Project Title : Public Transport Optimization

Problem:

With our changing habits, people are moving faster because of busy schedules and an increasingly fast-paced lifestyle. As a consequence, this requires smarter ways of managing our lives and maximising our time. ‘Internet of Things’ (IoT) is dramatically accelerating the pace of innovation in the public transportation industry and providing real solutions. Public transport refers to shared passenger transportation services like bus , trains , metro, trolleybus etc. There are many challenges that are faced by passengers while using public transportation . In this project, have a GPS system installed in your vehicles , it helps to find to track a stolen vehicles or it can be track a live location of the vehicle .

Design Component:

Arduino UNO

Wi-Fi Module ESP8266

GPS module

USB Cable

Connecting wires

Laptop

Power supply

16x2 LCD

Bread Board

Wi-Fi router

Working Principle:

O To track the vehicle, we need to find the Coordinates of Vehicle by using GPS module. GPS module communicates continuously with the satellite for getting coordinates.

O Then we need to send these coordinates from GPS to our Arduino by using UART. And then Arduino extract the required data from received data by GPS.

O The Arduino instructs the ESP8266 Wi-Fi Module to configure, connect and obtain the IP address from the router. Then after , the Arduino Board sets up GPS to obtain coordinates, and a “Page Refresh message” appears on the LCD.

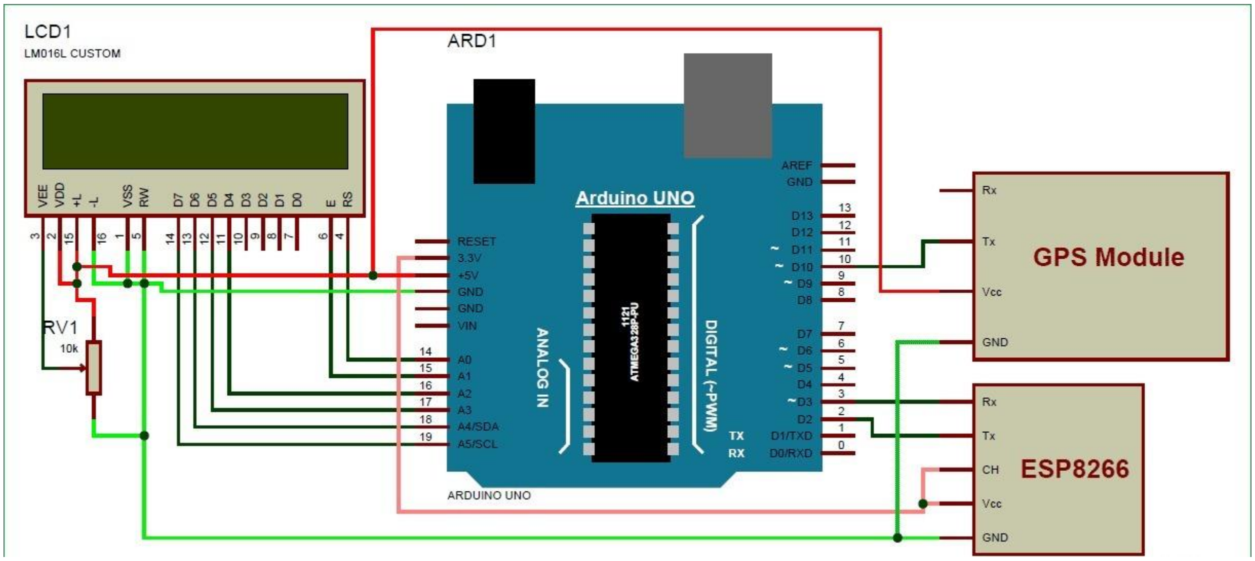
O This implies that the user must reload the page. When a user reloads the website, Arduino retrieves the GPS coordinates and transmits them through WiFi to a webpage (local server) along with some more data and a link to a Google map.

O The user will now receive the vehicle’s current location at the red spot on Google Maps after clicking this link, which will redirect them to the map with the coordinates.

Circuit Explanation:

In this circuit 'Vehicle Tracking using Google Maps project' is very simple and we mainly need an Arduino UNO, GPS Module and ESP8266 Wi-Fi module. There is a 16x2 LCD optionally connected for displaying the status. This LCD is connected at 14-19 (A0-A5) Pins of Arduino.

And the circuit diagram is attached below,



Here Tx pin of GPS module is directly connected to digital pin number 10 of Arduino. 12 Volt adaptor is used to power the GPS Module. Wi-Fi module ESP8266's Vcc and GND pins are directly connected to 3.3V and GND of Arduino and CH_PD is also connected with 3.3V. Tx and Rx pins of ESP8266 are directly connected to pin 2 and 3 of Arduino. Software Serial Library is also used here to allow serial communication on pin 2 and 3 of Arduino.

ESP8266 has two LEDs, one is Red, for indicating Power and second is Blue that is Data Communication LED. Blue LED blinks when ESP sends some data via its Tx pin. Also, do not connect ESP to +5 volt supply otherwise your device may damage. Here in this project, we have selected 9600 baud rate for all the UART communications.

Programming Explanation:

From this code, we have used Serial Software library to interface ESP8266 and GPS module with Arduino. Then we have defined different pins for both and initialize UART with 9600 baud rate . Also included Liquid Crystal Library for interface LCD with Arduino.

```
#include<SoftwareSerial.h>

SoftwareSerial Serial1(2,3);
SoftwareSerial gps(10,11);
#include<LiquidCrystal.h>
LiquidCrystal lcd(14,15,16,17,18,19);
```

After it, we need to define or declare variable and string for different purpose.

```
String webpage=””;
Int i=0,k=0;
Int  gps_status=0;
String name=”<p>1. Name: Your Name </p>”;   //22
String dob=”<p>2. DOB: 12 feb 1993</p>”;     //21
String number=”<p>4. Vehicle No.: RJ05 XY 4201</p>”;//29
String cordinat=”<p>Coordinates:</p>”;          //17
String latitude=””;
String logitude=””;
String gpsString=””;
Char *test=”$GPGGA”;
```

Then we have made some functions for different purposes like:

Function for getting GPS data with coordinates:

```
 Void gpsEvent()
{
  gpsString=””;
   while(1){
   while (gps.available()>0) {
  char inChar = (char)gps.read();
   gpsString+= inChar;
    if (I < 7) {
            if(gpsString[i-1] != test[i-1])
     {
       I=0;
    ….. ….
    ….. …..
```

Function for extracting data from GPS string and convert that data to decimal degree format from the decimal minute format, as explained earliar.

```
Void coordinate2dec()

{

 String lat_degree=””;

 For(i=18;i<20;i++)

  Lat_degree+=gpsString[i];

  String lat_minut=””;

 For(i=20;i<28;i++)

Lat_minut+=gpsString[i];

  ….. ….

  ….. …..
```

Function for sending commands to ESP8266 for configuring and connecting it with WIFI.

```
Void connect_wifi(String cmd, int t)

{

  Int temp=0,i=0;

  While(1)

  {

    Serial.println(cmd);

    Serial1.println(cmd);

    While(Serial1.available()>0)

    ….. ….

    ….. …..
```

Void show_coordinate() function for showing coordinate on the LCD and Serial Monitor and void get_ip() function for getting IP address.

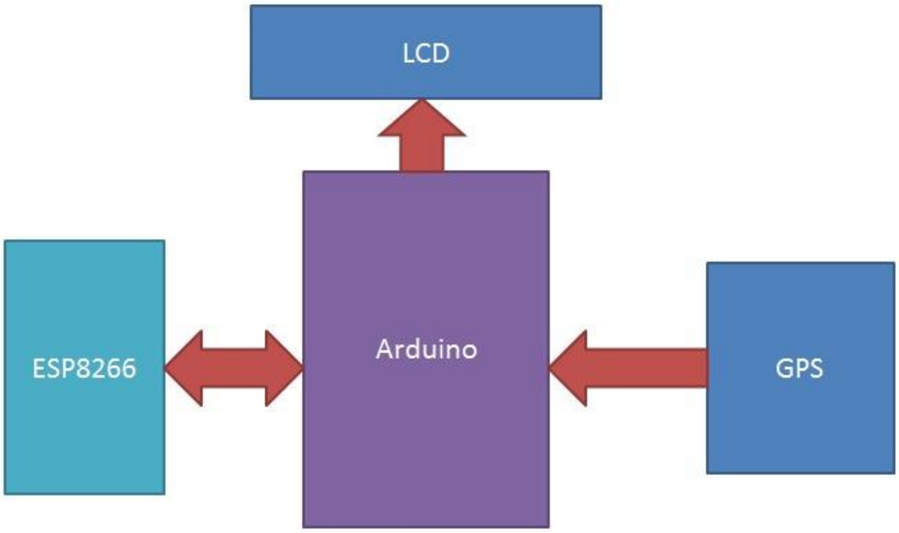
Void Send() function for creating a String of information that is to be sent to webpage using ESP8266 and void sendwebdata() Function for sending information string to webpage using UART

In void loop function Arduino continuously wait for request form webpage (Refreshing web page).

```
Void loop()
```

```
{
  K=0;
  Serial.println(“Please Refresh Ur Page”);
  Lcd.setCursor(0,0);
  Lcd.print(“Please Refresh  “);
  Lcd.setCursor(0,1);
  Lcd.print(“Your Web Page..  “);
  While(k<1000)
  ….. ….
```

Block Diagram:


LCD
ESP8266
Arduino
GPS