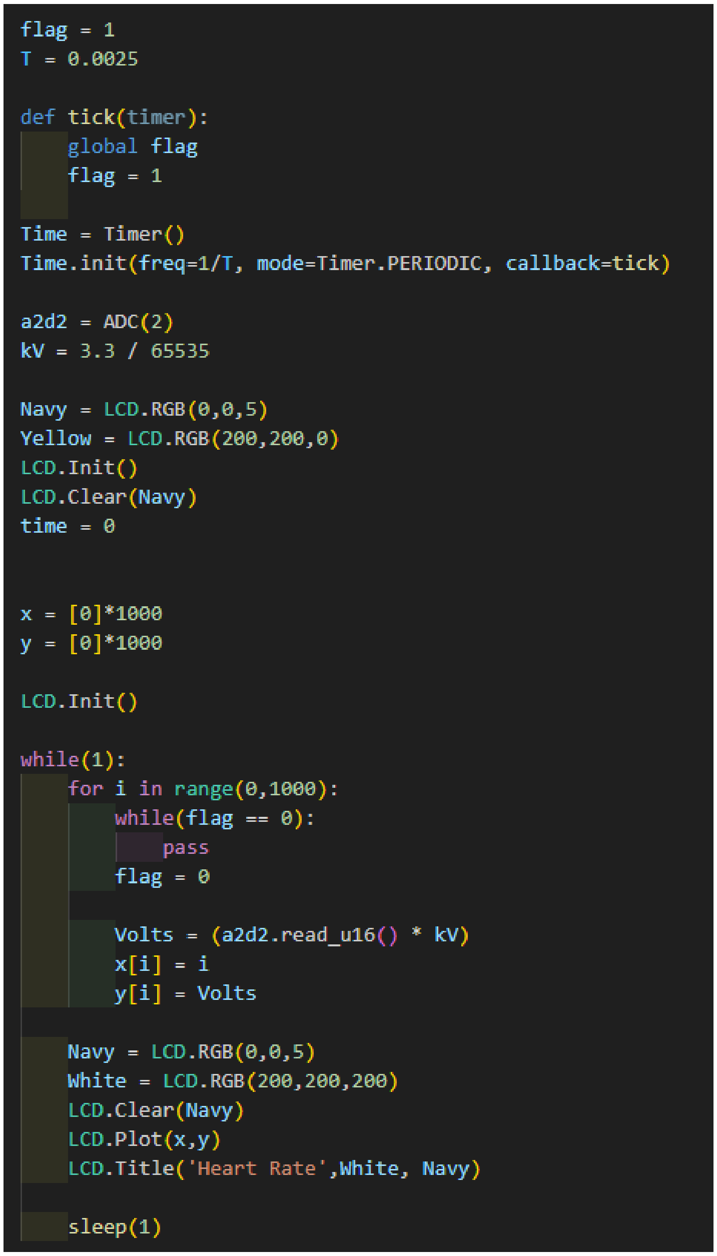
Heart Sensor

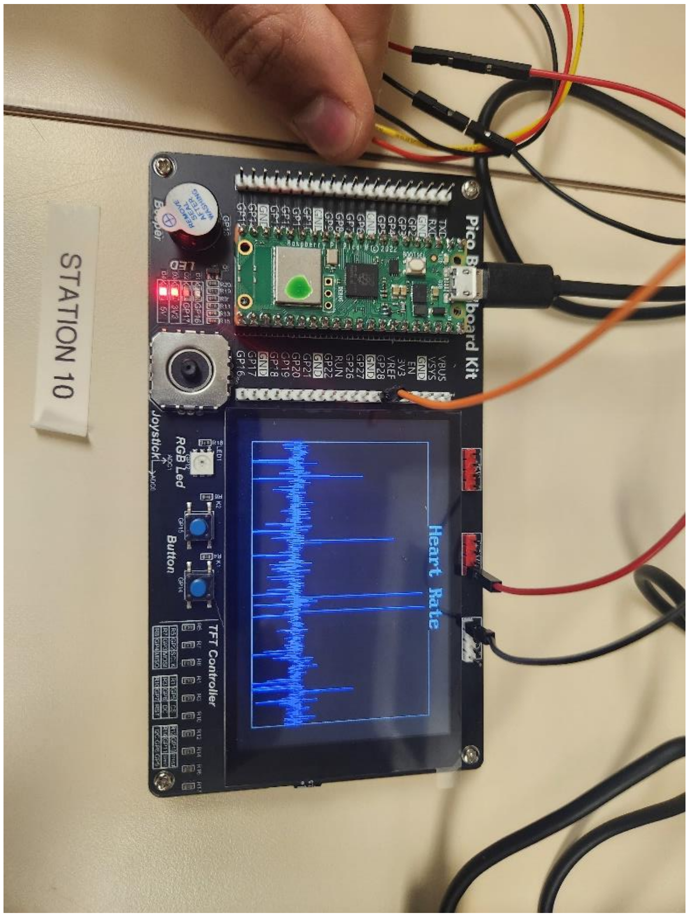
Write a Python program which

- Uses the Heart & Pulse sensor to record your pulse
- Detects each pulse automatically,
- Computes your heart rate in beats-per-minute, and
- Displays your pulse as a graph on the graphics display as well as your beats-per-minute.

1) Write a Python program which measures and displays your pulse
   - Give the program listing as well as the result on your graphics display



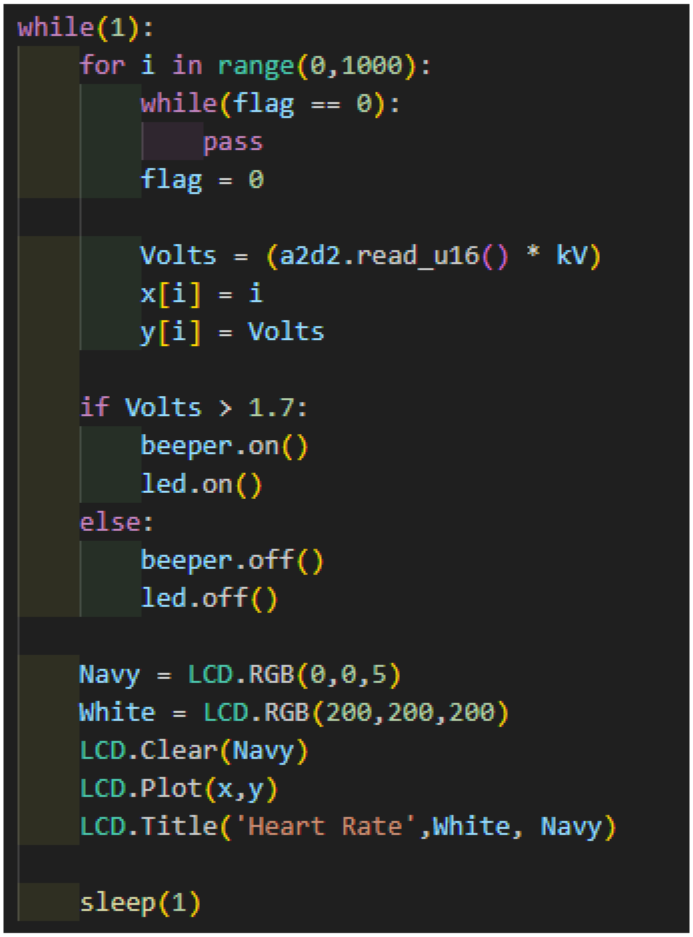
```
flag = 1
T = 0.0025

def tick(timer):
    global flag
    flag = 1

Time = Timer()
Time.init(freq=1/T, mode=Timer.PERIODIC, callback=tick)

a2d2 = ADC(2)
kV = 3.3 / 65535

Navy = LCD.RGB(0,0,5)
Yellow = LCD.RGB(200,200,0)
LCD.Init()
LCD.Clear(Navy)
time = 0


x = [0]*1000
y = [0]*1000

LCD.Init()

while(1):
    for i in range(0,1000):
        while(flag == 0):
            pass
        flag = 0

        Volts = (a2d2.read_u16() * kV)
        x[i] = i
        y[i] = Volts

    Navy = LCD.RGB(0,0,5)
    White = LCD.RGB(200,200,200)
    LCD.Clear(Navy)
    LCD.Plot(x,y)
    LCD.Title('Heart Rate',White, Navy)

    sleep(1)
```



hw_8a.py

2) Write a Python program which detects each beat

- Flash an LED for 100ms each pulse
- Beep the beeper for 100ms each pulse



```
while(1):
    for i in range(0,1000):
        while(flag == 0):
            pass
        flag = 0

        Volts = (a2d2.read_u16() * kV)
        x[i] = i
        y[i] = Volts

    if Volts > 1.7:
        beeper.on()
        led.on()
    else:
        beeper.off()
        led.off()

    Navy = LCD.RGB(0,0,5)
    White = LCD.RGB(200,200,200)
    LCD.Clear(Navy)
    LCD.Plot(x,y)
    LCD.Title('Heart Rate',White, Navy)

    sleep(1)
```

hw_8b.py

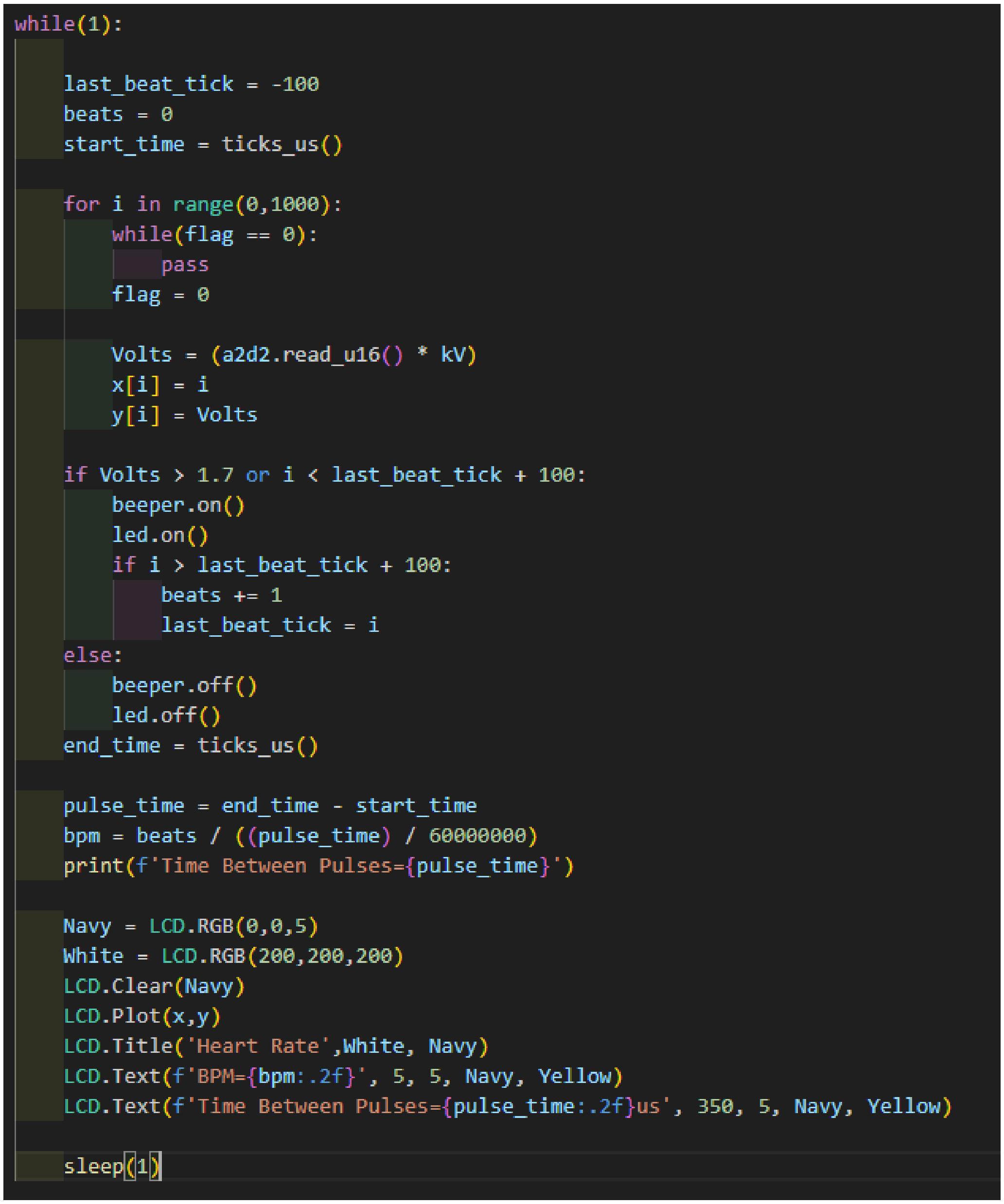
Was having trouble getting the heart rate sensor to read the voltage, so I could not properly test.

3) Write a Python program which measures the time between pulses with a resolution of 1us

- Display the results on the terminal window
- Give the results of your program

4) Modify this program to output on the graphics display

- The measured pulse signal as a graph
- The time between pulses in micro-seconds, and
- Your beats-per-minute, with a resolution of 0.01bpm

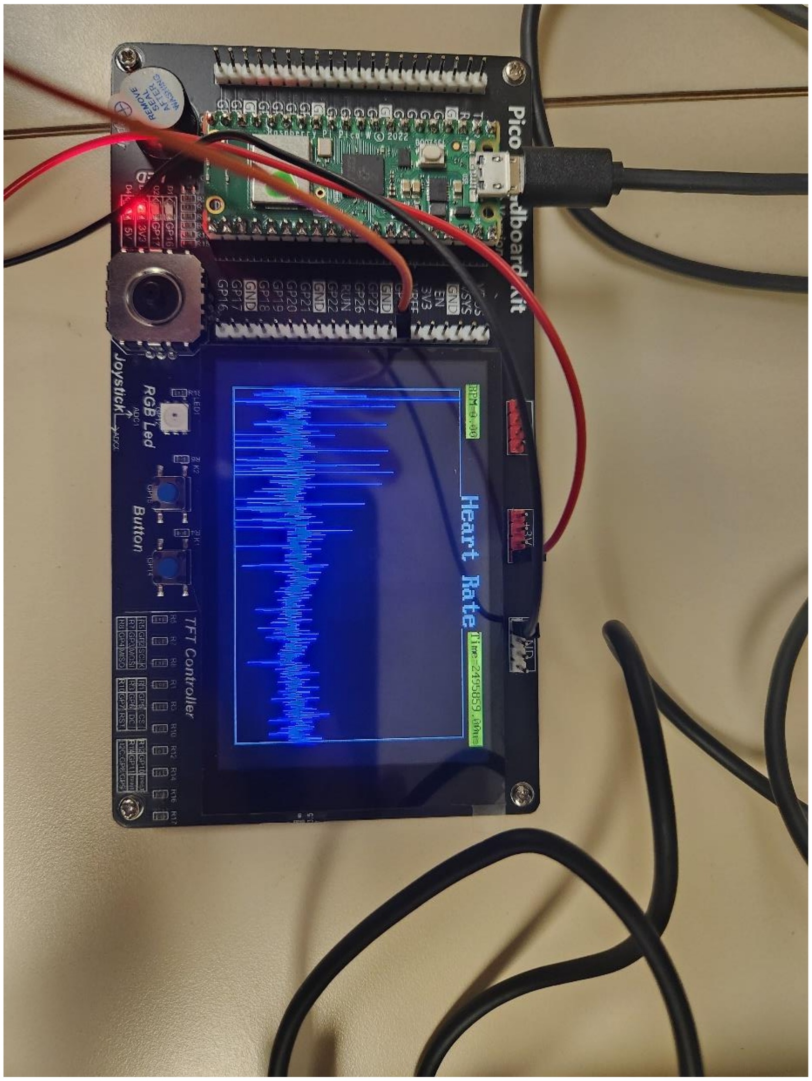


```
while(1):

    last_beat_tick = -100
    beats = 0
    start_time = ticks_us()

    for i in range(0,1000):
        while(flag == 0):
            pass
        flag = 0

        Volts = (a2d2.read_u16() * kV)
        x[i] = i
        y[i] = Volts

    if Volts > 1.7 or i < last_beat_tick + 100:
        beeper.on()
        led.on()
        if i > last_beat_tick + 100:
            beats += 1
            last_beat_tick = i
    else:
        beeper.off()
        led.off()
    end_time = ticks_us()

    pulse_time = end_time - start_time
    bpm = beats / ((pulse_time) / 60000000)
    print(f'Time Between Pulses={pulse_time}')

    Navy = LCD.RGB(0,0,5)
    White = LCD.RGB(200,200,200)
    LCD.Clear(Navy)
    LCD.Plot(x,y)
    LCD.Title('Heart Rate',White, Navy)
    LCD.Text(f'BPM={bpm:.2f}', 5, 5, Navy, Yellow)
    LCD.Text(f'Time Between Pulses={pulse_time:.2f}us', 350, 5, Navy, Yellow)

    sleep(1)
```

hw_8c.py

5) Demonstrate your progam

Since I could not get the demo to work, I instead took a photo of the screen

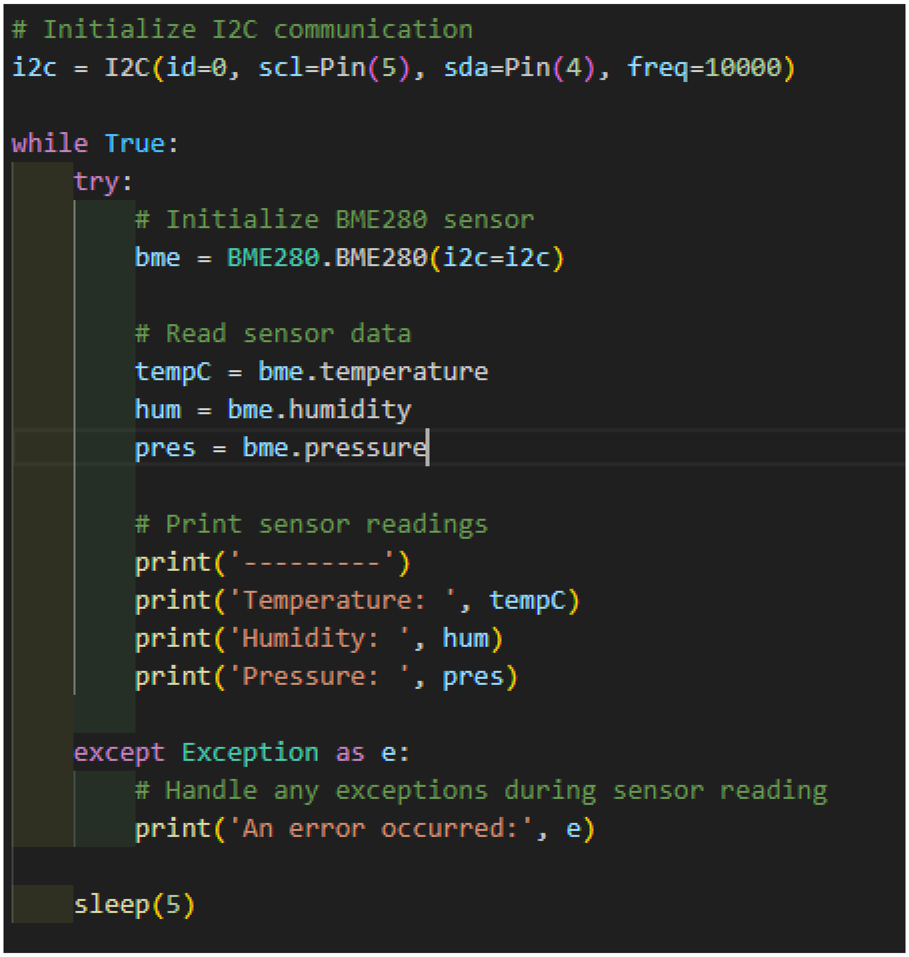


Weather Station

Use a BME280 sensor to measure temperature, humidity, and air pressure. Display this data on the screen.

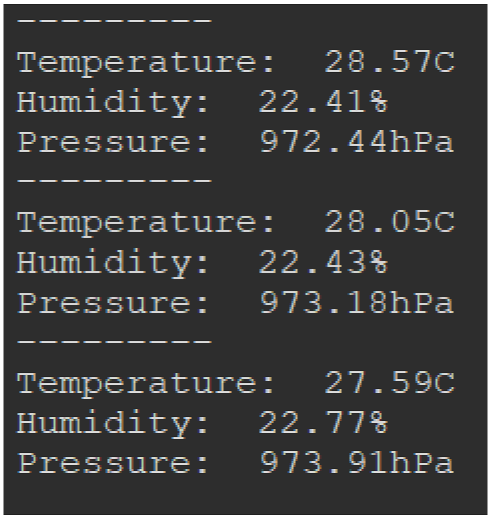
6) Write a Python program to read a BME280 sensor. Display as text on the LCD display

- Temperature
- Humidity, and
- Air pressure



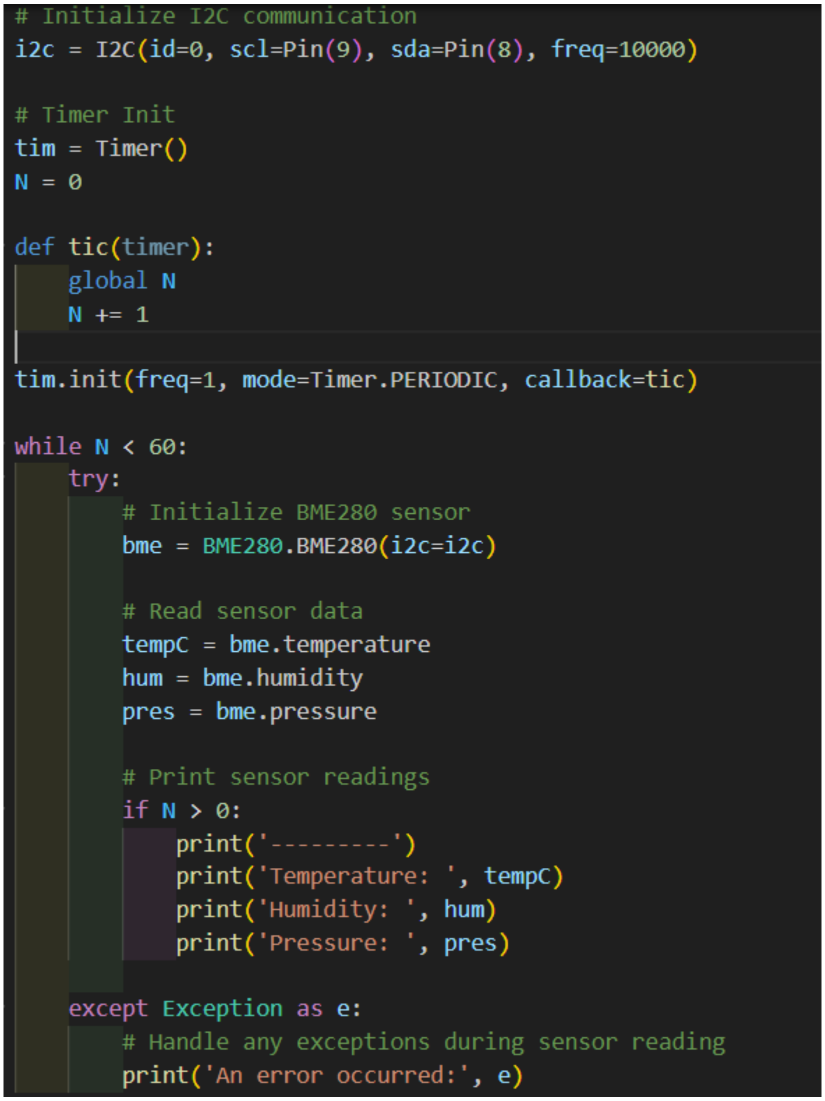
```python
# Initialize I2C communication
i2c = I2C(id=0, scl=Pin(5), sda=Pin(4), freq=10000)

while True:
    try:
        # Initialize BME280 sensor
        bme = BME280.BME280(i2c=i2c)

        # Read sensor data
        tempC = bme.temperature
        hum = bme.humidity
        pres = bme.pressure

        # Print sensor readings
        print('---------')
        print('Temperature: ', tempC)
        print('Humidity: ', hum)
        print('Pressure: ', pres)

    except Exception as e:
        # Handle any exceptions during sensor reading
        print('An error occurred:', e)

    sleep(5)
```

hw_8d.py



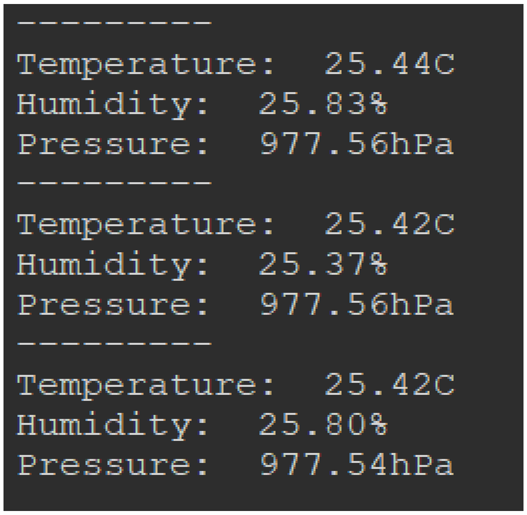
```
---------
Temperature:  28.57C
Humidity:  22.41%
Pressure:  972.44hPa
---------
Temperature:  28.05C
Humidity:  22.43%
Pressure:  973.18hPa
---------
Temperature:  27.59C
Humidity:  22.77%
Pressure:  973.91hPa
```

7) Modify this program to record temperature, pressure, and humidity for one minute with a sampling rate of one second (60 data points). After one second, display the data on the terminal window.



```
# Initialize I2C communication
i2c = I2C(id=0, scl=Pin(9), sda=Pin(8), freq=10000)

# Timer Init
tim = Timer()
N = 0

def tic(timer):
    global N
    N += 1

tim.init(freq=1, mode=Timer.PERIODIC, callback=tic)

while N < 60:
    try:
        # Initialize BME280 sensor
        bme = BME280.BME280(i2c=i2c)

        # Read sensor data
        tempC = bme.temperature
        hum = bme.humidity
        pres = bme.pressure

        # Print sensor readings
        if N > 0:
            print('---------')
            print('Temperature: ', tempC)
            print('Humidity: ', hum)
            print('Pressure: ', pres)

    except Exception as e:
        # Handle any exceptions during sensor reading
        print('An error occurred:', e)
```

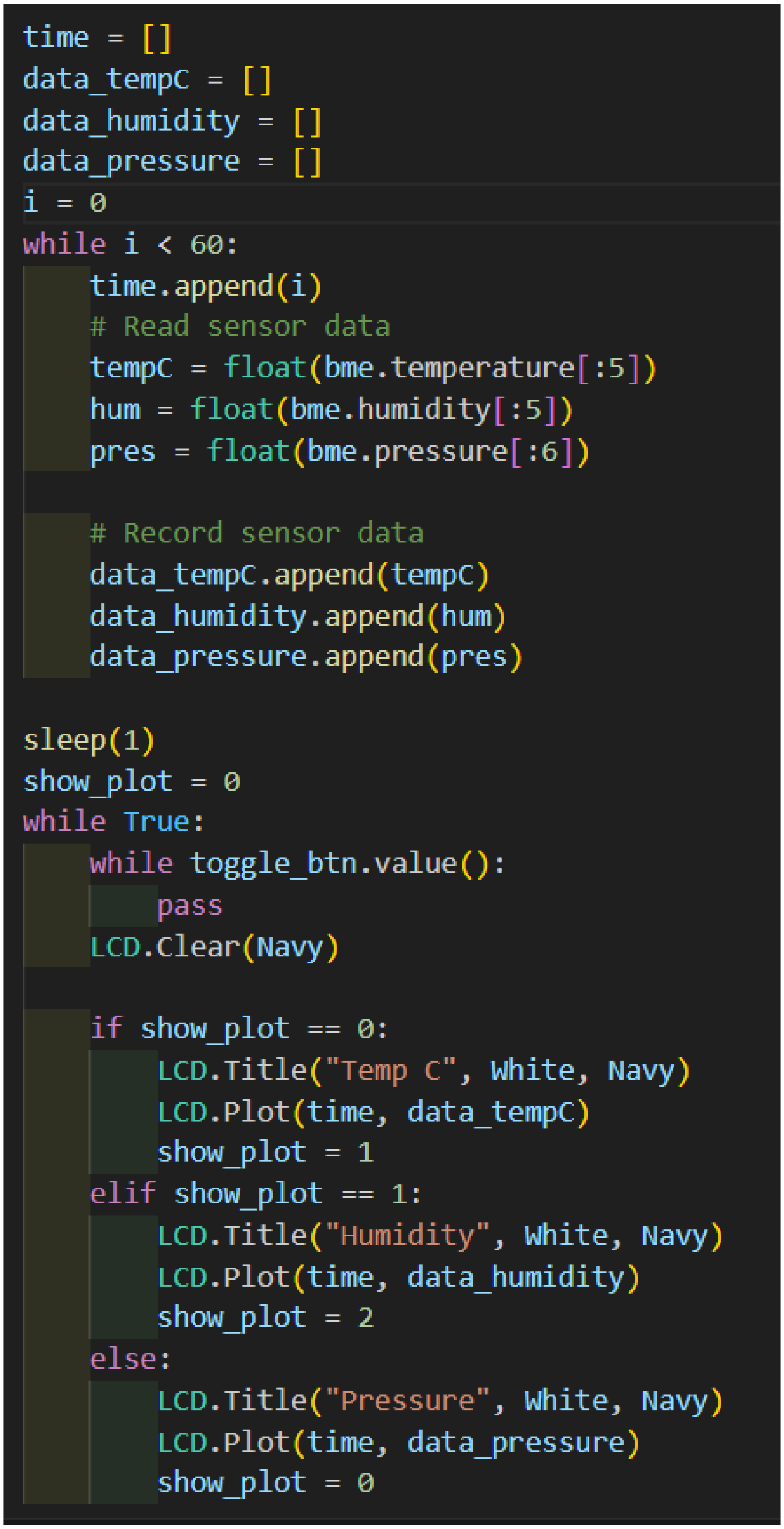
hw_8e.py



```
---------
Temperature:  25.44C
Humidity:  25.83%
Pressure:  977.56hPa
---------
Temperature:  25.42C
Humidity:  25.37%
Pressure:  977.56hPa
---------
Temperature:  25.42C
Humidity:  25.80%
Pressure:  977.54hPa
```

8) Modify this program to display each of these data sets as a graph on the LCD display,

- Start with displaying temperature for 60 seconds
- Each time you press GP15, you switch to the next data set (Temperature >> humidity >> pressure >> repeat)
- Give the results after one minutes of data collection (60 points)



```
time = []
data_tempC = []
data_humidity = []
data_pressure = []
i = 0
while i < 60:
    time.append(i)
    # Read sensor data
    tempC = float(bme.temperature[:5])
    hum = float(bme.humidity[:5])
    pres = float(bme.pressure[:6])

    # Record sensor data
    data_tempC.append(tempC)
    data_humidity.append(hum)
    data_pressure.append(pres)

sleep(1)
show_plot = 0
while True:
    while toggle_btn.value():
        pass
    LCD.Clear(Navy)

    if show_plot == 0:
        LCD.Title("Temp C", White, Navy)
        LCD.Plot(time, data_tempC)
        show_plot = 1
    elif show_plot == 1:
        LCD.Title("Humidity", White, Navy)
        LCD.Plot(time, data_humidity)
        show_plot = 2
    else:
        LCD.Title("Pressure", White, Navy)
        LCD.Plot(time, data_pressure)
        show_plot = 0
```

hw_8f.py

9) Demonstrate your program

- In-person or video

See Weather_Station.mp4