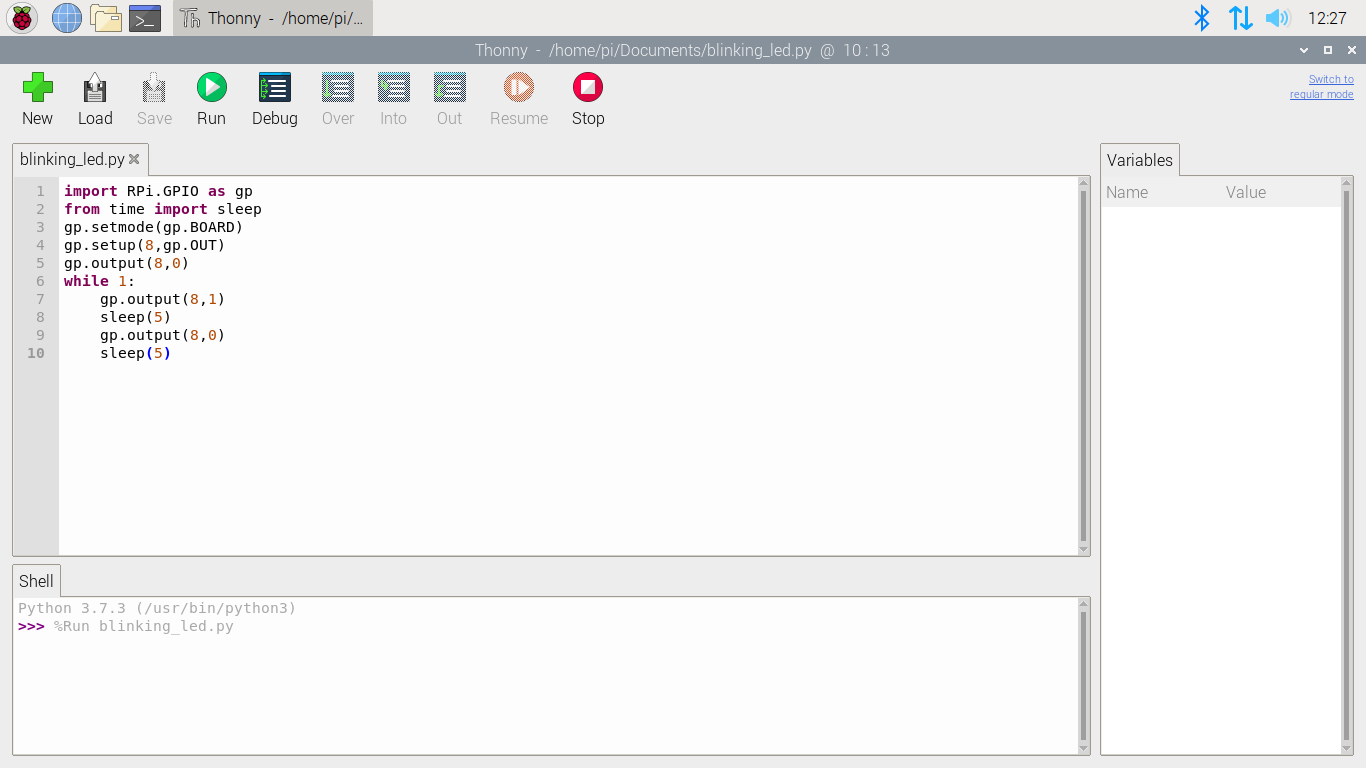
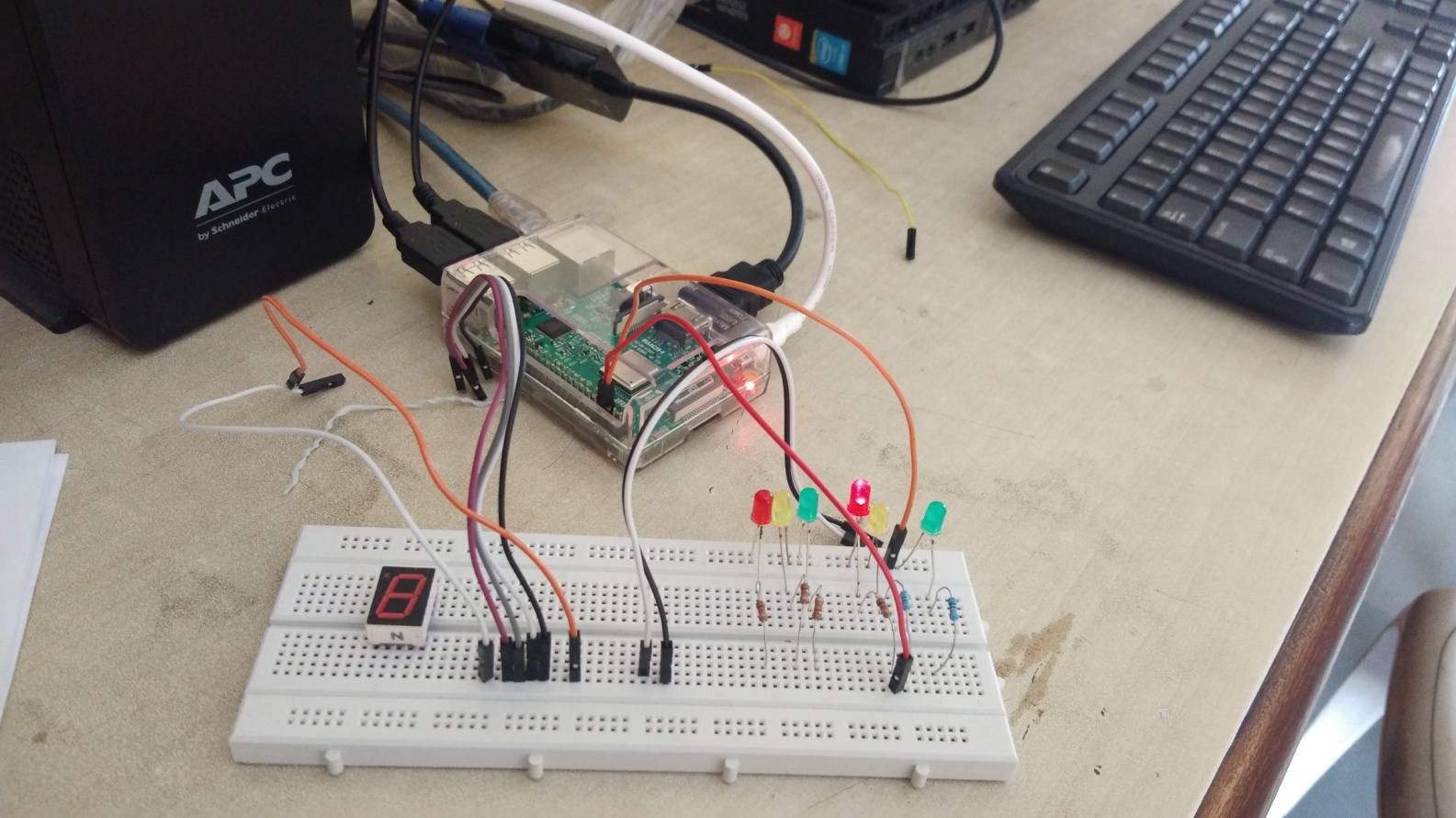
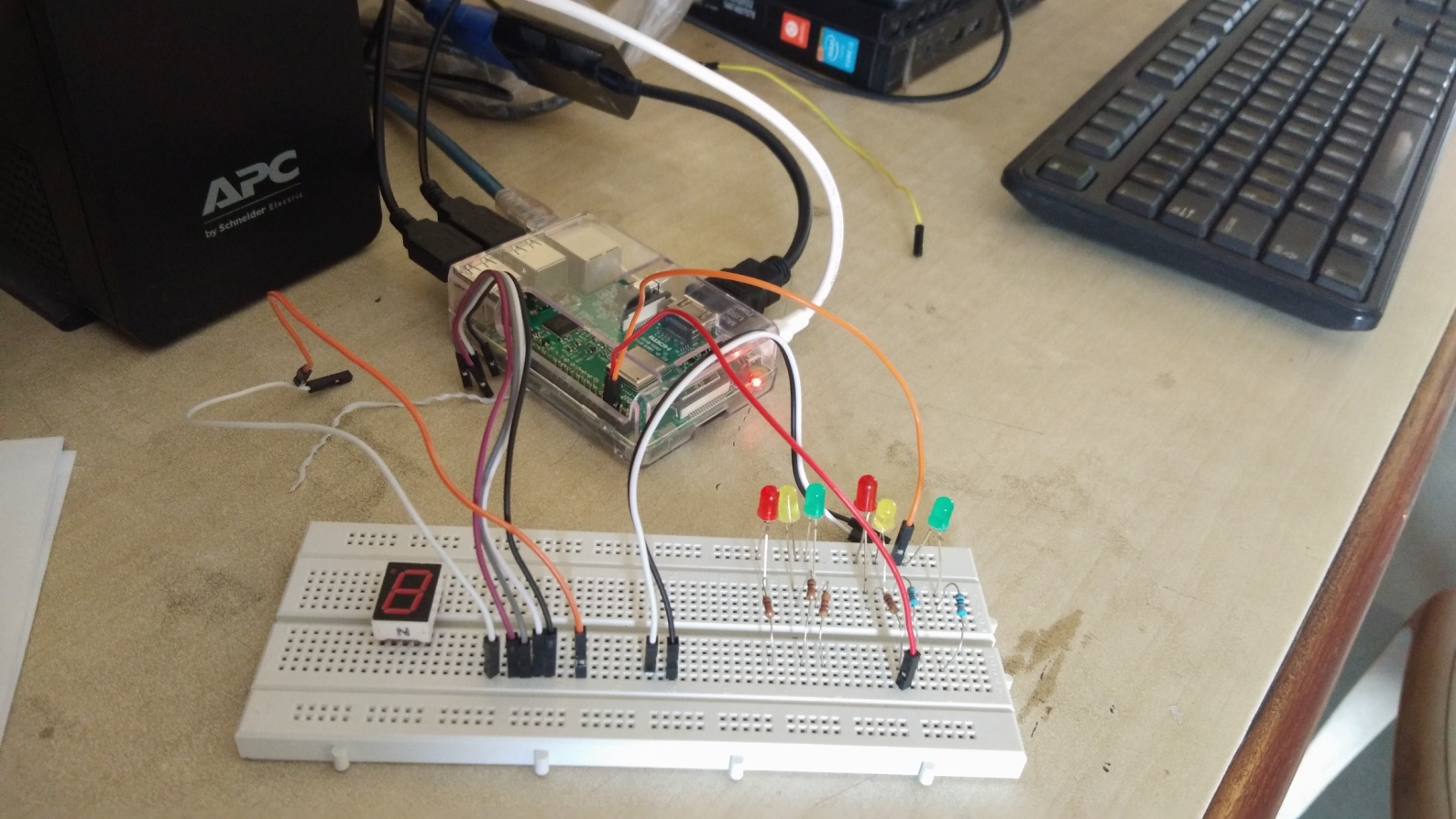
**LED Blinking using Raspberry Pi**

**Description:** The project is to teach students to handle Raspberry Pi programming. It is a simple programming for LED blinking.



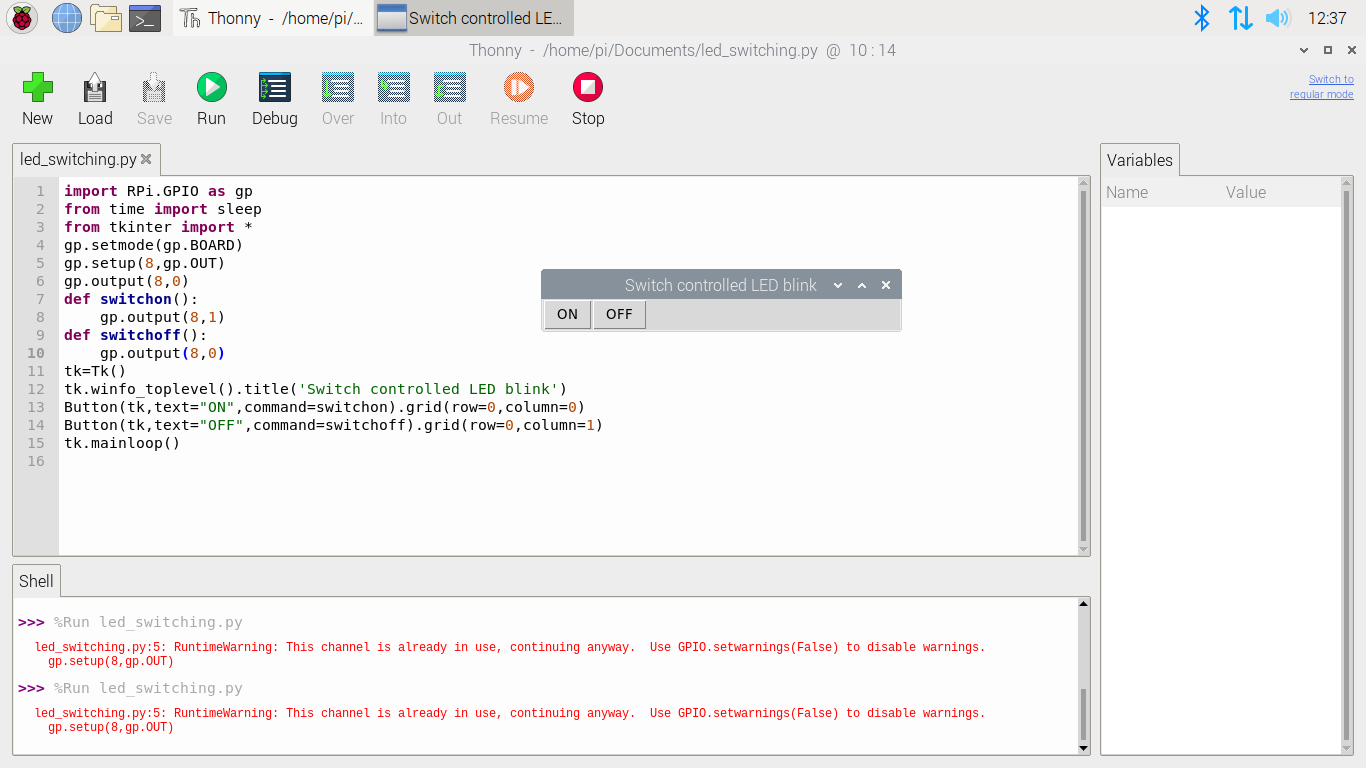
**Hardware Setup and Execution Image**



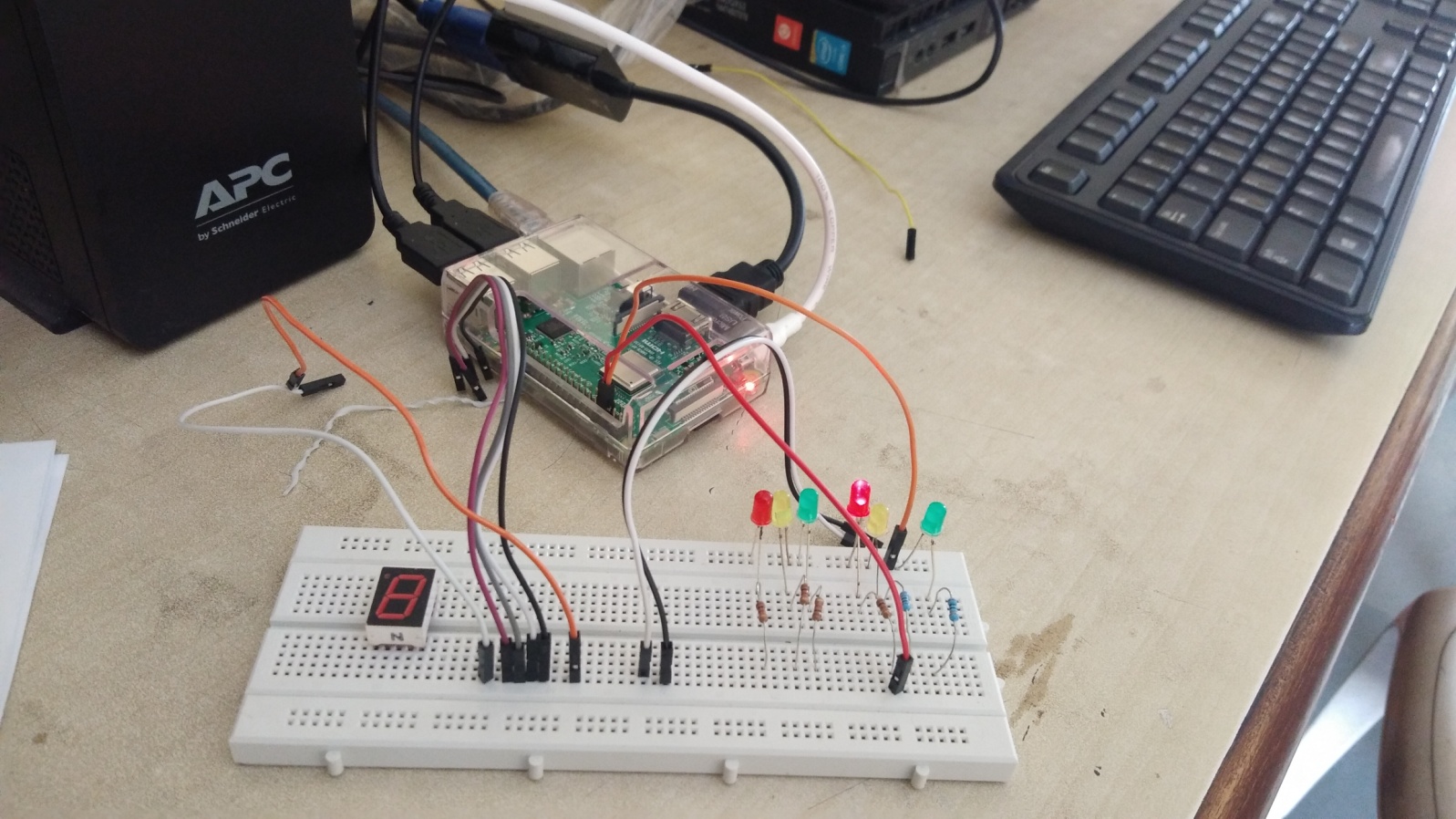


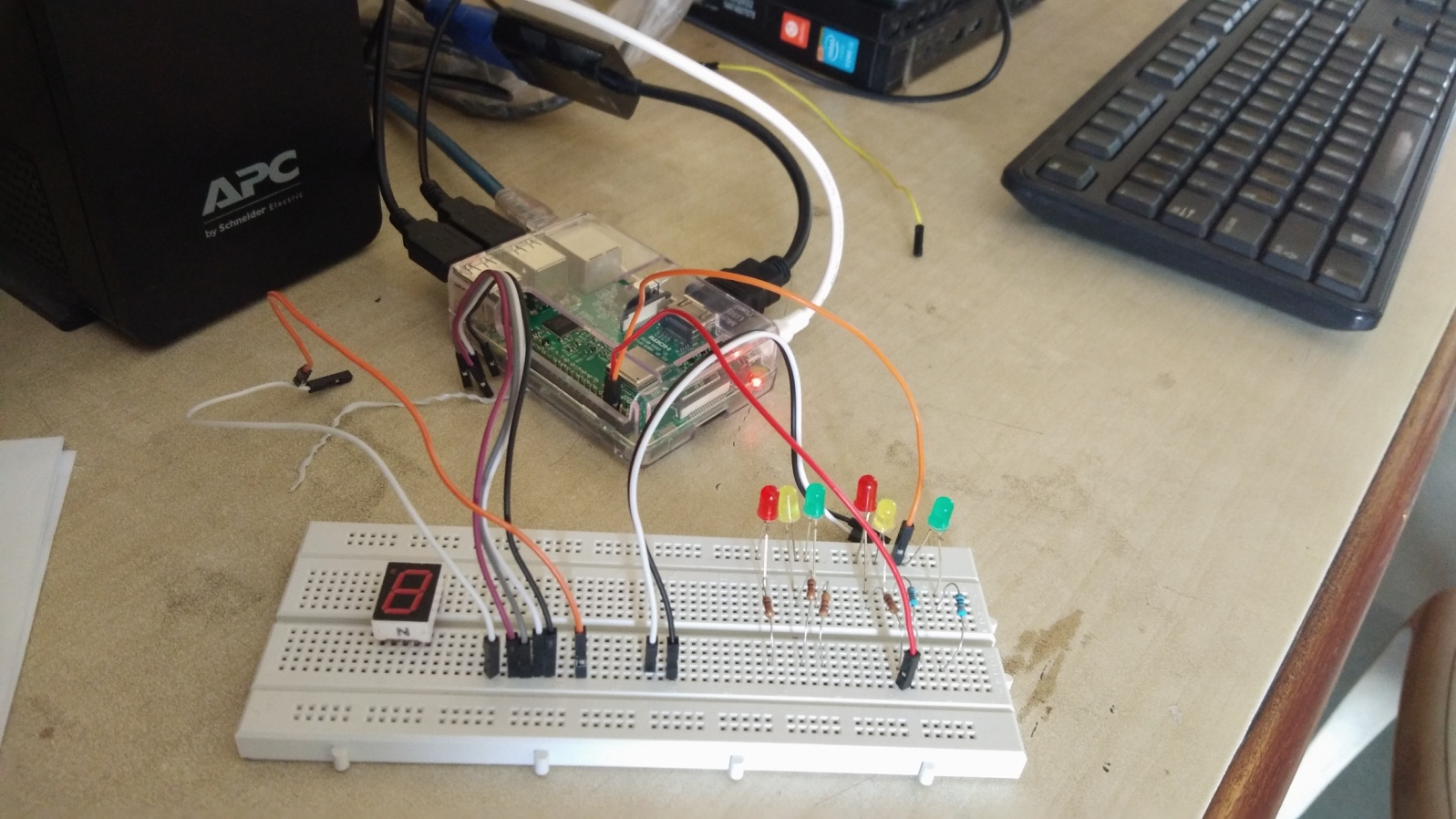
**Software based switch controlled LED Blinking using Raspberry Pi**

**Description:** The project is to train students to create simple UI in raspberry Pi using python to control the execution of LED blinking.



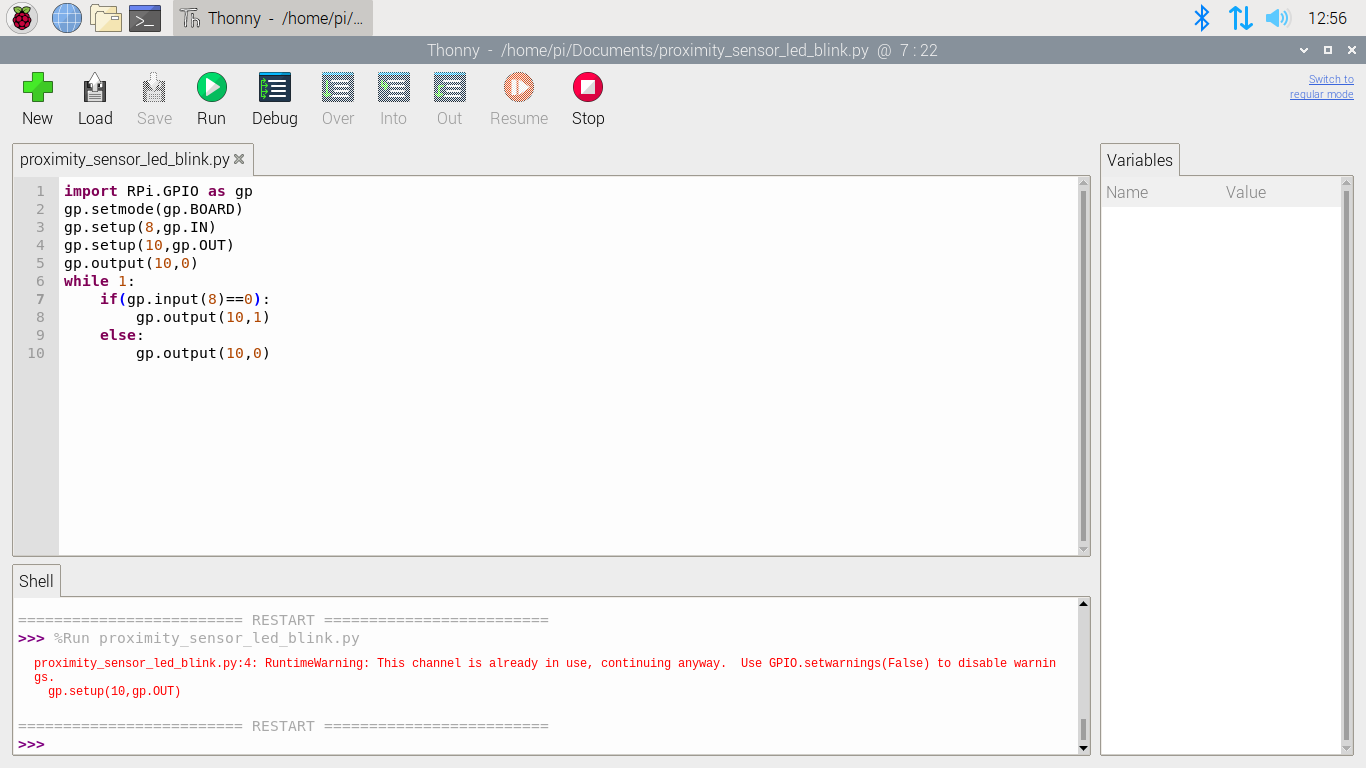
**Hardware Setup and Execution Image**



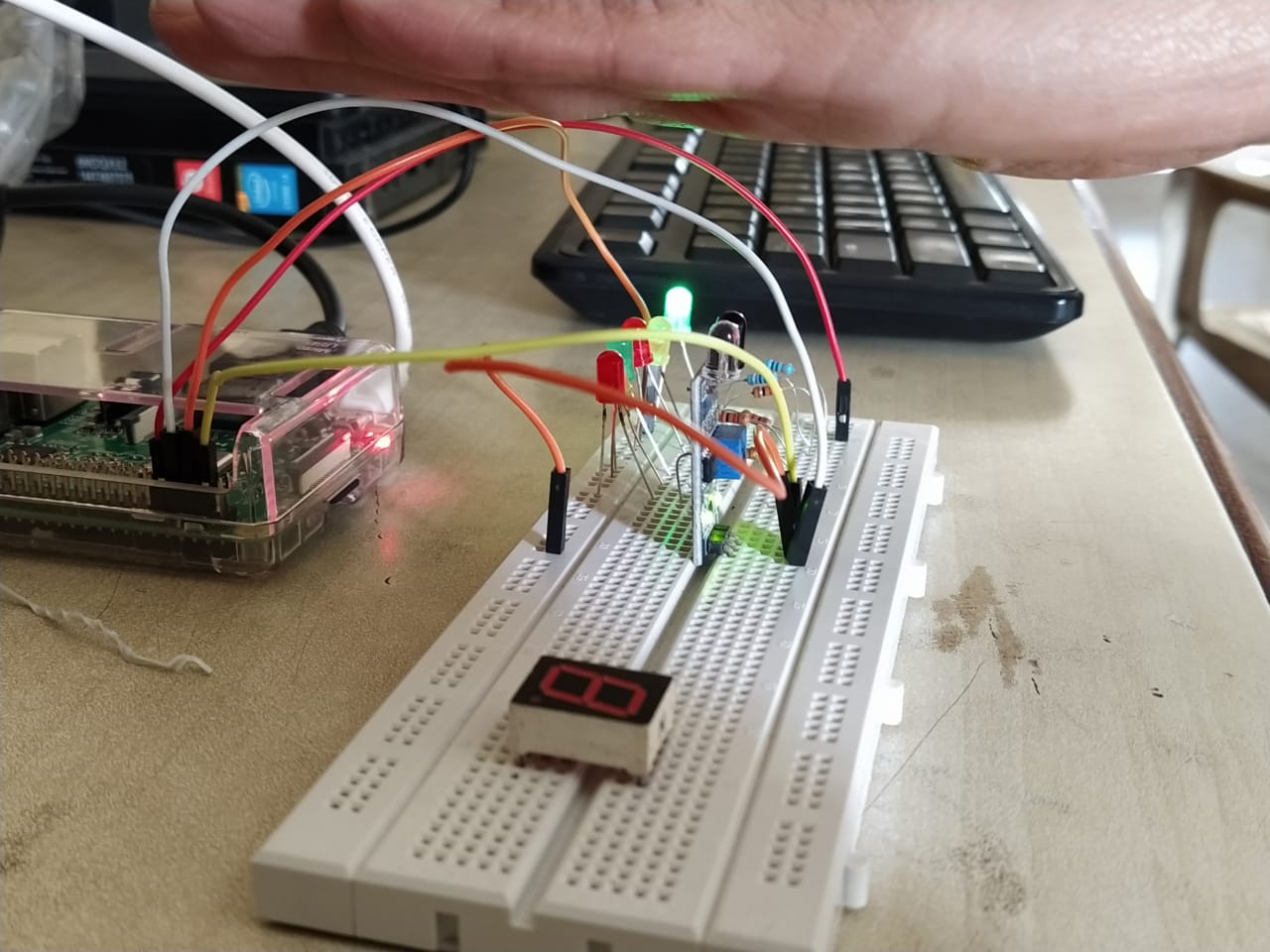


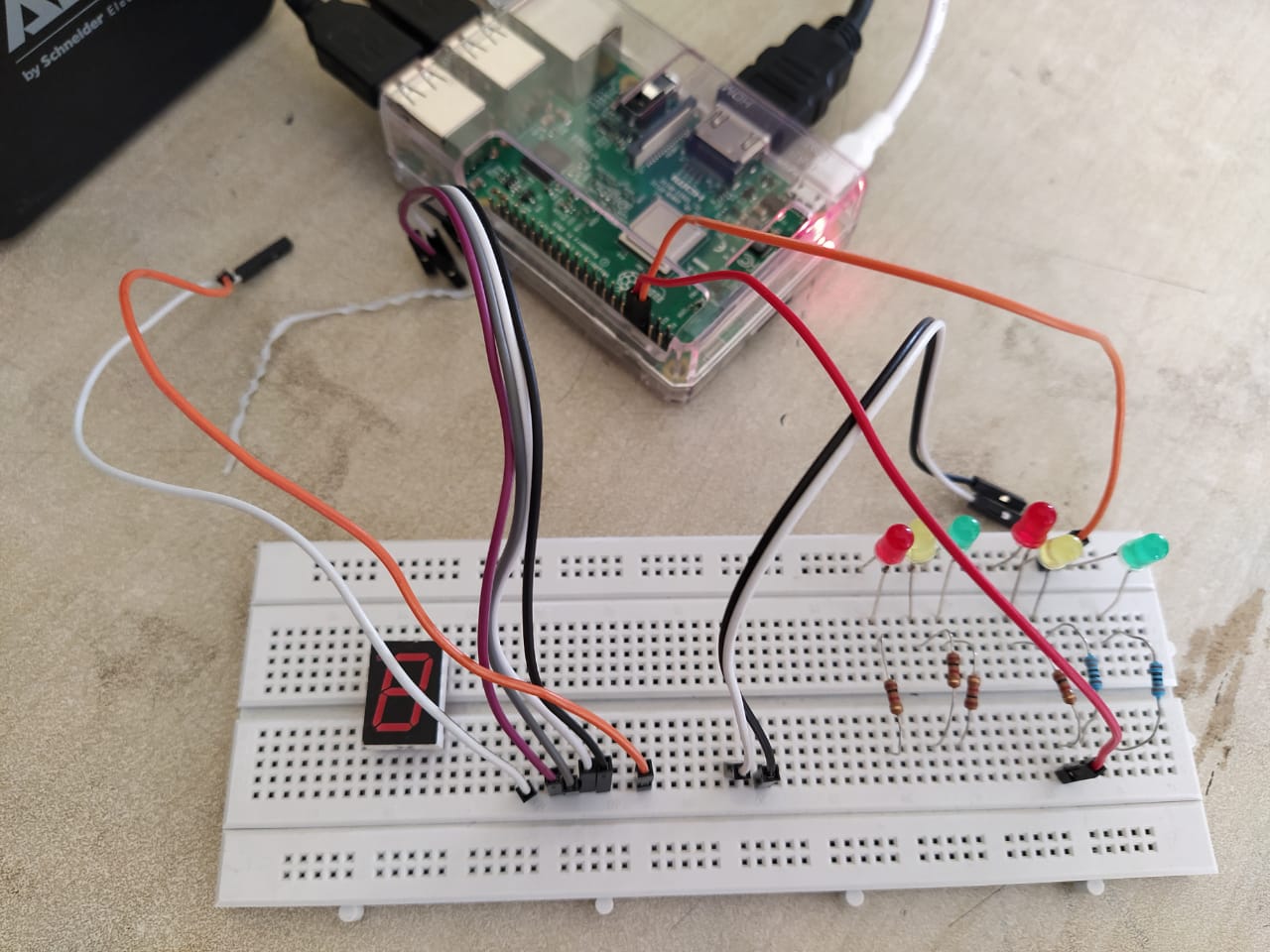
**Proximity Sensing by LED Blinking using Raspberry Pi**

**Description:** The project is to train students to interface proximity sensor with Raspberry Pi. Proximity sensing is indicated by LED blinking.



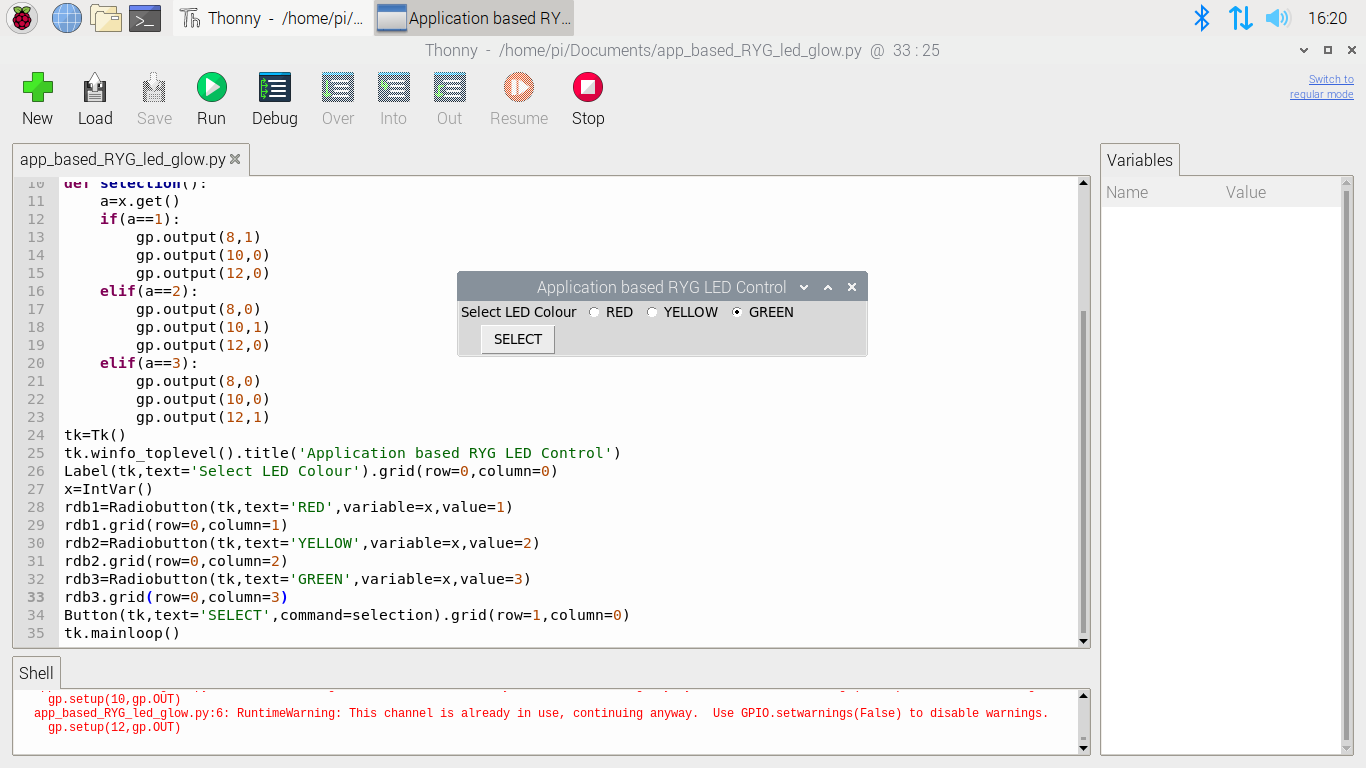
**Hardware Setup and Execution Image**



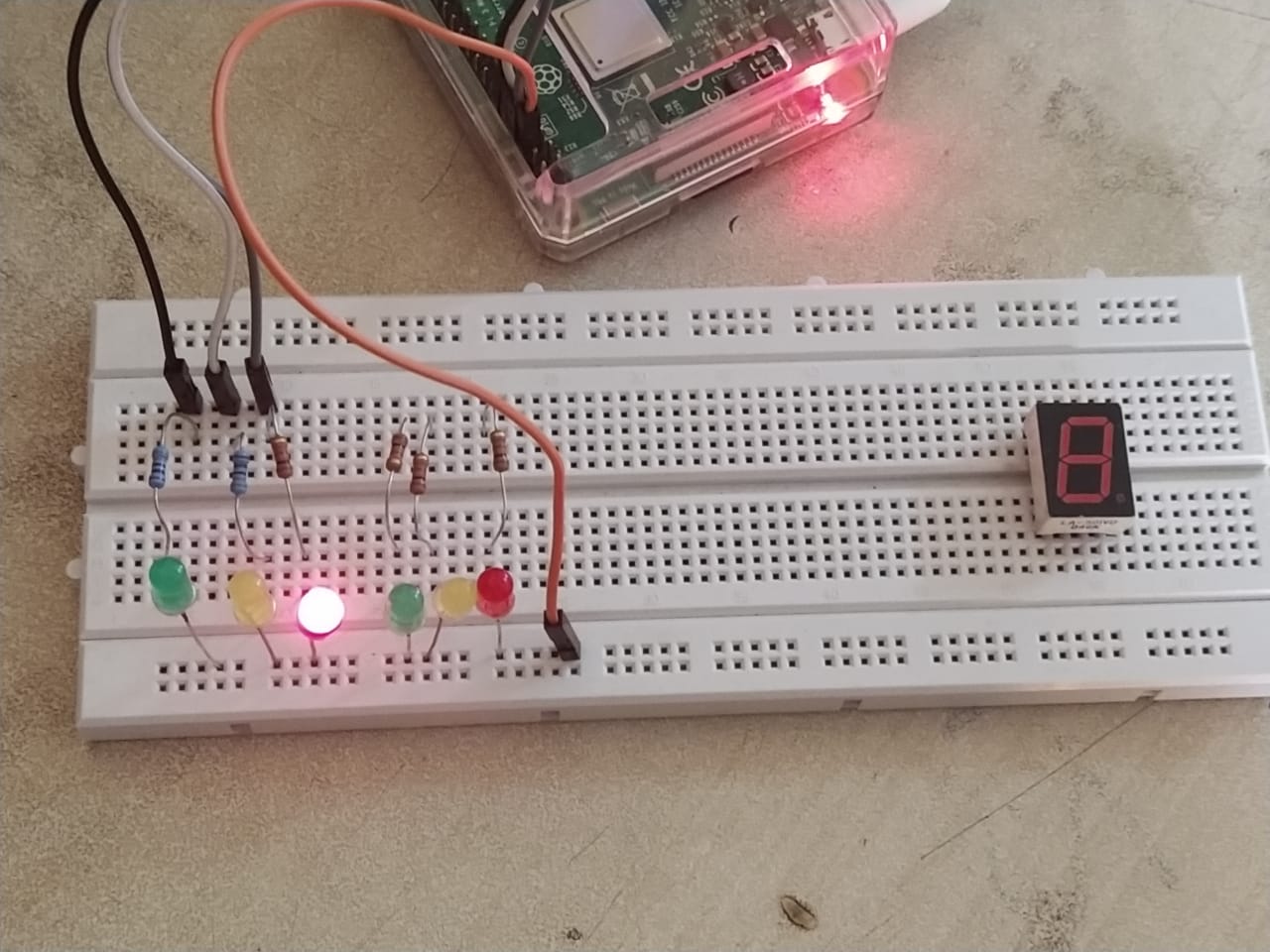


**Desktop application based RYG LED Glow using Raspberry Pi**

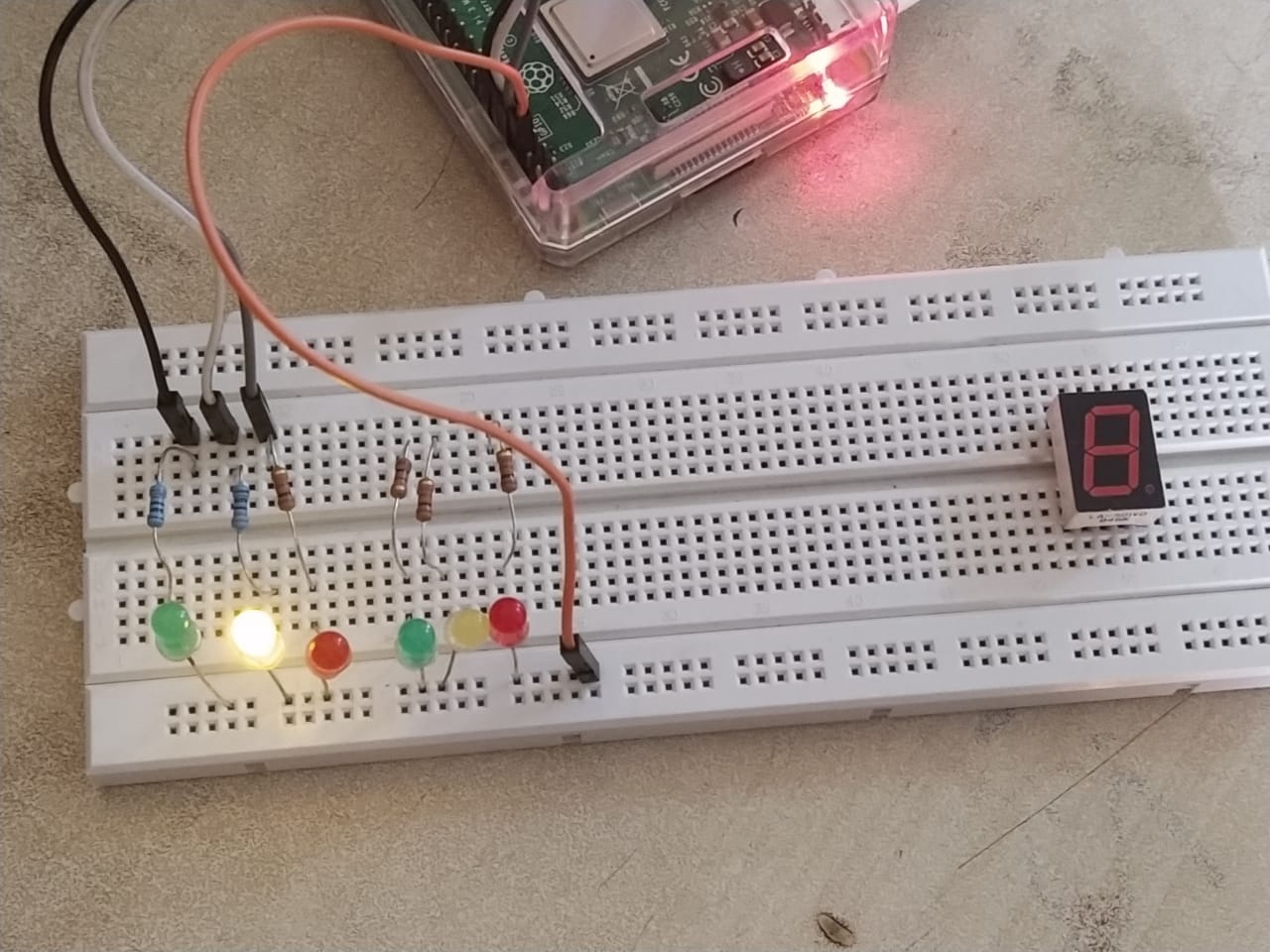
**Description:** The project is to train students to create desktop application using Raspberry Pi to control the glow of LEDs.



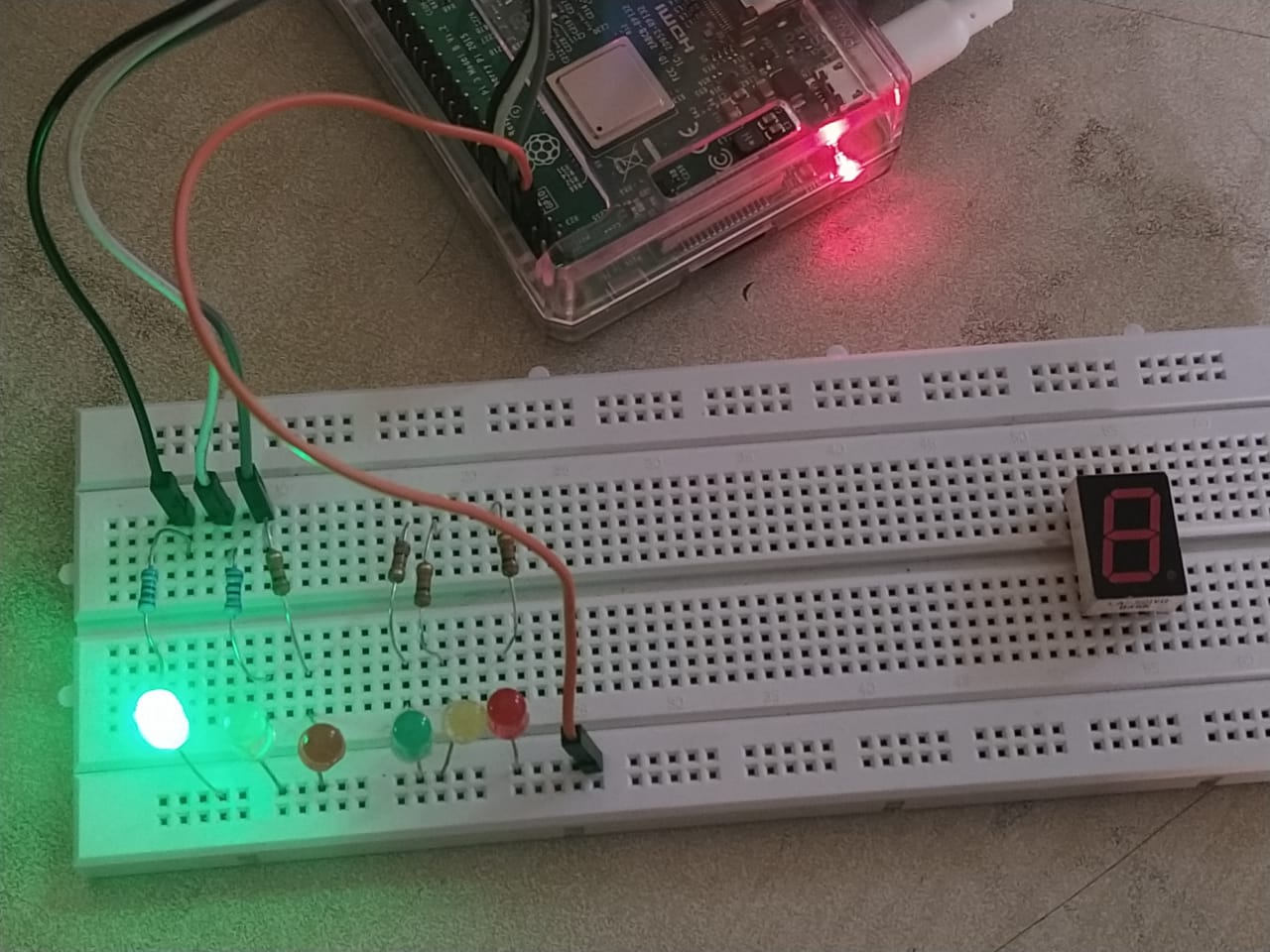
**Hardware Setup and Execution Image**



**RED LED Glow when RED button is selected**



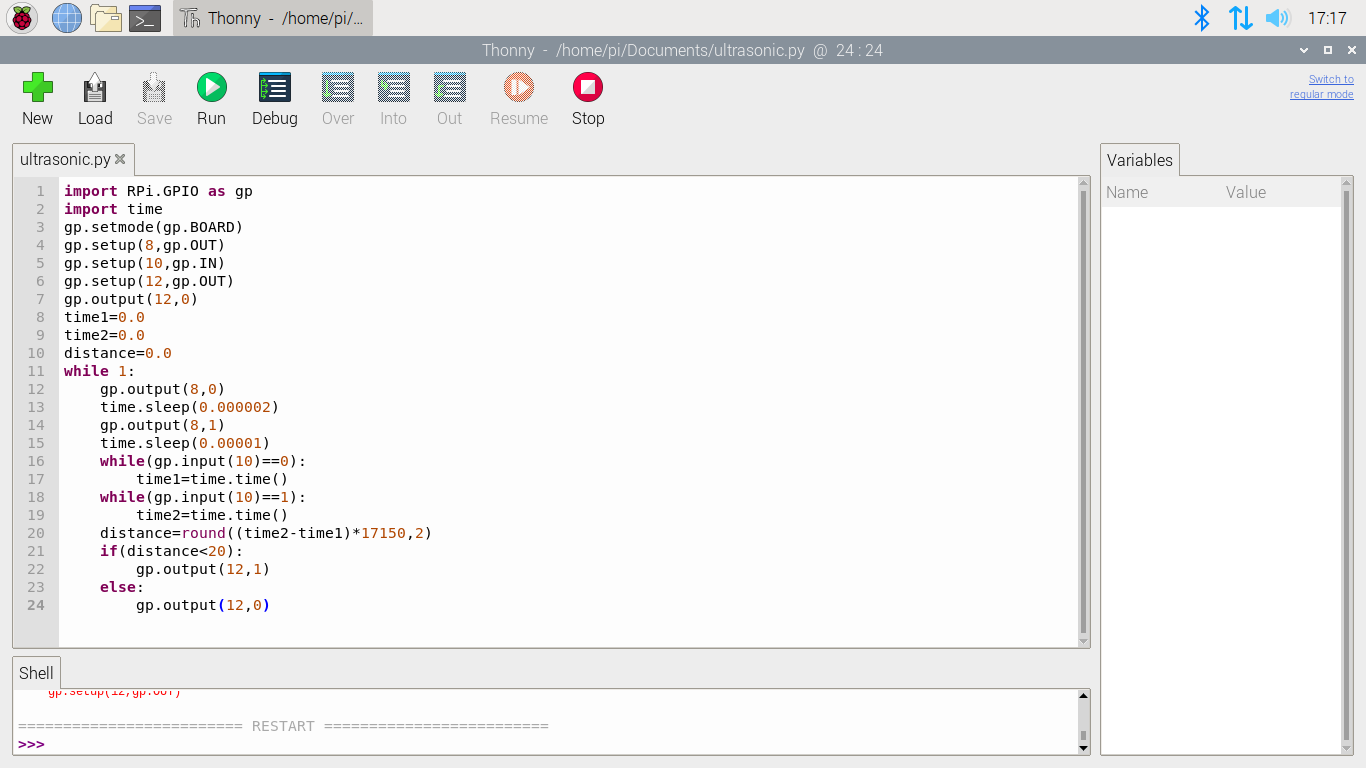
**YELLOW LED Glow when YELLOW button is selected**



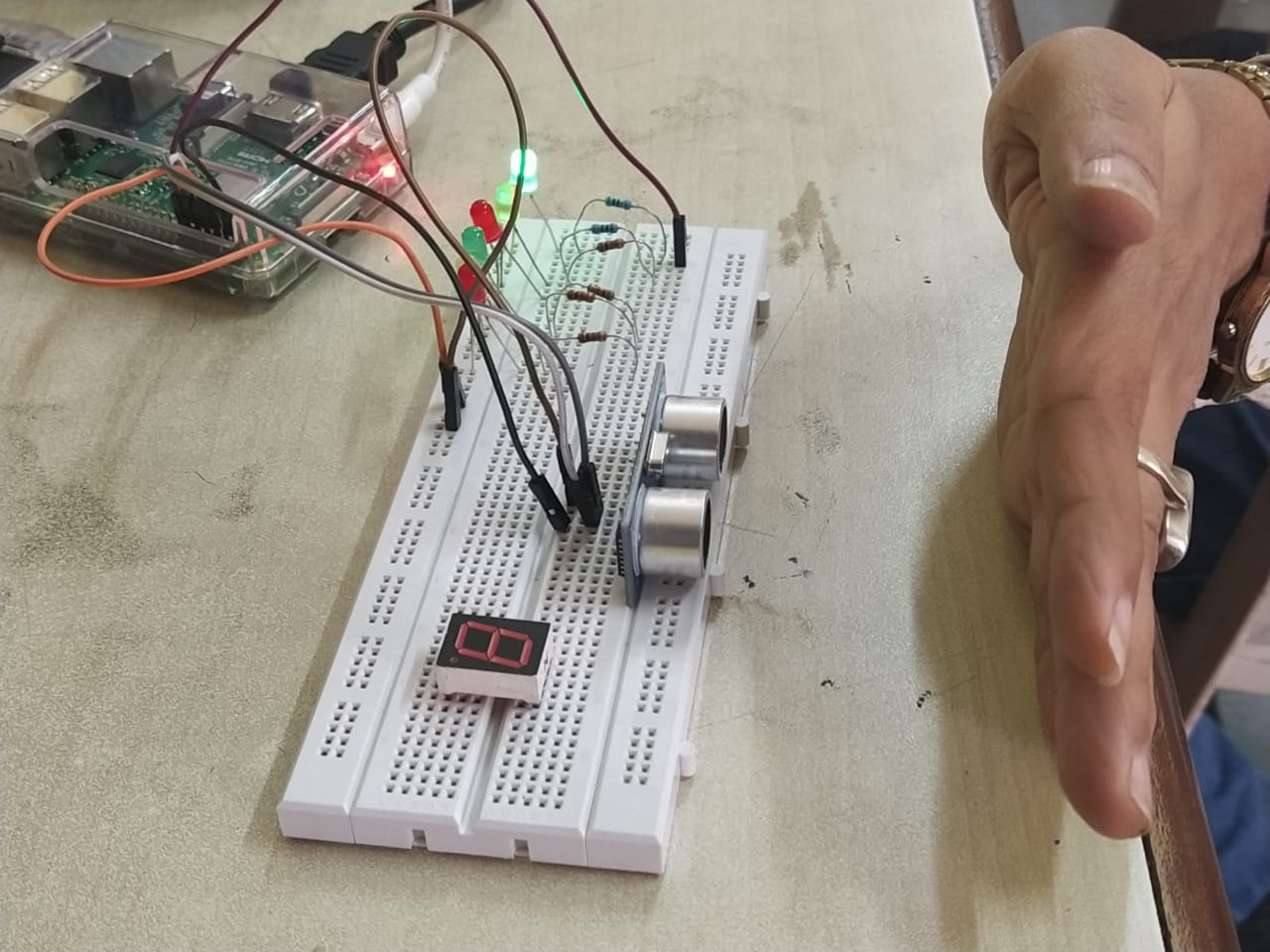
**GREEN LED Glow when GREEN button is selected**

**Ultrasonic Sensor Reading using Raspberry Pi**

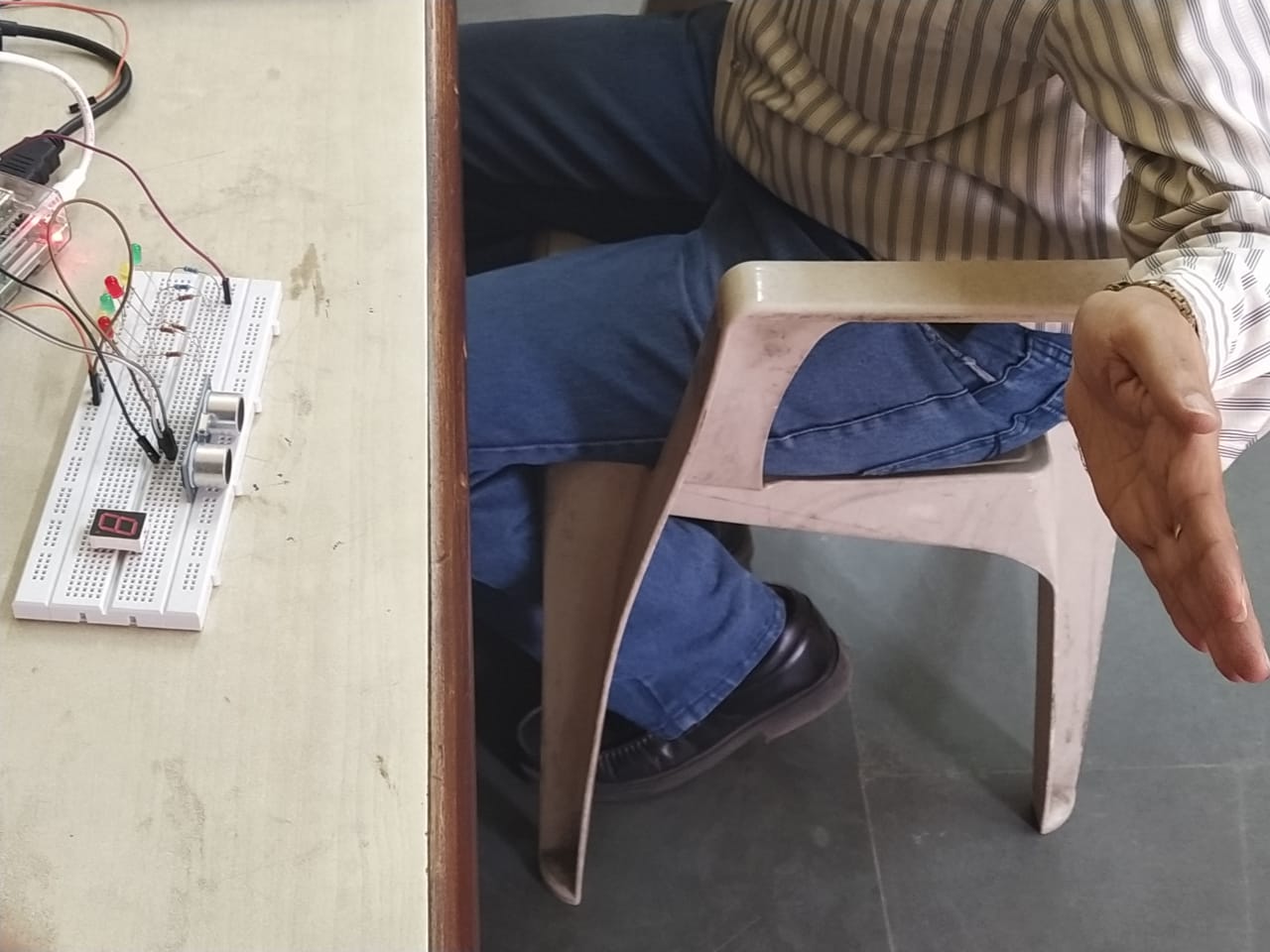
**Description:** This project is to teach the students to interface Ultrasonic sensor with Raspberry Pi.



**Hardware Setup and Execution Image**



**Green LED is glowing when the hand distance is less than 20c.m.**



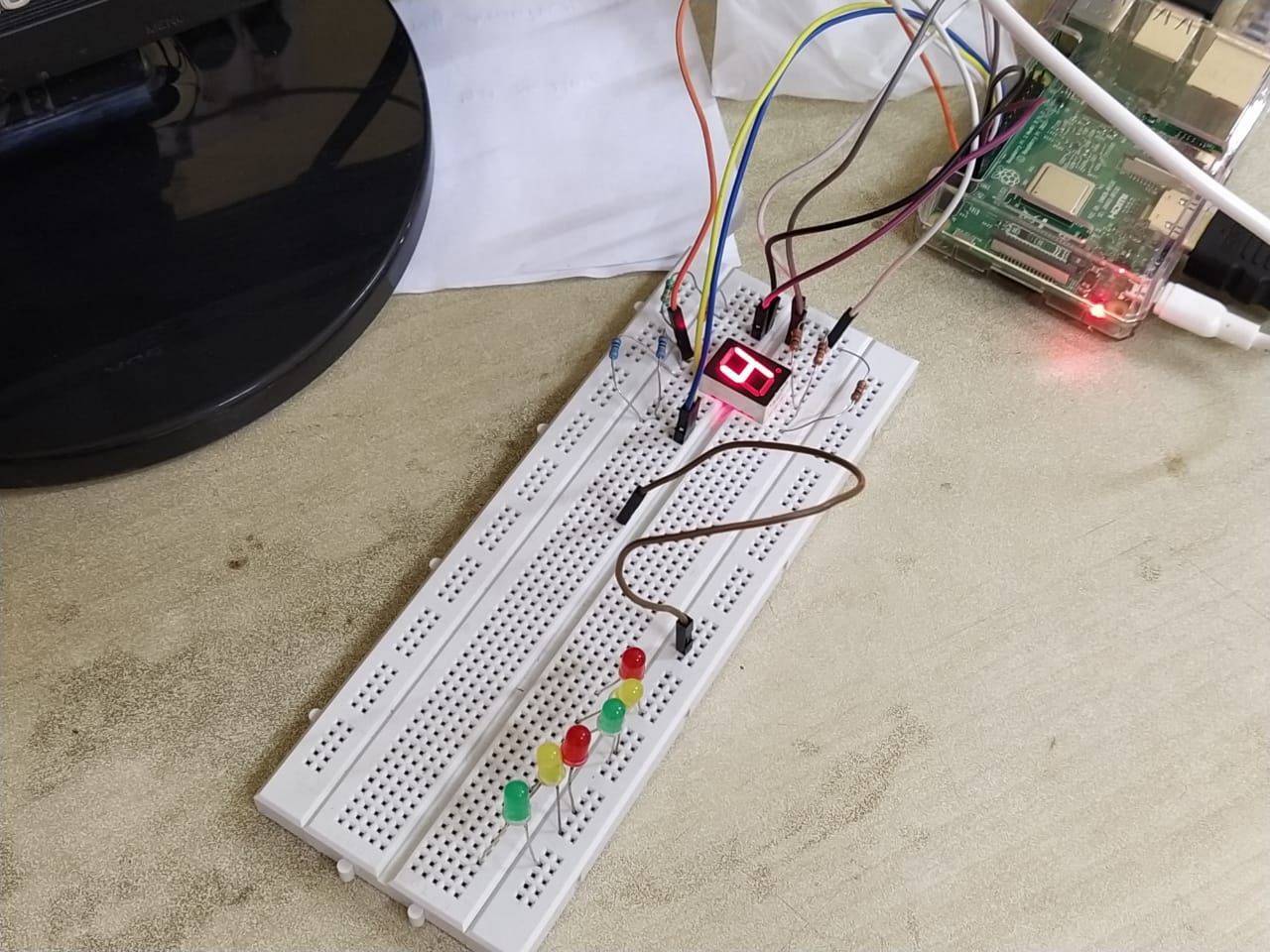
**Green LED is off when the hand distance is greater than 20c.m.**

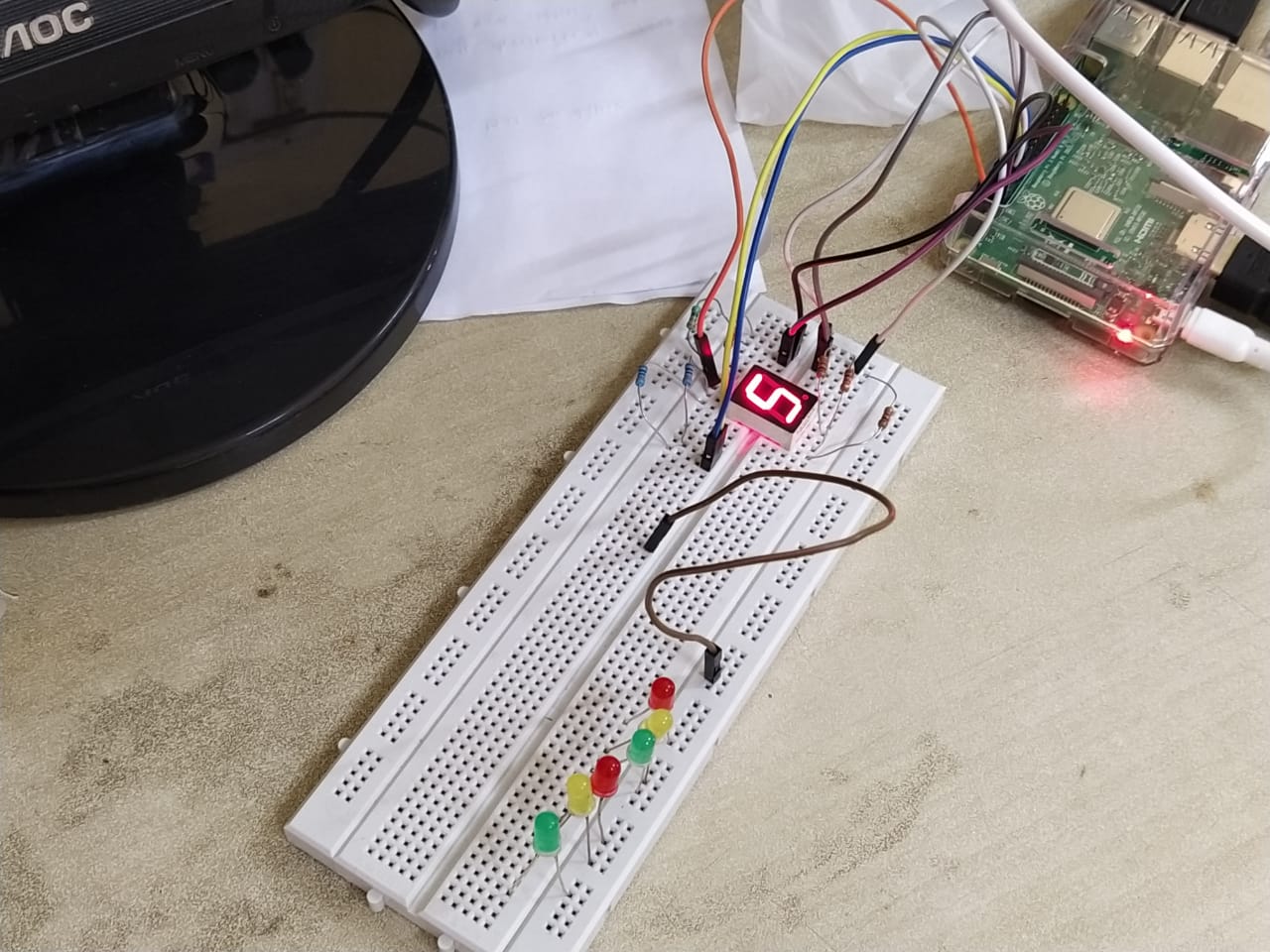
**Single Digit Counter Design with 7-segment LED using Raspberry Pi**

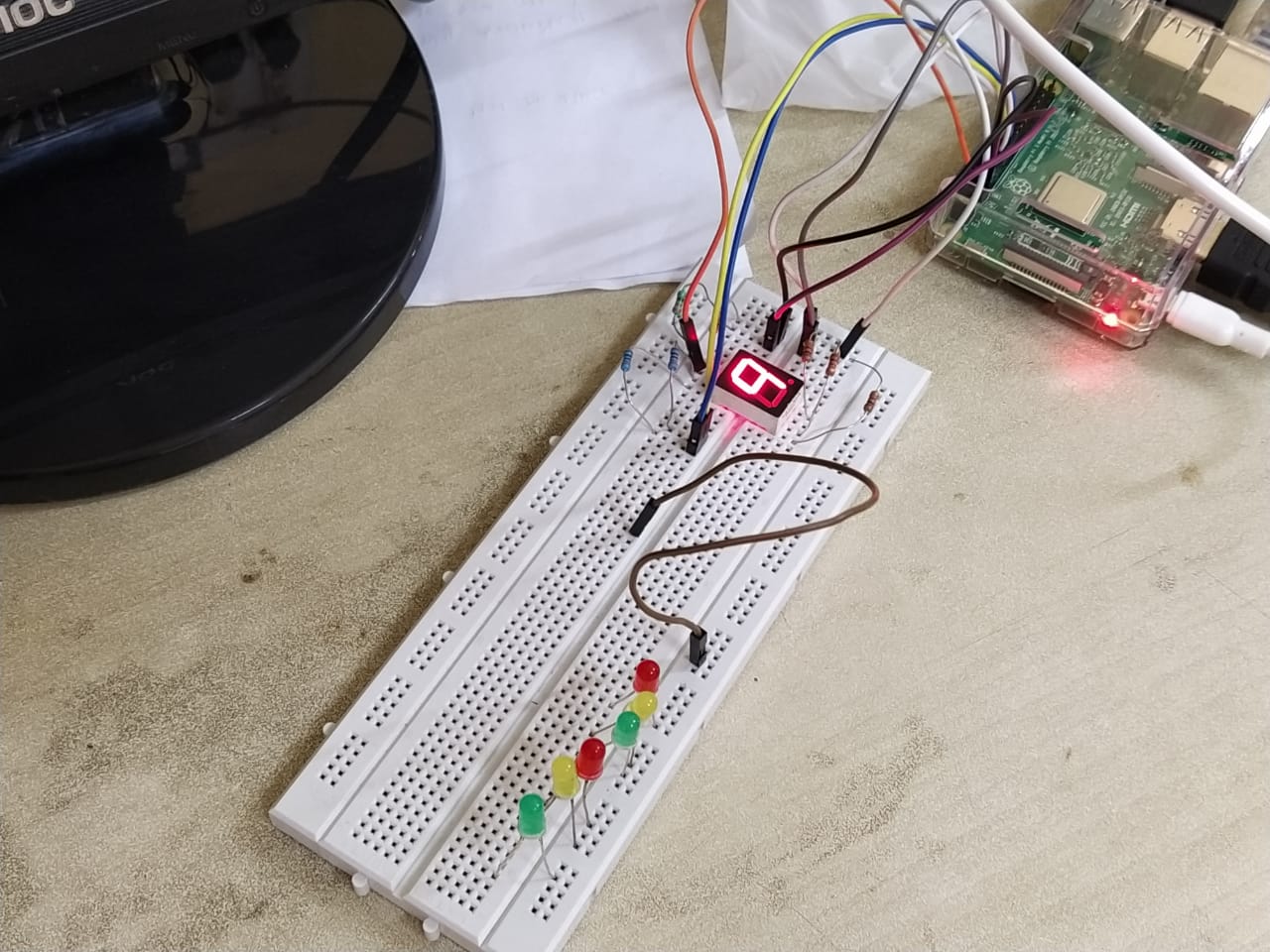
**Description:** This project is to train students to interface 7 segment display with Raspberry Pi and design a single digit counter using 7 segment display.



**Hardware Setup and Execution Image**







**Bit Rotation Program using Raspberry Pi**

**Raspberry Pi Code**

**import RPi.GPIO as gp**

**from tkinter import \***

**gp.setmode(gp.BOARD)**

**gp.setup(8,gp.OUT)**

**gp.setup(10,gp.OUT)**

**gp.setup(12,gp.OUT)**

**gp.output(8,1)**

**gp.output(10,0)**

**gp.output(12,1)**

**xy=[0 for i in range(3)]**

**xy[0]=1**

**xy[1]=0**

**xy[2]=1**

**def runcode():**

**a=x.get()**

**tmp=0**

**if(a==1):**

**tmp=xy[2]**

**for i in range(2):**

**xy[2-i]=xy[1-i]**

**xy[0]=tmp**

**elif(a==2):**

**tmp=xy[0]**

**for i in range(2):**

**xy[i]=xy[i+1]**

**xy[2]=tmp**

**gp.output(8,xy[0])**

**gp.output(10,xy[1])**

**gp.output(12,xy[2])**

**tk=Tk()**

**tk.title('Rotator')**

**x=IntVar()**

**rdb1=Radiobutton(tk,text='Rotate Left',variable=x,value=1)**

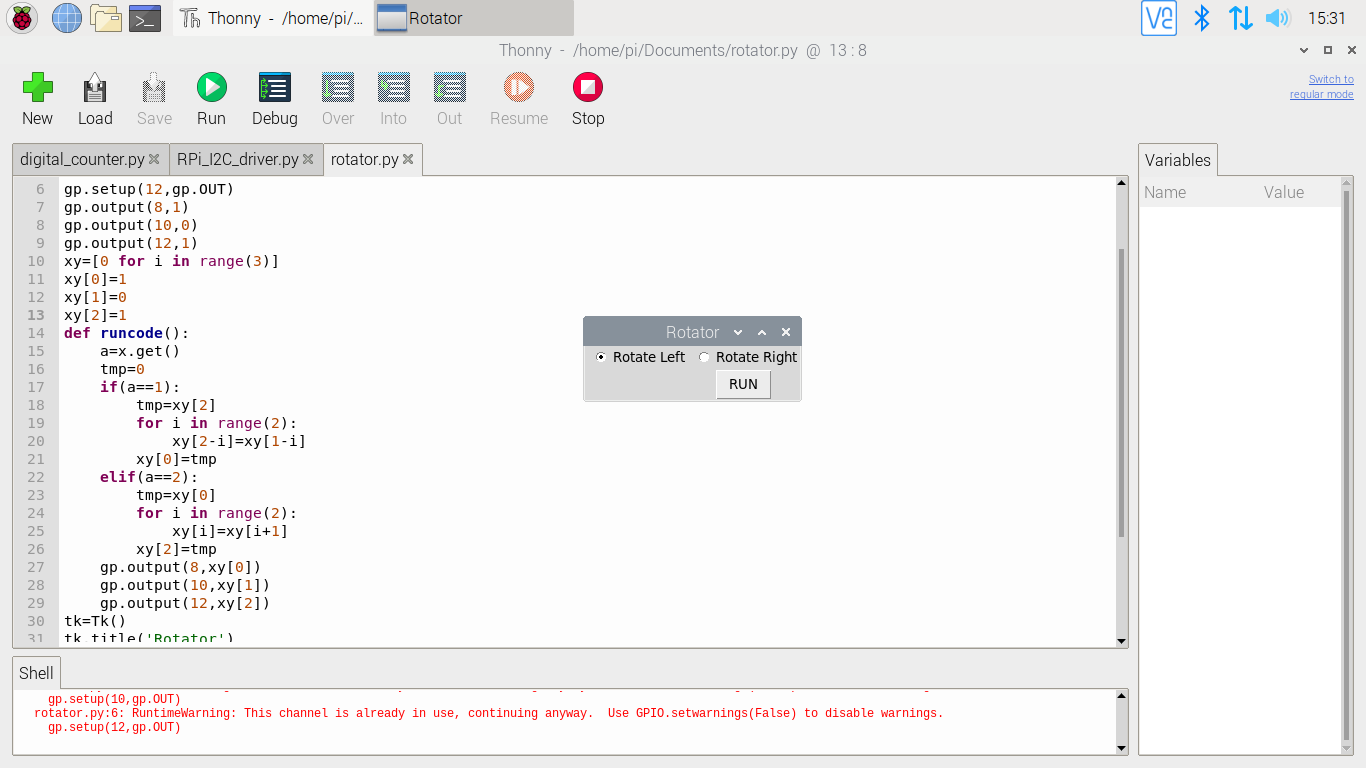
**rdb1.grid(row=0,column=0)**

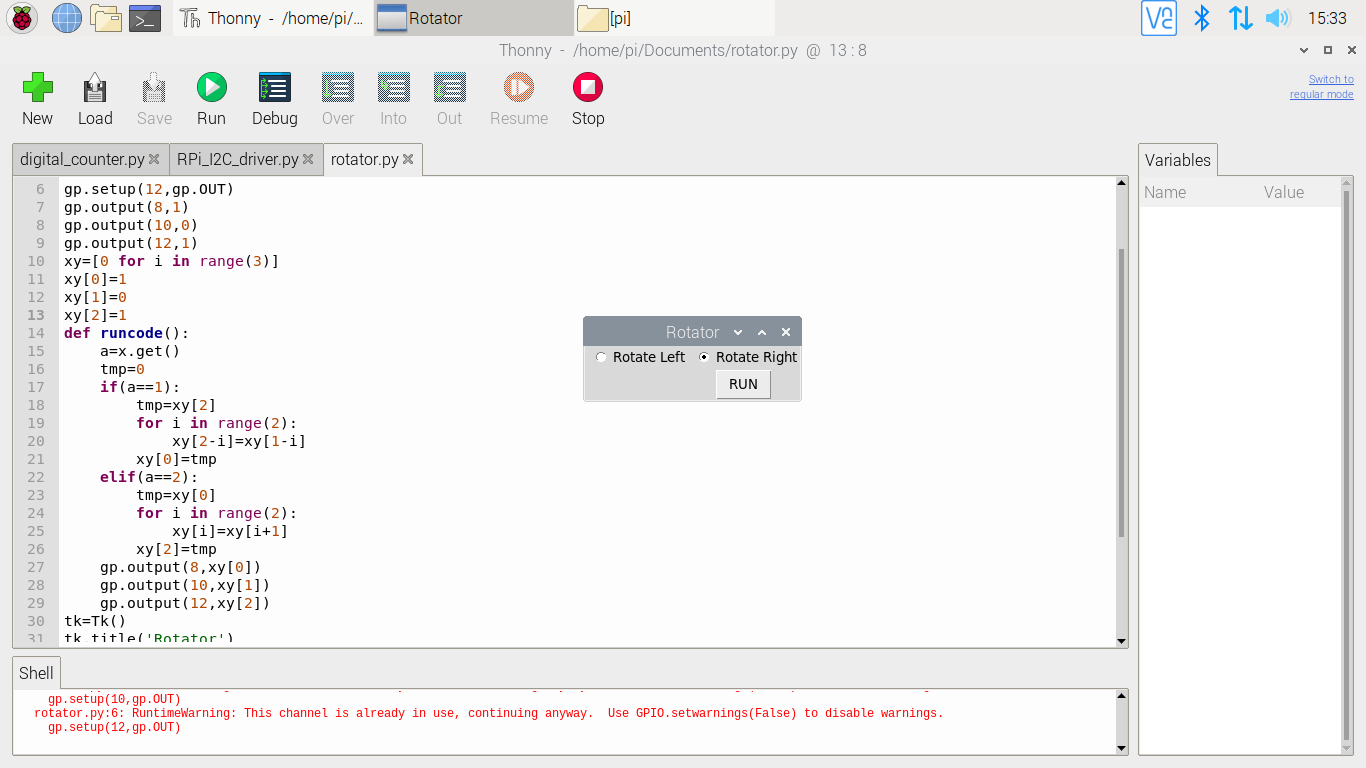
**rdb2=Radiobutton(tk,text='Rotate Right',variable=x,value=2)**

**rdb2.grid(row=0,column=1)**

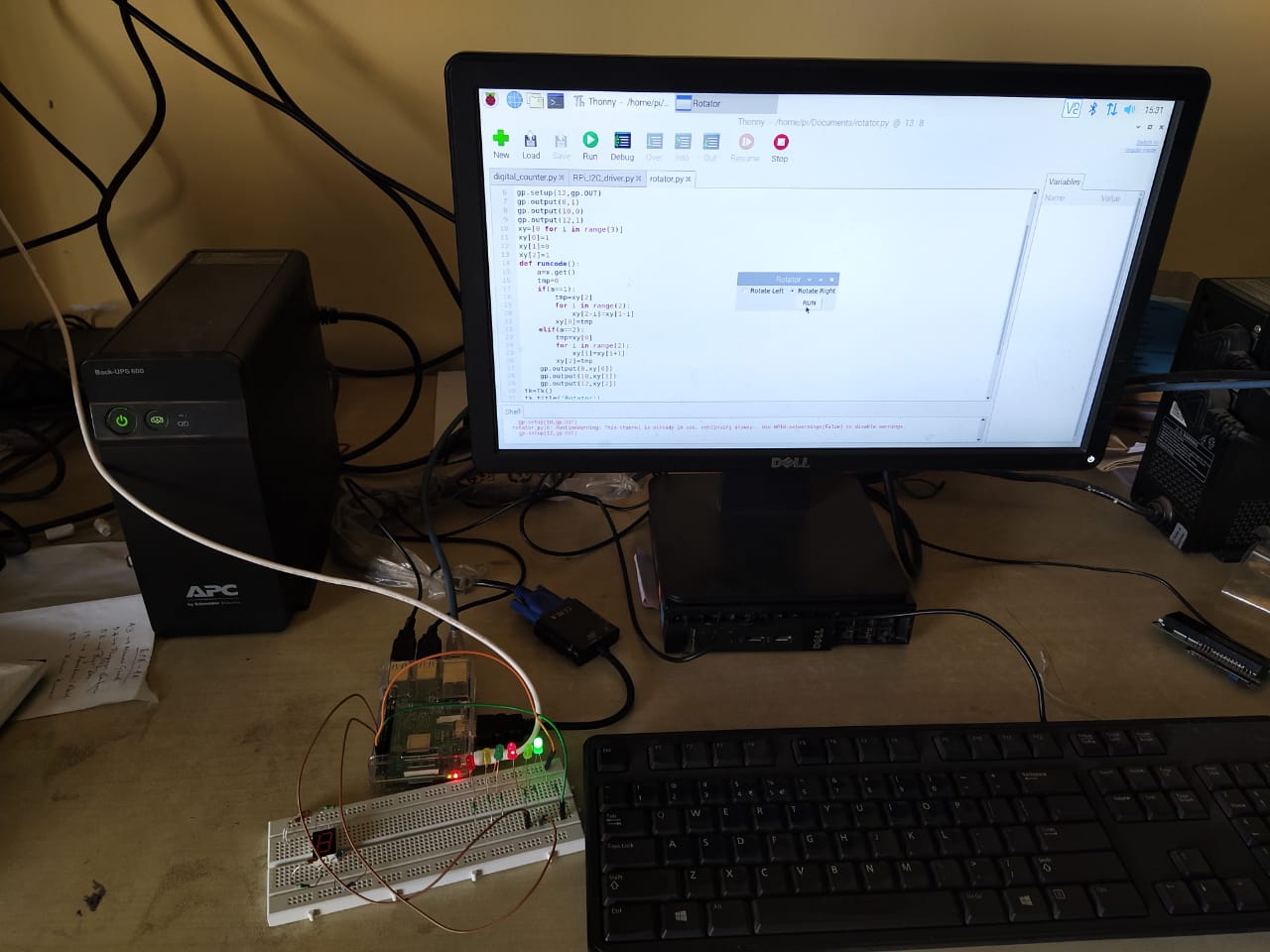
**Button(tk,text='RUN',command=runcode).grid(row=1,column=1)**

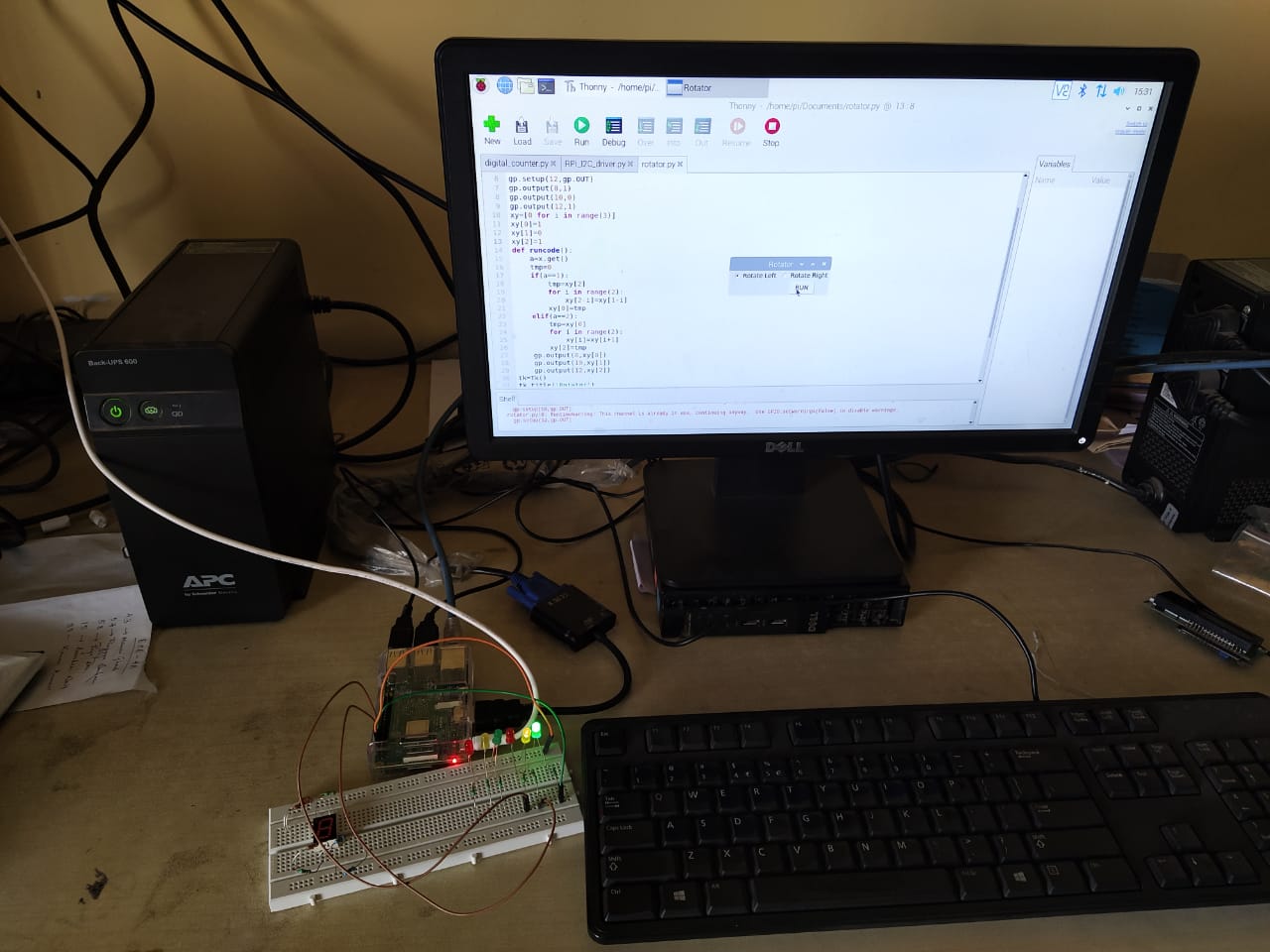
**tk.mainloop()**

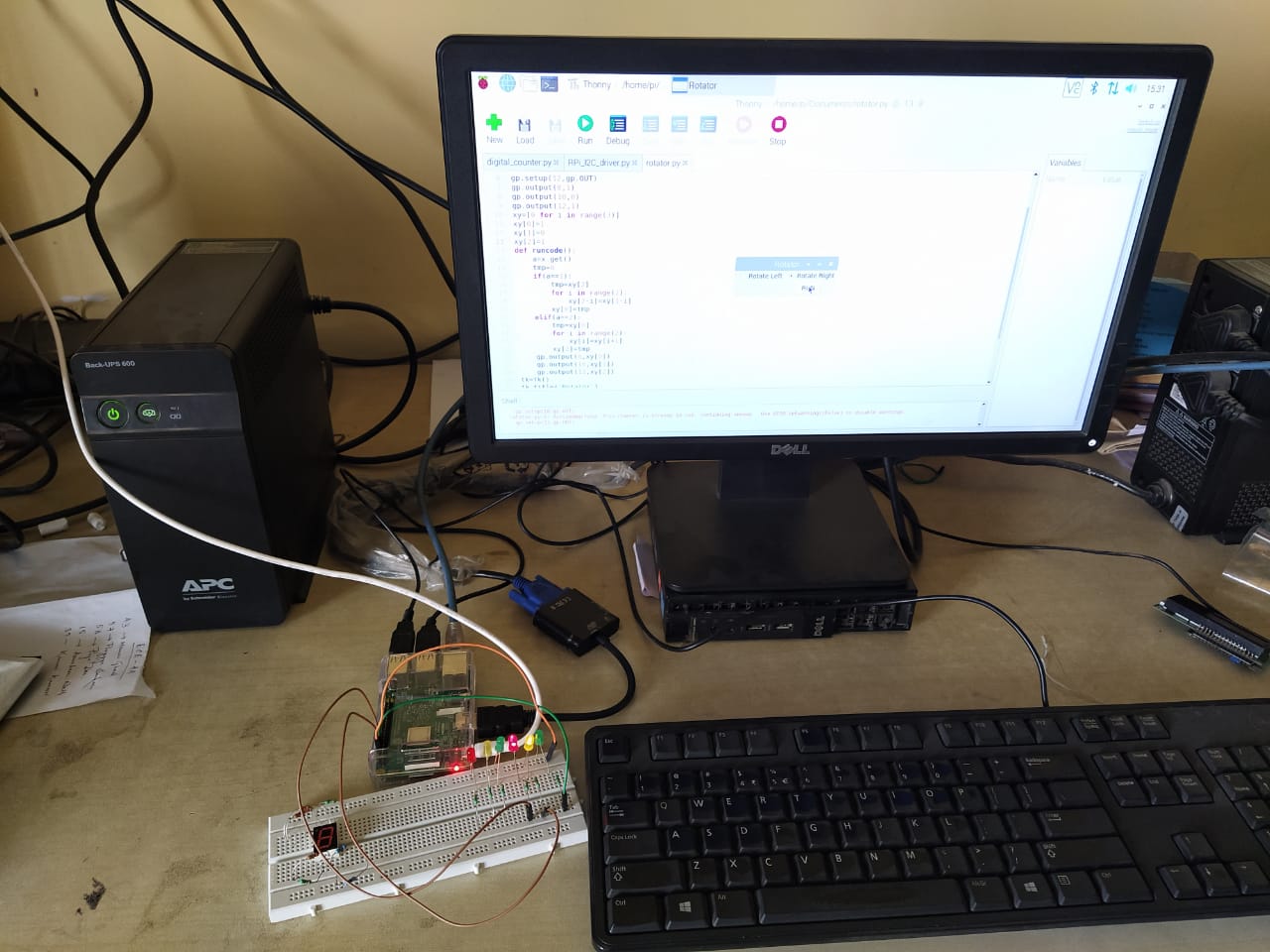


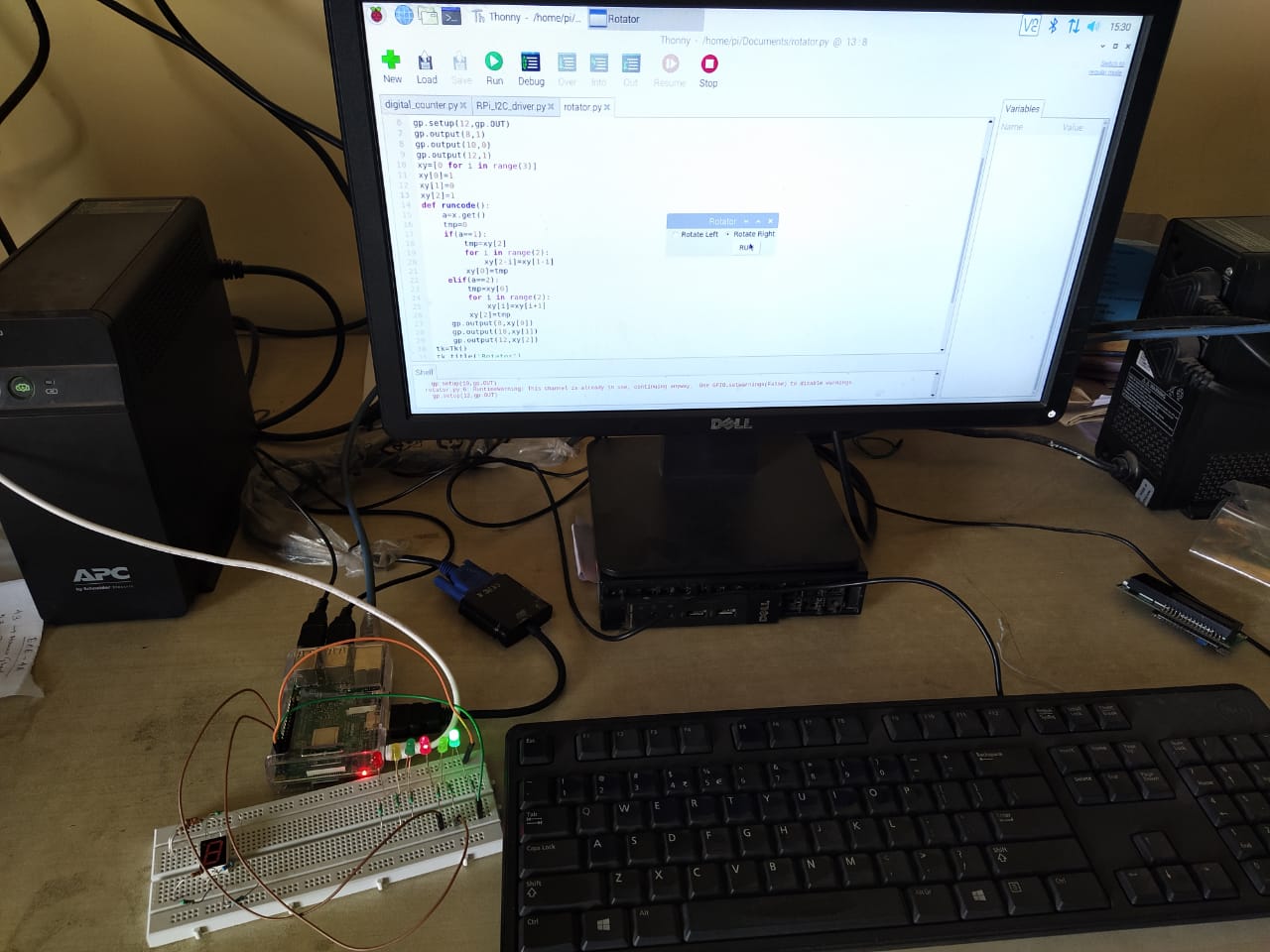


Raspberry Pi programming window









Raspberry Pi Hardware Setup

**Electronic Message Display for Visitors**

**Raspberry Pi Code:**

**import RPi\_I2C\_driver**

**import RPi.GPIO as gp**

**import time**

**gp.setmode(gp.BOARD)**

**gp.setup(11,gp.OUT)**

**gp.setup(13,gp.IN)**

**a=RPi\_I2C\_driver.lcd()**

**time1=0.0**

**time2=0.0**

**distance=0.0**

**while 1:**

**gp.output(11,0)**

**time.sleep(0.000002)**

**gp.output(11,1)**

**time.sleep(0.00001)**

**while (gp.input(13)==0):**

**time1=time.time()**

**while (gp.input(13)==1):**

**time2=time.time()**

**distance=round((time2-time1)\*17150,1)**

**if(distance<=20):**

**a.lcd\_display\_string('Welcome to JIS!',1)**

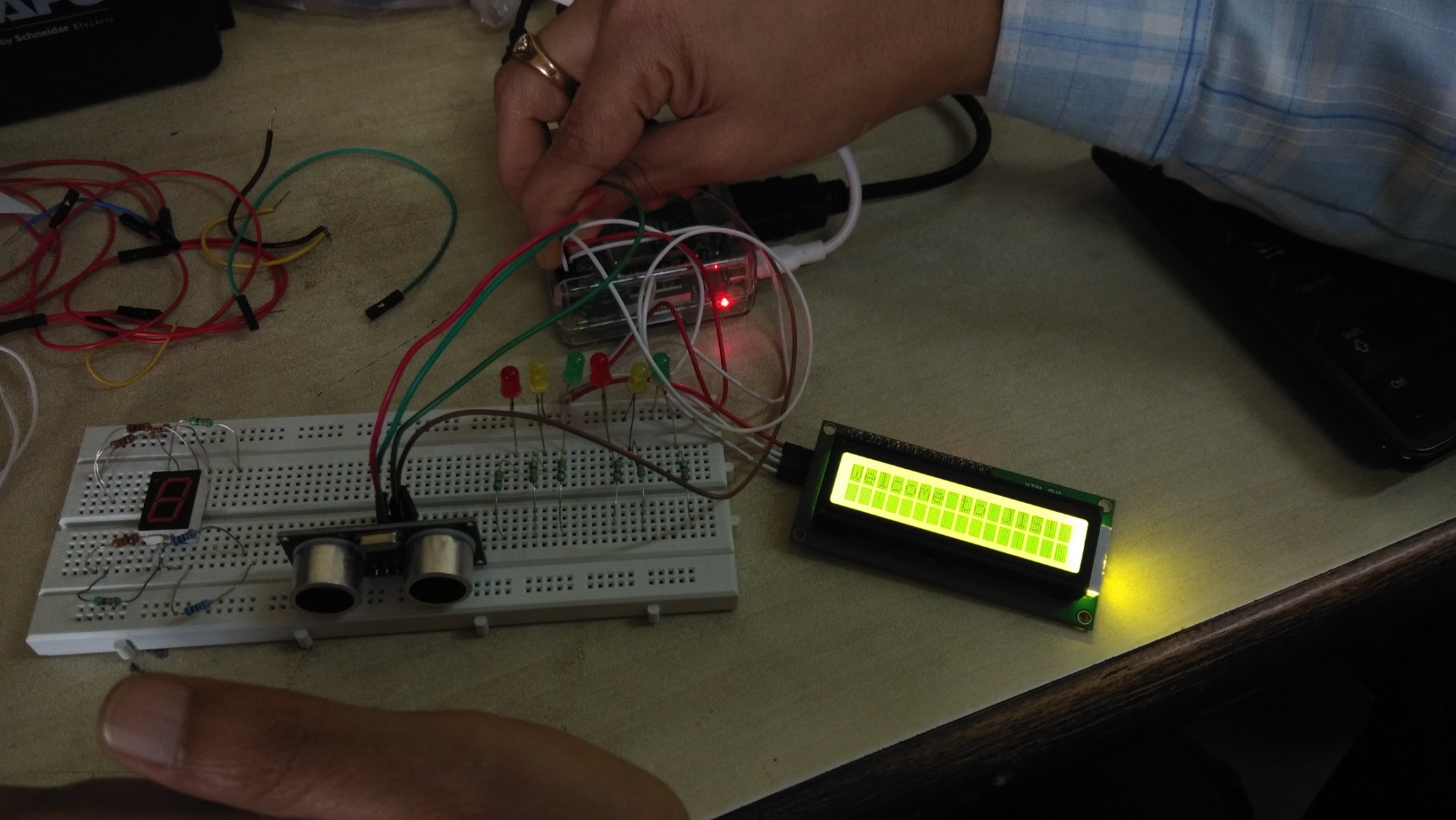
**a.lcd\_display\_string(' ',2)**

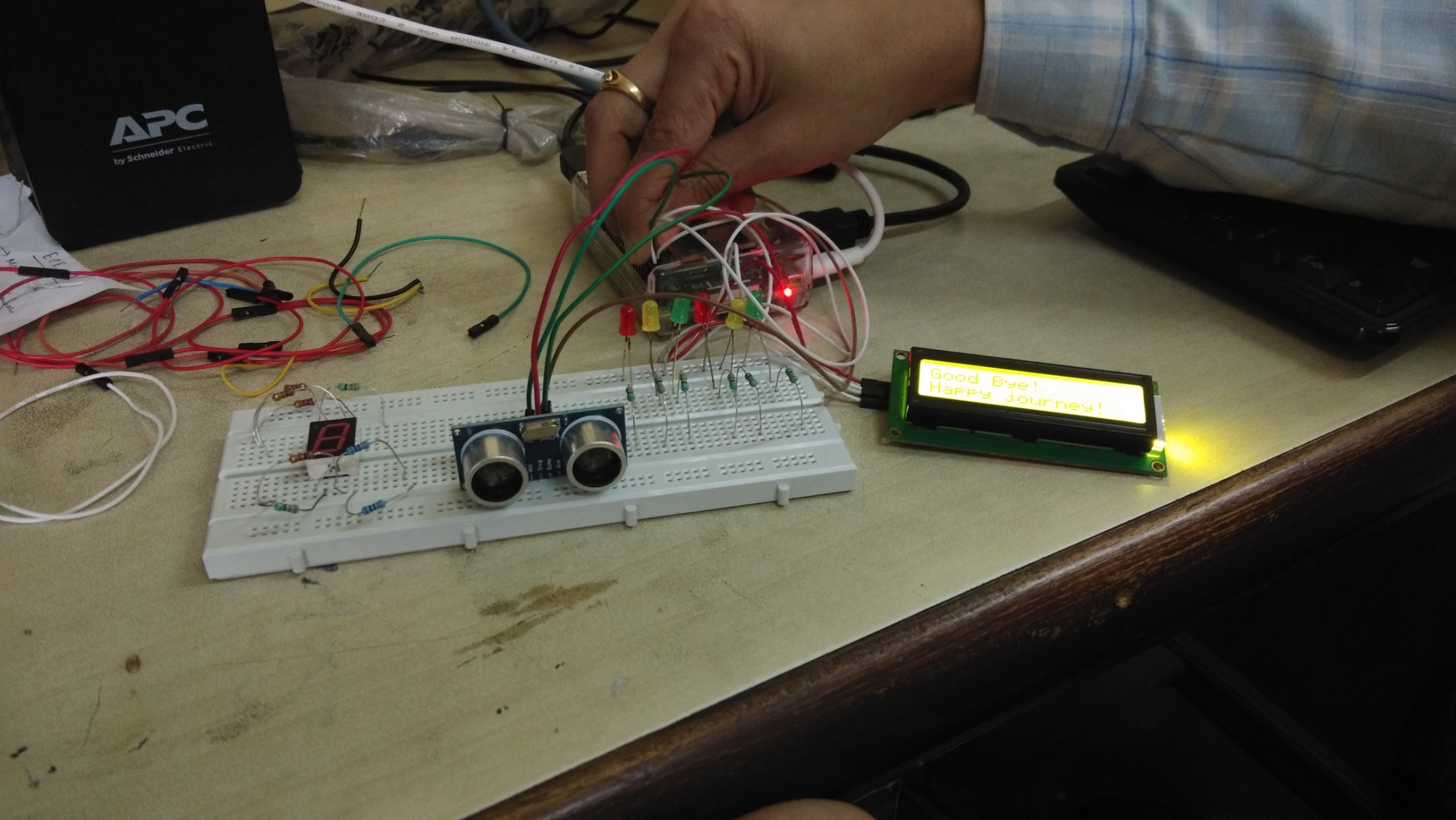
**else:**

**a.lcd\_display\_string('Good Bye! ',1)**

**a.lcd\_display\_string('Happy journey! ',2)**

**Hardware Setup**





(Check the Video <https://youtu.be/ID8TGyQNGWE>)

**Desktop Application based Faculty Name and Age Display System using Raspberry Pi**

**Raspberry Pi Code**

**import RPi\_I2C\_driver**

**from tkinter import \***

**from tkinter import ttk**

**import time**

**a=RPi\_I2C\_driver.lcd()**

**b=open('faculty\_name\_age','r')**

**fc=b.read().splitlines()**

**data={}**

**for value in fc:**

**name,age=value.split(',')**

**data[name]=age**

**print(data)**

**b.close()**

**nm=[]**

**ag=[]**

**for i in data.keys():**

**nm.append(i)**

**for i in data.values():**

**ag.append(i)**

**def showage():**

**a.lcd\_clear()**

**x=cmb.get()**

**tmp=0**

**for i in range(len(nm)):**

**if(x==nm[i]):**

**tmp=i**

**lbl.config(text=ag[tmp])**

**a.lcd\_display\_string(nm[tmp],1)**

**a.lcd\_display\_string('Age : '+ag[tmp],2)**

**tk=Tk()**

**tk.title('Staff Name and Age')**

**Label(tk,text='Select Faculty Name').grid(row=0,column=0)**

**cmb=ttk.Combobox(tk,values=nm)**

**cmb.grid(row=0,column=1)**

**Button(tk,text='Select Name',command=showage).grid(row=1,column=0)**

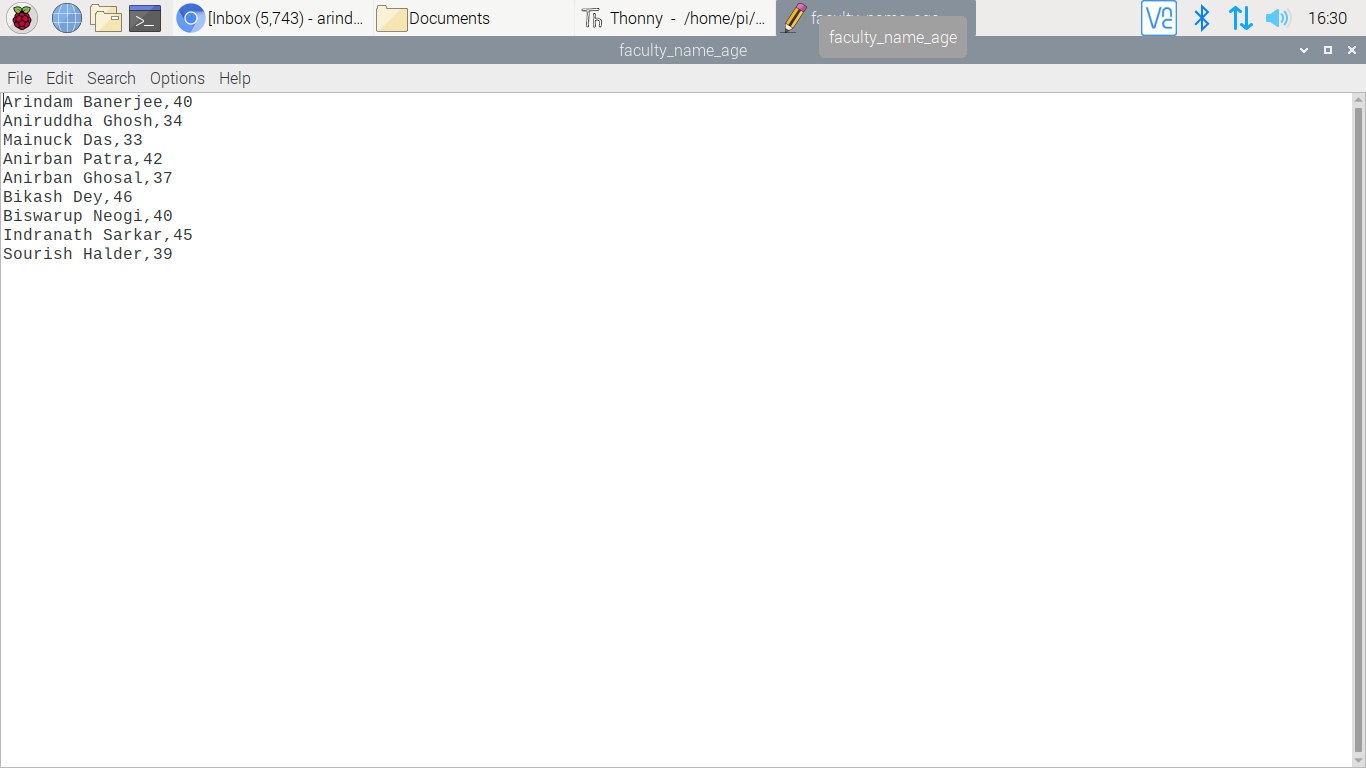
**Label(tk,text='Your age is:').grid(row=2,column=0)**

**lbl=Label(tk)**

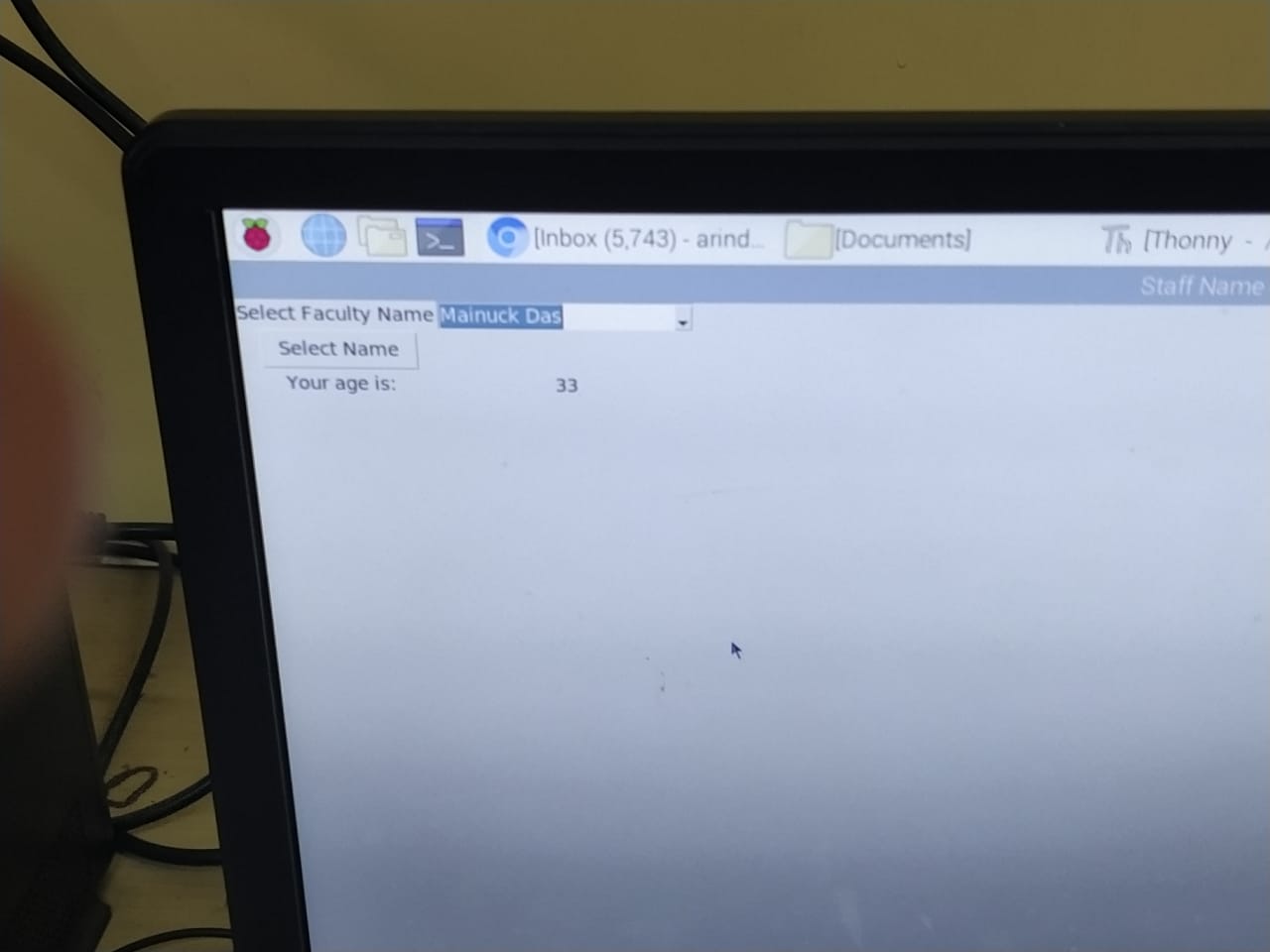
**lbl.grid(row=2,column=1)**

**tk.mainloop()**

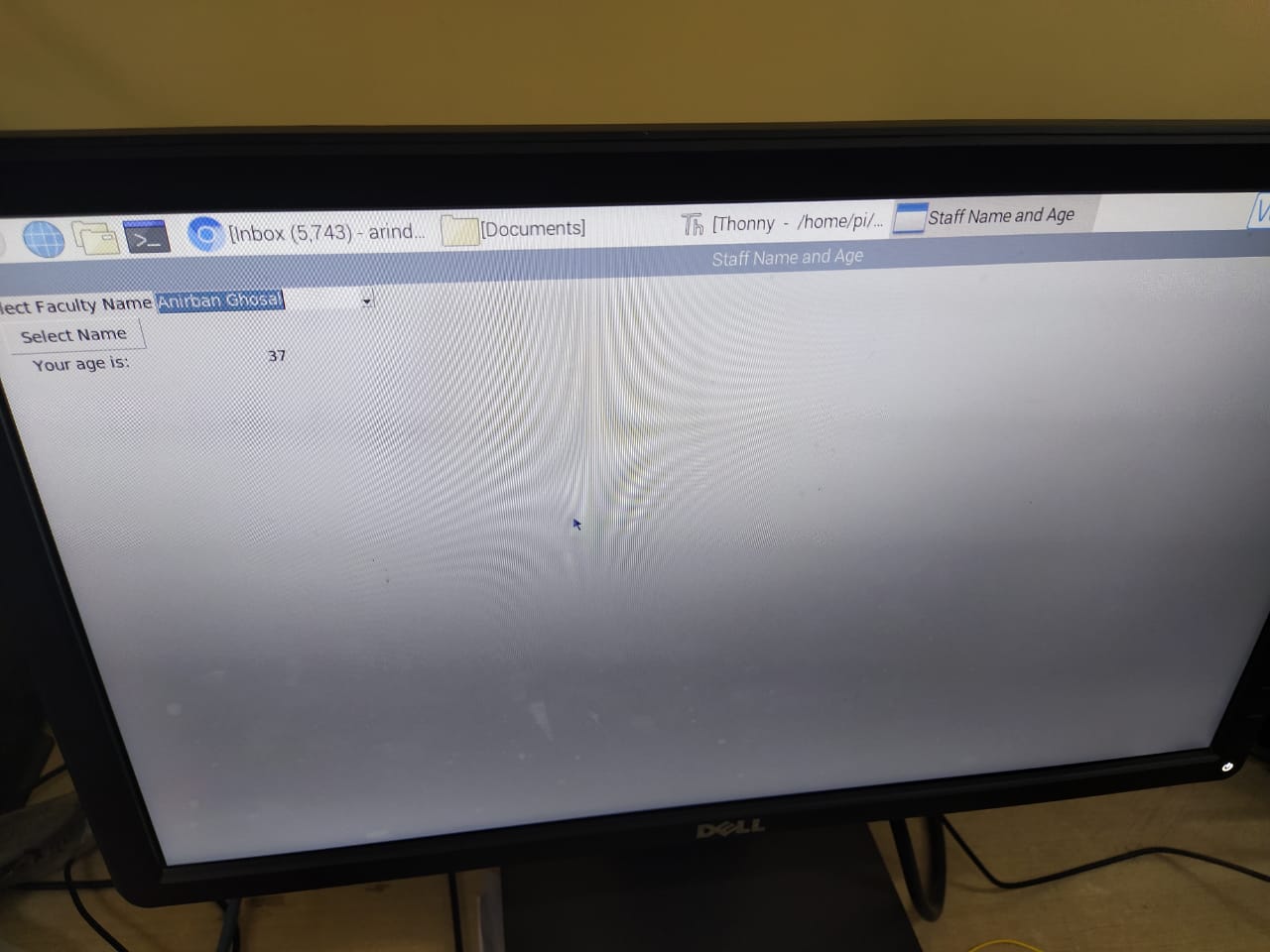
**Program window and hardware setup for the faculty details in Raspbian**



**Text Document for faculty name and age**

****

****

****

