CODE OPTIMIZATION AND DEBUGGING

ALERT MANAGEMENT SYSTEMS FOR FISHER MAN

INTRODUCTION:

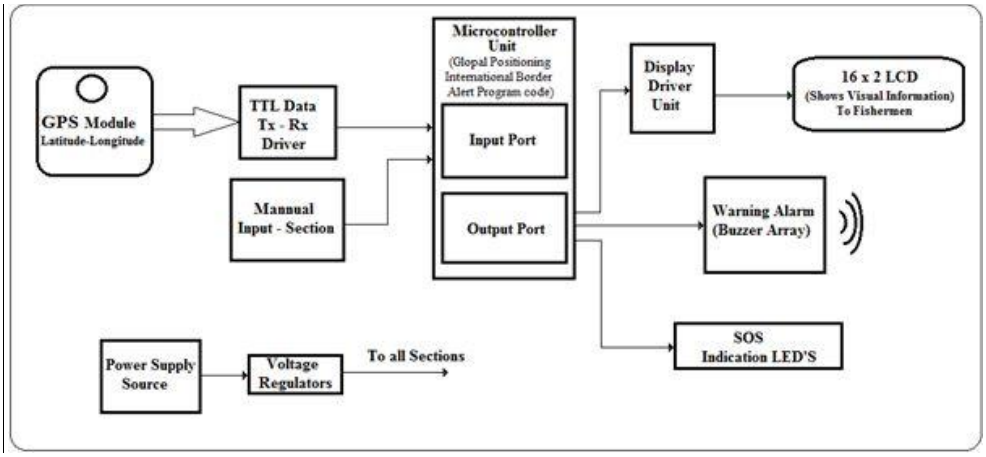
To save the lives of fishermans from the sudden weather changes and from crossing the borders of the neighbouring nations. To avoid these problems i created alert systems using Python programming language.

BORDER ALERT:

Increasing tension across the borders cause many conflicts among countries and India, Sri Lanka is no exception. The sea border between countries is not easily identifiable. The problem of Indian fishermen unintentionally crossing the border and getting abducted remains unsolved. Further, the inevitable harsh climate and the path misleading has cost many lives. Based on a census, so far in 2022, around 300 Indian fishermen have been apprehended. About 30% of the fishermen family has lost their beloved ones. We can design an embedded system

that notifies the fishermen about borders using GPS (global positioning system). We use the GPS receiver to find the location of fishermen and the longitude, latitude values are sent to micro-controller. The fishermen are alerted a few meters before the borders. On further moving the engine will stop.

BLOCK DIAGRAM:



PROGRAM:

```
import random
lat = random.randint(0, 1000000000000)
lon = random.randint(0, 1000000000000)
print (" latitude : ",lat ,"\n", "longitude : ",lon)
gpsValue = lat+lon
print (" gps loc value =",gpsValue)
if (gpsValue > 1000000000000) :
  print(" ALERT! YOU ARE CROSSING THE BORDER")
else :
  print(" KEEP GOING YOU ARE SAFE !")
```

```
 latitude :  22339590466
 longitude :  436525439630
 gps loc value = 458865030096
 KEEP GOING YOU ARE SAFE !
```

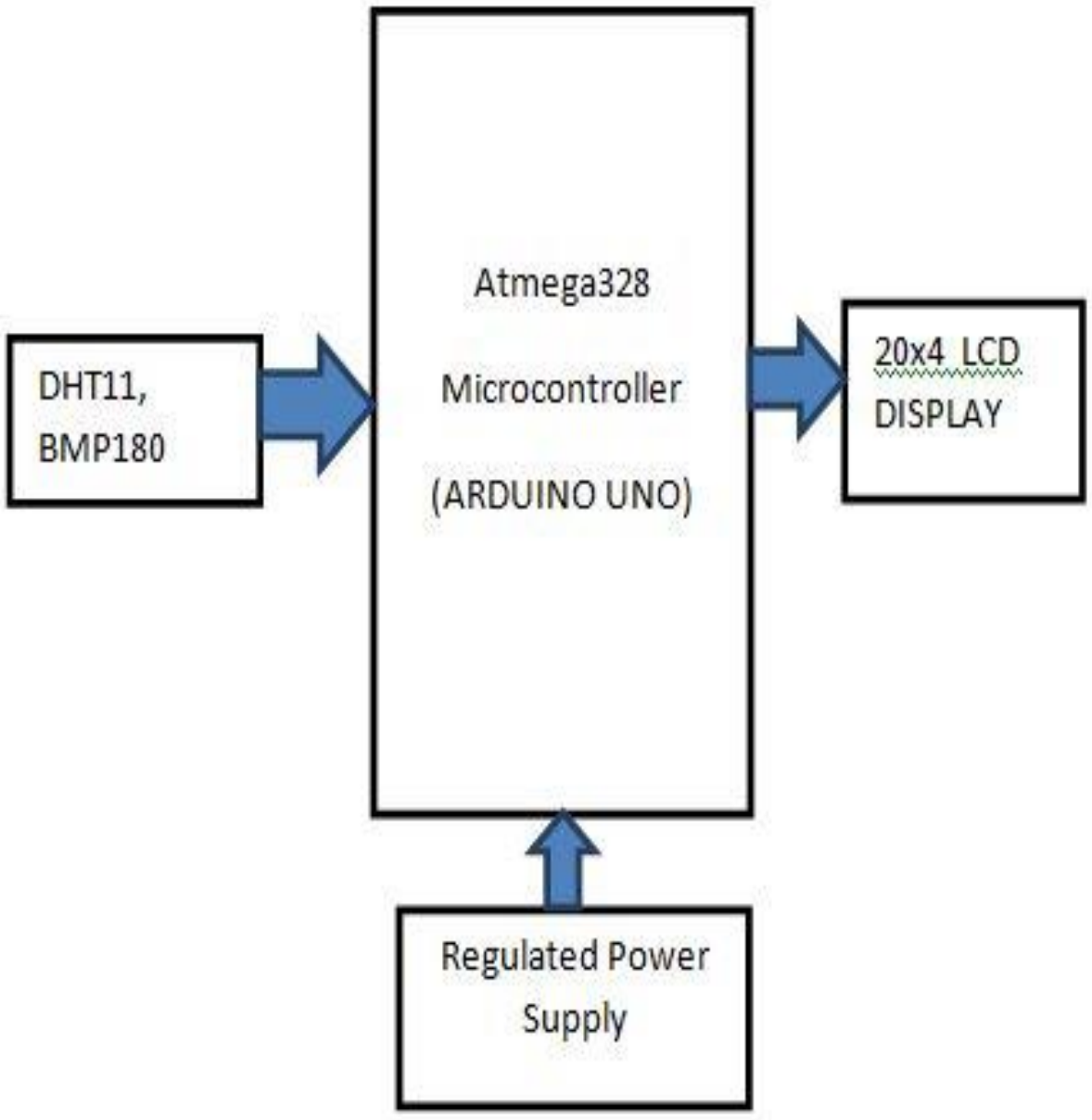
```
import random
lat = random.randint(0, 1000000000000)
lon = random.randint(0, 1000000000000)
print (" latitude : ",lat ,"\n", "longitude : ",lon)
gpsValue = lat+lon
print (" gps loc value =",gpsValue)
if (gpsValue > 1000000000000) :
  print("ALERT ! YOU ARE CROSSING THE BORDER")
else :
  print("KEEP GOING YOU ARE SAFE !")
```

```
 latitude :  212410947371
 longitude :  839460622022
 gps loc value = 1051871569393
ALERT ! YOU ARE CROSSING THE BORDER
```

WEATHER ALERT:

Area based alarm administrations are basic parts for fishermen, because of terrible atmosphere conditions.On considering the issue we proposed a minimal effort and simple climate Alert framework for angler's which is utilized to follow their family members, companions and other anglers in the event that some angler confronting any issues like unexpected climatic changes or crises mean this framework will support the angler.

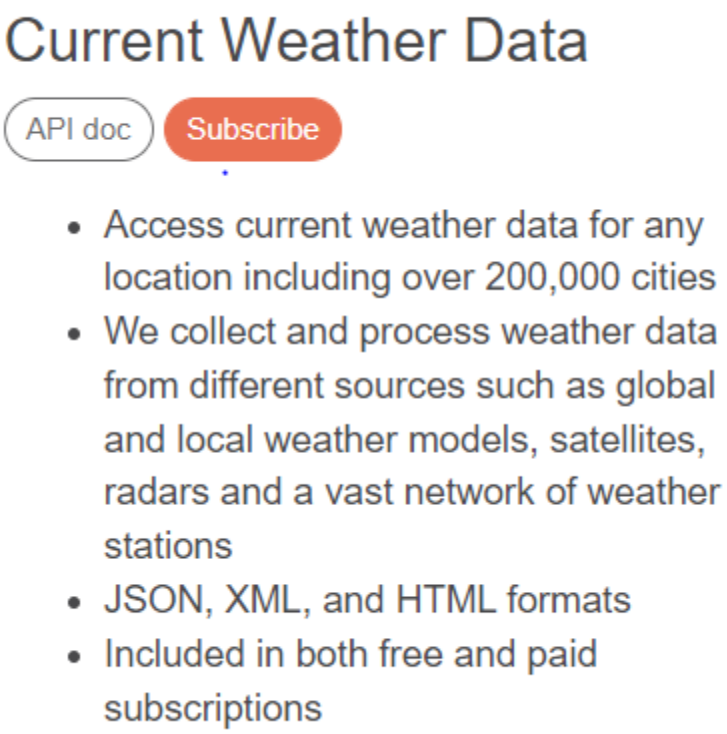
BLOCK DIAGRAM:



STEPS TO WRITE PROGRAM FOR WEATHER ALERT SYSTEM

STEP 1:

Open https://openweathermap.org/api



STEP 2: BASE URL

```
https://api.openweathermap.org/data/2.5/weather?
```

STEP 3: GENERATE API KEYS

You can generate as many API keys as needed for your subscription. We accumulate the total load from all of them.

Key	Name	Status	Actions
7f1b1d0cec406f012a56ecfae6f347b3	Default	Active	
6717b8305411335d42df5105b1cf2ceb	kiran	Active	

Create key

API key name

Generate

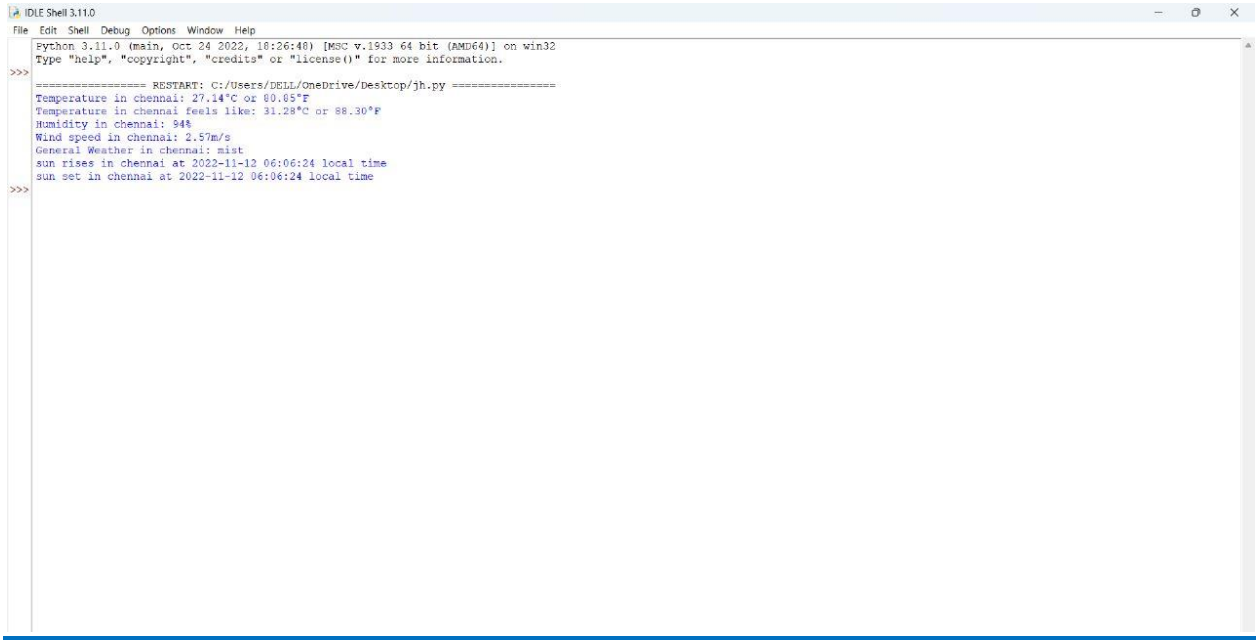
PROGRAM:

jh.py - C:/Users/DELL/OneDrive/Desktop/jh.py (3.11.0)

File Edit Format Run Options Window Help

```
import datetime as dt
import requests
from plyer import notification


BASE_URL ="https://api.openweathermap.org/data/2.5/weather?"
API_KEY ="6717b8305411335d42df5105b1cf2ceb"
CITY ="chennai"


def kelvin_to_celsius_fahrenheit(kelvin):
    celsius = kelvin - 273.
    fahrenheit = celsius *(9/5) + 32
    return celsius, fahrenheit


url = BASE_URL + "appid=" + API_KEY +"&q=" +CITY
response = requests.get(url).json()

temp_kelvin = response['main']['temp']
temp_celsius,temp_fahrenheit = kelvin_to_celsius_fahrenheit(temp_kelvin)
feels_like_kelvin = response['main']['feels_like']
feels_like_celsius,feels_like_fahrenheit = kelvin_to_celsius_fahrenheit(feels_like_kelvin)
wind_speed = response['wind']['speed']
humidity = response['main']['humidity']
description = response['weather'][0]['description']
sunrise_time= dt.datetime.utcfromtimestamp(response['sys']['sunrise']+response['timezone'])
sunset_time = dt.datetime.utcfromtimestamp(response['sys']['sunset']+response['timezone'])

print(f"Temperature in {CITY}: {temp_celsius:.2f}°C or {temp_fahrenheit:.2f}°F")
print(f"Temperature in {CITY} feels like: {feels_like_celsius:.2f}°C or {feels_like_fahrenheit:.2f}°F")
print(f"Humidity in {CITY}: {humidity}%")
print(f"Wind speed in {CITY}: {wind_speed}m/s")
print(f"General Weather in {CITY}: {description}")
print(f"sun rises in {CITY} at {sunrise_time} local time")
print(f"sun set in {CITY} at {sunrise_time} local time")
if (wind_speed>5.8):
    notification.notify(title="WIND WARNING",
                        message="Unstable weather please retreat",
                        timeout=10,
                        app_name="Live Weather Update")
elif(humidity>70):
     notification.notify(title="HUMIDITY WARNING",
                        message="Humidity higher than 70% please retreat",
                        timeout=10, app_name="Live Weather Update")
```

OUTPUT:



ALERT MESSAGE: