**testhcsr04.py**

**Page 1**

#Libraries

import RPi.GPIO as GPIO

import time

#set up components

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM) #using broadcom pin system GPIO

TRIG=23

ECHO=24

#Program Logic

#print("distance measurement in progress")

#print("waiting for sensor to settle")

while True:

GPIO.setup(TRIG,GPIO.OUT)

GPIO.setup(ECHO,GPIO.IN)

GPIO.output(TRIG,False)

time.sleep(0.2)

GPIO.output(TRIG,True)

time.sleep(0.00001)

GPIO.output(TRIG,False)

while GPIO.input(ECHO)==0:

pulse\_start=time.time()

#When the trigger pin sends out a pulse, the echo pin becomes high

#this will give us the pulse start time, the time when the echo pin was last low

#then, the Echo pin remains high until the pulse echo is received back

#by the receiver. At this point, the echo pin will drop to low - pulse\_end time

while GPIO.input(ECHO)==1:

pulse\_end=time.time()

pulse\_duration=pulse\_end-pulse\_start

distance=pulse\_duration\*34300

distance = distance / 2

distance=round(distance,2)

print("distance:",distance,"cm")

**Page 2**

**gmailwithapppassword.py**

#https://mail.google.com/mail/u/0/#inbox/KtbxLvhNVNlLqfmgZHbxdhVfpTHlCxQgqB?projector=1

import smtplib

from email.message import EmailMessage

def email\_alert(subject,body,to):

msg=EmailMessage()

msg.set\_content(body)

msg['subject']=subject

msg['to'] = to

user="praspberry060@gmail.com"

password="eajvptvfrrrgqvjh" #app password, not the sign in password

msg['from']=user

server = smtplib.SMTP("smtp.gmail.com",587)

server.starttls()

server.login(user,password)

server.send\_message(msg)

server.quit()

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

email\_alert('Test',"Hello World Again","iamssgoh@gmail.com")

**Page 3**

**hcsr04gmailalert.py (pg 1)**

#Libraries

import RPi.GPIO as GPIO

import time

from time import sleep

from datetime import datetime

import smtplib

from email.message import EmailMessage

#set up components

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM) #using broadcom pin system GPIO

TRIG=23

ECHO=24

#setup function

def email\_alert(subject,body,to):

msg=EmailMessage()

msg.set\_content(body)

msg['subject']=subject

msg['to'] = to

user="praspberry060@gmail.com"

password="eajvptvfrrrgqvjh" #app password, not the sign in password

msg['from']=user

server = smtplib.SMTP("smtp.gmail.com",587)

server.starttls()

server.login(user,password)

server.send\_message(msg)

server.quit()

#Program Logic

alert\_status=False

while True:

GPIO.setup(TRIG,GPIO.OUT)

GPIO.setup(ECHO,GPIO.IN)

GPIO.output(TRIG,False)

sleep(0.2)

GPIO.output(TRIG,True)

sleep(0.00001)

GPIO.output(TRIG,False)

while GPIO.input(ECHO)==0:

pulse\_start=time.time()

#When the trigger pin sends out a pulse, the echo pin becomes high

**hcsr04gmailalert.py (cont’d)**

**Page 4**

#this will give us the pulse start time, the time when the echo pin was last low

#then, the Echo pin remains high until the pulse echo is received back

#by the receiver. At this point, the echo pin will drop to low - pulse\_end time

while GPIO.input(ECHO)==1:

pulse\_end=time.time()

pulse\_duration=pulse\_end-pulse\_start

#distance=speed of sound \* time taken

distance=pulse\_duration\*34300

distance = distance / 2

distance=round(distance,2)

print("distance:",distance,"cm")

if distance <= 10: #BIN ALMOST FULL

print("distance:",distance,"cm")

if alert\_status==False:

alert\_status=True

#SEND ALERT TO MANAGER

subject="Smart Bin No 12345"

sent\_date=datetime.now()

"""

body="Alert sent on : " + str(sent\_date.year) +"-"+ str(sent\_date.month)+"-"+str(sent\_date.day) + "\n"

body += "Time : " + str(sent\_date.hour) + ":" + str(sent\_date.minute) + ":" + str(sent\_date.second) + "\n"

body += "Bin is full. Please clear it. Thanks"

"""

format = "%d-%b-%Y %H:%M:%S"

body="Alert sent on : " + sent\_date.strftime(format) +"\n"

body += "Bin is full. Please clear it. Thanks"

recipient ="iamssgoh@gmail.com"

email\_alert(subject,body,recipient)

else:

alert\_status=False

**Page 5**

**testtwilio.py**

#Install twilio library from Thonny

from twilio.rest import Client

account\_sid="AC8d44cd0f81dfc14a309031e7d70afcc8"

#AC8d44cd0f81dfc14a309031e7d70afcc8

authToken="52e0e15c3dc9b7997f4768f880a83ed7"

#52e0e15c3dc9b7997f4768f880a83ed7

client=Client(account\_sid,authToken)

message=client.messages.create(to="whatsapp:+6591080064",

from\_="whatsapp:+14155238886",

body="Hello World")

**hcsr04whatsappalert.py (pg 1)**

**Page 6**

#https://www.youtube.com/watch?v=Svl\_W81wUYU&t=503

#Libraries

import RPi.GPIO as GPIO

import time

from time import sleep

from datetime import datetime

import smtplib

from twilio.rest import Client

#set up components/system variables

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BCM) #using broadcom pin system GPIO

TRIG=23

ECHO=24

#twilio requires these authentications

account\_sid="AC8d44cd0f81dfc14a309031e7d70afcc8"

authToken="52e0e15c3dc9b7997f4768f880a83ed7"

client=Client(account\_sid,authToken)

#Program Logic

alert\_status=False

while True:

GPIO.setup(TRIG,GPIO.OUT)

GPIO.setup(ECHO,GPIO.IN)

GPIO.output(TRIG,False)

sleep(0.2)

GPIO.output(TRIG,True)

sleep(0.00001)

GPIO.output(TRIG,False)

while GPIO.input(ECHO)==0:

pulse\_start=time.time()

#When the trigger pin sends out a pulse, the echo pin becomes high

#this will give us the pulse start time, the time when the echo pin was last low

#then, the Echo pin remains high until the pulse echo is received back

#by the receiver. At this point, the echo pin will drop to low - pulse\_end time

while GPIO.input(ECHO)==1:

pulse\_end=time.time()

pulse\_duration=pulse\_end-pulse\_start

#distance=speed of sound \* time taken

distance=pulse\_duration\*34300

**hcsr04whatsappalert.py (cont’d)**

**Page 7**

distance = distance / 2

distance=round(distance,2)

print("distance:",distance,"cm")

if distance <= 10: #BIN ALMOST FULL

print("distance:",distance,"cm")

if alert\_status==False:

alert\_status=True

#SEND ALERT TO MANAGER

subject="Smart Bin No 12345"

sent\_date=datetime.now()

"""

body="Alert sent on : " + str(sent\_date.year) +"-"+ str(sent\_date.month)+"-"+str(sent\_date.day) + "\n"

body += "Time : " + str(sent\_date.hour) + ":" + str(sent\_date.minute) + ":" + str(sent\_date.second) + "\n"

body += "Bin is full. Please clear it. Thanks"

"""

format = "%d-%b-%Y %H:%M:%S"

body="Alert sent on : " + sent\_date.strftime(format) +"\n"

body += "Bin is full. Please clear it. Thanks"

recipient ="iamssgoh@gmail.com"

recipient="whatsapp:+6591080064"

sender="whatsapp:+14155238886"

message=client.messages.create(to=recipient,

from\_= sender,

body=body)

else:

alert\_status=False

**Date Formatting**

**Page 8**

**datetest.py**

#To learn more about date formatting

#https://www.w3schools.com/python/gloss\_python\_date\_format\_codes.asp

#https://www.tutorialkart.com/python/python-datetime/python-datetime-format/

from datetime import datetime

#current date and time

now = datetime.now()

#date and time format: dd/mm/YYYY H:M:S

format = "%d/%m/%Y %H:%M:%S"

#format datetime using strftime()

format = "%d-%b-%Y %H:%M:%S"

time1 = now.strftime(format)

print("Formatted DateTime:", time1)