**Vehicle Tracker and auto start of Vehical**

# **MICROPROCESSOR INTERFACE**



# **REPORT**

## Group Members

* **Usama Ahmed 01-132172-029**
* **Immad ul Haq 01-132172-051**
* **Abdul Haseeb Ahmed 01-132172-001**

# Table of Contents

|  |  |  |
| --- | --- | --- |
| S.NO | TOPICS | Page No |
| 1 | Introduction to IOT | 3 |
| 2 | Benefits of IOT | 4 |
| 3 | What is vehicle tracking system | 8 |
| 4 | Introduction to the Equipment Used | 9 |
| 5 | Microcontroller USED | 10 |
| 6 | GPS module used | 12 |
| 7 | CODE | 15 |
| 8 | Vehicle AutoStart | 18 |
| 9 | Conclusions | 19 |

# INTRODUCTION TO IOT

## Definition:

**IoT** is short for Internet of Things. The Internet of Things refers to the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems.

## Explanation:

How are you reading this post right now? It might be on desktop, on mobile, maybe a tablet, but whatever device you’re using, it’s most definitely connected to the internet.

An internet connection is a wonderful thing, it give us all sorts of benefits that just weren’t possible before. If you’re old enough, think of your cellphone *before* it was a smartphone. You could call and you could text sure, but now you can read any book, watch any movie, or listen to any song all in the palm of your hand. And that’s just to name a few of the incredible things your smartphone can do.

Image Credit: Samsung

The point is that connecting things to the internet yields many amazing benefits. We’ve all seen these benefits with our smartphones, laptops, and tablets, but this is true for everything else too. And yes, I do mean *everything*.

The Internet of Things is actually a pretty simple concept, **it means taking all the things in the world and connecting them to the internet**.

I think that confusion arises not because the concept is so narrow and tightly defined, but rather because it’s so broad and loosely defined. It can be hard to nail down the concept in your head when there are so many examples and possibilities in IoT.

To help clarify, I think it’s important to understand the benefits of connecting things to the internet. Why would we even want to connect everything to the internet?

# Benefits of IOT

Look around, we are hearing a lot of buzz about **IoT** (Internet of Things). The world is getting reliant on the internet which is a massive global network that allows people to communicate with each other. We can send emails and messages, use websites and post on social media to communicate with other people. To do all these activities, we use devices like PC, Laptop or mobile devices [smartphones and tablet]. **Internet of Things** is a big science which deals with the collected data, interconnectedness and devices.

There are three things which are required in the main process of IoT –People Devices (with sensors) Servers.

The things which are using the internet to make connections and respond to other devices, people or systems are called the **Internet of Things** (IoT). IoT is playing a significant role in our lives and serving as an ecosystem which interconnects physical objects and systems in order to perform task smartly.

It could be a car which can be monitored to check fuel or its engine condition or an AC which can be operated remotely [like when you are not at home and you have forgotten to off it, you can make it off or operate it without specific AC remote] or any physical object with built-in sensors. These objects have been assigned with an IP address and after building a robust connection, data transfer takes place with manual assistance or computing intervention. And then actions are generated.

Consumers are demanding devices which support IoT and organizations are trying to capitalize on it. In this post, we would like to throw light on IoT to make your vision clearer towards it.

IoT is considered as the biggest frontier which can improve our lives in many aspects. Those devices which have never been networked can get connected and respond just like the smart devices do, for e.g. your car, refrigerator, and home speakers. IoT is set to transform our world completely.

Let’s have a look at the following benefits of IoT as per the business’s perspective –

**1. Efficient resource utilization  
2. Reduced human efforts  
3. Lowers the cost and bring productivity  
4. Real-time marketing  
5. Decision analytics  
6. Better customer experiences  
7. High-quality data**

**How is IoT contributing to businesses?**

For any business, the customer base is an important factor to become successful. Your customers can make or break the value of your brand. If you are doing well and have good growth in business, it means your customers are already satisfied with your services. Why not you break your comfort zone and try to make them happy instead of just satisfy.

There is stiff competition in the market, and brands are regularly making strategies based on the latest trends of the Internet of Things to stay up than their competitors. They are applying almost every trend willingly in their strategies in order to retain their customers and increase the value of their brand as well.

**IoT** is serving as a great and modern platform or tool in today’s time to streamline customer’s service and engagement in an excellent way. Building real-time interactions with your customers is one of the best things in favour of brands and customers both.

Let’s see how it is doing well.

**Maktotter customer experience**

With the help of IoT, customers can feel more connected with the brand. In-built sensors will collect the data and the recorded data help brands pay attention to their customers’ needs. It will make the customer experience better with that brand.

**Interactive engagement**

Effective data analytics enables you to interact in real time with your customers by tracking location, timing and searching inputs. Dynamic interactions can be created and displayed in different screen sizes.

**Advanced functionalities**

Allowing advanced features in your will deliver wow experience in your customer services just like mobile payments option with the embedded sensor.

**The relation between IoT and cloud computing. How their synergism is benefiting us?**

**Cloud Computing** and **IoT (Internet of Things)** are different technologies. Both of them are providing many facilities in various aspects.

IoT is all about connectivity and real-time interactions whereas Cloud computing is about saving data securely on the cloud.

Machines or physical devices can get connected with other devices, systems and people to communicate easily in a network are IoTs.  
When devices are interconnected, data is shared between them and this data is required to make decisions or actions. So you can imagine how important the data is. Talking about data security and maintenance, the first and last name that strikes in our mind is Cloud. Yes, a cloud!

What will be the synergy if Cloud and IoT merge? Every field can be benefitted with the alliance of both technologies in many ways. Let’s peek into it –

**Mobile App development**

Mobile apps are the basic need of today’s business. The two most common platforms are – Android and iOS. In order to reduce the workload for developers, cross-platform app development techniques are being used frequently from past few years.

Developers can build apps using HTML5 and hybrid frameworks. With the help of mobile cloud and IoT based tools, they can use shared resources and important data to enhance teamwork and make their practices more effective.Additionally, developers can view completed modules of other developers in the team to suggest changes on development tasks if required, which brings a lot of transparency and efficiency in the system.

Moreover, they can develop mobile apps using [MEAP](https://www.xamarin.com/platform?gclid=CjwKCAiA4vbSBRBNEiwAMorER5vOEFobaVoH6SrxqyUHqlWy6pKLuN6IzyIDa6AbII2r-TgJLy03whoCykAQAvD_BwE) (Mobile Enterprise Application Platform) which is an IoT tool. It allows users, to have a wide range of cross-platform development services along with cloud, better user-interface and strong authentication system.

Smartphones have various features like GPS tracking, mobile gyroscope, adaptive brightness (could be adjusted itself by sensing light intensity), voice detection, face detection etc, all are based on IoT.

Applications that are built on sensing abilities will definitely require inbuilt sensors in Smartphone to function well. Smartphone manufacturers are focusing on adding these types of features in their devices to attract customers. So, everything is relating directly or indirectly to IoT.

Let’s understand it by an example –

A cab facility has announced an app. Let’s consider ‘**Uber**’ mobile app, which is[utilizing IoT and cloud (IoT Car)](https://www.smart-industry.net/iot-car-the-internet-of-things-is-already-here-and-its-called-uber/) both in its all operations. Customers can book rides and available drivers will pick them from their location.All details about the ride are shared with the driver, customer and to the main system’s cloud database to improve services, send discount coupons and many other alerts.

**Smart City**:

Many countries are trying to make smart cities to open a new opportunity for the citizens. The data is being stored in large data centres in order to make quality communication and to make more efficient use of Information.

Japan and Europe are collaborated to work on a research project, trying to create smarter cities by using tools that merge **cloud computing** with the **Internet of Things** and named [Clout project](https://www.eetimes.com/author.asp?section_id=36&doc_id=1324213) – which stands for Cloud of things. This project will bridge the cloud and IoT to establish a fresh platform for all data sources.

Now, these examples clearly define the synergy of both IoT and Cloud technologies. Developers are dedicatedly focusing on learning IoT and making their apps compatible with IoT devices to benefit the users.

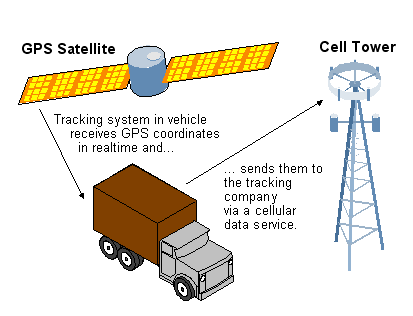
**Let’s wrap up –**Technologies are playing a crucial part in reinventing our experience on every activity and **Internet of Things** shares a great part in that. People are curious to know more about the Internet of Things. The biggest irony is that some of you do not want to dive into the depth but want to enjoy the comfort it offers. No matters, if you are a technology lover or a non-techie, it is benefitting all of us in many ways.

Google has launched many IoT devices till dates like Google Duo, Smart home devices, Voice assistant, Self-Driving Car and many more. Let’s hope for more in the coming years!!

# What is vehicle tracking system?

**Definition**

vehicle tracking use a combination of technologies to keep real-time tabs on the position of a vehicle or to construct a history of where a vehicle has been. These systems are used in a variety of industries, and they also a key part of most stolen vehicle recovery strategies. Most vehicle tracking systems use GPS technology, and some also make use of cellular or radio transmitters  
  
**Relay in Real Time**  
Vehicle tracking relies both on the Global Positioning satellites (GPS) and a cellular system. A tracking module in the vehicle continuously picks up the GPS coordinates that indicate the real-time location of the vehicle. Using a cellular data service, the coordinates are immediately transmitted to the tracking company's computers. Customers log in to the tracking company website to see their vehicles on road maps, similar to in-dash and handheld GPS-based navigation systems (see [navigation system](https://www.pcmag.com/encyclopedia/term/47674/navigation-system) and [GPS](https://www.pcmag.com/encyclopedia/term/43884/gps)).  
  
**No Relay - Report Later**  
There are non-real-time tracking systems in which the in-vehicle module records the coordinates, but does not relay them for real-time monitoring. The data are downloaded from the module when the vehicle is returned at the end of the trip. See [mobile positioning](https://www.pcmag.com/encyclopedia/term/47145/mobile-positioning).



**GPS Tracking in Real Time**

Real-time tracking uses both GPS and cellular systems to relay vehicle coordinates to the tracking service.

# Introduction to the Equipment used by us

* + - * **Node MCU (ESP8266)**



* + - * **[NEO-6M GPS Module](https://www.electroschematics.com/neo-6m-gps-module/)**

[](https://www.electroschematics.com/wp-content/uploads/2019/04/1-Ublox-NEO-6M-Module.jpg)

# Microcontroller Used

**Node MCU microcontroller board with ESP8266 and Lua is the microcontroller used by us.**

## **Details**

**NodeMCU microcontroller board with ESP8266 CP2102 module and Lua**

**Note:**NodeMCU is the name of both a firmware and a board  
**NodeMCU** is an open source IoT platform, whose firmware runs on Espressif's SoC Wi-Fi [ESP8266](https://www.elektor.com/search?q=8266), based on the ESP8266 [nonOS SDK](https://github.com/espressif/ESP8266_NONOS_SDK). Its hardware is based on the ESP-12 module. The scripting language is [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)) which allows to use many open source projects like [lua-cjson](https://github.com/mpx/lua-cjson) and [spiffs](https://github.com/pellepl/spiffs).

**Features:**

- Wi-Fi Module – ESP-12E module similar to ESP-12 module but with 6 extra GPIOs.  
- USB – micro USB port for power, programming and debugging  
- Headers – 2x 2.54 mm 15-pin header with access to GPIOs, SPI, UART, ADC, and power pins  
- Reset & Flash buttons  
- Power: 5V via micro USB port  
- Dimensions: 49 x 24.5 x 13 mm

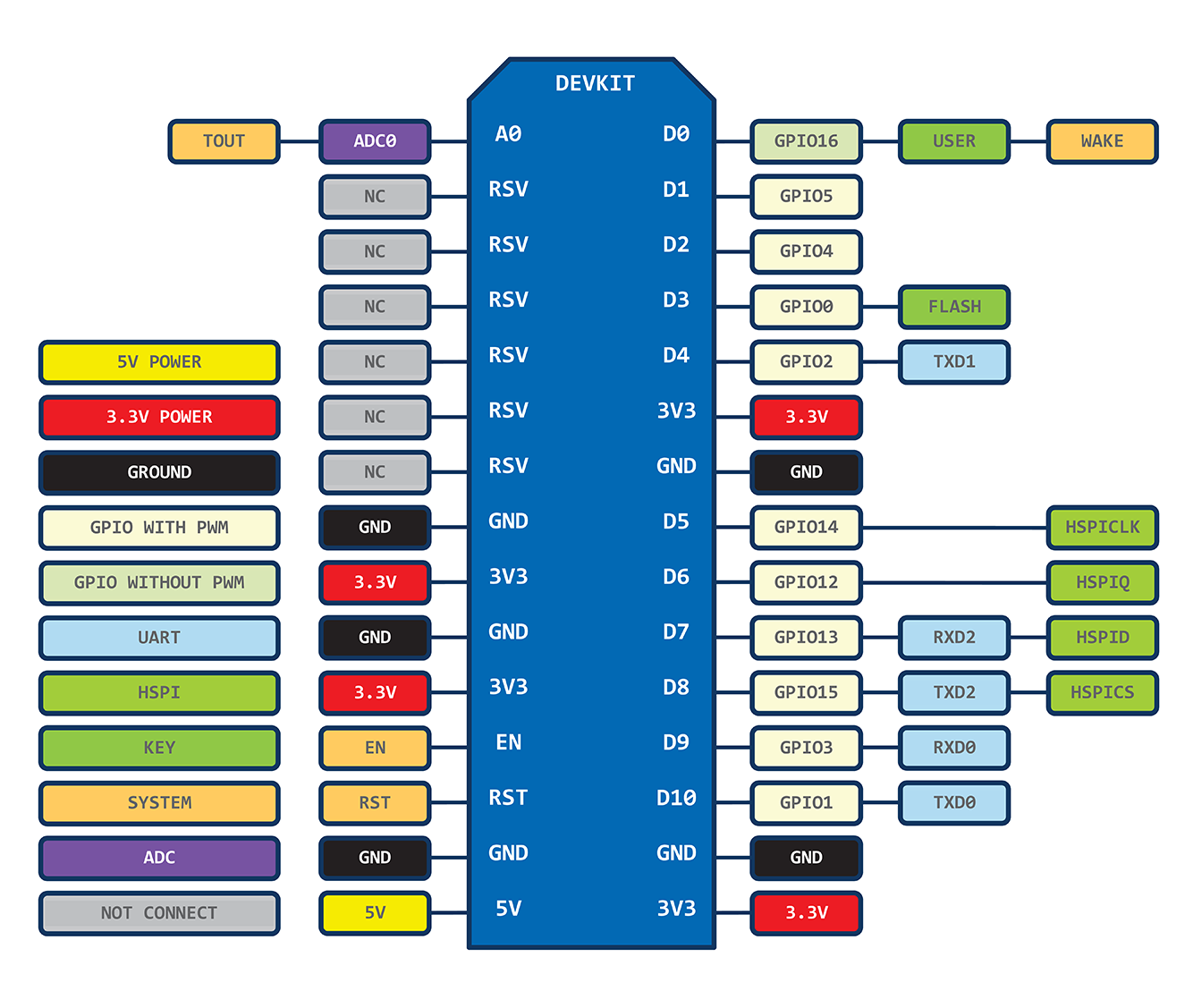
## **Specifications**

|  |  |
| --- | --- |
| **SKU** | 17952 |
| **Manufacturer** | Espressif |

##### **Figure 1. The NodeMCU DEVKIT board**



##### **Figure 2. The Node MCU pin schema**

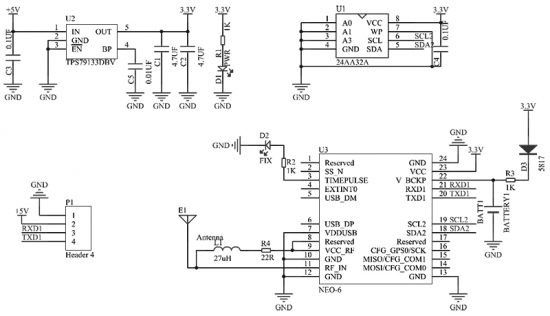


# GPS Module Used

**u-blox NEO-6M GPS module**

[](https://www.electroschematics.com/wp-content/uploads/2019/04/1-Ublox-NEO-6M-Module.jpg)

The u-blox NEO-6M GPS engine on these modules is quite a good one, and it also has high sensitivity for indoor applications. Furthermore, there’s one MS621FE-compatible rechargeable battery for backup and EEPROM for storing configuration settings. The module works well with a DC input in the 3.3- to 5-V range (thanks to its built-in voltage regulator). The original circuit diagram of the module, borrowed from the web, is shown below:

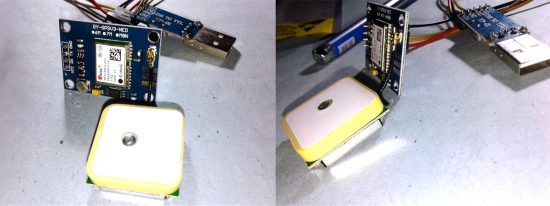
[](https://www.electroschematics.com/wp-content/uploads/2019/04/2-Ublox-NEO-6M-GPS-Module-Schematic.jpg)

As indicated, the GPS modules are based on the u-blox NEO-6M GPS engine. The type number of the NEO-6M is NEO-6M-0-001, and its ROM/FLASH version is ROM 7.0.3 (PCN reference UBX-TN-11047-1). The NEO-6M module includes one configurable UART interface for serial communication, but the default UART (TTL) baud rate here is 9,600. Because the GPS signal is right-hand circular-polarized (RHCP), the style of the GPS antenna will be different from the common whip antennas used for linear polarized signals. The most popular antenna type is the patch antenna. Patch antennas are flat, generally have a ceramic and metal body, and are mounted on a metal base plate. They are often cast in a housing. For more information about u-blox reference designs, see their [**website**](http://www.u-blox.com/). Remember, the position of the antenna mounting is very crucial for optimal performance of the GPS receiver. When using the patch antenna, it should be oriented parallel to the geographic horizon. The antenna must have full view of the sky, ensuring a direct line of sight with as many visible satellites as possible.

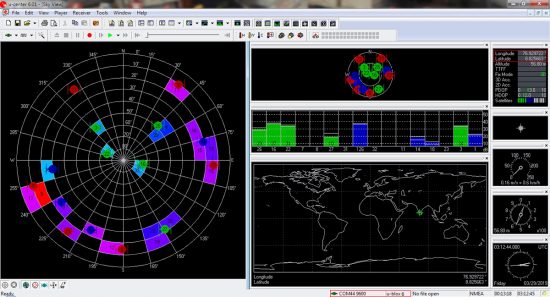
**Initial test setup**

For a quick test using your Windows computer, you just need to establish a serial communication with the GPS module using one USB-UART adapter like the PL2303 USB-to-Serial Converter module. The hardware setup is pretty simple:

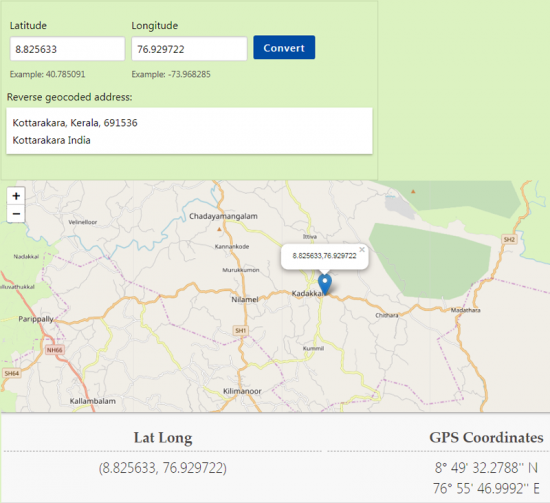
|  |  |
| --- | --- |
| **NEO-6M GPS Module** | **USB-to-Serial Converter** |
| TX | RX |
| RX | TX |
| GND | GND |
| VCC | 5 V |

[](https://www.electroschematics.com/wp-content/uploads/2019/04/3-Workbench-Snaps.jpg)

Next, download and install the Windows PC debug/evaluation tool “[**u-center**](http://wiki.sunfounder.cc/images/b/b5/U-center.zip)”. After successful installation, run the software (and the hardware) setup to transfer positioning data collected by the NEO-6M GPS module to the u-center software so that it can clearly display GPS data/information on the screen (see the below screen capture).

[](https://www.electroschematics.com/wp-content/uploads/2019/04/4-uCenter-Live.jpg)

Note that when the GPS module works, the green indicator on the GPS module will blink (the red one is for power-on indication), and the figures regarding the time, latitude, longitude, etc., will be displayed in the u-center software window. Finally, you should compare the figures shown by the software with the data collected by another trusty GPS device to ensure that your NEO-6M GPS module is perfect (and precise) in all respects. An alternative trick that I’d figured out to complete the quick evaluation process is the “Reverse Geocoded Address” method (see next image), as it’ll help to eliminate the requirement of a commercial GPS device. It’s worth a try!

[](https://www.electroschematics.com/wp-content/uploads/2019/04/5-Reverse-Geocoded-Address.png)

# CODE

**#include <TinyGPS++.h>**

**#include <SoftwareSerial.h>**

**#define BLYNK\_PRINT Serial**

**#include <ESP8266WiFi.h>**

**#include <BlynkSimpleEsp8266.h>**

**static const int RXPin = 4, TXPin = 5; // GPIO 4=D2(conneect Tx of GPS) and GPIO 5=D1(Connect Rx of GPS**

**static const uint32\_t GPSBaud = 9600; //if Baud rate 9600 didn't work in your case then use 4800**

**TinyGPSPlus gps; // The TinyGPS++ object**

**WidgetMap myMap(V0); // V0 for virtual pin of Map Widget**

**SoftwareSerial ss(RXPin, TXPin); // The serial connection to the GPS device**

**BlynkTimer timer;**

**float spd; //Variable to store the speed**

**float sats; //Variable to store no. of satellites response**

**String bearing; //Variable to store orientation or direction of GPS**

**char auth[] = "3xwFMAuNuj2lirMVCEWmXotTWaCkcjmz"; //Your Project authentication key**

**char ssid[] = "Immad's iPhone"; // Name of your network (HotSpot or Router name)**

**char pass[] = "O3365013493"; // Corresponding Password**

**//unsigned int move\_index; // moving index, to be used later**

**unsigned int move\_index = 1; // fixed location for now**

**void setup()**

**{**

**Serial.begin(9600);**

**Serial.println("Connecting to Device");**

**ss.begin(GPSBaud);**

**Blynk.begin(auth, ssid, pass);**

**timer.setInterval(5000L, checkGPS); // every 5s check if GPS is connected, only really needs to be done once**

**}**

**void checkGPS(){**

**if (gps.charsProcessed() < 10)**

**{**

**Serial.println(F("No GPS detected: check wiring."));**

**Blynk.virtualWrite(V4, "GPS ERROR"); // Value Display widget on V4 if GPS not detected**

**}**

**}**

**void loop()**

**{**

**while (ss.available() > 0)**

**{**

**Serial.println("DATA AVAILABLE!!");**

**// sketch displays information every time a new sentence is correctly encoded.**

**if (gps.encode(ss.read())){**

**Serial.println("Going to Display: ");**

**displayInfo();**

**}**

**}**

**Blynk.run();**

**timer.run();**

**}**

**void displayInfo()**

**{**

**if (gps.location.isValid() )**

**{//Serial.write(myserial.read());**

**Serial.println("Printing Coordinates: ");**

**float latitude = (gps.location.lat()); //Storing the Lat. and Lon.**

**float longitude = (gps.location.lng());**

**Serial.print("LAT: ");**

**Serial.println(latitude, 6); // float to x decimal places**

**Serial.print("LONG: ");**

**Serial.println(longitude, 6);**

**Blynk.virtualWrite(V1, String(latitude, 6));**

**Blynk.virtualWrite(V2, String(longitude, 6));**

**myMap.location(move\_index, latitude, longitude, "GPS\_Location");**

**spd = gps.speed.kmph(); //get speed**

**Blynk.virtualWrite(V3, spd);**

**sats = gps.satellites.value(); //get number of satellites**

**Blynk.virtualWrite(V4, sats);**

**bearing = TinyGPSPlus::cardinal(gps.course.value()); // get the direction**

**Blynk.virtualWrite(V5, bearing);**

**}**

**else {**

**Serial.println("We are in the Displaying section but Data is not valid yet: ");**

**}**

**/\***

**Serial.println("Printing Coordinates: ");**

**float latitude = (gps.location.lat()); //Storing the Lat. and Lon.**

**float longitude = (gps.location.lng());**

**Serial.print("LAT: ");**

**Serial.println(latitude, 6); // float to x decimal places**

**Serial.print("LONG: ");**

**Serial.println(longitude, 6);**

**Blynk.virtualWrite(V1, String(latitude, 6));**

**Blynk.virtualWrite(V2, String(longitude, 6));**

**myMap.location(move\_index, latitude, longitude, "GPS\_Location");**

**spd = gps.speed.kmph(); //get speed**

**Blynk.virtualWrite(V3, spd);**

**sats = gps.satellites.value(); //get number of satellites**

**Blynk.virtualWrite(V4, sats);**

**bearing = TinyGPSPlus::cardinal(gps.course.value()); // get the direction**

**Blynk.virtualWrite(V5, bearing);**

**Serial.print("check me Now");**

**\*/**

**}**

# Vehicle AutoStart

To automate a car engine (petrol) driven air compressor, the engine has an ecu which controlls the injectors and ignition (from the original donor car).  
I was thinking of just closing a relay for the ignition (wired into where the ignition switch was on the loom) then close another relay to bring in the starter, then measure the battery voltage with an a to d on a pic, when the engine fires the battery voltage rises by a quite a bit due to the releived load on the starter, and when this happens release the starter relay as the engine has started.  
The battery voltage drops to around 6 or 7 while while the engine cranks, then comes up to around 10 when the engine fires (before the alty kicks in).  
And I'd have to include a timer so the starter doesnt remain on too long, and on top of that maybe a number of tries counter resulting in an alarm if the engine wont go.

You can check if the engine has started by checking the frequency of the ignition coil or injector pulses. Many ECUs have a seperate clean 0/12v pulse to drive the tachometer.  
  
Put a secondary cutoff on a seperate series relay for the starter motor - something independent of the main start circuit which will act as a failsafe if the starter is operated for more than 10 seconds (for example).  
  
Also hook in a monitor for the oil pressure switch - if the engine oil runs low then you can cut out the engine.  
  
All pretty simple stuff and a lot of it is already done by remote start car alarm systems.

Commercial vehicles with a fancy start button cut out the starter as soon as the engine fires, I'd assumed it was done on starter current, but I spose the starter current is directly related to engine speed, and its engine speed we go off when we turn the key on our cars every day.  
I'll try that, maybe look for a 10% change in speed over a short time, the ecu doesnt have a pulse o/p I'll use an indirect pickup from the coil ht lead.  
And good advice on oil pressure, I can just use the engines existing switch, it does actually have an oil level sensor, I can read this before the engine starts.  
I dont really want to go to the expense of a battery isolator relay, those things are huge, however I can put 2 relays in line with the starter solenoid and just use one to operate the starter, and use the other if the starter relay doesnt disengage (welded contacts).  
I've had a starter relay seize on before, not very nice.

# Conclusion

This was the detailed report of the project that we were being assigned to implement. We in detailed mention all the work that we had performed. We submitted and gave viva of the tasks assigned to us to the instructor. It was a project done by the whole group.