**VOICE AND APP CONTROLLED CAR using Node MCU**

**A PROJECT REPORT**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

***Under the Guidance of***

**SOUVIK SARKAR**

***Project Carried Out At***

****

**Ardent Computech Pvt Ltd (An ISO 9001:2015 Certified)**

**SDF Building, Module #132, Ground Floor, Salt Lake City, GP Block, Sector V, Kolkata, West Bengal 700091**

***Submitted By***

**NILASREE HALDER**

**ARYA MUKHERJEE**

**DEBMALYA KONER**

**ANINDYA SARKAR**

**AMIT PAUL**

AOT Logo

**ACADEMY OF TECHNOLOGY**

**Grand Trunk Road, Aedconagar, near Adisaptagram Railway Station**

**JUNE – JULY 2019**

**In association with**

****

***(Note:All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)***

1. Title of the Project: **Voice and App controlled Car using Node MCU**

2. Project Members: Nilasree Halder

Arya Mukherjee

Debmalya Koner

Anindya Sarkar

Amit Paul

3. Name and Address of the Guide: **Mr. Souvik Sarkar**

Technical Head (Core Department)

Ardent Computech Pvt Ltd (An ISO 9001:2015 Certified)

Module 132, SDF Building, Sector 5, Salt Lake, Kolkata - 64

Ph.D**\***  M.Tech.**\*** B.E\***/**B.Tech.\*MCA**\*** M.Sc.**\***

Y

Y

Y

4. Educational Qualification of the Guide:

5. Working / Training experience of the Guide: **8 Years**

6. Project Version Control History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Authors** | **Description of Version** | **Date Completed** |
| Final | Nilasree Halder  Arya Mukherjee  Debmalya Koner  Anindya Sarkar  Amit Paul | Project Report | 6th July, 2019 |

1.

2.

3.

4.

5. **Mr. Souvik Sarkar**

Signatures of Team Members Signature of Project Approval Proposal Evaluator

Date: Date:

**For Office Use Only**

**Approved**

**Not Approved**

**Project Responsibility Form**

**VOICE & APP CONTROLLED CAR**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | SL.NO. | NAME OF MEMBER | RESPONSIBILITY | | 1 | Nilasree Halder | App implementation and Designing | | | 2 | Arya Mukherjee | App implementation and Designing | | | 3 | Debmalya Koner | Project Leader and Idea Implementation | | | 4 | Amit Paul | Analyst and Management | | | 5 | Anindya Sarkar | Coding and Designing | |   Each group member must participate in project development and developing the ideas for the required elements. Individual group members will be responsible for completing tasks which help to finalize the project and the performance. All group members must be assigned a task. | |  | | --- | |  | |

Date:

Name of the Students

1. Nilasree Halder
2. Arya Mukherjee
3. Debmalya Koner
4. Anindya Sarkar
5. Amit Paul

Signatures of the students

a.

b.

c.

d.

e.

**DECLARATION**

We hereby declare that the project work being presented in the project proposal entitled **“App AND VOICE CONTROLLED CAR USING Node MCU”** in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** at **ARDENT COMPUTECH PVT. LTD, SALTLAKE, KOLKATA, WEST BENGAL,** is an authentic work carried out under the guidance of **MR. SOUVIK SARKAR**. The matter embodied in this project work has not been submitted elsewhere for the award of any degree of our knowledge and belief.

Date:

Name of the Students

1. Nilasree Halder
2. Arya Mukherjee
3. Debmalya Koner
4. Anindya Sarkar
5. Amit Paul

Signature of the students

a.

b.

c.

d.

e.

****

**Ardent Computech Pvt Ltd (An ISO 9001:2008 Certified)**

**SDF Building, Module #132, Ground Floor, Salt Lake City, GP Block, Sector V, Kolkata, West Bengal 700091**

**CERTIFICATE**

This is to certify that this proposal of minor project entitled **“VOICE AND APP CONTROLLED CAR”** is a record of bona fide work, carried out by **1. DEBMALYA KONEr, 2. ANINDYA SARKAR, 3. NILASREE HALDER, 4.ARYA MUKHERJEE, 5.amit paul** under my guidance at **Ardent Computech Pvt Ltd**. In my opinion, the report in its present form is in partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY IN ELECTRONICS AND COMMUNICATION ENGINEERING** and as per regulations of the **Ardent*®.*** To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report.

**Guide / Supervisor**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mr. SOUVIK SARKAR**

Technical Head (CORE Department)

Ardent Computech Pvt Ltd (An ISO 9001:2015 Certified)

Module 132, SDF Building, Sector 5, Salt Lake, Kolkata - 64

**ACKNOWLEDGEMENT**

Success of any project depends largely on the encouragement and guidelines of many others. We take this sincere opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project work.

Our heartfelt thanks to Dr.Abhijit Banerjee, HOD of Electronics & Communication Engineering of ACADEMY OF TECHNOLOGY, for providing us the opportunity to develop the project at Ardent Computech Pvt.Ltd.

We would like to show our greatest appreciation to Mr. Souvik Sarkar, Project Manager at Ardent, Kolkata. We always feel motivated and encouraged every time by his valuable advice and constant inspiration; without his encouragement and guidance this project would not have materialized.

We wish to express our deep sense of gratitude to our internal guide, Mrs. Rakhi Neogy, Assistant Professor, AOT affiliated MAKAUT for her able guidance and useful suggestions, which helped us in completing the project work in time.

Words are inadequate in offering our thanks to the other trainees, project assistants and other members at Ardent Computech Pvt. Ltd. for their encouragement and cooperation in carrying out this project work. The guidance and support received from all the members and who are contributing to this project, was vital for the success of this project.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **S.NO.** | **Name of the Topic** |
| **1.** | **Introduction of Internet Of Things** |
|  |  |
| **2.** | **Project Introduction** |
|  |  |
| **3.** | **Project Component** |
| 3.1 | Hardware Component |
| 3.2 | Software Component |
|  |  |
| **4.** | **Description of Components** |
| 4.1 | NodeMCU |
| 4.2 | L298N |
| 4.3 | BO Motors |
| 4.4 | Arduino IDE |
| 4.6 | MIT App Inventor |
|  |  |
| **5.** | **IMPLEMENTATION OF THE PROJECT** |
|  | **Procedures** |
| **6.** | **Circuit Diagram** |
|  |  |
| 7. | **Project Coding** |
|  |  |
| **8.** | **Conclusion** |
|  |  |
| **9.** | **Future Scope & Further Enhancements** |
|  |  |
| **10.** | **Bibliography/References** |

* **INTRODUCTION**

**INTERNET OF THINGS:**

The Internet of things  is the extension of Internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity, and other forms of hardware (such as sensors), these devices can communicate and interact with others over the Internet, and they can be remotely monitored and controlled.

 The Internet of things has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), and others all contribute to enabling the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", covering devices and appliances (such as lighting fixtures, thermostats, home security systems and cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers.

The IoT concept has faced prominent criticism, especially in regards to privacy and security concerns related to these devices and their intention of pervasive presence.

**HISTORY:**

The concept of a network of smart devices was discussed as early as 1982, with a modified Coke vending machine at Carnegie Mellon University becoming the first Internet-connected appliance, able to report its inventory and whether newly loaded drinks were cold or not. Mark Weiser’s 1991 paper on ubiquitous computing, "The Computer of the 21st Century", as well as academic venues such as UbiComp and PerCom produced the contemporary vision of the IoT. In 1994, Reza Raji described the concept in IEEE Spectrum as "[moving] small packets of data to a large set of nodes, so as to integrate and automate everything from home appliances to entire factories". Between 1993 and 1997, several companies proposed solutions like Microsoft’s at Work or Novell’s  NEST. The field gained momentum when Bill Joy envisioned device-to-device communication as a part of his "Six Webs" framework, presented at the World Economic Forum at Davos in 1999.

The term "Internet of things" was likely coined by Kevin Ashton of Procter & Gamble, later MIT's Auto-ID Center, in 1999.  A research article mentioning the Internet of Things was submitted to the conference for Nordic Researchers in Norway, in June 2002,which was preceded by an article published in Finnish in January 2002. Defining the Internet of things as "simply the point in time when more 'things or objects' were connected to the Internet than people", Cisco Systems estimated that the IoT was "born" between 2008 and 2009, with the things/people ratio growing from 0.08 in 2003 to 1.84 in 2010.

**Trends and characteristics:**

The IoT's major significant trend in recent years is the explosive growth of devices connected and controlled by the Internet. The wide range of applications for IoT technology mean that the specifics can be very different from one device to the next but there are basic characteristics shared by most.

The IoT creates opportunities for more direct integration of the physical world into computer-based systems, resulting in efficiency improvements, economic benefits, and reduced human exertions.

The number of IoT devices increased 31% year-over-year to 8.4 billion in the year 2017 and it is estimated that there will be 30 billion devices by 2020.[]](https://en.wikipedia.org/wiki/Internet_of_things#cite_note-auto3-99) The global market value of IoT is projected to reach $7.1 trillion by 2020.

Intelligence

Ambient intelligence and autonomous control are not part of the original concept of the Internet of things. Ambient intelligence and autonomous control do not necessarily require Internet structures, either. However, there is a shift in research (by companies such as Intel) to integrate the concepts of the IoT and autonomous control, with initial outcomes towards this direction considering objects as the driving force for autonomous IoT.  Training an IoT device to behave smartly in such an environment cannot be addressed by conventional machine learning algorithms such as supervised learning. By reinforcement learning approach, a learning agent can sense the environment’s state and perform actions and learn through the maximizing accumulated rewards it receives in long term.

IoT intelligence can be offered at two basic levels: IoT devices,and Cloud computing. The need for intelligent control and decision at each level depends on the time sensitiveness of the IoT application. In the future, the Internet of Things may be a non-deterministic and open network in which auto-organized or intelligent entities and virtual objects will be interoperable and able to act independently (pursuing their own objectives or shared ones) depending on the context, circumstances or environments. Autonomous behavior through the collection and reasoning of context information as well as the object's ability to detect changes in the environment (faults affecting sensors) and introduce suitable mitigation measures constitutes a major research trend, clearly needed to provide credibility to the IoT technology. Modern IoT products and solutions in the marketplace use a variety of different technologies to support such context-aware automation, but more sophisticated forms of intelligence are requested to permit sensor units and intelligent cyber-physical systems to be deployed in real environments.

**Challenges faced by IOT**

As the Internet of Things (IoT) continues to steer operations in the 21st century, numerous challenges are coming to light.

While the IoT still has the potential to transform business for owners, employees and customers alike, those who already embrace this next-gen network still have some work to do.

Not only are they trying to make the most of IoT integration to benefit their own company, but they’re also treading new ground and serving as role models for those who have yet to take the plunge.

1. Meeting customer expectations:

In the 1990s, the widespread availability of internet access forever changed the way consumers shop. It also switched the customer’s focus from standardized, mass-produced goods to customized products and services.

With the year 2020 on the horizon, customers have higher expectations than ever before. According to a recent report by Salesforce, 57 percent of consumers are more interested in doing business with an innovative or forward-thinking company — and 50 percent won’t hesitate to switch brands if their needs go unmet.

2. Easing security concerns:

The IoT was initially touted as a hyper-secure network that was suitable for storing and transmitting confidential datasets. Although it’s true that the IoT is more secure than the average internet or LAN connection, it’s not exactly the bulletproof shell some users expected.

Some of the most significant security concerns involve both the IoT and the cloud. A recent analysis predicts a loss of up to $120 billion in economic fallout in the takedown of just one cloud datacenter.

Reports also state an annual economic cost of cybercrime at upward of $1 trillion — which is quite a leap for 2017’s record-setting figure of roughly $300 billion.

3. Keeping IoT hardware updated:

Regardless of how a company uses the IoT or the cloud, data integrity is a common challenge. With so much data coming in from multiple sources, it’s tough to separate useful, actionable information from irrelevant chatter.

It’s critical to calibrate your IoT sensors on a regular basis, just as you would any other kind of electrical sensor. Next-gen sensors are embedded in many different devices, including panel meters, chart recorders, current clamps, power monitors and more, and it’s difficult to synchronize the dataflow between all this hardware without the help of a professional team.

4. Overcoming connectivity issues:

In its current form, the IoT utilizes a centralized, server-client model to provide connectivity to the various servers, workstations and systems. According to updated reports from Gartner, more than 20 billion individual units will connect to the IoT by 2020. It’s just a matter of time before users start to experience significant bottlenecks in IoT connectivity, efficiency and overall performance.

5. Waiting for governmental regulation:

While some businesses immediately embraced the IoT, others are hesitant. In many cases, these businesses are waiting for government officials to intervene with new standards and regulations.

Complicating matters even further is the sheer amount of IoT-connected devices.

Although most experts agree that IoT regulation is a necessity, they have yet to formulate any standards or guidelines for the public to follow.

**PROJECT:**

INTRODUCTION Android is a very familiar word in the world today. Millions of devices are running the Google Android OS and millions are being developed daily. Google has made the Android development platform open to everyone around the world, so there are millions of developers. Although some developers just focus on building the apps or games for the android devices, there are numerous possibilities as well. One of the possibilities of Android development is its fusion with Arduino (a microprocessor), which in itself is a tiny computer. Possibilities from the combination of these two-development platforms cannot be derived from any permutation or combination logics. While there are many results already published, there are more innovations everyday. This thesis focuses on the outcome of the possible combination of Android and Node MCU. Although the project has been carried out numerous times before, this thesis gives detailed information on building a robot which can be controlled by any android device and also through the voice over the android device. This thesis includes two parts: i. the hardware and ii. the Software. The first part focuses on assembling the robot parts and building a circuit and the second part is about programming the interface on the android device. The thesis is being divided in two aspects of computing and the author had to focus on both aspects. The first part of the thesis, which is hardware, requires a brief knowledge of electronics circuit design and the second part (android interface) was designed using MIT App Inventor. This thesis starts with assembling the chassis for the robot in Part 1, designing the circuit in Arduino in Part 2 and building the android interface in Part 3.

* **PROJECT COMPONENT**

**HARDWARE:**

|  |
| --- |
| NodeMCU |
| L298N |
| BO Motors |
| Mini Bread Board |
| Jumper Wires |
| 9V Battery |

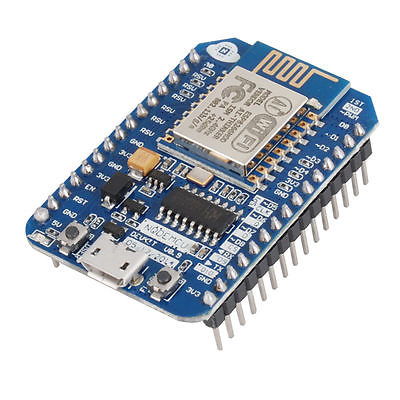
**SOFTWARE:**

|  |
| --- |
| Arduino IDE |
| MIT App Inventor |

* **DESCRIPTION OF COMPONENTS**

1. **Node-MCU:**

NodeMCU is an open source LUA based firmware developed for ESP8266 wifi chip. By exploring functionality with ESP8266 chip, NodeMCU firmware comes with ESP8266 Development board/kit i.e. NodeMCU Development board.



**NodeMCU Development Board/kit v0.9 (Version1)**

Since NodeMCU is open source platform, their hardware design is open for edit/modify/build.

NodeMCU Dev Kit/board consist of ESP8266 wifi enabled chip. The **ESP8266** is a low-cost Wi-Fi chip developed by Espressif Systems with TCP/IP protocol. For more information about ESP8266, you can refer ESP8266 WiFi Module.

There is Version2 (V2) available for NodeMCU Dev Kit i.e. **NodeMCU Development Board v1.0 (Version2)**, which usually comes in black colored PCB.

For more information about NodeMCU Boards available in market refer NodeMCU Development Boards

NodeMCU Dev Kit has **Arduino like** Analog (i.e. A0) and Digital (D0-D8) pins on its board.

It supports serial communication protocols i.e. UART, SPI, I2C etc.

Using such serial protocols we can connect it with serial devices like I2C enabled LCD display, Magnetometer HMC5883, MPU-6050 Gyro meter + Accelerometer, RTC chips, GPS modules, touch screen displays, SD cards etc.

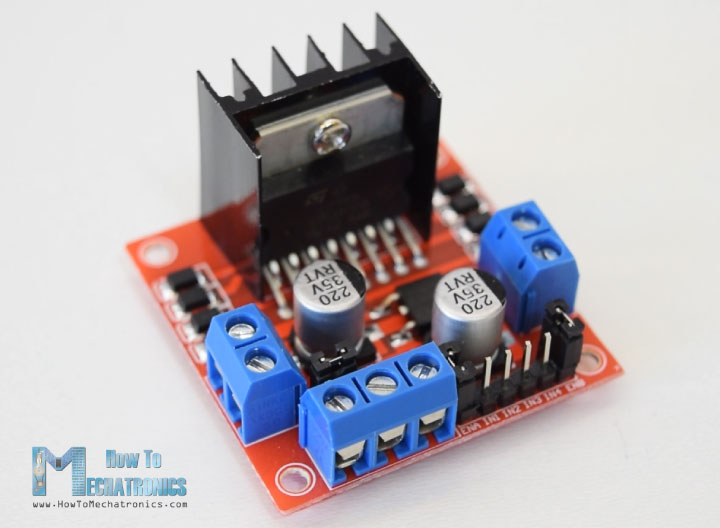
NodeMCU provides access to the GPIO (General Purpose Input/Output) and a pin mapping table is part of the API documentation.

|  |  |
| --- | --- |
| **I/O index** | **ESP8266 pin** |
| 0 [\*] | GPIO16 |
| 1 | GPIO5 |
| 2 | GPIO4 |
| 3 | GPIO0 |
| 4 | GPIO2 |
| 5 | GPIO14 |
| 6 | GPIO12 |
| 7 | GPIO13 |
| 8 | GPIO15 |
| 9 | GPIO3 |
| 10 | GPIO1 |
| 11 | GPIO9 |
| 12 | GPIO10 |

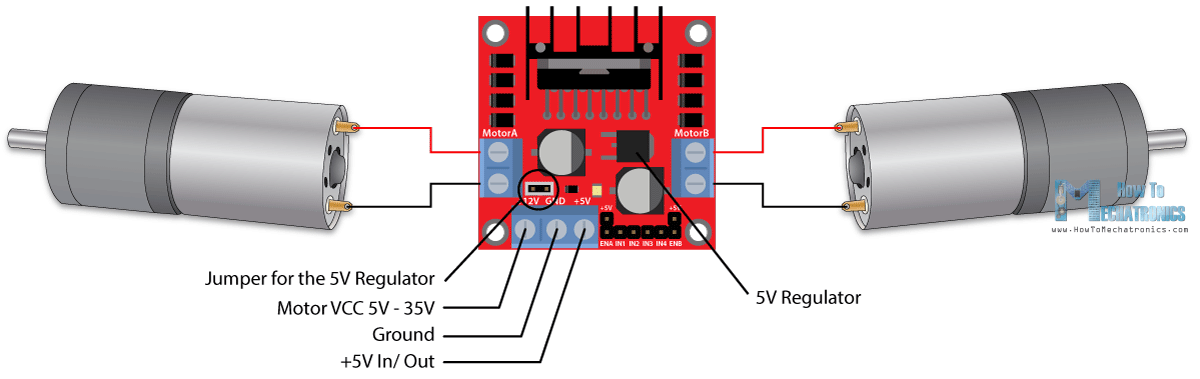
[\*] D0 (GPIO16) can only be used for GPIO read/write. It does not support open-drain/interrupt/PWM/I²C or 1-Wire.

1. **L298N:**

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A.



Let’s take a closer look at the pinout of L298N module and explain how it works. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output.

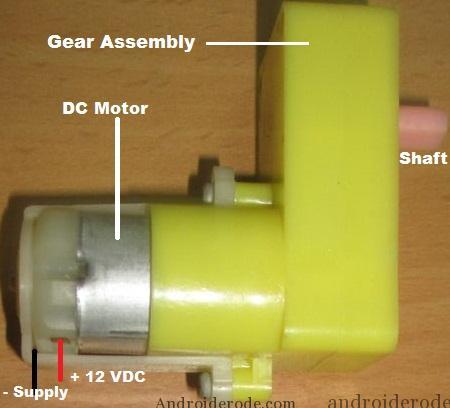


This depends on the voltage used at the motors VCC. The module have an onboard 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board. But if the motor voltage is greater than 12V we must disconnect the jumper because those voltages will cause damage to the onboard 5V regulator. In this case the 5V pin will be used as input as we need connect it to a 5V power supply in order the IC to work properly.

We can note here that this IC makes a voltage drop of about 2V. So for example, if we use a 12V power supply, the voltage at motors terminals will be about 10V, which means that we won’t be able to get the maximum speed out of our 12V DC motor.

1. **BO MOTORS**

DC motor (BO) Battery Operation. Dc motor converts electrical energy into mechanical energy.  Why DC gear motor used in robot Motor control circuit. DC MOTOR concept is where gears reduce the speed of the vehicle but increase its torque is known as gear reduction. In DC motor is assembled with multiple gear setup. Speed of motor is counted in terms of rotations of the soft per minute is called RPM. RPM means Revolution Per Minute. The setup assemble helps to increasing the torque and reduce the motor speed. All micro-controller based Robots this type of DC motor can be used.

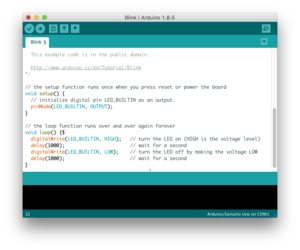
[](https://www.androiderode.com/dc-motor-dual-h-bridge-ic-l293d/)

DC motors were the first form of motor widely used, as they could be powered from existing direct-current lighting power distribution systems. A DC motor's speed can be controlled over a wide range, using either a variable supply voltage or by changing the strength of current in its field windings. Small DC motors are used in tools, toys, and appliances. The universal motor can operate on direct current but is a lightweight brushed motor used for portable power tools and appliances. Larger DC motors are currently used in propulsion of electric vehicles, elevator and hoists, and in drives for steel rolling mills. The advent of power electronics has made replacement of DC motors with AC motors possible in many applications.

1. **ARDUINO IDE:**

The **Arduino integrated development environment (IDE)** is a cross-platform application (for Windows, macOS, Linux) that is written in the programming language Java. It is used to write and upload programs to Arduino compatible boards, but also, with the help of 3rd party cores, other vendor development boards.[[2]](https://en.wikipedia.org/wiki/Arduino_IDE#cite_note-2)

The source code for the IDE is released under the GNU General Public License, version 2. The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User-written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub *main()* into an executable cyclic executive program with the GNU tool chain, also included with the IDE distribution. The Arduino IDE employs the program *avrdude* to convert the executable code into a text file in hexadecimal encoding that is loaded into the Arduino board by a loader program in the board's firmware.



* Arduino IDE is an open source software that is mainly used for writing and compiling the code into the Arduino Module.
* It is an official Arduino software, making code compilation too easy that even a common person with no prior technical knowledge can get their feet wet with the learning process.
* It is easily available for operating systems like MAC, Windows, Linux and runs on the Java Platform that comes with inbuilt functions and commands that play a vital role for debugging, editing and compiling the code in the environment.
* A range of Arduino modules available including Arduino Uno, Arduino Mega, Arduino Leonardo, Arduino Micro and many more.
* Each of them contains a microcontroller on the board that is actually programmed and accepts the information in the form of code.
* The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board.
* The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.
* This environment supports both C and C++ languages.

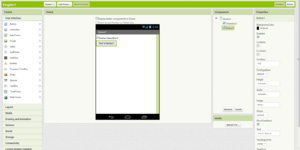
1. **MIT APP INVENTOR:**

**App Inventor for Android** is an open-source web application originally provided by Google, and now maintained by the Massachusetts Institute of Technology (MIT), which allows newcomers to computer programming to create software applications for the Android operating system (OS).

It uses a graphical interface very similar to Scratch and the Star Logo TNG user interface, which allows users to drag-and-drop visual objects to create an application that can run on Android devices. In creating App Inventor, Google drew upon significant prior research in educational computing, as well as work done within Google on online development environments.

App Inventor and the projects on which it is based are informed by constructionist learning theories, which emphasizes that programming can be a vehicle for engaging powerful ideas through active learning. As such, it is part of an ongoing movement in computers and education that began with the work of Seymour Papert and the MIT Logo Group in the 1960s and has also manifested itself with Mitchel Resnick's work on Lego Mindstorms and StarLogo.

App Inventor also supports the use of cloud data via an experimental FirebaseDB component.



To start making your application, click on the Projects menu on the top left of the screen and select the “Start new project” and then give the name of your project.

**Note: Your project name does not contain any Space. You can use underscore.**

**1. Designer Panel:**

Now, you are in the designer panel of your project, where you can layout the “user interfaces” of your app. You choose things for the user interface things like Buttons,Images, Text boxes to the palette.

Click and hold on the word “Button” in the palette, drag your mouse over to the Viewer. Drop the button and a new button will appear on the Viewer. You can change the properties of the components like height, width, text, color to the properties panel.

**2. Live Testing:**

You can test your app while you are building it. For this you need MIT AI2 Companion to connect your android phone or tablet.

The method for getting the AI2 Companion App is to download the app from the Play Store by searching for “MIT AI2 Companion” and install it on your phone.

To get code from App Inventor, click on the “Connect” button and select Al Companion. After installation open the app and scan QR code by clicking “Scan QR code”or you can type the code after that click on “connect with code” button. After Connecting, you can see the app on your phone.

**NOTE: Your phone and computer must both be on the same wireless network.and your phone’s wifi is on.**

**3. Block Editor:**

The Blocks Editor is where you program the behavior of your app. There are Built-in blocks that handle things like math, logic, and text with each components you have added. Click on the Button1 drawer. Click and hold the “when Button1.Click do” block. Drag it over to the workspace and drop it there. This is the block that will handle what happens when the button on your app is clicked. It is called an “Event Handler”. Now click on the Label1. Click and hold the “set Label1 text to” block. Drag it inside the button click, it will run when the button is pressed. At last, click on the text drawer, drag out a text block and plug it into the socket labeled to and write anything that you want to display. Click on the text block and write anything. Now go to your connected device and press on the button. You should see the text “This is my First App” on the screen.

**4. Sharing Apps:**

You can share your application source code (.aia form) that can be loaded into app inventor. Or you can share your app in an executable form (.apk) that can be installed on a device.

**IMPLEMENTATION OF THE PROJECT**

**PROCEDURES:**

We have used the IC L298N directly with the NodeMCU. The **L298N is** a dual H-**Bridge** motor driver which allows speed and direction control of two DC motors at the same time. The module **can** drive DC motors that **have** voltages between 5 and 35V, with a peak current up to 2A.

Our Robot has two wheels, that are driven by 2 DC motors:

* LEFT Motor
* RIGHT Motor

We have connected the motors to our H-Bridge . The NodeMCU have to have 6 GPIOs that have to command those motors.

For example, to drive LEFT motor FORWARD you must put:

* HIGH at pin D4 (left Motor +) and
* LOW at pin D3 (left Motor -)

For the RIGHT Motor you must do the opposite:

* HIGH at pin D8 (right Motor +) and
* LOW at pin D7 (left Motor -)

Due the way that my motors are assembly and the robot moves, the above combination have to drive both motors in the same direction pushing the Robot forward.

To control the H-Bridge properly, we had to work with the enable pins. We have defined 5 possible commands:

1. STOP
2. Turn to LEFT
3. Turn to RIGHT
4. REVERSE
5. FORWARD

The first command "STOP" is simple to accomplish. All H-Bridge inputs must be LOW, this way the motors have to not be moving:

The same way let's think about turn the robot to one direction, let's say LEFT. Consider that the robot is going Forward, if we want to turn to left, we can have two situations:

1. Stop LEFT motor and keep RIGHT going FW
2. Reverse LEFT Motor and keep RIGHT going FW

At first situation, the robot have to do an arc trajectory to the left. On the second, the robot have to turn in its axy to the left. We have implemented the second one.

1. Take the Kit (Chassis, 2 Wheels, 2 DC Motors)
2. Take out the Chassis cover (or leave it)
3. Solder 2 10 cm wires (Red and Black) at each motor
4. Fixed the motors at chassis
5. Assembly the coaster
6. Joint the motor wires to electronics (L298N)
7. Fix the 9V battery on the chassis

We connected the NodeMCU to our local WiFi and check its IP address.

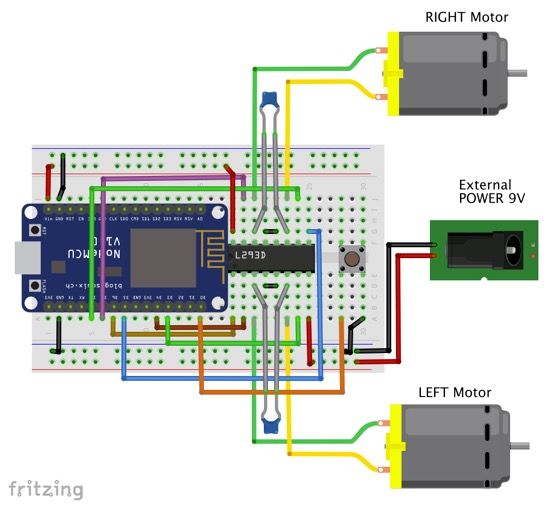
On the Serial monitor you can see its IP Address. Take note of it. We need it to connect with the Android App.

Uploaded the code to our NodeMCU and that's it!

We have used the MIT App Inventor to develop our Android App.

**Main Components on Screen** :

* User Interface:
  + Input of IP Address
    - TextBox named "IP\_Address"
  + 5 Command Buttons:
    - forward
    - reverse
    - right
    - left
    - stop
  + Voice Input Button

**CIRCUIT DIAGRAM**

L298N

**PROJECT CODING**

#include <ESP8266WiFi.h>

WiFiClient client;

WiFiServer server(80);

const char\* ssid = "recharge your jio";

const char\* password = "bond@007";

String command =""; // Command received from Android device

// Set Motor Control Pins

int rightMotor2 = 13; // D7 - right Motor -

int rightMotor1 = 15; // D8 - right Motor +

int leftMotor2 = 0; // D3 - left Motor -

int leftMotor1 = 2; // D4 - left Motor +

int eneLeftMotor = 12; // D6 - enable Mortor Left

int eneRightMotor = 14; // D5 - enable Mortor Right

void setup()

{

Serial.begin(115200);

pinMode(eneLeftMotor, OUTPUT);

pinMode(eneRightMotor, OUTPUT);

pinMode(leftMotor1, OUTPUT);

pinMode(leftMotor2, OUTPUT);

pinMode(rightMotor1, OUTPUT);

pinMode(rightMotor2, OUTPUT);

digitalWrite(eneLeftMotor,LOW);

digitalWrite(eneRightMotor,LOW);

digitalWrite(leftMotor1,LOW);

digitalWrite(leftMotor2,LOW);

digitalWrite(rightMotor1,LOW);

digitalWrite(rightMotor2,LOW);

connectWiFi();

server.begin();

}

void loop()

{

client = server.available();

if (!client) return;

command = checkClient ();

if (command == "forward") forwardMotor();

else if (command == "reverse") reverseMotor();

else if (command == "left") leftMotor();

else if (command == "right") rightMotor();

else if (command == "stop") stopMotor();

sendBackEcho(command); // send command echo back to android device

command = "";

}

/\* command motor forward \*/

void forwardMotor(void)

{

digitalWrite(eneLeftMotor,HIGH);

digitalWrite(eneRightMotor,HIGH);

digitalWrite(leftMotor1,HIGH);

digitalWrite(leftMotor2,LOW);

digitalWrite(rightMotor1,HIGH);

digitalWrite(rightMotor2,LOW);

}

/\* command motor backward \*/

void reverseMotor(void)

{

digitalWrite(eneLeftMotor,HIGH);

digitalWrite(eneRightMotor,HIGH);

digitalWrite(leftMotor1,LOW);

digitalWrite(leftMotor2,HIGH);

digitalWrite(rightMotor1,LOW);

digitalWrite(rightMotor2,HIGH);

}

/\* command motor turn left \*/

void leftMotor(void)

{

digitalWrite(eneLeftMotor,HIGH);

digitalWrite(eneRightMotor,HIGH);

digitalWrite(leftMotor1,LOW);

digitalWrite(leftMotor2,HIGH);

digitalWrite(rightMotor1,HIGH);

digitalWrite(rightMotor2,LOW);

}

/\* command motor turn right \*/

void rightMotor(void)

{

digitalWrite(eneLeftMotor,HIGH);

digitalWrite(eneRightMotor,HIGH);

digitalWrite(leftMotor1,HIGH);

digitalWrite(leftMotor2,LOW);

digitalWrite(rightMotor1,LOW);

digitalWrite(rightMotor2,HIGH);

}

/\* command motor stop \*/

void stopMotor(void)

{

digitalWrite(eneLeftMotor,LOW);

digitalWrite(eneRightMotor,LOW);

digitalWrite(leftMotor1,LOW);

digitalWrite(leftMotor2,LOW);

digitalWrite(rightMotor1,LOW);

digitalWrite(rightMotor2,LOW);

}

/\* connecting WiFi \*/

void connectWiFi()

{

Serial.println("Connecting to WIFI");

WiFi.begin(ssid, password);

while ((!(WiFi.status() == WL\_CONNECTED)))

{

delay(300);

Serial.print("..");

}

Serial.println("");

Serial.println("WiFi connected");

Serial.println("NodeMCU Local IP is : ");

Serial.print((WiFi.localIP()));

}

/\* check command received from Android Device \*/

String checkClient (void)

{

while(!client.available()) delay(1);

String request = client.readStringUntil('\r');

request.remove(0, 5);

request.remove(request.length()-9,9);

return request;

}

/\* send command echo back to android device \*/

void sendBackEcho(String echo)

{

client.println("HTTP/1.1 200 OK");

client.println("Content-Type: text/html");

client.println("");

client.println("<!DOCTYPE HTML>");

client.println("<html>");

client.println(echo);

client.println("</html>");

client.stop();

delay(1);

}

**CONCLUSION**

This project has been appreciated by all the users in the organization. It is easy to use, since it uses both the GOOGLE Speech Recognition and Manual Button mode based on the user. An user friendly app has been developed.The usage of WiFi module increases the efficiency, decreases the effort.. It has been thoroughly tested and implemented.

The project **“Voice and APP Controlled Car”** can be used to do any kind of work by just giving it voice commands by being anywhere near or far from the car only after uploading the required IP address which is easier and reduces the cost and complexity of wires .

The self-developed APP takes the IP address of the mobile with which its hotspot needs to be connected and then it can be used pretty easily and the car moves according to the users commands or the button mode.

**FUTURE SCOPE AND FURTHER ENHANCEMENTS**

In future, we would like to keep working on this project and make new additions to provide users with more advanced features and more detailed information. We have set our sights on the following additions in future-

* We would like to use an obstacle sensor to avoid accidents even when the user or the driver happens to give a wrong command.
* Even if the car meets an accident for that case we will have the car AI enabled so as soon as the cars meet with an accident it directly takes the user to a nearby hospital without the user’s commands.
* We will have a patient monitoring system within the car system to check the users health conditions. If its critical it will take the user to a nearby hospital.
* It can be used for physically challenged people and if the person is dumb we would have a motion sensor introduced with which he/she can travel here and there.
* We would like to introduce another feature with which the closest traffic system is accessed and according to its current state the car works so that there is no breach of the law and orders

**REFERENCES**

1. <https://www.youtube.com>
2. <https://www.wikipedia.org/>
3. <https://www.quora.com/>
4. <https://www.arduino.cc/>
5. <https://appinventor.mit.edu/explore/>

THANK YOU