

Real Time Vehicle Location Finder

Preety Singh

B.E. Final Year Student, Computer Science and Engineering, Government Engineering College, Bilaspur, India

Abstract: Locating a parked vehicle in a big parking ground has become an elusive task that troubles the vehicle owner and parking management equally. Though, the alarm system of some modern cars helps a bit, other vehicles generally do not have such inbuilt embedded feature in it. The problem becomes even worse at night time. This project proposes an IoT solution to the aforementioned problem that lets the user find the real time location of their vehicle and trace the path via map. It also shows an effective solution to one of the problem statements of GIH (Gujrat Industrial Hackathon) 2018-19, problem ID- GIH144. To get the real time location of vehicle we have made use of GPS Technology. To fetch the GPS data, we used Neo 6m GPS module that gives raw data in the form of NMEA Sentences. Interpretation and processing of this data is done through NodeMCU which is an open source Lua based firmware for the ESP8266 WiFi SoC from Espressif. To program the microcontroller i.e., coding, compiling and ultimately flashing this code, we used Arduino which is also an open source electronic prototyping platform. In programming, TinyGPS++ library is used to process NMEA sentences, SoftwareSerial library for making serial communication between the GPS module and microcontroller, and functions related to WiFi are handled through the library ESP8266WiFi. To provide user interface to the user, the application made use of Blynk platform.

Keywords: Alarm System, IoT, Real time location, GPS, Neo 6m GPS module, NMEA sentences, NodeMCU, Lua, Firmware, ESP8266, SOC, Flashing, Arduino, Prototyping, TinyGPS++, SoftwareSerial, Blynk.

I. INTRODUCTION

We have all been to parking ground to park our vehicle. How do we find our vehicle in a big parking lot, when there is a big ground having thousands of vehicles and we have to find our bike, scooter or cycle? As the population is increasing tremendously and so is the number of vehicles, the need for a proper management has become inevitable. There are a few probable solutions to this problem:

- **Manually:** We can go about the parking, looking for our vehicle by ourselves based on our memory of where we parked it last time.
- **Traditional Approach:** A person from the parking organization is assigned the task of making a receipt containing some data like vehicle number, row number, date etc., providing us with its copy, maintaining a register entry for the same and will locate our vehicle for us, if presented with this receipt.

First approach is too easy, but could be time consuming. If we don't remember where we parked it or if somehow vehicle moved from its original position, it would lend us in a big trouble. Inspecting each and every vehicle would be no better than BRUTE FORCE APPROACH. Second approach is somewhat better that a system has been implemented, but not all parking has this person, it requires a lot of data logging and delegating this task to a single person would make the system dependent on a single point- if it crashes, whole system will come down.

A better solution can be, to let the vehicle (actually the device embedded in it) take care of the fact where it has been parked, and let you know its current location in a single tap. So, no matter if the vehicle was displaced or attender is unavailable, we need not to bother. Just a single tap on our cell phone and we would know where it is.

The major processes involved to accomplish the objective are:

A. Fetching raw GPS data (NMEA sentences)

- Configuring GPS module (circuit along with antenna).
- PuTTY application – to view the GPS data and check for a fix.

B. Interpretation and Processing of raw GPS data

- Configuring Arduino for NodeMCU board – ESP8266WiFi library.
- Making Serial Communication between GPS module and NodeMCU using SoftwareSerial library.
- Installing TinyGPS++ for interpretation of NMEA sentences and calling related functions.



- Installing Blynk libraries to provide Remote Access and Control functions.
- Programming the microcontroller using Arduino.

C. Provision of User Interface

- Creating application with required Map widget (Google Map).
- Mapping the GPS data processed by microcontroller (location of vehicle) on map.
- Using inbuilt GPS sensor of cellphone to get our location coordinates and mapping it, to draw path.
- Sending the location coordinates via e-mail as a backup.

II. LITERATURE REVIEW

The problem has been solved by many organizations and other companies as well. One of the recent works is done by GOOGLE. Google Maps introduced this feature to save the parking location. To avail this feature, we simply need to tap the blue dot on the map which represents our current location and then click on “Save your parking” on the screen that appears. This location gets logged and remains there until we remove it. We can come back to it following the same map. The feature in itself is amazing, though, has its own shortcomings as:

- We need to save the location before leaving the vehicle in parking. If we forget doing that or run short of time, it cannot help us.
- The location saved, is the location where we were, at the time of parking and it has nothing to do with the location of vehicle. So, it cannot provide us with the real time location of the vehicle.

There are other such works that actually intend to make a vehicle tracking system. In paper [1], they have used AT89s52 microcontroller and a GSM modem along with GPS receiver to transmit the coordinates to the user and the value display includes LCD screen interfaced to the microcontroller. It's a pretty good work as with the help of GSM modem we don't need to rely on Internet but the LCD and GSM add an extra cost to the project making it expensive and since we only get location coordinates as a text message without a map interface we again need to use a third party Maps application like Google Maps which will require Internet. Since, our project uses NodeMCU which has built-in WiFi chip, we don't need GSM modem and the vehicle location can be accessed and monitored on map directly using our cell phone. NodeMCU is offered at a very low cost of around 3\$.

The motivation to this project and research comes from a problem statement of GIH (Gujrat Industrial Hackathon) 2018-19, problem ID- GIH144 which is available on [2].

III. ARCHITECTURE AND WORKING

A. Structural Constructs

1) Technology Used:

- GPS: Global Positioning System is a satellite radio navigation system. It is actually a network of 33 earth orbiting satellites that revolve round the earth making two complete revolutions each day, sending GPS data every 30 seconds. It is accessible to anyone having a GPS receiver, free of cost. An alternative could be GLONASS (GLOBAL Navigation Satellite System).

2) Hardware Used:

- NodeMCU: A low cost, open source, Lua based firmware, has associated development kit based on ESP-12 module of the ESP8266 which is a WiFi SoC. Since the board has embedded WiFi capability, we do not need a separate WiFi shield.
- Neo 6m GPS module: A complete GPS receiver with a built-in ceramic antenna (25 x 25 x 4mm) that provides strong capability of searching satellites, EEPROM and power and signal indicator.
- Jumper wires or one pin connectors (female to female): wires having connector pins at each end, so that it can be used for connecting two points to each other without soldering.
- Power source: The NodeMCU board can be charged via USB power supply or a LiPo (Lithium ion Polymer) battery or any other DC source that can provide 3.3V(as per the rating of board in data sheet) regulated power supply and must not exceed it.

3) Platform Used:

- Arduino: The IDE (Integrated Development Environment) to write, compile and upload code to the microcontroller.



- Blynk: platform that lets user make interfaces for monitoring and controlling electronic hardware projects from iOS or android device.

B. Working Mechanism

1) *Making Use of GPS*: Along with various other data, the two major components of GPS message that GPS satellites transmits are:

- Ephemeris data: this data is used to calculate the position of each satellite in the GPS constellation.
- Almanac: information about the time and status of the entire satellite constellation i.e., where each satellite should be at any moment of time.

2) *Circuit Design and Interfacing*: First of all, we need to solder berg stick on the 4 terminals of GPS receiver. Now, to make serial communication between NodeMCU and GPS receiver so that the GPS data can be processed by microcontroller, we connect them via jumper wires as:

TABLE I CIRCUIT CONNECTIONS

NodeMCU pin	GPS module pin
3V (any)	Vcc (power)
D1 (GPIO 5)	RX (Receiver)
D2 (GPIO 4)	TX (Transmitter)
Ground (any)	G (Ground)

While checking and uploading, we can power up the NodeMCU board via a USB cable connected to our laptop but while fixing it on our vehicle, this assembly would make a standalone device; We need a battery for it. We connect the positive terminal of battery to the Vin of NodeMCU and negative terminal to the any of the Ground pin of the board.

3) *Data Reception and Processing*: The GPS messages are transmitted in the form of microwave signals that can be received by any GPS receiver (actually, the antenna in it). Now, the function of this GPS module is to locate four or more of these GPS, calculate its distance to each of them with the help of their transmitted message (it internally uses a mathematical principle called 3-D Trilateration) and use this information to deduce its own location. Once this calculation has been done, it can tell us the latitude, longitude, altitude and other related information. To view this data on our laptop screen we can either use Serial Monitor of Arduino IDE or use puTTY application at a baud rate of 115200 or 9600 bps(default).

For the very first time, GPS module takes some time called TTFF (Time To First Fix) which is a measure of the time required for a GPS receiver to acquire satellite signals and navigation data, and use it to calculate a position solution known as Fix. This time may vary around 12.5 seconds or above. After a fix is obtained, we get a valid GPS data. This data is fed to the microcontroller for further use.

4) *Programming the Microcontroller*: In Stationary Mode (NodeMCU can connect to an existing WiFi), NodeMCU is programmed using TinyGPS++ library for parsing NMEA sentences, SoftwareSerial library for serial communication of data between microcontroller and GPS module, ESP8266WiFi library for data transmission over WiFi and others so that the device can connect to a WiFi and send this processed data to Blynk Server.

5) *Locating the Vehicle*: The circuit (GPS module along with programmed NodeMCU and power supply) is fixed at the vehicle. We simply need to turn on our cell phone data & open the application. The device needs a WiFi connection, which we can provide through our cell phone or use the WiFi of parking lot. Once the device establishes a connection with WiFi, the application notifies us of it and server asks the device for the GPS data (location of vehicle) and transmits it to the application over Internet. To get the location of user, the inbuilt GPS sensor of user's cell phone is used. Both of these locations are mapped in the application using Google Maps which also directs us to reach the vehicle. The location coordinates of vehicle will also be mailed to us. We can use this location coordinates in mail and put it in any Maps application to get alternate way of locating vehicle in case of any problem with the system, logging this data and can also be used to share it to someone else.

IV. CONCLUSION

We have made a stand-alone device that lets us know the real time location of our vehicle, be it parking or anywhere else and directs us to reach it using our cell phone via map. It is also capable of solving the problem statement of GIH 2018-19, problem ID - GIH144. It can also serve as an anti-theft mechanism for all sorts of vehicle. The solution is low cost with quite simple and convenient user interface. Vehicle owner can use it to locate their vehicle in a huge crowd of parked vehicles anytime without relying on anyone else, monitor it remotely and can also take preventive actions in case of suspect of it getting stolen. If incorporated with the parking management of parking lot, it would make the



parking much more plannable leading to efficient utilization of space, would save a substantial amount of time that could have otherwise been wasted in locating vehicle and at the same time providing a convenient way to reach each of the vehicle and thus a Congestion Free Parking System. The system offers the same accuracy as that of GPS module and the GPS signals, uses low power and makes negligible use of data storage.

ACKNOWLEDGMENT

First, I'd like to thank my project guide **Professor Alka Pandey** (Department of Computer Science & Engineering) for she helped me whenever I was stuck and that she reviewed my work with painstaking attention. I owe a huge debt to her. I am also grateful to my Head of Department, **Professor Sourabh Yadav** (Department of Computer Science & Engineering) for his unending support right from the beginning and correction to my slips. My Head of Institute **Dr. B.S. Chawla Sir** (Principal, Government Engineering College, Bilaspur) also deserve a special mention for motivating us through the entire journey. Finally, my heartfelt thanks to all the people who were part of this work in numerous ways.

REFERENCES

- [1]. Vehicle Tracking System Using GPS and GSM by Ravi Vijay, Sakshi Sharma, Deepak Mali and Rajesh Solanki, 2018 Available: https://www.academia.edu/12234429/Research_paper_on_VEHICLE_TRACKING_SYSTEM_USING_GPS_AND_GSM_
- [2]. Gujrat Industrial Hackathon Problem Statements, 2018-19. Available: http://ssipgujarat.in/gih/problem_statement.php
- [3]. "Exploring Arduino: Tools and Techniques for Engineering Wizardry", Jeremy Blum, July 2013.
- [4]. "The 8051 Microcontroller and Embedded Systems Using Assembly and C", Muhammad Ali Mazidi, 2008.

BIOGRAPHY



Ms. Preety Singh final year student of Bachelor of Engineering in Computer Science from Government Engineering College Bilaspur (CG) Affiliated to Chhattisgarh Swami Vivekananda Technical University, Bilai, 2016-20 batch. A passionate C Programmer and fond of working in Embedded Systems.