

INTERNAL LAB-2

AIM: Smart lighting system if person moving in the offices, residences between 11 pm and 5 am to alert the security people

Description:

In this experiment we use PIR sensor for detecting a person moving from one room to another room and on led when a person is in the room and off the led when there is no motion in the room.

PIR sensor is used for detecting infrared heat radiations. This makes them useful in the detection of moving living objects that emit infrared heat radiations.

The output of PIR sensor is on when it senses motion; whereas it is off when there is no motion

PIR sensors are used in many applications like for room light control using human detection, human motion detection for security purpose at home, etc.

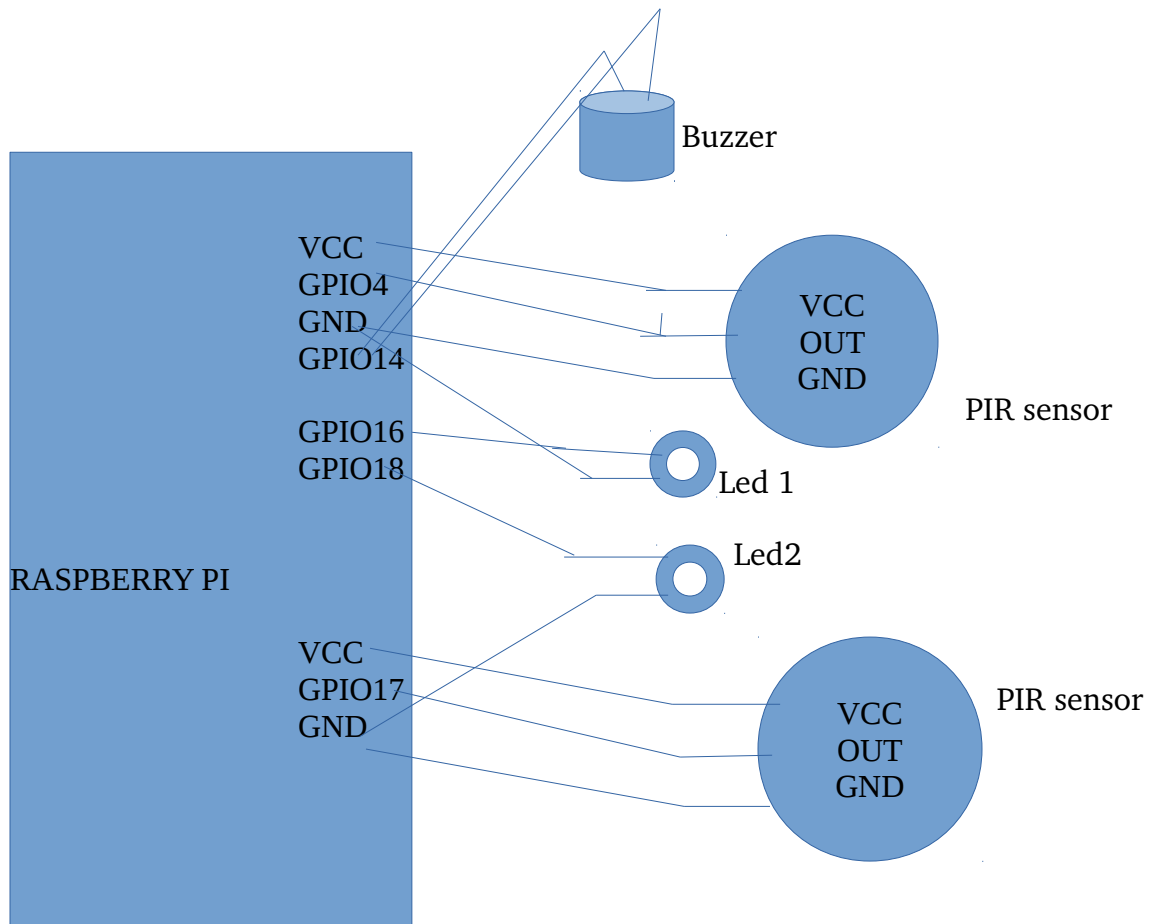
Whenever a motion is detected the buzzers give a beep sound and get off when there is no motion in the room.

Here, LED is connected to GPIO16 (pin no. 23) and GPIO18 (pin no. 24) whereas PIR output is connected to GPIO5.

Hardware Requirements:

- PIR sensor
- connecting wires
- buzzer
- led
- raspberry pi

Circuit Diagram:



CODE:

```
import RPi.GPIO as gp
from time import gmtime, strftime

#from firebase import firebase

gp.setmode(gp.BOARD)
gp.setwarnings(False)
room=[8,11]
led=[16,18]
buzz=37
for i in range(2):
    gp.setup(room[i],gp.IN)
for i in range(2):
```

```

    gp.setup(led[i], gp.OUT)
gp.setup(buzz, gp.OUT)
#for i in range(2):
#    gp.setup(buzz[i], gp.OUT)

if __name__=='__main__':

    while(True):
#        hr=strftime('%H:%M:%S', gmtime).split(':')[0]
        if(gp.input(room[0])):
            gp.output(led[0], True)
            gp.output(buzz, gp.HIGH)
#            time.sleep(1)
#            gp.output(buzz, gp.LOW)
        else:
            gp.output(led[0], False)
            gp.output(buzz, gp.LOW)
#            time.sleep(1)
#            gp.output(buzz, False)
        if(gp.input(room[1])):
            gp.output(led[1], True)
            gp.output(buzz, gp.HIGH)
#            time.sleep(1)
#            time.sleep(1)
#            gp.output(buzz[1], True)
        else:
            gp.output(led[1], False)
            gp.output(buzz, gp.LOW)
#            time.sleep(1)
#            gp.output(buzz[0], False)

#        time.sleep(2)

```