ULTRASONIC SENSOR-CODE USED:

#!/usr/bin/python

# Import required Python libraries

import RPi.GPIO as GPIO

import time

# We will be using the BCM GPIO numbering

GPIO.setmode(GPIO.BCM)

# Select which GPIOs you will use

GPIO\_BUZZER = 18

GPIO\_TRIGGER = 23

GPIO\_ECHO = 22

# Set BUZZER to OUTPUT mode

GPIO.setup(GPIO\_BUZZER, GPIO.OUT)

# Set TRIGGER to OUTPUT mode

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

# Set ECHO to INPUT mode

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

BUZZER=GPIO.PWM(GPIO\_BUZZER,1000)

# Measures the distance between a sensor and an obstacle and returns the measured value

def distance():

  # Send 10 microsecond pulse to TRIGGER

  GPIO.output(GPIO\_TRIGGER, True) # set TRIGGER to HIGH

  time.sleep(0.00001) # wait 10 microseconds

  GPIO.output(GPIO\_TRIGGER, False) # set TRIGGER back to LOW

  # Create variable start and assign it current time

  start = time.time()

  # Create variable stop and assign it current time

  stop = time.time()

  # Refresh start value until the ECHO goes HIGH = until the wave is send

  while GPIO.input(GPIO\_ECHO) == 0:

    start = time.time()

  # Assign the actual time to stop variable until the ECHO goes back from HIGH to LOW = the wave came back

  while GPIO.input(GPIO\_ECHO) == 1:

    stop = time.time()

  # Calculate the time it took the wave to travel there and back

  measuredTime = stop - start

  # Calculate the travel distance by multiplying the measured time by speed of sound

  distanceBothWays = measuredTime \* 33112 # cm/s in 20 degrees Celsius

  # Divide the distance by 2 to get the actual distance from sensor to obstacle

  distance = distanceBothWays / 2

  # Print the distance to see if everything works correctly

  print("Distance : {0:5.1f}cm".format(distance))

  # Return the actual measured distance

  return distance

# Calculates the frequency of beeping depending on the measured distance

def beep\_freq():

  # Measure the distance

  dist = distance()

  # If the distance is bigger than 50cm, we will not beep at all

  if dist > 50:

    return -1

  # If the distance is between 50 and 30 cm, we will beep once a second

  elif dist <= 50 :

    return 1

# Main function

def main():

  try:

    # Repeat till the program is ended by the user

    while True:

      # Get the beeping frequency

      freq = beep\_freq()

      # No beeping

      if freq == -1:

        GPIO.output(GPIO\_BUZZER, False)

        time.sleep(0.5)

      # Constant beeping

      elif freq == 1:

        GPIO.output(GPIO\_BUZZER, True)

        time.sleep(0.5)

        GPIO.output(GPIO\_BUZZER, False)

        time.sleep(freq)

  # If the program is ended, stop beeping and cleanup GPIOs

  except KeyboardInterrupt:

    GPIO.output(GPIO\_BUZZER, False)

    GPIO.cleanup()

# Run the main function when the script is executed

if \_\_name\_\_ == "\_\_main\_\_":

    main()