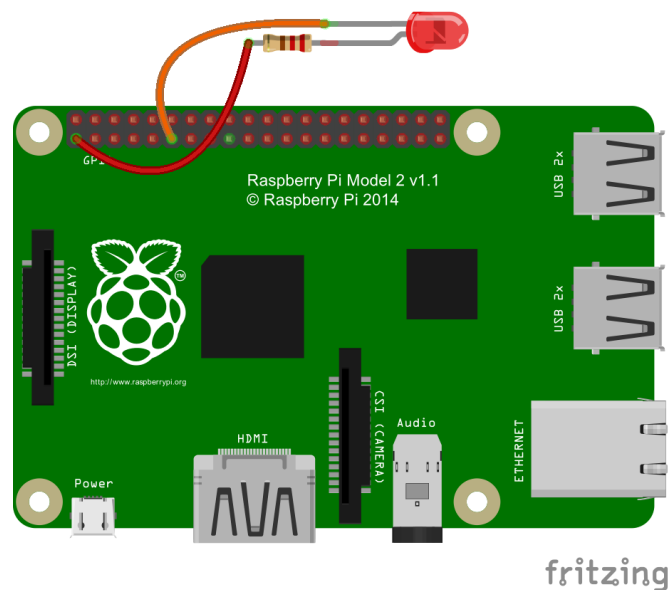
If you can see the information shown above, it indicates that the wiringPi has been installed successfully.

# How to use the wiringPi and the RPi.GPIO

Here we take a blinking LED for example to illustrate how to use the wiringPi C library and the RPi.GPIO Python module.

Step 1 : Build the circuit according to the following schematic diagram

Note : Resistance=220 $\Omega$



*For Python user:*

Step 2 : Create a file named led.py

```
$ sudo touch led.py
```

```
pi@raspberrypi /home $ ls
pi
pi@raspberrypi /home $ sudo touch led.py
pi@raspberrypi /home $ ls
led.py pi
pi@raspberrypi /home $
```

Step 3 : Open the file led.py with vim or nano

```
$ sudo vim led.py
```

Write down the following source code, then save and exit.

```
#!/usr/bin/env python
import RPi.GPIO as GPIO
import time

Led = 11      # pin11

def setup():
    GPIO.setmode(GPIO.BOARD)      # Numbering according to the physical location
    GPIO.setup(Led, GPIO.OUT)     # Set pin mode as output
    GPIO.output(Led, GPIO.HIGH)   # Output high level(+3.3V) to off the led

def loop():
    while True:
        print '...led on'
        GPIO.output(Led, GPIO.LOW) # led on
        time.sleep(0.5)
        print 'led off...'
        GPIO.output(Led, GPIO.HIGH) # led off
        time.sleep(0.5)

def destroy():
    GPIO.output(Led, GPIO.HIGH)     # led off
    GPIO.cleanup()                 # Release resource

if __name__ == '__main__':        # Program start from here
    setup()
    try:
        loop()
    except KeyboardInterrupt:     # Press 'Ctrl+C' to end the program
        destroy()
```

Step 4 : Run the program

```
$ sudo python led.py
```

```
pi@raspberrypi /home $ ls
led.py pi
pi@raspberrypi /home $ sudo python led.py
...led on
led off...
...led on
led off...
...led on
led off...
```

Press Enter, you should see that the LED is blinking. Press 'Ctrl+C', the program execution will be terminated.

### ***For C language user:***

Step 2 : Create a file named led.c

```
$ sudo touch led.c
```

```
pi@raspberrypi /home $ ls
led.py pi
pi@raspberrypi /home $ sudo touch led.c
pi@raspberrypi /home $
```

Step 3 : Open the file led.c with vim or nano

```
$ sudo vim led.c
```

Write down the following source code, then save and exit.

```
#include <wiringPi.h>
#include <stdio.h>

#define Led 0 //wiringPi GPIO0, pin11

int main(void)
{
    if(wiringPiSetup() == -1){
        printf("setup wiringPi failed !\n");
        return -1;
    }

    pinMode(Led, OUTPUT);

    while(1){
        digitalWrite(Led, LOW); //led on
        printf("led on...\n");
        delay(500);
        digitalWrite(Led, HIGH); //led off
        printf("...led off\n");
        delay(500);
    }

    return 0;
}
```

Step 4 : Compile the code

```
$ sudo gcc led.c -lwiringPi
```

```
pi@raspberrypi /home $ ls
led.c led.py pi
pi@raspberrypi /home $ sudo gcc led.c -lwiringPi
pi@raspberrypi /home $
```

After executing this command, you'll find a file named a.out appear in the current directory. It is an executable program.

```
pi@raspberrypi /home $ ls
a.out led.c led.py pi
pi@raspberrypi /home $
```

Step 5 : Run the program

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

```
pi@raspberrypi /home $ sudo ./a.out
led on...
...led off
led on...
...led off
```

Press Enter, you should see that the LED is blinking. Press 'Ctrl+C', the program execution will be terminated.

**Resources :**

<http://sourceforge.net/p/raspberry-gpio-python/wiki/Examples/>

<http://wiringpi.com/reference/>

**NOTE:**

Before learning the next courses, please copy the source code we provided to your Raspberry Pi's /home/ directory, or you can get the source code directly from our github repository:

**C Language Source Code :**

\$ git clone [https://github.com/adeept/Adeept\\_Ultimate\\_Starter\\_Kit\\_C\\_Code\\_for\\_RPi.git](https://github.com/adeept/Adeept_Ultimate_Starter_Kit_C_Code_for_RPi.git)

**Python Source Code :**

\$ git clone [https://github.com/adeept/Adeept\\_Ultimate\\_Starter\\_Kit\\_Python\\_Code\\_for\\_RPi.git](https://github.com/adeept/Adeept_Ultimate_Starter_Kit_Python_Code_for_RPi.git)