2

import time

import RPi.GPIO as gpio

gpio.setwarnings(False)

gpio.setmode(gpio.BOARD)

led1 = 15

led2 = 13

switch1 = 37

switch2 = 35

gpio.setup(led1,gpio.OUT,initial=0)

gpio.setup(led2,gpio.OUT,initial=0)

gpio.setup(switch1,gpio.IN)

gpio.setup(switch2,gpio.IN)

def glow\_led(event):

if event == switch1 :

gpio.output(led1, Truetime.sleep(3)

gpio.output(led1, False)

elif event == switch2 :

gpio.output(led2, True)

time.sleep(3)

gpio.output(led2, False)

gpio.add\_event\_detect(switch1, gpio.RISING ,

callback = glow\_led, bouncetime = 1)

gpio.add\_event\_detect(switch2, gpio.RISING ,

callback = glow\_led, bouncetime = 1)

try:

while(True):

time.sleep(1)

except KeyboardInterrupt:

gpio.cleanup()

print("")

3

import time

import RPi.GPIO as gpio

gpio.setwarnings(False)

gpio.setmode(gpio.BOARD)

led1 = 15

gpio.setup(led1,gpio.OUT,initial=0)

file1 = open('ledintervals.txt', 'r')

Lines = file1.readlines()

ON\_TIME = int(Lines[0].split("=")[1])

OFF\_TIME = int(Lines[1].split("=")[1])

try:

while(True):

gpio.output(led1,True)

time.sleep(ON\_TIME)

gpio.output(led1,False)

time.sleep(OFF\_TIME)

except KeyboardInterrupt:

gpio.cleanup()

print(""

4

4

import time

import RPi.GPIO as gpio

gpio.setwarnings(False)

gpio.setmode(gpio.BOARD)

relay1 = 38

gpio.setup(relay1,gpio.OUT,initial=0)

try:

gpio.output(relay1, True)

print("Relay is Switched On. Please Press

ctrl+c to exit")

time.sleep(15)

print("Relay is Switched Off.")

gpio.output(relay1, False)

except KeyboardInterrupt:

gpio.cleanup()

print("Program Exited")

print("")

5

from picamera import PiCamera

from time import sleep

import datetime

camera = PiCamera()

camera.start\_preview()

current\_date =

datetime.datetime.now().strftime('%d-%m-%Y

%H:%M:%S')

sleep(3)

camera.capture('/home/pi/Desktop/images/'+current

\_date+'.jpg')

camera.stop\_preview()

print("Image captured")

6

import RPi.GPIO as GPIO

import time

import datetime

led = 13

GPIO.setmode(GPIO.BOARD)

GPIO.setwarnings(False)

GPIO.setup(led, GPIO.OUT,initial=0)

GPIO.setup(led,GPIO.OUT)

from flask import Flask, render\_template

app = Flask( name )

@app.route('/')

def hello\_world():

return render\_template('web.html')

@app.route('/redledon')

def redledon():

GPIO.output(13, GPIO.LOW)

now = datetime.datetime.now()

timeString = now.strftime("%Y-%m-%d %H:%M")

templateData = {

'status': 'ON',

'time': timeString

}

return render\_template('web.html',

\*\*templateData)

@app.route('/redledoff') #Route for Turning

RedLed Off

def redledoff():

GPIO.output(13, GPIO.HIGH)

now = datetime.datetime.now()

timeString = now.strftime("%Y-%m-%d %H:%M")

templateData = {

'status': 'OFF',

'time': timeString

}

return render\_template('web.html',

\*\*templateData)

name == " main ":

app.run(debug = True, port = 4000,

host='0.0.0.0')

#templates/web.html

<html>

<body>

<title>Raspberry PI Remote Control</title>

<h1>Raspberry PI Remote Control</h1>

<h2>Light Status : {{status}}, Last Modified

: {{time}}</h2>

<form action="http://localhost:4000/redledon">

<input type="submit" value="Red LED On">

</form>

<form

action="http://localhost:4000/redledoff">

<input type="submit" value="Red LED Off">

</form>

</body>

</html>

7

import RPi.GPIO as gpio

import picamera

import time

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

from email.mime.base import MIMEBase

from email import encoders

from email.mime.image import MIMEImage

fromaddr = "<\*\*yourEmailAddress\*\*>"

toaddr = "<toAddress>"

mail = MIMEMultipart()

mail['From'] = fromaddr

mail['To'] = toaddr

mail['Subject'] = "Attachment"

body = "Please find the attachment

led=15

pir=12

HIGH=1

LOW=0

gpio.setwarnings(False)

gpio.setmode(gpio.BOARD)

gpio.setup(led, gpio.OUT)

gpio.setup(pir, gpio.IN)

data=""

def sendMail(data):

mail.attach(MIMEText(body, 'plain'))

print(data)

dat='%s.jpg'%data

print(data)

attachment = open(dat, 'rb')

image=MIMEImage(attachment.read())

attachment.close()

mail.attach(image)

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

server.starttls()

server.login(fromaddr, "alsdatasave2017")

text = mail.as\_string()

server.sendmail(fromaddr, toaddr, text)

server.quit()

def capture\_image():

data= time.strftime("Image was captured on

%H:%M:%S|%d\_%b\_%Y")

camera.start\_preview()

time.sleep(5)

print(data)

camera.capture('%s.jpg'%data)

camera.stop\_preview()

time.sleep(1)

sendMail(data)

gpio.output(led , 0)

camera = picamera.PiCamera()

camera.rotation=180

camera.awb\_mode= 'auto'

camera.brightness=55

while 1:

if gpio.input(pir)==1:

gpio.output(led, HIGH)

capture\_image()

while(gpio.input(pir)==1):

time.sleep(1)

else:

gpio.output(led, LOW)

time.sleep(0.01)

8

import time

import RPi.GPIO as gpio

from flask import Flask, render\_template

import datetime

app = Flask( name )

gpio.setwarnings(False)

gpio.setmode(gpio.BOARD)

led1 = 13

switch1 = 35

gpio.setup(led1,gpio.OUT,initial=1)

gpio.setup(switch1,gpio.IN)

light\_status = "OFF"

def glow\_led(event):

print("Entered Here")

global light\_status

if event == switch1 and light\_status == "OFF":

gpio.output(led1, False)

light\_status = "ON"

22

elif event == switch1 and light\_status ==

"ON":

gpio.output(led1, True)

light\_status = "OFF"

@app.route('/')

def ledstatus():

now = datetime.datetime.now()

timeString = now.strftime("%H:%M %d-%m-%Y")

templateData = {

'status' : light\_status,

'time': timeString

}

return render\_template('lightstatus.html',

\*\*templateData)

gpio.add\_event\_detect(switch1, gpio.RISING ,

callback = glow\_led, bouncetime =

100)

app.run(debug = True, port = 4000,

host='0.0.0.0')

#templates/lightstatus.html

<html>

<body>

<title>Raspberry PI Remote Light Status</title>

<h1>Raspberry PI Remote Control</h1>

<h2>Light Status : {{status}}, Last Seen :

{{time}}</h2>

<form action="http://localhost:4000">

<input type="submit" value="Get Light Status">

</form>

</body>

</html>

9

import socket

import Adafruit\_MCP3008

import Adafruit\_GPIO.SPI as SPI

import time

HOST = '127.0.0.1'

PORT = 4000

SPI\_DEVICE = 0

SPI\_PORT = 0

mcp =

Adafruit\_MCP3008.MCP3008(spi=SPI.SpiDev(SPI\_PORT,

SPI\_DEVICE))

try:

with socket.socket(socket.AF\_INET,

socket.SOCK\_STREAM) as s:

s.bind((HOST, PORT))

s.listen()

conn, addr = s.accept()

with conn:

print('Connected by', addr)

while True:

value = mcp.read\_adc(0)

print("Gas Value ", value , "units")

if(value >300):

data = "Alert".encode('utf-8')

conn.sendall(data)

time.sleep(3)

except KeyboardInterrupt:

s.close()

GPIO.cleanup()

30

#client program to alert through buzzer.

import socket

import RPi.GPIO as GPIO

import time

Buzzer = 36

HOST = '127.0.0.1'

PORT = 4000

GPIO.setmode(GPIO.BOARD)

GPIO.setup(36, GPIO.OUT)

GPIO.setwarnings(False)

try:

with socket.socket(socket.AF\_INET,

socket.SOCK\_STREAM) as s:

s.connect((HOST,PORT))

while True:

data = s.recv(1024).decode('utf-8')

print(data)

if(str(data) == 'Alert'):

print("ALert! Gas Leakage detected")

GPIO.output(36, True)

time.sleep(3)

GPIO.output(36, False)

time.sleep(3)

except KeyboardInterrupt:

s.close()

GPIO.cleanup()