CODINGS USED

**import serial**

**import RPi.GPIO as GPIO**

**import sys**

**import time**

**import datetime**

**import cv2**

**import numpy as np**

**import smtplib**

**from email.mime.multipart import MIMEMultipart**

**from email.mime.base import MIMEBase**

**from email.mime.text import MIMEText**

**from email.utils import COMMASPACE, formatdate**

**from email import encoders**

**GPIO.setmode(GPIO.BCM)**

**ser = serial.Serial('/dev/ttyUSB0', 9600)**

**GPIO.setup(16, GPIO.OUT)**

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

**GPIO.setup(20, GPIO.OUT)**

**GPIO.setup(21, GPIO.OUT)**

**LCD\_RS = 7**

**LCD\_E = 8**

**LCD\_D4 = 25**

**LCD\_D5 = 24**

**LCD\_D6 = 23**

**LCD\_D7 = 18**

**LCD\_WIDTH = 16**

**LCD\_CHR = True**

**LCD\_CMD = False**

**LCD\_LINE\_1 = 0x80**

**LCD\_LINE\_2 = 0xC0**

**# Timing constants**

**E\_PULSE = 0.0005**

**E\_DELAY = 0.0005**

**def main():**

**# Main program block**

**GPIO.setwarnings(False)**

**GPIO.setmode(GPIO.BCM) # Use BCM GPIO numbers**

**GPIO.setup(LCD\_E, GPIO.OUT) # E**

**GPIO.setup(LCD\_RS, GPIO.OUT) # RS**

**GPIO.setup(LCD\_D4, GPIO.OUT) # DB4**

**GPIO.setup(LCD\_D5, GPIO.OUT) # DB5**

**GPIO.setup(LCD\_D6, GPIO.OUT) # DB6**

**GPIO.setup(LCD\_D7, GPIO.OUT) # DB7**

**# Initialise display**

**lcd\_init()**

**def lcd\_init():**

**# Initialise display**

**lcd\_byte(0x33,LCD\_CMD) # 110011 Initialise**

**lcd\_byte(0x32,LCD\_CMD) # 110010 Initialise**

**lcd\_byte(0x06,LCD\_CMD) # 000110 Cursor move direction**

**lcd\_byte(0x0C,LCD\_CMD) # 001100 Display On,Cursor Off, Blink Off**

**lcd\_byte(0x28,LCD\_CMD) # 101000 Data length, number of lines, font size**

**lcd\_byte(0x01,LCD\_CMD) # 000001 Clear display**

**time.sleep(E\_DELAY)**

**def LOCAL(center, pixels):**

**out = []**

**for a in pixels:**

**if a >= center:**

**out.append(1)**

**else:**

**out.append(0)**

**return out**

**def get\_pixel\_else\_0(l, idx, idy, default=0):**

**try:**

**return l[idx,idy]**

**except IndexError:**

**return default**

**def lcd\_byte(bits, mode):**

**# Send byte to data pins**

**# bits = data**

**# mode = True for character**

**# False for command**

**GPIO.output(LCD\_RS, mode) # RS**

**# High bits**

**GPIO.output(LCD\_D4, False)**

**GPIO.output(LCD\_D5, False)**

**GPIO.output(LCD\_D6, False)**

**GPIO.output(LCD\_D7, False)**

**if bits&0x10==0x10:**

**GPIO.output(LCD\_D4, True)**

**if bits&0x20==0x20:**

**GPIO.output(LCD\_D5, True)**

**if bits&0x40==0x40:**

**GPIO.output(LCD\_D6, True)**

**if bits&0x80==0x80:**

**GPIO.output(LCD\_D7, True)**

**# Toggle 'Enable' pin**

**lcd\_toggle\_enable()**

**# Low bits**

**GPIO.output(LCD\_D4, False)**

**GPIO.output(LCD\_D5, False)**

**GPIO.output(LCD\_D6, False)**

**GPIO.output(LCD\_D7, False)**

**if bits&0x01==0x01:**

**GPIO.output(LCD\_D4, True)**

**if bits&0x02==0x02:**

**GPIO.output(LCD\_D5, True)**

**if bits&0x04==0x04:**

**GPIO.output(LCD\_D6, True)**

**if bits&0x08==0x08:**

**GPIO.output(LCD\_D7, True)**

**# Toggle 'Enable' pin**

**lcd\_toggle\_enable()**

**def lcd\_toggle\_enable():**

**# Toggle enable**

**time.sleep(E\_DELAY)**

**GPIO.output(LCD\_E, True)**

**time.sleep(E\_PULSE)**

**GPIO.output(LCD\_E, False)**

**time.sleep(E\_DELAY)**

**def lcd\_string(message,line):**

**# Send string to display**

**message = message.ljust(LCD\_WIDTH," ")**

**lcd\_byte(line, LCD\_CMD)**

**for i in range(LCD\_WIDTH):**

**lcd\_byte(ord(message[i]),LCD\_CHR)**

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

**try:**

**main()**

**lcd\_string(" VEGETABLE ",LCD\_LINE\_1)**

**lcd\_string("MONITORING SYSTEM ",LCD\_LINE\_2)**

**time.sleep(3) # 3 second delay**

**while True:**

**for i in range(1):**

**message = ser.readline()**

**for i in range(1):**

**message1 = ser.readline()**

**data1 = (message.decode("utf-8"))**

**data2 = (message1.decode("utf-8"))**

**d1=(int(data1))**

**d2=(int(data2))**

**print(d1)**

**print(d2)**

**lcd\_string("AIR QUALITY=%d "% d1,LCD\_LINE\_1)**

**lcd\_string("MQ-4 =%d"% d2, LCD\_LINE\_2)**

**GPIO.output(20, True)**

**GPIO.output(12, False)**

**GPIO.output(16, False)**

**GPIO.output(21, False)**

**time.sleep(1) # 3 second delay**

**if d1 > 200 and d1 < 300:**

**GPIO.output(20, False)**

**GPIO.output(21, True)**

**GPIO.output(12, True)**

**GPIO.output(16, False)**

**lcd\_string("BAD QUALITY-55% ",LCD\_LINE\_1)**

**lcd\_string("EMAIL SENDING ", LCD\_LINE\_2)**

**camera = cv2.VideoCapture(0)**

**for i in range(1):**

**return\_value, image = camera.read()**

**cv2.imwrite('opencv'+str(i)+'.png', image)**

**del(camera)**

**print('image capture...')**

**time.sleep(2)**

**GPIO.output(21, False)**

**fromaddr = "ssanthosh2k3@gmail.com"**

**toaddr = "vinothinivijay22@gmail.com"**

**msg = MIMEMultipart()**

**msg['From'] = fromaddr**

**msg['To'] = toaddr**

**msg['Subject'] = "QUALITY MONITORING SYSTEM"**

**body = "CAPTURED IMAGE AFTER DETECTING"**

**msg.attach(MIMEText(body, 'plain'))**

**filename = "opencv0.png"**

**attachment = open("/home/pi/PROJECT/opencv0.png", "rb")**

**part = MIMEBase('application', 'octet-stream')**

**part.set\_payload((attachment).read())**

**encoders.encode\_base64(part)**

**part.add\_header('Content-Disposition', "attachment; filename= %s" % filename)**

**msg.attach(part)**

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

**server.starttls()**

**server.login(fromaddr, "jodhmooqjmhkoopj")**

**text = msg.as\_string()**

**server.sendmail(fromaddr, toaddr, text)**

**server.quit()**

**print('mail sended...')**

**if d1>300:**

**GPIO.output(20, False)**

**GPIO.output(21, True)**

**GPIO.output(16, True)**

**GPIO.output(12, False)**

**lcd\_string("BAD QUALITY-80%",LCD\_LINE\_1)**

**lcd\_string("EMAIL SENDING ", LCD\_LINE\_2)**

**camera = cv2.VideoCapture(0)**

**for i in range(1):**

**return\_value, image = camera.read()**

**cv2.imwrite('opencv'+str(i)+'.png', image)**

**del(camera)**

**print('image capture...')**

**time.sleep(2)**

**GPIO.output(21, False)**

**fromaddr = "ssanthosh2k3@gmail.com"**

**toaddr = "vinothinivijay22@gmail.com"**

**msg = MIMEMultipart()**

**msg['From'] = fromaddr**

**msg['To'] = toaddr**

**msg['Subject'] = "QUALITY MONITORING SYSTEM"**

**body = "CAPTURED IMAGE AFTER DETECTING"**

**msg.attach(MIMEText(body, 'plain'))**

**filename = "opencv0.png"**

**attachment = open("/home/pi/PROJECT/opencv0.png", "rb")**

**part = MIMEBase('application', 'octet-stream')**

**part.set\_payload((attachment).read())**

**encoders.encode\_base64(part)**

**part.add\_header('Content-Disposition', "attachment; filename= %s" % filename)**

**msg.attach(part)**

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

**server.starttls()**

**server.login(fromaddr, "jodhmooqjmhkoopj")**

**text = msg.as\_string()**

**server.sendmail(fromaddr, toaddr, text)**

**server.quit()**

**print('mail sended...')**

**if d2 > 200 and d2 < 300:**

**GPIO.output(20, False)**

**GPIO.output(21, True)**

**GPIO.output(12, True)**

**GPIO.output(16, False)**

**lcd\_string("BAD QUALITY-55%",LCD\_LINE\_1)**

**lcd\_string("EMAIL SENDING ", LCD\_LINE\_2)**

**camera = cv2.VideoCapture(0)**

**for i in range(1):**

**return\_value, image = camera.read()**

**cv2.imwrite('opencv'+str(i)+'.png', image)**

**del(camera)**

**print('image capture...')**

**time.sleep(2)**

**GPIO.output(21, False)**

**fromaddr = "ssanthosh2k3@gmail.com"**

**toaddr = "vinothinivijay22@gmail.com"**

**msg = MIMEMultipart()**

**msg['From'] = fromaddr**

**msg['To'] = toaddr**

**msg['Subject'] = "QUALITY MONITORING SYSTEM"**

**body = "CAPTURED IMAGE AFTER DETECTING"**

**msg.attach(MIMEText(body, 'plain'))**

**filename = "opencv0.png"**

**attachment = open("/home/pi/PROJECT/opencv0.png", "rb")**

**part = MIMEBase('application', 'octet-stream')**

**part.set\_payload((attachment).read())**

**encoders.encode\_base64(part)**

**part.add\_header('Content-Disposition', "attachment; filename= %s" % filename)**

**msg.attach(part)**

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

**server.starttls()**

**server.login(fromaddr, "jodhmooqjmhkoopj")**

**text = msg.as\_string()**

**server.sendmail(fromaddr, toaddr, text)**

**server.quit()**

**print('mail sended...')**

**if d2>300:**

**GPIO.output(20, False)**

**GPIO.output(21, True)**

**GPIO.output(16, True)**

**GPIO.output(12, False)**

**lcd\_string("BAD QUALITY-80% ",LCD\_LINE\_1)**

**lcd\_string("EMAIL SENDING ", LCD\_LINE\_2)**

**camera = cv2.VideoCapture(0)**

**for i in range(1):**

**return\_value, image = camera.read()**

**cv2.imwrite('opencv'+str(i)+'.png', image)**

**del(camera)**

**print('image capture...')**

**time.sleep(2)**

**GPIO.output(21, False)**

**fromaddr = "ssanthosh2k3@gmail.com"**

**toaddr = "vinothinivijay22@gmail.com"**

**msg = MIMEMultipart()**

**msg['From'] = fromaddr**

**msg['To'] = toaddr**

**msg['Subject'] = "QUALITY MONITORING SYSTEM"**

**body = "CAPTURED IMAGE AFTER DETECTING"**

**msg.attach(MIMEText(body, 'plain'))**

**filename = "opencv0.png"**

**attachment = open("/home/pi/PROJECT/opencv0.png", "rb")**

**part = MIMEBase('application', 'octet-stream')**

**part.set\_payload((attachment).read())**

**encoders.encode\_base64(part)**

**part.add\_header('Content-Disposition', "attachment; filename= %s" % filename)**

**msg.attach(part)**

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

**server.starttls()**

**server.login(fromaddr, "jodhmooqjmhkoopj")**

**text = msg.as\_string()**

**server.sendmail(fromaddr, toaddr, text)**

**server.quit()**

**print('mail sended...')**

**except:**

**GPIO.cleanup()**

DHT-22 CODINGS

#include "DHT.h" \

#define DHTPIN 2

#define DHTTYPE DHT22

dht(DHTPIN, DHTTYPE);

void setup()

{

Serial.begin(9600);

Serial.println("DHTxx Robojax test!");

dht.begin();

}

void loop()

{

delay(2000);

Serial.print("Temperature: "); Serial.print(getTemp("c"));

Serial.print(" \*C ");

Serial.print(getTemp("f"));

Serial.println (" \*F");

Serial.println("-----------------");

Serial.print("Heat index: "); Serial.print(getTemp("hic"));

Serial.print(" \*C ");

Serial.print(getTemp("hif"));

Serial.println(" \*F");

Serial.print(getTemp("k"));

Serial.println(" \*K");

Serial.println("-----------------"); Serial.print("Humidity: ");

Serial.print(getTemp("h"));

Serial.println(" % "); Serial.println("===========================");

}

float getTemp(String req)

{

float h = dht.readHumidity();

float t = dht.readTemperature();

float f = dht.readTemperature(true);

float hif = dht.computeHeatIndex(f, h);

float hic = dht.computeHeatIndex(t, h, false);

if (isnan(h) || isnan(t) || isnan(f))

{

Serial.println("Failed to read from DHT sensor!"); return;

}

float k = t + 273.15;

if(req =="c")

{

return t

}

else if(req =="f")

{

return f;//

}

else if(req =="h")

{

return h;

}

else if(req =="hif")

{

return hif

}

else if(req =="hic")

{

return hic

else if(req =="k")

{

return k

}else

{

return 0.000

}

}

RELAY CODINGS

#include<dht.h>  
#define dht\_apin A0  
  
  
// you have to connect sensor input  
  
dht DHT  
  
const int pl=13; // heat  
const int p2=12; //cold  
  
  
  
  
void setup()  
{  
  
  Serial.begin(9600);  
   
  // to reboot system  
   
  delay(500);  
   
  Serial.println("Temperature Sensor\n\n);  
   
  delay(1000);  
   
  pinMode (p1, OUTPUT);  
   
  pinMode (p1, OUTPUT);  
  return 0.000;  
  
  
}  
void loop()  
  
{  
  DHT.read11 (dht\_apin):  
  
  
  Serial.print("TEMPERATURE=""):  
  
  Serial.print (DHT, temperature):  
  
  if (DHT.temperature<35)  
  
  {  
    digitalWrite(p1, HIGH); digitalWrite(p2, LOW);  
  
    digitalWrite(pl, LOW); digitalice (p2, WI):  
  }  
  else  
  {  
    digitalWrite(p1,LOW);  
    digitalWrite(p2,HIGH);  
  }  
   
  return 0.000;  
}