Project report submitted in partial fulfillment for the Degree of B. Tech in Applied Electronics & Instrumentation Engineering under Maulana Abul Kalam Azad University of Technology

RASPBERRY-PI BASED GESTURE CONTROLLED CAR

By

Abul Hassan (Roll No.11705515002)

Shibam Jana (Roll No. 11705515042)

Rishav Chakraborty (Roll No.11705515036)

Tridib Dey (Roll No. 11705515052)

Under Supervision of

Ms. NAIWRITA DEY

Assistant Professor

Department of AEIE, RCCIIT

DEPARTMENT OF APPLIED ELECTRONICS & INSTRUMENTATION ENGINEERING,

RCC INSTITUTE OF INFORMATION TECHNOLOGY, CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700015,

May, 2019

ACKNOWLEDGEMENT

It is a great privilege for us to express our profound gratitude to our respected teacher Ms. NAIWRITA DEY, Applied Electronics &Instrumentation Engineering, RCC Institute of Information Technology, for her constant guidance, valuable suggestions, supervision and inspiration throughout the course work without which it would have been difficult to complete the work within scheduled time.

We are also indebted to the Head of the Department,Prof.Arijit Ghosh,Applied Electronics & Instrumentation Engineering, RCC Institute of Information Technology for permitting us to pursue the project.

We would like to take this opportunity to thank all the respected teachers of this department for being a perennial source of inspiration and showing the right path at the time of necessity.

Abul Hassan (Roll No.11705515002)

Shibam Jana (Roll No. 11705515042)

Rishav Chakraborty (Roll No.11705515036)

TridibDey (Roll No. 11705515052)

RCC INSTITUTE OF INFORMATION TECHNOLOGY
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700 015
PHONE : 2323 2463 FAX : (033)2323 4668
E-mail : campus@rcciit.in Website : www.rcciit.org

CERTIFICATE OF APPROVAL

The project report titled "**Raspberry pie gesture controlled car**" prepared by Abul Hassan (Roll No.11705515002),Shibam Jana(Roll No.11705515042),Rishav Chakraborty(11705515036),Tridib Dey(Roll No.11705515052); is hereby approved and certified as a creditable study in technological subjects performed in a way sufficient for its acceptance for partial fulfilment of the degree for which it is submitted.

It is to be understood that by this approval, the undersigned do not, necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

[Ms. Naiwrita Dey]
Applied Electronics & Instrumentation Engineering

Examiner

Examiner

RCC INSTITUTE OF INFORMATION TECHNOLOGY
CANAL SOUTH ROAD, BELIAGHATA, KOLKATA – 700 015
PHONE : 2323 2463 FAX : (033)2323 4668
E-mail : campus@rcciit.in Website : www.rcciit.org

RECOMMENDATION

I hereby recommend that the project report titled "**Rasberry pi based gesture controlled car**" prepared by Abul Hassan (Roll No.11705515002),Shibam Jana(Roll No.11705515042),Rishav Chakraborty(11705515036),Tridib Dey(Roll No.11705515052): be accepted in partial fulfillment of the requirement for the Degree of Bachelor of Technology in Applied Electronics & Instrumentation Engineering, RCC Institute of Information Technology.

[Ms. Naiwrita Dey]

[Mr. Arijit Ghosh]
Applied Electronics & Instrumentation Engineering

Abstract

In this project a novel approach has been proposed about the controlling of a cart using human hand gestures. The proposed scheme is divided into three parts. Data glove is developed by detecting six different human hand gestures by using digital accelerometer and node mcu and the hand gestured command was sent to the receiver part which is basically controlled by the raspberry pi. Different statistical features are considered for feature recognition. The cart has been driven using raspberry pi. The system is made wireless. MQTT protocol is used for sending command from data glove to cart. An wireless network is developed with same gateway address . Digital 3 axis accelerometer ADXL345 is used here. Node mcu and Raspberry pi 3B model is used for the project.

List of Figures:

List of table:

Content

Chapter 1: Introduction

There are many ways to control a robot we had heard , like them a robot can be controlled using our body movements too, even our several hand gestures can control a robot or more interestingly it can be controlled by our brain signals too which is also known as nuro signals, like without giving any effort to our body one can operate some moving thing according to our intention. This project is also based upon the concept of gesture controlled things. The main purpose or the utility of the project should be to help the handicapped people, basically who can't through their legs. The main attraction of our project is a wheel-chair which can be controlled by the movement of the handicapped person's hand palm. That concept of that work is actually divided into three parts, like Transmitting section, Networking section and Receiver section. Basically the part of gesture recognition and transmitting our hand gesture command which will forward to the final car belong to the transmitting section and the signal will received at the receiver section. In a very short manner it can be said an accelerometer will sense our hand gesture and send it to node mcu, which is going to transmit that signal to the receiver section. The signal transmission will occur absolutely in a wireless manor to avoid the complex arrangement of wires. The final controller will receive the transmitted signal and it will operate the motor wheels. That was the basic idea of our project which is actually a demonstration of man machine interface.That is actually a demonstration a wireless communication system.These kind of transmissions are based on several protocol through which information can be transferred between two different controllers, which is a very modern technique we had applied here to make our project more attractive. There was a little description of the basic idea and some inner operation of that project at the introduction part. Here some elaborated description which has provided here. As it hasmentioned earlier that the total work has divided, the total operation of this project activity into threeparts. Where the initial one is to data acquisition from accelerometer and encode the accelerometer data by Node-mcu and finally transmit the data to raspberry pi. The final one is to receive the transmitted data and decode it and use it to drive the motor driver [1]. So initially comes that part where the command was given to an object, means the gesture recognition. Here ADXL345 had used, which was a digital accelerometer. It has digital interface supply voltage pin(vcc), a ground pin, chip select (cs) pin, INT1 and INT2 (interrupt 1 and 2), SDO (serial data output), SCL (Serial communication clock) and SDA (serial data input). Each pin of ADXL345 accelerometer has their own specifications which will be shown at the operation of accelerometer [2]. The accelerometer was set at the upper part of our palm, because the upper part of the palm is a very suitable position and that part is very nearer to our wrist. So when the wrist had moved to left, right, up and down direction accordingly the accelerometer will move. The accelerometer had set on a hand-glove at the right hand. According to the direction wise movement the accelerometer will give some data according to the gesture direction which will be represented in a graphical manor.From the several data of the accelerometer of the four above mentioned directions or the gestures we have locked those data to an excel sheet and also made a graphical representation of those data [3]. After locking the raw data of our hand gestures by ADXL345 accelerometer, the data was prepared for wireless transmission. For that there have to encode that data. For this step there Node MCU (ESP 8266) was used. With the help of suitable codes for the three axis data of accelerometer and here as a software Arduino Ide used for this purpose [4]. Here there have used basically ESP8266 model of node MCU which is basically a Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. ESP8266 is a 3V WiFi module very popular for its Internet of Things applications. ESP 8266 maximum working Voltage is 3.6V and its very important. The ESP8266WebServer library allows us to run an ESP8266 as a basic webserver and access point. This can process data received from a remote sensor over Wi-Fi without connecting the devices to a network or router [5].So based upon this feature the data transfer had been possible to the other controller we had used. As the other controller which was known as Raspberry pi3 B+, which is indeed a totally different kind of controller than the Node MCU we had used, so, here a protocol has used which was liable to represent same kind of data which any one can see at node MCU and Raspberry pi too. This protocol is known as MQTT protocol. This is also a wireless telemetry protocol known as Message Queuing Telemetry Transport. It is a lightweight messaging protocol that provides resource-constrained network clients with a simple way to distribute telemetry information. The protocol, which uses a publish/subscribe communication pattern, is used for machine-to-machine (M2M) communication and plays an important role in the internet of things. It enables a resource constrained IOT device to send information to a server that works as MQTT message broker. This broker pushes the information to those clients which are previously subscribed [6]. There are several important steps which are important for enable this protocol. So there it was necessary to install some packages and client. There it was also necessary to follow several commands. At the pi command window we had updated the signing key using suitable commands. After there was added the mosquitto repository to apt-get and then further downloaded the repository list file for MQTT. After that at the MQTT installation part there was updated the apt-get source list and after that there was installed the package of MQTT broker. As this project is quite python based paho-mqtt was installed to replace the old version of mosquito. After that MQTThad configured and after making set all these things it was tested the whole set up by using two command window by saying “Hello pi” which was visualized in both command window [7]. After those steps there comes the matter of arduino ide installation. There were initially installed several libraries from Arduino library and after that there’s toran some codes and in this way the interfacing part was done [8]. After that there comes the final most steps. After achieving the command of accelerometer by a prolonged process by Raspberry pi there was a suitable circuital connection made among the motor driver had used and the motors which are connected with wheels and most importantly with the pi board which is the controller [9]. After that was shown according to the commends means according to the movements of the human palm, which were previously assigned the car like setup moved [10].In this paper,a method is presented for a gesture recognition using a 3-axis sensor.In this system,the user gives the input signal,and it is received by the 3-axis sensor,the raw data is then adjusted by means of the acceloremeter ,then the data is reduced dimensionally by using principal component analysis(PCA),then linear discriminant to complete the feature extraction.The feature data set is then rearranged for further analysis,then pattern classification is done using support vector machine(SVM),then labels are assigned as labels for training the appropriate models on them,finally the result is analysed to give the accuracy [11].With the aim of easing Human Machine Interaction (HMI),hand Gesture Recognition Systems (GRS) have emerged as anew feasible method to naturally communicating withmachines. HMI can be achieved by two ways: vision-based and wearablesensor-based,the former is the better method to employ,as it does not require wearing complex sensors,thereby rendering it as convenient.However,certain restrictions arise from using this system such as the fixed camera position,low light conditions for non detection of imaging capabilities of the camera,in this situation using a sensor-based system is much feasible.Here,an accelerometer sensor is used for the gesture recognition system.For pattern recognition,feature extraction method is used.A number of common features were extracted from the raw data of the accelerometer in each of the 3 axis(x,y and z),namely mean,difference,standarddeviation,variance,root mean square,average of absolute deviation.All of these features were taken from each of the 3 axis of the accelerometer.All of this features are extracted from the data of the accelerometer and some of this features are clubbed together in a set. Different methods are used to find the feature set with the highest recognition rate.[12]

The key of gesture recognition is to extract effective features which reflect the motion characteristics of different gestures.Since a complex gesture is segmented into several basicgestures, here we just extract the features of basic gestures.When a user performs gestures on 3Dspace the movement is sensed by anaccelerometer. Then the acquired data is processed andclassified into a gesture through the gesture recognitionalgorithm. Finally, the corresponding function is executed andfeedback to the users.[13-14]A number of different features are to be extracted from the data acquired from the accelerometer,namely: mean,difference,standarddeviation,variance,root mean square,average of absolute deviation.

All of these features are performed on the acquired data from the accelerometer and clubbed into different datasets.These datasets are then used to find the gesture recognition rate by using various statistical feature extraction methods.The features are then again clubbed into different datasets.This process continues repeating until the dataset with the highest gesture recognition rate is found.It is seen that at the beginning there were a considerable number of datasets,but these continued to be reduced in number,after they were clubbed into different datasets and they were applied different statistical feature extraction methods.A dimension reduction could be seen from the original feature dataset and the final feature datasets.[15-16]The raw data is taken from the accelerometer ADXL345 for all the three axes(x,y,z).The accelerometer is interfaced with a nodemcu,which is a arduino board capable of transmitting data(in this case,accelerometer data)wirelessly to any other device.The data is gathered from the Arduino IDE from the serial monitor.The accelerometer data is organized into their respective axes on an MS Excel spreadsheet.A special software extension called PLX-DAQ allows the data from the excel speadsheet to be accessed by the Arduino code ,which then performs the necessary operations to extract the required features. Parallax Data Acquisition tool (PLX-DAQ) software add-in for

Microsoft Excel acquires up to 26 channels of data from any Parallax microcontrollers and drops the numbers into columns as they arrive. PLX-DAQ provides easy spreadsheet analysis of data collected in the field, laboratory analysis of sensors and real-time equipment monitoring.

Chapter 2: Methodology

Gesture Controlled Car is controlled by simple human gestures. The user just needs to wear a gesture device in which a sensor is included. The sensor will record the movement of hand in a specific direction which will result in the motion of the car in the respective directions. We can control the car using accelerometer sensors connected to a hand glove. The sensors are intended to replace the remote control that is generally used to run the car. It will allow user to control the forward, backward, leftward and rightward movements, while using the same accelerometer sensor to control the throttle of the car.

In case of Gesture Control Robot using Raspberry Pi 3 B+ , there have several steps which must be follow to run the motor and to operate the system successfully.

<u>Flow Chart :</u>

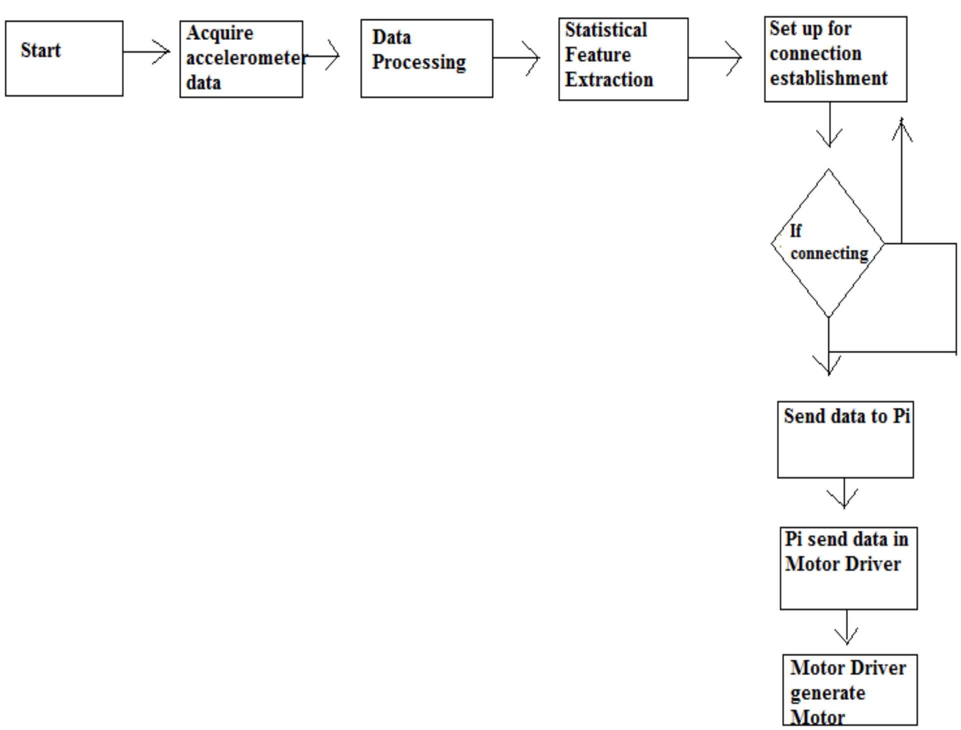


Fig 2.1 Block diagram of Gesture Controlled Car

In this project the methodology divides into three parts. These are :

A.. Data glove

B. Data transmission through wifi using MQTT protocol:

To establish the communication channel between the transmission and receiver section the MQTT protocol has used. There basically the Ip address based data transfer were done from Nodemcu to Raspberry pi. Both were connected through a common host which was basically the mobile phone's hotspot. Through which the connection was established.

C Cart/Car

Part 1: Data gloves

The data glove has motion sensors (accelerometer,nodemcu)on the back of the hand and on the hand movement to provide information regarding motion and orientation of the hand .Hand movement are controlling the direction of the robot in different side by sensing the sensor.Sensors mounted on the glove send signals to a processing unit, worn on the user'sforearm that translates hand postures into data. An RF transceiver, also mounted on the user, transmits the encoded signals representing the hand postures and dynamic gestures to the car via RF link.Communications, conveying intentions, distributed environment,gestureshuman-computer interactions, human-robot interaction.

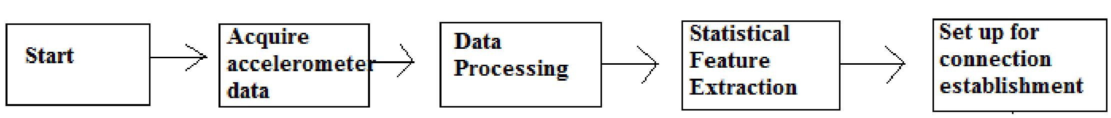


Fig 2.2: Steps of feature extraction

Node MCU :

It is used as controller for data glove. Node MCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espress if Systems, and hardware which is based on the ESP-12 module. Node MCU and Accelerometer are connected through wifi.

<u>Accelerometer</u> :

Accelerometers are electromechanical devices that sense either static or dynamic forces of acceleration. Static forces include gravity, while dynamic forces can include vibrations and movement.

we will interface our accelerometer ADXL 345 and MPU 6050 to our raspberry pi and check the readings of the sensor. After the raspberry pi is being installed with latest operating system and python in it, since we are going to use python code here.

Part 2: Data transmission through wifi using MQTT protocol

Part 3: Cart

The car have been created by the one wood board and four D.C motors are tide by the iron wire below the board and four motors are connected with four wheels. The upper part of the frame is contain raspberry pi 3 B+,L293d motor driver both device are connected and the motor connected with motors. Using battery to generated the car.

Development of cart model :

Cart is run properly by several steps which are

- firstlyone wood board and four D.C motors are tide by the iron wire below the board and four motors are connected with four wheels.
- Secondly raspberry pi 3 B+, L293d motor driver both device are connected and the motor driver connected with motors.
- Atlast battery is used to generate the robot.

.

Chapter 3: Data glove

A data glove is an interactive device, resembling a glove worn on the hand, which facilitates tactile sensing and fine-motion control in robotics and virtual reality. Fine-motion control involves the use of sensors to detect the movements of the user's hand and fingers, and the translation of these motions into signals that can be used by a virtual hand (for example, in gaming) or a robotic hand (for example, in remote-control surgery).

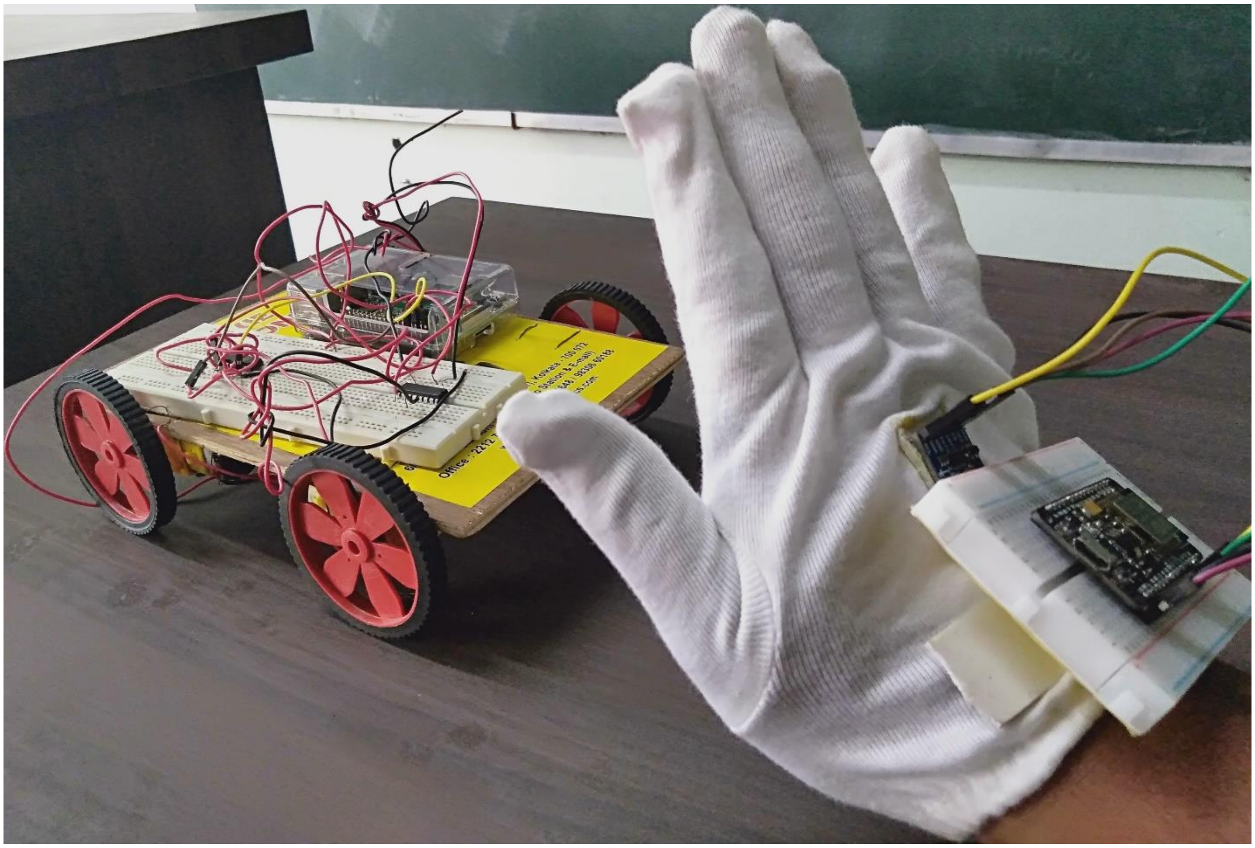


Fig 3.1 Data glove

The data glove has motion sensors(accelerometer,nodemcu)on the back of the hand and the hand movement provides information regarding motion and orientation of the hand .Hand movement are controlling the direction of the robot in different side by sensing the sensor. Sensors mounted on the glove send signals to a processing unit,NodeMcu worn on the user’s forearm that translates hand postures into data. The NodeMcu sends the data wirelessly via wifi to the raspberry pie which receives the data via MQTT protocol.The raspberry pie receives this data and sends the appropriate code to the motor driver,which drives the motors.

Feature Extraction

A feature is an attribute or property shared by all of the independent units on which analysis or prediction is to be done.

Any attribute could be a feature, as long as it is useful to the model.The purpose of a feature, other than being an attribute, would be much easier to understand in the context of a problem. A feature is a characteristic that might help when solving the problem.Results can be improved using constructed sets of application-dependent features, typically built by an expert. One such process is called feature engineering.

Feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is a dimensionality reduction process, where an initial set of raw variables is reduced to more manageable groups (features) for processing, while still accurately and completely describing the original data set.

When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features (also named a feature vector). Determining a subset of the initial features is called feature selection. The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data.

Feature extraction involves reducing the amount of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power, also it may cause a classification algorithm to overfit to training samples and generalize poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. Many machine learning practitioners believe that properly optimized feature extraction is the key to effective model construction.

General dimensionality reduction techniques are used such as:

- Independent component analysis
- Isomap
- Kernel PCA
- Latent semantic analysis
- Partial least squares
- Principal component analysis
- Multifactor dimensionality reduction
- Nonlinear dimensionality reduction
- Multilinear Principal Component Analysis
- Multilinear subspace learning
- Semidefinite embedding
- Autoencoder

With the aim of easing Human Machine Interaction (HMI),hand Gesture Recognition Systems (GRS) have emerged as anew feasible method to naturally communicating with machines. HMI can be achieved by two ways: vision-based and wearable sensor-based,the former is the better method to employ,as it does not require wearing complex sensors,thereby rendering it as convenient.However,certain restrictions arise from using this system such as the fixed camera position,low light conditions for non detection of imaging capabilities of the camera,in this situation using a sensor-based system is much feasible.Here,an accelerometer sensor is used for the gesture recognition system.For pattern recognition,feature extraction method is used.A number of common features were extracted from the raw data of the accelerometer in each of the 3 axis(x,y and z),namely mean,difference,standard deviation,variance,average of absolute deviation.All of these features were taken from each of the 3axis of the accelerometer.All of this features are extracted from the data of the accelerometer and some of this features are clubbed together in a set. Different methods are used to find the feature set with the highest recognition rate.[21]

The key of gesture recognition is to extract effective features which reflect the motion characteristics of different gestures.Since a complex gesture is segmented into several basic gestures, here we just extract the features of basic gestures.

Figure below shows an overview of the proposed framework for gesture interactive. When gestures are performed by a user on 3Dspace the movement is sensed by accelerometer Then the acquired data is processed and classified into a gesture through the gesture recognition algorithm. Finally, the corresponding function is executed and feedback to the users.

Feature Extraction :

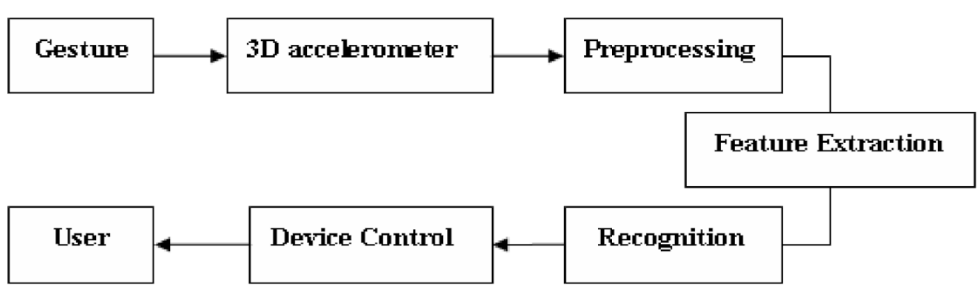


Fig 3.2 Block diagram of Feature extraction

A number of different features are to be extracted from the data acquired from the accelerometer,namely,mean,difference,standard deviation,variance,average of absolute deviation.

Time Domain features are extracted directly from the accelerometer raw data. Due to this reason, these features are the simplest and most efficient to compute. In this study the most common features in gesture recognition system(GRS) were extracted mean, difference,standard deviation,variance,average of absolute deviation.

Frequency domain features are very useful for developing the means to identifying physical activities such as walking, running or cycling. However, these kinds of features may be less helpful in hand gesture recognition because the duration is significantly shorter and the gesture may not present any significance in the frequency domain. To study the meaningfulness of these features in a gesture recognition system, the following features were added to the previous set, Energy, power band (0-2,75Hz), max entropy (0-1-2-3Hz).
All of these features are performed on the acquired data from the accelerometer and clubbed into different datasets.These datasets are then used to find the gesture recognition rate by using various statistical feature extraction methods.The features are then again clubbed into different datasets.This process continues repeating until the dataset with the highest gesture recognition rate is found.[22]

It is seen that at the beginning there were a considerable number of datasets,but these continued to be reduced in number,after they were clubbed into different datasets and they were applied different statistical feature extraction methods.A dimension reduction could be seen from the original feature dataset and the final feature datasets.

The raw data is taken from the accelerometer ADXL345 for all the three axes(x,y,z).The accelerometer is interfaced with a nodemcu,which is a arduino board capable of transmitting data(in this case,accelerometer data)wirelessly to any other device.The data is gathered from the Arduino IDE from the serial monitor.

The accelerometer data is organized into their respective axes on an MS Excel spreadsheet.A special software extension called PLX-DAQ allows the data from the excel speadsheet to be accessed by the Arduino code ,which then performs the necessary operations to extract the required features.

Up to 26 channels of data from any Parallax microcontrollers could be accessed by the Parallax Data Acquisition tool (PLX-DAQ) software add-in for Microsoft Excel which acquires and drops the numbers into columns as they arrive. Easy spreadsheet analysis of data collected in the field, laboratory analysis of sensors and real-time equipment monitoring is provided by PLX-DAQ .

PLX-DAQ is a Parallax microcontroller data acquisition add-on tool for Microsoft Excel. Any of the microcontrollers connected to any sensor and the serial port of a PC can now send data directly into Excel. PLX-DAQ has the following features:

- Data as it arrives in real-time using Microsoft Excel could be plotted or graphed.
- Up to 26 columns of data could be recorded.
- Data could be marked with real-time (hh:mm:ss) or seconds since reset.
- Could be Read/Written on any cell on a worksheet.
- Capable of Reading/Setting any of 4 checkboxes on control the interface.
- Baud rates up to 128K.
- Com1-15 are supported.

With the help of plx-daq,the sensor data could be datalogged from the accelerometer data(x,y,z)and it would be sent to the Arduinocode,which performs the necessary actions to extract the appropriate features from the dataset.

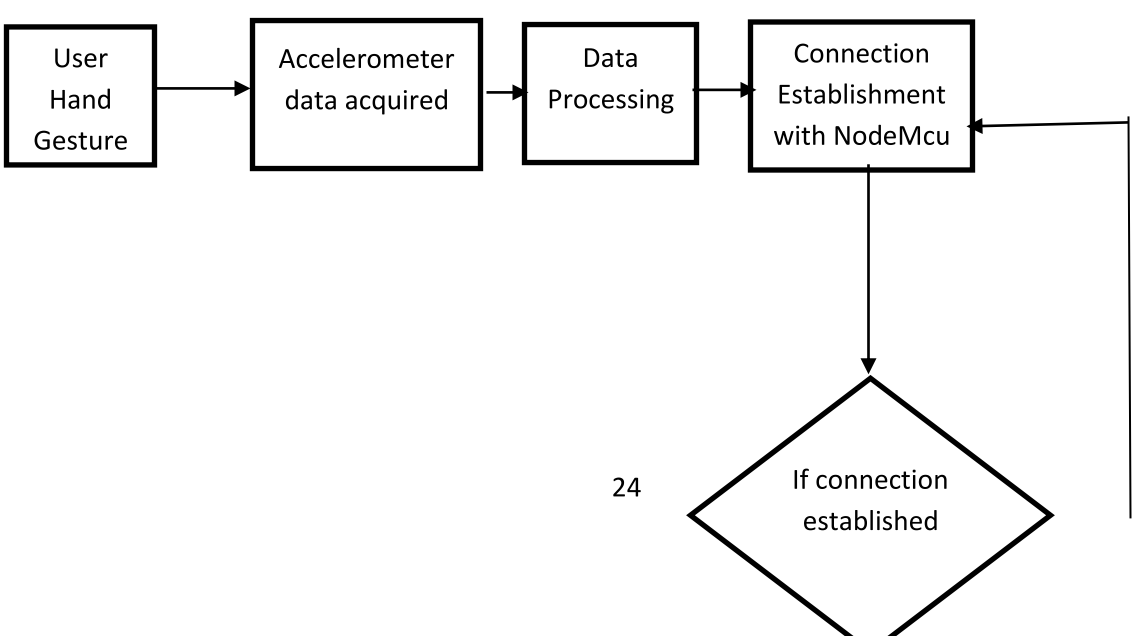
One of the most important things in the robot design is the possibility to check and control the values feeded by the analogic sensors, i.e. accelerometers, gyros, IR sensors. In order to view the sensor values ,the Serial Port is used that could be displayed in the Serial Monitor inside the Arduino IDE environment. But it isn't possible to show charts or save the data read from the sensors. There are several software that allow to show the graphs derived from the sensors data, and

Usually Processing or similar language are used that has a requirement of always to write some code.Appropriate code is written to change the charts or to implement new lines in the graphs.

The ideal situation would be to use a software that everyone knows: Microsoft Excel.
If it were possible to use Excel to receive the data from the serial port, it would be easy to make very nice and user friendly charts.

start again from the first position, in order to avoid a graph too large. The command format are: Serial.println (ROW, SET, 2) which put the cursor in the second line next step.
The software is called PLX-DAQ and it is a free software. It works only in Windows. In practice it is a little software in VBA that adds some features to Excel to receive and elaborate real time data. If the data are in Excel it is also possible to save them in a file and it is possible to use the amazing function library already present in the spreadsheet software. The documentation about PLX-DAQ is complete and clear.

The PLX-DAQ takes commands in uppercase and every data row have to end with a ‘carriage return’. The serial port speed required by PLX-DAQ is particular, because it derives from Parallax world. In any case the rate is from 9600 bit/sec to 128.000 bit/sec. The speed of 128.000 bit/sec works fine in the Arduino.



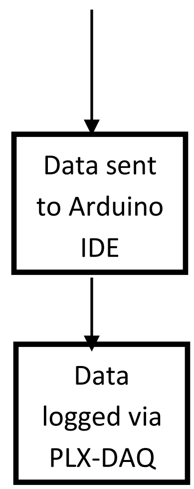


Fig 3.3 Flow chart of total connection establishment.

The main commands are:

- LABEL used to define the column headings. The command format are:

Serial.println (“LABEL, INT_COLUMN”);

- DATE, TIME that allows the serial port to send data to Excel. The first field is always TIME, then the fields of interest (val). The command format are: Serial. print (“DATE, TIME,”); Serial.println (val)

ROW, SET, k, to define the next line to write. It is useful if it is required to plot n data and then go back to first row and cycle.

Statistical Features:-

A number of statistical features were extracted namely mean, difference, standard deviation,variance,average of absolute deviation.

Formulaes for different statistical features:-

Let there be variables x1,x2,x3,…..,xn.Then,

Mean: $\bar{x} = \frac{\sum \boldsymbol{xi}}{\boldsymbol{n}}$ (1)

Difference=x1 − x2 − x3 − ⋯ xn (2)

Standard deviation: $\sigma = \sqrt{\sum(\boldsymbol{x} - \bar{x})\text{^}2/\boldsymbol{n}}$ **(3)**

Variance: $\sigma = \frac{\sum \boldsymbol{X^2}}{\boldsymbol{N}} - \mu\text{^}\mathbf{2}$

Average of absolute deviation: $\sum|\boldsymbol{xi} - \bar{x}|/\mathbf{n}$ **(4)**

Description of Statistical Features:-

Gesture	Gesture Description	Mean	Difference	Standard Deviation	Variance	Average of absolute deviation
G1	Rest1	35674.91	-21297923.00	32505.42	1056602536	32376.96
G2	Wrist up	35674.91	-21297923.00	32505.42	1056602536.31	32376.96
G3	Wrist down	140.03	-83599.00	83.68	7003.15	71.95
G4	Rest2	43748.62	-26117927.00	30761.05	946242424.47	29001.80
G5	Wrist left	2740.81	-1636262.00	12849.88	165119321.28	5048.05
G6	Wrist right	43533.85	-25989710.00	30840.24	951120140.02	29148.68

Table 1: Description of statistical features

Plot of different gestures along the three axes:

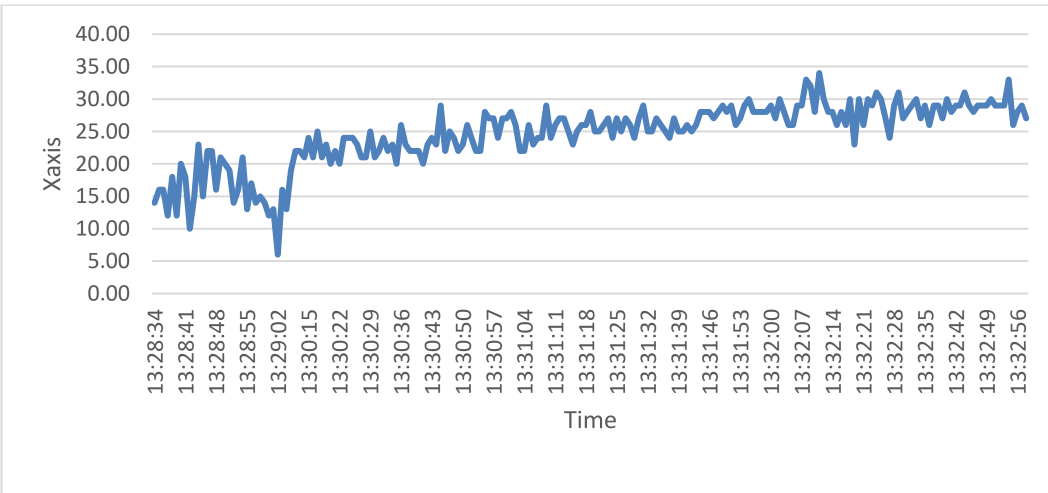


Fig 3.4: Raw data of X-axis in gesture G1:

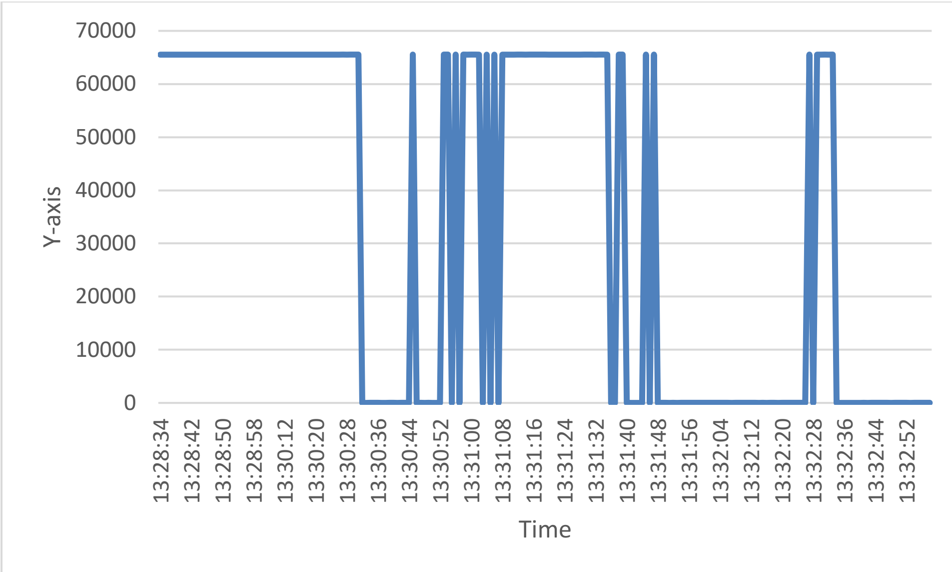


Fig 3.5 Raw data of Y-axis in gesture G1

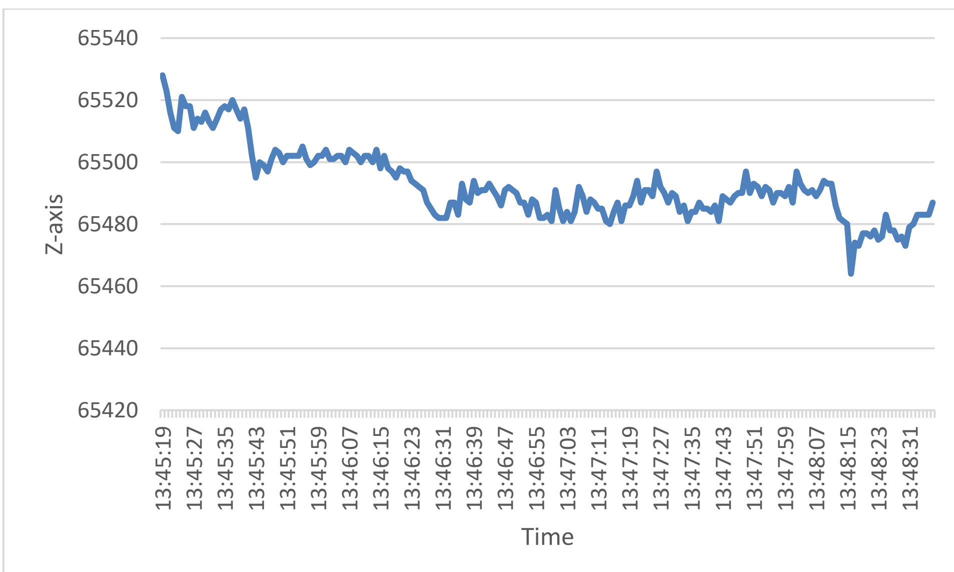


Fig 3.6 Raw data of Z-axis in gesture G1

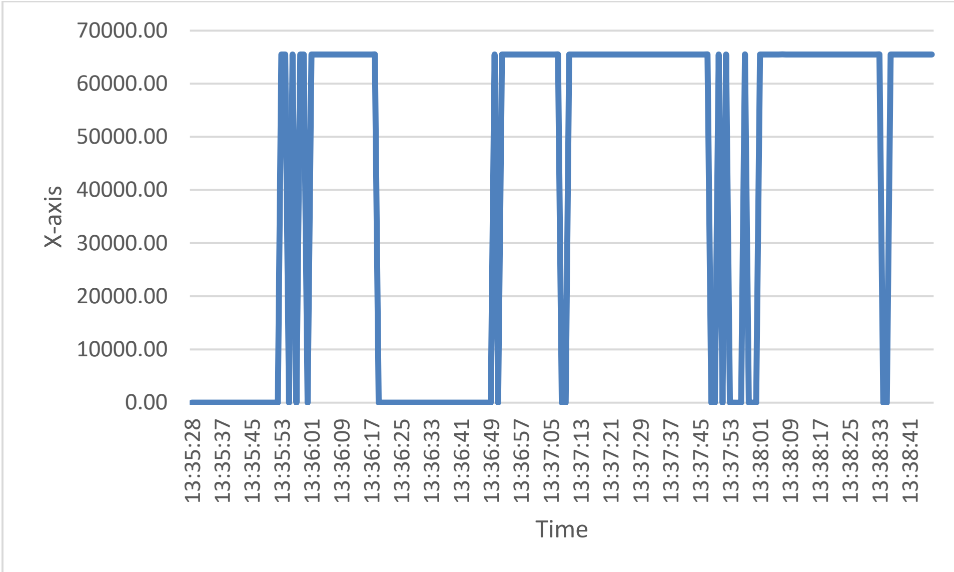


Fig 3.7 Raw data of X-axis in gesture G2

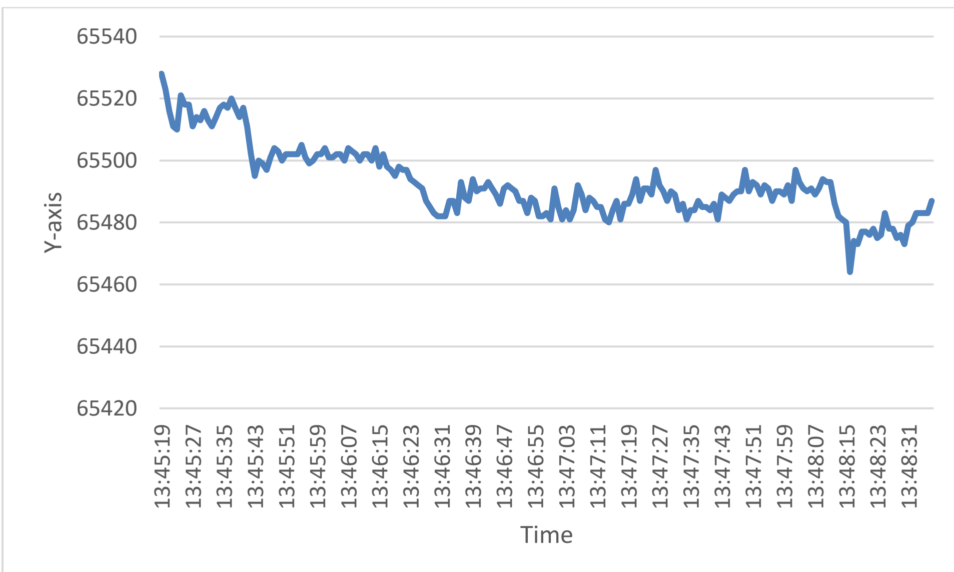


Fig 3.8 Raw data of Y-axis in gesture G2

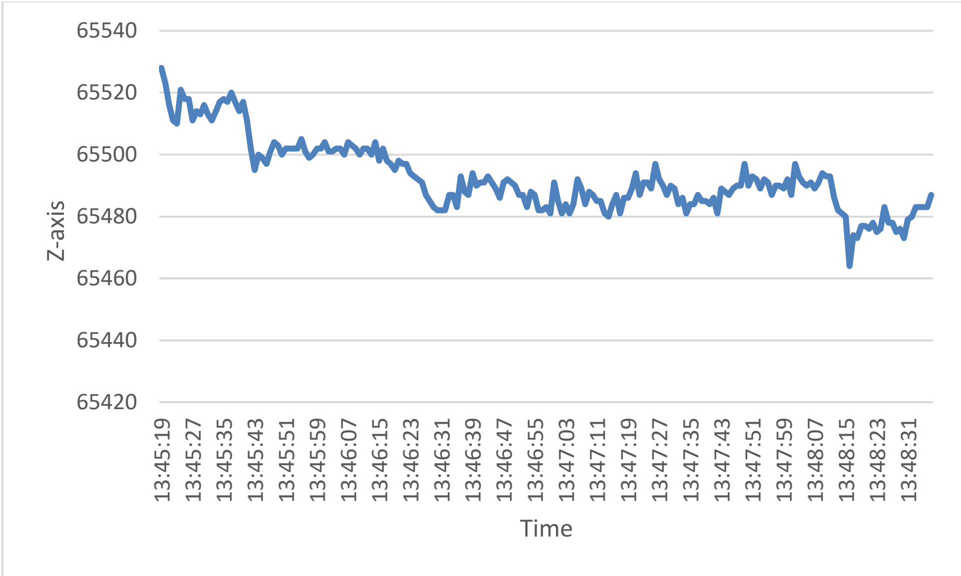


Fig 3.7 Raw data of Z-axis in gesture G2

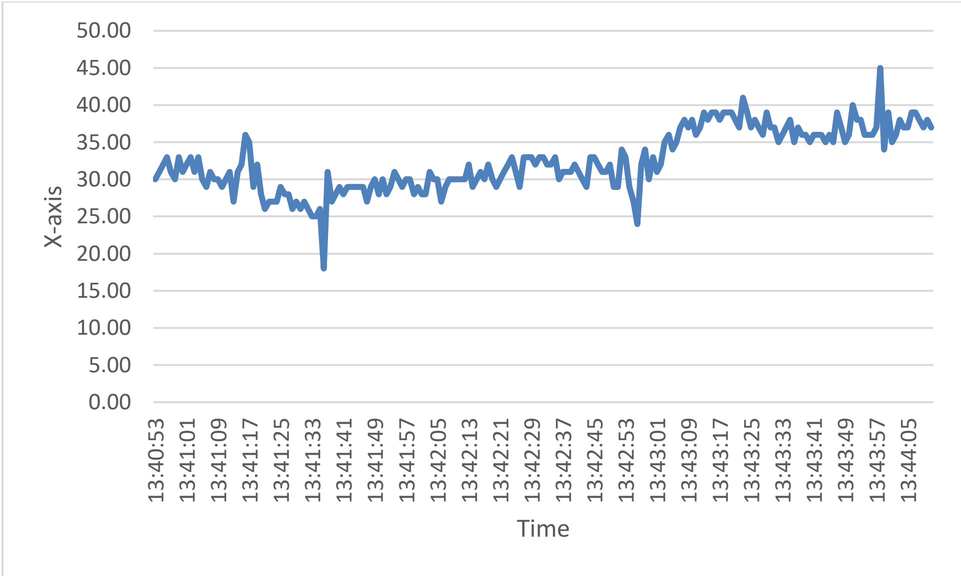


Fig 3.8 Raw data of X-axis in gesture G3

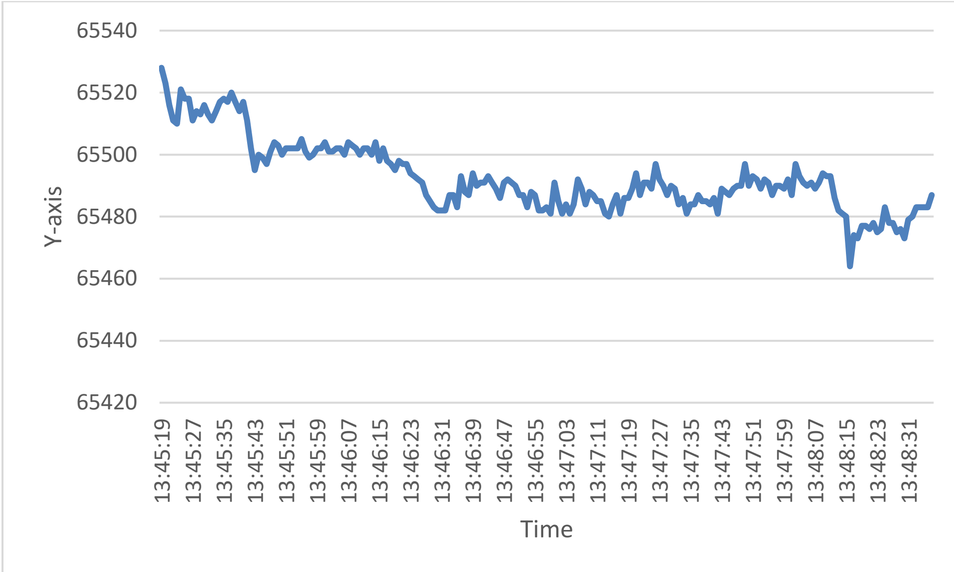


Fig 3.9 Raw data of Y-axis in gesture G3

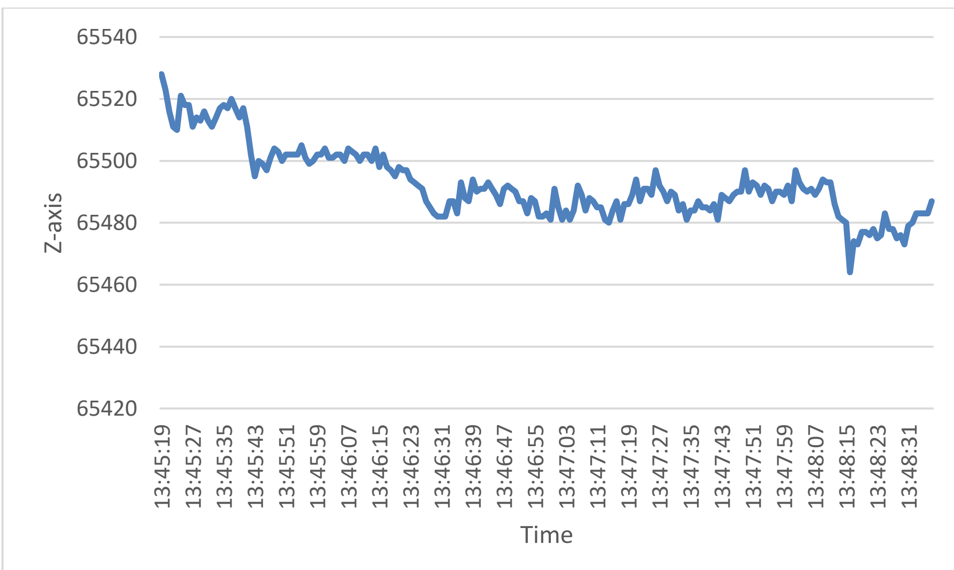


Fig 3.10 Raw data of Z-axis in gesture G3

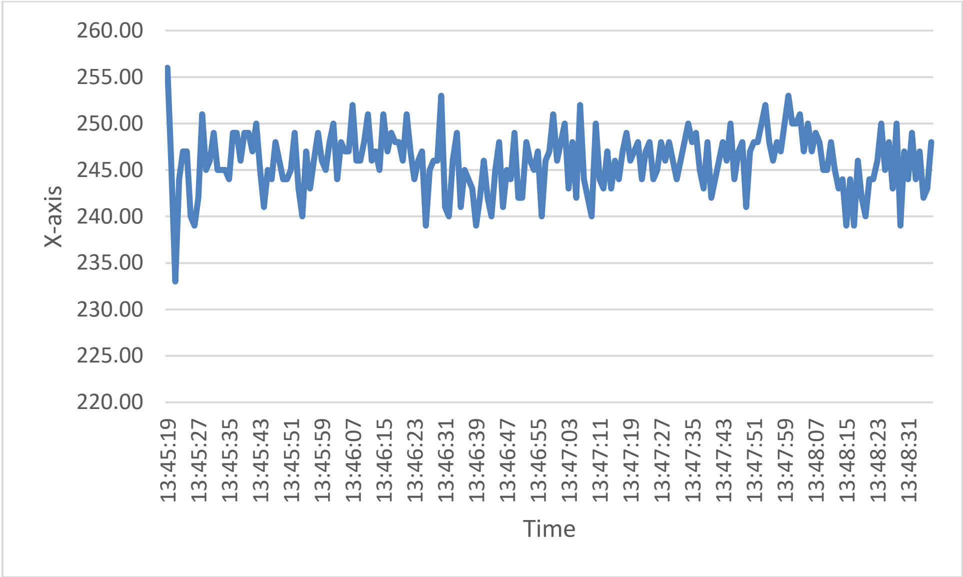


Fig 3.11 Raw data of X-axis in gesture G4

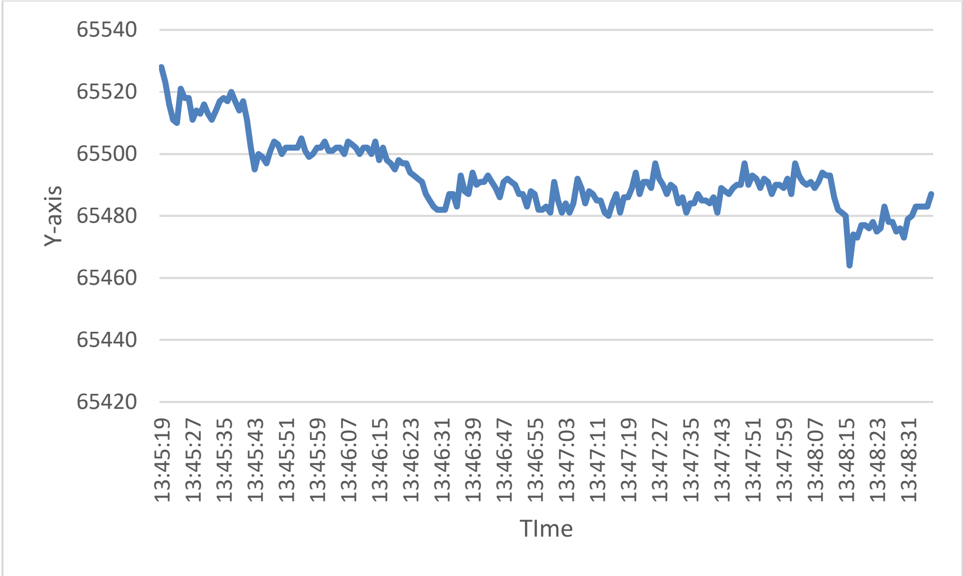


Fig 3.12 Raw data of Y-axis in gesture G4

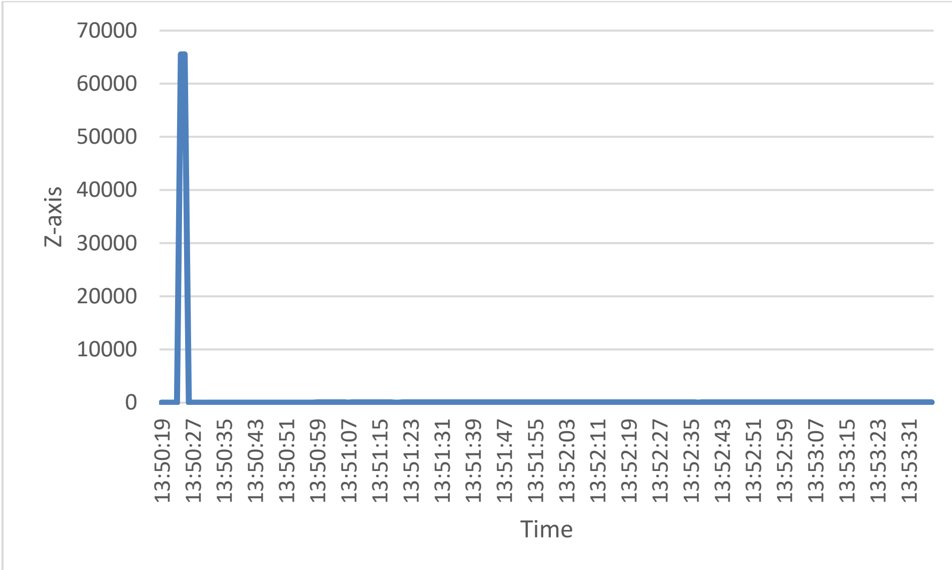


Fig 3.13 Raw data of Z-axis in gesture G4

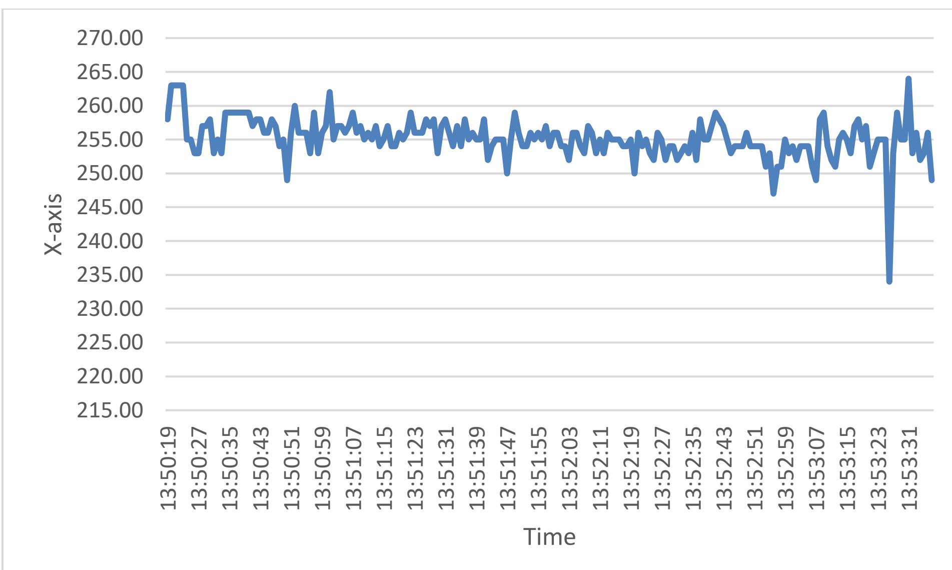
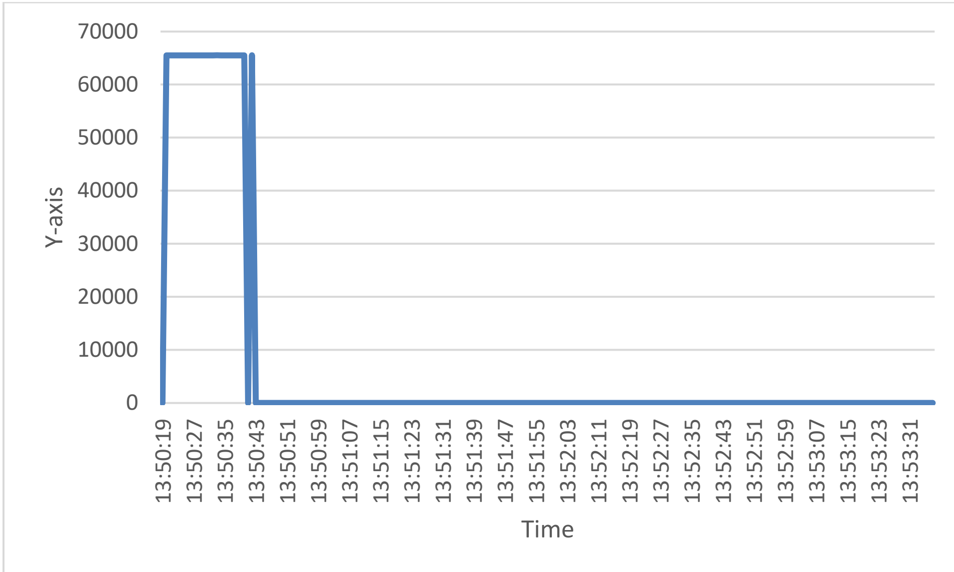


Fig 3.14 Raw data of X-axis in gesture G5



Raw data of Z-axis in gesture G5:

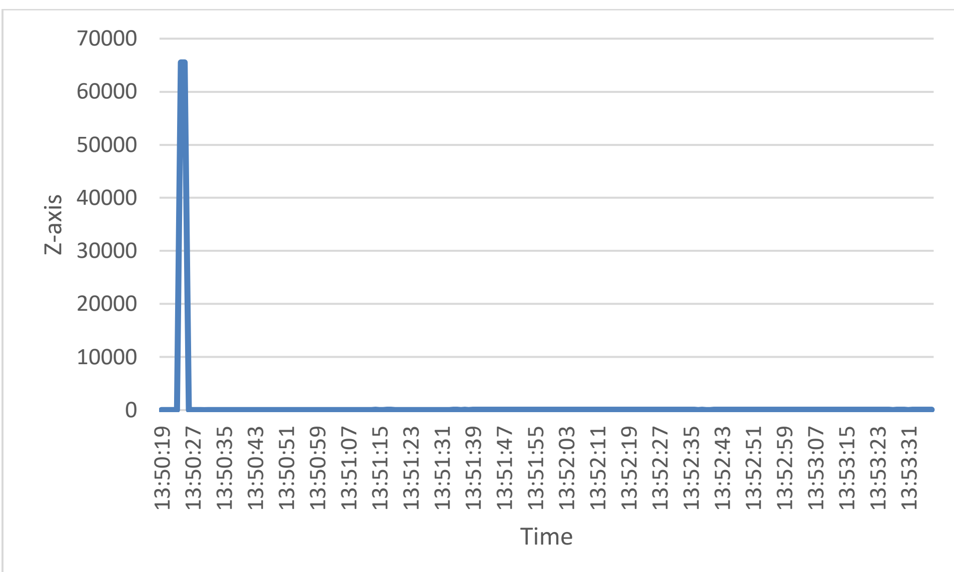


Fig 3.15 Raw data of Y-axis in gesture G5

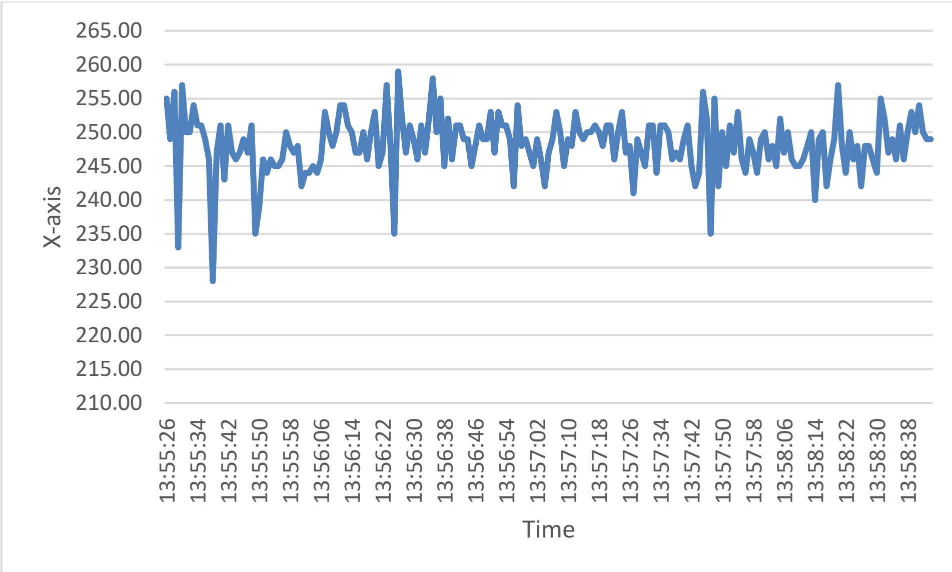


Fig 3.16 Raw data of X-axis in gesture G6

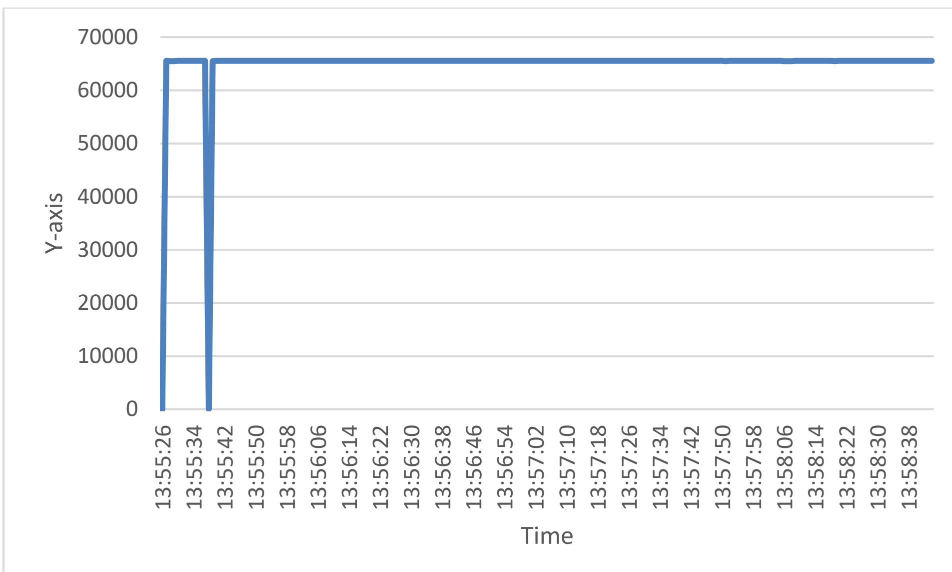


Fig 3.17 Raw data of Y-axis in gesture G6

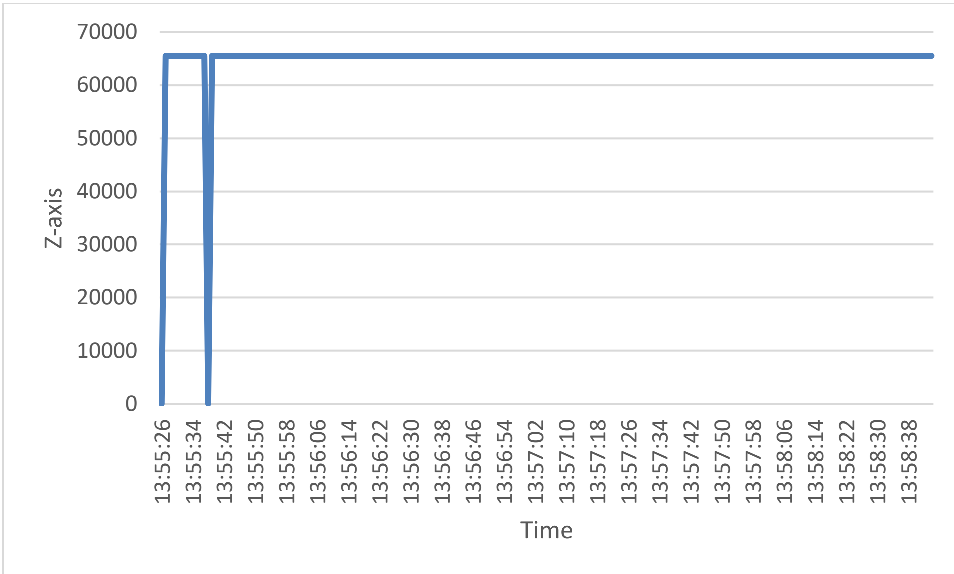


Fig 3.18 Raw data of Z-axis in gesture G6

Chapter 4: MQTT Protocol

This protocol actually acted like a bridge in our project, which has basically interfaced two controller which we had used in our project. MQTT is known as “Message Queuing Transport Telemetry system”. It is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium.

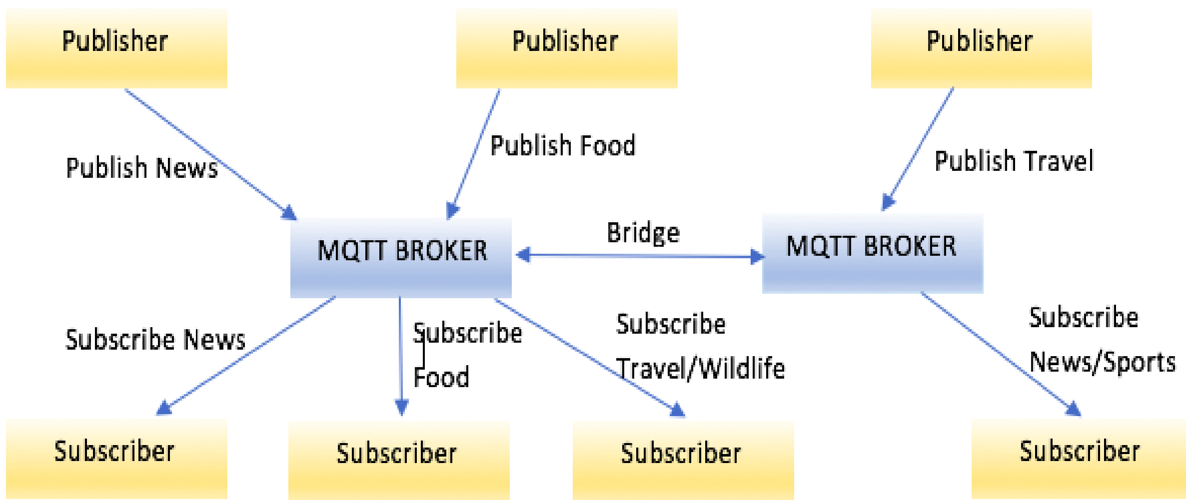


Fig 4.1.: MQTT publishing and subscribing technique

It is slide wise publish and subscribe system where we can publish and receive messages as a clients.So it becomes more easy to establish a communication between multiple devices. This messaging protocol is actually has a low bandwidth. It is therefore a standardized network protocol with which short messages / commands can be transmitted. The big advantage is that the built-in wifi adapters (for example in the Raspberry Pi or in the ESP8266) are used for the internet connection. More accessories and complicated wiring is not necessary! This makes it possible to send the data via the local network or the Internet.
In detail, the transfer consists of three different components:

- **Publisher**: Sends messages.
- **Broker**: Forwards messages to registered subscribers.
- **Subscriber**: Receives messages through the broker.

WHAT CAN BE DONE :

1. A command signal can be sent to control an output.

2. Data can be read from the sensor and publish it

BASIC CONCEPTS:

This MQTT protocol is based upon some basic concepts which represents its total activities. These concepts are like,

Publish and Subscribe

Messages

Topics.

Broker

Let’s know about those topics.

PUBLISH and SUBSCRIBE:

This concept is also based on message transferring, like when a main device is trying to publish some messages to other devices and those devices as a return subscribe those particular topics.So we can exchange messages. So what is **topic** here,

TOPIC :

Topics are the way to rise interest for incoming messages or we want to specify where we want to publish messages. Topics can be represented by strings separated by **slashes “/”.** This Slash indicates the topic level. In natural topics are case sensitive.

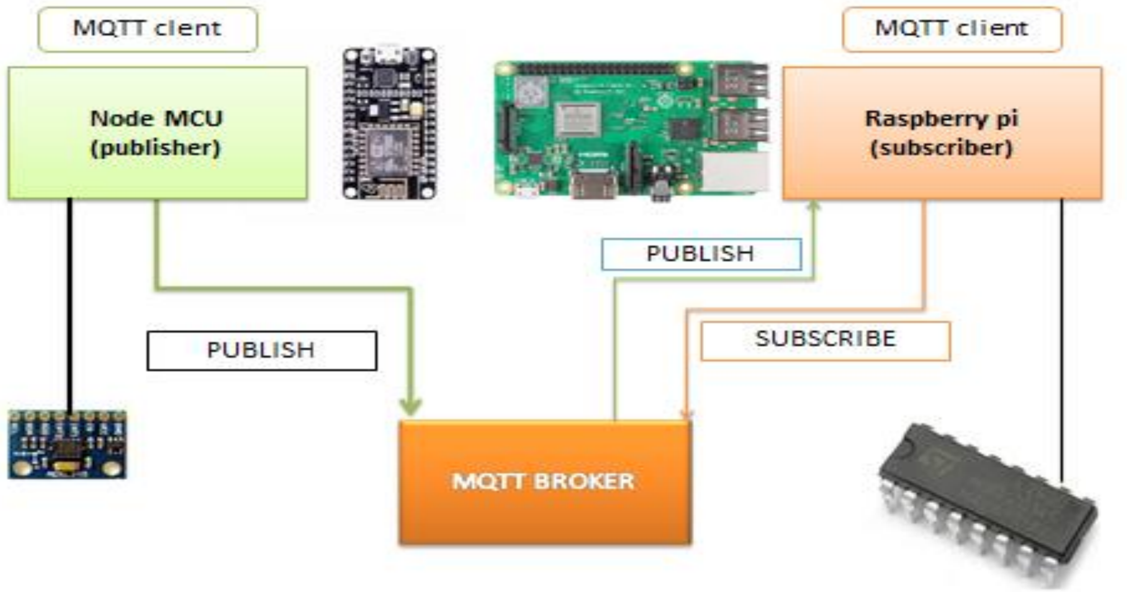


Fig. 4.2: Block diagram of MQTT setup

<u>MQTT Broker :</u>

Sometime a vender's Cloud service becomes more useless and an imperfect media to interconnect several devices like several sensors to more devices. With the flow of growing technologies the system has changed, basically IOT based interfacing is more smart way to for the interfacing applications. The broker is at the heart of any publish/subscribe protocol. Depending on the implementation, a broker can handle up to thousands of concurrently connected MQTT clients. The broker is responsible for receiving all messages, filtering the messages, determining who is subscribed to each message, and sending the message to these subscribed clients. The broker also holds the sessions of all persisted clients, including subscriptions and missed messages. Another responsibility of the broker is the authentication and authorization of clients. Usually, the broker is extensible, which facilitates custom authentication, authorization, and integration into backend systems.

Benefits Of MQTT Broker:

1. Eliminate Insecure connections and creates an easy media.
2. Easily Scales
3. Manages all client connection states perfectly and avoid complexcities.
4. Reduces the network station.

The broker is the central hub through which every message must pass. Therefore, it is important that your broker is highly scalable, integratable into backend systems, easy to monitor, and (of course) failure-resistant. HiveMQ meets these requirements by using state-of-the-art event-driven network processing, an open plugin system, and standard monitoring providers.

So basically in this project the main purpose to use MQTT (Message Queuing Transport Telemetry) protocol is to interface Node mcu ESP8266 with raspberry pi. So through internet we tried to connect several devices which I have already mentioned.So through this arrangement several devices can be able to communicate between each other. Here the interfacing is actually works through transmitting messages from publisher to broker and broker republishes the messages to the subscribers. So when the initial device publish some message to a broker then that device will be known as publisher and the message will be published at the broker, which is known as the intermediate state . When broker will republish those messages to the final elements then the final element will be known as Subscriber which we had mentioned earlier.

So implement this operation we had to perform more tasks like we had to install several libraries with the help of suitable code. We had used this code basically for remote action purpose.

So, if we consider the installation of MQTT at Node MCU at arduino IDE then from the initial stage we have to download some libraries like MQTT library and SLEEPY DOG libraries from the library manager (pictures are given bellow)

Here we had used a suitable code for MQTT server. There some specific things are mentioned, like, the library which we had installed we had mentioned. Here we had also mentioned the wifi password and also the IP address we had mentioned. And also the properties of message receive for the command of accelerometer also mentioned there.

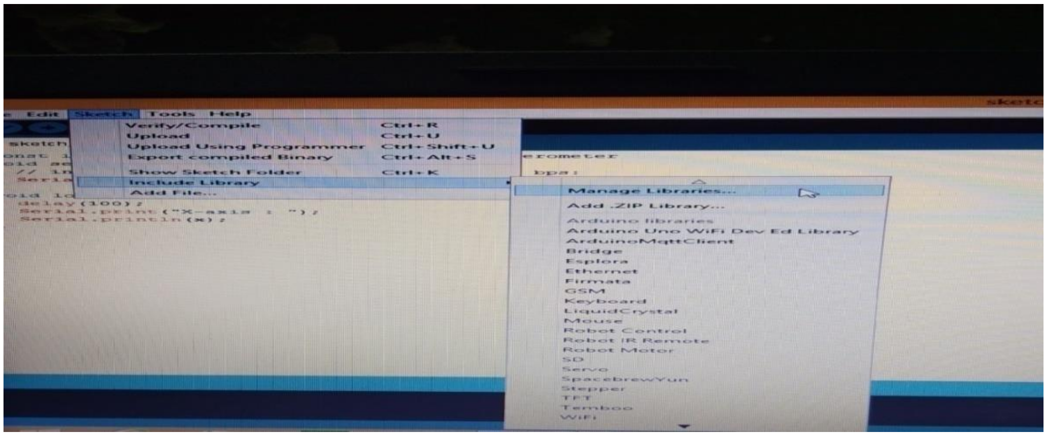


Fig 4.3: Managing MQTT libraries

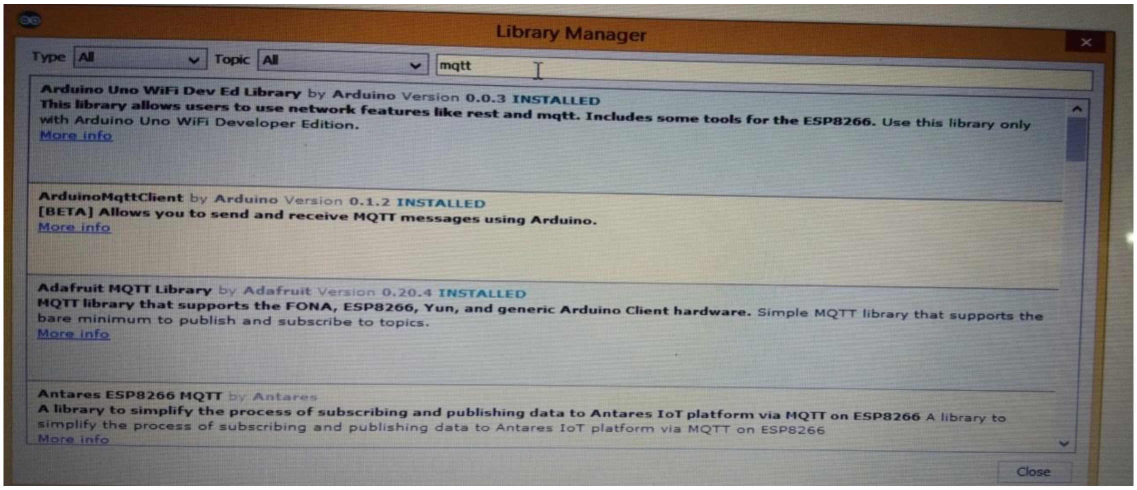
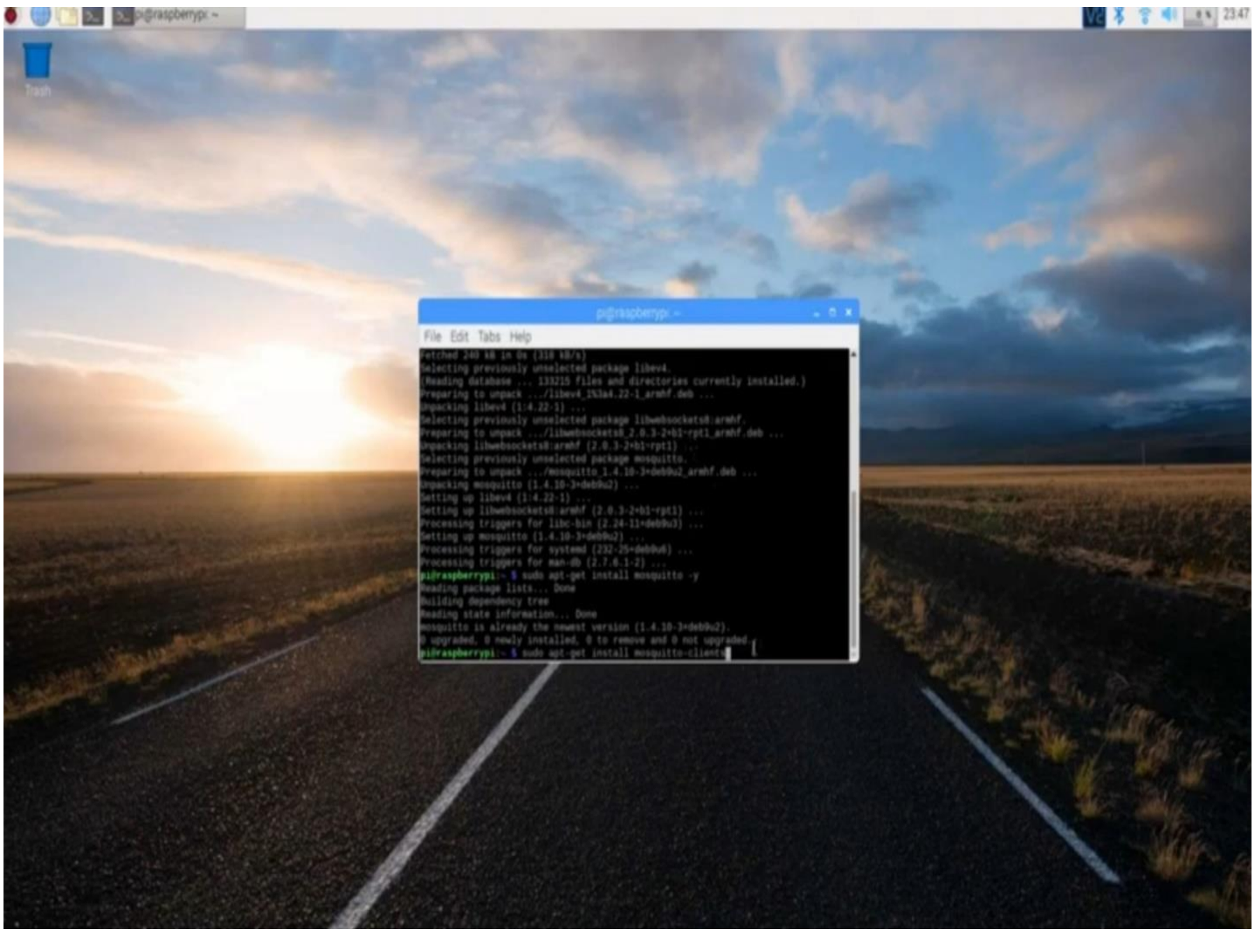


Fig 4.4: MQTT installation at Node MCU

RASPBERRY PI Installation :

The latest version of raspbian is installed for raspberry pi 3b+ and after that it starts with the command . Initially it updates the system by suitable python code. After updating the system it updateds the system repositories, because with out any updated repositories the Mosquitto server will not be installed. After that a jpj key is installed. After getting the key,it is added at MQTT repository. Then the source list is updated. After downloading the repository the system is updated again. After that there came the most essential step, step to install Mosquitto server. MQTT server is installed by suitable commands. After that they were installed few more setup of mosquito. After that there were few more steps performed to make some changes to the repository. After that the server was restarted again and it had opened a terminal again to check whether it was working or not. By typing hello command it is examined that it was perfectly



working. Most importantly it was required to login at the new terminal and it is typed at initial terminal and the response was seen in a new terminal.

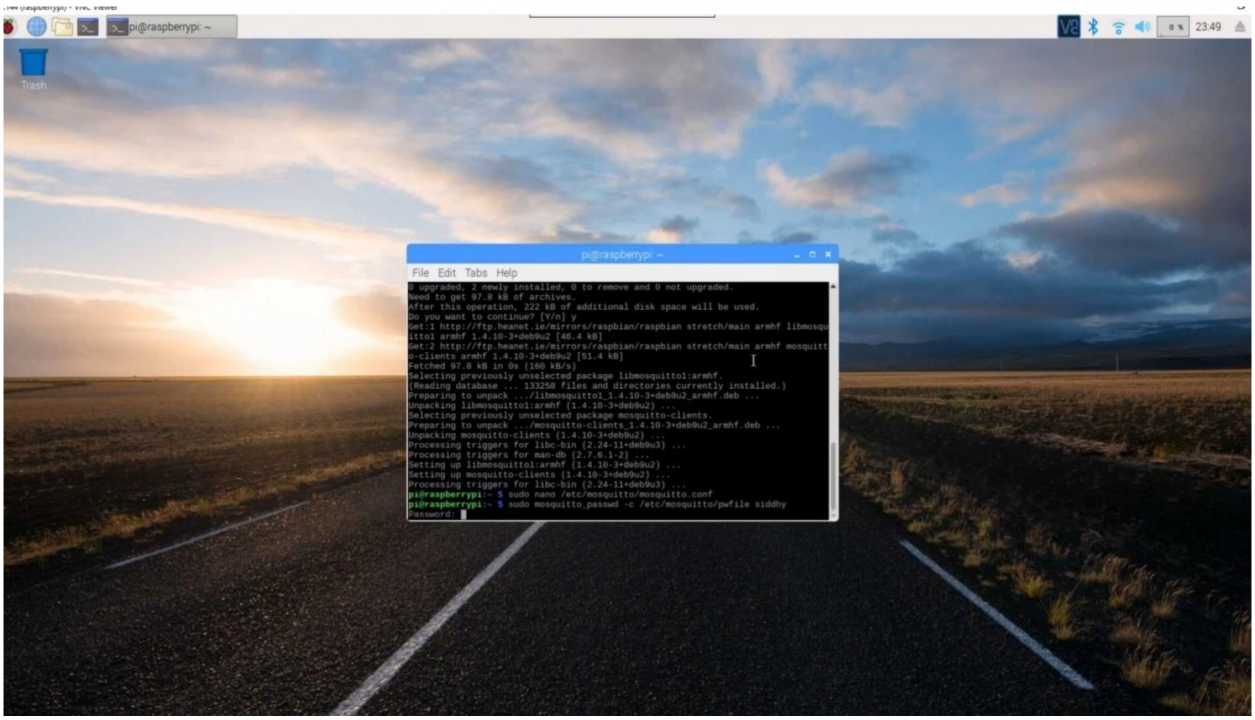


Fig 4.5 :Mqtt installation ad raspberry pi lxt terminal

Benefits and Cause to use MQTT protocol:

There are some basic cause that using MQTT protocol is very much beneficial. These are,

- Push delivery of messages by MQTT is low latency, two way communication
- Network has used efficiently in this case
- There is reliable delivery over fragile network
- Decoupling of publishers and subscriber, one to many only
- Here message broker needs, which makes a suitable medium of remote or cloud communication
- Here the cloud server is actually the broker
- It requires TCP, which is larger than UDP, which makes MQTT more easier to operate than others.
- Even though MQTT messaging uses an unsecured TCP, anyone can be able to encrypt data with TLS/SSL Internet security to make it robust, when implementing for the

mission critical business. So there is an absolute encryption based on the resourcefulness of the system and security mandate.

- MQTT defines three QoS which can cater to based on the importance of each messages and the repetitiveness of the messages in the environment.

  At Most Once – Client configured with QoS level 0 will publish messages only once. This is just a confirmation message which will not be stored or responded by the receiver Client.

  At least Once – Client configured with QoS level 1 will publish messages more than once. This message will be stored by the sender until a confirmation message is received from the other Client.

  Exactly Once – Client configured with QoS level 2 ensures publishing the message exactly once. This is the most secure level of publishing messages.

- **Last WILL** helps in knowing whether the particular client is available or not. It is not worth waiting for something that won't happen. The listeners can be put on the power saver mode with interval based wake up to check the publisher availability.

  Retained messages will help subscribers receive messages that were published some time before.

  These messages highly decouple both the publisher and the subscriber to work independent of each device.

Node MCU to pi MQTT set up :

We are going to setup MQTT server and client on Raspberry Pi using Mosquitto and paho-mqtt libraries. We will read a button and if it is pressed we will Toggle LED on Arduino side.

Whereas, on Arduino we will use Adafruit MQTT library. We will use a sensor like LDR (here we can use any other sensor),read it and publish it as soon as it changes to Raspberry Pi.

Components needed:

Raspberry Pi

Node MCU

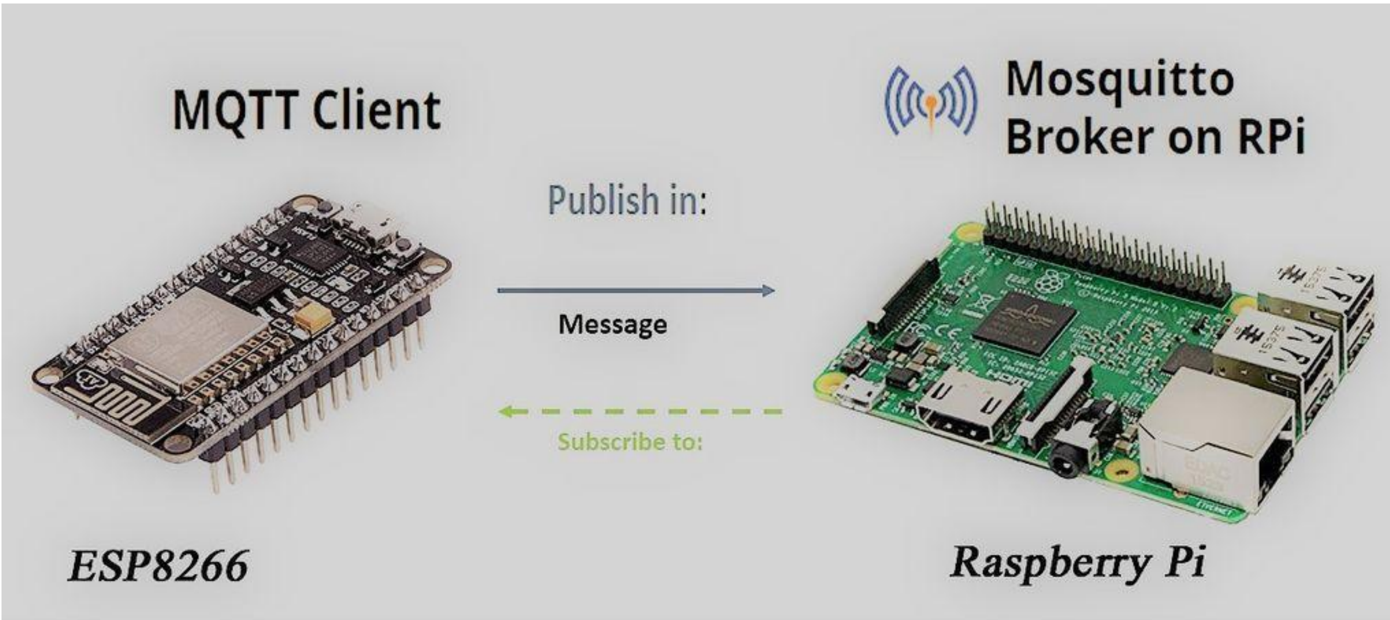


Fig 4.6Network configuration of Node MCU to pi MQTT setup

Step 1: Adding Libraries for Esp8266

First navigation should be to "Manage Libraries..." using the Arduino IDE and searching for "MQTT" by Adafruit.

After successful installation, Addition of "MQTT_ Node MCU

Step 2: Install MQTT Server and Client on Raspberry Pi

Using these commands it is possible to publish -messages from raspberry pi

Step 3: Run Python Script

MQTT_ Pi is downloaded. If everything is fine, sensor data should be seen on a terminal.

Wireless Transmission :

The another and most important part of this project is to establish a wireless transmission path between the sender junction and the receiver junction. To build that communication channel, which is actually an Ip address based communication path, is based upon several Ips , like_

1. Ip(1) = Here IP(1) is actually known as Internet protocol address of Raspberry pi 3b+.
2. Ip(2) = Here IP(2) is actually known as Internet protocol address of Node MCU.
3. Ip(3) = Here IP(3) is actually known as Internet protocol address of Mobile Hotspot.
4. Ip(4) = Here IP(4) is actually known as Internet protocol address of Computer system.

Steps for Wireless Transmission:

Step 1:

To build the communication channel here initially there have to ping from node MCU to Raspberry pi. Using mobile hotspot (IP3) the computer as well as the raspberry pi had to connect. The raspberry pi was connected via VNC viewer. At VNC viewer the IP address of the raspberry pi and the password was required.

Step 2:

After opening the raspberry pi window using VNC ,arduino ide was opened, connecting the machine with Node MCU.

Step 3:

At the time of implementing the code initially needed the raspberry pi Ip address and the password and the host name which is basically device hotspot name and the password was also used.

Step 4:

Secondly the nodemcu username and password was also used at the code.

As it is a transmitter side from this side a suitable command given through that code which had applied for mqtttransmission.

After the compilation and uploading the code at the serial monitor simultaneously shown ,

1. Connecting to MQTT server
2. Connected to MQTT server

Step 5:

At the LXT terminal of raspberry pi suitable command was given and then whatever shown in the serial monitor particularly that thing will shown in the LXT terminal. It means the connection had made and the wireless transmission going to happen.

IP Based Interface:

IP address is known as Internet Protocol version Address . The main purpose for usingIp addresses to show to differentportal of two different controllers in a one monitor .To simultaneously ping from one screen to another. So, lets know what is IP address.

Computers connected to the Internet must speak the "Internet language" called the "Internet Protocol" or simply "IP." Each computer is assigned a unique address somewhat similar to a street address or telephone number. Under the current system there are four numbers that range from 0 to 255 (Example: 206.156.18.122). Every computer, whether it functions as a web site, is being used by a web surfer, is a mail server, and/or is used for any other function, has an IP address so it can communicate across the Internet. Communication is accomplished by sending pieces of information called "packets" that include the IP address of the destination computer.

What is IP gate way ?

A default gateway serves as an access point or IP router that a networked computer uses to send information to a computer in another network or the internet. Default simply means that this gateway is used by default, unless an application specifies another gateway. The default server does not even need to be a router; it may be a computer with two network adapters, where one is connected to the local subnet and the other is connected to an outside network.

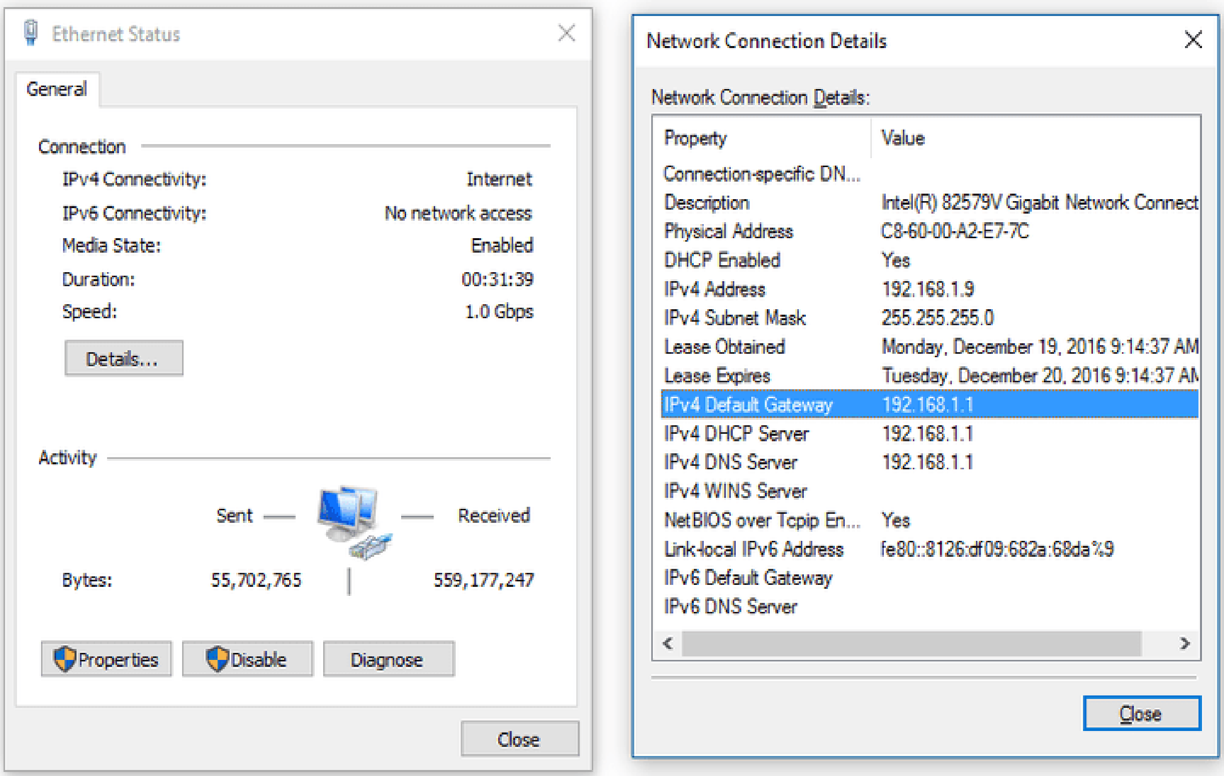


Fig 4.7: static ip setting

So now there comes the term DNS. So Domain Name Server (DNS) is an Internet service that translates *domain names* into IP addresses. Because domain names are alphabetic, they're easier to remember. The Internet however, is really based on IP addresses. Every time a user uses a domain name, therefore, a DNS service must translate the name into the corresponding IP address. For example, the domain name *www.example.com* might translate to *198.105.232.4*.

The DNS system is, in fact, its own network. If one DNS server doesn't know how to translate a particular domain name, it asks another one, and so on, until the correct IP address is returned.

So here we have considered 4 IP addresses _

1. IP1 = IP address of raspberry pi
2. IP2 = IP address of node mcu
3. IP3 = IP address of laptop
4. IP4 = IP address of mobile set

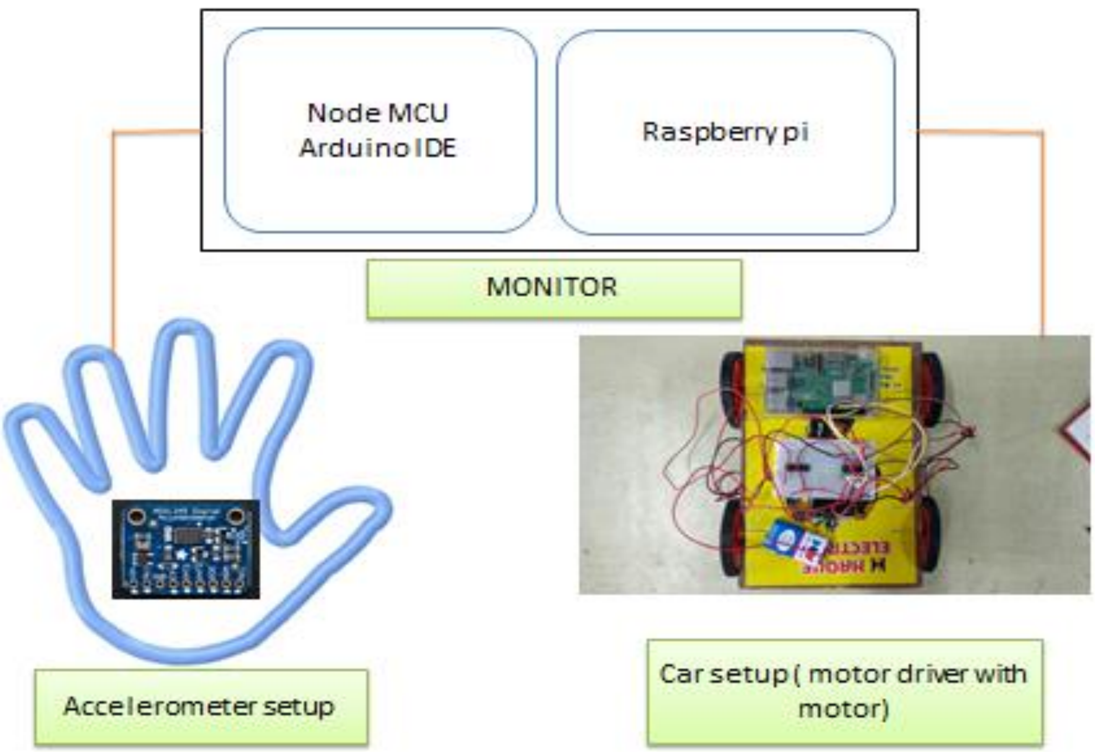


Fig 4.8: IP based wireless communication

There get the objective here there several steps to follow :

Step 1: Switch on the mobile hotspot. It has a separate IP address (which is basically ip4)

Step 2: Here raspberry pi is connected to the serial monitor (IP1).

Step 3: By giving the command IP config to the command window ip configuration can be checked.

Step 4: At the pi terminal command window find the find the IP2 by If configure command.

Step 5 : Match those and access the signals comes from the accelerometer and run the motor .

Chapter 5: Hardware Prototype

The hardware component of the system consist of raspberry pi 3 B+ with 8 GB memory card, Node mcu, Accelerometer ADXL 345, L293D motor driver, Few jumper wires, 12v dc motor, Rechargeable batteries as a power supply, Hand gloves and small size wood board.

Hardware Requirement

Requirement	Name of component/software
Hardware	ADXL345 : Accelerometer Connecting Wires
	NodeMCU
	Raspberry pi 3B+
	BreadBoard, connecting wires
	DC motor and L293D as driver
	Micro USB Cable
Software	Arduino IDE,PythonIdle 3.7

Table 2: Hardware requirement

Part A: Prototyping of Data glove

The data glove has motion sensors(accelerometer,nodemcu)on the back of the hand and on the hand movement to provide informa-tion regarding motion and orientation of the hand .Hand movement are controlling the direction of the robot in different side by sensing the sensor. Sensors mounted on the glove send signals to a processing unit, worn on the user's forearm that translates hand postures into data. An RF transceiver, also mounted on the user, transmits the encoded signals representing the hand postures and dynamic gestures to the robot via RF link.Communications, conveying intentions, distributed environment,gestures, human-computer interactions, human-robot interactions.

Circuit Connection:(Node MCU with Accelerometer)

The Accelerometer module has 8 pins

Node MCU	Accelerometer
GND	GND
3V	VCC
3V	CS
GND	SDO
D1	SDA
D2	SDL

Table 3 :Node MCU with Accelerometer interfacing

Note : Since NodeMCU has only one Analog Pin, Before we get started with coding we need Arduino IDE.To download Arduino IDE and for NodeMCU set.

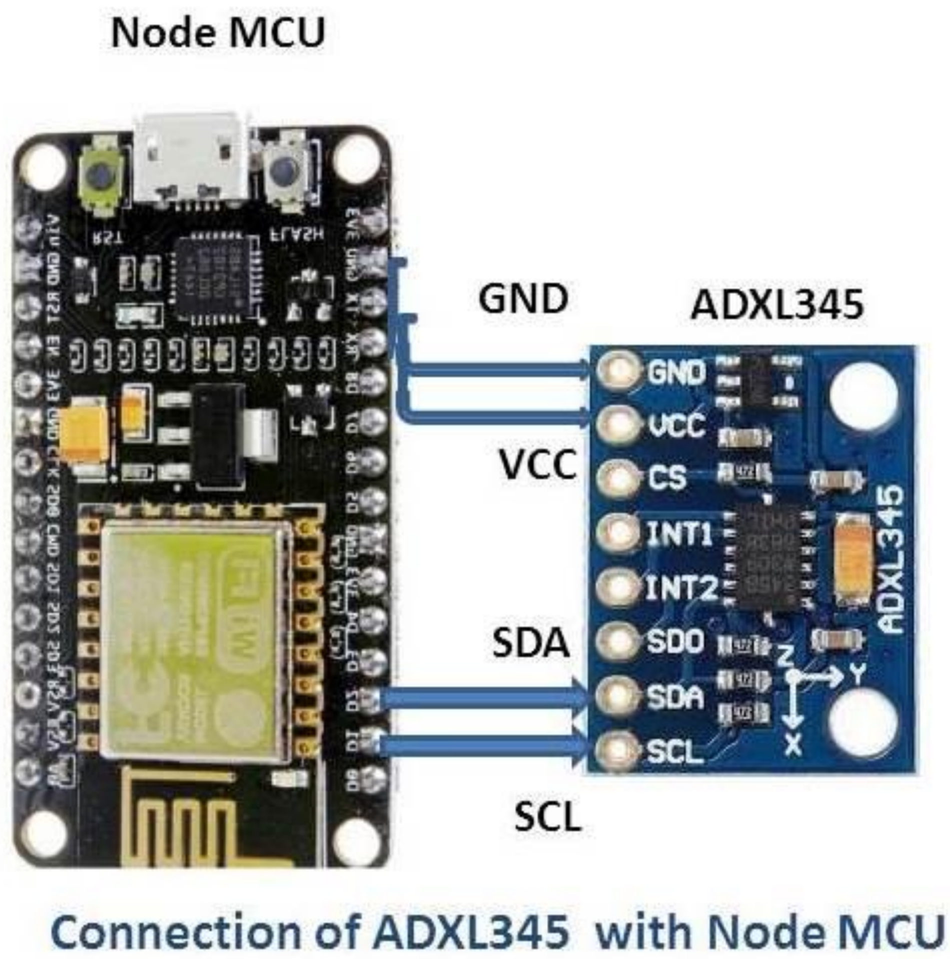


Fig 5.1: Node MCU connection with ADXL 345

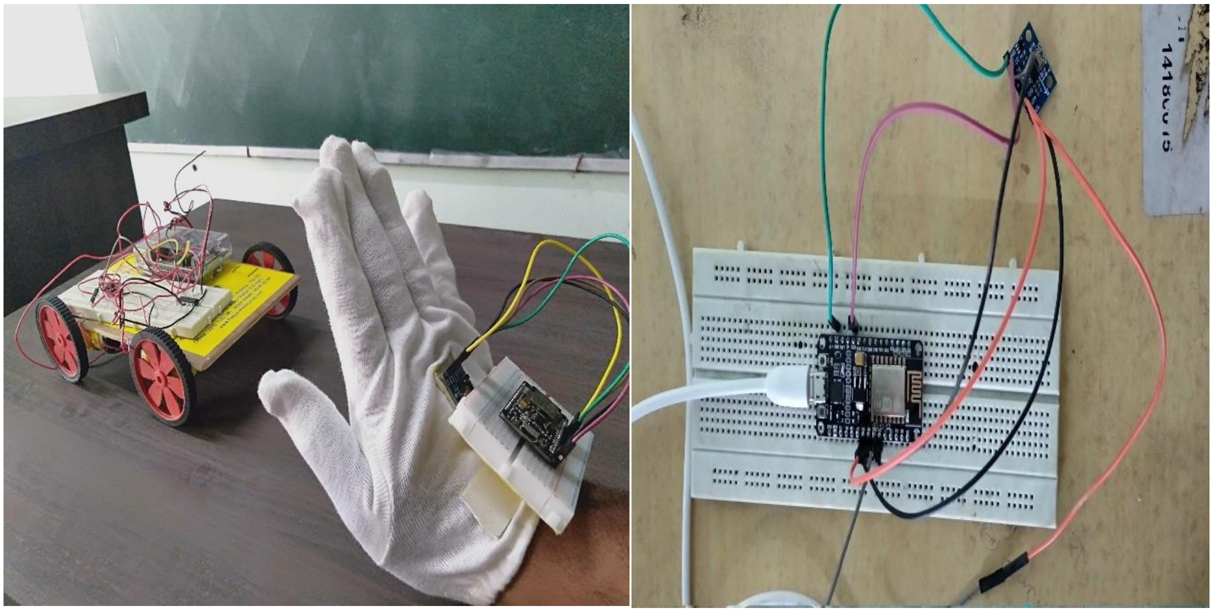


Fig 5.2 Circuit connection of Nodemcu with Raspberry pi

Setup of Network configuration:

<u>Node MCU to pi MQTT set up :</u>

The MQTT server and client on Raspberry Pi using Mosquitto and paho-mqtt libraries are set up by us. Button will read by us and if it is pressed it will Toggle LED on Arduino side.

Whereas, on Arduino Adafruit MQTT library will be used. A sensor will used like LDR (here we can use any other sensor),read it and publish it as soon as it changes to Raspberry Pi.

We need:

Raspberry Pi

NodeMCU

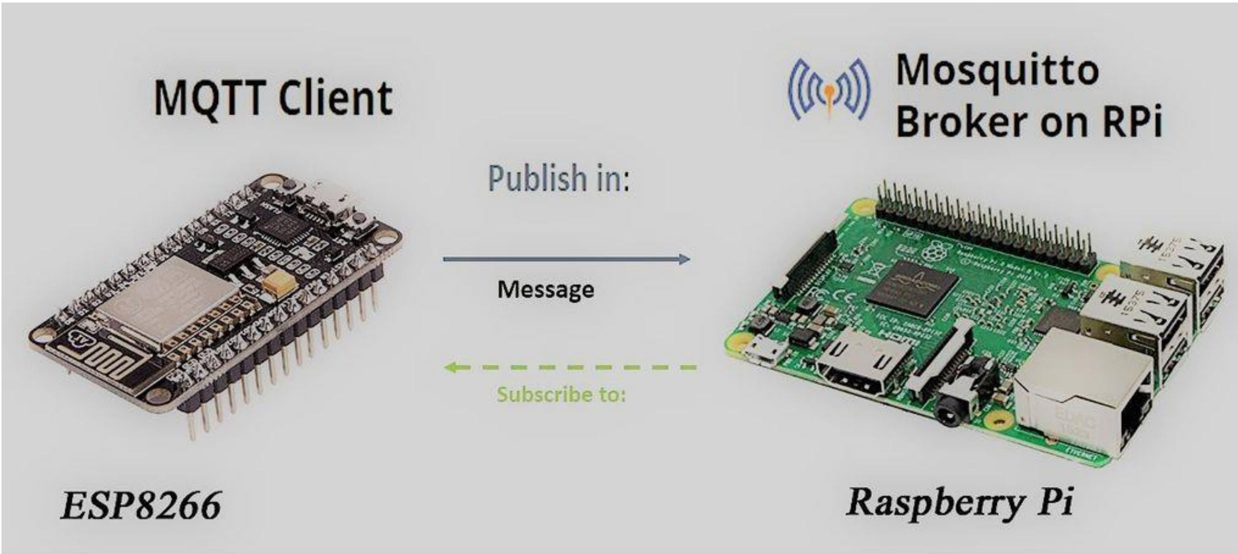


Fig 5.3 Network configuration of Node MCU to pi MQTT setup

Step 1: Adding Libraries for Esp8266

First navigation should be to "Manage Libraries..." using the Arduino IDE and searching for "MQTT" by Adafruit.

After successful installation, Addition of "MQTT_ Node MCU

Step 2: Install MQTT Server and Client on Raspberry Pi

Using these commands it is possible to publish -messages from raspberry pi

Algorithm flow chart:

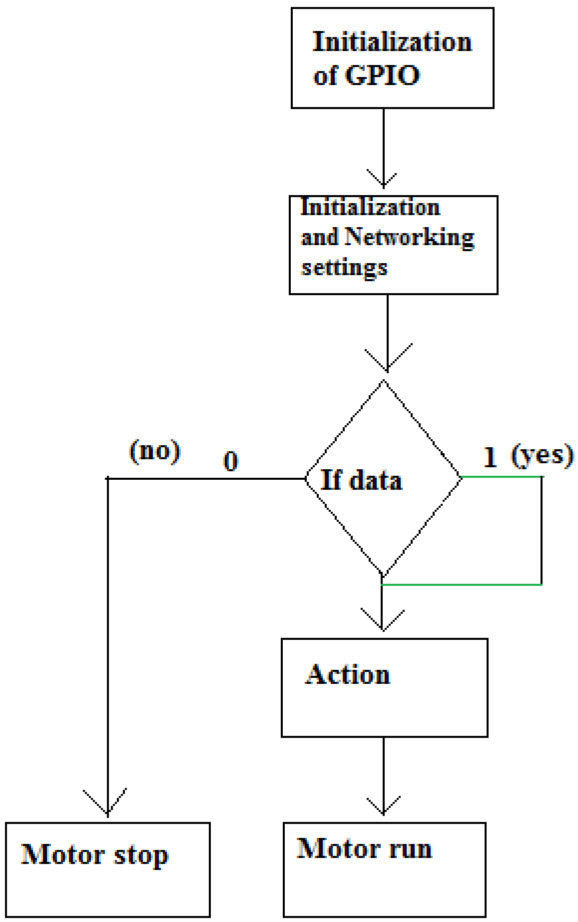


Fig 5.4 Algorithm flow chart of Receiver end

Part C:Prototyping of cart

So here we have considered 4 IP addresses –

IP ADDRESS NUMBER	COMPONENT
IP1	Raspberry pi 3B+
IP2	Node MCU
IP3	Laptop
IP4	Mobile set

Table3: Table of assigned IP addresses

Set up of raspberry pi 3 B

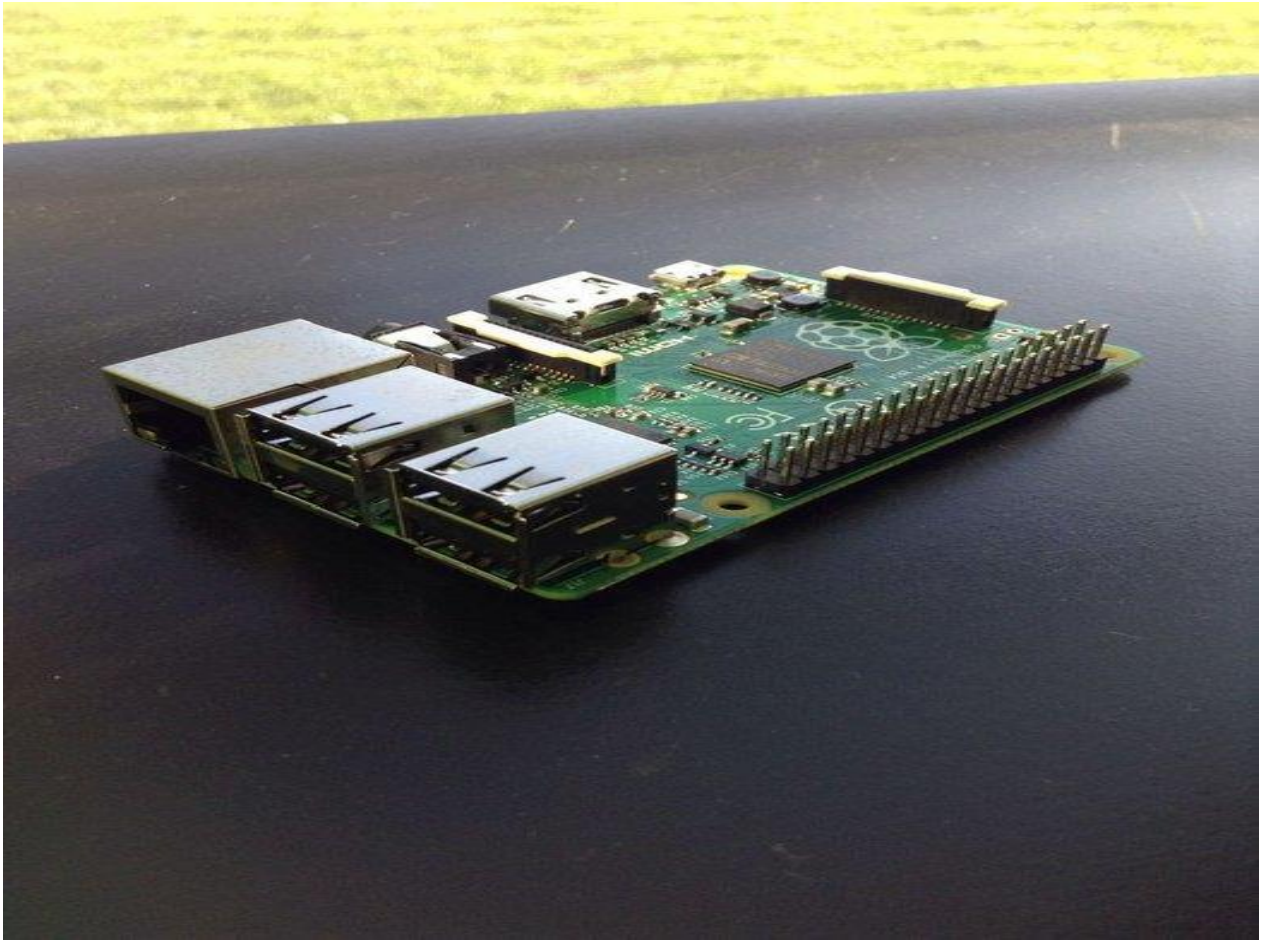


Fig 5.5 Raspberry Pi 3B Model

The Raspberry Pi Model B+ is a credit card sized computer with a Micro SD Card as a hard drive and 512MB of RAM. we install a Linux based software onto the SD Card and insert it to the bottom of the Pi. There are GPIO ports on the right hand side of the Pi that can be hooked up to LEDs, motors, sensors, etc. we can also program the Pi to do anything we want.

Rasberry pi setup:

Step 1: Gathering Parts

Fig 5.6: Gathered components

Firstly getting our Pi, it doesn't come with anything. Here's a list of parts needed:

•Raspberry Pi Model B+

•USB to Micro USB cable

•Micro SD Card

•USB Wall Charger (recommended 5V 1A US)

•Keyboard & Mouse

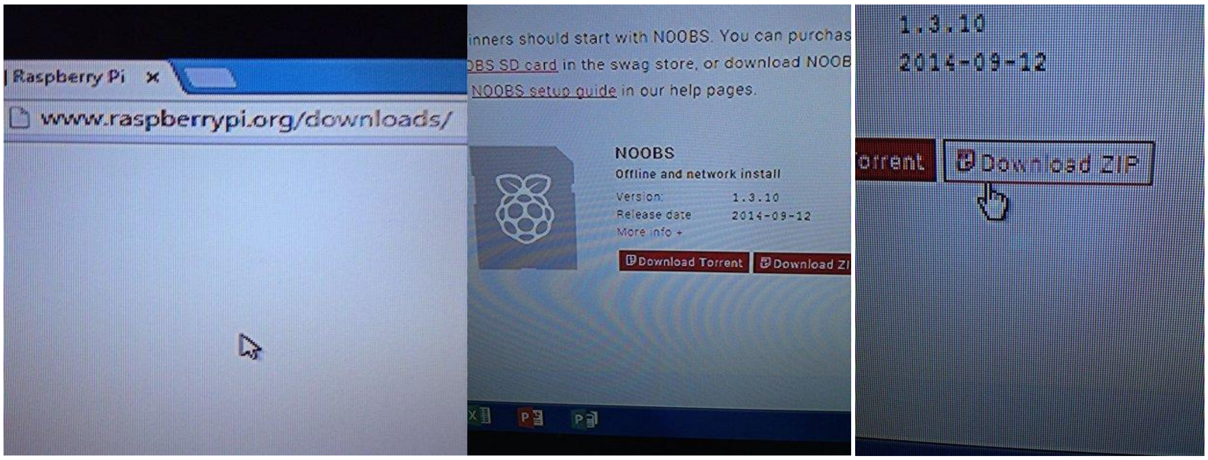
•Ethernet Cable or USB Wi-Fi dongle

•TV or Monitor

•HDMI Cable

•Computer with an SD Card reader

Step 2: Downloading Software



Here will need SD Card, the Raspberry Pi, and a computer with an SD reader. On our computer, go to http://raspberrypi.org/downloads and install the .zip file of NOOBS. This one is recommended, other than NOOBS Lite. It might take a while to download, depending on internet speed. After it's done downloading, insert the SD card into our computer. It should pop up a window of the card. Unzip the NOOBS file and copy all the items IN THE FOLDER and paste them into the SD Card. DO NOT COPY JUST THE FOLDER AND PASTE IT INTO THE SD CARD. After it's done copying, click Safely Remove Device and remove the SD Card. Put the SD Card in the back bottom place of our Pi.

Step 3: Setting Up the Pi

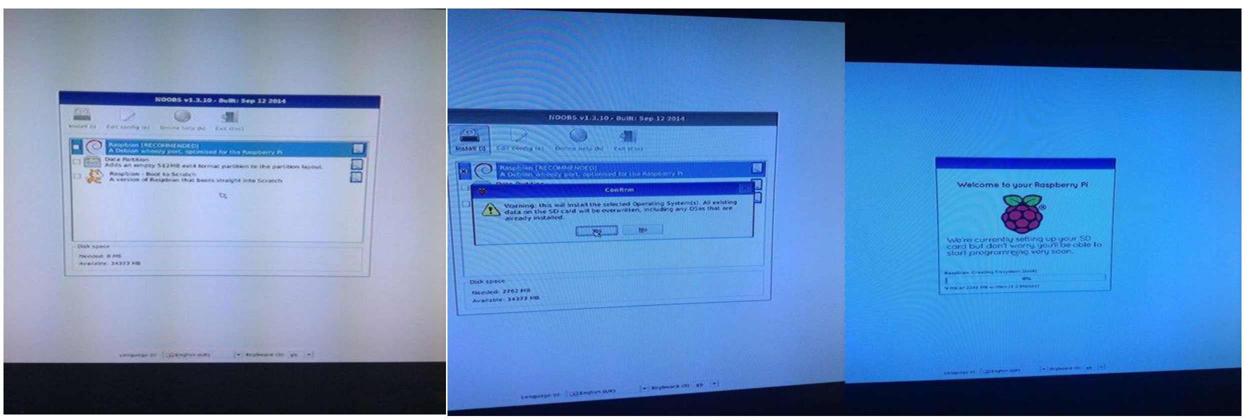


Fig 5.8: Setting up pi.

Once have put the SD card into the Pi, plug in the HDMI, keyboard, mouse, Ethernet or USB Wi-Fi dongle, and lastly the power.

On the screen there should be a box to install which software we want on our Pi..Another box will come up white a loading bar. This will usually take a while, mine took 20-25 minutes. After that, yet another box will come up. This one we are forced to only use the arrow keys on the keyboard. It has some things we might have questions about. If we think one of them is interesting, go ahead and press enter on one. If we don't want to do anything, and just get to the main screen, go all the way down to the button that says, Done. When we push the button, there should be some code that pops up at the bottom. Once it has raspberrypi@ ~ $, type in startx.

That will prompt us to the main screen. When we first power down ourPi, and then power it back up, we will have some code again. This time when it stops at Username, type in pi. Then when it says Password, type in raspberry. When type in the password, it won't come up on the screen. When done typing in all the stuff, press Enter on the keyboard.

Step 4: setting up the cart

The robot have been created by the one wood board and four D.C motors are tide by the iron wire below the board and four motors are connected with four wheels. The upper part of the frame is contain raspberry pi 3 B+,L293d motor driver both device are connected and the motor driver connected with motors. Using battery to generated the robot.

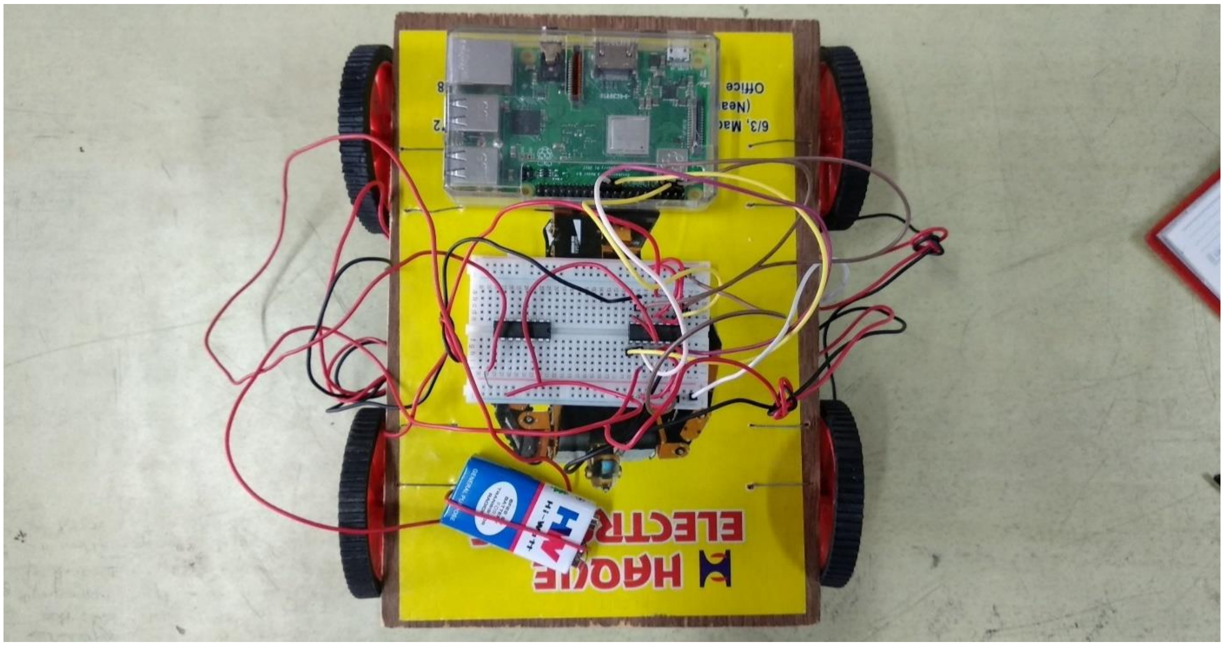


Fig 5.9: Robot formation using raspberry pi 3 B+ and L293d motor driver

Circuit diagram of motor with pi :

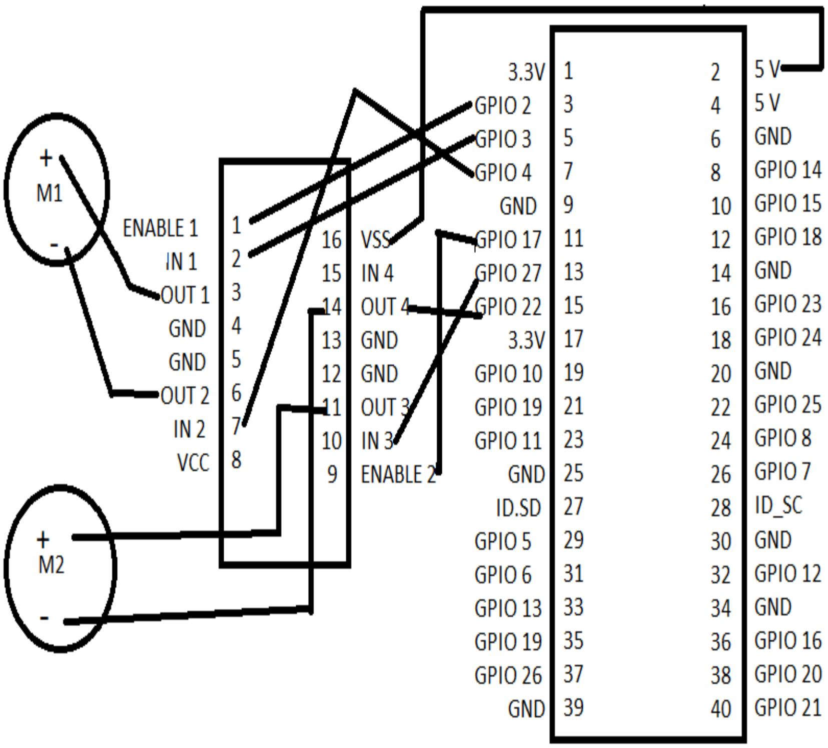


Fig 5.10 Circuit connection diagram of motor with Rasberry Pi

L293d motor driver connection:

Here in gesture control robot the motor driver connected with the raspberry pi 3B+ and D.C motor. The motor driver four output pins are connected with two motor and two enable pins and four input pins are connected with the raspberry pi GPIO pins. Four ground pins are connected with bread board ground and the VSS pin are connected with the voltage pin of raspberry pi 3.

<u>Motors Movement For The Different Direction Of Robot:</u>

Gesture	Handshape	Action of motor

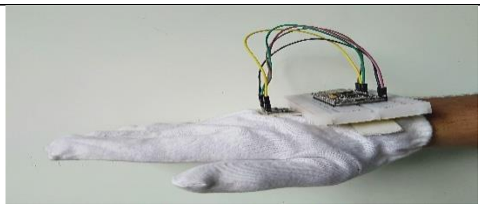
		M1	M2	M3	M4
G1(Moves in forward direction)		High	High	High	High
G2(Moves in backward direction)		High	High	High	High
G3(Movement stop)		Low	Low	Low	Low
G4(Moves in right direction)		Low	High	Low	High
G5(Moves in left direction)		High	Low	High	Low
G6(Movement stop)		Low	Low	Low	Low

Table 4 : Table of several Hand Gestures

<u>DESCRIPTION OF MOTORS :</u>

- Robot moves in forward direction - run all four motor in forward direction

- Robot moves in backward direction - run all four motor In backward direction

- Robot moves in right direction – run motor 1 and 3 in forward direction and motor 2 and 4 in backward direction.

- Robot moves in left direction – run motor 2 and 4 in forward direction and motor 1 and 3 in backward direction.

- Robot movement is stop – all four motor is stop

Chapter 6 : Experimental Result and conclusion

As it has followed the above mentioned aspects like methodology, features extraction these all gives a clear idea about the flow of this particular project work. From the Hardware prototype , the detail description of the several hardware setup.The accelerometer was set at the upper part of our palm, because the upper part of the palm is a very suitable position and that part is very nearer to our wrist. So when the wrist had moved to left, right, up and down direction accordingly the accelerometer will move. The accelerometer had set on a hand-glub at the right hand. According to the direction wise movement the accelerometer will give some data according to the gesture direction which will be represented in a graphical manor.After locking the raw data of our hand gestures by ADXL345 accelerometer, the data was prepared for wireless transmission. For that there have to encode that data. For this step there Node MCU (ESP 8266) was used. With the help of suitable codes for the three axis data of accelerometer and here as a software Arduino Ide used for this purpose. After there was added the mosquitto repository to apt-get and then further downloaded the repository list file for MQTT. After that at the MQTT installation part there was updated the apt-get source list and after that there was installed the package of MQTT broker. As this project is quite python based paho-mqtt was installed to replace the old version of mosquito. After that MQTT had configured and after making set all these things it was tested the whole set up by using two command window by saying “Hello pi” which was visualized in both command window.

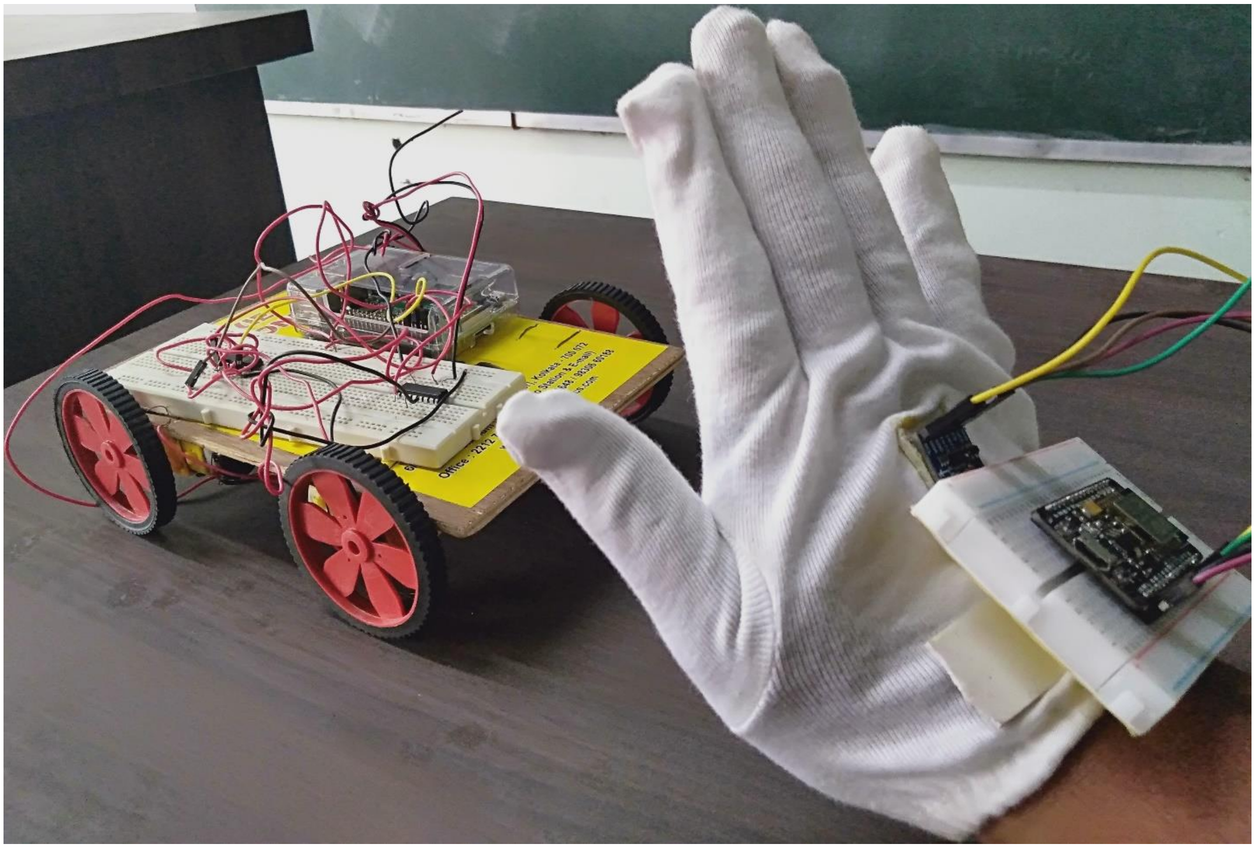


Fig 6.1: direction wise hand movement

After that steps there comes the matter of arduino ide installation. There were initially installed several libraries from Arduino library and after that there's to ran some codes and in this way the interfacing part was done . After that there comes the final most steps. After achieving the command of accelerometer by a prolonged process by Raspberry pi there was a suitable circuital

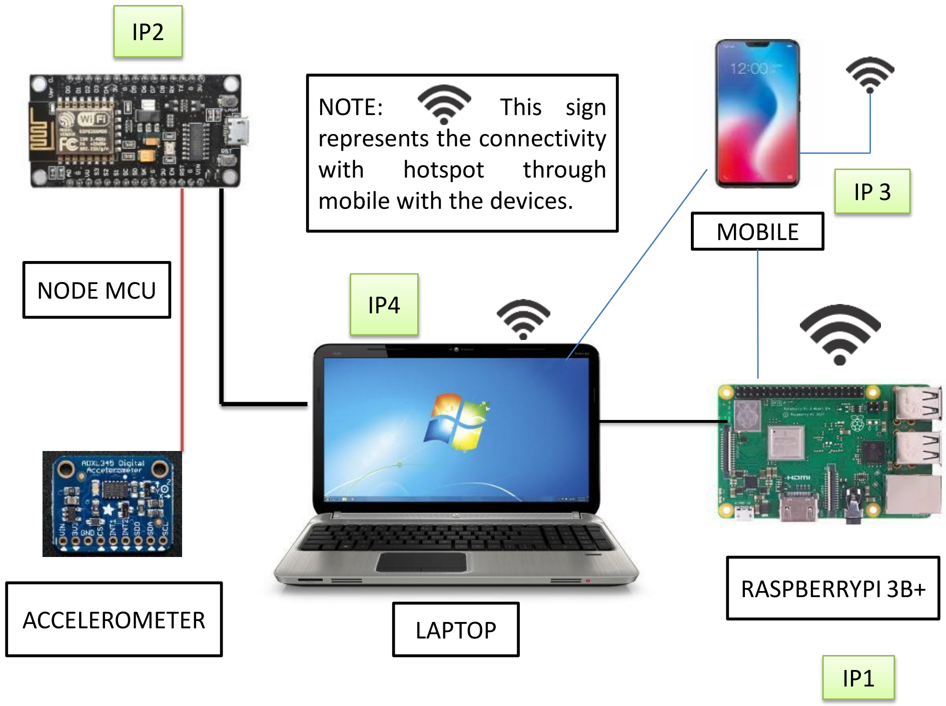


Fig 6.2: Wireless network set up of the proposed scheme

connection made among the motor driver had used and the motors which are connected with wheels and most importantly with the pi board which is the controller . After that was shown according to the commends means according to the movements of the human palm, which were previously assigned the car like setup moved.

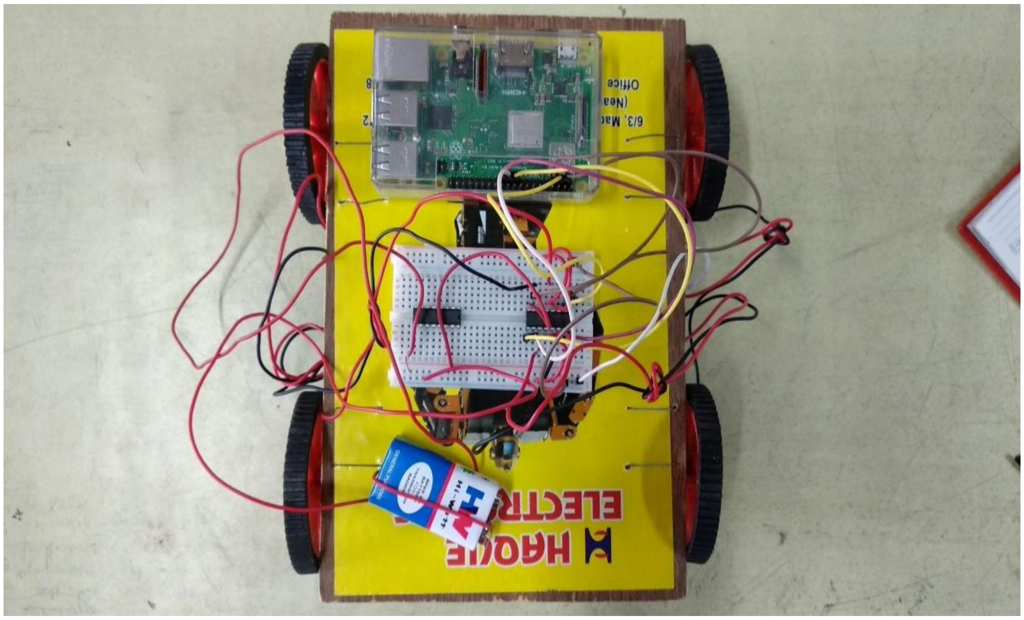


Fig 6.3: total setup

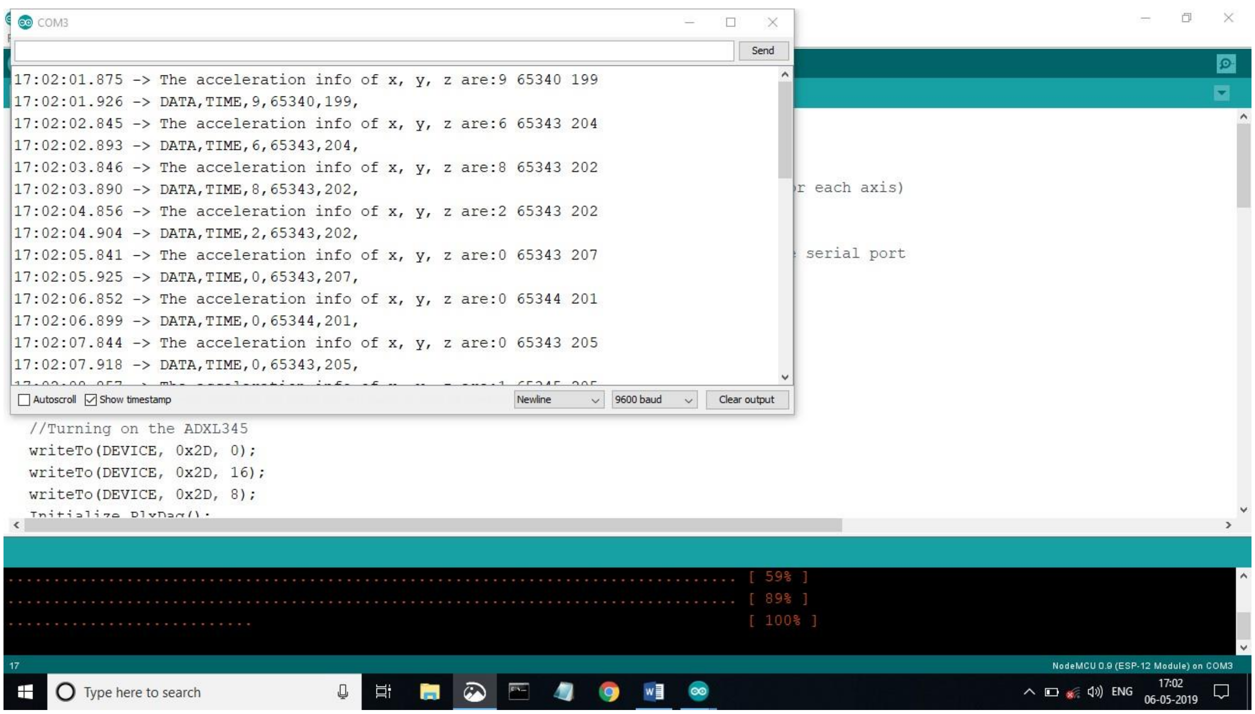


Fig 6.4 Accelerometer reading on serial plotter

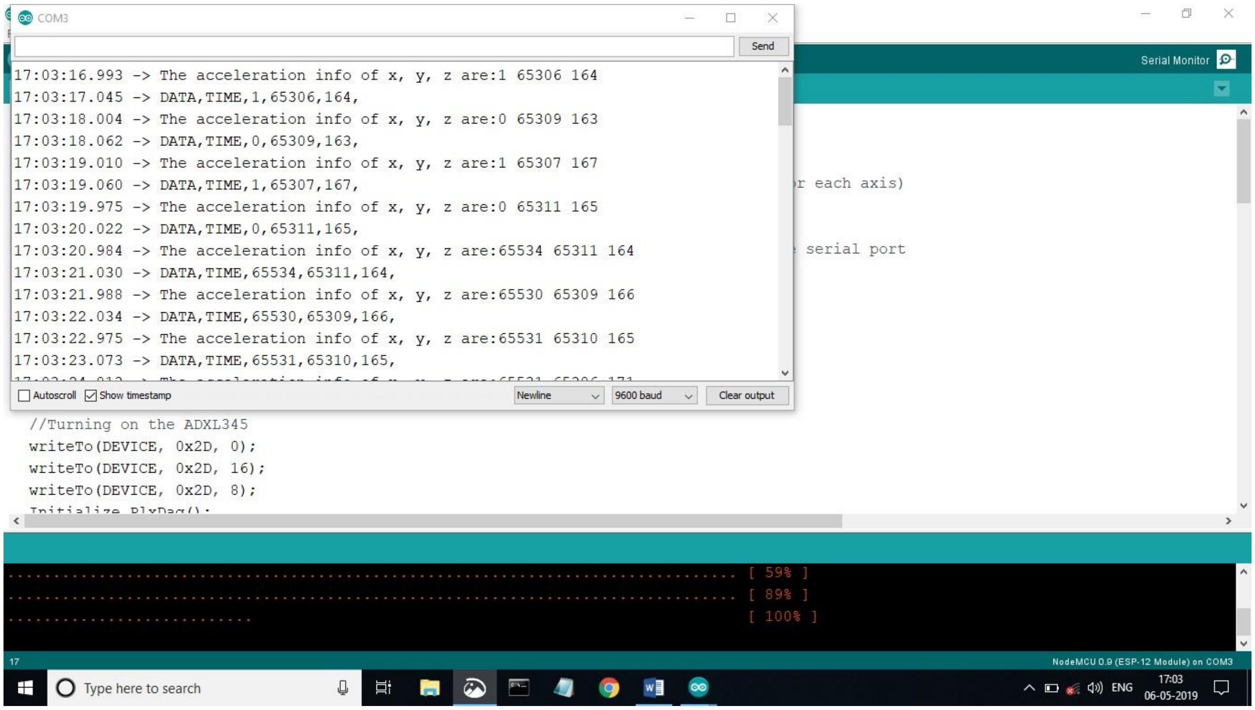


Fig 6.5 Accelerometer reading on serial plotter

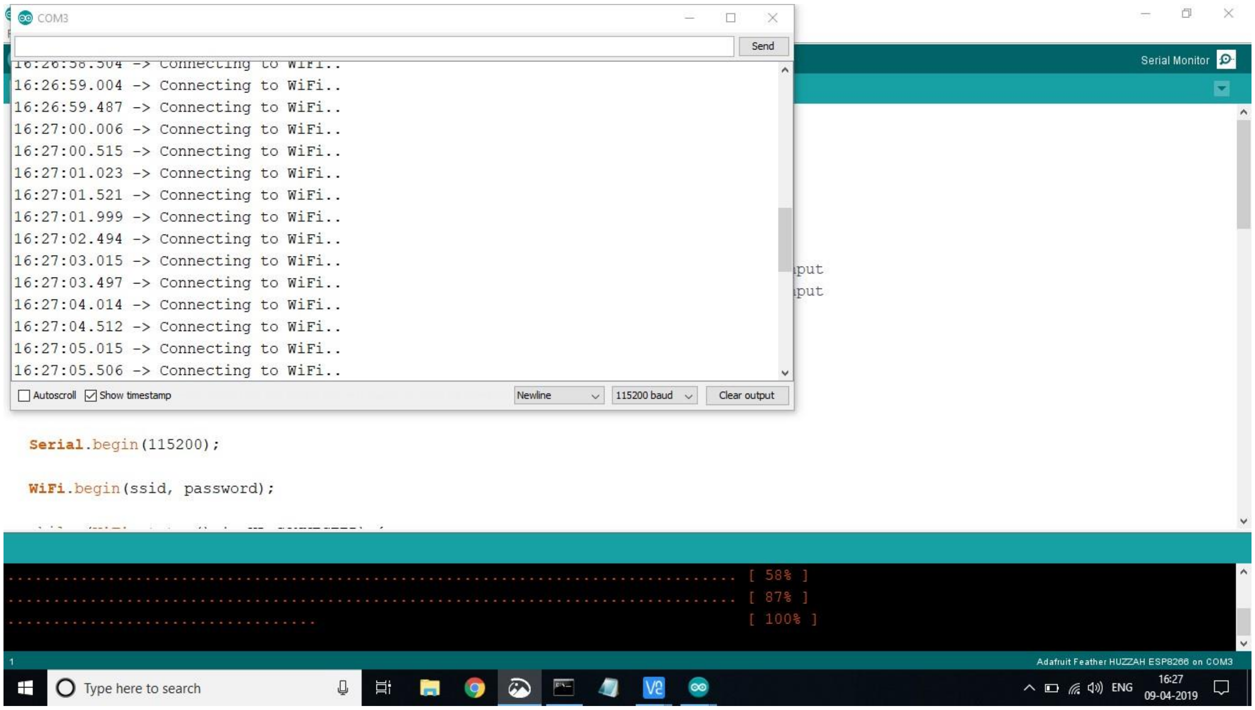


Fig 6.6 Showing connectivity from NodeMcu to raspberyy pie

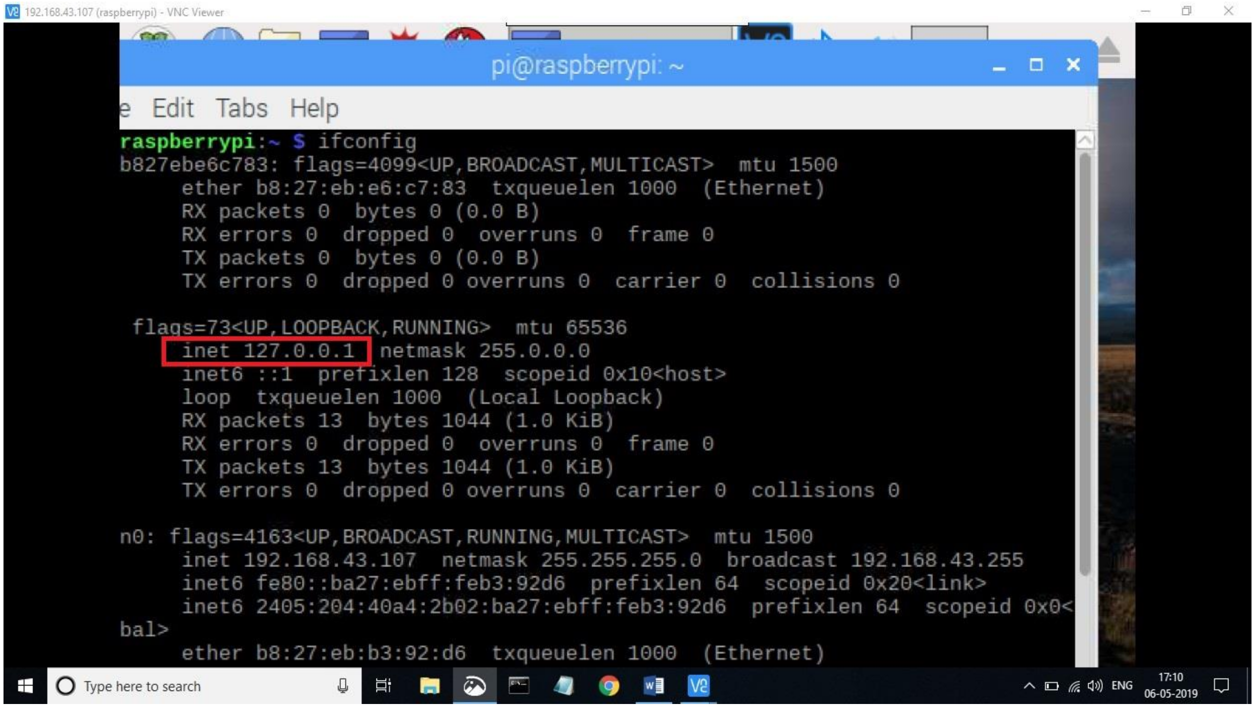


Fig 6.7 Ip Address of Raspberry Pi

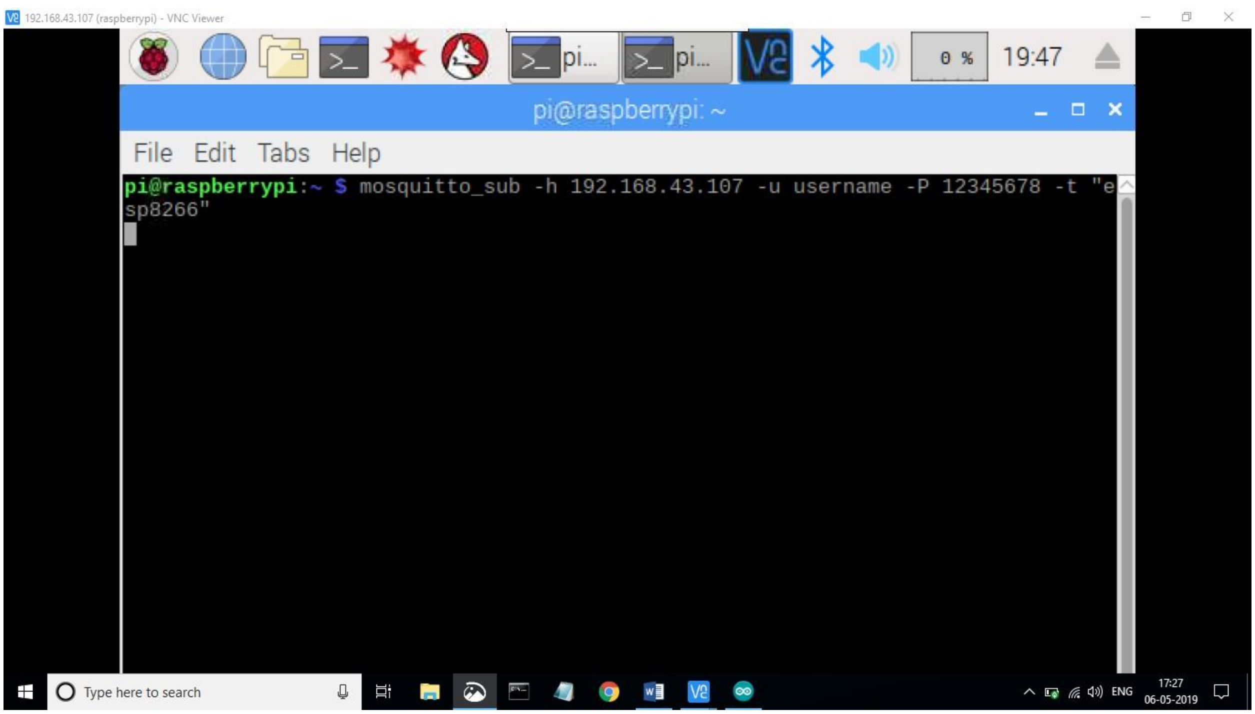


Fig 6.8 Showing the received signal at LXT terminal.

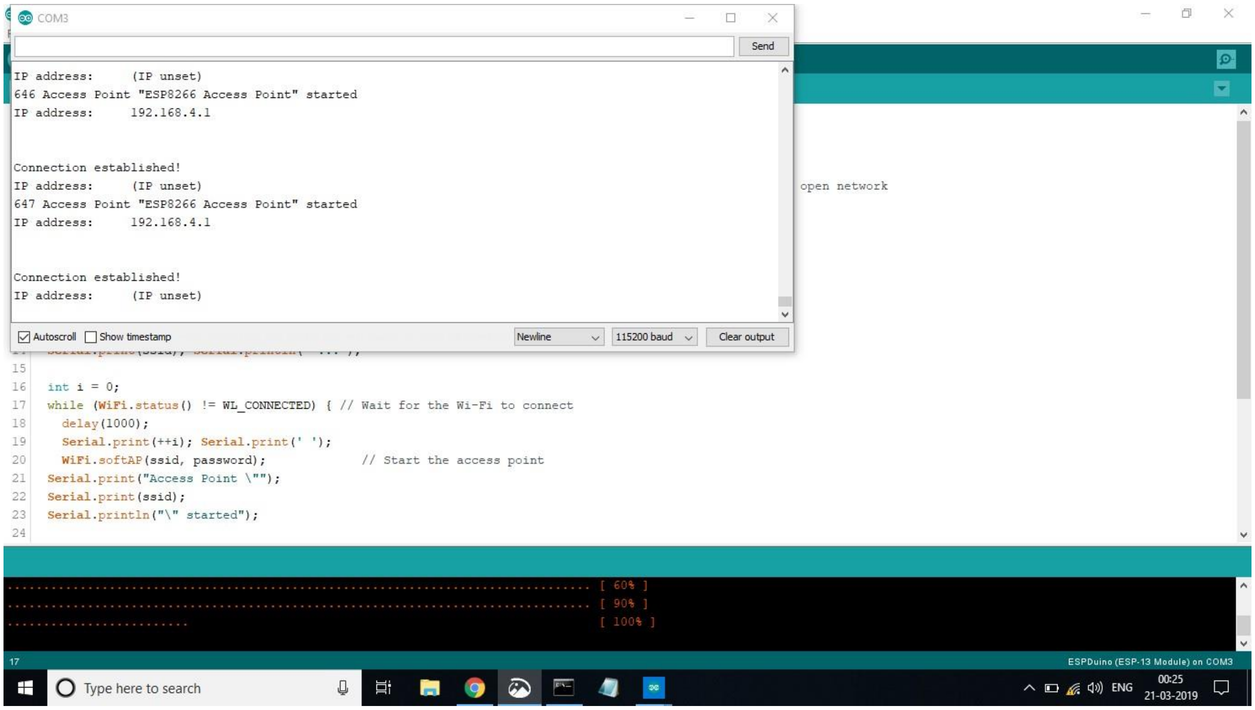


Fig 6.9 NodeMcu IP Address

Conclusion:

From the above real time pictures of this project it can be concluded that depending upon different statistical features different gesture is recognized using Node MCU and data is transmitted to raspberry pi for control of cart. MQTT protocol is implemented for transmission.

In feature this work can be extended with the help of machine learning for more precise feature extraction and control of the car.

References:

[1] Kausthub N P, 1Rounak S, 2Bhakthavathsalam R, 2Gowranga K H, 2 Saqquaf S M “Hand GestureInitiated Motor Operations for Remote Robot Surveillance System using Raspberry Pi Processor on aZigBee Communication Link” International Journal of Innovative Research in Electronics and Communications (IJIREC) Volume 2, Issue 5, July 2015, PP 55-62 ISSN 2349-4042

[2] Akram Bayat , Marc Pomplun, Duc A. Tran “A Study on Human Activity Recognition Using Accelerometer Data from Smartphones”, The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC-2014) .

[3] JiahuiWu1 , Gang Pan1 , Daqing Zhang2 , Guande Qi1 , and Shijian Li1 “Gesture Recognition with a3-D Accelerometer”, Springer-Verlag Berlin Heidelberg 2009

[4] Manan Mehta, “A BREAKTHROUGH IN WIRELESS SENSOR NETWORKS AND INTERNETOF THINGS”, International Journal of Electronics and Communication Engineering & Technology (IJECET) Volume 6, Issue 8, Aug 2015,

[5] M. Todica, “Distance measurement with SR HC 04, NodeMcu ESP 8266 and Blynk”, 24 July 2018

[6] DipaSoni, AshwinMakwana “A SURVEY ON MQTT: A PROTOCOL OF INTERNET OFTHINGS(IOT)”, DipaSoni on 12 April 2017.

[7] Nita Kalambe, Prof. Dhruv Thakur, Prof. Shubhankar Paul, “Review of Microstrip Patch AntennaUsing UWB for Wireless Communication Devices”, IJCSMC, Vol. 4, Issue. 1, January 2015, ISSN 2320–088X

[8] Ullas B S ,Anush S , Roopa J , GovindaRaju M, “Machine to Machine Communication forSmart Systems using MQTT”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,(An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 3, March 2014

[9] ArKarKyaw , Hong Phat Truong, Justin Joseph, “Low-Cost Computing Using Raspberry Pi 2Model B” , Manuscript submitted March 1, 2017; accepted June 21, 2017

[10] NitiRajendra Patel, “Interactive Interface for DC Motor using GUI with Raspberry Pi Controller”,Interactive Interface for DC Motor using GUI with Raspberry Pi Controller, (An ISO 3297: 2007 Certified Organization) Vol. 5, Issue 6, June 2017

[11] Fang-Ting Liu, Yong-Ting Wang, Hsi-Pin Ma, “Gesture Recognition with Wearable 9-axis

Sensors” Manuscript submitted June 1, 2016; accepted October 3, 2017,

[12] GorkaMarqués, KoldoBasterretxea “Efficient Algorithms for Accelerometer-based Wearable HandGesture Recognition Systems”, 978-1-4673-8299-1/15 $31.00 © 2015 IEEE

[13] Zhenyu He, LianwenJin, Lixin Zhen, Jiancheng Huang, “Gesture recognition based on 3Daccelerometer for cell phones interaction”, 978-1-4244-2342-2/08/$25.00 ©2008 IEEE.

[14] Zhenyu He, LianwenJin, Lixin Zhen, Jiancheng Huang, “Gesture recognition based on 3Daccelerometer for cell phones interaction”, 978-1-4244-2342-2/08/$25.00 ©2008 IEEE.

[15]GorkaMarqués ,KoldoBasterretxea “Efficient Algorithms for Accelerometer-basedWearable Hand Gesture Recognition Systems”, 978-1-4673-8299-1/15 $31.00 © 2015 IEEE

[16]GorkaMarqués ,KoldoBasterretxea “Efficient Algorithms for Accelerometer-basedWearable Hand Gesture Recognition Systems”, 978-1-4673-8299-1/15 $31.00 © 2015 IEEE

[17] https://robottini.altervista.org/arduino-and-real-time-charts-in-excel

[18]https://www.parallax.com/downloads/plx-daq

[19] https://www.parallax.com/downloads/plx-daq

[20] https://www.parallax.com/downloads/plx-daq

[21] https://en.wikipedia.org/wiki/Principal_component_analysis

[22] https://en.wikipedia.org/wiki/Feature_extraction

Appendix A Data sheet of ADXL345