

CONSTRUCTION OF A GESTURE- CONTROLLED ROBOTIC CAR USING NODEMCU

By

OKUBOYEJO, ADEBOLANLE OLUWASEYI

17010302001

**A PROJECT SUBMITTED TO THE DEPARTMENT OF PHYSICS, COLLEGE OF BASIC
AND APPLIED SCIENCES.**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF
DEGREE OF BACHELOR OF SCIENCE IN PHYSICS.**

May 2021.

DECLARATION

I hereby declare that this project has been written by me and is a record of my own research work. It has not been presented in any previous application for a higher degree of this or any other University. All citations and sources of information are clearly acknowledged by means of reference.

.....

OKUBOYEJO, ADEBOLANLE OLUWASEYI

.....

Date

CERTIFICATION

This is to certify that the content of this project entitled ‘**CONSTRUCTION OF A GESTURE- CONTROLLED ROBOTIC CAR USING NODEMCU**’ was prepared and submitted by **OKUBOYEJO, ADEBOLANLE OLUWASEYI (MATRIC NO. 17010302001)** in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE IN PHYSICS**.

The original research work was carried out by him under my supervision and is hereby accepted.

_____ (Signature and Date)

Dr. R.O. Akande

Supervisor

_____ (Signature and Date)

Dr. S.O Kareem

Head of Department

DEDICATION

I dedicate this project to the almighty God for His loving kindness and protection over my life, also for giving me the inspiration for this project and the grace for its success, may His name be highly exalted.

I also dedicate this project to all my family members especially my mother for her unending support both financially and spiritually towards the success of this project.

Finally, I dedicate this project to the Department of Physics in order to inspire other physics student to work on more project like this, as the first student to have ever done a project like this in the Department of Physics, Mountain Top University.

ACKNOWLEDGEMENTS

I thank God over the life of my mother Mrs. Abiola Oboh, for her financial support towards this project and for believing in me always. I also thank my aunt Mrs Talabi nee Okuboyejo for her words of encouragements and her support towards the completion of my undergraduate programme. I appreciate the support of every of my family member.

I would like to extend my gratitude to the Chancellor, Dr D.K. Olukoya, the Vice-Chancellor, Prof E. Ayolabi, and the Dean of the College of Basic and Applied Sciences, Dr. Ofudje, for their persistent endeavors towards our education.

I also extend my sincere thanks to the Head of the Department, Physics, Dr. S.O. Kareem, for the constant support.

It is indeed a pleasure to mention about Dr. R.O. Akande, project supervisor and Mr. F. Olawole, project guide who have always been patient enough to make the complexities of the project and relentlessly supported me throughout the project.

My thanks to the teaching and non-teaching staff of the Physics Department of Mountain Top University, who provided necessary resources for this project.

LIST OF TABLES

Table No.	Title	Page No.
2.1	NodeMCU Development Board Pinout Configuration.....	7
2.2	Details of the pins of L298N.....	9

LIST OF FIGURES

Figure No.	Title	Page no.
2.1	A NodeMCU Device.....	7
2.2	A L298N Motor driver and its pinout	9
2.3	A L298N Module Pinout Configuration	9
2.4	A Simple DC Motor	10
3.1	Illustration of the flow of program	13
3.2	Attachment of the motors and wheels to the chassis	14
4.1	Circuit Diagram.....	15
4.2	Image of the robotic car completely connected.....	16
4.3	Image of the Arduino IDE.....	17
4.4	Image of successful connection of the NodeMCU to hotspot.....	17
4.5	Image of the Gesture control app.....	18

ABBREVIATIONS

MCU – Micro-Controller Unit

RTOS – Real Time Operating System

UART – Universal Asynchronous Receiver-Transmitter

Wi-Fi – Wireless Fidelity

SPI – Serial Peripheral Interface

HMI – Human Machine Interface

DC – Direct Current

RC – Radio Controlled

IDE – Independent Development Environment

MIT – Massachusetts Institute of Technology

GSM – Global Service for Mobile

TABLE OF CONTENTS

	Page No.
DECLARATION.....	i
CERTIFICATION.....	ii
DEDICATION.....	iii
ACKNOWLEDGEMENT.....	iv
LIST OF TABLES.....	v
LIST OF FIGURES.....	vi
ABBREVIATIONS.....	vii
ABSTRACT.....	xi
CHAPTER 1.....	1
1.0 INTRODUCTION.....	1
1.1 Background of study.....	1
1.2 Problem Definition.....	1
1.3 Aim.....	2
1.4 Objectives.....	2
1.5 Applications.....	2
CHAPTER 2.....	3
2.0 LITERATURE REVIEW.....	3
2.1 Rope and Strings.....	3
2.2 Strings and Winder.....	3
2.3 Strings and Wheels.....	4
2.4 Battery-Powered.....	4
2.5 RC Vehicles.....	4

2.6 Gesture-Controlled Vehicles using Arduino.....	5
2.7 Gesture-Controlled Vehicles using Node MCU.....	6
2.8 Other Complimentary parts.....	6
2.8.1 ESP8226 Node MCU.....	6
2.8.2 L328N Motor Driver.....	8
2.8.3 DC Motors.....	10
CHAPTER 3.....	11
3.0 MATERIALS AND METHODS.....	11
3.1 Project Design.....	11
3.2 Methodology.....	12
3.2.1 Gyroscope's Data.....	12
3.2.2 Wi-Fi Network.....	12
3.2.3 Signal processing.....	12
3.3 Components Assembly.....	13
3.3.1 Chassis, Motors and Wheels.....	13
3.3.2 Powering the Motors.....	13
3.3.3 Powering the L298N Motor Driver.....	13
3.3.4 Powering the Node MCU.....	14
3.3.5 Remotely Controlling the L298N Motor Driver.....	14
CHAPTER 4.....	15
4.0 IMPLEMENTATION AND RESULT.....	15
4.1 Circuit Engineering.....	15
4.1.1 Description of Circuit.....	15
4.2 Coding of the Node MCU.....	16

4.3 Gesture control application development.....	18
4.4 Result.....	18
CHAPTER 5.....	19
5.0 CONCLUSION AND RECOMMENDATION.....	19
5.1 Conclusion.....	19
5.2 Recommendations.....	19
REFERENCES.....	20
APPENDIX.....	22

ABSTRACT

Robots are now an essential part in the modern society. Nowadays robots are everywhere, their application are far reaching and can encompass a variety of different machines with different purposes. The major goal of the project work is to control the movement of a robot through gestures. The system is made up of two major components: An android device with an app capable of reading the gyroscope sensor embedded in the device, which is controlled by tilting the device. The tilting of the device sends data signals through a Wi-Fi network. This signal is processed with the assistance of the NodeMCU, which instructs the robot to move in the specified direction by sending voltages to the motor driver. The essential operation of the robot is the transmission of data signals from an android device, readings, to the NodeMCU board installed on the robot. The software compiled in the NodeMCU executes according to that value, causing the robot to work properly while connected to the hotspot of the android device. The gyroscope sensor operates on two axis. One axis controls the forward or reverse speed, while the other controls the turning mechanism. Wi-Fi controlled robots can be very useful in rescuing individuals from collapsed buildings due to natural disasters, and can also be implemented in almost any sector in the world.

CHAPTER 1

INTRODUCTION

1.1 Background of study

Today, robotics is one of the most sophisticated fields of technology. A robot is often an electromechanical machine that can execute tasks automatically. Robots can be fully autonomous or semi-autonomous. An autonomous robot is not directed by people and acts on its own pre-programmed decisions based on detecting its surroundings. The bulk of industrial robots are self-contained because they must work at fast speeds and with excellent accuracy. However, semi-autonomous robots, also known as human-controlled robots, operate under the control of humans through the use of a control system as an interface. Voice recognition, touch-controlled, and gesture-controlled systems are some of the most widely utilized control methods.

Historically, significant attempts have been made to create intelligent and natural interfaces between people and computer-based systems that are based on human gestures. Gestures can come from any body action or state, although they most often come from the face or hand. Gesture recognition is a method for a computer to comprehend human body language. Gestures offer both people and computers with an intuitive interface. As a result, such gesture-based interfaces can not only replace standard interface devices, but also be used to enhance their capabilities.

1.2 Problem Definition

In the early days, the only method to interact with a robot was to program, which took a lot of time and effort. The gesture-controlled robot was developed as science and robotics progressed. This robot will be controlled with a gyroscope sensor supported android device, the tilting of the device will be utilized as input for the robot's movement, which is the main idea behind this project. In this project, a system (robotic vehicle) will be developed which can identify human interaction and perform the objectives that have been set to it. The robotic vehicle will receive signals sent by the android device which will control its movement.

1.3 Aim

The project's aim is to make a robotic car and control the motions of the car with an android device which has a gyroscope sensor to execute predetermined gestures.

1.4 Objectives

The goal of this project is to create a robotic battery-powered automobile that can be operated wirelessly by gesture. While doing so, we will also need to:

- Analyze the energy consumption rate of the four motors that drive the wheels over a specified period of time. The motors that use the least amount of energy will be considered.
- Research the most efficient, low-cost, and high-memory-capacity electronic board.
- Research the most effective, low-cost, long-range, energy-efficient communication technologies.
- Analyze the sensors that will be on board to allow the robotic automobile to be easily maneuvered. For example, we may use similar sensors to pre-determine the terrain in which the robot would operate.

1.5 APPLICATIONS

Gesture control allows handicapped and physically challenged persons to perform tasks such as driving a vehicle and maneuvering a wheelchair with great convenience.

Gestures can be used to control interactions for entertainment purposes, such as gaming, in order to make the game player's experience more interactive and immersive. This is currently being used by the Tech company "*UltiGesture*."

CHAPTER 2

LITERATURE REVIEW

Similar research from the past are discussed in this chapter, and some of the methodologies and working concepts employed are reviewed.

2.1 ROPE AND STRINGS

The early semi-autonomous cars were completely mechanical and were mostly utilized as children's toys. The toy car is comprised of wheels, a string, and a rope and can only move in one direction. When the user pulls back on the rope, tension is created on the string, and when released, the vehicle travels forward at a constant speed that decreases as the tension on the string decreases.

Problems

- The rope becomes slack and may be cut.
- It covers a very small distance.
- It can only travel in one direction.

2.2 STRINGS AND WINDER

Later, modifications were made, and a toy car capable of moving in both directions was developed. This version was similar to the rope and strings model except for the addition of a winder, which was necessary since the rope used in the previous version could only be pulled in one direction and was not particularly strong. The winder is rotated in the opposite direction that the user desires the vehicle to go. The user then releases the winder, and the vehicle goes at a high speed that decreases as the strain on the string decreases and ultimately stops.

Problems

- The winder is easily damaged.
- It only travels a small distance.

2.3 STRINGS AND WHEELS

Further improvements were made to the previously described toy vehicles, and the usage of strings and wheels was discarded. The vehicle was constructed with just wheels and springs linked to the wheels, but previous versions had their string put only at the back wheels, which controlled the vehicle's motions. The springs in this variant were located at the front and rear wheels. The vehicle is dragged backward or pushed forward to travel ahead or backward at a constant speed that decreases as the tension on the string decreases.

Problems

- The wheels deteriorate due to excessive dragging.
- It travels a short distance.

2.4 BATTERY-POWERED

Mechanical toy vehicles became obsolete, and electrical-powered toy vehicles were developed. The car was constructed with DC motors that controlled the wheels, a battery for power, and a switch to turn the vehicle on and off. The motors are connected to the power supply through connecting wires, and the power source is then connected to the switch. When the switch is turned on, the battery gives power to the motors that drive the wheels, allowing the vehicle to continue going ahead. If the vehicle hits a wall, the battery will continue to supply power to the motors until it reaches its limit, at which point it will stop. This depletes the battery.

Problems

- It cannot be used in every environment.
- It is power consuming.
- It drains the battery in a very short period

2.5 RC VEHICLES

A few years later, technology evolved, allowing the use of wireless controllers to control the movement of a vehicle, commonly known as Radio Controlled (RC) vehicles. This device is made up

of two parts: the transmitter and the receiver. The receiver is built within the car. The vehicle is made up of four DC motors, one for each wheel, a receiver, batteries as a power source, a circuit board, and an on/off switch. The controller, or transmitter, contains buttons or joysticks to control the vehicle's motions, a transmitter, batteries to power it up, and a switch to turn it on/off. When a button is pushed, the transmitter sends a certain amount of electrical pulses corresponding to that action via the atmosphere in the form of radio waves. Once the RC vehicle receives the radio waves, the motors start to do the desired action. The power supply provides electricity to all functioning elements, including the motor, and the circuit board interprets the electrical pulses to activate the vehicle.

Problems

- It uses a lot of power.
- The distance between the transmitter and receiver is limited.
- The human-machine interaction is unnatural.

2.6 GESTURE-CONTROLLED VEHICLES USING ARDUINO

This project is an enhancement to the previous model. It is a gesture-controlled robotic vehicle, which means it can travel in different directions simply on hand gestures. This is likewise separated into two parts: a transmitter attached to a wearable hand glove and a receiver installed in the vehicle. An RF transmitter module, an encoder IC, an accelerometer, and batteries serve as the transmitter's power supply. The vehicle is comprised of a microcontroller, a decoder IC, DC motors, batteries as a power supply, and an RF receiver module. The whole circuit that records the user's hand motions is encased in a hand glove. To conceal electrical information, the user wears a hand glove that is portable. The data collected by the sensors embedded in the hand gloves is analyzed and translated into an intelligible format for the robotic car's receiver. Whether the user moves his or her hand or not, the transmitter gadget transmits all of the time. Every time a signal is transmitted, the receiver receives it and transforms it into movement or no movement of the robotic vehicle. This transmitter device will serve as a data transmitter for wireless communication. Once the data is received by the receiver module, which is connected to the microcontroller, it is processed and delivered to the microcontroller. The commands will be deduced by the microcontroller, and the motor drivers will be activated to operate the motor for various tasks on the robot.

2.7 GESTURE-CONTROLLED VEHICLES USING NODE MCU

This project is similar to the previous project because they are both gesture-controlled. The difference between them is the mode of transmission, the previous project operates by sending electrical impulses, while this project operates by sending signals from an android device to a WIFI enabled device. This project seeks to control a robotic car with gesture. The sensors for detecting gesture are Accelerometer, gyroscopes and many others. In this project, we will implement the gyroscope built-into Android devices. This gyroscope will be accessed via an Android application which will be downloaded from the Google Play Store. Any gyroscope, be it the one built-into Android devices or not, measures the rate of rotation in rad/s around a device's x, y, and z axis. It measures angular velocity. The sensor's coordinate system is the same as the one used for the acceleration sensor. Some applications require the feature of tilting the device to interact with their content, a gyroscope sensor will be needed to complete such actions.

Although this is not the first time this project has been undertaken in the globe, many people from various schools and nations have participated in it successfully.

2.8 Other Complimentary Parts

2.8.1 ESP8266 Node MCU

The Node MCU is a micro-controller capable of communicating wirelessly with other devices. The Node MCU ESP8266 development board comes with the ESP-12E module containing the ESP8266 chip which is a Tensilica Xtensa 32-bit LX106 RISC microprocessor. This microprocessor supports RTOS and operates at 80MHz to 160 MHz adjustable clock frequency. NodeMCU has 128 KB RAM and 4MB of Flash memory to store data and programs. It has high processing power with in-built Wi-Fi / Bluetooth. It also has a Deep Sleep Operating features make it ideal for IoT projects. Since an IoT device has to operate 24 hours a day and 7 days a week, this Deep Sleep feature wakes the device whenever there is a command, therefore helps save power consumption. NodeMCU can be powered using a Micro USB jack and VIN pin (External Supply Pin). It supports UART, SPI, and I2C interface.

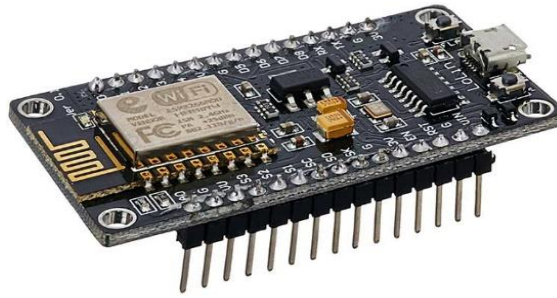


Figure 2.1: A NodeMCU device

Table 2.1: NodeMCU Development Board Pinout Configuration

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	<p>Micro-USB: NodeMCU can be powered through the USB port</p> <p>3.3V: Regulated 3.3V can be supplied to this pin to power the board</p> <p>GND: Ground pins</p> <p>Vin: External Power Supply</p>
Control Pins	EN, RST	The pin and the button resets the microcontroller
Analog Pin	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		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

2.8.2 L298N Motor Driver

The L298N Motor Driver module consists of an L298N Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit 78M05 Voltage regulator will be enabled only when the jumper is placed. When the power supply is less than or equal to 12V, then the internal circuitry will be powered by the voltage regulator and the 5V pin can be used as an output pin to power the microcontroller. The jumper should not be placed when the power supply is greater than 12V and separate 5V should be given through 5V terminal to power the internal circuitry. ENA and ENB pins are speed control pins for Motor A and Motor B while IN1& IN2 and IN3 & IN4 are direction control pins for Motor A and Motor B.

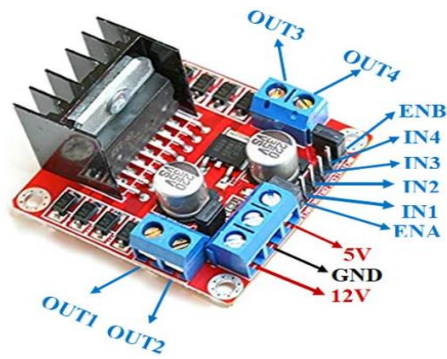


Figure 2.2: A L298N Motor Driver and its pinouts

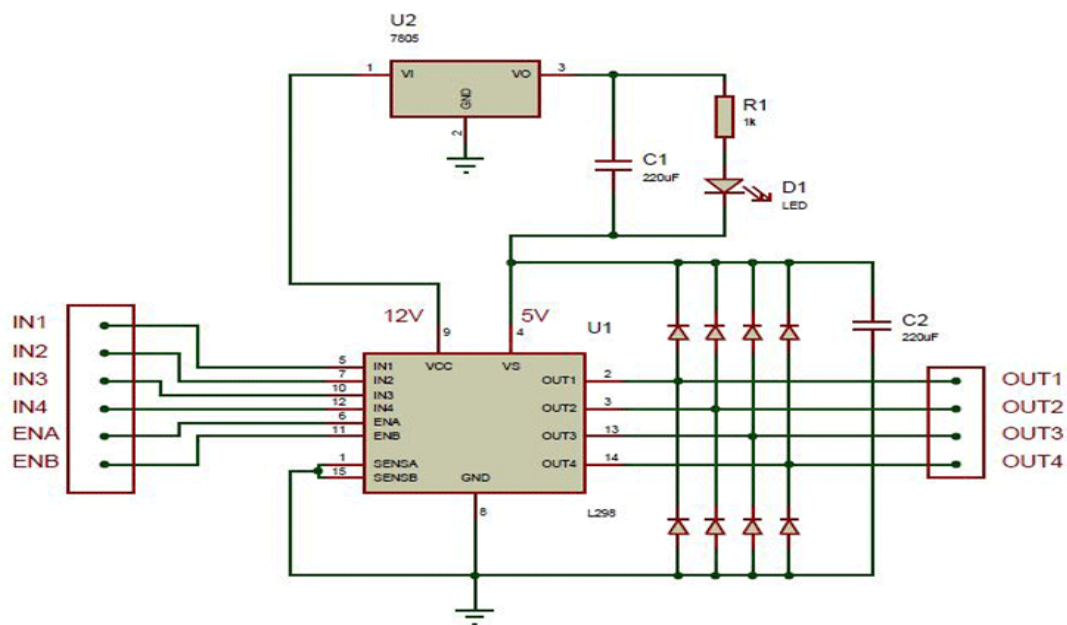


Figure 2.3: An L298N Module Pinout Configuration

Table 2.2: Details of the pins of L298N

Pin Name	Description
IN1 & IN2	Motor A input pins. Used to control the spinning direction of Motor A
IN3 & IN4	Motor B input pins. Used to control the spinning direction of Motor B

ENA	Enables PWM signal for Motor A
ENB	Enables PWM signal for Motor B
OUT1 & OUT2	Output pins of Motor A
OUT3 & OUT4	Output pins of Motor B
12V	12V input from DC power Source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground pin

2.8.3 DC Motors

A machine that converts DC power into mechanical power is known as a DC motor. Its operation is based on Faraday's principle which states that when a current carrying conductor is placed in a magnetic field, the conductor experiences a mechanical force. DC motors have a revolving armature winding but non-revolving armature magnetic field and a stationary field winding or permanent magnet. Different connections of the field and armature winding provide different speed/torque regulation features. The speed of a DC motor can be controlled by changing the voltage applied to the armature or by changing the field current.



Figure 2.4: A Simple DC Motor

CHAPTER 3

MATERIALS AND METHODS

3.1 Project Design

This project is designed to be in two major parts: Software and Hardware. Listed below are the hardware and software used in the project. The hardware used is divided into Receiver and Transmitter Sections:

Hardware

- Receiver
 - Esp8266 Node MCU
- Transmitter Section
 - Android device with a gyroscope sensor and hotspot capability

Software

- Arduino IDE
- MIT App Inventor

Other Equipment

- Buck Converter
- L298N Motor driver
- Motor Chassis
- 4 Dc motors and 4 wheels
- Jumper wires
- Batteries
- Vero Board
- Soldering iron and lead

3.2 Methodology

3.2.1 Gyroscope's data

In this project, we plan to control a robotic car with our gesture. To do this we need a gyroscope whose output data will change with our gesture. Since android devices, these days, come with a gyroscope, we will use an already developed mobile application: MIT App

Inventor to access the gyroscope's data. This mobile application runs on an android device that has a gyroscope sensor built-in into it. All that the MIT App Inventor does is to get the gyroscope's data, process the data and transmits over any WiFi network.

3.2.2 Wi-Fi Network

A very important requirement, for transmitting the gyroscope data over WiFi network, is that the MIT App Inventor and any receiving device (NodeMCU in our case) must be on the same Wi-Fi network. The Android phone's hotspot provided the network to connect both NodeMCU and the Android MIT App Inventor application together.

3.2.3 Signal Processing

The signals from MIT App Inventor application are then processed, before transmission into:

- A move forward code
- A go backward code
- A turn left code
- A turn right code

These codes are then interpreted by the NodeMCU and converted into voltages which are sent into the motor driver to move the robotic car in accordance to the direction of the tilting of the Android Phone. Figure 3.1 below illustrates the encoding and decoding of the codes.

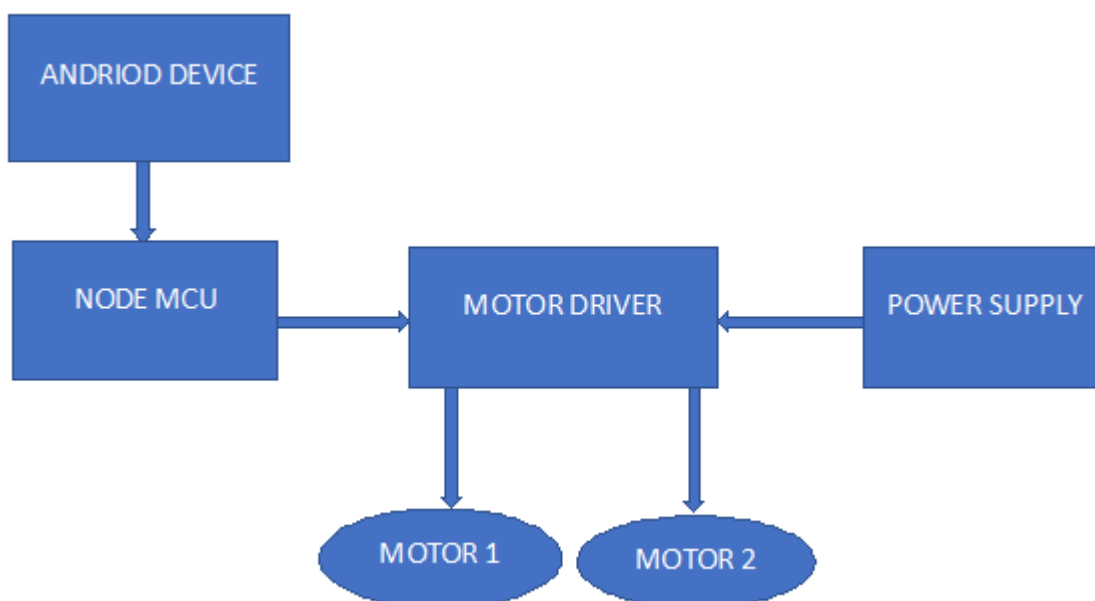


Figure 3.1: Illustration of the flow of program

3.3 Components Assembly

3.3.1 Chassis, motors and wheels

There are four wheels and four motor. Each of the wheels and motors are connected in pairs and attached to the chassis.

3.3.2 Powering the motors

Connecting wires are soldered to the positive and negative part of the motors and connected to the outputs of the motor driver. The two left wheels are connected together in positive-to-positive and negative-to-negative manner. The same form of connection is done for the two right wheels. In total, there are now one positive and one negative from the left wheels. This is connected to one section of the L298N motor controller. Similarly, there are one positive and one negative wire from the right wheels. This is connected to another section of the L298N motor controller.

3.3.3 Powering the L298N motors driver (or controller)

The motor controller is powered by a DC-DC Buck converter. This Buck converter is an adjustable DC-DC step down module. It was set to 9V.

3.3.4 Powering the NodeMCU

The L298N motor controller has a 5V output source. Since the NodeMCU operates on 5V, the NodeMCU was connected to the 5V output source on the L298N motor controller.

3.3.5 Remotely controlling the L298N motors driver (or controller)

As someone makes a gesture, the Android phone moves.



Figure 3.2: Attachment of the motors and wheels to the chassis

CHAPTER 4

IMPLEMENTATION AND RESULT

4.1 CIRCUIT ENGINEERING

In this section a circuit will be modelled to enable the motor driver to receive instructions from the node MCU to control the motors to move in the specified directions.

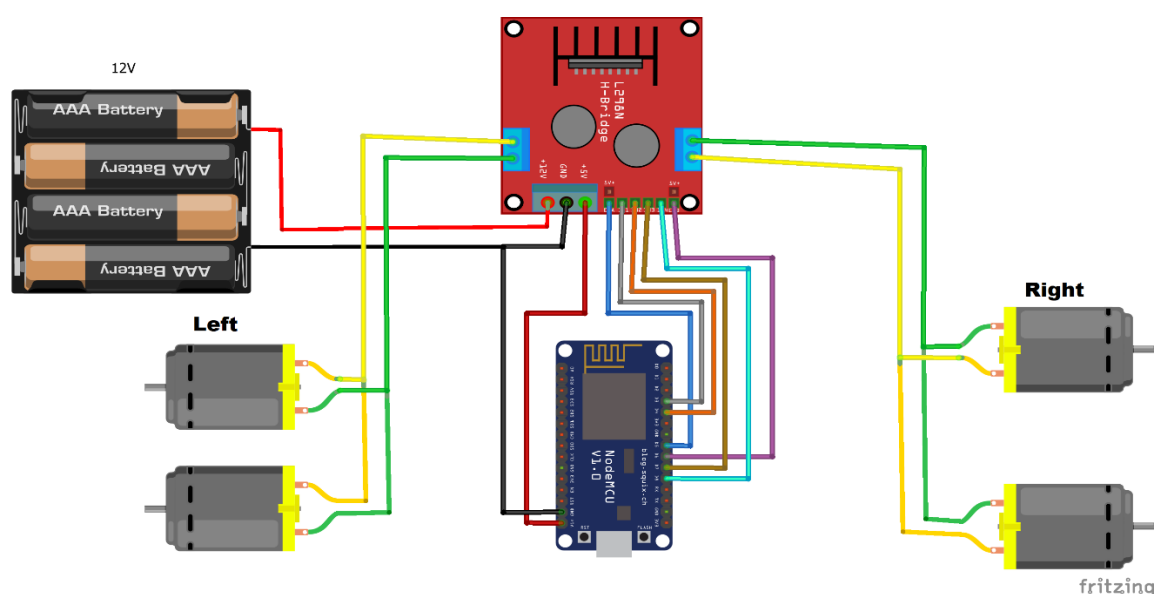


Figure 4.1: Circuit Diagram

4.1.1 Description of Circuit

The circuit above has been designed to function according to the aim of this project which is wirelessly controlling the movement of a robotic car with the aid of gestures from the gyroscope sensor embedded in an android device. In the circuit above the motors are connected to the motor driver, the positive and negative of the two left motors are connected to the output 1 and 2 of the motor driver respectively, the positive and negative of the two right motors are connected to the output 3 and 4 of the motor driver respectively. The motor driver is connected to the node MCU, IN1 and IN2 of the motor driver are connected to pin 3 and 4 of the node MCU to control the left motors, IN3 and IN4 of the motor driver are connected to pin 7 and 8 of the node MCU to control the right motors, ENA and ENB of the

motor driver are connected to pin 5 and 6 of the node MCU to control the speed and direction of the motors. The vin pin of the node MCU is connected to the 5v outlet of the motor driver, the ground pin of the node MCU is connected to the ground outlet of the motor driver and negative of the battery and 12v of the motor driver to the positive of the battery.

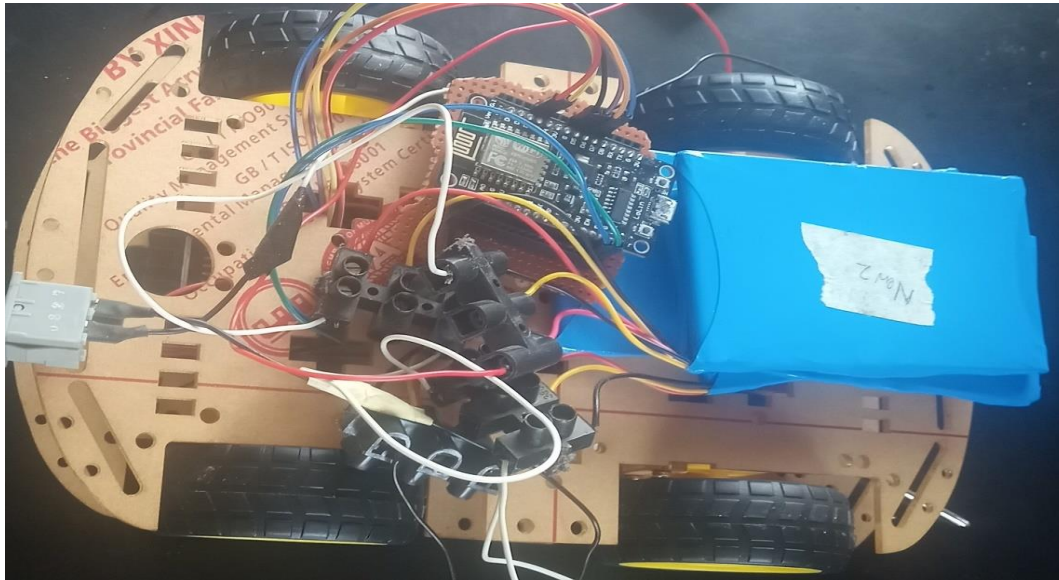


Figure 4.2: Image of the robotic car completely connected

4.2 Coding of node MCU

The objective of this section is to write and upload the code for programming the node MCU to enable it to receive instructions from the android device, interpret it and control the motor driver according to the interpretation which is to move the motors in the specified direction. An Arduino IDE with board esp8266 will be used to program the node MCU.

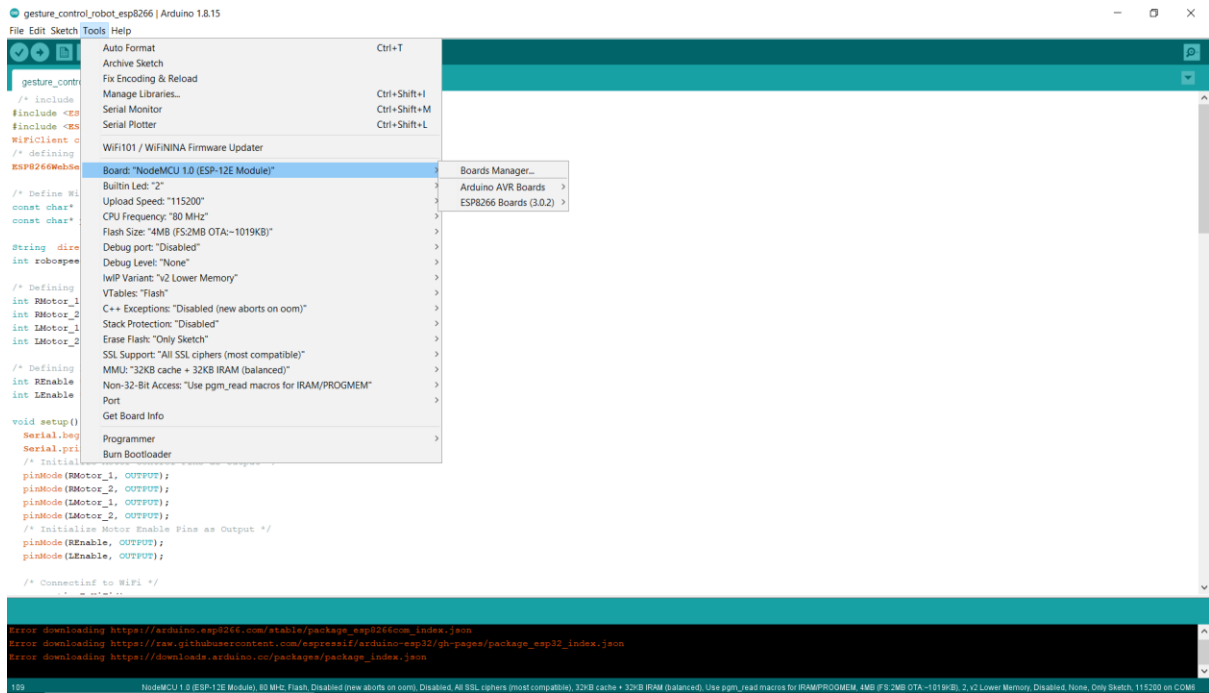


Figure 4.3: Image of the Arduino IDE

When the code as been written, it will be compiled to check for errors, if there are no errors, the node MCU is then connected to a computer via a USB cord, once the device is detected, the appropriate port is selected then the code is uploaded. After the code has been successfully uploaded, the node MCU will automatically connect to the mobile hotspot of the android device via its WIFI module to receive instructions, the serial monitor on the Arduino IDE is opened to confirm the successful connection of the node MCU to the android device’s hotspot and an IP address is also displayed which will be put into the application on the android device to interface it with the node MCU.

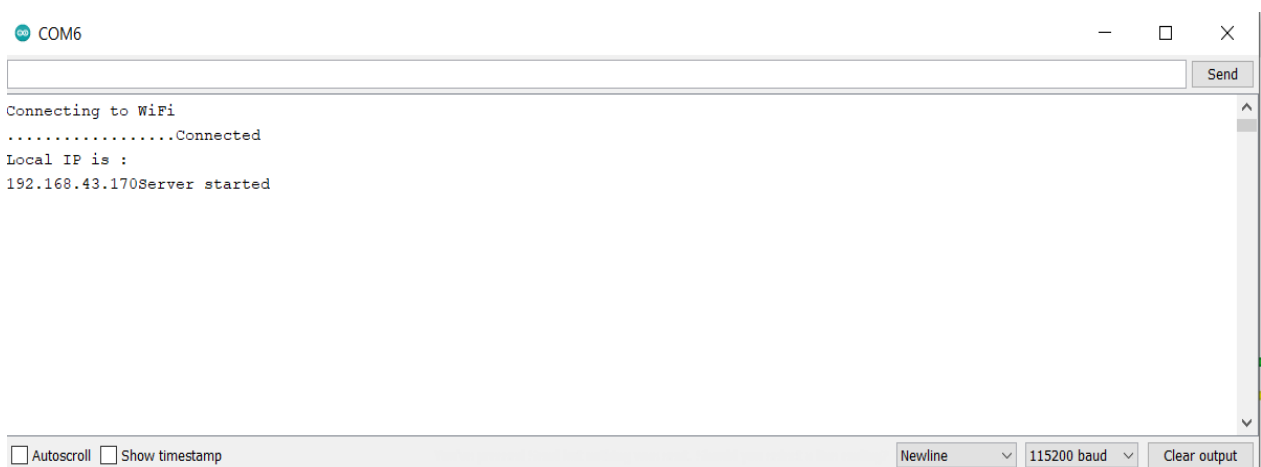


Figure 4.4: image of successful connection of the NodeMCU to hotspot

4.3 Gesture Control Application Development

In this section an application will be developed to use the gyroscope sensor embedded in the android device to read and effectively communicate with the node MCU. The development of this app will be implemented using the “MIT App Inventor” website. The app will be developed in such a way that it will connect to the node MCU using an IP address, measure the readings of the gyroscope sensor as the device is being tilted in different directions and sending the readings to the node MCU to interpret and move the motors in the specified directions.



Figure 4.5: image of the gesture control app

4.4 RESULT

After the successful assembling of the components, circuit engineering, programming of the node MCU and development of the gesture control app. The android device successfully controlled the movements of the robotic car in different directions. It also covered a long distance. The android device could control the robotic car from any distance as long as it stays connected to the mobile hotspot of the device. Once there is a break in connection the device disconnects, the robotic car stops responding until the node MCU re-connects to the hotspot of the android device. The batteries last long and supply the motor driver with enough voltage.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

The objective of this project which is to control a robotic car using gestures was achieved with just a few problems along the way. The robot is showing proper responses whenever the device is tilted in any direction. The car moves forward whenever the device is tilted downward, moves backward whenever the device is tilted upward, moves left and right whenever the device is tilted left and right respectively and stops whenever the device is parallel to the surface of the earth. The connection between the app and the node MCU is fast, and the robotic car can be controlled from a long range. The problems encountered were with soldering some of the parts to the vero board and bad connecting wires, which we overcame. This project has therefore proven+ that an android device can be used to control a robot wirelessly.

5.2 RECOMMENDATIONS

The on-board batteries occupy a lot of space and are also quite heavy. Some alternate power source for the batteries or replace the current DC Motors with ones which require less power can be used.

In terms of range the WIFI connection used is limited to the distance between the android device and the node MCU, this can be upgraded by using a GSM module which will provide a wider range of connectivity.

The robotic car could only be controlled when it is in an area of sight by the user, this can be fixed by adding on-board cameras to monitor the movement of the car even when it is out of sight.

REFERENCES

1. Wi-Fi technology -<https://smallbusiness.chron.com/introduction-wifi-technology-62018.html>
2. Parihar, Y. S. (2019). *Internet of Things and Nodemcu A review of use of Nodemcu ESP8266 in IoT products. JETIR,6(6)*
3. L298n Motor Driver <https://components101.com/modules/l298n-motor-driver-module>
4. "Robotic Gesture Recognition (1997)" by Jochen Triesch and Christoph Von Der Malsburg <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.37.5427>>
5. "Real-Time Robotic Hand Control Using Hand Gestures" by Jagdish Lal Raheja, Radhey Shyam, G. Arun Rajsekhar and P. Bhanu Prasad
6. "Hand Gesture Controlled Robot" by Bhosale, P.S., Bunage, Y.B. and Shinde, S.V.
7. <http://www.robotplatform.com/howto/L293/motor_driver_1.htm>
8. http://en.wikipedia.org/wiki/Gesture_interface
9. 'Wi-Fi Robot For Video Monitoring & Surveillance System' by Pavan.C, Dr. B. Siva kumar, *International Journal of Scientific & Engineering Research Volume 3, Issue 8, August-2012 ISSN 2229-5518*
10. C.W.Warren , S.G.Buckley created "Mobile robotic over wifi network using arduino technology".
11. Harshit Gulati, Shriyannsh Vaishya created "Bluetooth and wifi control rescue robot"
12. Wu Xing-Han, Su Mu-Chun, and Wang Pa-Chun, "A hand-gesture-based control interface for a car-robot", *Institute of Electrical and Electronics Engineers.2010; 4(10): 4644-648. 10.*
13. Sathiyarayanan Mithileysh, Azharuddin Syed, Kumar Santhosh, Khan Gibran, "Gesture controlled robot for military purpose", *International Journal for Technological Research in Engineering.2014; 1(11): 1300-303. 11.*

14. Mansuri Riyaz, Vakale Sandesh, Shinde Ashish, Patel Tanveer, “Hand gesture control robot vehicle”, *International Journal of Electronics & Communication Technology*.2013; 4(2): 77-80.

15. <https://en.wikipedia.org/wiki/NodeMCU>

16. <https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet>

17. <https://components101.com/modules/l293n-motor-driver-module>

18. <https://en.wikipedia.org/wiki/Robotics>

APPENDIX

Node MCU CODE

```
/* include library */
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
WiFiClient client;
/* defining server port */
ESP8266WebServer server(80);

/* Define WiFi Credentials */
const char* ssid = "Kvng aiseal"; /* Your SSID */
const char* password = "boppob123456"; /* Your Password */

String direction = "";
int robospeed = 800; /* Defining speed limit min 400 to max 1023 */

/* Defining right and left motor pins */
int RMotor_1 = 0; /* Right Motor GPIO0(D3) */
int RMotor_2 = 2; /* Right Motor GPIO2(D4) */
int LMotor_1 = 13; /* Left Motor GPIO13(D7) */
int LMotor_2 = 15; /* Left Motor GPIO15(D8) */

/* Defining L298N enable pins */
int REnable = 14; /* Right Motor Enable Pin GPIO14(D5) */
int LEnable = 12; /* Right Motor Enable Pin GPIO12(D6) */

void setup() {
  Serial.begin(115200);
  Serial.println("GPIO test!");
  /* Initialize Motor Control Pins as Output */
```

```

pinMode(RMotor_1, OUTPUT);
pinMode(RMotor_2, OUTPUT);
pinMode(LMotor_1, OUTPUT);
pinMode(LMotor_2, OUTPUT);
/* Initialize Motor Enable Pins as Output */
pinMode(REnable, OUTPUT);
pinMode(LEnable, OUTPUT);

/* Connectinf to WiFi */
connectingToWiFi();

server.on("/", handleRequest);
server.onNotFound(handleRequest);

/* Starting Server */
server.begin();
Serial.println("Server started");

/* Starting Server */
analogWrite(REnable, robospeed);
analogWrite(LEnable, robospeed);
}
void loop() {

/* Handling Request */
server.handleClient();
String direction = server.arg("direction");
Serial.println(direction);

if (direction == "F") {
    move_forward(); /* If direction F moves forward */
}
}

```

```

}
else if (direction == "B") {
    move_backward(); /* If direction B moves backward */
}
else if (direction == "R") {
    turn_right(); /* If direction R turns right */
}
else if (direction == "L") {
    turn_left(); /* If direction L turns left */
}
else if (direction == "S") {
    move_stop(); /* If direction S stops moving */
}
}

```

```

void handleRequest() {
    if ( server.hasArg("direction") )
    {
        Serial.println(server.arg("direction"));
    }
    server.send ( 200, "text/html", "" );
    delay(1);
}

```

```

void connectingToWiFi() {
    delay(3000);
    WiFi.disconnect();
    delay(1000);
    Serial.println("Connecting to WiFi");
    WiFi.begin(ssid, password);
    while (!(WiFi.status() == WL_CONNECTED)) {

```

```
    delay(1000);
    Serial.print("...");
}
Serial.println("Connected");
Serial.println("Local IP is : ");
Serial.print((WiFi.localIP().toString()));
delay(5000);
}
```

```
/* Move Forward */
```

```
void move_forward() {
    digitalWrite(RMotor_1, LOW);
    digitalWrite(RMotor_2, HIGH);
    digitalWrite(LMotor_1, HIGH);
    digitalWrite(LMotor_2, LOW);
}
```

```
/* Move Backward */
```

```
void move_backward() {
    digitalWrite(RMotor_1, HIGH);
    digitalWrite(RMotor_2, LOW);
    digitalWrite(LMotor_1, LOW);
    digitalWrite(LMotor_2, HIGH);
}
```

```
/* Turn Right */
```

```
void turn_right() {
    digitalWrite(RMotor_1, LOW);
    digitalWrite(RMotor_2, HIGH);
    digitalWrite(LMotor_1, LOW);
    digitalWrite(LMotor_2, HIGH);
}
```

```
}

/* Turn Left */
void turn_left() {
    digitalWrite(RMotor_1, HIGH);
    digitalWrite(RMotor_2, LOW);
    digitalWrite(LMotor_1, HIGH);
    digitalWrite(LMotor_2, LOW);
}
```

```
/* Stop Move */
void move_stop() {
    digitalWrite(RMotor_1, LOW);
    digitalWrite(RMotor_2, LOW);
    digitalWrite(LMotor_1, LOW);
    digitalWrite(LMotor_2, LOW);
}
```