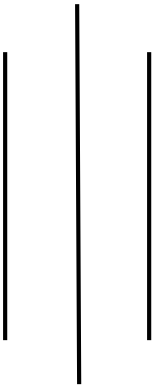
A Thesis for the Fulfilment of Bachelor's Degree

VEHICLE MONITORING AND ACCIDENT ALERT SYSTEM USING IOT



Submitted By:

Ramesh Subedi

Student ID: 17-09-002

Submitted To:

Wakkanai Hokusei Gakuen University

Faculty of Integrated Media

Supervisor:

Assoc. Prof. Yojiro Harie

Co-Supervisor:

Prof. Hiroyasu Asami

Prof. Tomoharu Ando

ABSTRACT

The Internet of Things (IoT) plays a prominent role in today's world. Everything is being smart and intelligent. IoT represents the idea of connecting physical objects to other objects over the Internet. It has created an ecosystem that links many systems to give smart performances in every task. This project is about the design and implementation of Vehicle monitoring and Accident alert system using modern IoT devices. It comprises of integration between a GPS receiver, gyro sensor, LCD display, a microcontroller and Blynk application. This project can be divided into two main parts i.e. the hardware development part which includes the wiring connection of GPS receiver, LCD display, Gyro sensor with a microcontroller and the software development part which includes developing source code for microcontroller.

The advancement in the field of transportation has not only increased the use of vehicles, but also increased the risk of road accident. In Nepal, due to the bad road network in the hilly regions, there is always a risk of tilting and overturning of vehicle, which causes accident. We often hear the news of vehicle being out of contact for many days. Every year, many people lose their life for not being rescued in time. Although we see various scientifically advanced tracking system introduced in technically developed countries, these advanced system are not in reach for the underdeveloped countries since they are expensive to purchase and difficult to implement. Therefore there is a need of a cheap and affordable system that helps to monitor the position of vehicle in real time and also alert when something uncertain happens.

The main objective of this research is to develop an inexpensive system that can monitor different vehicle behavior like position, speed, location, inclination, etc. in real time, suggest the driver for safe driving, and alert the concerned authorities in case of sudden changes in the vehicle behavior. If we are able to monitor the vehicle behavior, we can conduct the immediate rescue operations in case of accident, which will minimize the possible damages of accident.

ACKNOWLEDGEMENT

I would like to express my deep gratitude and sincere thanks to Prof. Dr. Bishnu Prasad Gautam for his encouragement, valuable suggestions and motivation after enrolling Wakkanai Hokusei Gakuen University, Faculty of Integrated Media. I sincerely appreciate his generosity by taking me into his fold for which I shall remain indebted to him.

I extent my appreciation to Assoc. Prof. Yojiro Harie, my project supervisor, for his professional guidance throughout the project. I take this opportunity to express my deep sense of gratitude to his invaluable guidance, encouragement, enormous motivation, which has sustained my efforts throughout the project. Similarly, I would like to thank Prof. Hiroyasu Asami, Prof. Andoh Tomoharu, and Prof. Saga Takahiro for their advice and assistance throughout the study. Also many thanks to all the members of IoT Seminar for their valuable opinions and suggestions regarding the project.

I can't forget to offer my sincere thanks to all the teachers and members of Wakkanai Hokusei Gakuen University, who helped and supported me throughout the academics and extra-curricular activities.

Finally, I wish to thank my parents and friends for their support and encouragement throughout my study.

Table of Contents

FIGURE INDEX

CHAPTER 1: INTRODUCTION

The Internet of Things (IoT) represents the idea of connecting physical objects to other objects over the Internet. IoT plays a prominent role in today's world. Everything is being smart and intelligent. IoT is making our lives better in so many ways, and it will continue to do so. Over the last decades, IoT is growing in importance, both for the industrial as well as everyday use. The IoT not only connect the electronic appliances with the internet, but also controls the appliances through the Internet. Moreover, with the help of GPS enabled devices, we can digitally monitor and control the activities all over the world. Nowadays, GPS enabled devices like smartphones, watches, telematics are widely used for the purpose of tracking, mapping, and navigation. However, these scientifically advanced devices are not found everywhere since they are very expensive and everyone cannot afford to buy those devices. Therefore there is a need of a system that is cheap and easy to implement, which is the main aim of this research.

There are a lot of system already introduced for tracking the vehicle and monitoring the activities. Most of the system uses GPS receiver to extract the location data and processes the data over the internet. Many system uses GSM module to transmit the location data in the form of message using a mobile network. But nowadays, with the widespread use of wireless internet protocols like Wi-Fi, people can easily get access over the internet and pass information quickly to desired place. On the other hand, we can see various accident alert system that processes sensor values and quickly inform the concerned authority when accident occurs. If we can make a system that can alert the user/authority from taking certain action before the accident, we might be able to prevent the accident, which is the aim of this research.

In this project, we will be working on a system that monitors the position of vehicle in real time using Blynk API, creates threshold for alert and accident, alerts the driver when the vehicle is performing unusual behavior, and also notifies the concerned authority when vehicle encounters an accident. Since we have started from the research level, this project covers only minor accidents like over sliding and over turning of vehicle that only causes minor damages and injuries. The experiment for this project is performed in a bicycle since we do not have other vehicles, but we can implement this system in other vehicles too using similar concept introduced in this system.

CHAPTER 2: RESEARCH BACKGROUND

In this research, we are mainly focused on making a simple Vehicle monitoring and alert system that is cheaper and easy to use. We can give the threshold value for the alert and accident case through our coding and remotely monitor the real time location of the vehicle using mobile application. Moreover, in case of accident, we can send the alert messages to the concerned authority through mobile application and also through a direct email notification.

2.1 IoT (Internet of Things)

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. An IoT ecosystem consists of a web-enabled smart devices that use embedded systems, such as processors, sensors and communication hardware, to collect, send and act on data they acquire from their environment [1]. The figure below shows the applications of IoT [2].

Figure 1 Applications of IoT

2.2 GPS

The Global Positioning System (GPS) is one of the global navigation satellite systems (GNSS) owned by the United States government and operated by United States Space Force [3]. This system provides users with positioning, navigation, and timing (PNT) services and mainly consists of three segments. The figure below shows three segments of GPS [4].

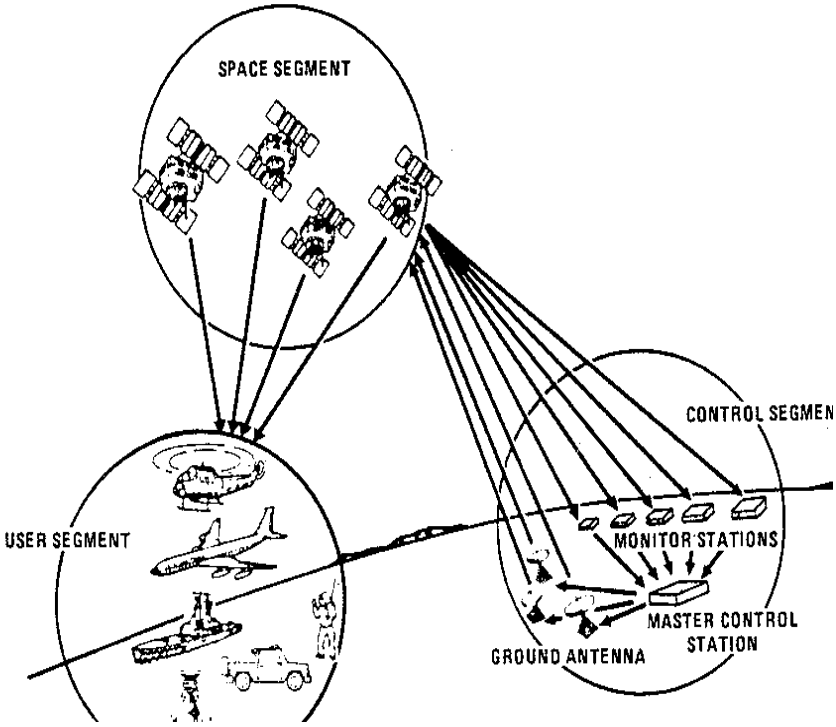


Figure 2 Three segment of GPS

1. Space segment:
   It consists of constellation of 24 satellites which transmit radio signals to users.
2. Control segment:
   It includes the control stations and monitor which maintain satellites in their orbit with command maneuvers and also adjust the satellite clocks. It tracks GPS satellites, uploads updated navigational data, and maintains health status of the satellite constellation [5].
3. User segment:
   It includes GPS receiver which receive signals from GPS satellites and uses the data to calculate the position and time.

In this project, we are going to work on the user segment. We will use a GPS receiver to extract the location data and use the data in our tracking system. GPS receiver uses 3D trilateration to determine their position on earth surface.

2.3 TinyGPS++

It is an arduino library for parsing NMEA data streams provided by GPS modules. NMEA, an acronym for the National Marine Electronics Association, is a standard data format supported by all GPS. By using TinyGPS++ library, we can easily extract the position, date, time, altitude, speed, and course from GPS devices. TinyGPS++ is the immediate inheritor of TinyGPS, a popular compact parser that is used in Arduino installations around the world. TinyGPS++ is not quite as 'tiny' as its older sibling, but its powerful and extremely easy-to-use new object model and useful new feature set make it an attractive alternative. Although TinyGPS++ shares much the same compact parsing engine with TinyGPS, its programmer interface is somewhat more intuitive. The main TinyGPS++ object contains several core sub-objects like location, date, time, speed, course, altitude, satellites, and hdop [6].

2.4 MPU6050_light

It is an Arduino library for light and fast communication with the MPU6050. We can retrieve accelerometer and gyroscope measurement from MPU6050. This data is processed using a complementary filter to provide and estimation of tilt angles on X and Y with respect to the horizontal frame. This library is compatible with all the architectures so we can use it in all Arduino boards [7].

2.5 Literature Review

Various researches have been conducted to make vehicle monitoring and accident alert system. The “IOT BASED SMART VEHICLE MONITORING SYSTEM” [8] proposes the use of IoT technology by using sensor acquainted Raspberry Pi for accident detection and machine learning algorithm to find the severity of accident. The system also uses GPS and GSM module for finding the location and communicate through the cellular network.

“Real Time Vehicle Tracking System using GSM and GPS Technology-An Anti-theft Tracking System” [9] proposed a design of a vehicle tracking system using GPS and GSM technology to continuously monitor the moving vehicle and report the status of vehicle on demand.

“Vehicle Accident Detection System using Internet of Things (VADS -IoT)” [10] aims to develop a system using a vibration sensor to determine the collision impact of accident and a gyro sensor to determine x-y displacement of the vehicle. The location captured by GPS after the accident is transmitted to the emergency response department via a GSM module.

After studying and analyzing various related research paper, I was motivated to do something related about the vehicle monitoring and accident alert system. Also we have realized the necessity of some uniqueness in the existing system that were developed before. We realized that our system will be more effective and unique if we are able to alert the driver of the vehicle for safe driving before the accident, which will prevent the accident, rather than getting prepared for conducting rescue operations after the accident.

CHAPTER 3: RESEARCH ISSUES AND PROPOSED SOLUTION

3.1 Research Issue

Over the last decades, the use of vehicle is rapidly increasing, which on the one hand is making the life of human easier while on the other hand the risk of road accident is also increasing. According to WHO, approximately 1.3 million people die each year as a result of road accident and more than half of all road accident deaths are among vulnerable road users like pedestrians, cyclists, and motorcyclists [16]. Every day we hear the news of road accident and many people lose their life for not being rescued in time. Therefore safe driving has been a challenge in transportation sector. Different alert systems are invented, which help to alert the owner and other concerned authority immediately after the accident which in turn will help to perform the rescue operation and save the life of victims, as in [8], [9], [10]. Although these systems are effective as we can perform the rescue operations immediately after the accident, I think it is better try to avoid the accident from happening by alerting the driver immediately when the vehicle shows unusual behavior and suggest the driver for safe driving. If we are able to develop such a system, we might be able to prevent the accident, rather than preparing for the rescue operations after the accident.

The other issue is that in context of our country Nepal, most of the areas in the hilly regions are not connected with proper road network yet. So in these areas only limited vehicles are used. The roads are sloppy and poorly maintained. The road gets further deteriorated during the rainy seasons due to continuous rainfall and sudden landslides. In this case, many vehicles have to stop in the middle. Sloppy and poorly maintained road causes inclining and overturning of the vehicle. There are many cases where the vehicles have fallen down into the river and disappeared for many days. And due to the lack of proper access to rural areas, the search of victims and investigation cannot be carried out immediately. That is why the challenge of tracking the vehicles seems important. If the tracking system is introduced in such vehicles, we can monitor the vehicle from anywhere and there will be easiness to carry out the rescue operations in case of uncertainties.

Of course, there are many tracking and alert system introduced in technologically/economically rich countries. However, these systems are very expensive to afford and difficult to implement in the country of poor economy. Therefore we need a system that is less expensive and easy to implement.

3.2 Proposed Solution

In order to solve the issues, we have realized that both the monitoring and alert system is necessary. After reviewing various research paper [8], [9], [10], we decided to combine some concepts of vehicle tracking and accident alert system and also introduce a new approach of safe driving that makes our system smart and unique. Therefore our approach is to develop an integrated system using modern IoT devices that monitors position of vehicle in real time, alert the driver for safe driving if the vehicle is showing unusual behavior and also alert the concerned authority to carry out the rescue operations immediately if accident occurs.

CHAPTER 4: METHODOLOGY

The methodology implemented during the research is illustrated using the figure below:

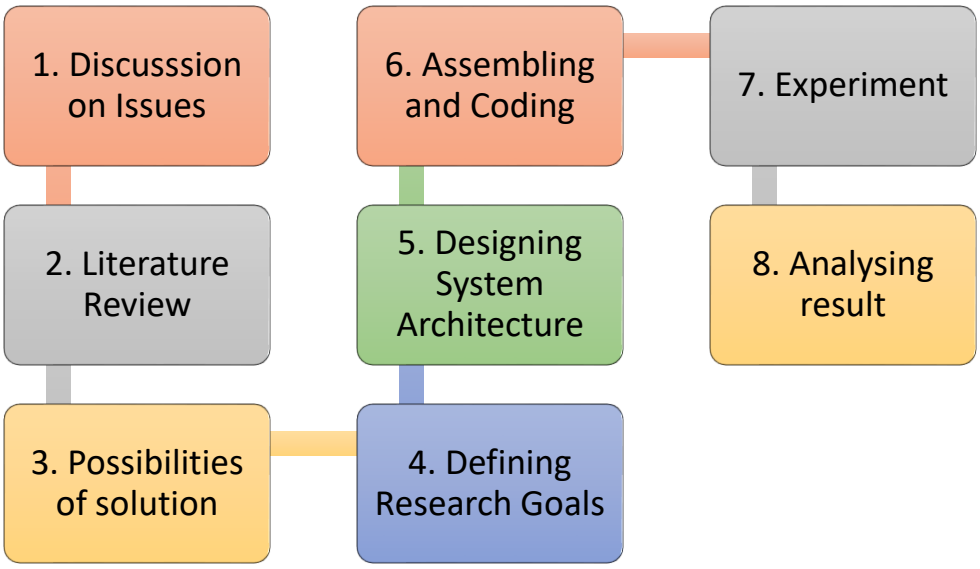


Figure 3 Methodology flow diagram of Vehicle Monitoring and Accident Alert System

The research issues were discussed with the professors and students of IoT Seminar. Then we go through some of the related research and analyzed them properly. After that we discussed on our major research goals and decided to design a Vehicle monitoring and Accident alert system by using IoT devices. Then we worked on assembling microcontroller and different sensors and coding using Arduino IDE. During the development stages, we performed the experiment many times using a Bicycle, debug the errors and analyze the result.

CHAPTER 5: SYSTEM OVERVIEW

This research mainly consists of 3 different sections:

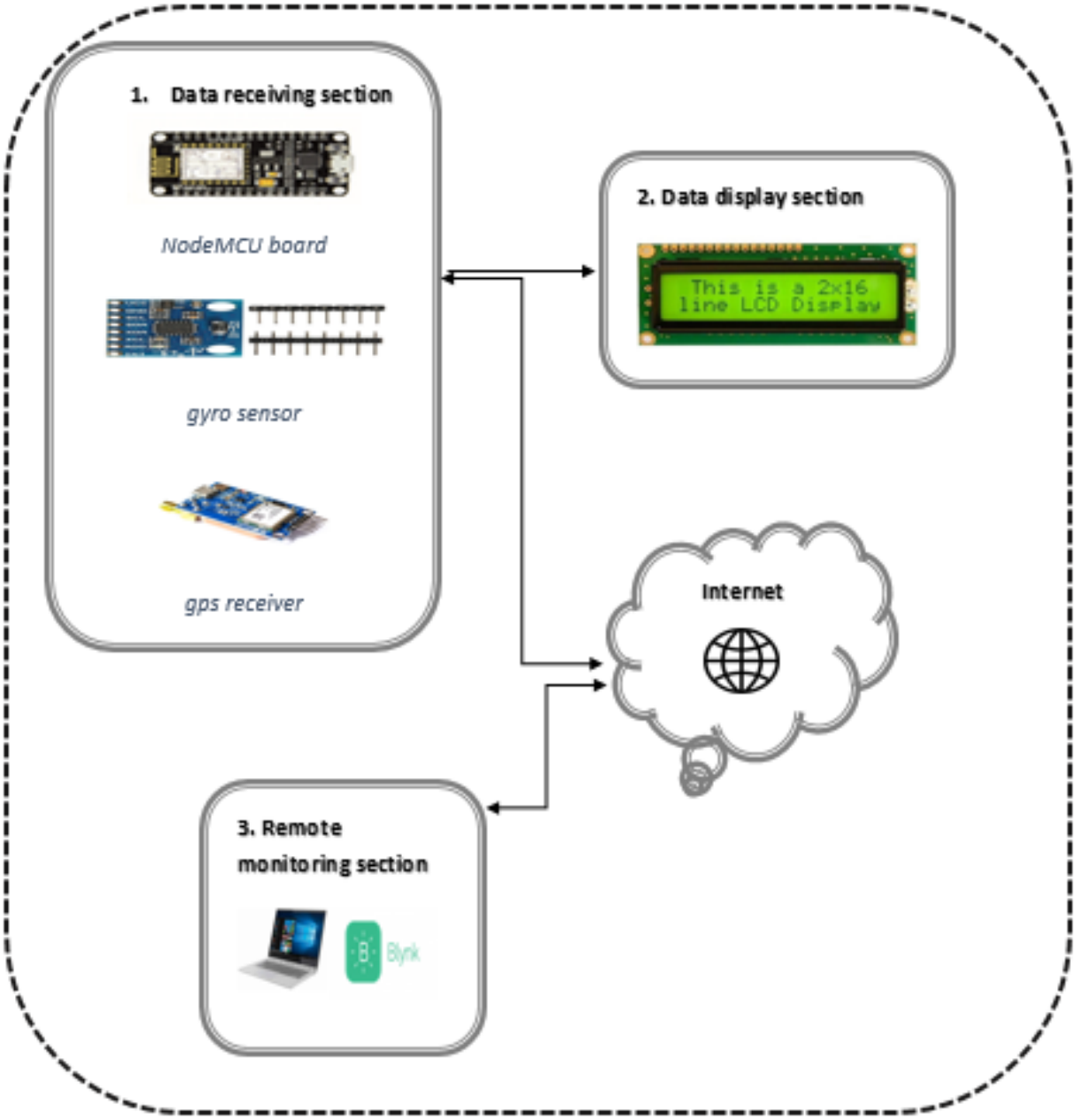


Figure 4 System Overview diagram of Vehicle Monitoring and Accident Alert System

The data receiving section consists of NodeMCU as a microcontroller, a GPS receiver to extract location data, Gyro sensor to extract the angle of inclination of vehicle. NodeMCU is connected to a Wi-Fi or mobile hotspot. The Data display

section consists of a LCD display that shows the current angle readings of gyro sensor and also displays the messages according to the program written in the microcontroller. The remote monitoring section consists of mobile phones or PC that have Blynk application installed. Any alert messages and notification can be obtained through Blynk application and also through the Email.

5.1 Hardware used

1. NodeMCU ESP8266:

NodeMCU is an open-source Lua based firmware and development board specially designed for IoT based Applications. It consists of firmware that runs on the ESP8266 Wi-Fi SoC from Espressif Systems having Tensilica Xtensa 32-bit LX106 RISC microprocessor and hardware that is based on ESP-12 module. The microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. It has 128 KB RAM and 4MB of Flash memory to store data and programs. Its high processing power with in-built Wi-Fi/Bluetooth and Deep Sleep Operating features make it ideal for IoT projects [11]. The NodeMCU Development Board can be easily programmed with Arduino IDE.

Specification and Features:

- Microcontroller: Tensilica 32-bit RISC CPU Xtensa LX106
- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- Digital I/O Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4MB
- SRAM: 64KB
- Clock Speed: 80 MHz

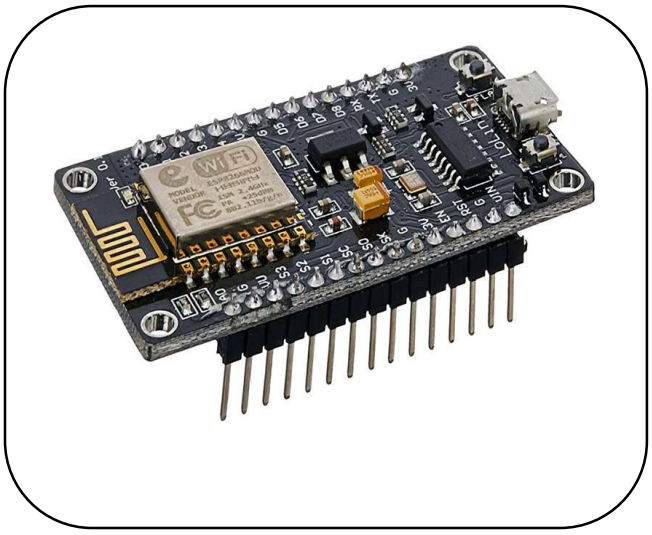


Figure 5 NodeMCU ESP8266

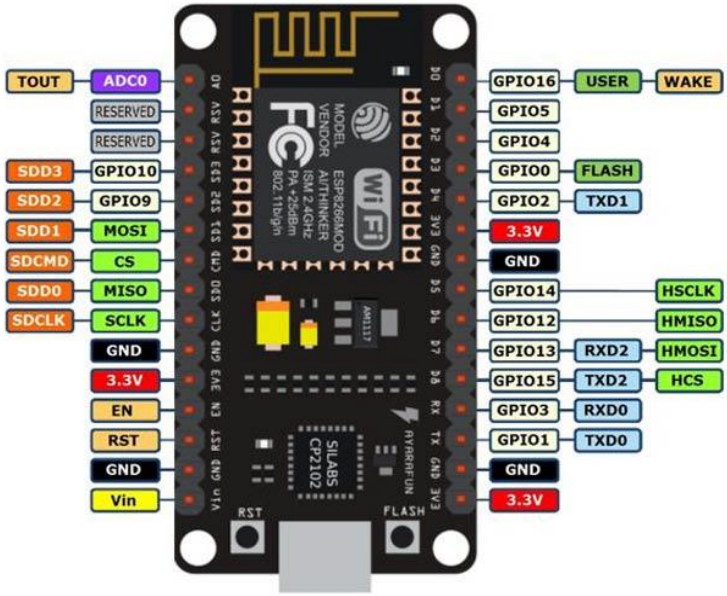


Figure 6 NodeMCU Pinout

2. GPS module:

   The GPS devices utilizes data from satellites to locate a specific point on the Earth in a process called trilateration. The GPS modules are easily compatible with Arduino and some other microcontrollers like NodeMCU. The GPS receiver is capable of solving the navigation equations to determine the user position, velocity and precise time by processing the signal broadcasted by GPS satellites. In this research we are going to use a NEO-M8 module as a receiver. The receiver can be extended with an external antenna for good data signal. The figure below is a NEO-M8N-0-10 ublox receiver.

Figure 7 NEO-M8N-0-10

3. MPU6050:

   MPU6050 is a complete 6-axis motion tracking device. It consists of 3-axis Gyroscope, 3-axis accelerometer and digital motion processor all in small package. It has I2C bus interface to communicate with the microcontrollers. The MPU6050 consist of 3-axis gyroscope with Micro Electro Mechanical System (MEMS) technology. It is used to detect the rotational velocity or rate of change of angular position over time, along the X, Y, and Z axes [12]. The figure below is a MPU6050 sensor.

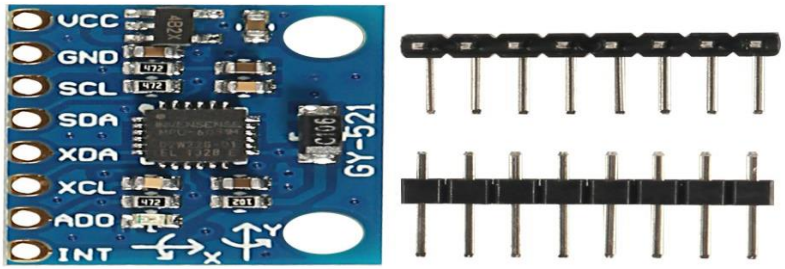


Figure 8 MPU6050

4. I2C LCD:

   Since the wiring between Arduino and a normal LCD is complicated, LCDI2C is created to simplify the wiring. The LCDI2C is composed of a normal LCD, I2C module and a potentiometer. The I2C module has an inbuilt PCF8574 I2C chip that converts I2C serial data to parallel data for LCD display. These modules are supplied with a default I2C address of either 0x27 or 0x3F. In the module we use in this project, it consists of 3 sets of pads labelled as A0, A1, and A2, hence the default address is 0x3F. The module has a contrast adjustment potentiometer on underside of display. We can adjust the screen to display the text accordingly [13].

   Features of I2C:

   - Operating voltage of 5V
   - Backlight and contrast is adjusted by potentiometer
   - Serial I2C control of LCD display using PCF8574
   - Come with 2 IIC interface, which can be connected by output line

- Compatible for 16x2LCD
- With this I2C interface module, we can realize the data display via only 2 wires

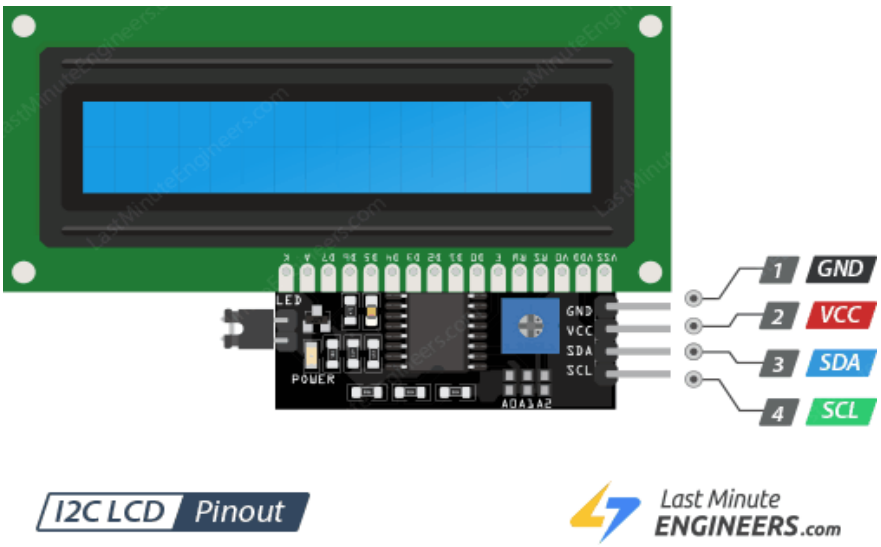


Figure 9 I2C LCD Display Pinout

5.2 Software used

1. Arduino IDE
   The Arduino Integrated Development Environment – or Arduino Software (IDE) contains a text editor for writing code, a message, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them [14].

2. Blynk
   Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, and it can store data, visualize it and do many other cool things [15]. The major components of Blynk are:
   - Blynk App: It allows you to create amazing interfaces for your projects using various widgets and pins.
   - Blynk server: It is responsible for all the communications between the smartphone and hardware. We can use Blynk cloud or run private Blynk server locally.

- Blynk libraries: It enables communications with the server and process all the incoming and outcoming commands for all popular hardware platforms. The figure below shows the working of Blynk:

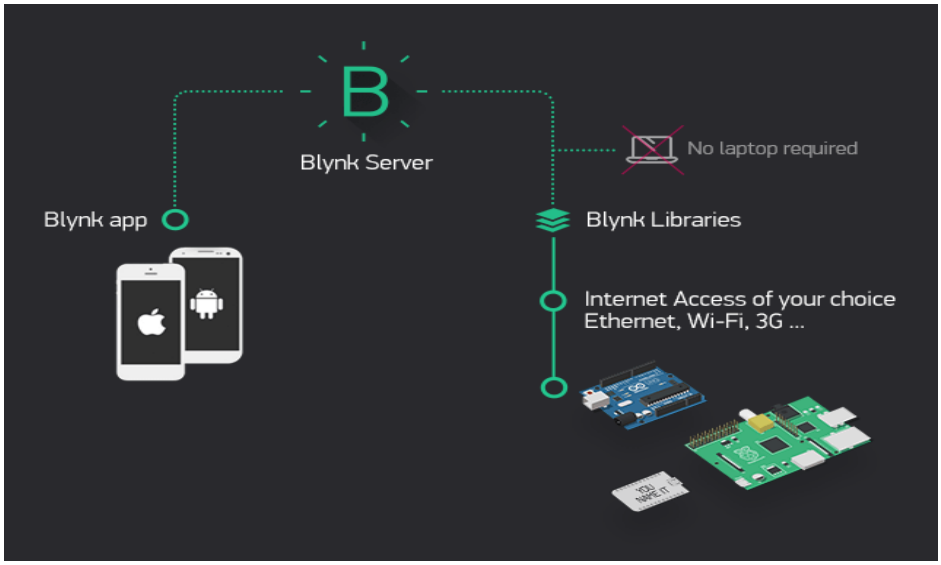


Figure 10 Working of Blynk

5.3 Coding with NodeMCU

We have used Arduino IDE for coding with the microcontroller NodeMCU. We have used following libraries in our code.

- MPU6050_light.h
  This is a library for light and fast communication with MPU6050. It is used to retrieve accelero and gyro data from MPU6050 and compute tilt angles along X and Y axes.
- TinyGPS++.h
  This is a library for parsing NMEA data streams provided by GPS modules.
- LiquidCrystal_I2C.h
  This is a library for I2C LCD displays. This library allows us to control I2C displays with functions similar to LiquidCrystal library.
- BlynkSimpleEsp8266.h
  This is the library to connect ESP8266 to the Blynk server.
- ESP8266WiFi.h
  This library provides ESP8266 specific Wi-Fi routines that we are calling to connect to the network.

CHAPTER 6: SYSTEM ARCHITECTURE

The overall architecture of Vehicle monitoring and Accident alert system is illustrated as:

- Vehicle Monitoring
- Alert for safe driving
- Alert after the accident
- Data display section
- Circuit diagram and threshold value

6.1 Vehicle Monitoring

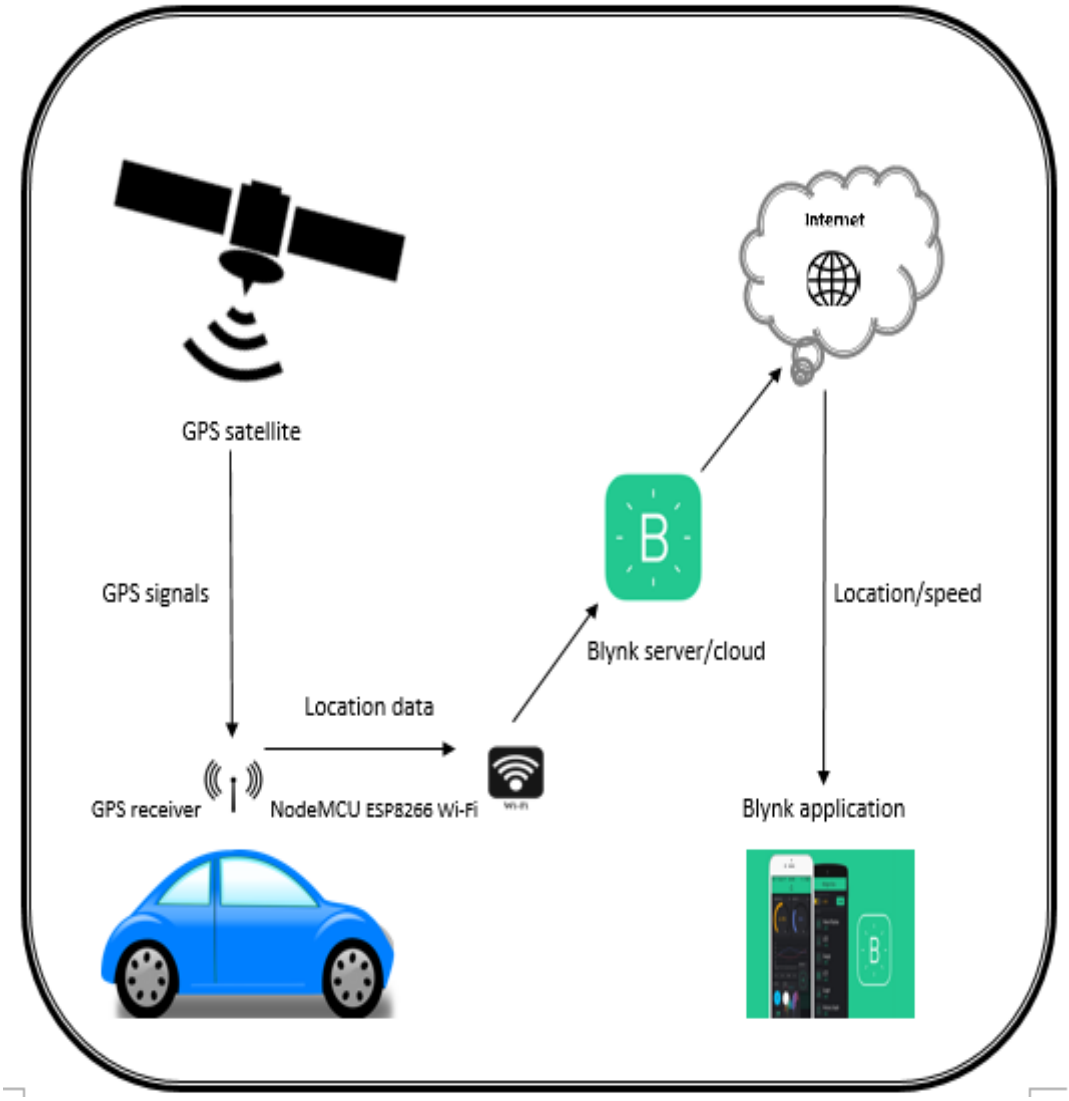


Figure 11 System architecture of Vehicle Monitoring section

This section includes the real time monitoring of the vehicle with the help of Blynk application. This section includes a NodeMCU microcontroller and GPS device which is installed somewhere in the vehicle. The GPS receiver is further extended with the help of an external antenna which is kept under the open sky for good signal. The GPS module continuously receives data signals in NMEA format. By using TinyGPS++ library, we will extract only the essential data i.e. latitude, longitude, satellites tracked, and speed. The extracted data is send to Blynk server/cloud with the help of NodeMCU that consists of ESP8266 Wi-Fi chip. The data can be accessed and visualized by using the Blynk mobile application through the embedded virtual pins and Google map widget. The position of vehicle can be monitored regularly in real time.

The connection between the microcontroller and GPS receiver is as follow:

NodeMCU microcontroller	**GPS receiver**
3v3	Vcc
GND	GND
D7	Tx
D8	Rx

6.2 Alert for safe driving

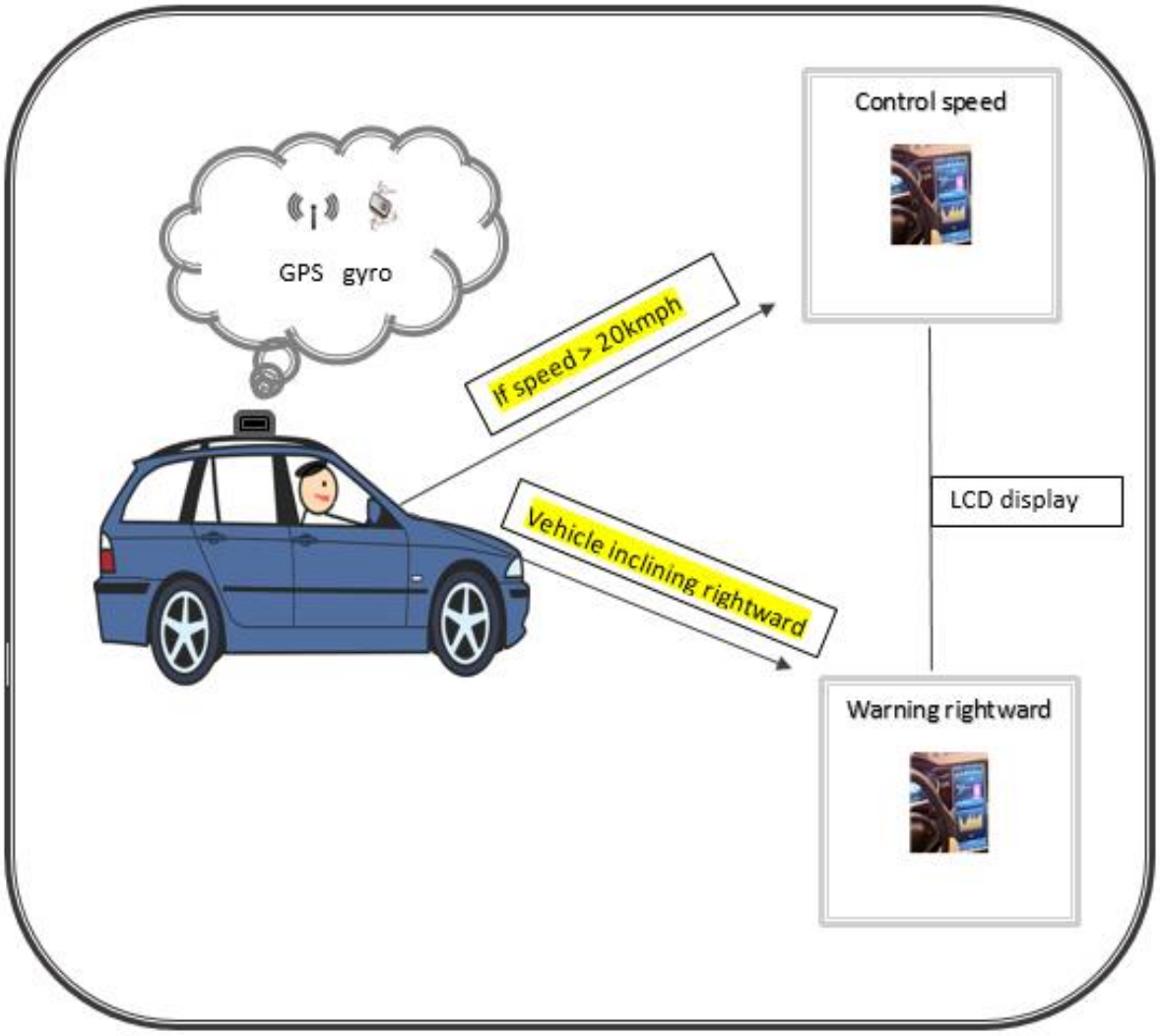


Figure 12 System architecture of Alert for safe driving

This section comprises of alert/notification system that identifies current condition and suggest the driver immediately if the vehicle is showing unusual behavior. This section includes a microcontroller, gps module and MPU6050 sensor that is installed in the vehicle. It also includes I2C LCD display that is visible from driver. The LCD is capable of displaying the current inclination readings of vehicle along x and y axis. In case of normal driving, the display will continuously show a message “Normal Riding”. When the vehicle is showing unusual behavior like leaning and inclining towards certain direction, for example, inclination towards right direction, the display will show alert message “Warning Rightward” and alert the driver for safe driving. Also if the speed of vehicle cross the threshold value, for example, if speed is over 20 kmph, the display will show alert message “Control speed”.

6.3 Alert after the accident

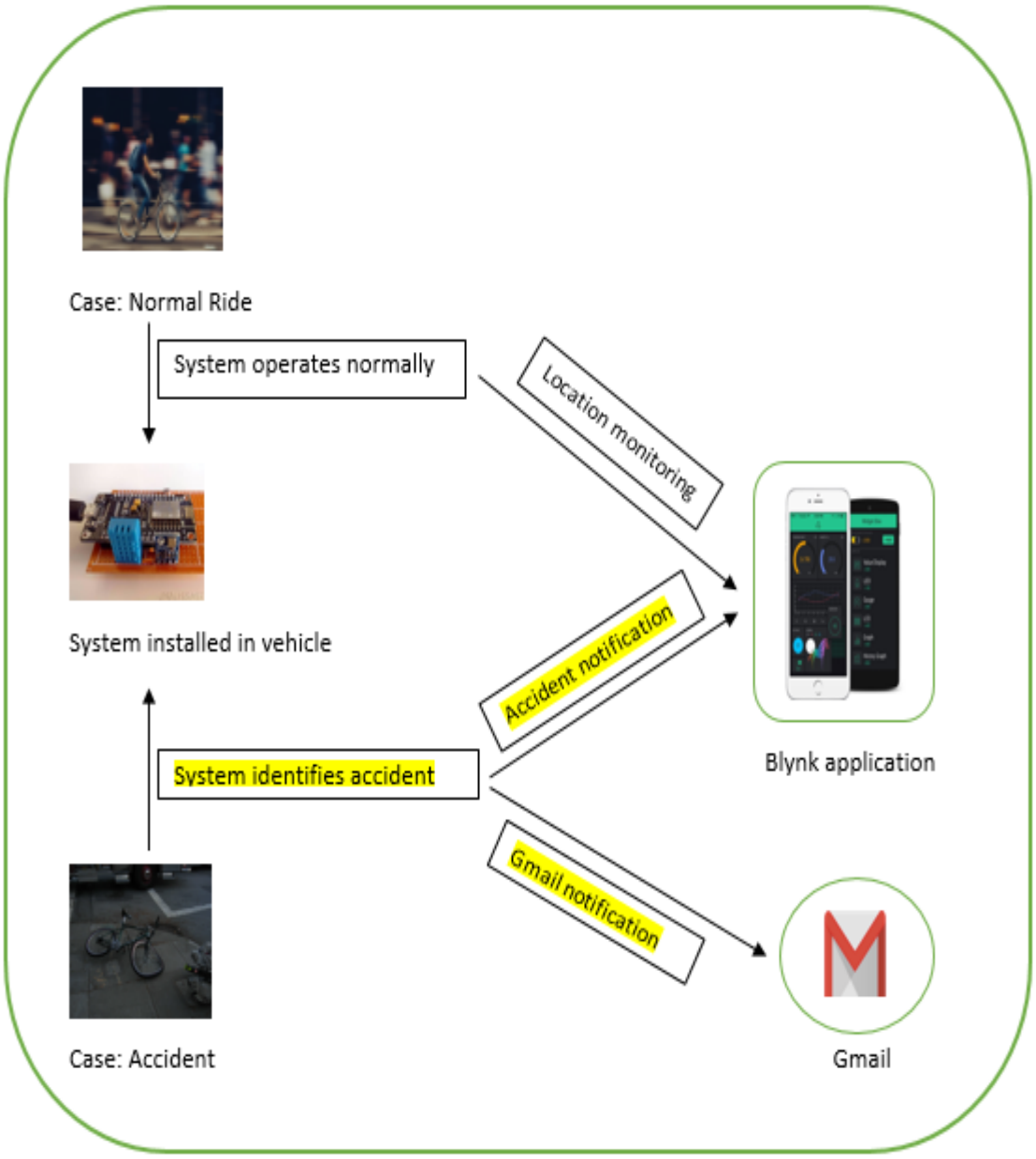


Figure 13 Alert after the accident

This section is designed as a backup of section 2. Although section 2 alerts the driver for safe driving and prevents accident, however during uncertainties and accidents, we need an immediate response or rescue operations so that the victim gets less affected. This section mainly comprises of alert/notification system that takes place if the vehicle encounters an accident. The vehicle is said to meet an accident if it is falling/inclining to the ground. The condition for accident is defined by putting a

threshold value of inclination angle along x and y axis. Therefore we will have four threshold values for front, back, right and left inclination. If the angle of inclination of vehicle exceeds the threshold value, the system identifies the condition as an accident. After that it quickly notifies about the accident to the concerned authority who is monitoring the vehicle with a blynk message notification. Also an immediate email is sent to the registered email address to carry out the rescue operations. After receiving the blynk notification, the concerned authority can view the final position of vehicle where the accident occurred through the virtual pins and Google map and hence carry out further actions. The connection between the microcontroller and MPU6050 sensor is as follow:

NodeMCU microcontroller	MPU6050
3v3	Vcc
GND	GND
D1	SCL
D2	SDA

6.4 Data display section

This section is mainly designed to alert the driver of vehicle by showing an alert message if vehicle is showing unusual behavior and to suggest the driver for safe driving. This section includes an I2C LCD display that is made visible for the driver of the vehicle. The display is directly connected to the microcontroller and is used to display the inclination readings of the vehicle which is calculated using MPU6050 sensor. The display shows an alert message if the vehicle is inclining to certain direction. It also shows an alert message if the speed of vehicle crosses the threshold speed. The connection between the microcontroller and I2C board is as follow:

NodeMCU microcontroller	I2C board
3v3	Vcc
GND	GND
D1	SCL
D2	SDA

6.5 Circuit diagram and threshold value

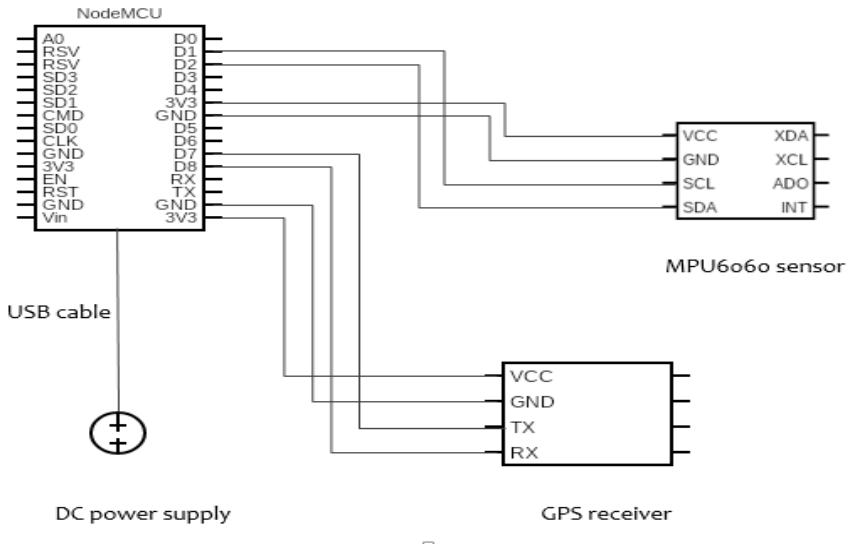


Figure 14 Circuit diagram

In this system, we have set some threshold values for determining certain conditions which are described below:

- Threshold for speed:
  We have defined an alert condition for speed and put the threshold value as 20 kmph. It means when the speed of vehicle crosses the threshold value of 20kmph, then the system identifies the condition as alert case and quickly notifies the driver of vehicle about the over speed by displaying an alert message “Control >>> speed” in the LCD display.
- Threshold for angle along x-axis:

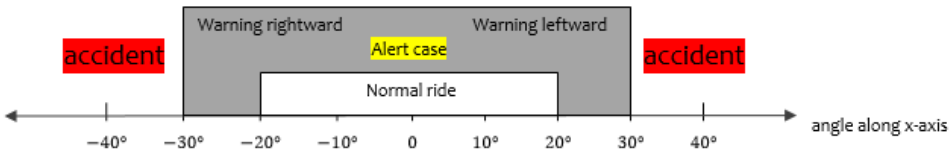


Figure 15 Threshold for angle along x-axis

We have defined a threshold condition for normal riding if the angle of inclination along the x-axis i.e. {X <= 20 && X >= -20}. Also we have set the threshold condition for the alert case along x-axis. If the angle of inclination along x-axis i.e. {X >=21 && X <= 30}, the condition is identified as alert case of leftward inclination since the vehicle is inclining leftward. Hence a warning message will be displayed in the LCD as "Warning Leftward". And if {X <= -21 && X >= -30}, it is identified as alert case of rightward inclination and hence a warning message is displayed in the LCD as "Warning Rightward".

- Threshold for angle along y-axis:

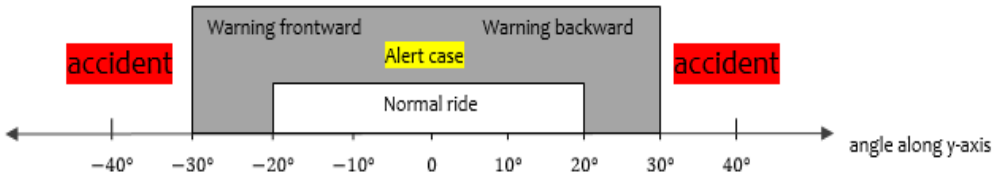


Figure 16 Threshold for angle along y-axis

We have defined a threshold condition for normal riding if the angle of inclination along the y-axis i.e. {Y <= 20 && Y >= -20}. Also we have set the threshold condition for the alert case along y-axis. If the angle of inclination along y-axis i.e. {Y >=21 && Y <= 30}, the condition is identified as alert case of backward inclination since the vehicle is inclining backward. Hence a warning message will be displayed in the LCD as "Warning Backward". And if {Y <= -21 && Y >= -30}, it is identified as alert case of frontward inclination and hence a warning message is displayed in the LCD as "Warning Frontward".

CHAPTER 7: IMPLEMENTATION AND RESULT

After defining and working on the system architecture, there was a challenge to implement the system practically. We have conducted the project in a research level and now on we needed to see if the system was practically working or not. Since it is very expensive to burrow the vehicle like buses, cars and it is also risky to bear the expenses for big vehicles, we decided to implement it using our own bicycle.

The figure below shows the System that we have developed using a NodeMCU ESP8266, MPU6050, GPS module and Blynk API:

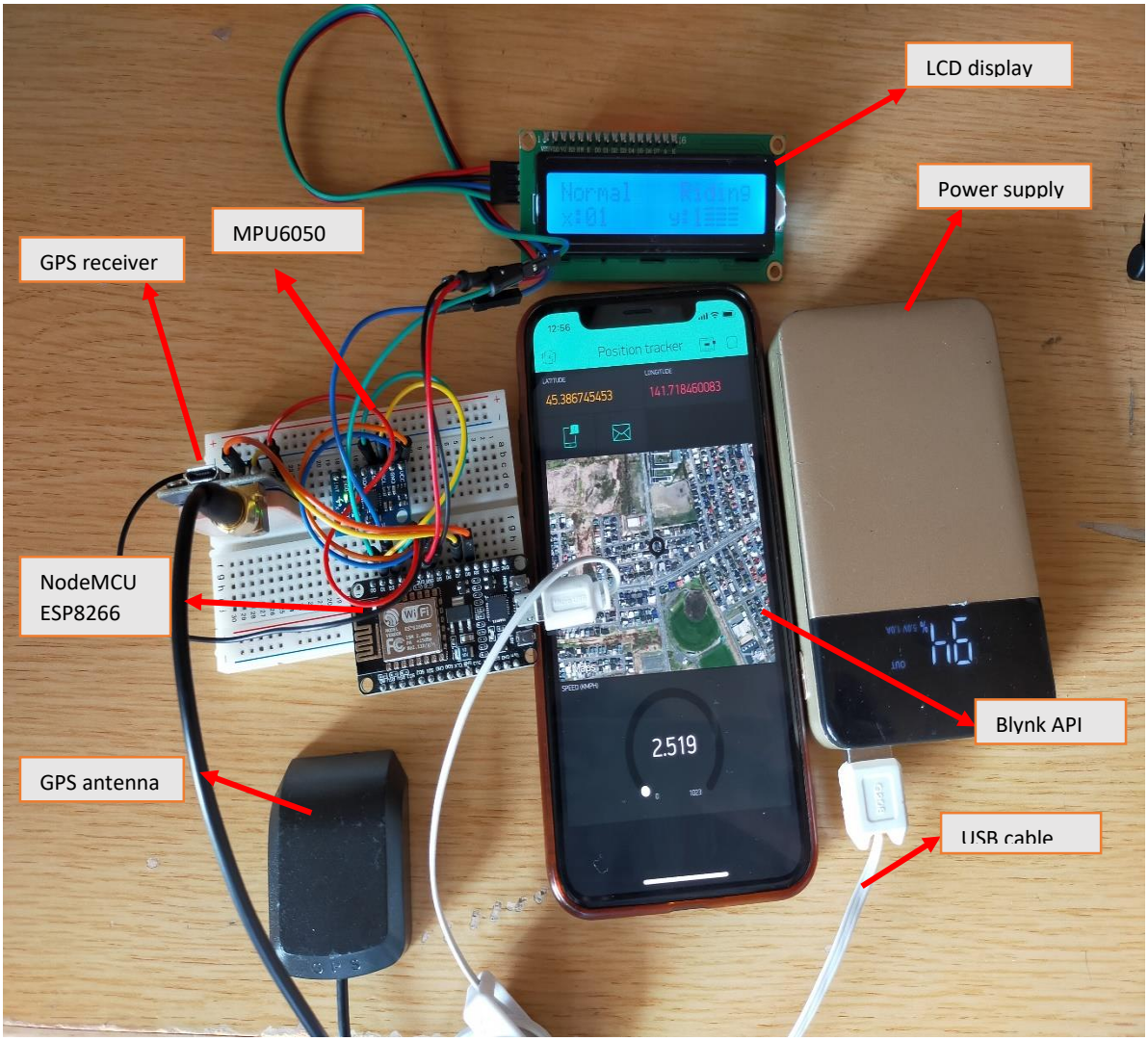


Figure 17 Overall view of the system we assembled

The figure below shows the interface that we prepared in Blynk API using virtual pins and Google map:

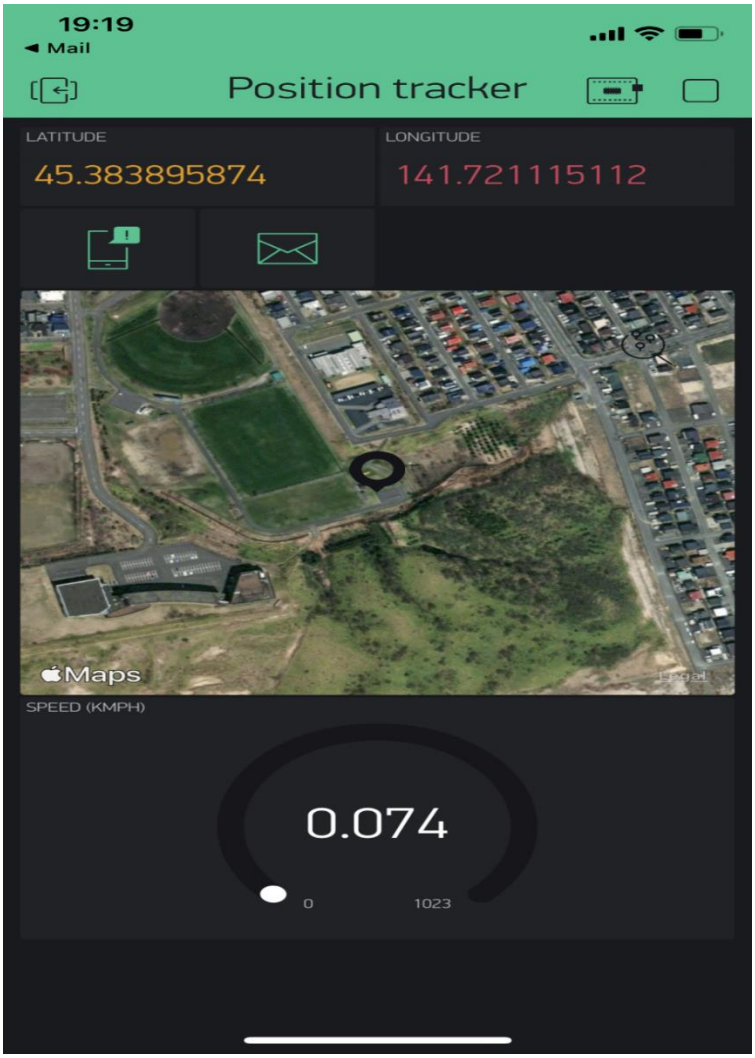


Figure 18 Blynk API for the project

The blynk interface includes the virtual pins V1 and V2 that shows the latitude and longitude values respectively. Similarly email and notification are also installed. There is a map widget for visualizing the real time location and also a gauge to show the speed of the vehicle in kmph.

The figure below shows how the devices are set up in our bicycle:

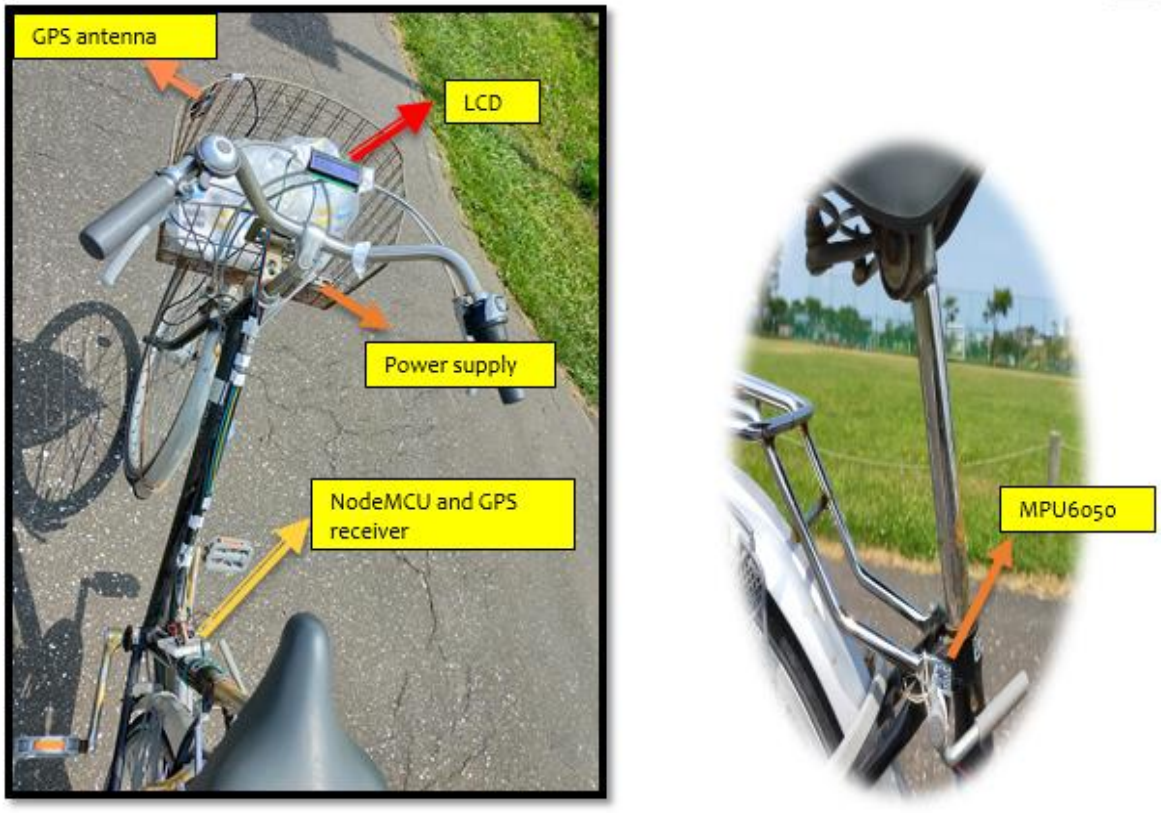


Figure 19 Set up of the devices

The LCD is set up in a position that is visible from rider. Also the MPU6050 sensor is set up below the seat and calibrated with initial readings 0 along x and y axis.

The figure below shows the essential gps data and angle of inclination data that we extracted from our gps receiver and gyro sensor respectively. We can see the values of Roll and Pitch as inclination angle along x and y axis respectively. Also the latitude, longitude and speed is also extracted using gps. The extracted data looks like below in the serial monitor of Arduino IDE:

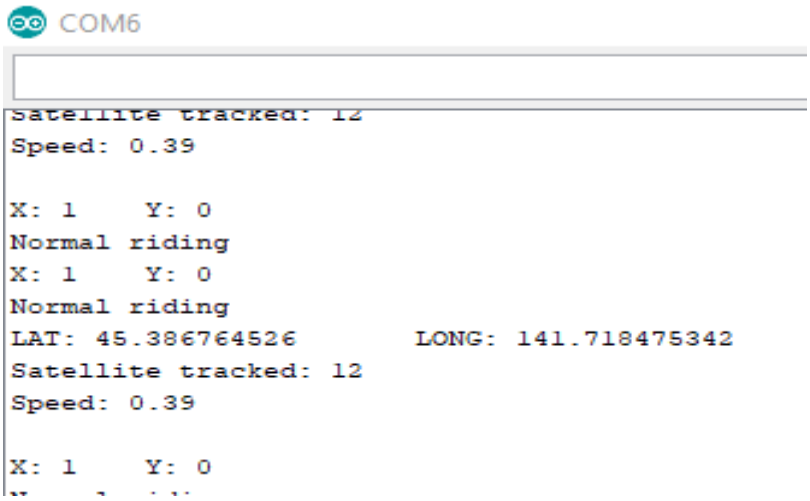


Figure 20 Data visualization in the serial monitor of Arduino IDE.

The figure below shows the live data monitoring of the vehicle through the embedded virtual pins and Google map.

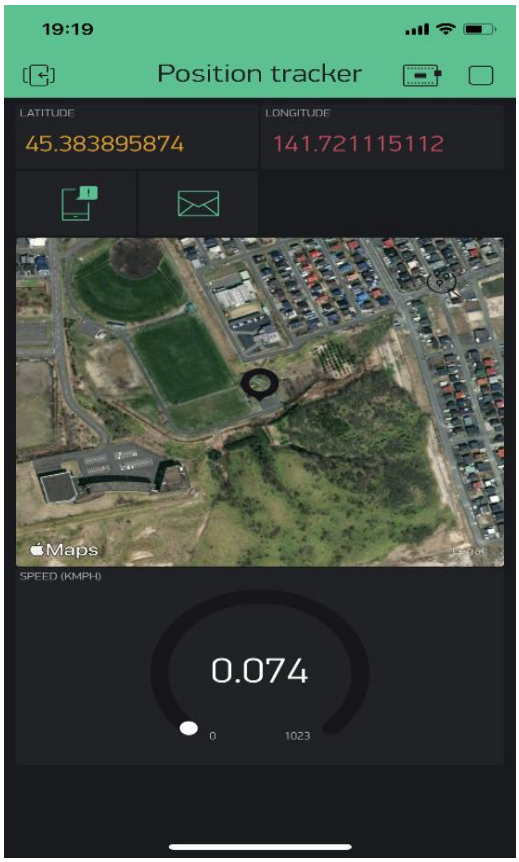


Figure 21 Live monitoring of vehicle through Blynk API

The figure below is the case of normal riding. We normally drove the bicycle and we could view “normal riding” in LCD and also inclination values along x and y axis.

Figure 22 Normal riding of vehicle

The figures below are the alert cases for directional inclination. Since the vehicle is inclining towards left, the LCD shows a warning message “Warning Leftward” and “Warning Rightward” when it is inclining right.

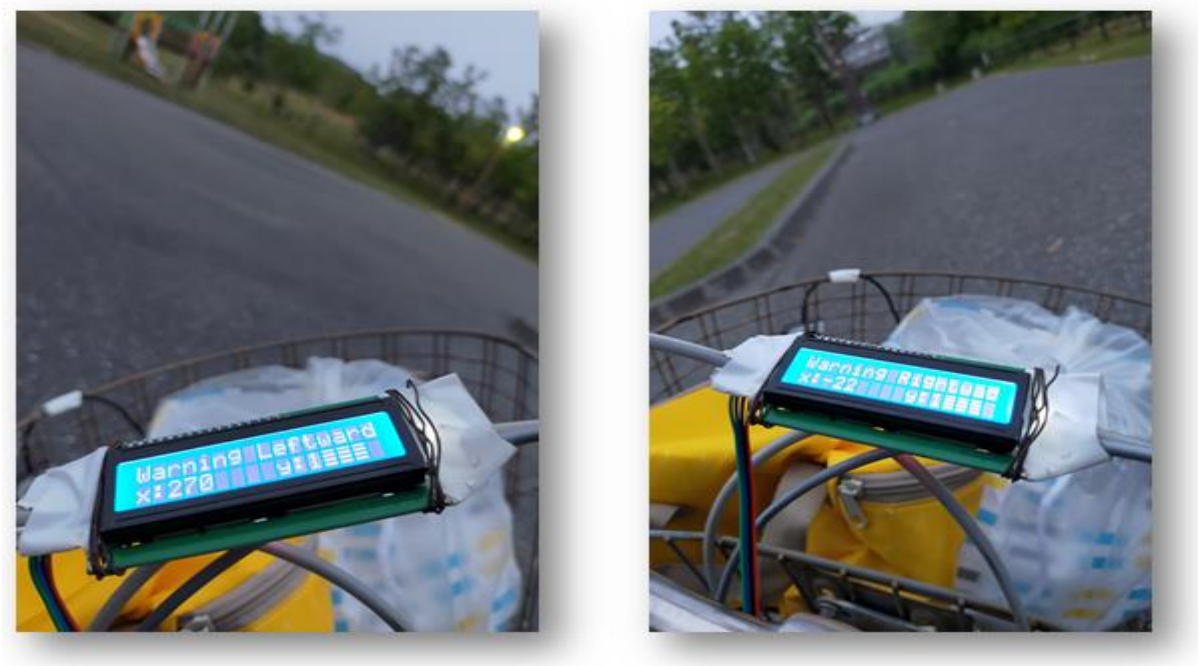


Figure 23 Directional inclination of vehicle

The figure below is the case when speed of vehicle crosses the threshold value. We have set the threshold value for speed as 20 kmph, so when the bicycle is running with speed over 20kmph as in the first figure below, a message is immediately displayed in the LCD as “CONTROL>>>>SPEED” as in the third figure below. The speed of vehicle can be viewed through the blynk API too, as in the second figure.

Figure 24 Speed of vehicle is over threshold, Blynk view, LCD view

The figure below is the case of accident. When the bicycle encountered an accident, we received a blynk notification in our mobile. After that we visualized the final position where the accident occurred. The third figure below shows that after the accident, an email is delivered immediately at the same time to the registered email address indicating “Accident occurred, Rescue needed!”

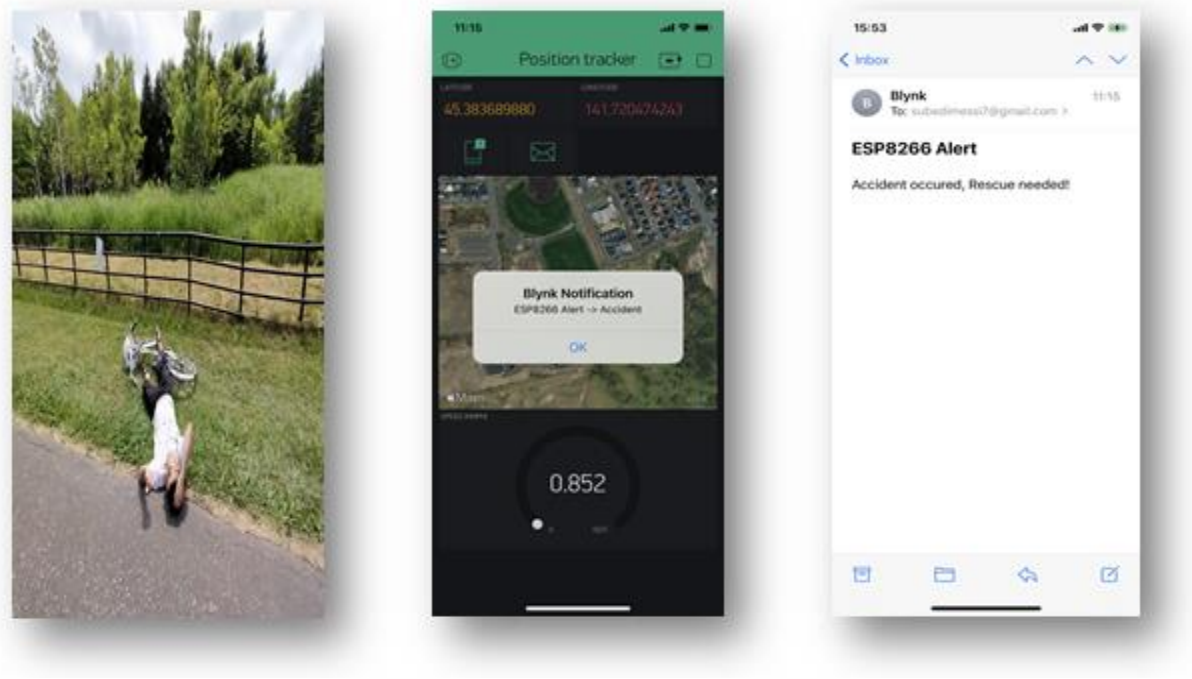


Figure 25 Case of accident, Blynk notification, and Gmail notification

CHAPTER 8: CONCLUSION

In this research, we developed a Vehicle monitoring and accident alert system based on IoT, using NodeMCU ESP8266, MPU6050 sensor, GPS receiver and Blynk mobile application. Although the system is practically implemented using a bicycle, which is comparatively smaller than other vehicles, there will not be any problem to implement it in other big vehicles, however the position to setup the device maybe different. The system can monitor the position of vehicle, wherever it is, in a real time using Blynk API. Also the concept of safe driving is introduced in this research as the system is able to identify the condition where the vehicle is showing unusual behavior or very near to encounter accident, and immediately notify the driver of the vehicle for safe driving. However if the driver is unable to prevent the accident, there is also a backup plan in the system as the system quickly identifies the accident and notifies the concerned authority about the accident hence the authorities can carry out further actions or rescue operations accordingly.

The vehicle monitoring and accident alert system that we designed in this research is mainly expected to assist the Vehicle Rental organization since they need to keep regular record of their vehicle. Furthermore, in context of our country Nepal where the vehicle operates under the direction of Vehicle management committee, this system might help the management committee in monitoring and scheduling the vehicle services more appropriately. Regarding the rural context, if the system is implemented in such vehicles that run in rural areas, immediate rescue operations can be conducted in case of emergency and accident hence save the life of victims.

CHAPTER 9: LIMITATION AND FUTURE WORK

In this research, we are able to monitor the real time position of vehicle using Blynk API. In future, our works will be more focused on creating our own web Interface and be able to monitor the vehicle in real time. The other limitation of this research is that the term “Accident” in “Vehicle Monitoring and Accident Alert System” is mainly based only on angle of inclination of vehicle from the normal position. The angle of inclination is not just a sufficient parameter to determine an accident. Other parameters like vibration, shock, fire, sudden break, etc., which are not introduced in this system, might also play a vital role in determining the accident, which we realized during the experimenting phase of our research. For example, we can introduce a vibration/shock sensor to obtain the shock value during the collision of vehicle and define a criteria of accident. Due to the lack of time, we were not able to introduce these parameters in our system. It can also be a good platform for the junior students to work on the other parameters of accident and make the system more effective. Hence our future work will be focused on introducing other parameters of accident in our system.

REFERENCES

[1] IoT, https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT (04 July 2021)

[2] Designing Efficient Access Control to Comply Massive-Multiservice IoT over Cellular Networks - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/IoT-Applications-Vertical_fig1_320830857 (04 July 2021)

[3] GPS, https://en.wikipedia.org/wiki/Global_Positioning_System (05 July 2021)

[4] GPS segment, http://allaboutgps101.blogspot.com (05 July 2021)

[5] Control Segment, https://www.gps.gov/systems/gps/control/ (05 July 2021)

[6] TinyGPS++, http://arduiniana.org/libraries/tinygpsplus/ (10 July 2021)

[7] MPU6050_light, https://github.com/rfetick/MPU6050_light (10 July 2021)

[8] Mallidi, S Kumar Reddy. (2018). IOT BASED SMART VEHICLE MONITORING SYSTEM. International Journal of Advanced Research in Computer Science. 9. 738-741. 10.26483/ijarcs.v9i2.5870

[9] Maurya, Kunal & Singh, Mandeep & Jain, Neelu. (2012). Real Time Vehicle Tracking System using GSM and GPS Technology-An Anti-theft Tracking System. International Journal of Electronics and Computer Science Engineering. 1.

[10] J. Singh, V. Velu and U. Nirmal. (2021). Vehicle Accident Detection System using Internet of Things (VADS -IoT). IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE). pp. 353-359. 10.1109/ISCAIE51753.2021.9431813.

[11] NodeMCU ESP8266, https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet (11July 2021)

[12] MPU6050, https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module (12July 2021)

[13] I2C LCD display, https://lastminuteengineers.com/i2c-lcd-arduino-tutorial/ (15 July 2021)

[14] Arduino software, https://www.arduino.cc/en/Guide/Environment (16 July 2021)

[15] Blynk, https://docs.blynk.cc/ (17 July 2021)

[16] Accident data, https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries (27 July 2021)

[17] GPS tracker with GSM/GPRS using Blynk and SMS, https://www.circuitschools.com/gps-tracker-with-gsm-gprs-using-blynk-with-sms-and-calling-features/ (4 July 2021)

[18] NodeMCU GPS Tracker, https://www.electroniclinic.com/nodemcu-gps-tracker-using-arduino-nodemcu-esp8266-and-blynk/ (5 July 2021)

[19] Accident Alert System using IoT and Mobile application, https://www.ijrte.org/wp-content/uploads/papers/v7i5s2/ES2059017519.pdf (5 July 2021)

APPENDIX

Arduino Code

gyro_gps.ino

```
#include "Wire.h"
#include <MPU6050_light.h>
#include <Blynk.h>
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F, 16, 2);

MPU6050 mpu(Wire);

static const int RXPin = 13, TXPin = 15;   // GPIO 13=D7(conneect Tx of GPS) and
GPIO 15=D8(Connect Rx of GPS)
static const uint32_t GPSBaud = 9600;

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;

char auth[] = "authentication key";          //Your Project authentication key
char ssid[] = "Ramesh";                         // (HotSpot or Router name)
char pass[] = "password10";                          // Corresponding Password

//unsigned int move_index;        // moving index, to be used later
unsigned int move_index = 1;      // fixed location for now

void(* resetFunc) (void) = 0;
```

```
void setup()
{
  Serial.begin(115200);
  Serial.println();
  ss.begin(GPSBaud);
  Blynk.begin(auth, ssid, pass);
  Timer.setInterval(5000L, checkGPS); // every 5s check if GPS is connected

  lcd.init();   // initializing the LCD
  lcd.backlight(); // Enable or Turn On the backlight

  Wire.begin();
  byte status = mpu.begin();
  Serial.print(F("MPU6050 status: "));
  Serial.println(status);
  while(status!=0){ }

  Serial.println(F("Calculating offsets, don't move MPU6050"));
  delay(1000);
  mpu.calcOffsets();
  Serial.println("Done!\n");
}

void checkGPS(){
  if (gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected: check wiring."));
      Blynk.virtualWrite(V1, "GPS ERROR");
  }
}

void displayInfo()
{
  if (gps.location.isValid() )
  {
    float latitude = (gps.location.lat());
    float longitude = (gps.location.lng());
```

```
   Serial.print("LAT: ");
   Serial.print(latitude, 9);
   Serial.print("\tLONG: ");
   Serial.println (longitude, 9);
   Serial.print("Satellite tracked: ");
   Serial.println(gps.satellites.value());
   Blynk.virtualWrite(V1, String(latitude, 9));
   Blynk.virtualWrite(V2, String(longitude, 9));
   myMap.location(move_index, latitude, longitude, "GPS_Location");
   spd = gps.speed.kmph();
   Serial.print("Speed: ");
   Serial.println(gps.speed.kmph());
   Blynk.virtualWrite(V3, spd);
  }
 Serial.println();
}

void loop()
{
   while (ss.available() > 0)
   {
     if (gps.encode(ss.read()))
       displayInfo();
  }

  short int X,Y;
  X = mpu.getAngleX();
  Y = mpu.getAngleY();
  mpu.update();
  Serial.print("X: ");
  Serial.print(X);
  Serial.print("\tY: ");
  Serial.println(Y);

   lcd.setCursor(0, 1);
   lcd.print("x:");
```

```
   lcd.print(X);
   lcd.setCursor(9,1);
   lcd.print("y:");
   lcd.println(Y);

  lcd.setCursor(0, 0);
  if (spd > 20){
   lcd.print("CONTROL>>>>SPEED");
  }
  else if (X <= 20 && X >= -20 && Y <= 20 && Y >= -20) {
   Serial.println("Normal riding");
   lcd.print("Normal    Riding");
  }
  else if(Y >=21 && Y <= 30) {
   Serial.println("Warning! The vehicle is tilting backward, control!");
    lcd.print("Warning Backward");
  }
  else if(Y <= -21 && Y >= -30) {
   Serial.println("Warning! The vehicle is tilting frontward, control!");
   lcd.print("Warning Frontwad");
  }
  else if(X >=21 && X <= 30) {
   Serial.println("Warning! The vehicle is tilting towards left, control!");
   lcd.print("Warning Leftward");
  }
  else if(X <= -21 && X >= -30){
  Serial.println("Warning! The vehicle is tilting towards right, control!");
  lcd.print("Warning Rightwad");
  }
  else {
   Serial.println("Accident Occured!");
   lcd.print("Accident occured");
   Blynk.email("subedimessi7@gmail.com", "ESP8266 Alert", "Accident
occured, Rescue needed!");
   Blynk.notify("ESP8266 Alert -> Accident");
   delay(120000);
   resetFunc();
```

```
   }

  Blynk.run();
  Timer.run();
}
```