1.PROGRAM

#converting voice message to text form

import speech\_recognition as sr

# Record Audio

r = sr.Recognizer()

with sr.Microphone() as source:

r.adjust\_for\_ambient\_noise(source,duration=5)

r.dynamic\_energy\_threshold = True

print("say something")

audio = r.listen(source)

# Speech recognition using Google Speech Recognition

try:

# for testing purposes, we're just using the default API key

# to use another API key, use `r.recognize\_google(audio, key="GOOGLE\_SPEECH\_RECOGNITION\_API\_KEY")`

# instead of `r.recognize\_google(audio)`

print("You said: " + r.recognize\_google(audio))

except sr.UnknownValueError:

print("Google Speech Recognition could not understand audio")

except sr.RequestError as e:

print("Could not request results from Google Speech Recognition service; {0}".format(e))

# out put

ALSA lib pcm.c:2495:(snd\_pcm\_open\_noupdate) Unknown PCM cards.pcm.rear

ALSA lib pcm.c:2495:(snd\_pcm\_open\_noupdate) Unknown PCM cards.pcm.center\_lfe

ALSA lib pcm.c:2495:(snd\_pcm\_open\_noupdate) Unknown PCM cards.pcm.side

ALSA lib pcm\_route.c:867:(find\_matching\_chmap) Found no matching channel map

ALSA lib pcm\_route.c:867:(find\_matching\_chmap) Found no matching channel map

ALSA lib pcm\_route.c:867:(find\_matching\_chmap) Found no matching channel map

ALSA lib pcm\_route.c:867:(find\_matching\_chmap) Found no matching channel map

say something

You said: hello

**2.PROGRAM FOR ULTRASONIC SENSOR:**

|  |
| --- |
| import RPi.GPIO as GPIO |

|  |
| --- |
| import time |

|  |
| --- |
| import os |

|  |
| --- |
|  |

|  |
| --- |
| GPIO.setmode(GPIO.BCM) |

|  |
| --- |
|  |

|  |
| --- |
| GPIO\_TRIGGER = 18 |

|  |
| --- |
| GPIO\_ECHO = 17 |

|  |
| --- |
|  |

|  |
| --- |
| GPIO.setup(GPIO\_TRIGGER, GPIO.OUT) |

|  |
| --- |
| GPIO.setup(GPIO\_ECHO, GPIO.IN) |

|  |
| --- |
|  |

|  |
| --- |
| def distance(): |

|  |
| --- |
| GPIO.output(GPIO\_TRIGGER, True) |

|  |
| --- |
|  |

|  |
| --- |
| # set Trigger after 0.01ms to LOW |

|  |
| --- |
| time.sleep(0.00001) |

|  |
| --- |
| GPIO.output(GPIO\_TRIGGER, False) |

|  |
| --- |
|  |

|  |
| --- |
| StartTime = time.time() |

|  |
| --- |
| StopTime = time.time() |

|  |
| --- |
|  |

|  |
| --- |
| while GPIO.input(GPIO\_ECHO) == 0: |

|  |
| --- |
| StartTime = time.time() |

|  |
| --- |
|  |

|  |
| --- |
| # save time of arrival |

|  |
| --- |
| while GPIO.input(GPIO\_ECHO) == 1: |

|  |
| --- |
| StopTime = time.time() |

|  |
| --- |
|  |

|  |
| --- |
| # time difference between start and arrival |

|  |
| --- |
| TimeElapsed = StopTime - StartTime |

|  |
| --- |
| # multiply with the sonic speed (34300 cm/s) |

|  |
| --- |
| # and divide by 2, because there and back |

|  |
| --- |
| distance = round((TimeElapsed \* 34300) / 2,2) |

|  |
| --- |
|  |

|  |
| --- |
| return distance |

|  |
| --- |
|  |

|  |
| --- |
| if \_\_name\_\_ == '\_\_main\_\_': |

|  |
| --- |
| try: |

|  |
| --- |
| while True: |

|  |
| --- |
| dist = distance() |

|  |
| --- |
| print ("Measured Distance = %.1f cm" % dist) |

|  |
| --- |
| time.sleep(5) |

|  |
| --- |
| if(dist<=50): |

|  |
| --- |
| print("object is near by") |

|  |
| --- |
| os.system("fswebcam -F 4 --fps 20 -r 800\*600 /home/pi/vvv/"+str(dist)+".jpg") |

|  |
| --- |
| print("pic Taken") |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
| # Reset by pressing CTRL + C |

|  |
| --- |
| except KeyboardInterrupt: |

|  |
| --- |
| print("Measurement stopped by User") |

|  |
| --- |
| GPIO.cleanup() |

**OUTPUT:**

distance: 56cm

distance: 12cm

distance: 10cm

distance: 23cm

3.PROGRAM

import matplotlib.pyplot as plt

x=[1,2,3,4,5]

y=[-3,-6,-9,12,15]

plt.plot(x,y)

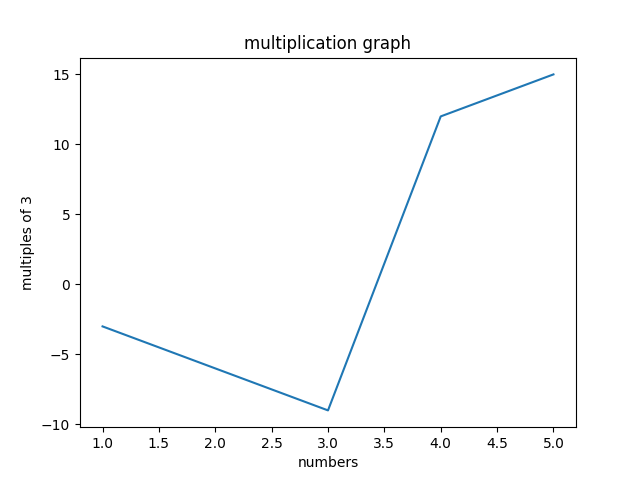
plt.xlabel("numbers")

plt.ylabel("multiples of 3")

plt.title("multiplication graph")

plt.show()

out put



input

import matplotlib.pyplot as plt

#x-cordinates of left side of bars

left=[1,2,3,4,5,6,7]

#heights of bars

height =[5,10,15,20,25,30,35]

#labels for bars

tick\_label =['A','B','C','D','E','F','G']

#plotting a bar chart

plt.bar(left, height, tick\_label =tick\_label,width=0.7,color=['blue','red'])

#naming the x-axis

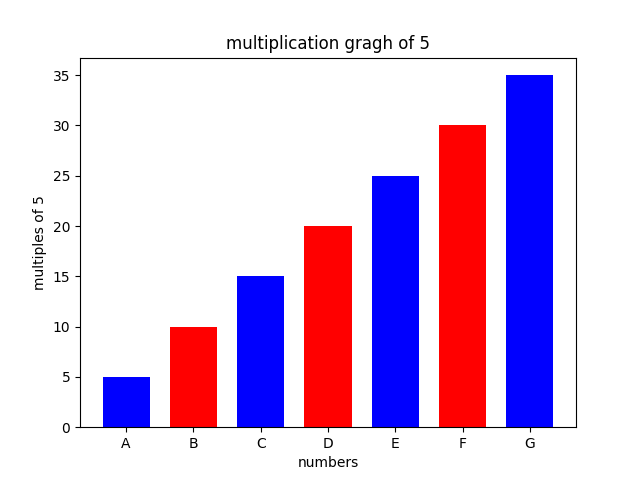
plt.xlabel("numbers")

plt.ylabel("multiples of 5")

plt.title("multiplication gragh of 5")

plt.show()

output



import matplotlib.pyplot as s

left = [15,25,35,45]

height = [10,50,90,10]

tick\_label = ['2000', '2005', '2010', '2015']

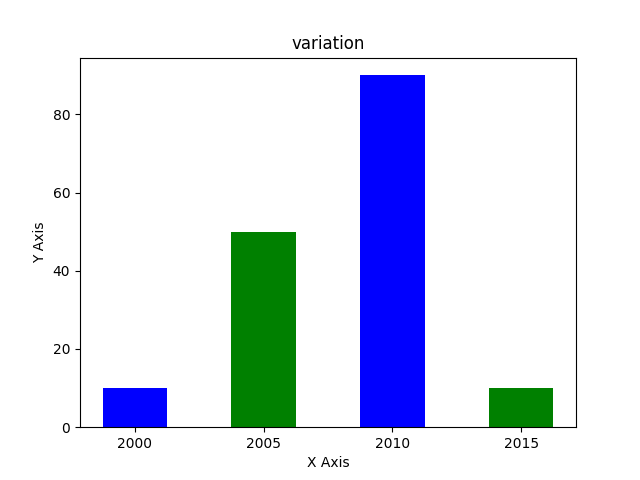
s.bar(left, height,tick\_label = tick\_label,width=5,color = ['blue','green'])

s.xlabel("X Axis")

s.ylabel("Y Axis")

s.title("variation")

s.show()



import matplotlib.pyplot as plt

x=[1,2,3]

y=[5,9,2]

plt.plot(x,y)

plt.xlabel("X Axis")

plt.ylabel("Y Axis")

plt.title("graphs")

plt.show()

