

**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

**Minor Project Report**

**On**

**Automatic Vehicle Barrier**

**Submitted By:**

Nabin Paneru (THA075BEI025)

Samrat Kafle (THA075BEI037)

Shiv Ranjan Gupta (THA075BEI039)

Rikrish Shrestha (THA074BEX030)

**Submitted To:**

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

February 27, 2022

Shape, arrow

Description automatically generated  
**TRIBHUVAN UNIVERSITY**

**INSTITUTE OF ENGINEERING**

**THAPATHALI CAMPUS**

**Minor Project Report**

**On**

**Automatic Vehicle Barrier**

**Submitted By:**

Nabin Paneru (THA075BEI025)

Samrat Kafle (THA075BEI037)

Shiv Ranjan Gupta (THA075BEI039)

Rikrish Shrestha (THA074BEX030)

**Submitted To:**

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal  
  
In partial fulfillment for the award of the Bachelor’s Degree in Electronics, Communication and Information Engineering

**Under the Supervision of**

Associate Professor Er. Shanta Maharjan  
  
February 27, 2022

# DECLARATION

We hereby declare that the report of the project entitled “**Automatic Vehicle Barrier**” which is being submitted to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus,** in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in **Electronics and Communication Engineering**, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work.

Nabin Paneru (THA075BEI025) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Samrat Kafle (THA075BEI037) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Shiv Ranjan Gupta (THA075BEI039) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Rikrish Shrestha (THA074BEX030) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Date**: February, 2022

# CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the **Department of Electronics and Computer Engineering, IOE, Thapathali Campus**, a minor project work entitled “**Automatic Vehicle Barrier**” submitted by **Nabin Paneru**, **Samrat Kafle**, **Shiv Ranjan Gupta** and **Rikrish Shrestha** in partial fulfillment for the award of Bachelor’s Degree in Electronics and Communication Engineering. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor’s degree of Electronics and Communication Engineering

Project Supervisor

Assistant Professor Er.Shanta Maharjan

Department of Electronics and Computer Engineering, Thapathali Campus

External Examiner

Prof.

Department of

Project Co-ordinator

Mr. Umesh Kanta Ghimire

Department of Electronics and Computer Engineering, Thapathali Campus

Mr. Kiran Chandra Dahal

Head of the Department,

Department of Electronics and Computer Engineering, Thapathali Campus

February, 2022

# COPYRIGHT

The author has agreed that the library, Department of Electronics and Computer Engineering, Thapathali Campus, may make this report freely available for inspection. Moreover, the author has agreed that the permission for extensive copying of this project work for scholarly purpose may be granted by the professor/lecturer, who supervised the project work recorded herein or, in their absence, by the head of the department. It is understood that the recognition will be given to the author of this report and to the Department of Electronics and Computer Engineering, IOE, Thapathali Campus in any use of the material of this report. Copying of publication or other use of this report for financial gain without approval of the Department of Electronics and Computer Engineering, IOE, Thapathali Campus and author’s written permission is prohibited.

Request for permission to copy or to make any use of the material in this project in whole or part should be addressed to department of Electronics and Computer Engineering, IOE, Thapathali Campus.

# ACKNOWLEDGEMENT

We would like to thank our Department of Electronics & Computer Engineering, Thapathali Campus for providing us with the proper guidance, platform, and an exciting opportunity to increase our productivity through this Minor project.

We would also like to thank and express our gratitude towards our supervisor Associate Professor Er. Shanta Maharjan sir for his helpful suggestion and guidance throughout project proposal.

Nabin Paneru (THA075BEI025)

Samrat Kafle (THA075BEI037)

Shiv Ranjan Gupta (THA075BEI039)

# ABSTRACT

Barrier management is one of the critical problems. As the number of parking increased, the manpower required for management for barrier also increased significantly. This paper offers the use of sensors and android app in a barrier management system using algorithms. MIT App inventor algorithm techniques are used to develop an application that is used to manage the barrier. The sensor is used for car detection in Slot. These Ultrasonic sensors are connected to NodeMCU.

*Keywords: Authentication, Barrier, Internet of Things, MIT App inventor, Ultrasonic sensor*

Table of Contents

[DECLARATION ii](#_Toc96978081)

[CERTIFICATE OF APPROVAL iii](#_Toc96978082)

[COPYRIGHT iv](#_Toc96978083)

[ACKNOWLEDGEMENT v](#_Toc96978084)

[ABSTRACT vi](#_Toc96978085)

[List of Figures x](#_Toc96978086)

[List of Tables xi](#_Toc96978087)

[List of Abbreviations xii](#_Toc96978088)

[1. INTRODUCTION 1](#_Toc96978089)

[**1.1** **Background** 1](#_Toc96978090)

[**1.2** **Motivation** 2](#_Toc96978091)

[**1.3** **Problem Statement** 2](#_Toc96978092)

[**1.4** **Project Objectives** 3](#_Toc96978093)

[**1.5** **Project Scope and Applications** 3](#_Toc96978094)

[2. LITERATURE REVIEW 4](#_Toc96978095)

[3. REQUIREMENT ANALYSIS 6](#_Toc96978096)

[**3.1** **Hardware Requirement** 6](#_Toc96978097)

[**3.2** **Software Requirement** 8](#_Toc96978098)

[**3.3** **Feasibility Study** 10](#_Toc96978099)

[3.3.1 Economic Feasibility 10](#_Toc96978100)

[3.3.2 Operational Feasibility 10](#_Toc96978101)

[3.3.3 Technical Feasibility 10](#_Toc96978102)

[4. SYSTEM ARCHITECTURE AND METHODOLOGY 11](#_Toc96978103)

[**4.1** **Block Diagram** 11](#_Toc96978104)

[**4.2** **Working Principle** 11](#_Toc96978105)

[4.2.1 Car Entering Module 16](#_Toc96978106)

[4.2.2 Car Exiting Module 16](#_Toc96978107)

[**4.3** **Flowchart** 17](#_Toc96978108)

[5. IMPLEMENTATION DETAILS 18](#_Toc96978109)

[**5.1** **NodeMCU** 18](#_Toc96978110)

[**5.2** **Power Management for the circuit components** 21](#_Toc96978111)

[5.2.1 Managing Power for NodeMCU 21](#_Toc96978112)

[5.2.2 Managing Power for Ultrasonic Sensors 21](#_Toc96978113)

[5.2.3 Managing Power for Servo motor 22](#_Toc96978114)

[5.2.1 Management of Power Supply for LED 22](#_Toc96978115)

[**5.3** **Creating an App for control** 24](#_Toc96978116)

[**5.4** **HTTP (Hypertext Transfer Protocol)** 25](#_Toc96978117)

[**5.5** **HTML(Hypertext Markup Language)** 26](#_Toc96978118)

[**5.6** **Use of Libraries** 27](#_Toc96978119)

[6. RESULT AND ANALYSIS 29](#_Toc96978120)

[**6.1** **Ultrasonic Sensor** 29](#_Toc96978121)

[6.1.1 Output of Ultrasonic Sensor 29](#_Toc96978122)

[6.1.2 Analysis 30](#_Toc96978123)

[**6.2** **Servo Motor** 31](#_Toc96978124)

[6.2.1 Output of Servo motor 31](#_Toc96978125)

[6.2.2 Analysis 31](#_Toc96978126)

[**6.3** **Error Analysis** 33](#_Toc96978127)

[6.3.1 Providing power to Servo 33](#_Toc96978128)

[6.3.2 Providing power to the Ultrasonic Sensor 34](#_Toc96978129)

[6.3.3 Assembling the circuit components together 34](#_Toc96978130)

[**6.4** **Simulation of a parking system prototype** 34](#_Toc96978131)

[7. FUTURE ENHANCEMENT 39](#_Toc96978132)

[8. CONCLUSION 40](#_Toc96978133)

[9. APPENDICES 41](#_Toc96978134)

[**APPENDIX A: PROJECT BUDGET** 41](#_Toc96978135)

[**APPENDIX B: PROJECT SCHEDULE** 42](#_Toc96978136)

[**APPENDIX C: CIRCUIT DESIGN DIAGRAMS** 43](#_Toc96978137)

[**APPENDIX D: MODULE SPECIFICATION** 44](#_Toc96978138)

# List of Figures

[Figure 1‑1: Automatic vehicle barrier 2](#_Toc96978196)

[Figure 3‑1: NodeMCU 6](#_Toc96978197)

[Figure 3‑2: Ultrasonic Sensor 6](#_Toc96978198)

[Figure 3‑3: Servo Motor 7](#_Toc96978199)

[Figure 3‑4: Jumper Wire 7](#_Toc96978200)

[Figure 3‑5 : Matrix Board 8](#_Toc96978201)

[Figure 3‑6: MIT App Inventor Designer Screen 9](#_Toc96978202)

[Figure 3‑7: MIT App Inventor Blocks Editor Screen 9](#_Toc96978203)

[Figure 4‑1: System Block Diagram 11](#_Toc96978204)

[Figure 4‑2: Timing diagram of the Ultrasonic Sensor 12](#_Toc96978205)

[Figure 4‑3: No rotation of barrier for parking occupied 13](#_Toc96978206)

[Figure 4‑4: Rotation of barrier for parking unoccupied 14](#_Toc96978207)

[Figure 4‑5: No Action for incorrect Wi-Fi Password 15](#_Toc96978208)

[Figure 4‑6: Perform Action for correct Wi-Fi Password 15](#_Toc96978209)

[Figure 4‑7: Flowchart for entering 17](#_Toc96978210)

[Figure 5‑1: NodeMCU Pin out Diagram 18](#_Toc96978211)

[Figure 5‑2: Voltage Drop Across NodeMCU 21](#_Toc96978212)

[Figure 5‑3: Voltage Drop Across Ultrasonic Sensors 21](#_Toc96978213)

[Figure 5‑4: Initial voltage drop across Servo 22](#_Toc96978214)

[Figure 5‑5: Voltage Drop Across Servo for Maximum Rotation 22](#_Toc96978215)

[Figure 5‑6: Multimeter reading when LED is on 23](#_Toc96978216)

[Figure 5‑7: User Interface Screen for Mobile Application 24](#_Toc96978217)

[Figure 5‑8: NodeMCU as HTTP Server using Wi-Fi AP mode 25](#_Toc96978218)

[Figure 5‑9: HTTP GET Request Method 26](#_Toc96978219)

[Figure 6‑1: Placement of the ultrasonic sensor 29](#_Toc96978220)

[Figure 6‑2: Output of the Ultrasonic Sensor 29](#_Toc96978221)

[Figure 6‑3: Waveform of the Trigger Input signal 30](#_Toc96978222)

[Figure 6‑4: Waveform of Echo Output Signal for a Trigger Input 30](#_Toc96978223)

[Figure 6‑5: Path for data flow for servo at exit 31](#_Toc96978224)

[Figure 6‑6: PWM pulse for servo rotation 32](#_Toc96978225)

[Figure 6‑7: Closed loop servo motor working mechanism 33](#_Toc96978226)

[Figure 6‑8: No changes in the received echo signal for a pulse of Trigger signal 34](#_Toc96978227)

[Figure 6‑9: Testing one or more simulated sensors 35](#_Toc96978228)

[Figure 6‑10: Entry Position of Vehicle 36](#_Toc96978229)

[Figure 6‑11: Ticket info 37](#_Toc96978230)

[Figure 6‑12: Truck going to park 37](#_Toc96978231)

[Figure 8‑1: Circuit Diagram 43](#_Toc96978232)

# List of Tables

[Table 9‑1: Budget Analysis 41](#_Toc96978184)

[Table 9‑2: Gantt chart 42](#_Toc96978185)

[Table 9‑3: NodeMCU specifications 44](#_Toc96978186)

[Table 9‑4: Pin Description of NodeMCU 45](#_Toc96978187)

[Table 9‑5: Ultrasonic Sensor Pin Configuration 46](#_Toc96978188)

[Table 9‑6: Electrical Specification of Ultrasonic Sensor 46](#_Toc96978189)

[Table 9‑7: Pin Configuration of Servo Motor 47](#_Toc96978190)

[Table 9‑8: Electrical Specification of Servo Motor 47](#_Toc96978191)

# List of Abbreviations

ADC Analog to Digital Converter

IOT Internet of Things

IR Infrared

ITS Information Technology

NodeMCU Node Micro Controller Unit

# INTRODUCTION

## **Background**

In recent years, the IOT has been applied in many ways. The barrier management system is one part of the technology of the IOT. The concept of the IOT starts from a device that can be traced, controlled, or monitored over the internet. IOT extends the use of Internet providing the communication, and thus inter-network of the devices and physical objects, or ‘Things’. The two prominent words in IOT are “internet” and “things”. Internet means a vast global network of connected servers, computers, tablets and mobiles using the internationally used protocols and connecting systems. Internet enables sending, receiving, or communicating of information. Thing in English has number of uses and meanings. Dictionary meaning of ‘Thing’ is a term used to reference to a physical object, an action or idea, situation or activity, in case when we do not wish to be precise. IOT, in general consists of inter-network of the devices and physical objects, number of objects can gather the data at remote locations and communicate to units managing, acquiring, organizing and analyzing the data in the processes and services. It provides a vision where things (wearable, watch, alarm clock, home devices, surrounding objects with) become smart and behave alive through sensing, computing and communicating by embedded small devices which interact with remote objects or persons through connectivity

One of the systems of barrier management is to control the barrier via handheld device. This is related to barrier problems in which one of them is the allocation of human manpower for the barrier management and tends to more difficult along with the increasing number of parking systems and barrier. Problems related to barrier management can be solved if the management can control the barrier through the application. As the result, the concept of the IOT applies to the barrier management.

Physical Object + Controller, Sensor and Actuators + Internet = Internet of Things

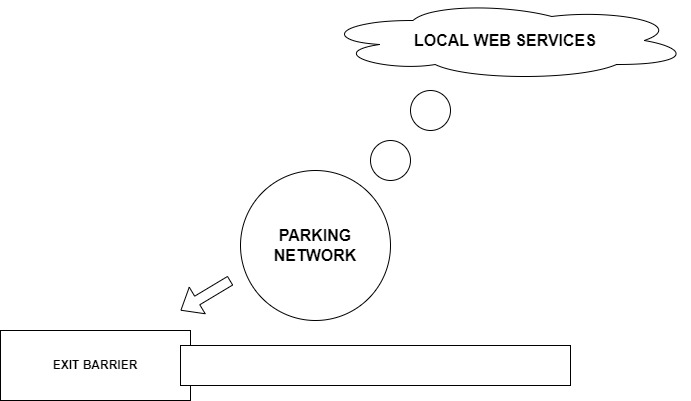


Figure 1‑1: Automatic vehicle barrier

## **Motivation**

Since the first meeting of our team members, we asked each other that we want to do this project. As a conclusion of that meeting, we came to know that all of us are excited to do something related to automated system. Our country uses manual system for management of barrier, watching present scenario and being an aspirant of Electronics and Communication engineering, we had a feeling of helping barrier management by using this technology in their organization. It is observable that the IOT is an important field that the whole world seeks to make the best use of it in order to decrease the use of human manpower that increases the overall cost and decreases the efficiency of barrier management. For this reason automatic vehicle barrier system is presented as a solution for barrier management. The objective now is to develop an app interface for barrier management.

## **Problem Statement**

The use of human manpower can be reduced by an automated system that manages the barrier system. The fact that the efficiency of an automated system is more than manual human managed system is known to all. The manual management of barrier can be time consuming and less efficient, Here we going to see how to reduce the barrier management problem automating the car parking using offerings that are transforming cities by improving infrastructure, creating more efficient and cost-effective municipal services, enhancing public transportation, reducing traffic congestion, and keeping citizens safe and more engaged in the community.

## **Project Objectives**

The main objectives of our project are listed below:

* To develop a system that increases efficiency of barrier management.
* Decrease the use of human manpower and automation of work.

## **Project Scope and Applications**

This project can be used by anyone desiring to build a parking area. Hotels, shopping malls, Cinema Halls, Government, and private offices where many people come in their personal vehicle this project can help guide the parking management team on the aspects of barrier management

It can be used for parking management by enhancing the features of the system by using Machine Learning Algorithm for obstacle avoidance in the parking. Furthermore, RFID tag can be used for authentication of the user also an application interface can be developed for monitoring of spots and from the perspective of user booking the parking slot.

\

# LITERATURE REVIEW

In this age of increasing modernization, demand for automated system is highly increasing. Due to limitations in overall trained workforce and allocation of trained workforce to higher priority works; we have explored automated system for security control issues.

In [1], monitoring of empty space is performed by the end user and monitoring environmental conditions, where light and temperature are performed by an authorized person. Infrared sensors are used when indoor parking and ultrasonic sensors are used when outdoor parking is available. Light and heat are obtained by binding according to the action of light and heat respectively.

In [2], vehicles are detected using ultrasonic sensors. The program uses a supervised learning algorithm to find stationary and empty parking spaces.

In [3], Automatic number plate camera is used to effectively manage, monitor and protect parking spaces, Android application is used to make it easier for drivers to remember their parking space, however, There are no private parking detector centers, details for incoming drivers about the current state of parking spaces.

In [4] Intelligent parking using IOT technology helps design and build up a real parking system that provides information on vacant spaces and helps the user. This paper uses a computer view to find a vehicle number for security improvements.

In [5], parts of a good parking system are a parking meter, Wi-Fi enabled laptop. The parking meter contains ultrasonic sensors to determine the position of the seat in the parking lot, an alarm IC is used for improper parking. A motor vehicle license plate is also obtained using a camera.

In [6], the Hough circle modification is applied to images to determine the number of unused parking spaces. Hough modification is used for shape detection. This system is designed specifically for four wheels.

In [7], smart parking systems have a magnetic sensor with transceiver 802.15.4, which is used to detect the number of empty parking spaces. Barrier collects this information and sends it to an Internet server. Zigbee is used for wireless communication. Parking can be booked in advance. System performance is performed using a genetic algorithm.

In [8], an ultrasonic sensor is used to locate the parking space. The mobile app and online portal are being upgraded to display the number of vacant sites. Users can save the parking space in advance and pay for it. Users are provided with a barcode to enter the parking lot with an automatic collection. Offline users can provide cash at the time of login.

# REQUIREMENT ANALYSIS

## **Hardware Requirement**

Following Hardware Components are required for Automatic Vehicle Barrier System.

1. NodeMCU

NodeMCU is open source platform. NodeMCU Dev Kit/board consists of ESP8266 WI-FI enabled chip. It offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor. ESP8266 NodeMCU has powerful on-board processing and storage capabilities that allow it to be integrated with the sensors specific devices through its GPIOs with minimal development up-front and minimal loading during runtime.



Figure 3‑1: NodeMCU

1. Ultrasonic Sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).



Figure 3‑2: Ultrasonic Sensor

1. Servo Motor

A servomotor is **a** rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity, and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It can rotate with great precision. It is just made up of a simple motor which runs through a servo **mechanism**. A servo motor usually comes with a gear arrangement that allows us to get a very high torque servo motor in small and lightweight packages.



Figure 3‑3: Servo Motor

1. Jumper Wire

A jumper wire is an electrical wire, or group of them in a cable, with a connector or pin at each end, which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering. Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.



Figure 3‑4: Jumper Wire

1. Matrix Board

Matrix boards are prototype circuit boards used for laying out electrical components securely, and can be used for both high frequency and analogue circuit building. When using a matrix board, electrical parts are soldered together and joined with tinned copper wire.

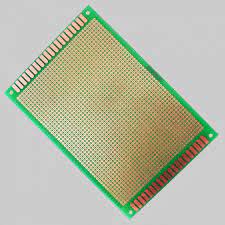


Figure 3‑5 : Matrix Board

## **Software Requirement**

Following Software Components are required for Automatic Vehicle Barrier System.

1. Arduino IDE

Arduino IDE is an open-source cross platform application that is used to write and upload programs to the microcontrollers that are compatible with the Arduino. It supports both the C as well as C++ languages. In our project, we use this software to program NodeMCU.

1. C programming Language

C is a procedural programming language. The major features of C language include low-level access to memory, simple set of keywords, and clean style, these features make C language suitable for system programming like operating system or compiler development.

1. MIT App inventor

MIT App Inventor is an intuitive, visual programming environment that allows everyone to build a fully functional app for smart phones and tablets. Its blocks-based tool facilitates the creation of complex, high-impact apps in significantly less time than traditional programming environments. It uses a graphical interface which helps user to create application just by knowing execution logic.

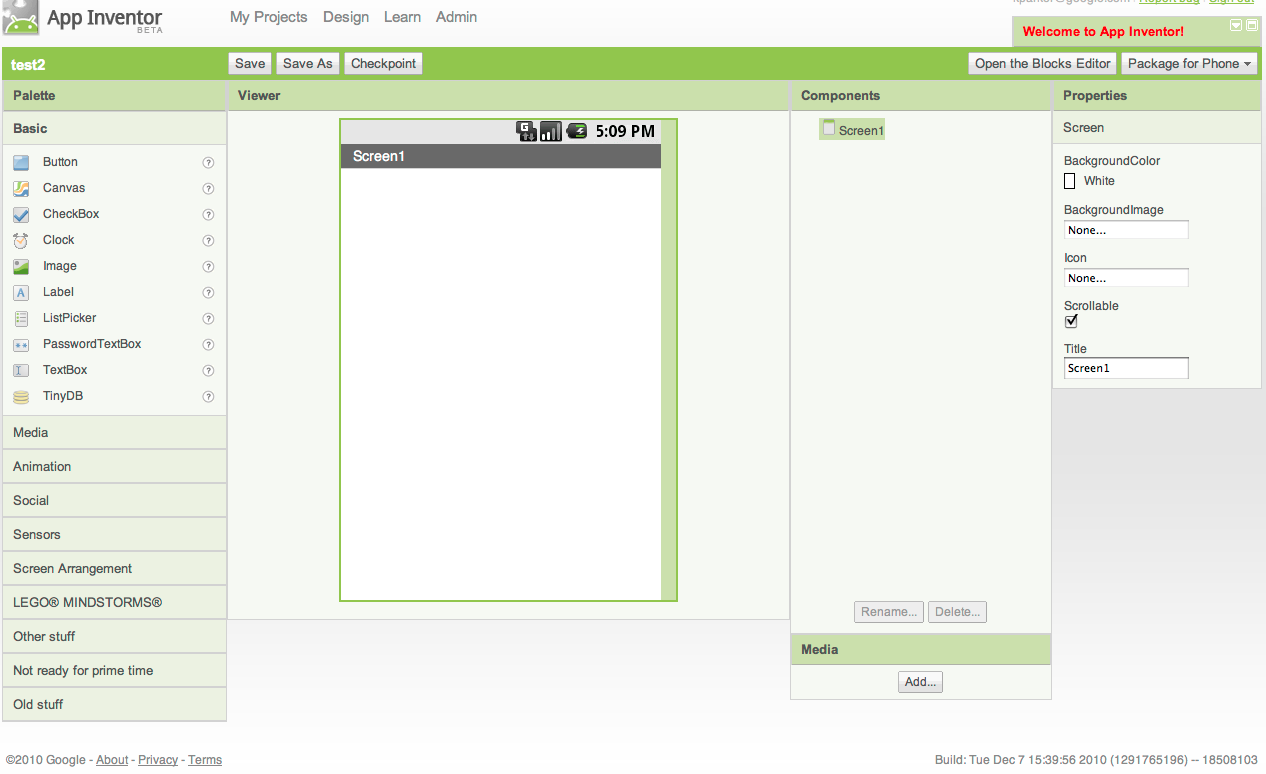


Figure 3‑6: MIT App Inventor Designer Screen

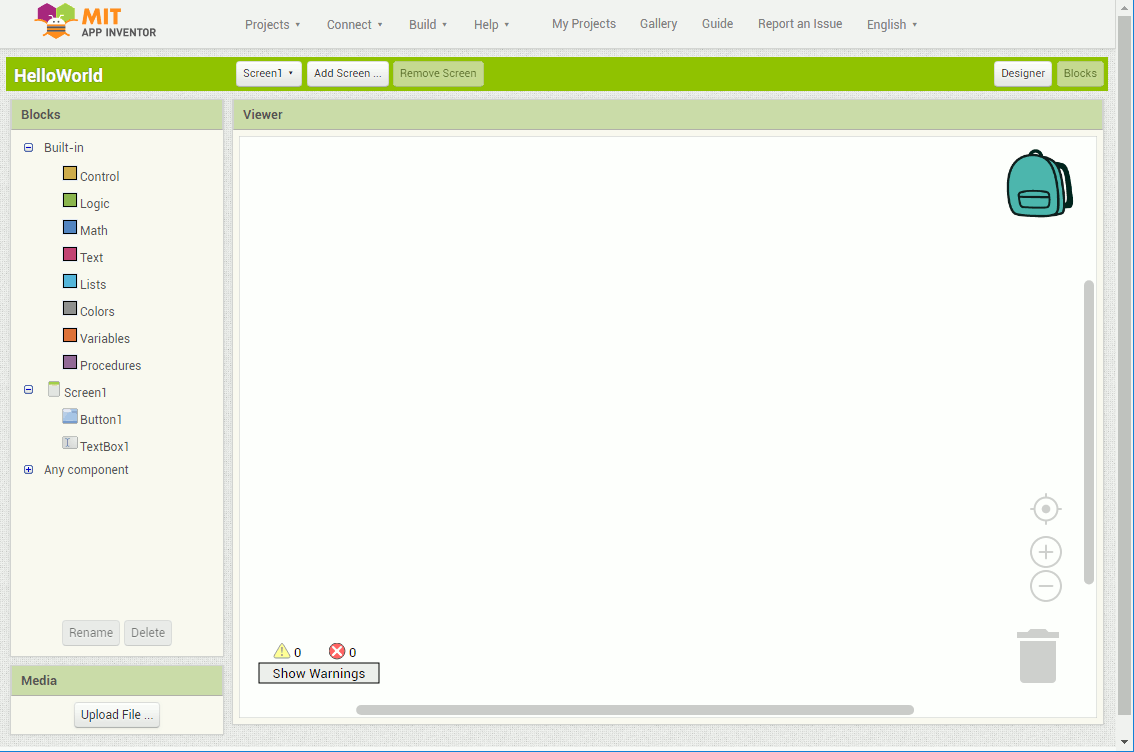
In the Designer Screen we identify and select the required tools. Here the two tools required for automating the barrier are two buttons. One of the button is used for closing the barrier where as the other is used for opening the barrier.

Figure 3‑7: MIT App Inventor Blocks Editor Screen

In the blocks editor screen we define the algorithm of the system.

1. Unity

Unity is an all purpose game engine that supports 2D and 3D graphics, drag and drop functionality and scripting through C.Unity is particularly popular for mobile game development and much of their focus is on mobile platforms. Unity is an adequate platform for developing 2D games even when compared to other dedicated 2D engines, particularly if you plan to release the game across multiple mobile devices.

## **Feasibility Study**

### Economic Feasibility

The task in this study is to assess the financial aspects exemplified by determining the economic feasibility of implementing the system. The system is more hardware than software. So, it will be economically feasible. But to make the model run, we need to program it which is considerable. For the deployment, we can use NodeMCU which is a very low-cost microcontroller and Ultrasonic sensors which is available at affordable price.

### Operational Feasibility

Operational feasibility refers to the evaluation which analyzes how well a system operates. The NodeMCU was designed to help students to create automated systems in an easy and low-cost manner. So, the system will be easy to use and understand and someone with knowledge of microcontroller and sensors can easily operate.

### Technical Feasibility

This type of feasibility is done to make sure whether the available resources would be able to work in the existing infrastructure or not. It also compares the level of technology available for the users and the level of technology required for the system development. Here, the level of technology consists of programming language, hardware and open source app development software. Since, we are able to fulfill the level of technologies required; the project was concluded to be technically feasible.

# SYSTEM ARCHITECTURE AND METHODOLOGY

## **Block Diagram**

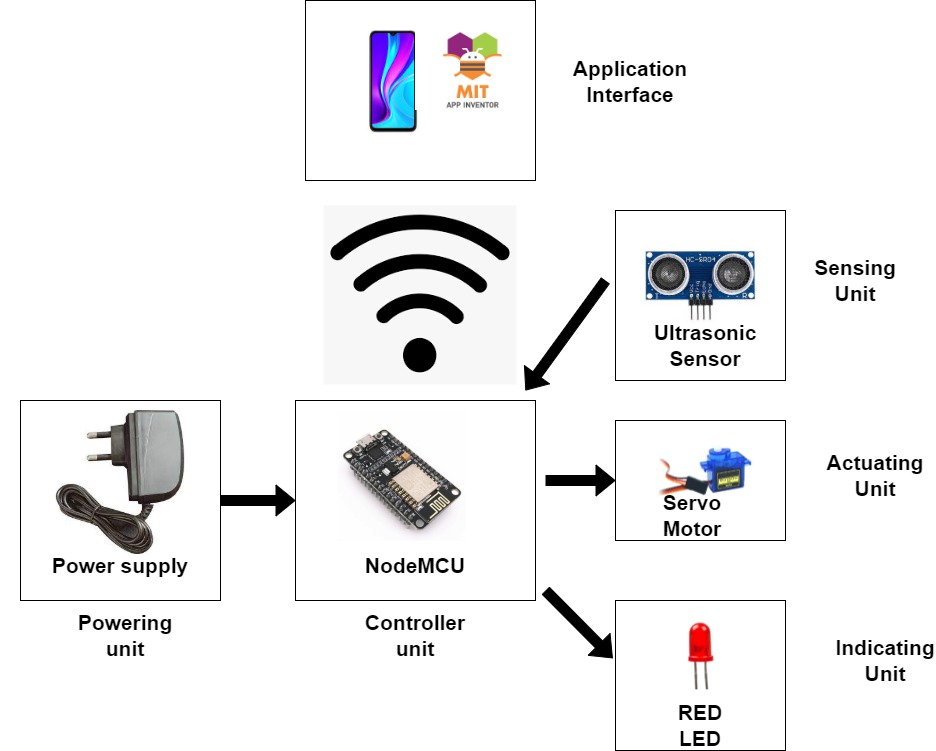


Figure 4‑1: System Block Diagram

## **Working Principle**

In this project of our Automatic Vehicle Barrier System we have NodeMCU as the central controller, three ultrasonic sensors, two servo motor also we have created a separate app to control the barrier at the exit.

We will have to understand how the Ultrasonic sensor module works. It consists of two digital pins trigger as input and echo as output. It also consists of VCC and GND pins used for power supply.

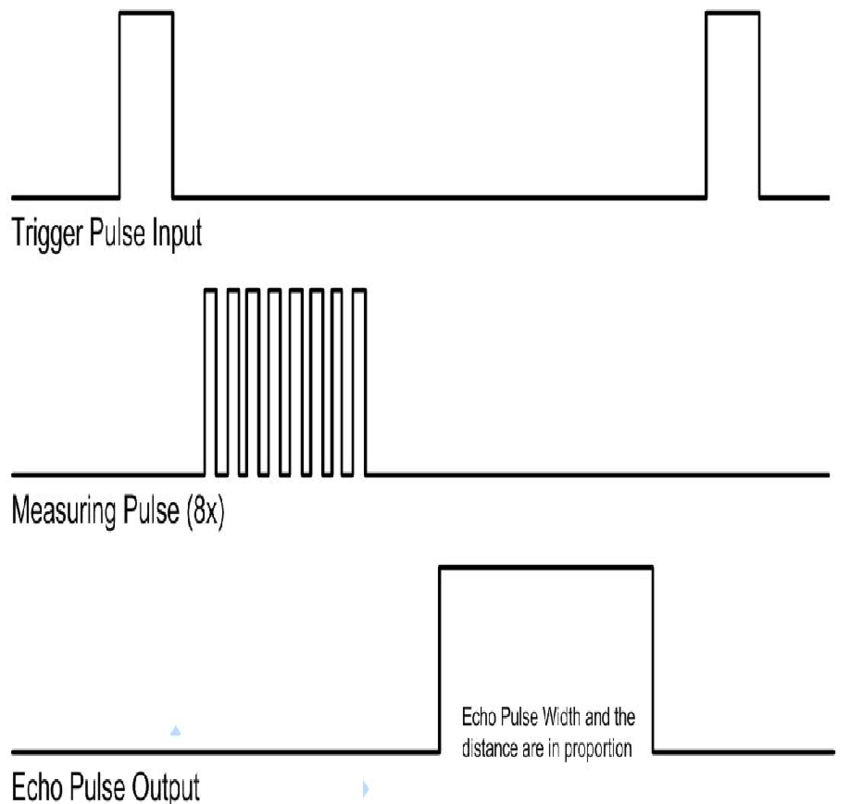


Figure 4‑2: Timing diagram of the Ultrasonic Sensor

Using IO trigger for at least 10us high level signal, The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back, if the signal is back, through high level , time of high output IO duration is the time from sending ultrasonic to returning. Test distance = (high level time \* velocity of sound (340M/S)) /2.

Hence any object near to the ultrasonic sensor is detected by it. Now, we use an algorithm whereby the servo motor at entry is rotated to let the vehicle in only when the ultrasonic sensor at the parking lot detects there is no vehicle in the parking lot. Also the servo motor at exit barrier is rotated based on its control from the mobile application.

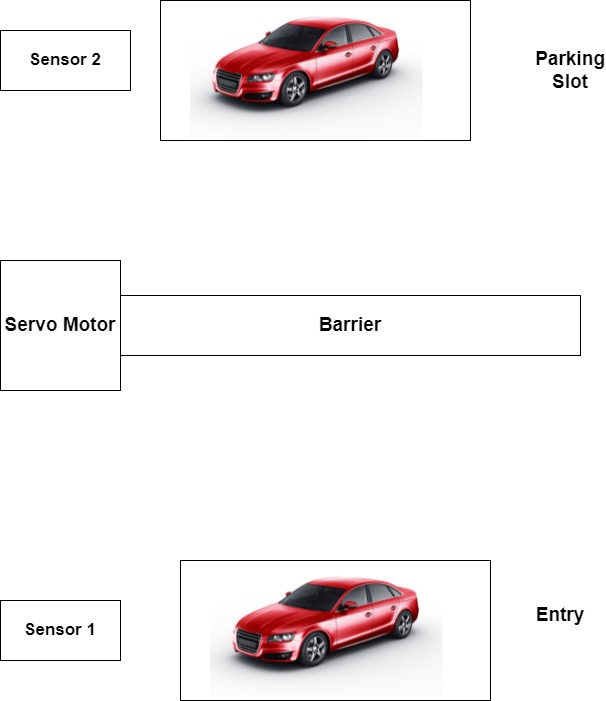


Figure 4‑3: No rotation of barrier for parking occupied

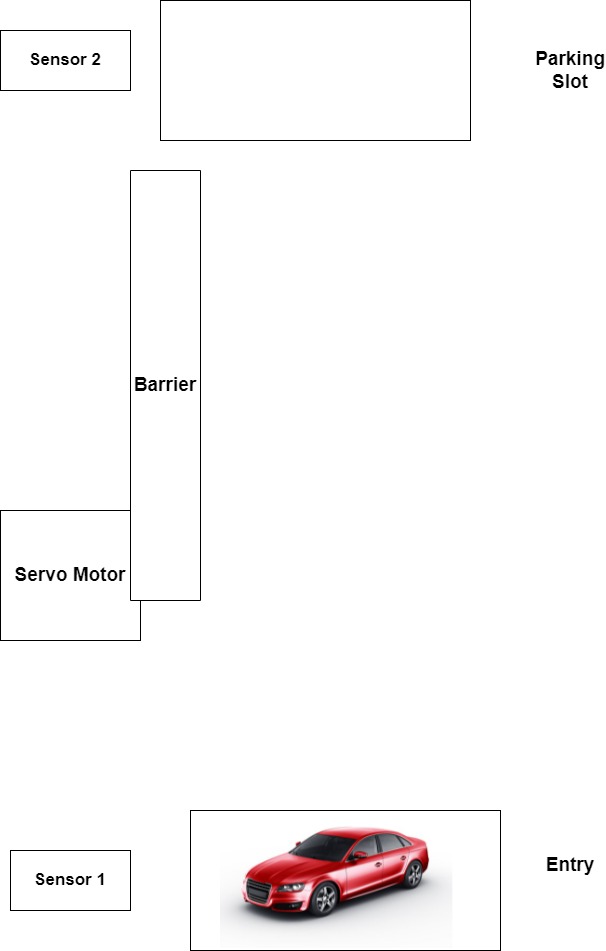


Figure 4‑4: Rotation of barrier for parking unoccupied

There are three conditions to be followed:

* Condition 1:

If parking slot is occupied, do not rotate the barrier as shown in Figure 4-3

I.e. Sensor 1 d<Threshold when Sensor 2 d< Threshold

* Condition 2:

If parking slot is unoccupied, rotate the barrier.

I.e. Sensor 1 d<Threshold when Sensor 2 d>Threshold

Here the NodeMCU is used as a Wi-Fi Access Point whereby we have defined certain Wi-Fi SSID and Password so that only the Authenticated system with the correct

Wi-Fi Password can control the barrier as shown in the Figures below. Here in the Figure 4-5 it is shown that no action is performed when the device is not connected to the hotspot created by the NodeMCU. In the Figure 4-6 it is shown that action is performed when the device is connected to the hotspot created by the NodeMCU.

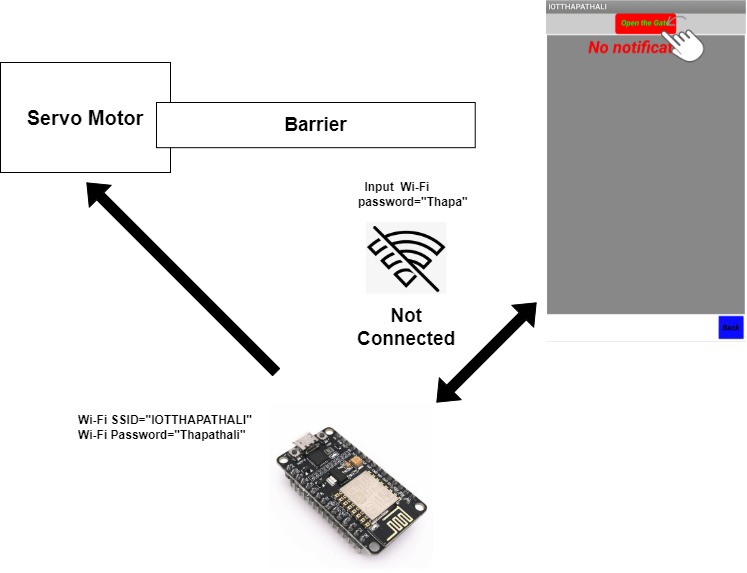


Figure 4‑5: No Action for incorrect Wi-Fi Password

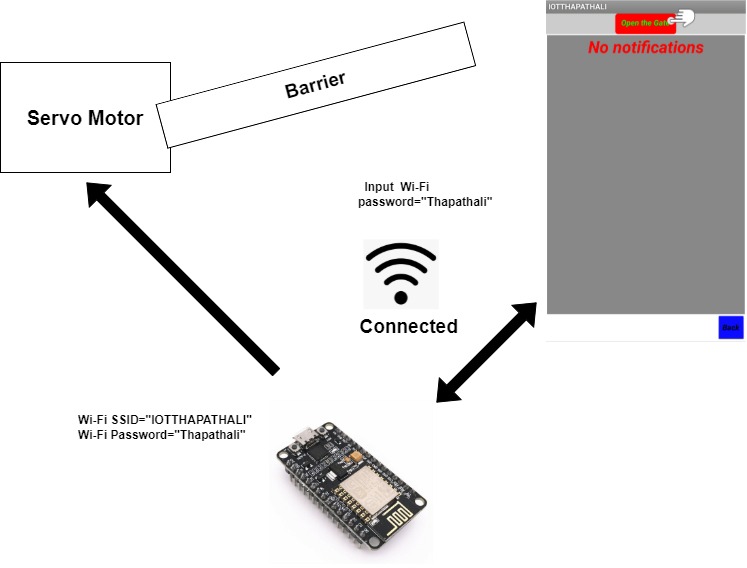


Figure 4‑6: Perform Action for correct Wi-Fi Password

### Car Entering Module

As the car reaches the barrier the ultrasonic sensor in the barrier detects it. The ultrasonic sensor in the parking lot detects whether there is vehicle or not and based on detection of vehicle, the barrier is automated. Now if the vehicle is at entry while other vehicle is being parked the driver is notified through the blinking of red led.

### Car Exiting Module

It’s upon the central server whether or not to rotate the servo to open the barrier.

## **Flowchart**

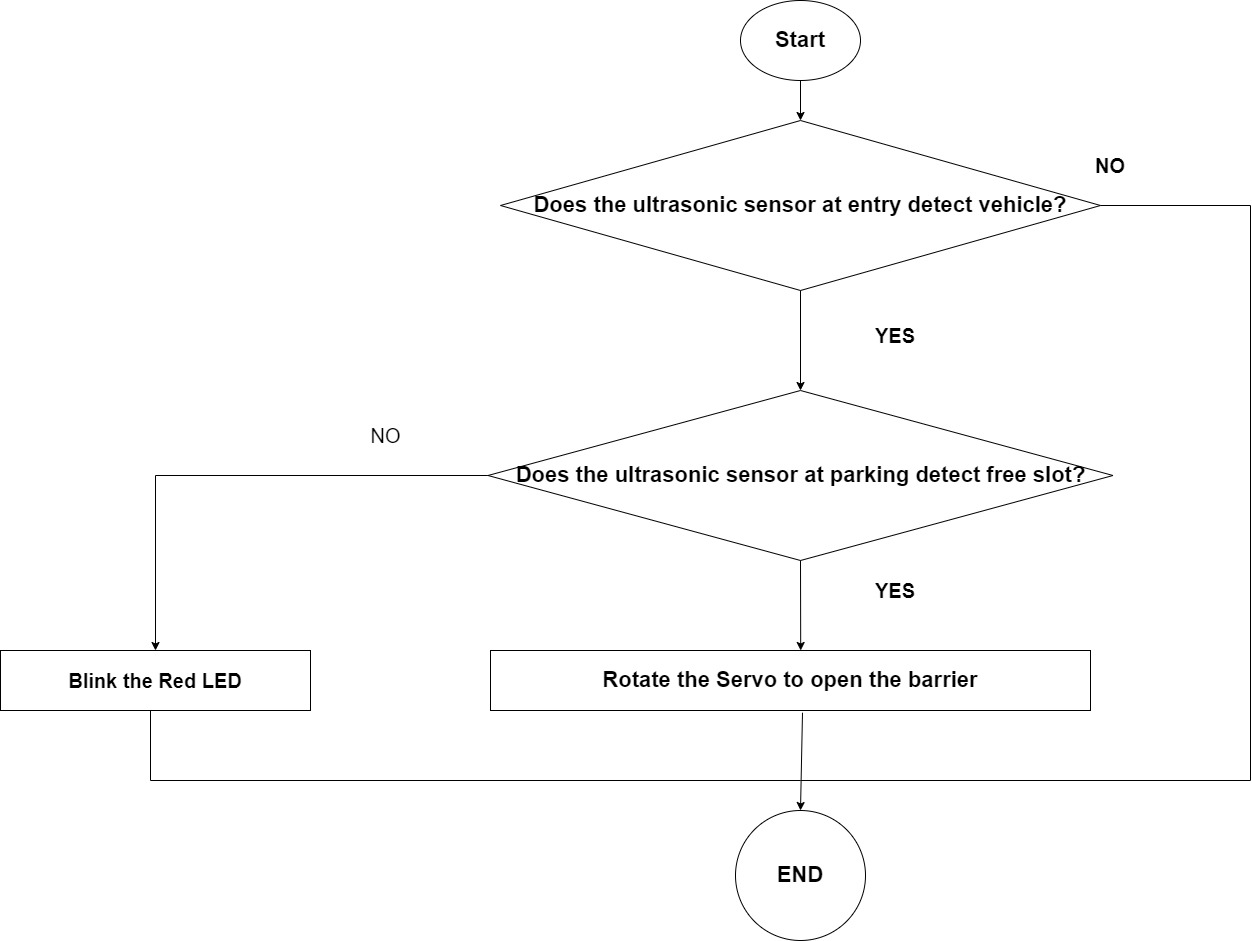


Figure 4‑7: Flowchart for entering

# IMPLEMENTATION DETAILS

## **NodeMCU**

This Project presents the interface of the barrier system to the management team using an authenticated mobile application which is run only when the input password of the management matches the NodeMCU Wi-Fi password here the NodeMCU is programmed as a WIFI Access point. The NodeMCU is also programmed as a web server for connection to the application interface for performing desired action as specified in the application.

The NodeMCU consists of ESP8266 WI-FI Module, 9 digital pins and one Analog pin as shown below in its pin out diagram.



Figure 5‑1: NodeMCU Pin out Diagram

Our project makes the use of ESP8266 WI-FI Module and 9 digital pins (GPIO) as explained below.

**Digital Pin D0 (GPIO16)**

Here the Digital Pin D0 is used for giving the trigger input pulse to the **Ultrasonic Sensor** at the entry.

**Digital Pin D1 (GPIO05)**

Here the Digital Pin D1 is used for taking the echo output pulse from the **Ultrasonic Sensor** at the entry.

**Digital Pin D2 (GPIO04)**

Here the Digital Pin D2 is used for giving the trigger input pulse to the **Ultrasonic Sensor** at the parking slot.

**Digital Pin D3 (GPIO00)**

Here the Digital Pin D3 is used for taking the echo output pulse from the **Ultrasonic Sensor** at the parking lot.

**Digital Pin D4 (GPIO02)**

Here the Digital Pin D4 is used for giving the trigger input pulse to the **Ultrasonic Sensor** at the exit.

**Digital Pin D5 (GPIO14)**

Here the Digital Pin D5 is used for taking the echo output pulse from the **Ultrasonic Sensor** at the exit.

**Digital Pin D6 (GPIO12)**

Here the Digital Pin D6 is used for giving the PWM pulse to the **Servo Motor** at the entry.

**Digital Pin D7 (GPIO13)**

Here the Digital Pin D7 is used for giving the PWM pulse to the **Servo Motor** at the entry.

**Digital Pin D8 (GPIO15)**

Here the Digital Pin D8 is connected to the long leg of the **LED**.

**Vin**

The positive terminal of external power supply is connected to this pin for powering the NodeMCU.

## **Power Management for the circuit components**

### Managing Power for NodeMCU

The NodeMCU is capable of operating only when an input voltage of 3.3V to 5V is given. It is tolerable to up to 10V. We have given the power to NodeMCU through it’s Vin and Gnd pins.

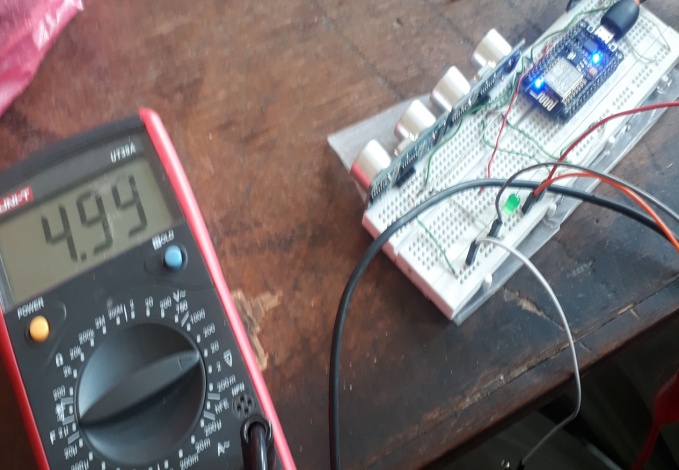


Figure 5‑2: Voltage Drop Across NodeMCU

### Managing Power for Ultrasonic Sensors

The Ultrasonic Sensor is compatible with an input voltage of 3.3V to 5V and its minimum current requirement is **15mA**.The NodeMCU is capable of providing the voltage and current that is required for the single ultrasonic sensor but in our project we have used three ultrasonic sensors. We have used an external power supply of **5V** to power the Ultrasonic Sensor.

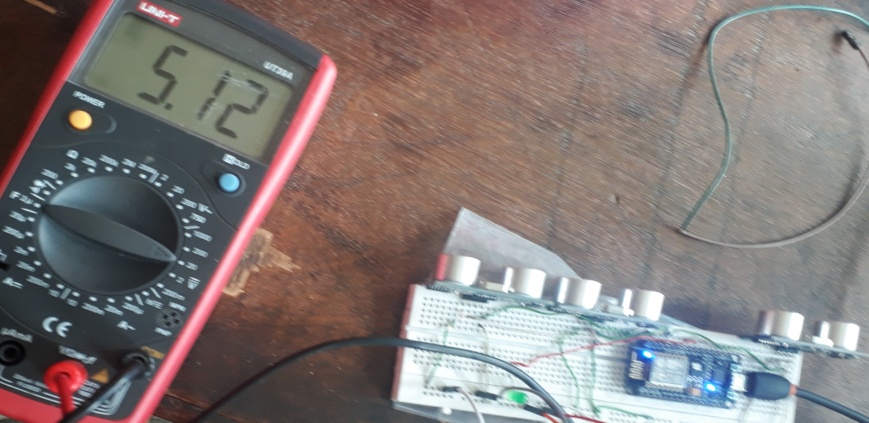


Figure 5‑3: Voltage Drop Across Ultrasonic Sensors

### Managing Power for Servo motor

The Servo is capable to operate only when an input voltage of 4.8V to 6V is given and **PWM** signal given to it ranges between the frequency of 50 and 60 HZ. Its minimum current requirement is **10mA**.The current requirement for maximum rotation of the servo arm is **200mA**.The NodeMCU is not capable of providing the voltage and current that is required for the servo-motor, hence, we have used an external power supply of **5V** and **2A**.Here it is seen that initial voltage of 4.16V drops to 2.60V when servo arm rotates to its maximum as shown in figures below.

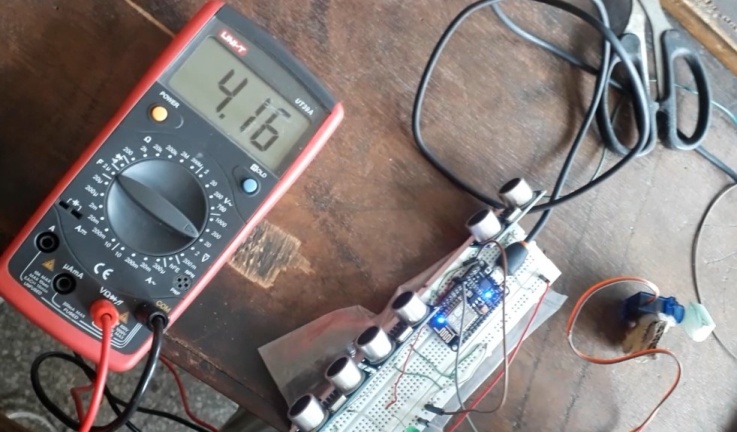


Figure 5‑4: Initial voltage drop across Servo



Figure 5‑5: Voltage Drop Across Servo for Maximum Rotation

### Management of Power Supply for LED

It is noticed that during the time when LED is off the voltage required is 0V and the voltage at time of LED on is **2.52V** as shown in the figure below.It is connected in parallel to power supply of 5V and 2A which is also given to Servo Motor and the ultrasonic sensor.

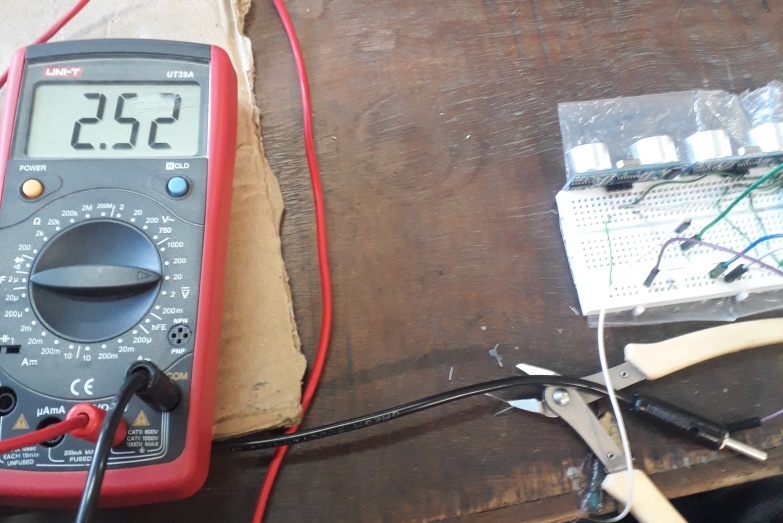


Figure 5‑6: Multimeter reading when LED is on

## **Creating an App for control**

A control of the barrier is the main theme of our project for this we have developed app through MIT App Inventor.

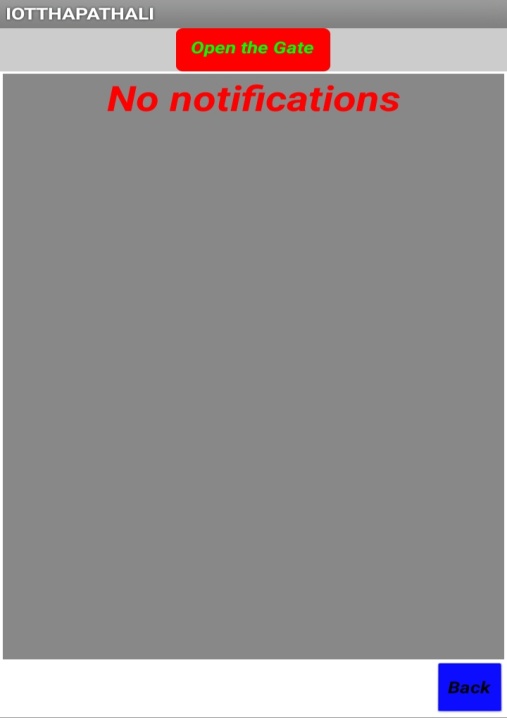


Figure 5‑7: User Interface Screen for Mobile Application

For controlling the barrier (Servo) in the user interface we have included one button for barrier open .

Here since the NodeMCU is being used as the web server the Url of the NodeMCU is specified in the backend of the application for performing the desired task and based on the request made by the client(user) the application performs the desired task i.e. control of the barrier.

For programming NodeMCU as a web server we have included certain part of web development, for which we have used Hypertext Markup Language (HTML).

## **HTTP (Hypertext Transfer Protocol)**

The Hypertext Transfer Protocol (HTTP) is a standard application layer protocol that functions as a request-response protocol between server and client.

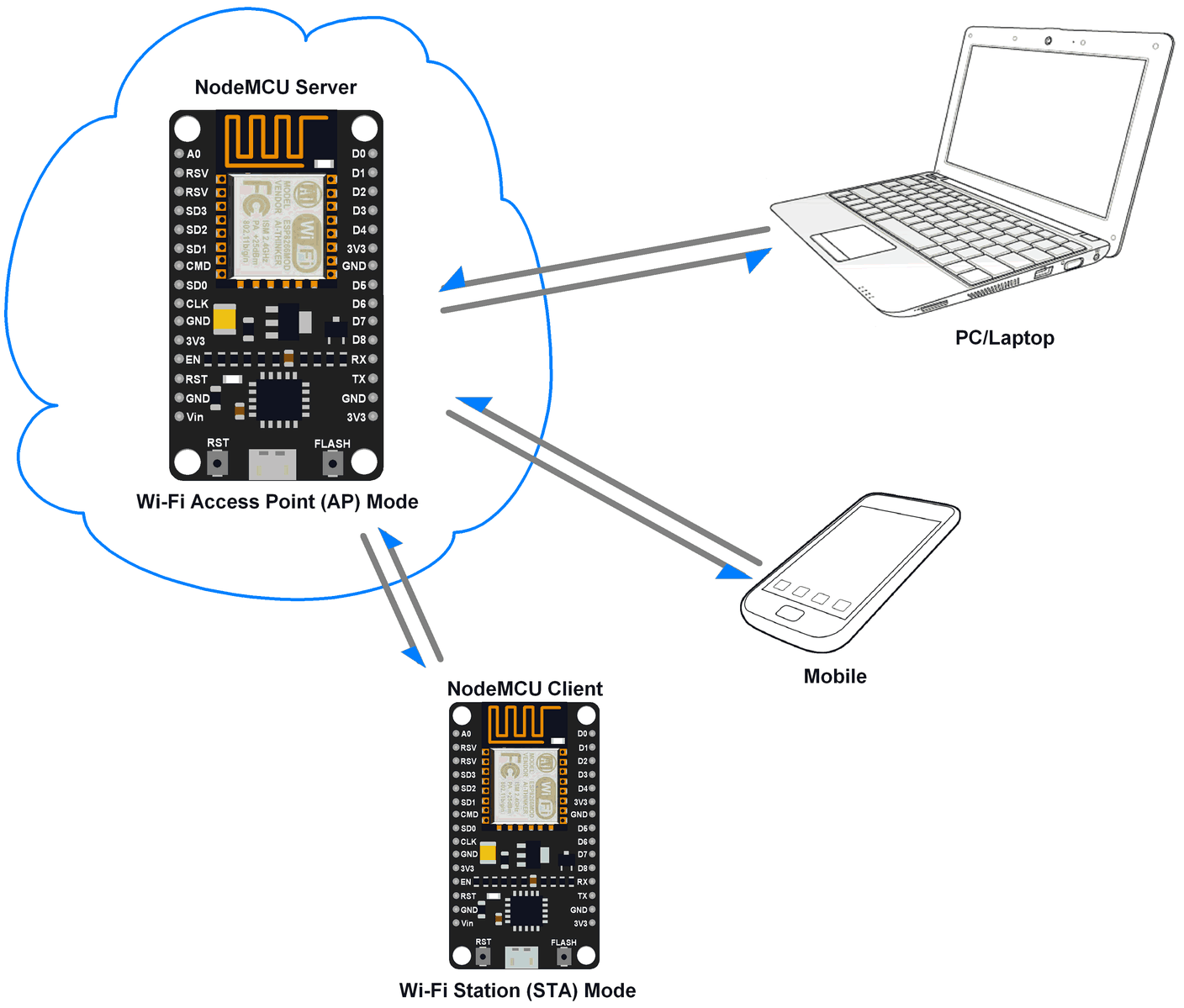


Figure 5‑8: NodeMCU as HTTP Server using Wi-Fi AP mode

Through the HTTP protocol, resources are exchanged between client devices and servers over the internet. Client devices send requests to servers for the resources needed to load a web page; the servers send responses back to the client to fulfill the requests.

To make the request, the client uses components of a URL (Uniform Resource Locator), which includes the information needed to access the resource. The components of a URL explain URLs.

A correctly composed HTTP request contains the following elements:

1. A request line.
2. A series of HTTP headers, or header fields.
3. A message body, if needed.

Here in backend of our application we have specified URL as **http://192.168.4.1/ON?** for a particular action.

Now in the web server the corresponding request is read as GET /ON HTTP/1.1

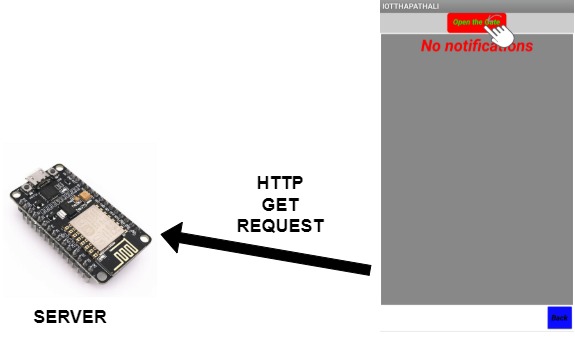


Figure 5‑9: HTTP GET Request Method

The **HTTP GET method** requests a representation of the specified resource. Requests using GET should only be used to request data (they shouldn't include data).

The HTTP headers for a client's request contain information that a server can use to decide how to respond to the request. The HTTP header used here is HTTP/1.1.

## **HTML(Hypertext Markup Language)**

HTML (Hypertext Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. HTML tags are like keywords which define that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

<div>

The tag defines a division or a section in HTML document.

<body>

This tag defines the document's body.

<form id>

Specifies the form of element the <input> element belongs to.

<input >

The class attribute assigns one or more classnames to the <input> tag.

<value>

Defines value corresponding to request by client.

<br>

This tag is used for inserting line break.

## **Use of Libraries**

To supplement our programming, we used the in-built libraries.

E.g. ESP8266WiFi Library, Servo Library

To use this library we used

#include <ESP8266WiFi.h>, #include <Servo.h>

These are the functions in ESP8266WiFi library:

**WiFiServer server(80);**

The server responds to clients (in this case - application) on port 80, which is a standard port web browsers talk to web servers.

**WiFi.softAP(AP\_NameChar, WiFiPassword);**

Enables to use NodeMCU as a Software Enabled Access Point.

AP\_NameChar= Used to set name of Wi-Fi Access Point

Password =Used to set Password of Wi-Fi Access point

**server.begin();**

Initialize server for listening to client request

**WiFiClient client = server.available();**

Gets a client that is connected to the server and has data for reading.

**request = client.readStringUntil('\r')**

Read the first line of the request.

These are the functions in Servo library:

**Servo x;**

x = the name that you'll use when calling the servo to do something.

**x.attach(y)**

x = the name that you'll use when calling the servo to do something.

y = the pin to attach servo.

**x.write(pos);**

x = the name that you'll use when calling the servo to do something.

pos = the angle that you want your servo motor to go. (it has to be something between 0 and 180)

Other than that, we installed a CH340 Driver, CH340 chip is used by a number of NodeMCU compatible boards to provide USB connectivity.

# RESULT AND ANALYSIS

## **Ultrasonic Sensor**

### Output of Ultrasonic Sensor

For our project, we needed to use the output of Ultrasonic Sensor. So we interfaced Ultrasonic Sensor with NodeMCU whose output when obstacle is near is given as below:



Figure 6‑1: Placement of the ultrasonic sensor

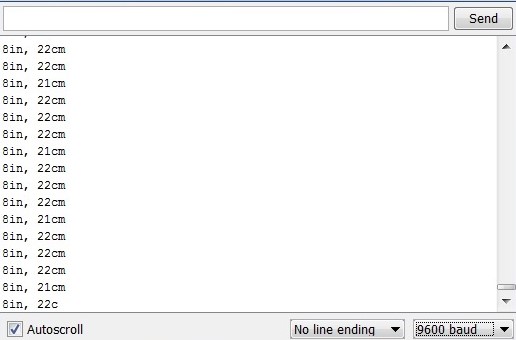


Figure 6‑2: Output of the Ultrasonic Sensor

### Analysis

The ultrasonic sensor uses trigger pin for giving input trigger pulse and draws echo output pulse from the echo pin

**Trigger Signal**

Here, the trigger input signal given is 10Microsecond signal. Its corresponding waveform as seen in the oscilloscope is as shown below.

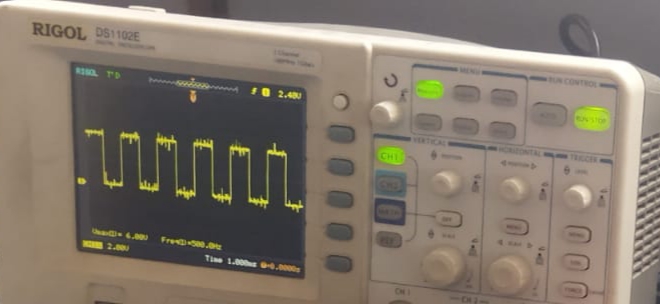
s

Figure 6‑3: Waveform of the Trigger Input signal

**Echo Signal**

The corresponding Echo signal for a particular instance of the Trigger Signal is as shown below.

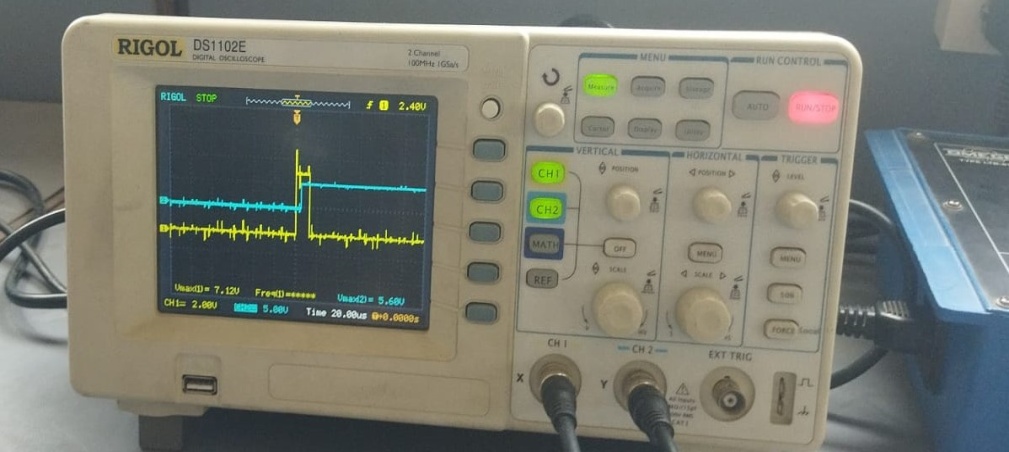


Figure 6‑4: Waveform of Echo Output Signal for a Trigger Input

Here the change in echo pulse indicates that an object is detected. Total time of object detection can be noted from this waveform. Now since we have calculated the time duration’s’ during which echo is high and the velocity of ultrasonic wave ’v’ i.e. 340m/s. now the total distance of separation between the object and the sensor is calculated as

Distance (d) = (velocity (v)\*time duration (t))/2;

## **Servo Motor**

### Output of Servo motor

Currently, the servo motor rotates in between 0 and 180 degree according to vehicle detected or in accordance with control from the mobile application

### Analysis

Servo motor at the entry is to be automated in accordance with presence of vehicle at parking.For the servo motor at exit it is controlled through the mobile application here the NodeMCU is being programmed as a web server whereby it receives the HTTP GET request from the client through the mobile application and responds accordingly as shown in the figure below.

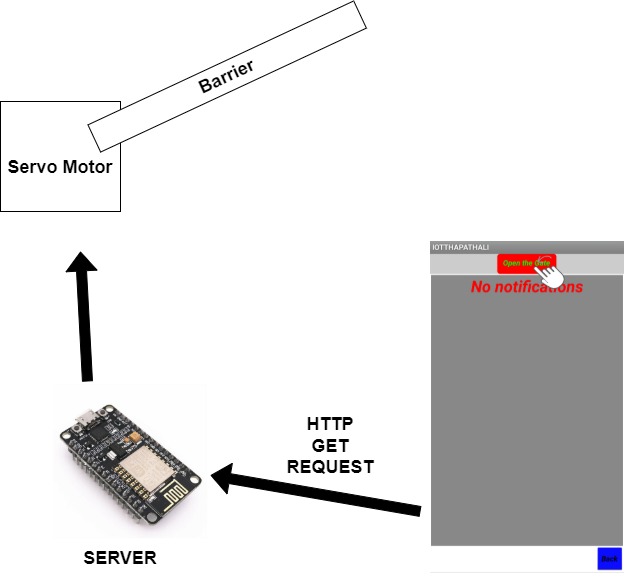


Figure 6‑5: Path for data flow for servo at exit

Servo motors are controlled by via a pulse width modulated (PWM) signal. Servo motors usually have three wires: power, ground and the control signal.Most servos fixedly rotate between 0° and 180° - starting and ending at fixed points relative to the motor.

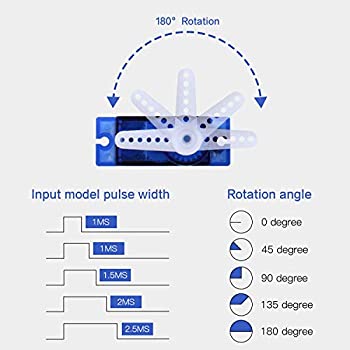


Figure 6‑6: PWM pulse for servo rotation

They accept pulses within a fixed range commonly between 500 and 2500us as shown in Figure 6-5. To put it all together, say we send a 500us width pulse to a servo accepting pulses between 500us and 2500us. The servo will rotate its arm to the 0° position in response - no matter which position the arm was in before. It will respond with appropriate increments when the pulse width is increased up until 2500us, then it will stop moving.

When the servo is receiving signals continuously, it will apply force to attempt to stay in the position that is being signaled. When the servo is unpowered and sent no signals, it won’t actively try to restore position. Manually moving the servo arm is possible when unpowered, but it should not be done as it can damage the servo.

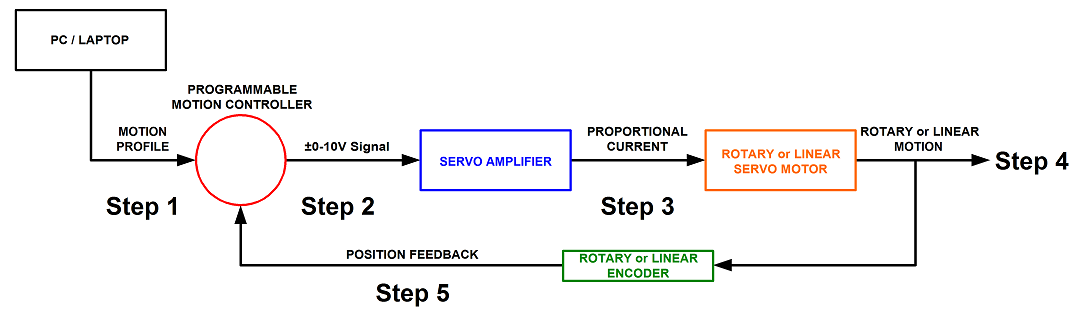


Figure 6‑7: Closed loop servo motor working mechanism

A Servo motor SG90 is a closed-loop servomechanism that uses position feedback to control its motion and final position. The input to its control is a signal (either analogue or digital) representing the position commanded for the output shaft.

The Servo motor SG90 is paired with some type of position encoder to provide position and speed feedback. In the simplest case, only the position is measured. The measured position of the output is compared to the command position, the external input to the controller.

If the output position differs from that required, an error signal is generated which then causes the motor to rotate in either direction, as needed to bring the output shaft to the appropriate position. As the positions approach, the error signal reduces to zero and the motor stops.

The very simplest servomotors use position-only sensing via a potentiometer and bang-bang control of their motor; the motor always rotates at full speed (or is stopped).

## **Error Analysis**

### Providing power to Servo

Initially we tried to power the entire system by using the NodeMCU.but later on we found that maximum voltage that can be supplied by NodeMCU is **3.3V** which is insufficient to drive the servo motor as it requires the potential difference across it to be **4.8-6V**,also at maximum rotation condition of servo arm it was noted that maximum current required was roughly equal to **200mA** but the maximum current that can be supplied by NodeMCU is **12mA**.

### Providing power to the Ultrasonic Sensor

One Ultrasonic sensor can be interfaced through the NodeMCU but in our project we have used three Ultrasonic sensors. Hence, we had to provide an external power for it’s smooth operation but the ultrasonic sensor didn’t respond accordingly. Later on we found out that due to excessive changes in the power supply during its testing, the receiver of ultrasonic sensor wasn’t receiving the reflected ultrasonic waves as shown in figure below.

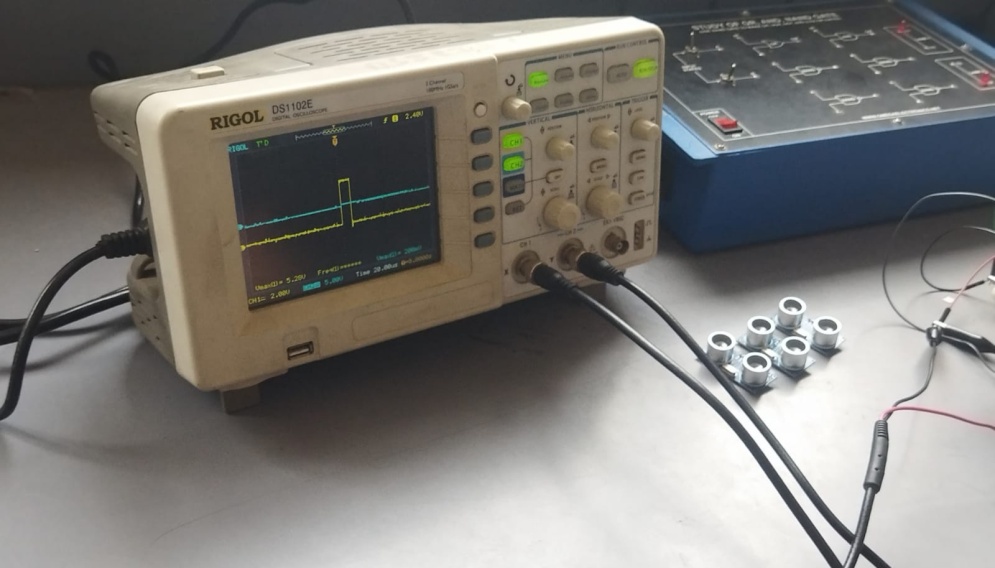


Figure 6‑8: No changes in the received echo signal for a pulse of Trigger signal

### Assembling the circuit components together

Initially we tested the circuit components, but while assembling them together we faced problem of power management. The servo wasn’t responding in accordance to its program. The ultrasonic sensor didn’t detect the obstacle, also due to insufficient power supply the NodeMCU kept turning off.

## **Simulation of a parking system prototype**

Though it was difficult for us to implement a parking system prototype but we tried simulation of an actual parking system. First of all we need to understand why is simulation required? Simulation allows you to explore ‘what if’ questions and scenarios without having to experiment on the system itself & helps you to gain insight into which variables are most important to system performance. We were able to test what if we used 3 or more sensors to check the spatial position of incoming car without having to actually buy more sensors.

Prototyping can be cost-prohibitive in complex systems and can significantly delay projects due to timing and supply chain. Simulation and modeling offer a viable alternative for running experiments, clinical studies, and stressing the design to limit loads in a virtual environment, all with relatively accurate results. Some Electronic equipment are expensive while some are hard to fetch, but simulation has helped to keep the project moving forward in designing code and architecture section while hardware section is being delayed.

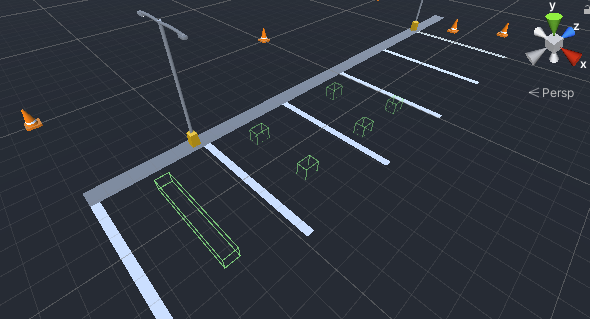


Figure 6‑9: Testing one or more simulated sensors

1. Process

The data can be taken from real sensor and sent to the simulated environment to see how it would look like or data can also taken from simulated sensors to do anything you intended to do quickly without any setup. Simulation was done in Unity Engine . Uduino Library was used to communicate between NodeMCU and Unity. 3 sensors (entry gate, exit gate, parking spot sensor) take data from real world and send it to Unity for better visualization. The 3d models (cars) are positioned based on data from those sensors and for gate open/close mechanism, positive signal is taken when user presses button in our app.

As this is a simulation, we made a simple ticket system, parking slot display system and top camera view to easily identify empty parking spaces which is display in a large billboard.



Figure 6‑10: Entry Position of Vehicle

A User/Vehicle comes to the gate and sensor detects the car. Currently in real prototype, no vehicle information is taken.

In the simulation, Vehicle information like vehicle registration number, parking type ( if its a car, Truck-heavy, or a bike). A ticket is issued in the account of that registration number . And the gate is opened to let the vehicle in. Then, the parking billboard updates the number of available slots available which can be seen in fig as green billboard. This feature was tested to know if incoming cars would get required info before even entering the parking area. More feedbacks are needed to know if this feature is valuable or not.

The 3 buttons seen in bottom left of Figure 6-10 is solely made for the purpose of simulating/creating new vehicles to enter the parking area. The large billboard on the right shows a live top down view of the entire parking area. This is also a feature that is made to check how people respond to it and it’s value to the project.

Following screenshot shows the number of variables that a ticket contains when a vehicle enters parking lot.

Exit-time (outTime in Figure), Payment method, Price are fields that are filled out only when the vehicle tries to exit parking lot.

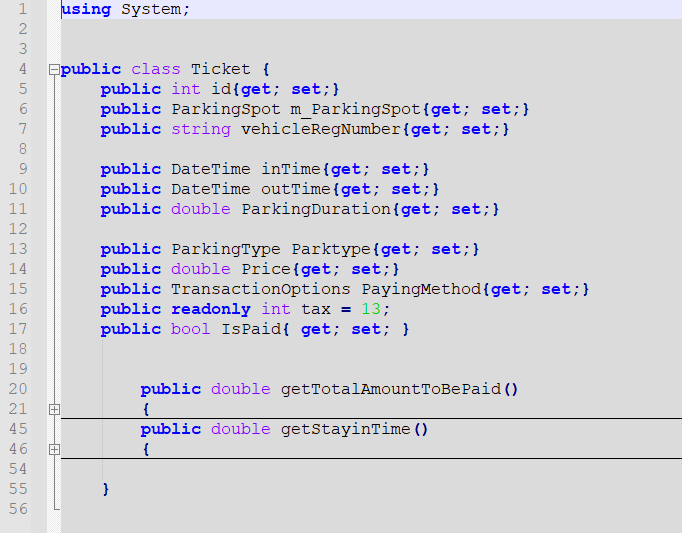


Figure 6‑11: Ticket info



Figure 6‑12: Truck going to park

In Fig 6-9 we see how both billboards updates its display when there are vehicles in the area.

When the car returns to exit gate, a ticket is provided to vehicle driver . The price of parking is determined by the duration and type of vehicle (truck , car , bike) with added tax. The driver must pay else the exit gate won’t be opened.

The simulation can be done without any hardware input as well. In this case, we manually control the car in the environment and check various cases to see if the parking system works as intended.

1. RESULTS:

A working simulated environment was made that can take data from real sensors as well as create its own dummy data to easily test various normal to extreme cases which would be usually be difficult in real-world scenarios.

# FUTURE ENHANCEMENT

We have presented a prototype for our project, and thus we have room for further improvements in our project. The components used in the project have their limitations, which has affected the efficient and effective functioning of the project as a whole. Servo Motor is the main component required for the control of the barrier. Servo Motor showed a lot of problems due to its large current requirement, and we have aimed to find a more stable power source like SMPS (Switch Mode Power Supply) The efficiency is as high as 80 to 90%.As it has less heat generation; less power wastage. The device is compact and small in size. It has provision for providing the required number of voltages.

The use of ultrasonic sensor seems to be inefficient as it is unable to distinguish the obstacle. In-order to recognize the vehicle using the ultrasonic sensor we could use it for UV radar so that the gate is automated only when there is an obstacle in the specified radar. Camera Module can be used for more accurate vehicle detection. RFID card reader is also the other alternative.

A major feature which we had to cut down due to time constraint was connecting our project to shopping mall, offiices. The inclusion of this feature in our project would enable the parking management to effectively manage parking in their organization We also can add a feature which sends notification alert message to the management in case there is vehicle at entry and exit and display the available and parked slots.

The system we designed involves multiple sensors and peripherals interfaced to a microcontroller unit, and the entire system is soldered onto a matrix board, thus the form factor of the project is quite uncomfortable for daily use. To further improve on this, we can remove the use of NodeMCU in favor of a smaller microcontroller and a separate Wi-Fi module like nRF24L01, and reduce the size to that of a wearable device, which would make the project more usable in daily lives.

These are some of the enhancements and features we can add to improve our project

# CONCLUSION

Automatic Vehicle Barrier is an IOT-based system which is used to effectively manage the vehicle barrier. The project involves two sections; hardware and software sections. The software section involves a NodeMCU server for retrieving data from the hardware, and an MIT app, which acts as a frontend for managing barrier at exit.

The interface along with the output received from the project justifies the usefulness of the project. Being an IOT-based system, whereby NodeMCU is used as a Wi-Fi hotspot it makes it easier for an organization to maintain authentication for barrier control through the Wi-Fi password.

This project has been a great learning experience for us. We learned many things about hardware and embedded systems, and about reducing hardware complexity, and using less hardware as possible to achieve our required task. The project has been helpful in teaching us the basics of project management and also making us use design patterns in software development, which enabled us to develop our app more efficiently and effectively.

Thus, the output we obtained from our project were desirable, which proved the usefulness of our project.

# APPENDICES

## **APPENDIX A: PROJECT BUDGET**

The estimated cost for project completion is extremely low as the circuit can be developed using the breadboard using simple elements like sensors motor and programming can be done in NodeMCU.

Table 9‑1: Budget Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **Components Used** | **No of Components** | **Price** |
| 1 | NodeMCU | 1 | 550 |
| 2 | Ultrasonic Sensor Module | 3 | 600 |
| 3 | Servo motor SG90 | 2 | 400 |
| 4 | Breadboard | 1 | 125 |
| 5 | Miscellaneous | | 1,000 |
| Total | | | 2675 |

## **APPENDIX B: PROJECT SCHEDULE**

Table 9‑2: Gantt chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Task /Timeline | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 | Week 12 |
| Brain Storming and Topic Discussion |  |  |  |  |  |  |  |  |  |  |  |  | |
| Familiarization of tools |  |  |  |  |  |  |  |  |  |  |  |  | |
| Designing |  |  |  |  |  |  |  |  |  |  |  |  | |
| Assembling hardware components together |  |  |  |  |  |  |  |  |  |  |  |  | |
| Coding Microcontroller |  |  |  |  |  |  |  |  |  |  |  |  | |
| Testing and finalizing the project |  |  |  |  |  |  |  |  |  |  |  |  | |
| Documentation |  |  |  |  |  |  |  |  |  |  |  |  | |

## **APPENDIX C: CIRCUIT DESIGN DIAGRAMS**

The Circuit Diagram of our project is as below:

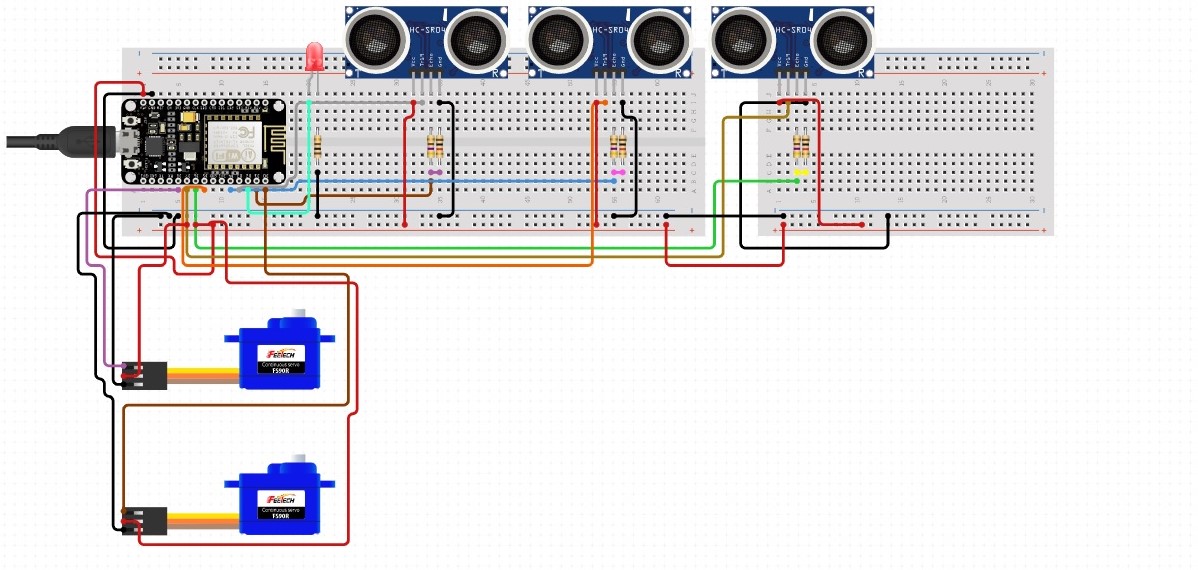


Figure 8‑1: Circuit Diagram

**APPENDIX D: MODULE SPECIFICATION**

Table 9‑3: NodeMCU specifications

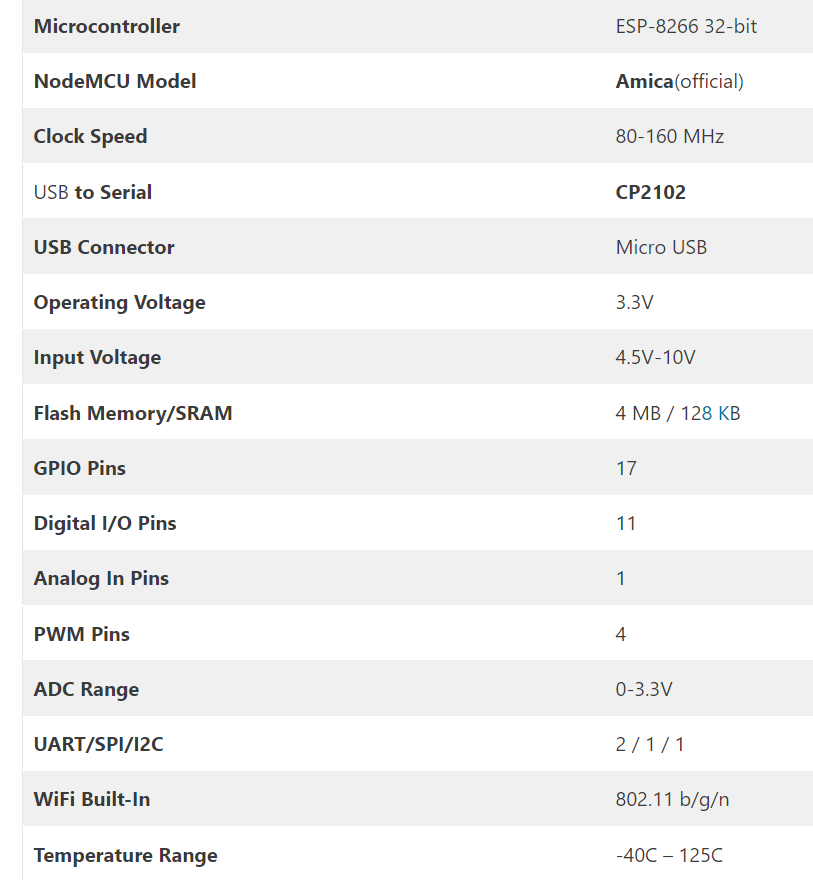


Table 9‑4: Pin Description of NodeMCU

|  |  |  |
| --- | --- | --- |
| Pin Category | Name | Description |
| Power | Micro-USB,3.3V,GND,Vin | **Micro-USB**: NodeMCU can be powered through  the USB port  **3.3V:** Regulated 3.3V can be supplied to this pin to power the board  **GND:** Ground pins  **Vin:**External Power Supply |
| Control Pins | EN,RST | The pin and the button resets the microcontroller |
| Analog Pins | A0 | Used to measure analog voltage in the range of 0-3.3V |
| GPIO Pins | GPIO1 to GPIO16 | NodeMCU has 16 general purpose input-output pins on its board |
| SPI Pins | SD1,CMD,SD0,CLK | NodeMCU has four pins available for SPI communication. |
| UART Pins | TXD0,RXD0,TXD2,RXD2 | NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program. |
| I2C Pins |  | D1,D2 is used for connecting SCL,SDA pins of OI2C devices |

Table 9‑5: Ultrasonic Sensor Pin Configuration

|  |  |
| --- | --- |
| **VCC** | For Power supply |
| **Trigger** | Giving the trigger input |
| **Echo** | Taking the echo output |
| **Gnd** | Used for ground |

Table 9‑6: Electrical Specification of Ultrasonic Sensor

|  |  |
| --- | --- |
| **Working Voltage** | DC 5 V |
| **Working Current** | 15mA |
| **Working Frequency** | 40Hz |
| **Max Range** | 4m |
| **Min Range** | 2 cm |
| **MeasuringAngle** | 15 degree |
| **Trigger Input Signal** | 10uS TTL pulse |
| **Echo Output Signal** | Input TTL lever signal and the range in proportion |
| **Dimension** | 45\*20\*15mm |

Table 9‑7: Pin Configuration of Servo Motor

|  |  |
| --- | --- |
| **PWM** | Generation of required pulse |
| **VCC** | Power Supply |
| **GND** | Used as Ground |

Table 9‑8: Electrical Specification of Servo Motor

|  |  |
| --- | --- |
| **Speed(sec)** | 0.1 |
| **Torque(kg-cm)** | 2.5 |
| **Weight(g)** | 14.7 |
| **Voltage** | 4.8-6 |

**References**

|  |  |
| --- | --- |
| [1] | Pranav Chippalkatti, Ganesh Kadam, and Vrushali Ichake “I-SPARK: IOT based Smart Parking System” in International Conference on Advances in Communication and Computing Technology (ICACCT), 2018, pp. 473-477 |
| [2] | Cristian Roman , Ruizhi Liao, and Peter Ball “Detecting On-Street Parking Spaces in Smart Cities: Performance Evaluation of Fixed and Mobile Sensing Systems” in IEEE Transactions on intelligent Transportation Systems,2018, vol. 19, pp. 2234-2245 |
| [3] | Mohammed Y Aalsalem, WazirZada Khan, Khalid Mohammed Dhabbah“An Automated Vehicle Parking Monitoring and Management System Using ANPR Cameras” July 1-3, 2015 ICACT2015, Faculty of Computer Science & Information System, Jazan University, Kingdom of Saudi Arabia 2015 |
| [4] | Rachapol Lookmuang, K. N. (2018).Smart Parking Using IOT Technology., IEEE, 6. |
| [5] | Pampa Sadhukhan “An IOT-based E-Parking System for Smart Cities” in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 1062-1066 |
| [6] | Ming Wang, Huifang Dong, Xu Li, Liangliang Song, and Dandan Pang “A Novel Parking System Designed for Smart Cities” in Chinese Automation Congress, 2017, pp. 3429-3434. |
| [7] | Anak Trivedi, Dr. Mandalapu Sarada Devi, and Dave Dhara “OpenCV and Matlab based Car Parking System Modulefor Smart City using Circle Hough Transform” in International Conference on Energy, Communication, Data Analytics and Soft Computing (lCECDS-2017), 2017, pp. 2461-2464. |
| [8] | Ilhan Aydin, Mehmet Karakose, and Ebru Karakose “A Navigation and Reservation Based Smart Parking Platform Using Genetic Optimization for Smart Cities” in 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG), 2017, pp. 120-124 |