**Hardware using Raspberry Pi Zero**



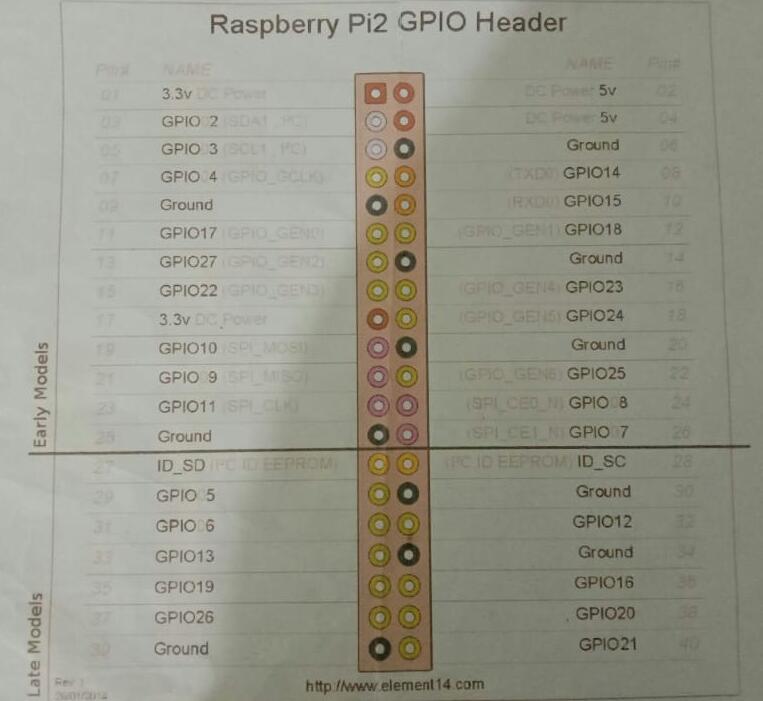
**fig:Raspbery pi zero board:**

**Introduction to raspberrypi zero:**

The Raspberry Pi Zero W extends the Pi Zero family. It have a functionality of 802.11 b/g/n wireless LAN , Bluetooth 4.1, Bluetooth Low Energy(BLE) and it also has : 1GHz, single-core CPU, Mini HDMI and USB on-The-Go ports, Micro USB power, HAT-compatible 40-pin header and CSI camera connector and it also have Buit-In BCM43143 WiFi chip. It works on 5volts

below figure shows the pin discription of the raspberypi. the raspberypi will have 40pins in that it have 26 GPIO pins and two dc power pins and 6 ground pins so on.

**Fig. pin description:**



**LED Blinking**

LED stands for light emitting diode.LED lighting products produce light approximately 90% more efficiently than incandescent light bulbs.below connection shows the led interface with raspberypi board .The raspberypi board pin nuber 3 will connected to the led positive terminal and pin number 39 will connected to the led's negative terminal. below program will show the blinking of led when run the program the led start to turn on after one second of delay the led will turn off and after one second of delay the led will turn on again.

**Example:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

led=3

GPIO.setup(led,GPIO.OUT)

while(1):

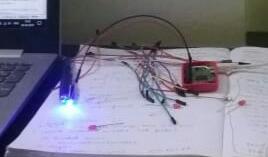
GPIO.output(led,1)

time.sleep(1)

GPIO.output(led,0)

time.sleep(1)

**output:**



**Pushbutton:**

A push-button or simply button is a simple switch mechanism for controlling some aspect of a machine  or a process. The push button is used for to blink the led if press the push button the led is turn on to releases the push button the led turns off.

**Example:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

sw=3

led=5

i=0

GPIO.setup(led,GPIO.OUT)

GPIO.setup(sw,GPIO.IN)

while(1):

var=GPIO.input(sw)

print(var)

if(var==0)and (i==0):

GPIO.output(led,1)

time.sleep(1)

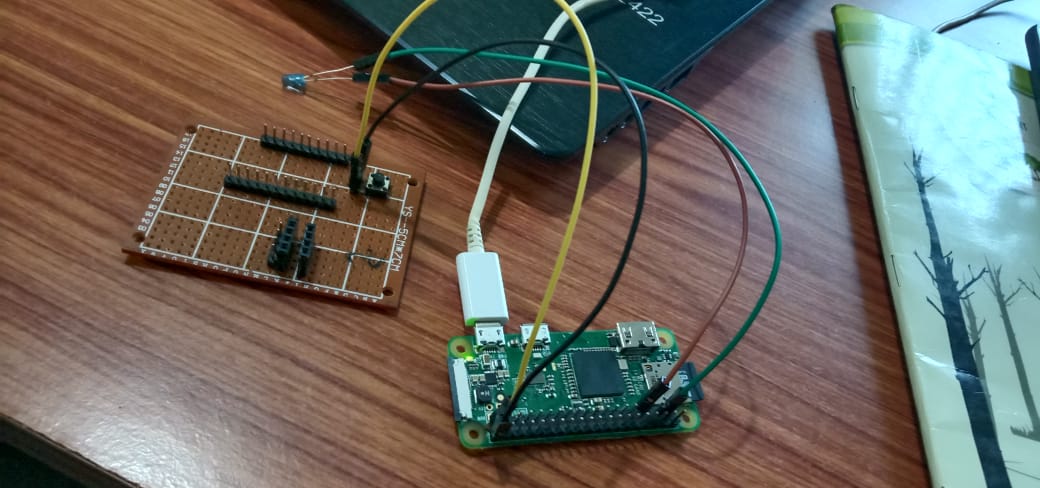
i=i+1

elif(var==0)and(i==1):

GPIO.output(led,0)

time.sleep(1)

i=0



**Dc motor:**

To interface the dc motor to the raspberypi board use the dc motor driver the driver will have the terminals of in1,in2 and power suply pin. The dc driver will drive the motor . we will write a program for direction of the dc motor if we given a input one it rotate on clockwise if we given a input as zero it rotate anticlockwise.

**Example:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

sw=7

i=0

in1=3

in2=5

GPIO.setup(sw,GPIO.IN,pull\_up\_down=GPIO.PUD\_UP)

GPIO.setup(in1,GPIO.out)

GPIO.setup(in2,GPIO.out)

while(1):

var=GPIO.input(sw)

if(var==0)and(i==0)

GPIO.output(in1,1)

GPIO.output(in2,0)

time.sleep(1)

i=i+1

elif(var==0)and(i==0)

GPIO.output(in1,0)

GPIO.output(in2,1)

time.sleep(1)

i=i+1

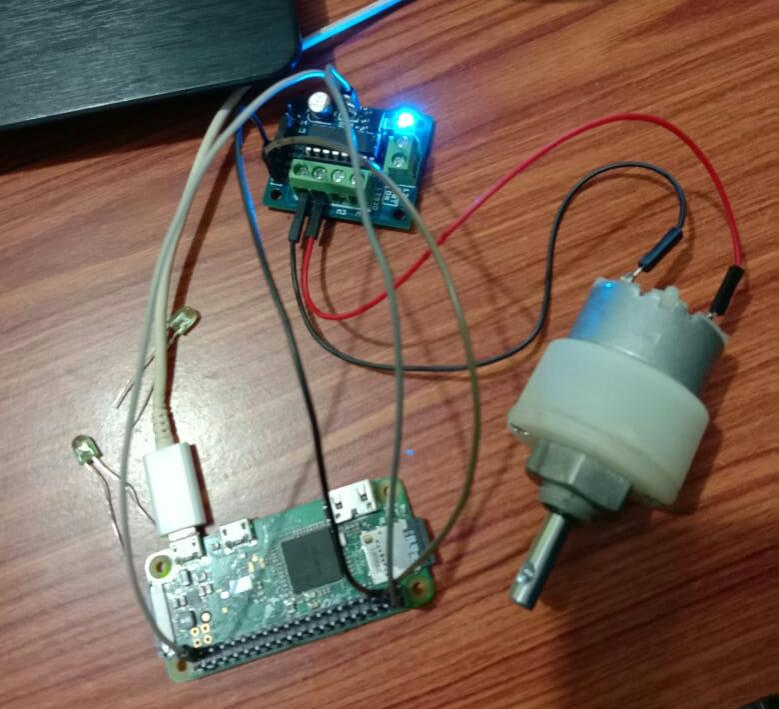
elif(var==0)and(i==0)

GPIO.output(in1,0)

GPIO.output(in2,0)

time.sleep(1)

**output:**



**LCD display:**

In this experiment we use 16x2 lcd display. The 16x2 means it have 2rows and 16 colomns in each row we can display the 16 charector in the total display we can display the 32 charector and the lcd will stores the values in each row 40 charectors.

in lcd display have d4-d7 data pins and enable pin the enable and data pins are connected to the GPIO pins.

**Example:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

en,rs,d4,d5,d6,d7=3,5,40,11,13,15

GPIO.setup(en,GPIO.OUT)

GPIO.setup(rs,GPIO.OUT)

GPIO.setup(d4,GPIO.OUT)

GPIO.setup(d5,GPIO.OUT)

GPIO.setup(d6,GPIO.OUT)

GPIO.setup(d7,GPIO.OUT)

def enable():

GPIO.output(en,1)

time.sleep(0.05)

GPIO.output(en,0)

time.sleep(0.05)

def cmd(x):

GPIO.output(rs,0)

GPIO.output(d4,0)

GPIO.output(d5,0)

GPIO.output(d6,0)

GPIO.output(d7,0)

if x & 0x10==0x10:

GPIO.output(d4,1)

if x & 0x20==0x20:

GPIO.output(d5,1)

if x & 0x40==0x40:

GPIO.output(d6,1)

if x & 0x80==0x80:

GPIO.output(d7,1)

enable()

GPIO.output(d4,0)

GPIO.output(d5,0)

GPIO.output(d6,0)

GPIO.output(d7,0)

if x & 0x01==0x01:

GPIO.output(d4,1)

if x & 0x02==0x02:

GPIO.output(d5,1)

if x & 0x04==0x04:

GPIO.output(d6,1)

if x & 0x08==0x08:

GPIO.output(d7,1)

enable()

def data(y):

GPIO.output(rs,1)

GPIO.output(d4,0)

GPIO.output(d5,0)

GPIO.output(d6,0)

GPIO.output(d7,0)

if y & 0x10==0x10:

GPIO.output(d4,1)

if y & 0x20==0x20:

GPIO.output(d5,1)

if y & 0x40==0x40:

GPIO.output(d6,1)

if y & 0x80==0x80:

GPIO.output(d7,1)

enable()

GPIO.output(d4,0)

GPIO.output(d5,0)

GPIO.output(d6,0)

GPIO.output(d7,0)

if y & 0x01==0x01:

GPIO.output(d4,1)

if y & 0x02==0x02:

GPIO.output(d5,1)

if y & 0x04==0x04:

GPIO.output(d6,1)

if y & 0x08==0x08:

GPIO.output(d7,1)

enable()

def string(msg):

for i in range(len(msg)):

data(ord(msg[i]))

def init():

cmd(0x33)

cmd(0x32)

cmd(0x28)

cmd(0x06)u

cmd(0x0c)

cmd(0x01)

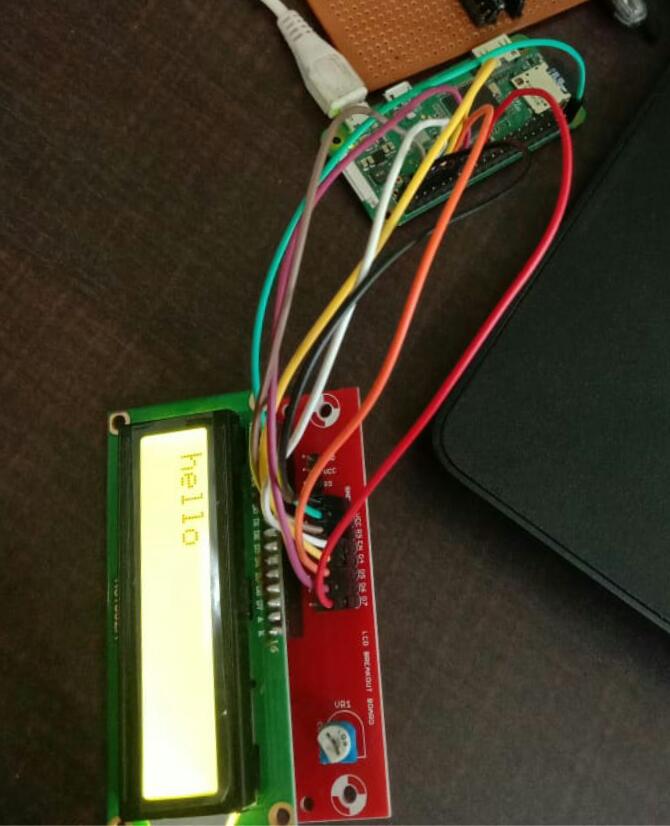
while(1):

init()

cmd(0x80)

string("ANTP")

time.sleep(1)



**Ultrasonic sensor:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

trig=3

echo=5

GPIO.setup(7,GPIO.OUT)

GPIO.setup(11,GPIO.OUT)

GPIO.setup(trig,GPIO.OUT)

GPIO.setup(echo,GPIO.IN)

while(1):

GPIO.output(trig,1)

time.sleep(0.5)

GPIO.output(trig,0)

while(GPIO.input(echo)==0):

start=time.time()

while(GPIO.input(echo)==1):

stop=time.time()

elapse=stop-start

distance=(elapse\*34300)/2

print distance

if(distance>=100):

GPIO.output(7,1)

time.sleep(1)

GPIO.output(7,0)

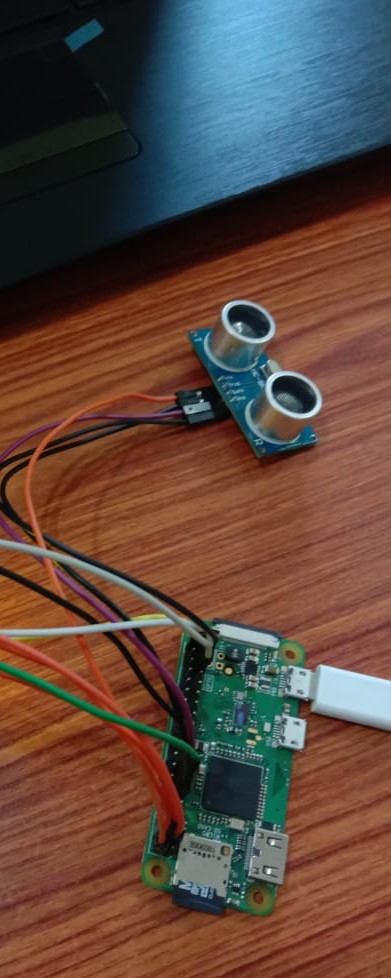
else:

GPIO.output(11,1)

time.sleep(1)

GPIO.output(11,0)

**Output:**

****

**Temperature sensor:**

Temperature sensor is used to check the temperature and humidity.here we use DHT sensor.we have to install a library function to use a temperature sensor.here we need not define gpio pins.it is setup by default.

**Example:**

Import Adafruit\_DHT

Import RPI.GPIO as gpio

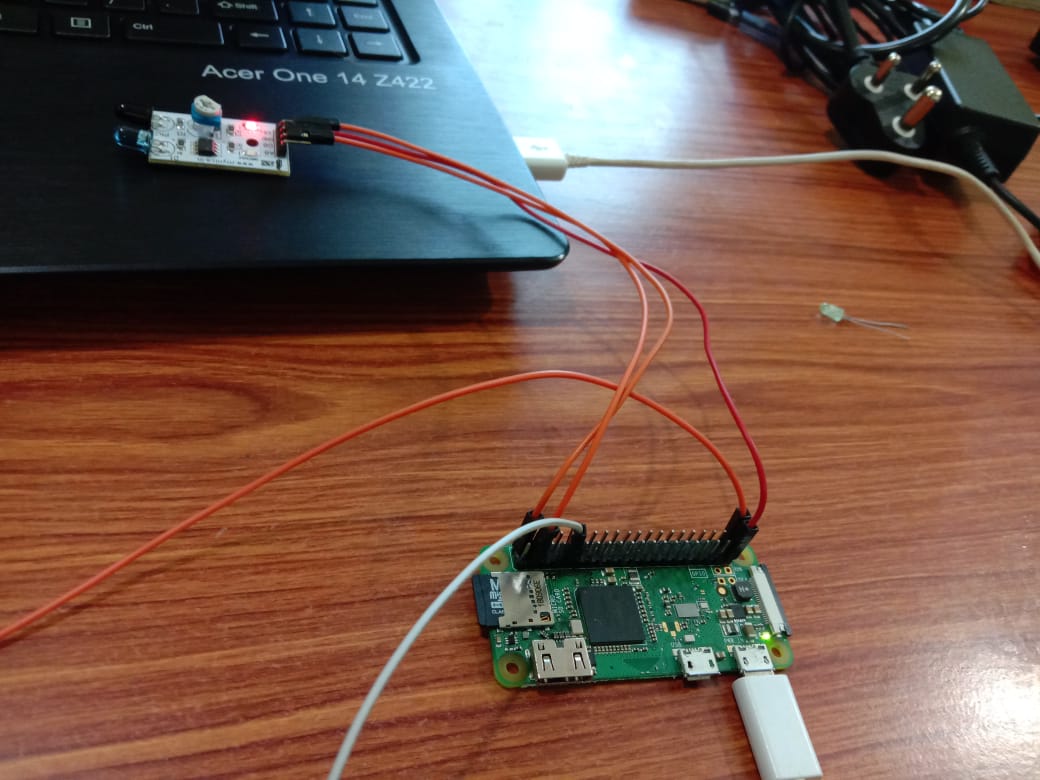
Import time

While(1):

Temperature , humidity =Adfruit-DHT.read-retry

Print temperature , humidity

**Output:**

****

We also displayed temperature and humidity using lcd as below

**Example:**

Import Adafruit\_DHT

From lcd import\*

Import RPi.GPIO as gpio

Import time

While 1:

temp,hem=Adafruit\_DHT.read\_retry(11,4)

print “temperature=”,temp

print “humidity=”hem

init()

cmd(0x80)

string(“temperature”)

string(str(temp))

cmd(0xc0)

string(“humidity”)

string(str(hem))

time.sleep(1)

**Relay:**

A relay is an operated switch. It consists of a set of input terminals for a single or multiple control signals, and a set of operating contact terminals. The switch may have any number of contacts in multiple contact forms, such as make contacts, break contacts, or combinations there of.Relays are used where it is necessary to control a circuit by an independent low-power signal, or where several circuits must be controlled by one signal. In the below experiment we use the relay for the controlling or switching of 230v bulb as shown in below.

**Example:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

sw=15

re=3

GPIO.setup(re,GPIO.OUT)

GPIO.setup(sw,GPIO.IN,pull\_up\_down=GPIO.PUD\_UP)

while(1):

var=GPIO.input(sw)

print var

if(var==0):

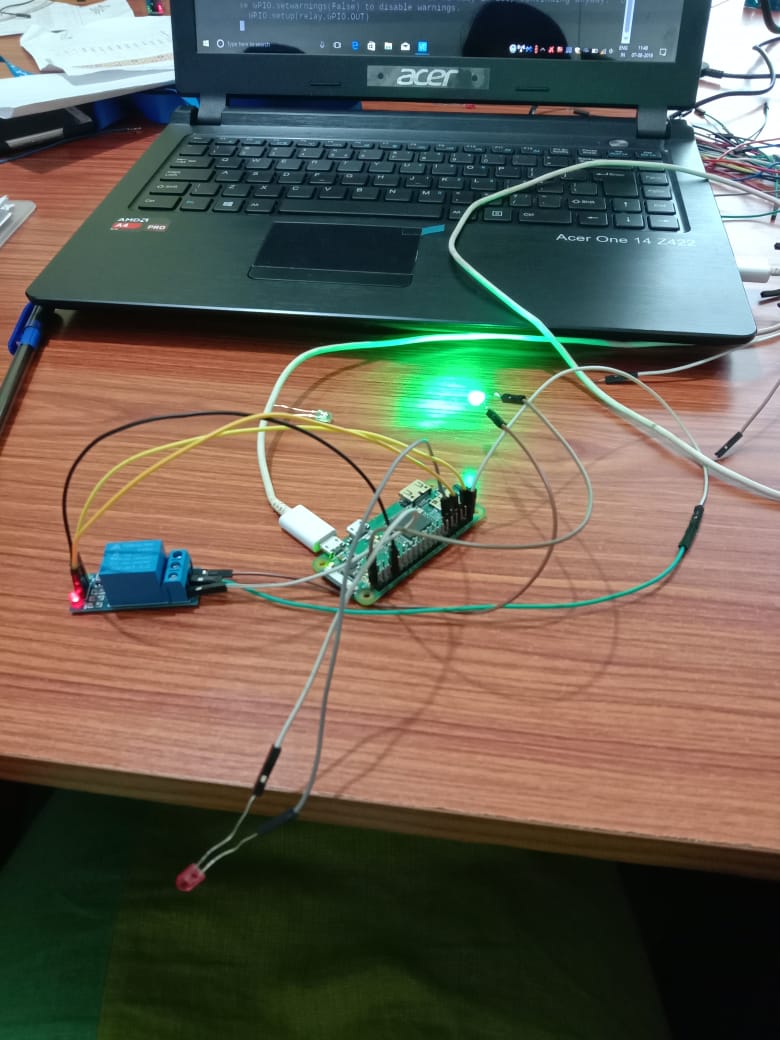
GPIO.output(re,1)

time.sleep(0.5)

else:

GPIO.output(re,0)

time.sleep(0.5)

**Output:** 

The below figure shows that controlling 230 volts of bulb using relay.



**IR sensor(Infrared sensor):**

Infrared sensor it will use for the object detection. in the IR sensor will spread the signals if the object will apear in the range of the IR waves it detects the abstractors.

**Example:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

led=5

ir=3

i=0

count=0

GPIO.setup(ir,GPIO.IN)

GPIO.setup(led,GPIO.OUT)

while(1):

var=GPIO.input(ir)

# print var

if(var == 1):

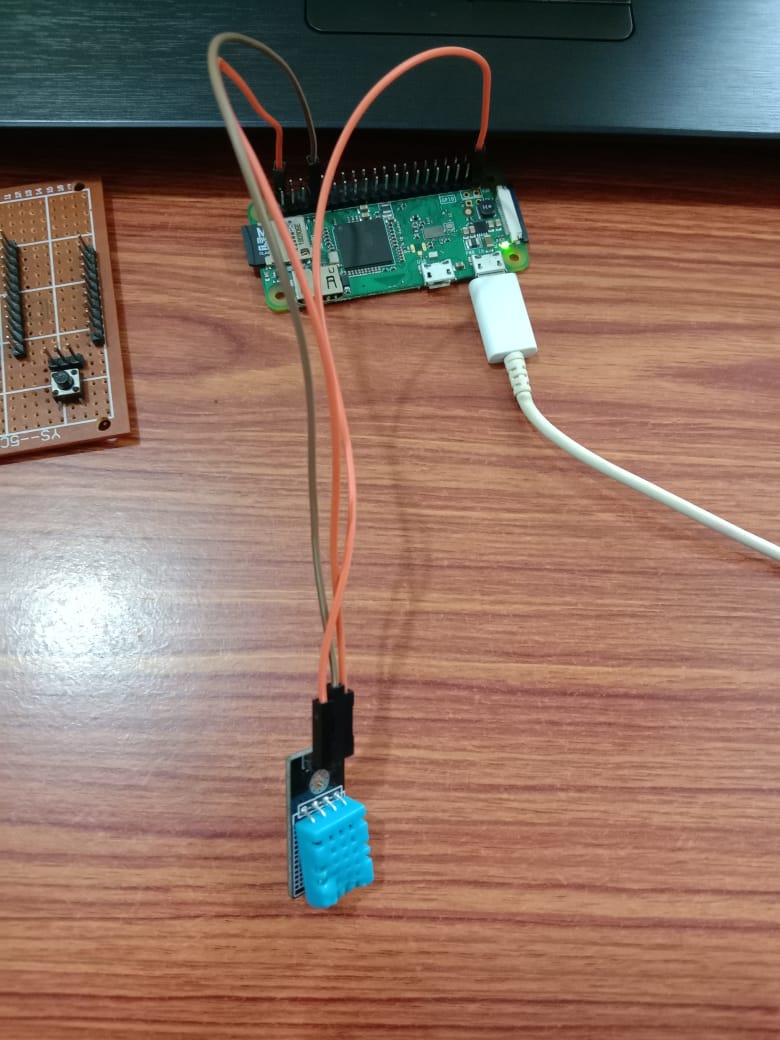
var = GPIO.input(ir)

if(var == 0):

count = count + 1

print count

**output:**

****

**RFID Reader:**

A radio frequency identification reader (RFID reader) is a device used to gather information from an RFID tag, which is used to track individual objects. Radio waves are used to transfer data from the tag to a reader.  
  
RFID is a technology similar in theory to bar codes. However, the RFID tag does not have to be scanned directly, nor does it require line-of-sight to a reader. The RFID tag it must be within the range of an RFID reader, which ranges from 3 to 300 feet, in order to be read. RFID technology allows several items to be quickly scanned and enables fast identification of a particular product, even when it is surrounded by several other items.

**Example:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

GPIO.setup(3,GPIO.OUT)

GPIO.setup(7,GPIO.OUT)

from mfrc522 import SimpleMFRC522

reader=SimpleMFRC522()

while(1):

text="log"

reader.write(text)

id,text=reader.read()

print id

# print text

if id == 80852164341:

GPIO.output(3,1)

time.sleep(1)

GPIO.output(3,0)

print "correct"

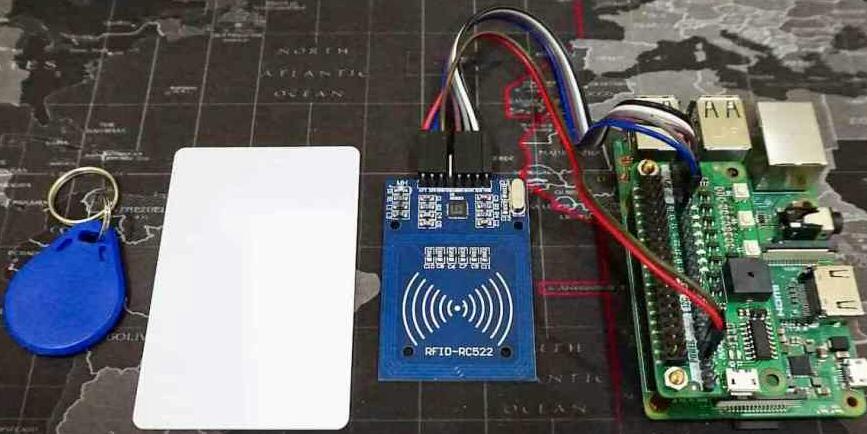
else:

GPIO.output(7,1)

time.sleep(1)

GPIO.output(7,0)

print "authentication\_error"

**output:**

**Fire sensor:**

A fire detector is a sensor designed to detect and respond to the presence of a fire allowing fire detection. The fire sensor will detect the fire and given the output as one when it detect the fire, it display the output as zero if absence of fire.

**Example:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

ir=3

i=5

i1=7

GPIO.setup(ir,GPIO.IN)

GPIO.setup(i,GPIO.OUT)

GPIO.setup(i1,GPIO.OUT)

while(1):

var=GPIO.input(ir)

print(var)

if(var==0):

GPIO.output(i,1)

GPIO.output(i1,0)

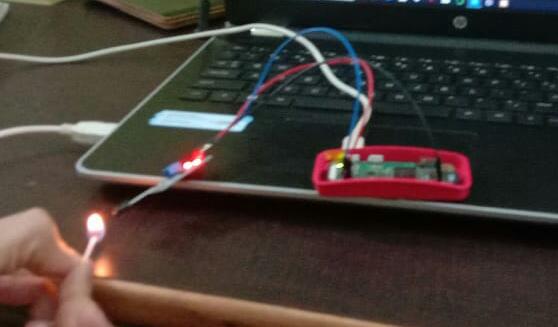
time.sleep(1)

else:

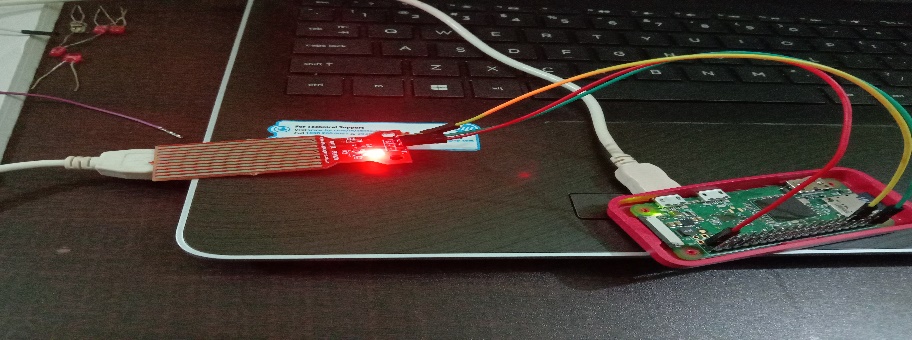
GPIO.output(i,0)

GPIO.output(i1,0)

**Output for fire sensor:**



**Output for water sensor:**

****

**Bluetooth:**

Bluetooth is a wireless technology standard for exchanging data between fixed and mobile devices over short distances using short-wavelength UHF radio waves in the industrial, scientific and medical radio bands, from 2.400 to 2.485 GHz, and building personal area networks(PANs).To use or interface with a bluetooth firt we download the bluetooth serial controller in the mobile and it will controll the bluetooth operation. The bluetooth is interfaces to the bluetooth with raspberypi board bluetooth will have 6pins in that only 4pins are connected.TXD pin ofbluetooth is connected to raspberypi RXD pin vice versa.

**Example:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

sw=7

i=0

in1=3

in2=5

GPIO.setup(sw,GPIO.IN,pull\_up\_down=GPIO.PUD\_UP)

GPIO.setup(in1,GPIO.OUT)

GPIO.setup(in2,GPIO.OUT)

while(1):

var=GPIO.input(sw)

if(var==0)and(i==0):

GPIO.output(in1,1)

GPIO.output(in2,0)

time.sleep(1)

i=i+1

elif(var==0)and(i==1):

GPIO.output(in1,0)

GPIO.output(in2,1)

time.sleep(1)

i=i+1

elif(var==0)and(i==2):

GPIO.output(in1,0)

GPIO.output(in2,0)

time.sleep(1)

**output:** 

**Hexakeypad:**

The keypad is an arrangement of 16 push button switches in the form of a 4x4 matrix. The keypad consists of numbers 0,1,2,3,4,5,6,7,8,9 and letters A, B, C, D,\*, #. Column scanning method is used to identify the pressed key. Below program shows that when we press the key pad then it will recognize the key and display that and the hexa keypad will interface with a raspberypi board, the hexa keypad will have 8 pins in that 8 pins 4pins for rows and 4pins for coloms and that 8 pins are connected to the raspberypi GPIO pins . First we high the row pins and send zero to the colomn pins the colomn pins are output pins and the row are input pins so when we press the key the key pressed colomn become hight and then the pi will read that value and compare to the given sequence and given the output.

**Example:**

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

MATRIX=[[1,2,3,'A'],[4,5,6,'B'],[7,8,9,'C'],['\*',0,'#','D']]

ROW=[3,5,7,11]

COL=[13,15,40,38]

ls=[1,2,3,4]

v=[0,0,0,0]

a=0

for j in range(4):

GPIO.setup(COL[j],GPIO.OUT)

GPIO.output(COL[j],1)

for i in range(4):

GPIO.setup(ROW[i],GPIO.IN,pull\_up\_down=GPIO.PUD\_UP)

while(1):

for j in range(4):

GPIO.output(COL[j],0)

for i in range(4):

if GPIO.input(ROW[i])==0:

print MATRIX[i][j]

time.sleep(0.2)

v[a]=MATRIX[i][j]

a=a+1

while(GPIO.input(ROW[i])==0):

pass

GPIO.output(COL[j],1)

if a==4:

if v==ls:

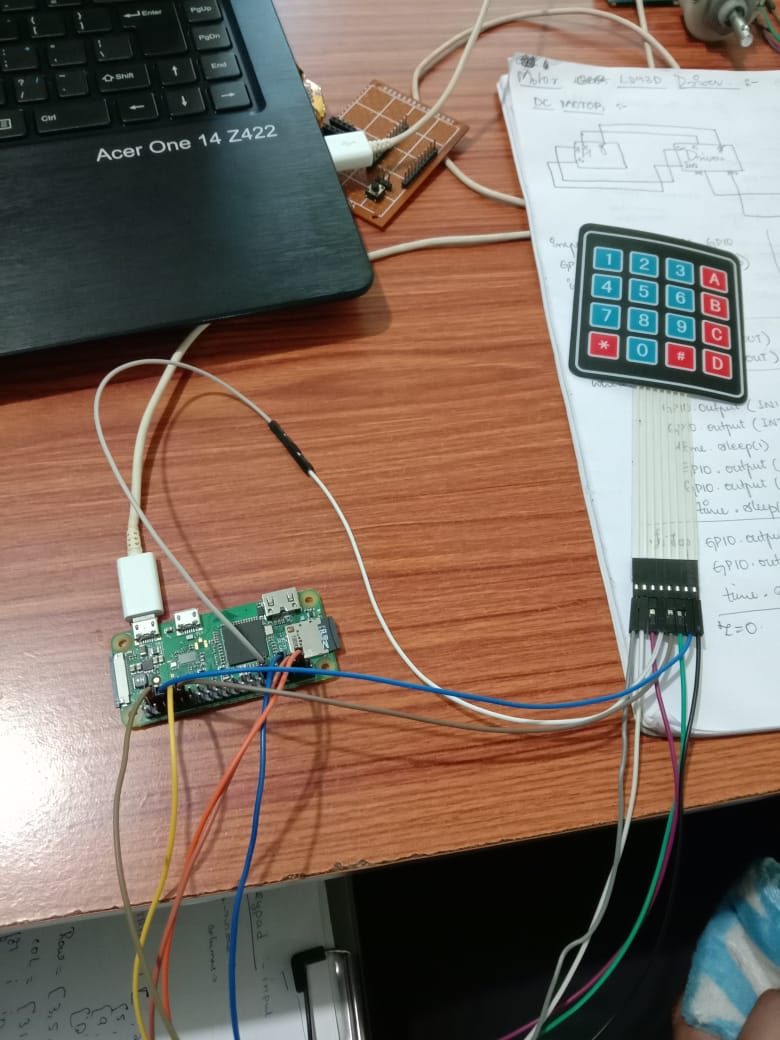
print "match"

else:

print "not match"

a=0

**output:**



**GSM Model:**

GSM is a mobile communication modem; it is stands for global system for mobile communication (GSM). The idea of GSM was developed at Bell Laboratories in 1970.  It is widely used mobile communication system in the world. GSM is an open and digital cellular technology used for transmitting mobile voice and data services operates at the 850MHz, 900MHz, 1800MHz and 1900MHz frequency bands. The GSM model will consists of 4 pins that are TXD,RXD,VCC and GND. The transmitter pin connected to the raspberypi's recever pin and GSM's recever pin connected to the raspberypi's transmitter pin and the GSM have one sim to communicate and it will work on the baud rate of 9600 b/s.

**Example:**

import RPi.GPIO as GPIO

import time

import serial

ser=serial.Serial('/dev/ttyS0',9600)

def msg():

ser.write('AT\r')

time.sleep(1)

ser.write('AT+CMGF=1\r')

time.sleep(1)

ser.write('AT+CMGS="9113647067"\r')

time.sleep(1)

ser.write('hello')

time.sleep(1)

ser.write('\x1A') # ascii value of ctrl z

time.sleep(1)

def call():

ser.write('ATD9113647067;\r')

def hang():

ser.write('ATH')

def ans():

ser.write('ATA\r')

while(1):

ans()

print"messge sent"

time.sleep(1)

dc

import RPi.GPIO as GPIO

import time

GPIO.setmode(GPIO.BOARD)

sw=7

i=0

in1=3

in2=5

GPIO.setup(sw,GPIO.IN,pull\_up\_down=GPIO.PUD\_UP)

GPIO.setup(in1,GPIO.OUT)

GPIO.setup(in2,GPIO.OUT)

while(1):

var=GPIO.input(sw)

if(var==0)and(i==0):

GPIO.output(in1,1)

GPIO.output(in2,0)

time.sleep(1)

i=i+1

elif(var==0)and(i==1):

GPIO.output(in1,0)

GPIO.output(in2,1)

time.sleep(1)

i=i+1

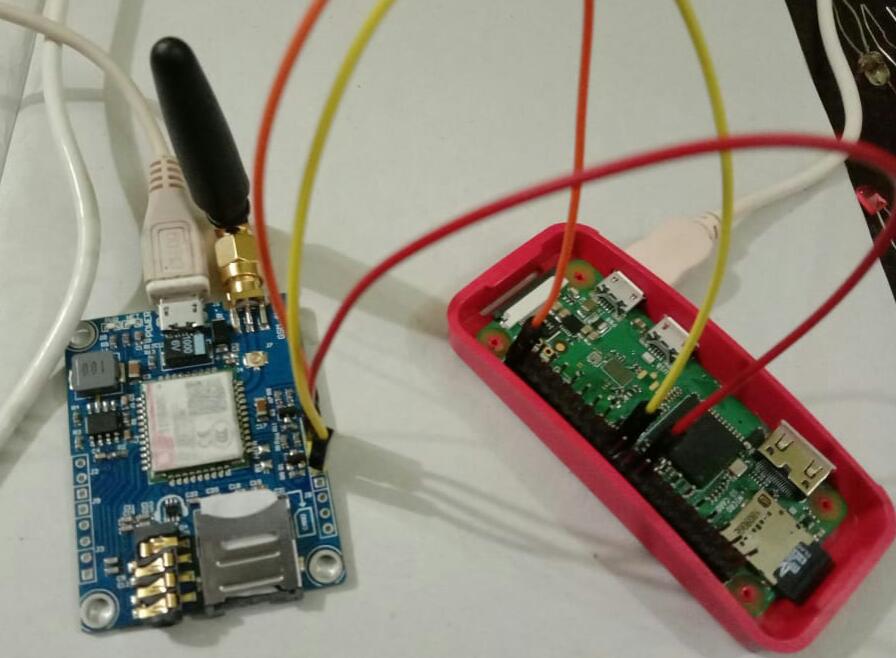
elif(var==0)and(i==2):

GPIO.output(in1,0)

GPIO.output(in2,0)

time.sleep(1)

**output:**

****