SRI HARI A S
sriharias2204@gmail.com

<u>Real-Time GPS Tracking System Using Raspberry Pi - 4</u>

<u>Outcome</u> – By executing the following program, we will get the coordinates of our current location, the coordinates include our:

- Latitude
- Longitude
- Altitude
- Fix quality
- Number of satellites
- Horizontal Dilution of Precision (HDOP)

"Fix quality" refers to the quality or reliability of the GPS position fix obtained from the satellite signals. It indicates the level of confidence or accuracy associated with the reported position. The fix quality is typically provided as part of the NMEA (National Marine Electronics Association) sentences output by GPS receivers.
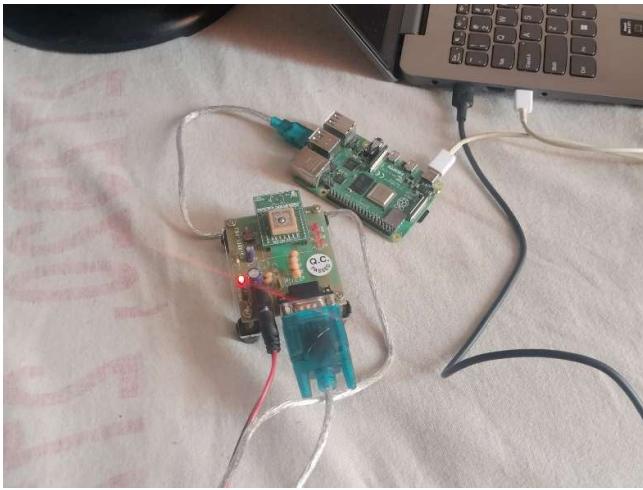
"Number of satellites" refers to the count of satellites from which the GPS receiver is receiving signals at a given moment. In the context of GPS data, the number of satellites provides information about the quality and reliability of the position fix obtained by the GPS receiver.

"Horizontal Dilution of Precision" (HDOP) is a measure of the geometric quality of the satellite constellation relative to the receiver's position. It indicates the degree of precision degradation in the horizontal position fix due to the satellite geometry.

Components used-

Raspberry pi 4	
L80 GPS Module with base board	
Usb to RS232 converter	
Power cables for Raspberry pi4 and GPS module	

Overall Connection:

Power cables connected to the Raspberry pi and the GPS moule,
USB to RS232 convertor is used to connect the Raspberry pi 4 and
the GPS module.

Reason for using Raspberry pi 4:

Raspberry Pi 4 involves GPS data acquisition, processing and provides a flexible and cost-effective solution with ample opportunities for experimentation, prototyping, and development.

Getting started with software the setup –

There are a few python libraries that has to be installed for the proper working of the program.

In the terminal use this command:-

<div align="center"><strong style="color:green">"pip install pyserial "</div>

This command is used to interface python with the serial ports of the Raspberry pi 4.

<div align="center"><strong style="color:green">"pip install pynmea2"</div>

pynmea2 is a Python library designed to parse NMEA sentences, which are standard data formats used in GPS and marine navigation systems. NMEA sentences contain information such as GPS coordinates (latitude and longitude), time, speed, altitude, and more.

After installing pyserial, we can run the python code given.

In case the program throws an error of version compatibility issue, it is recommended to check the version of pyserial and install the correct version.

Steps for doing it is (in the terminal window):
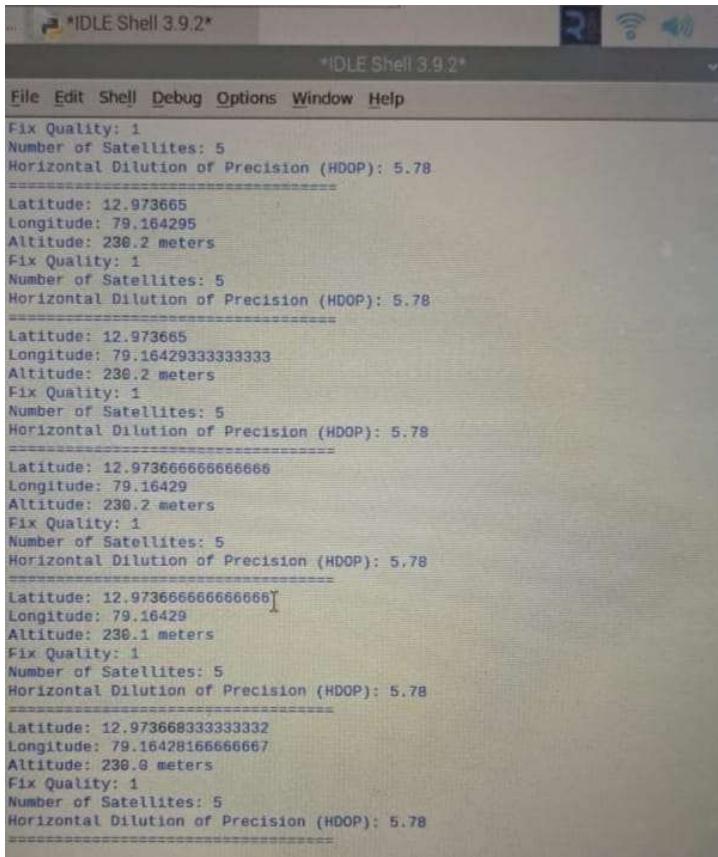
<div align="center"><strong style="color:green">"pip show pyserial"</div>

This command displays the current version of the pyserial

Then use this command to install the correct version:

<div align="center"><strong style="color:green">"pip install pyserial==version"</div>

The output is expected to look similar to this image:

DISPLAYING THE COORDINATED IN GOOGLE MAPS AND CONSTANTLY UPDATING MAPS

Outcome: By executing the given programs, the location coordinates given by the GPS module will continuously be updated in Google Maps.

For the given problem statement, the components used and the connections remain the same.

There a few more python libraries to be installed. We will also be using a HTML and JavaScript program to automatically keep updating the google maps

In the command line:

<p align="center">"**pip install Flask**"</p>

This command is used to install the Flask library

<p align="center">"**pip install flask-socketio**"</p>

flask-socketio is a Flask extension that adds WebSocket support to your Flask application. WebSocket is a communication protocol that provides full-duplex communication channels over a single TCP connection.

We have to save the HTML file in a templates folder in the same location where the python code is saved.

Steps to create template folder:

<p align="center">"**mkdir /path/to/your/desired/location/templates**"</p>

Use this command to create templates folder in the location where the python code is saved. Replace path/to/your/desired/location with the path where the python code is saved.

Enter the API key you acquired, in the place of "HereIenteredmyAPIKey"

After creating the templates file, the HTML code has to be copied into a new file index_socket.io

Save this file as 'index_socket.html' then move it into the templates folder by using the following command.

<p align="center">"**mv index_socket.html templates/**"</p>

The required python code for this application is given.

After the given steps are followed, run the python file.

- **If any port error occurs make sure that the port specified in the python code and the HTML code are the same.**
- **In the given code port:5000 has been used.**
- **If any problem still persists try using ports 8000, 8080**
- **Make sure to update the latest ports in both the python and the HTML code.**

Open the Web browser and type the given address.

<p align="center">"http://127.0.0.1:5000"</p>

In case the ports have been changed then the corresponding port has to be specified in the address instead of 5000.

If any error related as Page not found or error related to IP address, try changing the DNS of the Raspberry pi.

Use the following commands:

<p align="center">"cat /etc/resolv.conf"</p>

<p align="center">"nameserver 8.8.8.8"</p>

Using Google's public DNS servers.

After making the change in the DNS, the system has to be restarted.

<p align="center">"sudo systemctl restart networking"</p>

After executing these steps in case there are any errors, the program will work properly and the Maps will keep updating the location every time there is any change in location received by the GPS module.