

# GPS Data Logger and Data Visualizer

<sup>1</sup>Shubham Rajendrakumar Ware, <sup>2</sup>Mehul Mahesh Jain, <sup>3</sup>Shruti Sanjay Bharati

<sup>1,2,3</sup> Information Technology, Terna Engineering College, Nerul, India

\*\*\*

## Abstract –

Many GPS Data Loggers and Visualizers are available to the public to process the data in the background and give the finished output to the user. This approach makes the data gathered opaque to the user and hence the data cannot be used for implementations other than designed by the software which is in use. This makes the availability of the data to the user crucial for advanced modification or analysis. There are few projects which use this concept of making the data available to the user. By keeping the software scalable and robust, we can implement further analytical solutions to the data. This paper mainly contributes to these points while changing the traditional opaque method to the new scalable, portable and robust structure.

**Keywords – Visualizers, Interpolation, Dataset, Protocols**

## I. INTRODUCTION

Positioning System to identify the location of a vehicle, person, or item with which it is attached and records that location at regular intervals. The data for GPS modules originates from satellite locations; with a growing number of satellites linked to GPS, more exact data may be collected. The method works by determining the time it takes packets to get at the GPS device by analyzing the distance between multiple satellites. The recorded position data can be stored in the tracking unit or sent to a central location database or an internet-connected computer via the device's cellular (GPRS), radio, or satellite modem. This enables the asset's location to be represented against a map backdrop in real-time or after the track has been processed using specialized software. A GPS tracking system can work in several different ways. GPS devices often are employed in the commercial sector to track the location of cars as they move. Some technologies may save the data within the GPS tracking system (known as passive tracking) while some others routinely transfer the data to a centralized database or system via a modem within the GPS unit. A GPS tracking unit is a device that uses the Global

## II. LITERATURE SURVEY

[1] When utilising computer languages, visualising GPS data might be difficult. Some Python packages may be used for this. In certain circumstances, installing and using them might be difficult if you simply have a small task to do. This post will demonstrate a simple and effective approach for visualising GPS data using Python and Open Street Maps. There are three major components that must be included, 1. Map image - a map in an image format such as.png, .jpg, etc. 2. GPS records - records made up of (latitude, longitude) pairs. 3. Geographical coordinates - converting pixels to geographic coordinates. Simply choose a different region of the map by choosing "Manually select a new area." In this scenario, write down the coordinates of the top left and bottom right corners (45.8357,15.9645) and (45.6806,16.1557). This article showed a basic yet efficient way for visualising the GPS dataset. To create a framework, we used Python and OSM. [2] This research focuses on a methodology for assessing the accuracy of portable, common GPS devices for vehicle chase. The experiment was dispensed employing a GPS tracking device BT-338(X) on vehicles adored cars, bikes, buses and it recorded the vehicles' geographical position, velocity, and altitude within the road network. Information pre-processing and improvement are required, yet supplemental information. The GPS observation data accurately determined the cars' true whereabouts. There have been no obvious intrusive factors in the tests' environment, such as densely settled areas, woods, magnetic fields, etc. However, the somewhat poor identification of the bike' position by trees and in the proximity of magnetic fields demonstrates that the GPS' positional quality is extremely liable to the environment. Brakatsoulas et al.(2005), Taylor et al. (2006), and Quddus et al. (2006) conferred map-matching techniques to resolve this downside (2007). We have a tendency to feel that location-based GPS chase information may be valuable for routing, quality pattern detection, and general journey navigation. However, in situations demanding great geographical precision, such as parking, emergency rescue, and the like, the accuracy is insufficient. The limited effective lifespan of GPS devices is a disadvantage (Ryan et al. 2004). The data gathering portion of the experiment in this research lasted at most two hours; the length of the devices is unimportant here because the operating time for the device is around 11 hours after being fully replaced and in continuous mode. [3] The primary goal of this review article is to address many challenges that arise while processing GPS tracking data and, as a result, establish a basic strategy for data processing. The software code used in the

procedure is available for public use as a reference for anybody interested in using it. The study makes use of real-world GPS tracking data from 300 automobiles that were initially acquired to examine CO2 emissions caused by commerce.

The procedure applies methods to match GPS data with the road network in order to improve the data performance based on a clear definition of movement. [4] For a completely unrelated objective, I conducted research on shrinking and zooming digital images using the interpolation process. This article provides zooming applications based on geometric and linguistic zooming algorithms, such as classic zoom, spotlight zoom, and glance zoom. In image processing, picture zooming is a critical technique. In general, zooming entails two steps: The addition of additional element locations and, as a result, the attribution of a grey level to those places. The next step in image downsizing is row and column deletion, which is the analogous process of pixel replication in these cases. Zoom-in and zoom-out features were dynamically enforced in the file zooming programmes. The visual outcome of the additive interpolation algorithm is superior than the closest purpose interpolation method since it is continuous.

### III. PROPOSED SYSTEM

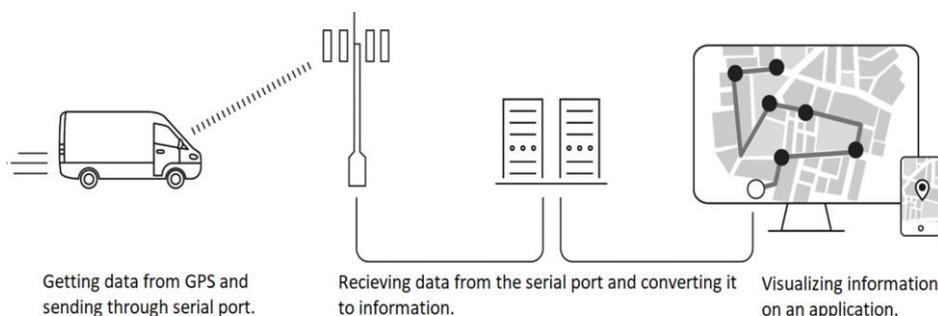


Fig: Proposed Block Diagram.

The proposed system is to get the data through a GPS sensor. Better the sensor, more accurate the reading will be logged. We can install the sensor on any entity such as a logistics unit and track the path travelled by the device in given time. After installing the sensor to the object the data is then captured using an API where the noise will be reduced and stored to a file. We log the data through a serial port and store it in a .csv file. The logging parameters will be taken from the user with the help of a user interface developed which is scalable and robust and has space to add features. A visualization feature will be integrated in the user interface for ease of access of features and the user does not have to access different softwares. We can implement many extra features for data representation, here we visualize the path taken by the device with a simple arrow starting from the origin to the endpoint of the data logged. Hence getting a full route of the device travelled and then can be used for further analysis.

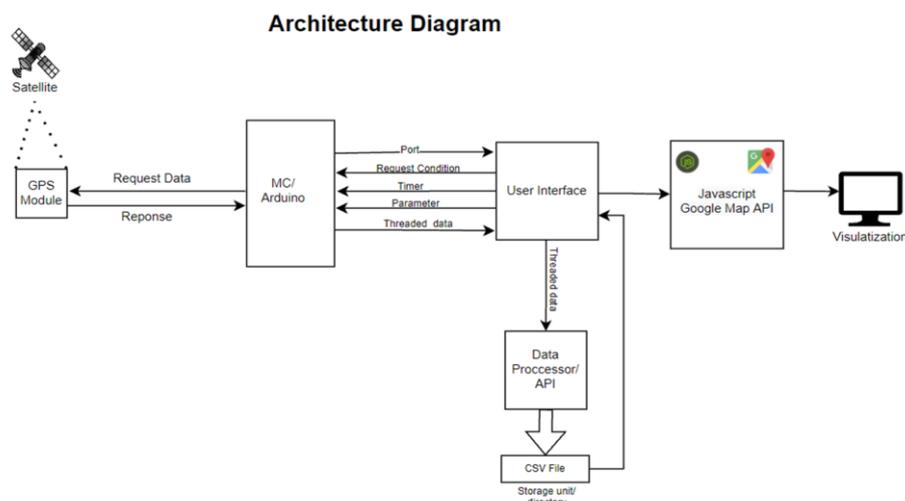


Fig: Architecture diagram.

The system has 7 different sub modules, namely GPS sensor, microcontroller, User Interface, File reader, Storage unit, Gmaps Javascript API, Browser. Each has its own independent functionality and hence multiple hardwares are compatible with the system.

#### IV. IMPLEMENTATION

*Technologies used -*

The system consists of 3 main components *i.e.* GPS Module, User Interface and Visualizer.

##### 1. GPS Module –

- In this project we had used a Ublox Neo 9M GPS Module.
- It is a cheap and portable module.
- It has excellent RF integration mitigation.
- Maximum position availability with concurrent reception of 4 GNSS.
- Low Power consumption.

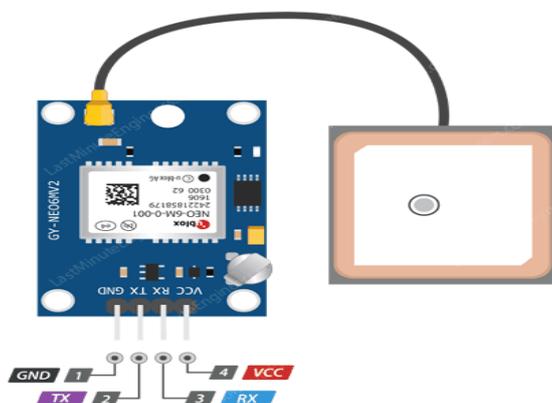


Fig: GPS Ublox Neo 9M.

This device may be used for asset tracking, telematics, navigation, wearables, cameras, unmanned vehicles, automotive, road vehicle navigation, and autonomous driving, among other things. Users of GNSS anticipate instant position data, which is not always achievable due to poor signal circumstances; as a result, location, help, and correction services have become critical features. Ublox has the most innovative solutions for these issues.

##### 2. User Interface –

For this project we chose C# as the programming language as it consists of many built in libraries for data manipulation, cross threading, serial port communication and Windows Form Format. In the user interface there consists of 2 modes for selection of logging modes. The inputs are handled by event handlers and the interface is dynamic. Options for selecting the multiple logging formats are available such as only coordinates, only coordinates and time and coordinates, time and altitude. After selection of logging format, we have to specify the interval between the logging of data will take place. Selection of serial port on which the device is connected to the computer on which the software is installed in case multiple devices are connected to the system. Lastly an export button to save the data logged by the software in a specified location on the system memory.

After selection of all these parameters we can start the script to log and save the data. Pressing the stop button will terminate the script from logging. A Visualize button to start the visualization of the logged data in a separate window of any browser using Google Maps JavaScript API. This will summarize the functioning of the features provided in this software.

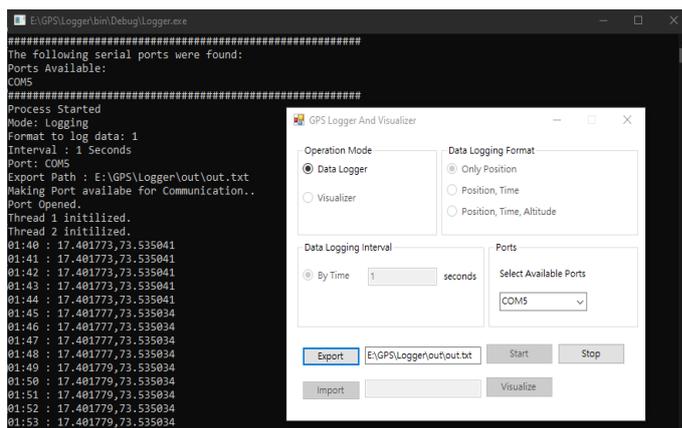


Fig: Data Logging and Visualizer User Interface.

A console window for monitoring the data logged in real time and to keep track of errors occurring during runtime in case any module fails.

### 3. Visualizer –

Google Maps provides a javascript API to load an array and to implement using Polylines. This paper uses javascript to get the data logged from a .csv file and loads it into the API. We can add multiple overlays to make the data visualization more detailed.

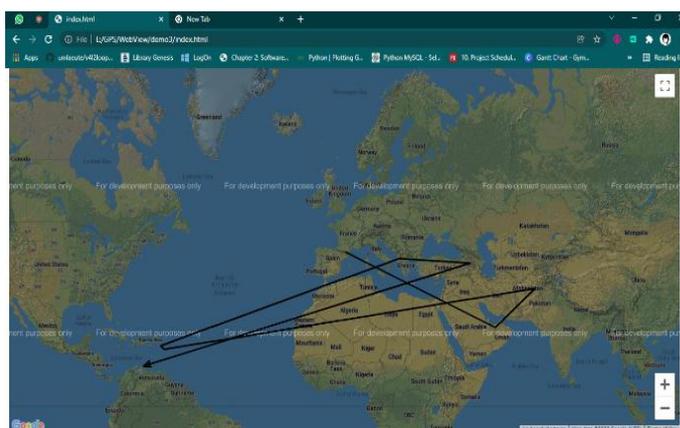
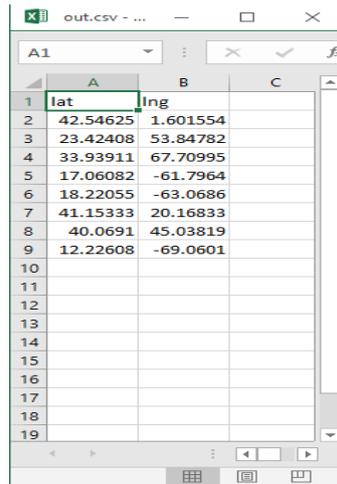


Fig: Data Visualizer.

### 4. File structure .csv –

.csv is a standard file structure to save comma separated values in a file for ease of access and simple file handling. .csv file reader can be implemented in any programming language and hence can be used by multiple programs to analyze the data. In this paper the .csv is passed to a javascript which further reads and interprets the data and visualizes it in browser windows using HTML and CSS.



	A	B	C
1	lat	lng	
2	42.54625	1.601554	
3	23.42408	53.84782	
4	33.93911	67.70995	
5	17.06082	-61.7964	
6	18.22055	-63.0686	
7	41.15333	20.16833	
8	40.0691	45.03819	
9	12.22608	-69.0601	
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			

Fig: .csv File structure.

### Methodology –

GPS sensor and microcontroller are connected using RX, TX, GND, VIN. Logged data from the GPS are communicated to the microcontroller as a Request and Response protocol. The microcontroller requests data from the sensor while the gps is gathering the data through the satellites. This data stream is running from the power up moment till the power is cut off to the microcontroller. As the data request sequence can be initiated by the UI anytime, the sensor cannot be kept in a standby mode.

The microcontroller and the user interface has sub functions i.e port communication, request sender, timer, parameters, threaded communication. The user interface checks on which port the microcontroller is connected and establishes a connection to communicate. The user interface and the serial communication is established on two different threads which helps in two separate functions to run simultaneously hence seamless running of the program. If both the tasks are carried out on the same thread the program will run a single function at a single time which results in data loss and a StreamWriter cannot be initialized on the same thread as the data is still in receiving mode. We need the StreamWriter to be run parallelly as if the connection fails during logging the data received from the sensor does not get lost. We run the StreamWriter parallel for live data saving as soon as a valid data entry occurs in the data stream of the serial port. The first thread sends the request to the microcontroller with the parameters selected by the user on the user interface. This thread is solely used for the base program and the send request, no data logging is done on this thread, while the second thread is used for logging the data from the sensor. A delegate is used to transfer the data for processing from the second thread to the first. A timer is started on the start of the program to keep the time parameter in check and to rearrange the coordinates incase of logging errors. As the microcontroller can either send or receive at a single time, hence we developed a protocol such that the stream is in response mode until a new set of parameters is not sent to the microcontroller. The stream is checked for an end bit to confirm the end of a single stream. This confirms a meaningful response from the microcontroller and now can be written by the StreamWriter.

The StreamWriter has access to the storage units and enables customized saving of files in any directory on the disk. The saved file is then provided to a javascript which converts it into a json string. This string can be used by Gmaps API to visualize in a certain way. In this case a line connecting the coordinates on Google Maps.

## V. TECHNICAL CHALLENGES

### Threading –

Getting the data from the serial port in a thread and using cross threading to use the serial data to write in a file. This challenge was overcome by using a delegate to temporarily save in and further forwarding to another function for processing as direct handling of data from a cross-thread is not possible.

### GPS Locking –

The GPS sensors have modes to get a fix on the satellites. Generally it takes around 1-2 minutes for a hot start. Hot start is a condition when the sensor has got a fix/lock on a satellite in the past sometime. While a fresh start means the sensor has no prior knowledge of the positions of the satellites hence it takes considerable time to get the first lock on a satellite.

Generally it takes around 5-10 minutes for a fresh start in a metropolitan city where there are many servers which save the location of the satellites and help in getting the first fix while the backup servers are less in rural and distant areas which contributes to large fix time. This fix/lock time can be between 30 mins to 1 day. The next fix is frequent and depends on the availability of satellites and weather conditions in the area. The sensor data drastically depends on the weathering conditions and plays a crucial role in accuracy and noise volume.

## VI. CONCLUSION

We have concluded the non-opaque methodology will give more analytic features to the user. This enabled the user to access the data and make custom representations to it for user friendly visualization. This system allows the developers and users to generate their personalized data, then to integrate this data through the use of sophisticated visualization tools. Furthermore, the approach enables the data to be shared and compared with expert systems and AI. The system uses off-the-shelf infrastructure, protocols and a newly developed prototype which allows gathering of data and then can be analysed.

## VII. REFERENCES

- [1]Leo Tišljarić , "GPS-visualization-Python" 2021 MIT License Copyright (c) 2021
- [2] 42 Broadway Suite 12-400, New York, NY, 10004, US© 2021 The ThingsBoard Authors
- 3]Nikolai Janakiev " Geospatial-storytelling" The MIT License (MIT) Copyright (c) 2016
- [4]Zhao, Xiaoyun & Carling, Kenneth & Håkansson, Johan. (2014). Reliability of GPS based traffic data: an experimental evaluation. 10.13140/2.1.2960.2242.
- [5] Cai, Li & Zhou, Yifan & Liang, Yu & He, Jing. (2018). Research and Application of GPS Trajectory Data Visualization. Annals of Data Science. 5. 10.1007/s40745-017-0132-1.
- [6]Shveta Chadda, "Zooming Techniques for Digital Images: A survey" IJCST Vol. 3, Issue 1, Jan. - March 2012[
- [7]Schwab, Michail & Hao, Sicheng & Vitek, Olga & Tompkin, James & Huang, Jeff & Borkin, Michelle. (2019). Evaluating Pan and Zoom Timelines and Sliders. 10.1145/3290605.3300786.
- [8] Wang, Junqiu & Yagi, Yasushi. (2008). Integrating Color and Shape-Texture Features for Adaptive Real-Time Object Tracking. IEEE transactions on image processing : a publication of the IEEE Signal Processing Society. 17. 235-40. 10.1109/TIP.2007.914150.
- [9]Yang sa, "Improved bilinear interpolation method for image fast processing" intelligent computation technology and automation (icicta), 2014 ,7th international conference , oct. 2014, pp. 308-311