

```
import ADS1256
import SH1106
import config1
import subprocess
import RPi.GPIO as GPIO
import SH1106OLED
import time
import configOLED
import traceback
import numpy as np
import pandas as pd
import joblib
import httpplib2
import urllib3
import thingspeak
import smtplib

from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.image import MIMEImage
from PIL import Image, ImageDraw, ImageFont

channel_id= 2053898
write_key1='XSQN721HU79GG18H'
a=2


ADC = ADS1256.ADS1256()
ADC.ADS1256_init()
disp = SH1106OLED.SH1106()
ADC_Value = ADC.ADS1256_GetAll()
```

### **#GPIO define**

RST\_PIN = 25

CS\_PIN = 8

DC\_PIN = 24

### **#joystick**

KEY\_UP\_PIN = 6

### **#button key**

KEY1\_PIN = 21

KEY2\_PIN = 20

KEY3\_PIN = 16

disp = SH1106OLED.SH1106()

disp.Init()

### **# Clear display.**

disp.clear()

GPIO.setmode(GPIO.BCM)

GPIO.setup(KEY\_UP\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) #

Input with pull-up

GPIO.setup(KEY1\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) #

Input with pull-up

GPIO.setup(KEY2\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) #

Input with pull-up

GPIO.setup(KEY3\_PIN, GPIO.IN, pull\_up\_down=GPIO.PUD\_UP) #

Input with pull-up

### **# Create blank image for drawing.**

```
image = Image.new('1', (disp.width, disp.height), "WHITE")
```

```
# Get drawing object to draw on image.
```

```
draw = ImageDraw.Draw(image)
```

```
font = ImageFont.truetype('Font.ttf',25)
```

```
font10 = ImageFont.truetype('Font.ttf',10)
```

```
draw.text((6,10),'JUZTblow', font = font, fill = 0)
```

```
draw.text((95,38),'Sugar', font = font10, fill = 0)
```

```
disp.ShowImage(disp.getbuffer(image))
```

```
time.sleep(5)
```

```
disp.clear()
```

```
image1 = Image.new('1', (disp.width, disp.height), "WHITE")
```

```
draw = ImageDraw.Draw(image1)
```

```
font = ImageFont.truetype('Font.ttf', 15)
```

```
font10 = ImageFont.truetype('Font.ttf',10)
```

```
font20 = ImageFont.truetype('Font.ttf',10)
```

```
draw.text((25,5), 'Instructions ', font = font, fill = 0)
```

```
draw.text((12,25),'To Be Followed',font = font, fill = 0)
```

```
disp.Init()
```

```
disp.ShowImage(disp.getbuffer(image1))
```

```
time.sleep(5)
```

```
disp.clear()
```

```
image2 = Image.new('1', (disp.width, disp.height), "WHITE")
```

```
draw = ImageDraw.Draw(image2)
```

```
font = ImageFont.truetype('Font.ttf', 14)
```

```
font10 = ImageFont.truetype('Font.ttf',10)
```

```
font20 = ImageFont.truetype('Font.ttf',10)
```

```
draw.text((15,20), 'Open the Knob', font = font, fill = 0)
```

```
draw.text((2,1), '1', font = font, fill = 0)
draw.text((4,5),'chamber carefully ',font = font10, fill = 0)
disp.Init()
disp.ShowImage(disp.getbuffer(image2))
time.sleep(5)
disp.clear()
```

```
image3 = Image.new('1', (disp.width, disp.height), "WHITE")
draw = ImageDraw.Draw(image3)
font = ImageFont.truetype('Font.ttf', 14)
font10 = ImageFont.truetype('Font.ttf',15)
font20 = ImageFont.truetype('Font.ttf',10)
draw.text((1,1), '2', font = font, fill = 0)
draw.text((35,20),'Blow into',font = font10, fill = 0)
draw.text((35,40),'Chamber',font = font, fill = 0)
draw.text((4,5),'chamber carefully ',font = font10, fill = 0)
disp.Init()
disp.ShowImage(disp.getbuffer(image3))
time.sleep(5)
disp.clear()
```

```
image4 = Image.new('1', (disp.width, disp.height), "WHITE")
draw = ImageDraw.Draw(image4)
font = ImageFont.truetype('Font.ttf', 14)
font10 = ImageFont.truetype('Font.ttf',10)
font20 = ImageFont.truetype('Font.ttf',10)
draw.text((1,1), '3', font = font, fill = 0)
draw.text((15,20), 'Close the Knob', font = font, fill = 0)
draw.text((18,40), 'while blowing',font = font, fill = 0)
```

```
disp.Init()
disp.ShowImage(disp.getbuffer(image4))
time.sleep(5)
disp.clear()
```

```
image7 = Image.new('1', (disp.width, disp.height), "WHITE")
draw = ImageDraw.Draw(image7)
font = ImageFont.truetype('Font.ttf', 14)
font10 = ImageFont.truetype('Font.ttf',10)
font20 = ImageFont.truetype('Font.ttf',10)
draw.text((5,12), 'Before fasting - BF ', font = font, fill = 0)
draw.text((5,32),'After fasting - AF',font = font, fill = 0)
disp.Init()
disp.ShowImage(disp.getbuffer(image7))
time.sleep(5)
disp.clear()
```

```
image6 = Image.new('1', (disp.width, disp.height), "WHITE")
draw = ImageDraw.Draw(image6)
font = ImageFont.truetype('Font.ttf', 14)
font10 = ImageFont.truetype('Font.ttf',10)
font20 = ImageFont.truetype('Font.ttf',10)
draw.text((20,1), 'After Blowing', font = font, fill = 0)
draw.text((14,25),'Select BF or AF',font = font, fill = 0)
draw.text((22,47),'to continue',font = font, fill = 0)
disp.Init()
disp.ShowImage(disp.getbuffer(image6))
time.sleep(5)
```

while 1:

if GPIO.input(KEY1\_PIN): # button is released

print()

else:

draw.ellipse((70,0,90,20), outline=255, fill=1) #A button filled

ADC\_ch0 = ADC\_Value[0]\*3.3/0x7ffff\*1000

ADC\_value=[[round(ADC\_ch0,2)]]

ADC\_value1=str(ADC\_value)

from joblib import load

regr=joblib.load('ANN\_Training.joblib')

predicted\_output=regr.predict(ADC\_value)

predicted=np.round(predicted\_output,2)

for output in predicted:

glucose=str(output)

print(glucose)

a=output

image1 = Image.new('1', (disp.width, disp.height), "WHITE")

draw = ImageDraw.Draw(image1)

font = ImageFont.truetype('Font.ttf', 10)

font10 = ImageFont.truetype('Font.ttf',13)

font20 = ImageFont.truetype('Font.ttf',10)

draw.text((5,5),'Before fasting is selected', font = font, fill = 0)

draw.text((20,22),'Are you sure?', font = font10, fill = 0)

draw.text((1 ,45),'press OK to continue',font = font10, fill = 0)

disp.ShowImage(disp.getbuffer(image1))

time.sleep(1)

```

if(GPIO.input(KEY2_PIN)):
    print()
else:
    draw.ellipse((100,20,120,40), outline=255, fill=1) #B button filled
    ADC_ch0 = ADC_Value[0]*3.3/0x7ffff*1000
    ADC_value_A=[[round(ADC_ch0,2)]]
    ADC_value1_A=str(ADC_value_A)
    from joblib import load
    regr_A=joblib.load('ANN_Training.joblib')
    predicted_output_A=regr_A.predict(ADC_value_A)
    disp.Init()
    # Clear display.
    disp.clear()
    predicted_A=np.round(predicted_output_A,2)
    for output_A in predicted_A:
        glucose_A=str(output_A)
        print(glucose_A)
    output=0
    a=output_A
    image1 = Image.new('1', (disp.width, disp.height), "WHITE")
    draw = ImageDraw.Draw(image1)
    font = ImageFont.truetype('Font.ttf', 11)
    font10 = ImageFont.truetype('Font.ttf',13)
    font20 = ImageFont.truetype('Font.ttf',10)
    draw.text((5,1), 'After fasting is selected', font = font, fill = 0)
    draw.text((25,22), 'Are you sure?', font = font10, fill = 0)
    draw.text((1,45),'press OK to continue',font = font10, fill = 0)

```

```
draw.text((60,30), '(mg/dL)', font = font10, fill = 0)
disp.ShowImage(disp.getbuffer(image1))
time.sleep(1)
```

```
if GPIO.input(KEY3_PIN):# button is released
    print()
else:
    if(a==output):
        image2 = Image.new('1', (disp.width, disp.height), "WHITE")
        draw = ImageDraw.Draw(image2)
        font = ImageFont.truetype('Font.ttf', 15)
        font10 = ImageFont.truetype('Font.ttf',10)
        font20 = ImageFont.truetype('Font.ttf',10)
        draw.text((20,22), 'Measuring...', font = font, fill = 0)
        disp.ShowImage(disp.getbuffer(image2))
        time.sleep(5)
        image1 = Image.new('1', (disp.width, disp.height), "WHITE")
        draw = ImageDraw.Draw(image2)
        image1 = Image.new('1', (disp.width, disp.height), "WHITE")
        draw = ImageDraw.Draw(image1)
        font = ImageFont.truetype('Font.ttf', 14)
        font10 = ImageFont.truetype('Font.ttf',12)
        font20 = ImageFont.truetype('Font.ttf',10)
        draw.text((1,1), 'Before fasting:', font = font10, fill = 0)
        draw.text((10,23), 'Blood Sugar Value:', font = font10, fill = 0)
        draw.text((20,45),str(glucose),font = font, fill = 0)
        draw.text((60,45), '(mg/dL)', font = font10, fill = 0)
        disp.ShowImage(disp.getbuffer(image1))
```



```

time.sleep(1)
print("Acetone(ppb): " + ADC_value1)
print("Blood Sugar Value(mg/dL): " + glucose)
def measure(channel):
    try:
        response = channel.update( {'field2': output})
    except:
        print("Connection Failure")
if __name__ == "__main__":
    channel = thingspeak.Channel(channel_id,write_key1)
    measure(channel)
    sender_email = "giumtech5@gmail.com"
    sender_password = "hmpd hhjc fayd oxpe"
# Email details
    recipient_email = "divyachitra391@gmail.com"
    subject = "Alert!!!"
    output=40
    if output>=250:
        body = "\nHigh Sugar level detected.\nPlease consult your Doctor."
        sugar="Sugar level = "
        url1="\nChannel_ID = "
        url='2053898'
        message1 = MIMEMultipart()
        message1["From"] = sender_email
        message1["To"] = recipient_email
        message1["Subject"] = subject

# Add the message body

```

```
message1.attach(MIMEText(sugar + glucose + body + url1 + url,
"plain"))
```

```
with smtplib.SMTP("smtp.gmail.com", 587) as smtp:
```

```
    smtp.starttls()
```

```
    smtp.login(sender_email, sender_password)
```

```
    # Send the email
```

```
    smtp.sendmail(sender_email, recipient_email,
message1.as_string())
```

```
    print("Email sent successfully!")
```

```
elif output<=50:
```

```
    body = "\nLow Sugar level detected.\nPlease consult your Doctor."
```

```
    sugar="Sugar level = "
```

```
    url1="\nChannel_ID = "
```

```
    url='2053898'
```

```
    message1 = MIMEMultipart()
```

```
    message1["From"] = sender_email
```

```
    message1["To"] = recipient_email
```

```
    message1["Subject"] = subject
```

```
    message1.attach(MIMEText(sugar + glucose + body + url1 + url,
"plain"))
```

```
    # Add the message body
```

```
    message.attach(MIMEText(body, "plain"))
```

```
    with smtplib.SMTP("smtp.gmail.com", 587) as smtp:
```

```
        smtp.starttls()
```

```

smtp.login(sender_email, sender_password)

# Send the email

smtp.sendmail(sender_email, recipient_email,
message1.as_string())

print("Email sent successfully!")

elif(a==output_A):
    image2 = Image.new('1', (disp.width, disp.height), "WHITE")
    draw = ImageDraw.Draw(image2)
    font = ImageFont.truetype('Font.ttf', 15)
    font10 = ImageFont.truetype('Font.ttf',10)
    font20 = ImageFont.truetype('Font.ttf',10)
    draw.text((20,22), 'Measuring...', font = font, fill = 0)
    disp.ShowImage(disp.getbuffer(image2))
    time.sleep(5)
    image1 = Image.new('1', (disp.width, disp.height), "WHITE")
    draw = ImageDraw.Draw(image1)
    font = ImageFont.truetype('Font.ttf', 14)
    font10 = ImageFont.truetype('Font.ttf',12)
    font20 = ImageFont.truetype('Font.ttf',10)
    draw.text((1,1), 'After fasting:', font = font10, fill = 0)
    draw.text((10,25), 'Blood Sugar Value:', font = font10, fill = 0)
    draw.text((20,45),str(glucose_A),font = font, fill = 0)
    draw.text((60,45), '(mg/dL)', font = font10, fill = 0)
    disp.ShowImage(disp.getbuffer(image1))
    time.sleep(1)
    print("Acetone(ppb): " + ADC_value1_A)

```

```

print("Blood Sugar Value(mg/dL): " + glucose_A)
def measure(channel):
    try:
        response = channel.update( {'field3': output_A})
    except:
        print("Connection Failure")
if __name__ == "__main__":
    channel = thingspeak.Channel(channel_id,write_key1)
    measure(channel)
    sender_email = "giumtech5@gmail.com"
    sender_password = "hmpd hhjc fayd oxpe"
    # Email details
    recipient_email = "divyachitra391@gmail.com"
    subject = "Alert!!!"
    output_A=45
    if output_A>=250:
        body = "\nHigh Sugar level detected.\nPlease consult your Doctor."
        sugar="Sugar level = "
        url1="\nChannel_ID = "
        url='2053898'
        message1 = MIMEMultipart()
        message1["From"] = sender_email
        message1["To"] = recipient_email
        message1["Subject"] = subject

        # Add the message body
        message1.attach(MIMEText(sugar+ glucose_A + body + url1 + url,
"plain"))

        # Create a SMTP connection

```

```

with smtplib.SMTP("smtp.gmail.com", 587) as smtp:
    smtp.starttls()
    smtp.login(sender_email, sender_password)

    # Send the email

    smtp.sendmail(sender_email, recipient_email,
message1.as_string())

    print("Email sent successfully!")
elif output_A<=50:
    body = "\nLow Sugar level detected.\nPlease consult your Doctor."
    sugar="Sugar level = "
    url1="\nChannel_ID = "
    url='2053898'
    message1 = MIMEMultipart()
    message1["From"] = sender_email
    message1["To"] = recipient_email
    message1["Subject"] = subject

    # Add the message body

    message1.attach(MIMEText(sugar+ glucose_A + body + url1 + url,
"plain"))

# Create a SMTP connection

with smtplib.SMTP("smtp.gmail.com", 587) as smtp:
    smtp.starttls()
    smtp.login(sender_email, sender_password)

```

**# Send the email**

```
smtp.sendmail(sender_email, recipient_email,  
message1.as_string())
```

```
print("Email sent successfully!")
```