THE HONG KONG POLYTECHNIC UNIVERSITY

Department of Electronic and Information Engineering

EIE568 IoT – Tools and Applications

Laboratory 3b

Objectives:

(i) To write programs to control the peripheral equipment.

(ii) To design and implement an IoT system.

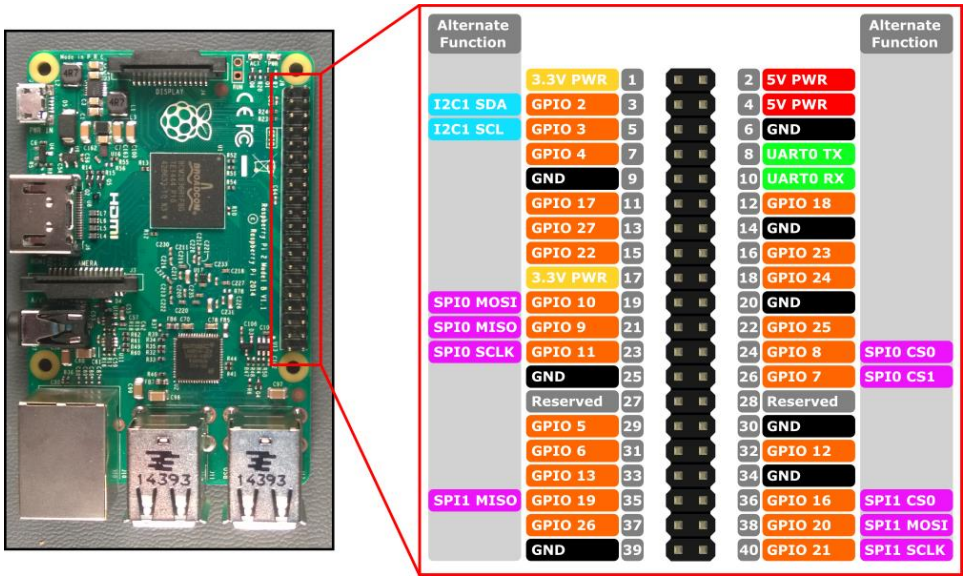
Introduction:

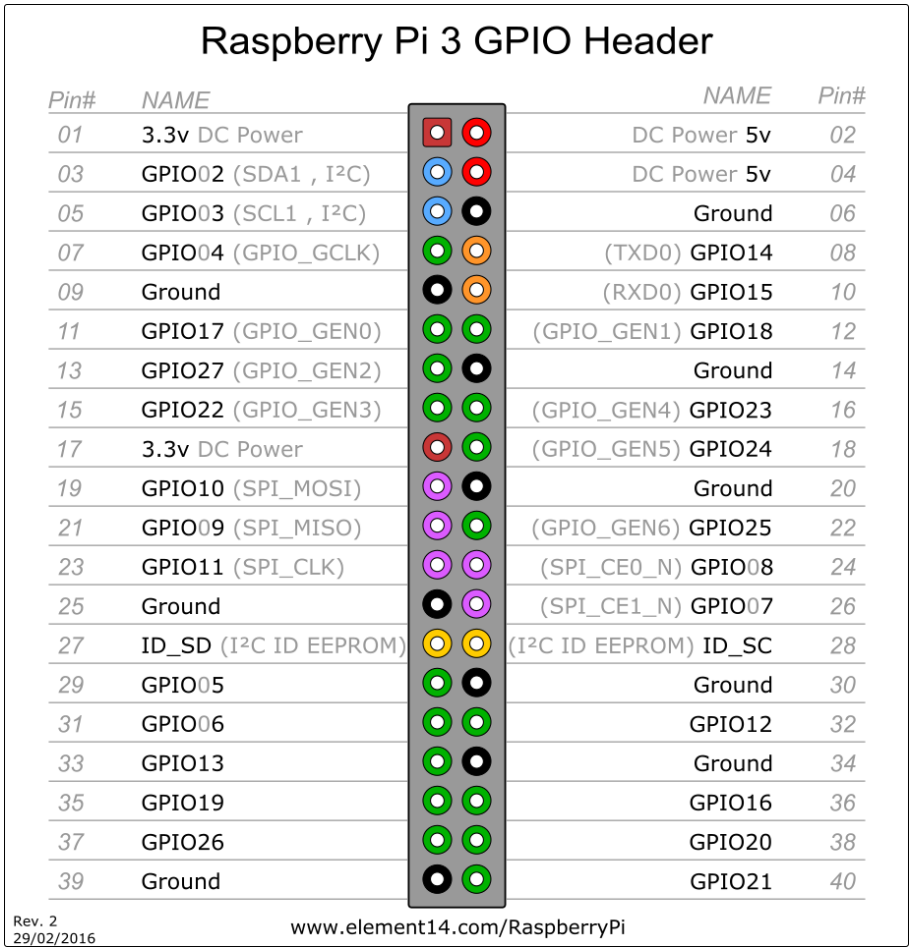
This lab exercise provides an introduction for the Raspberry Pi and peripheral equipment. Students are asked to learn through firsthand experience on using Raspberry Pi to control the peripheral equipment and communicate with mobile terminal user.

Task 3. Pi GPIO Pins controller

This task is to be familiar with the Pi GPIO Pins. After connecting the peripheral equipment via GPIO pins, we can write scripts to control it.

Controlling Pi3 GPIO Pins





Parts list:

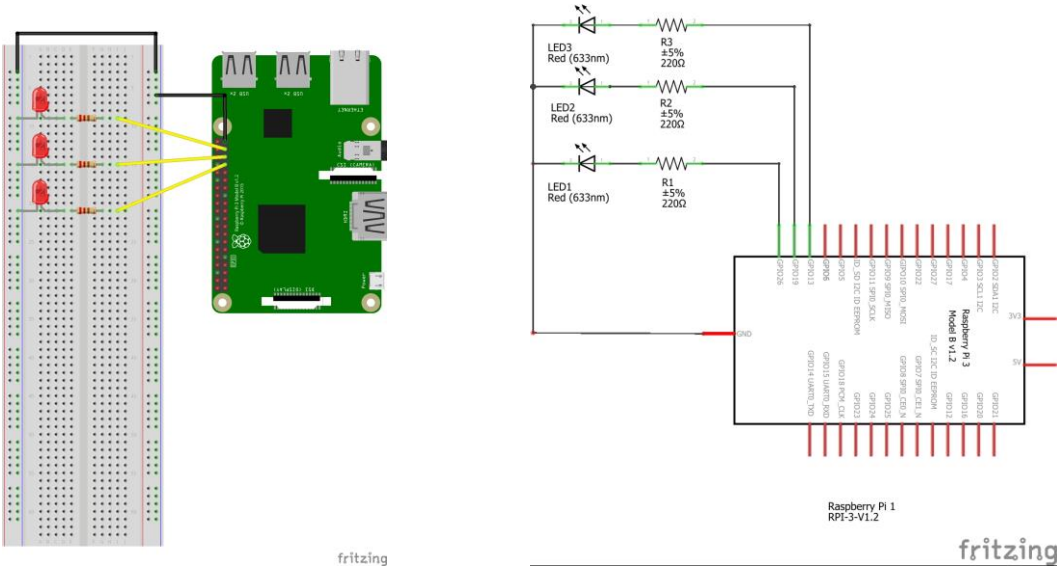
Raspberry Pi X 1

LED X 3

Resistor (220 ohm) X 3

The following table helps you determine the pin number and GPIO number for the connections of three LEDs. The long pin of the LED is positive (anode), and the short one is negative (cathode).

Parts	**Pin Number**	**GPIO Number**
LED1	Pin 37	GPIO 26
LED2	Pin 35	GPIO 19
LED3	Pin 33	GPIO 13



Once the hardware is ready, we can proceed with the Python program. In this program we have to read the data (message) sent from the Telegram bot and toggle the LED accordingly. The program will primarily check for two words, they are ‘on’ and ‘off’**.** Once detecting either one of these two words, it will look for other keywords like ‘led1’, ‘led2’ and ‘led3’. The respective led will be toggled only if the word is detected. We will also update a string for the detected words to send a message back to telegram bot.

For each message we send from the mobile, there should be a chat id and text. This chat id is required by the program to reply back to the sender. We save the chat id and message as below.

```
chat_id = msg[‘chat’][‘id’]
text = msg[‘text’]
```

Create a file “Led_test.py” and edit it.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)

led1 = 26
led2 = 19
led3 = 13
GPIO.setwarnings(False)
GPIO.setup(led1, GPIO.OUT)
GPIO.output(led1, 0)
GPIO.setup(led2, GPIO.OUT)
GPIO.output(led2, 0)
GPIO.setup(led3, GPIO.OUT)
GPIO.output(led3, 0)

while True:
    GPIO.output(led1,1)
    time.sleep(1)
    GPIO.output(led1,0)
    time.sleep(1)
    GPIO.output(led2,1)
    time.sleep(1)
    GPIO.output(led2,0)
    time.sleep(1)
    GPIO.output(led3,1)
    time.sleep(1)
    GPIO.output(led3,0)
    time.sleep(1)
```

Run the script and observe the LED.

Now let’s control the LED via our mobile phone. Create a file “3led.py” and edit it.

```
import time, datetime
import RPi.GPIO as GPIO
import telepot
from telepot.loop import MessageLoop
led1 = 26
led2 = 19
led3 = 13
now = datetime.datetime.now()
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

GPIO.setup(led1, GPIO.OUT)
GPIO.output(led1, 0)
GPIO.setup(led2, GPIO.OUT)
GPIO.output(led2, 0)
GPIO.setup(led3, GPIO.OUT)
GPIO.output(led3, 0)

def action(msg):
        chat_id = msg['chat']['id']
        text = msg['text']
        print ('Received: %s' % text)
        if 'on' in text:
                message = "Turned on "
                if 'led1' in text:
                        message = message + "led1 "
                        GPIO.output(led1, 1)
                if 'led2' in text:
                        message = message + "led2 "
                        GPIO.output(led2, 1)
                if 'led3' in text:
                        message = message + "led3 "
                        GPIO.output(led3, 1)
                if 'all' in text:
                        message = message + "all "
                        GPIO.output(led1, 1)
```

```
            GPIO.output(led2, 1)
            GPIO.output(led3, 1)
        message = message + "light(s)"
        telegram_bot.sendMessage (chat_id, message)
if 'On' in text:
        message = "Turned on "
        if 'led1' in text:
            message = message + "led1 "
            GPIO.output(led1, 1)
        if 'led2' in text:
            message = message + "led2 "
            GPIO.output(led2, 1)
        if 'led3' in text:
            message = message + "led3 "
            GPIO.output(led3, 1)
        if 'all' in text:
            message = message + "all "
            GPIO.output(led1, 1)
            GPIO.output(led2, 1)
            GPIO.output(led3, 1)
        message = message + "light(s)"
        telegram_bot.sendMessage (chat_id, message)

if 'off' in text:
        message = "Turned off "
        if 'led1' in text:
            message = message + "led1 "
            GPIO.output(led1, 0)
        if 'led2' in text:
            message = message + "led2 "
            GPIO.output(led2, 0)
        if 'led3' in text:
            message = message + "led3 "
            GPIO.output(led3, 0)
        if 'all' in text:
            message = message + "all "
            GPIO.output(led1, 0)
            GPIO.output(led2, 0)
```

```
                GPIO.output(led3, 0)
            message = message + "light(s)"
            telegram_bot.sendMessage (chat_id, message)
        if 'Off' in text:
            message = "Turned off "
            if 'led1' in text:
                message = message + "led1 "
                GPIO.output(led1, 0)
            if 'led2' in text:
                message = message + "led2 "
                GPIO.output(led2, 0)
            if 'led3' in text:
                message = message + "led3 "
                GPIO.output(led3, 0)
            if 'all' in text:
                message = message + "all "
                GPIO.output(led1, 0)
                GPIO.output(led2, 0)
                GPIO.output(led3, 0)
            message = message + "light(s)"
            telegram_bot.sendMessage (chat_id, message)

telegram_bot = telepot.Bot('527769403:AAF-DECm9ky_BoPSeUxIzJLis3TXUdSXXXX')
print (telegram_bot.getMe())
MessageLoop(telegram_bot, action).run_as_thread()
print ('Up and Running....')
while 1:
    time.sleep(10)
```

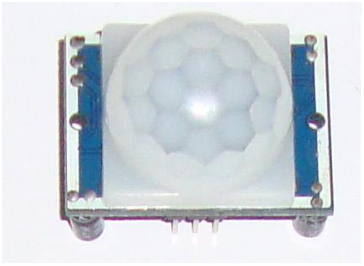
Run the script and control the LED with mobile phone.

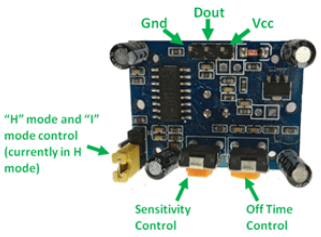
Task 4. Interfacing the PIR Motion Sensor to the Raspberry Pi

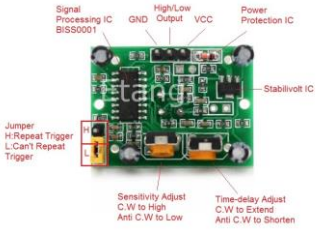
PIR (passive infrared) motion sensor detects any movement of objects, human or animals. Mostly they are used in automatically activated lighting and burglar alarm systems.

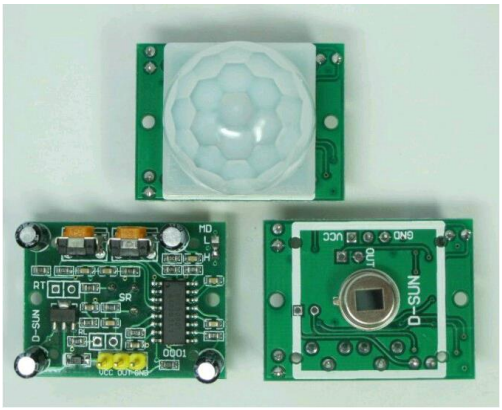
Every object with temperature above absolute zero emits heat in form of infrared radiation. PIR motion sensor detects change in the infrared radiation impinging on it. When any object or human passes in the front of the PIR sensor, the temperature in the sensor's field of view will rise from ambient temperature to object temperature and then back again to ambient temperature. PIR sensor converts this temperature change to change in output voltage and hence the motion is detected.

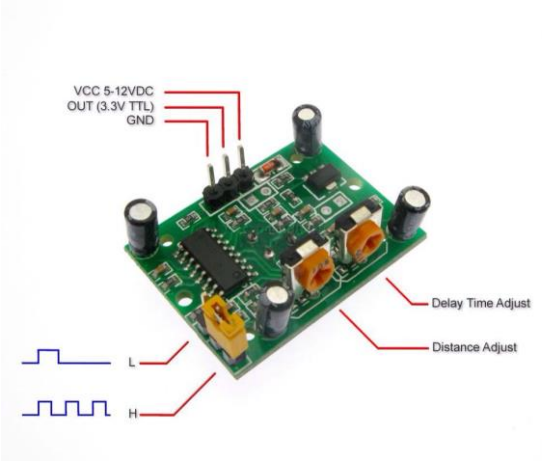
HC-SR501 PIR Sensor









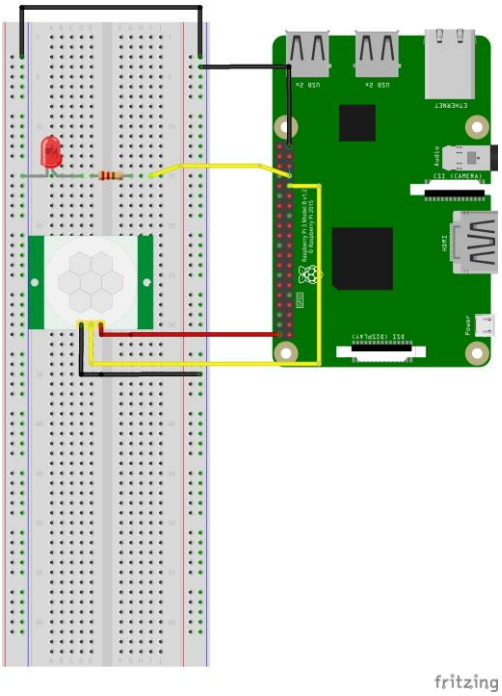


Pin configuration

The Sensor has three pins. Power (VCC), Ground (GND) and the output (OUT) pin which gives logic high if motion is detected.

Most PIR sensors have adjustable screws on them for adjusting both the time and sensitivity. The time will allow you to set a delay before it goes off after sending a high signal. The sensitivity defines the maximum distance that motion can be detected.

Connect your Raspberry Pi GPIO to the PIR motion sensor as shown below:



Parts list:

Raspberry Pi	X 1
LED	X 1
Resistor (220 ohm)	X 1
HC-SR501 PIR Sensor	X1

Connect the LED to GPIO13

Connect the PIR pin2(Data output) to GPIO6

To control the PIR sensor, create a file “pirtest.py” and edit it.

```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(6, GPIO.IN)
GPIO.setup(13, GPIO.OUT)
while True:
        i=GPIO.input(6)
        if i==0:
                print('Motion ended',i)
                GPIO.output(13, 0)    #Turn OFF LED
```

```
        time.sleep(1)
    elif i==1:
        print('Motion detected',i)
        GPIO.output(13, 1)    #Turn ON LED
        time.sleep(1)
```

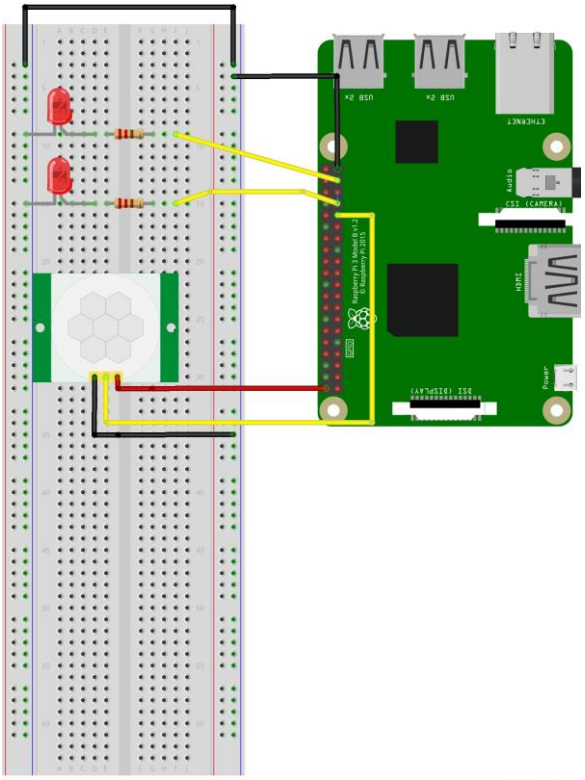
Run the script and test the PIR sensor.

Exercise 2

To enable communication with the mobile phone so that remote users can receive warning when motion is detected, create a file “pir_bot_test.py” and edit it.
Run the script and observe the result. Demonstrate your result to the tutor.

Exercise 3

To enable motion detection, communication and LED on/off functions simultaneously, connect the Raspberry Pi and peripheral equipment as follows:



Parts list:

Raspberry Pi	X 1
LED	X 2
Resistor (220 ohm)	X 2
HC-SR501 PIR Sensor	X1

Connect LED1 to GPIO13
Connect LED2 to GPIO26
Connect PIR pin2(Data output) to GPIO6

Create a file “pir_bot_led_test.py” and edit it so that one LED will indicate the motion detection while the other LED will indicate the communications.
Run the script and observe the result. Demonstrate your result to the tutor.

Some more information on sensor devices:

Picamera (Ref: http://picamera.readthedocs.io/)

1. Open up your Raspberry Pi Camera module. Be aware that the camera can be damaged by static electricity. Before removing the camera from its grey anti-static bag, make sure you have discharged yourself by touching an earthed object (e.g., a radiator or PC Chassis).

2. Install the Raspberry Pi Camera module by inserting the cable into the Raspberry Pi. The cable slots into the connector situated between the Ethernet and HDMI ports, with the silver connectors facing the HDMI port.

3. Boot up (power on) the Raspberry Pi.

4. From the prompt, run "***sudo raspi-config***". If the "camera" option is not listed,

you will need to run a few commands to update your Raspberry Pi. Run "***sudo apt-get update***" and "***sudo apt-get upgrade***".

5. Run "***sudo raspi-config***" again - you should be able to see the "camera" option.

6. Navigate to the "camera" option and enable it. Select “Finish” and reboot your Raspberry Pi.

Buzzer

The long pin of the buzzer is positive, and the short one is negative. This is also marked on the bottom of the buzzer. The positive pin of the buzzer is wired to digital IO (GPIO) pin of Raspberry Pi, and the negative pin is connected to GND of the Raspberry.

Mini Project

After going through all the lab materials (Labs 1 to 3), you should have acquired basic knowledge on controlling various sensor devices, and handling the communications from sensors to gateway, and then the Internet. You can analyze the collected sensor data through machine learning, and respond to them through mobile bot program. Apply what you have learnt to propose a novel IoT application in the mini project.