# 

# A PROJECT REPORT

#### ON

#### 

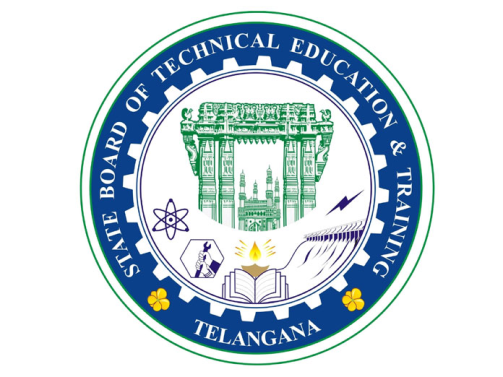
**“Wi-Fi Controlled Robot Car Using Node MCU”**

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF

**DIPLOMA**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**



# SUBMITTED TO

**STATE BOARD OF TECHNICAL EDUCATION, TELANGANA**

**SUBMITTED BY**

|  |  |
| --- | --- |
|  |  |

1. **H.ADARSH 19054-EC-049**
2. **C.ARJUN 19054-EC-056**
3. **K.ASHALATHA 19054-EC-051**
4. **MSVK.BHARGAV 19054-EC-045**
5. **Y.DIVYA 19054-EC-044**
6. **MD.HASSAN AHMED 19054-EC-037**
7. **CH.TEJASWINI 19054-EC-043**
8. **G.YASHASWINI 19054-EC-050**

**GUIDED BY**

## **MR. NARESH GOUD SIR (LECTURER/ECE)**

## (Approved by TS SBTET, Hyderabad)

## GOVERNMENT INSTITUTE OF ELECTRONICS, SECUNDERABAD

### EAST MAREDPALLY, SECUNDERABAD-500026

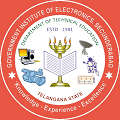
### 2019 -2022

**GOVERNMENT INSTITUTE OF ELECTRONICS, SECUNDERABAD**

**(Approved by TS SBTET, Hyderabad)**

#### East marredpally, Secunderabad-500026

#### 2019-2022



*CERTIFICATE*

# This is to certify that the project report entitled “Wi-Fi CONTROLLED ROBOT CAR USING NODE MCU” was successfully completed by Students of fifth semester Diploma in Electronics and Communication Engineering.

1. **H.ADARSH**
2. **C.ARJUN**
3. **K.ASHALATHA**
4. **Y.DIVYA**
5. **MSVK.BHARGAV**
6. **MD.HASSAN AHMED**
7. **CH.TEJASWINI**
8. **G.YASHASWINI**

In partial fulfillment of the requirement for the award of diploma in *Electronics and Communication Engineering* GIOE (TS-SBTET, HYD) during the Academic Year 2021-22 as Per Curriculum C-18 DECE.

Name of Guide Name of H.E.C.E

**B.NARESH GOUD CH. RAJENDRA PRASAD M. Tech**

External Examiner Principal

**Mr. VENKATESWARA PRASAD M. tech**

**ACKNOWLEDGMENT**

We owe our gratitude to **Sri.G.Venkateswara prasad** M.tech PRINCIPAL of

Govt. Institute of electronics Secunderabad for extending the college facilities pursuit of our project so far and for his kind patronage.

We wish to record our profound gratitude **to Sri. CH.Rajendra Prasad** M.tech **H.O.D** of electronics and instrumentation engineering of Govt.Institute of electronics secunderabad for his love, motivation and encouragement.

We are indebted to our project guide **B.Naresh** b.techlecturer in Electronics and communication engineering of Govt.Institute of Electronics. We feel it’s an immense pleasure to be indebted to our guide for her valuable support, encouragement. We thank her for her and continuous guidance towards this project.

We are very grateful and would like to express our deepest gratitude to our class in charge **B.Naresh goud** b.tech lecturer in Electronics and Communication Engineering for her help, guidance, support, continuous encouragement and the confidence she has entrusted upon us. Her constant encouragement and the motivation which she inculcated during the course of our work helped us to look forward to the timely completion of our project with more enthusiasm and boost confidence within ourselves.

We are deeply grateful to all the staff members of ECE department, of Govt.Institute of electronics, secunderabad who have helped directly or indirectly during the project.

we would like to express our acknowledgement to our **PARENTS** for their everlasting

love, dreams and sacrifices they made throughout their lives to make us see this day. We

cannot find the appropriate words which could properly describe our appreciation for their

endless devotion, support and faith in my abilities in achieving our goal.

**ABSTRACT**

Robots have suddenly started grabbing attention in this world for their applications. Wi-Fi controlled robots or NodeMCU helps to rescue individuals from collapsed buildings due to natural disasters. Currently the operations of Robots are controlled by remote sensor technology. The proposed concept of this modified Wi-Fi controlled robots or node MCU Robot used for navigation or location identification. By Introducing this navigation and location identification concepts in robots can save human beings from a risky and harmful natural disasters especially that happens in power plants.

The Field of robotics has been exponential growth with the amalgamation of multiple domains. The holistic approach is proving to be a boon, where communication engineering, mechanical engineering, embedded system and so many more are together creating robot with high flexibility. Present technologies can only control robots up to a radius of 500 meters, but our aim in this project is to target controlling a robot from a remote location which is more than 1000 meters away. We use Wi-Fi as the medium foe communication. We are achieving this operation by communicating using two computers. These computers connected to the Wi Fi pass on serial data and also communication is established between one computer located near the robot and a microcontroller present, which control its trajectory. Moreover robot will be having its own senses to dodge obstacle, which will also give about its position. Future scope of this project is to establish the same connection but make the bot intelligent and autonomous.

**TABLE OF CONTENTS P.NO**

**i)** CERTIFICATE

**ii)** ACKNOWLEDGEMENT

**iii)** ABSTRACT

**CHAPTER-1**

* 1. INTRODUCTION .
  2. HISTORY.
  3. MOTIVATION.
  4. CHALLENGES.
  5. OBJECTIVE.
  6. EXISTING SYSTEM.
  7. PROJECT AIM.

**CHAPTER-2**

**2.1** BLOCK DIAGRAM .

**2.2** DESCRIPTION OF BLOCK DIAGRAM.

**2.3** CIRCUIT DIAGRAM .

**2.4** CIRCUIT DESCRIPTION.

**CHAPTER-3**

**3.1** SOURCE CODE.

**3.2** INSTALLATION OF CODE.

**CHAPTER-4**

**4.1** LIST OF COMPONENTS USED IN WIFI CONTROLLED ROBOT USING NODE MCU

**4.2** FUNCTION AND WORKING OF EACH COMPONENT.

**CHAPTER-5**

**5.1** ADVANTAGES.

**5.2** APPLICATIONS.

**5.3**  FUTURE EXPANSION.

**CHAPTER-6**

* 1. CONCLUSION
  2. REFERENCES.

**CHAPTER-1**

# 1.1 INTRODUCTION:

# 

# In the present day, technology has so improved that an Unmanned Aerial Vehicle (UAV) also called as Drone can be controlled from a distance ranging from 2km to 20,000km. The Mars Rover, which was sent to Mars to explore various features of the planet is an autonomous robot which is programmed such that it performs the desired task as it is intended to do. There are many such systems which are controlled either by Radio Frequency transmission or by creating intelligence. Robots are called Non autonomous robots. These robots have the programming logic to do the desired task but the decision power lies in the hand of controller (human) handling the robot. Here the interface can be made using two methods: A. Wired –The connection between controller and robot is maintained using wired interface. This interfaces can be serial or parallel but the technology is transmission of signal, which is sent in the form of specific pattern to the robot to carry out the specific task, these patterns with the help of a microcontroller governing its motion. B. Wireless –Here the connection between controller and robot is achieved by wireless interface such as: ➢ Bluetooth ➢ Wi-Fi The underlying technology is transmission of signals wireless in air by transmitter which is captured by the receiver and sent to microcontroller mounted on the robot to carry out the task. At the present demand for robots in this developing world to carry out work effectively and accurately, the development of cost effective robots is necessary

# This robot is controlled by motor drive and node mc devices have transmitters and receivers to make the job. The transmitter (mobile hotshot) send the command to receivers. The input command are processed by components and jobs done by custom bot .Some also use the term industrial Internet interchangeably with IoT. Specialize in robotic solutions for the home with our snow removal robots, remote controlled lawnmowers and even a robotic cooler to provide you with refreshment and entertainment. In the military travel and operate in dangerous areas and in medical industry to assist in procedures.

**1.2 HISTORY:**

It’s often said industrial robots have made their biggest mark in the automotive world but it took many decades of refinement for them to get there. How long has it been since robots got their start? The most basic ideas originate in Leonardo's time!

The modern idea for the robot made its first appearance[in a play](http://www.umich.edu/~engb415/literature/pontee/RUR/RURsmry.html) in 1921. In this production, robots were mechanical workers who helped humans – but they eventually revolted and took over the world.

To say this is an inauspicious beginning would be an understatement. Still, real-life technology soon began to catch up with the concept.

Prototype industrial robots were deployed in General Motors facilities as early as 1961. These first robots mainly performed spot welding. Their success soon attracted attention from Ford.

In 1969, the[Stanford Arm](http://infolab.stanford.edu/pub/voy/museum/pictures/display/1-Robot.htm) was developed. With six degrees of freedom, it was capable of tasks earlier robots couldn’t perform.

In 1974, it was followed by the[Silver Arm](https://www.used-robots.com/education/the-history-of-industrial-robots) from MIT. Using embedded pressure-sensitive sensors and a microprocessor, this new arm was far more versatile. It opened the way for a years-long robot boom with 30% year-on-year growth.

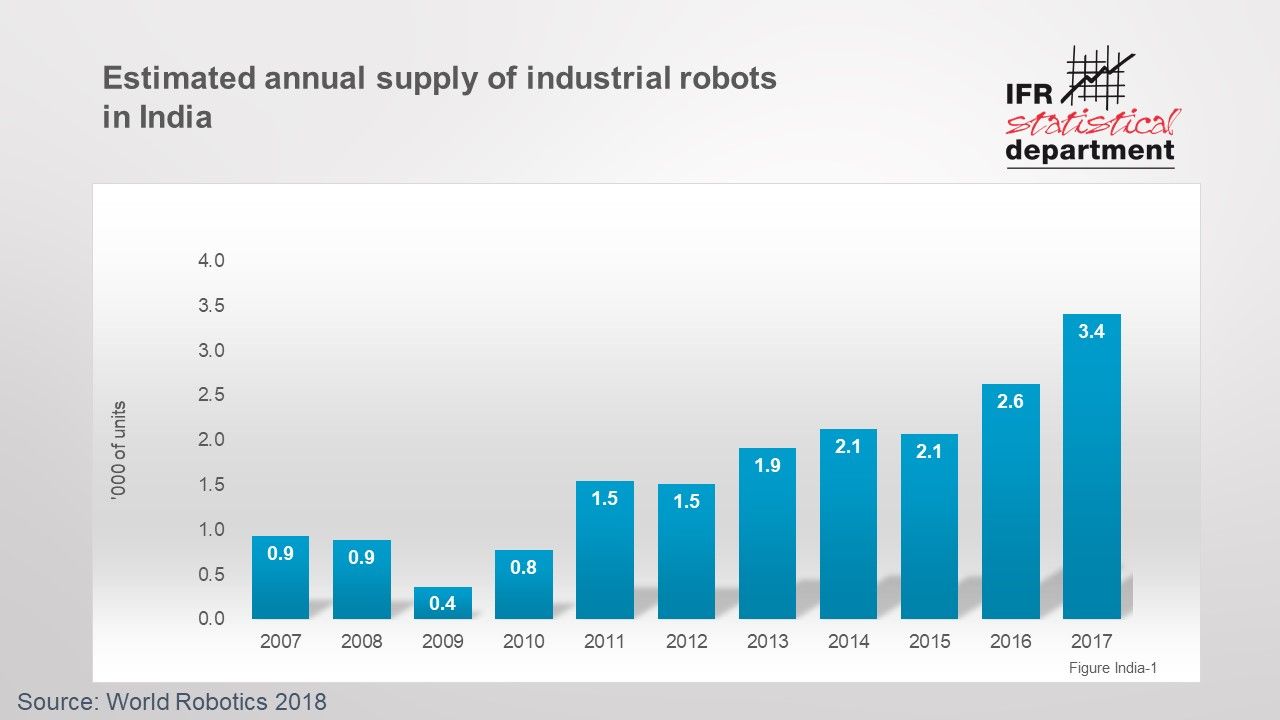
By the 1980s, billions of dollars were spent by companies worldwide to automate basic tasks in their assembly plants. Although automation system deployment did dip in the 1990s, innovative technology has caused it to rebound.

Today, robots are an essential part of making automotive plants competitive. With interest in building plants throughout China, the stock of industrial robots in that country is expected to grow rapidly. In recent years, over half of industrial robot purchases in North America have been[made by automakers](https://www.bastiansolutions.com/blog/index.php/2015/09/17/industrial-robotics-automotive-industry/#.WGRPGFMrIjK).

Today’s robots are far more sophisticated than their predecessors. Many are semi-autonomous, with machine vision systems to interact within a changing environment. Some can even work side-by-side with humans. All signs suggest we are in the middle of a new industrial robot boom!

**1.3 MOTIVATION:**

* It is observable that the robot car is an important field that the whole world seeks to make the best use of it in order to decrease the cost and use of fossil fuels such as petroleum, diesel, kerosene etc that pollutes the environment and bio diversity.
* So that, the objective now is to increase the efficiency of bot in order to decrease its high initial cost.
* For this reason, Wi-Fi car is presented as one of the solutions.



**1.4 CHALLENGES:**

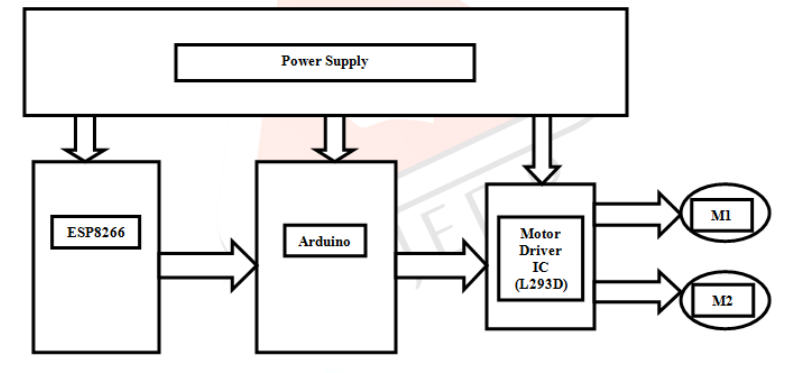
* We found some syntax errors in the code and we debugged the errors and modified the code.
* We replaced the 1500mAh batteries with 2600mAh batteries so that life of the battery will be long lasting.

**1.5 OBJECTIVE:**

The motion of robot controlling via internet is one of the easy means as it requires the user to access the designated webpage to guide it. This system can be used in Defence applications,unmanned applications. Further, the size of device can be miniaturized based upon specific applications.

# 1.6 EXISTING SYSTEM:

* Working of the Wi-Fi controlled robot is very easy, we just need to Drag or Slide the joystick in the direction, where we want to move the Robot. If we want to move the Robot in Forward direction then we need to drag the Joystick ‘circle’ in Forward direction. Like we can move the Robot in Left, Right and Backward direction by Dragging the Joystick in respective direction. Now as soon as we release the Joystick, it will come back to center and Robot. Blink App sends values from Two Axis Joystick to Arduino through Wi-Fi medium. Arduino receive the values, compare them with predefined values and move the robot accordingly in that direction



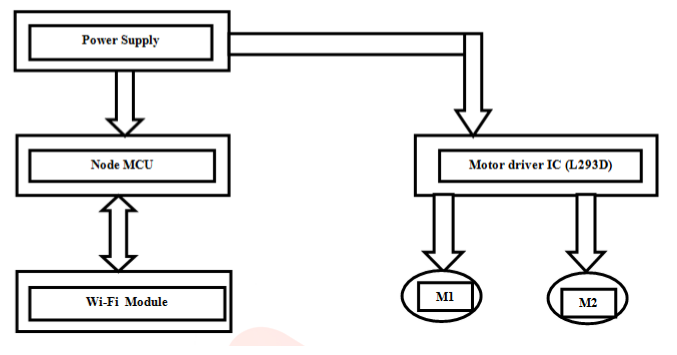
* risk from a financier’s viewpoint.

**1.7 PROJECT AIM:**

Present technologies can only control robots up to a radius of 500 meters, but our aim in this project is to target controlling a robot from a remote location which is more than 1000 miles away. We use WIFI as the medium foe communication. We are achieving this operation by communicating using two computers.

**CHAPTER-2**

**2.1 BLOCK DIAGRAM:**

****

**2.2 DESCRIPTION OF BLOCK DIAGRAM:**

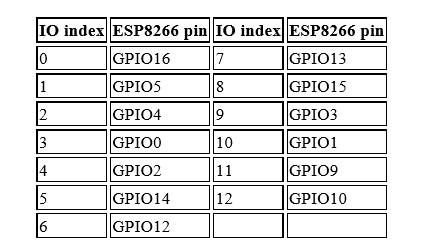
**Power Supply:-**

To drive the vehicle, we are using a 12 volt rechargeable battery which gives the maximum speed and torque. The Arduino board and Ethernet Shield is energized by a 5 volt battery which synchronizes with its operating voltage. Hence we are using dual power source to meet our requirements

**Node MCU:-**

Node MCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif NonOS SDK for ESP8266. It uses many open source projects, such as lua-cjson, and spiffs.

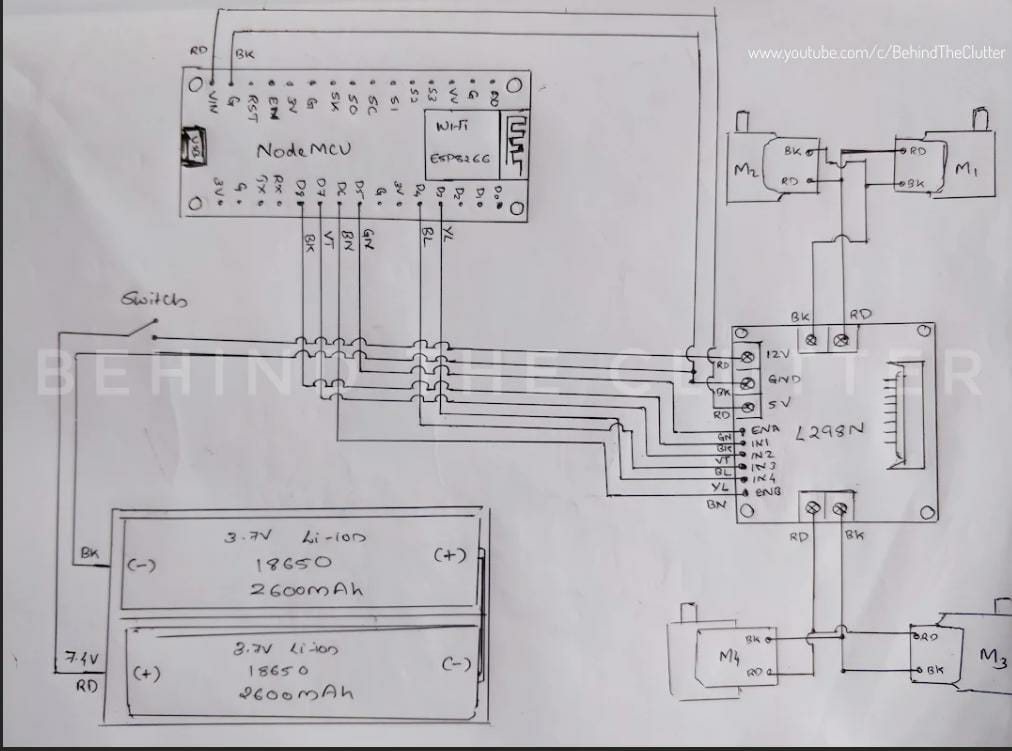
TABLE:



**Motor and Motor Driver L293D:-**

An electric motor is an electrical machine that converts electrical energy into mechanical energy. L293D is a typical Motor driver or Motor Driver IC which allows DC motor to drive on either direction. L293D is a 16-pin IC which can control a set of two DC motors simultaneously in any direction. It means that you can control two DC motor with a single L293D IC. Dual H-bridge Motor Driver Integrated Circuit(IC).The l293d can drive small and quiet big motors as well, check the Voltage Specification at the end of this page for more info.

**2.3 CIRCUIT DIAGRAM:**

****

**2.4 CIRCUIT DESCRIPTION:**

Circuit Diagram of Wi-Fi controlled robot is given below. We mainly need a NODEMCU and ESP8266 Wi-Fi module. ESP8266’s Vcc and GND pins are directly connected to 3.3V and GND of Node MCU and CH\_PD is also connected with 3.3V. Transmitter and Receiver pins of ESP8266 are directly connected to pin 2 and 3 of NODEMCU. Software Serial Library is used to allow serial communication on pin 2 and 3 of Arduino. We have already covered the Interfacing of ESP8266 Wi-Fi module to NODEMCU in detail. A L293D Motor Driver IC is used for driving DC motors Input pins of motor driver IC is directly connected to pin 8, 9, 10 and 11 of NODEMCU. And DC motors are connected at its output pins. Here we have use 9 Volt battery for driving the Circuit and DC motors.

**CHAPTER-3**

**3.1 SOURCE CODE:**

#define ENA 14 // Enable/speed motors Right GPIO14(D5)

#define ENB 12 // Enable/speed motors Left GPIO12(D6)

#define IN\_1 15 // L298N in1 motors Right GPIO15(D8)

#define IN\_2 13 // L298N in2 motors Right GPIO13(D7)

#define IN\_3 2 // L298N in3 motors Left GPIO2(D4)

#define IN\_4 0 // L298N in4 motors Left GPIO0(D3)

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

String command; //String to store app command state.

int speedCar = 800; // 400 - 1023.

int speed\_Coeff = 3;

const char\* ssid = "NodeMCU Car";

ESP8266WebServer server(80);

void setup() {

pinMode(ENA, OUTPUT);

pinMode(ENB, OUTPUT);

pinMode(IN\_1, OUTPUT);

pinMode(IN\_2, OUTPUT);

pinMode(IN\_3, OUTPUT);

pinMode(IN\_4, OUTPUT);

Serial.begin(115200);

// Connecting WiFi

WiFi.mode(WIFI\_AP);

WiFi.softAP(ssid);

IPAddress myIP = WiFi.softAPIP();

Serial.print("AP IP address: ");

Serial.println(myIP);

// Starting WEB-server

server.on ( "/", HTTP\_handleRoot );

server.onNotFound ( HTTP\_handleRoot );

server.begin();

}

void goAhead(){

digitalWrite(IN\_1, LOW);

digitalWrite(IN\_2, HIGH);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, LOW);

digitalWrite(IN\_4, HIGH);

analogWrite(ENB, speedCar);

}

void goBack(){

digitalWrite(IN\_1, HIGH);

digitalWrite(IN\_2, LOW);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, HIGH);

digitalWrite(IN\_4, LOW);

analogWrite(ENB, speedCar);

}

void goRight(){

digitalWrite(IN\_1, HIGH);

digitalWrite(IN\_2, LOW);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, LOW);

digitalWrite(IN\_4, HIGH);

analogWrite(ENB, speedCar);

}

void goLeft(){

digitalWrite(IN\_1, LOW);

digitalWrite(IN\_2, HIGH);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, HIGH);

digitalWrite(IN\_4, LOW);

analogWrite(ENB, speedCar);

}

void goAheadRight(){

digitalWrite(IN\_1, LOW);

digitalWrite(IN\_2, HIGH);

analogWrite(ENA, speedCar/speed\_Coeff);

digitalWrite(IN\_3, LOW);

digitalWrite(IN\_4, HIGH);

analogWrite(ENB, speedCar);

}

void goAheadLeft(){

digitalWrite(IN\_1, LOW);

digitalWrite(IN\_2, HIGH);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, LOW);

digitalWrite(IN\_4, HIGH);

analogWrite(ENB, speedCar/speed\_Coeff);

}

void goBackRight(){

digitalWrite(IN\_1, HIGH);

digitalWrite(IN\_2, LOW);

analogWrite(ENA, speedCar/speed\_Coeff);

digitalWrite(IN\_3, HIGH);

digitalWrite(IN\_4, LOW);

analogWrite(ENB, speedCar);

}

void goBackLeft(){

digitalWrite(IN\_1, HIGH);

digitalWrite(IN\_2, LOW);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, HIGH);

digitalWrite(IN\_4, LOW);

analogWrite(ENB, speedCar/speed\_Coeff);

}

void stopRobot(){

digitalWrite(IN\_1, LOW);

digitalWrite(IN\_2, LOW);

analogWrite(ENA, speedCar);

digitalWrite(IN\_3, LOW);

digitalWrite(IN\_4, LOW);

analogWrite(ENB, speedCar);

}

void loop() {

server.handleClient();

command = server.arg("State");

if (command == "F") goAhead();

else if (command == "B") goBack();

else if (command == "L") goLeft();

else if (command == "R") goRight();

else if (command == "I") goAheadRight();

else if (command == "G") goAheadLeft();

else if (command == "J") goBackRight();

else if (command == "H") goBackLeft();

else if (command == "0") speedCar = 400;

else if (command == "1") speedCar = 470;

else if (command == "2") speedCar = 540;

else if (command == "3") speedCar = 610;

else if (command == "4") speedCar = 680;

else if (command == "5") speedCar = 750;

else if (command == "6") speedCar = 820;

else if (command == "7") speedCar = 890;

else if (command == "8") speedCar = 960;

else if (command == "9") speedCar = 1023;

else if (command == "S") stopRobot();

}

void HTTP\_handleRoot(void) {

if( server.hasArg("State") ){

Serial.println(server.arg("State"));

}

server.send ( 200, "text/html", "" );

delay(1);

}

**3.2 CODE INSTALLATION:**

The Arduino Web Editor allows you to write code and upload sketches to any official Arduino board from your web browser (Chrome, Firefox, Safari and Edge). We used Google Chrome.

This IDE (Integrated Development Environment) is part of Arduino Create, an online platform that enables developers to write code, access tutorials, configure boards, and share projects. Designed to provide users with a continuous workflow, Arduino Create connects the dots between each part of a developer's journey from inspiration to implementation. Meaning, you now have the ability to manage every aspect of your project right from a single dashboard.

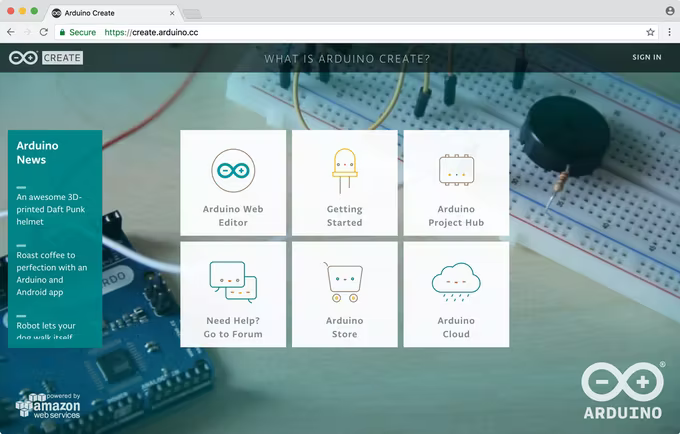


Figure: 3.1 Arduino web editor page

The Arduino Web Editor is hosted online, therefore it will always be up-to-date with the latest features and support for new boards. This IDE lets you write code and save it to the cloud, always backing it up and making it accessible from any device. It automatically recognizes any Arduino and Genuino board connected to your PC, and configures itself accordingly. An Arduino account is all you need to get started.

**Sign up to Arduino**

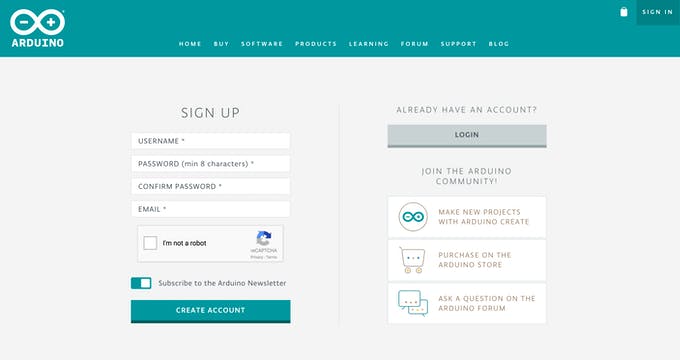
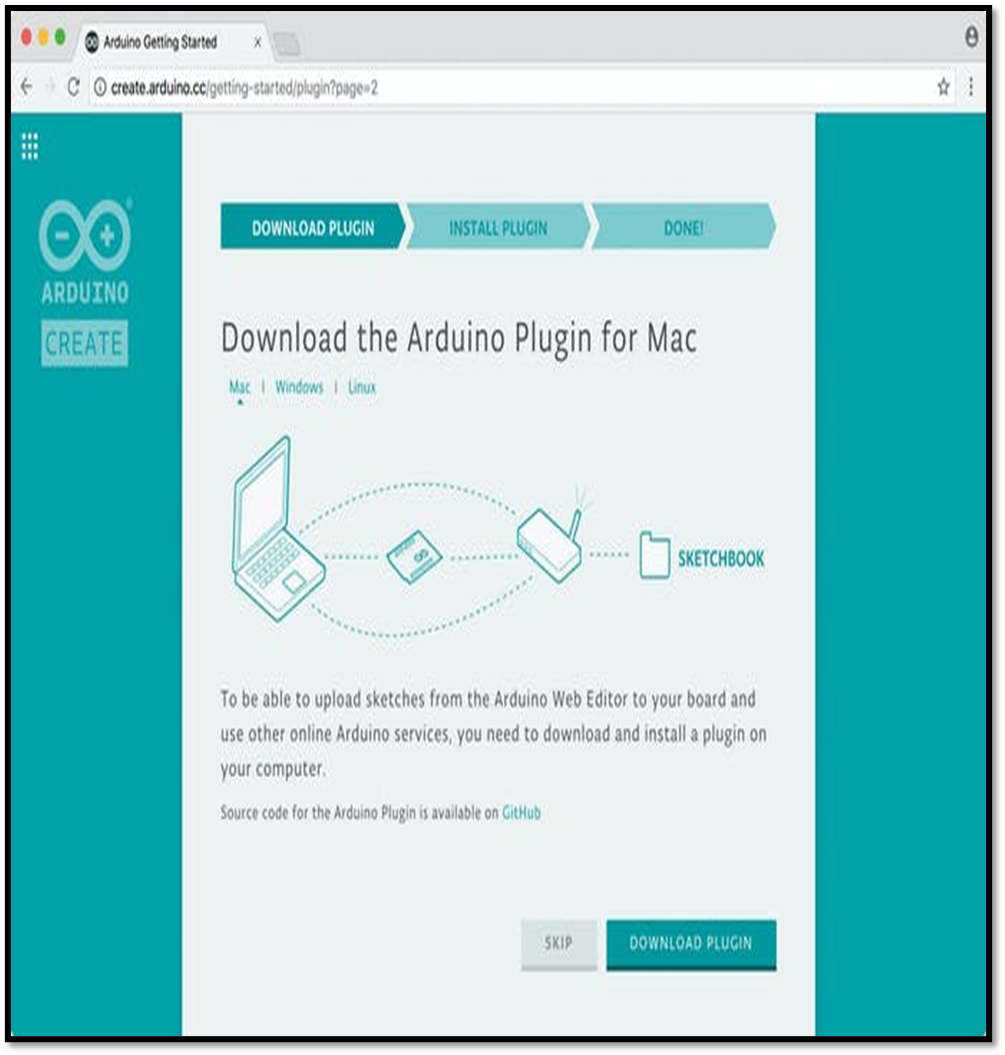
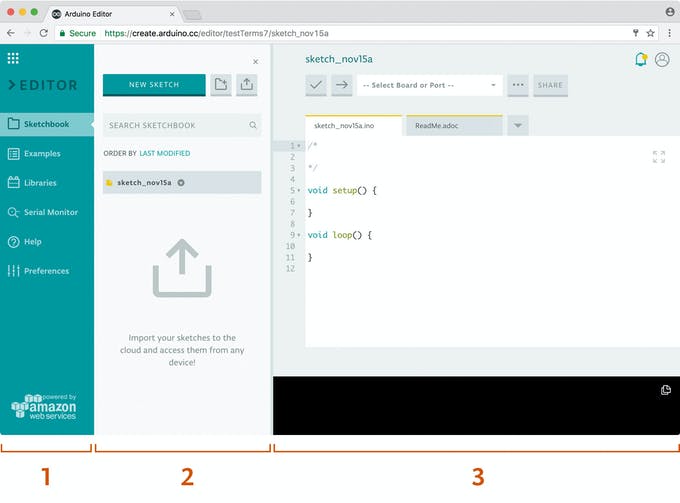
Go to the Arduino page and complete the registration form, then hit the ‘create account’ button. You will receive an email with a link to activate your account. Select the link and a new page will open with your confirmed account information.

Figure:3.2 sign in page

Note that you can also use this account to write posts on [Arduino’s official Forum](http://forum.arduino.cc/index.php?board=101.0), buy products on the [online store](https://store.arduino.cc/), add comments on the [blog](https://blog.arduino.cc/), as well as create tutorials on Arduino project hub (YUY!). **Log in the Arduino Web Editor** Once you have successfully registered for an Arduino account, go to create.arduino.cc/editor. After accepting the term and conditions, you should receive an email with a link to the Forum where you can report any bugs you may encounter and share your feedback.Once you have successfully registered for an Arduino account, go to [create.arduino.cc/editor.](https://create.arduino.cc/editor) After accepting the term and conditions, you should receive an email with a link to the [Forum](http://forum.arduino.cc/index.php?board=101.0) where you can report any bugs you may encounter and share your feedback. **Pick your Platform: Windows, Mac or Linux** The Arduino Web Editor can run on a variety of Platforms. If you are using Windows, Mac or Linux follow a simple flow to install the [Arduino Web Editor plugin](https://github.com/arduino/arduino-create-agent), which permits you to upload sketches from the browser onto your boards. At the end of the plugin installation is redirected to next page.

**Pick your Platform: Chrome Os** Using the Arduino Web Editor on Chrome OS requires the Arduino Create App to be able to upload sketches to a board. You can [install the Create App in the Chrome Web Store.](https://chrome.google.com/webstore/detail/elmgohdonjdampbcgefphnlchgocpaij) **Arduino Web Editor anatomy.** After logging in, you are ready to start using the Arduino Web Editor. The web app is divided into three main columns.

**The first column lets you to navigate between:**

* Your Sketchbook: a collection of all your sketches (‘sketch’ is what programs you upload on your board are called).upload on your board are called).

Figure 3.3 Arduino software menu

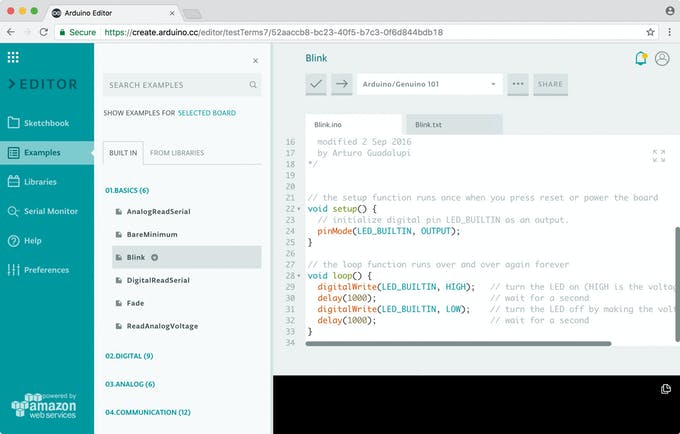
* Examples: read-only sketches that demonstrate all the basic Arduino commands (built-in tab), and the behavior of your libraries (from the libraries tab)
* Libraries: packages that can be included to your sketch to provide extra functionalities
* Serial monitor: a feature that enables you to receive and send data to your board via the USB cable
* Help: helpful links and a glossary about Arduino terms
* Preferences: options to customize the look and behavior of your editor, such as text size and color theme

When selected, every menu item shows its options in a side panel (second column).

The third column, the code area, is the one you will use the most. Here, you can write code, verify it and upload it to your boards, save your sketches on the cloud, and share them with anyone you want.

Make your board blink from the browser

When you’ve set up your online IDE, make sure your computer can talk to the board.

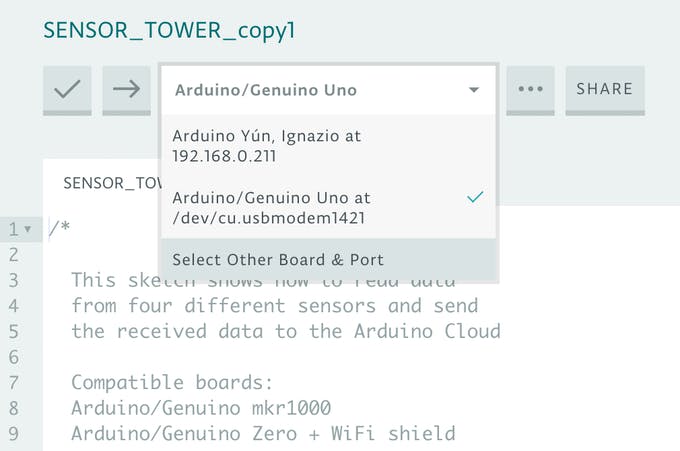


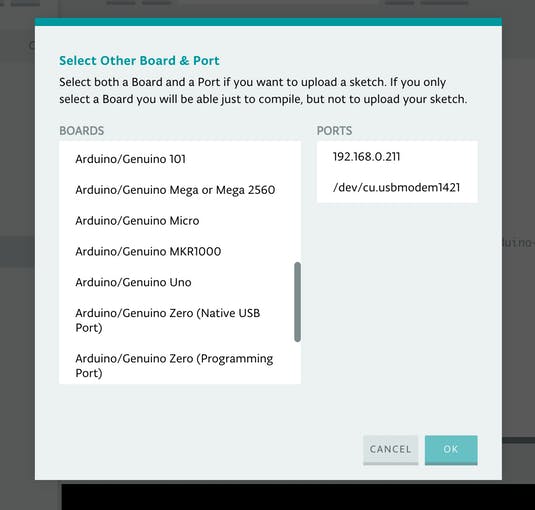
* Double check if the Web Editor is displayed the way you like, check the *Preferences panel* for a few options
* Connect your Arduino or Genuino board to your computer. Boards and serial ports are auto-discovered and selectable in a single dropdown. Pick the Arduino/Genuino board you want to upload to from the list.
* For easy understanding: Choose ‘examples on the menu on the left, then ‘basics and blink’. The [Blink sketch](https://create.arduino.cc/example/builtin/01.Basics%5CBlink/Blink/preview) will be displayed in the code area.
* To upload it to your board, press the ’upload’ button (arrow right) near the dropdown. A ‘busy’ label replaces the buttons during code verifying and uploading. If the upload is successful, the message success: done uploading ‘ will appear in the bottom output area.
* Once the upload is complete, you should then see on your board the yellow LED with an L next to it start blinking. If this is the case that indicates, you have successfully programmed your board to blink its on-board LED!
* Sometimes your brand-new Arduino/Genuino is already programmed with the Blink sketch, so you can’t tell if you are truly in control. If this is the case, change the delay time by changing the number in the parenthesis to 100, and upload the Blink sketch again. Now the LED should blink much faster.
* **What boards are supported?**

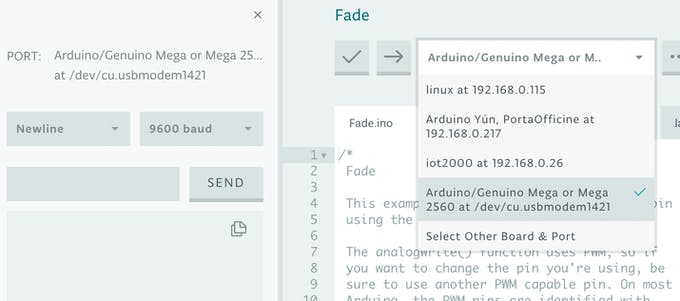
On the online IDE we are able to automatically detect the kind of board and the port it is connected to without you having to individually select them.

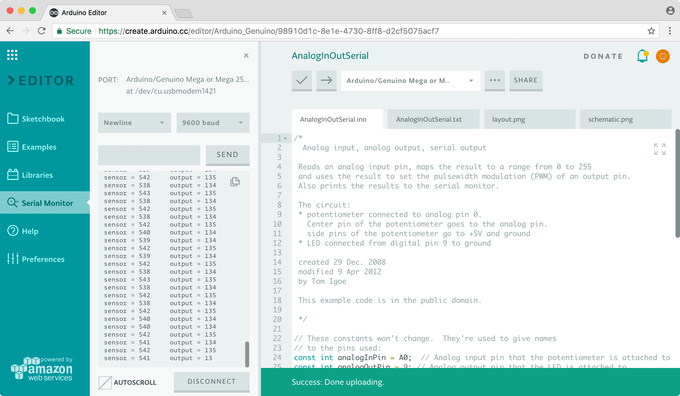
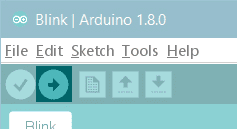
We are currently supporting only Official Arduino/Genuino boards, and a few Certified and At Heart boards. You can find the full list of supported boards by clicking on the boards dropdown and hitting ‘select other board & port’.

In this popup window you also have the freedom to select any board if you want to verify if your code works for a specific product, but you don't have it at hand. If the board you have connected is not recognized, you can also manually select it here, don't forget to also select the port.

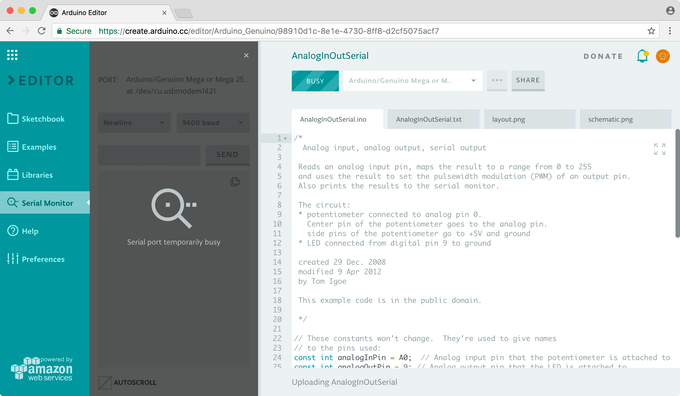




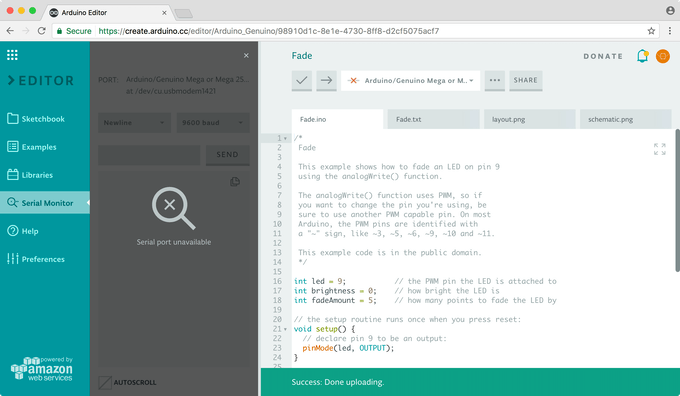
The full list of supported hardware is displayed in the popup window. We are working on expanding this list. **Serial Monitor** on the Arduino Web Editor the Serial Monitor is available on the side bar, when you click on it a panel will be displayed. The Serial Monitor reads the board selected at that moment in the Board dropdown.If you need to change the board (for instance you have more than one board connected to your PC), just select the one you would like to read from in the dropdown. The PORT info’s in the panel will be updated as a result.



The code is uploading the panel will be grayed out as in the following image.



If no board is connected at that time, the Serial Monitor will be grayed out and unavailable.



If you are using other programs that communicate via that port (for instance Processing or theDesktop IDE), a message saying that the port is used by another software will appear. If you wish to use the port with another software or just want the flow of output values to stop, you can press the *Disconnect* button on the panel.

## **Step 1: Connect Your NodeMCU to the Computer**

Use the USB cable to connect your NodeMCU to the computer,you will see the blue onboard LED flicker when powered up, but they will not stay lit

## **Step 2: Install the COM/Serial Port Driver**

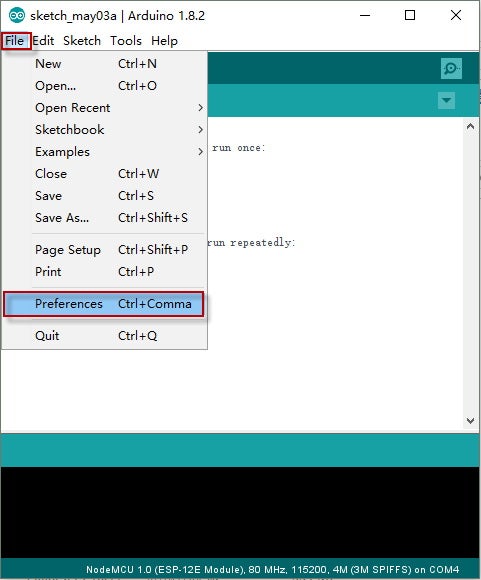
In order to upload code to the ESP8266 and use the serial console, connect any data-capable micro USB cable to ESP8266 IOT Board and the other side to your computer’s USB port.

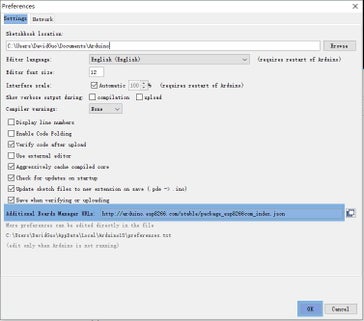
The new version NodeMCUv1.0 comes with the CP2102 serial chip,you can download and install the driver from:[https://www.silabs.com/products/development-tools/...](https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers). The NodeMCUv0.9 comes with the CH340 serial chip,you can download and install the driver from:[https://github.com/nodemcu/nodemcu-devkit/tree/mas...](https://github.com/nodemcu/nodemcu-devkit/tree/master/Drivers)

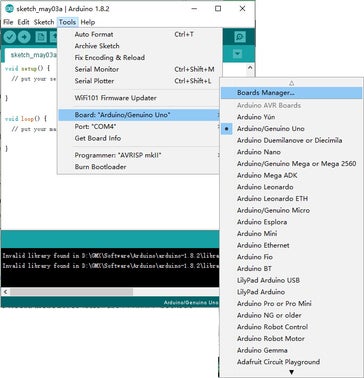
## **Step 3: ​Install the Arduino IDE 1.6.4 or Greater**

Download Arduino IDE from Arduino.cc (1.6.4 or greater) – don’t use 1.6.2! You can use your existing IDE if you have already installed it. You can also try downloading the ready-to-go package from the ESP8266-Arduino project, if the proxy is giving you problems

## **Step 4: ​Install the ESP8266 Board Package**

[](https://content.instructables.com/ORIG/F75/FYRT/J7QGGU8U/F75FYRTJ7QGGU8U.jpg?auto=webp&frame=1&fit=bounds&md=1edfd54931b4cbdba26dfda43e918119)

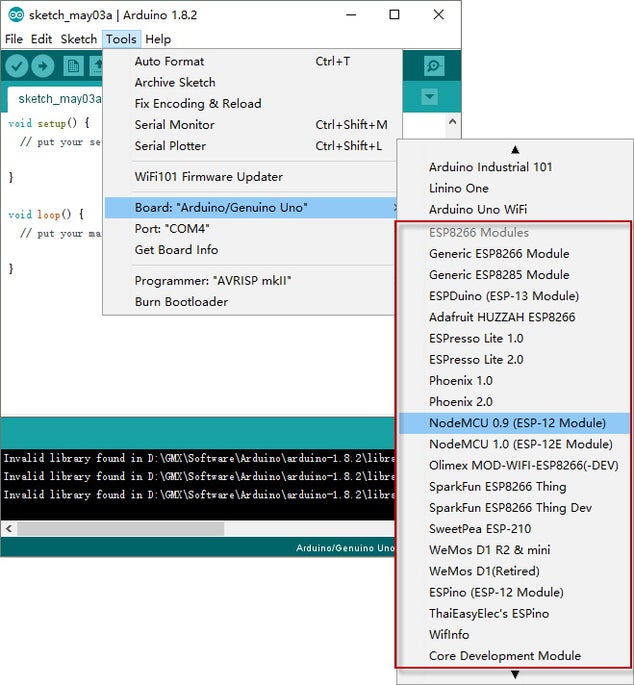
[](https://content.instructables.com/ORIG/F4L/BKVJ/J7QGGU8X/F4LBKVJJ7QGGU8X.jpg?auto=webp&frame=1&fit=bounds&md=15adda4693ff5b192f84b8b3d893cc25)

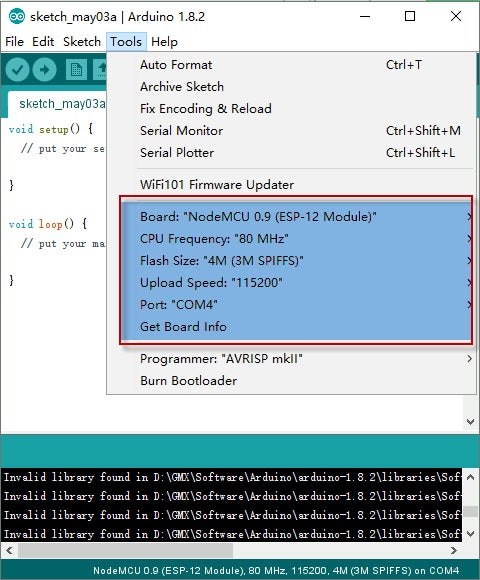
[](https://content.instructables.com/ORIG/FKY/D5SW/J7QGGU8Y/FKYD5SWJ7QGGU8Y.jpg?auto=webp&frame=1&fit=bounds&md=a07a01401c17a724e073dfe27fbbc28f)

2 More Images

Enter http://arduino.esp8266.com/stable/package\_esp8266... into Additional Board Manager URLs field in the Arduino v1.6.4+ preferences (Open Arduino IDE–>File–>Perferences–>Settings). Enter the link and click “OK” to save your changes. Next, use the Board Manager to install the ESP8266 package Enter the Boards Manager and find the board type as below: Scroll the Broads Manager screen down to the bottom, you will see A module called “esp8266 by esp8266 Community” (see following picture), select the latest version and click “Install“. The ESP8266 package has benn installed successfully. Note: You’d better close the Arduino IDE and restart it again.

## **Step 5: Setup ESP8266 Support**

[](https://content.instructables.com/ORIG/F97/U1S7/J7QGGU99/F97U1S7J7QGGU99.jpg?auto=webp&frame=1&fit=bounds&md=736510e12dd6fa679aaa14300db3dcba)

[](https://content.instructables.com/ORIG/FQI/6TB5/J7QGGU9J/FQI6TB5J7QGGU9J.jpg?auto=webp&frame=1&fit=bounds&md=c538609e7ca0c66ec8e17ec1c9e8b56d)

When you’ve restarted, select NodeMCU 0.9 (or NodeMCU 1.0) from the Tools->Board dropdown Config the Board menu and choose the right Port for your device. CPU Frequency：80MHz,Flash Size：4M（3M SPIFFS）,Upload Speed：115200 Now just proceed as the Arduino: Start your sketching! Note: 115200 baud upload speed is a good place to start – later on you can try higher speeds but 115200 is a good safe place to start.

**CHAPTER-4**

**5.1 COMPONETS REQUIRED:**

The main components which are used to develop WIFI controlled car using node mcu are:

|  |  |
| --- | --- |
| **COMPONENTS** | **QUANTITY** |
| NODE MCU ESP8266 | 1 |
| 4 wheel chassis set | 1 |
| 2600 mAh Li-ion batteries | 3 |
| L298N motor driver IC | 1 |
| Connecting wires | As required |
| 3 set battery holder | 1 |
| Switch | 1 |
| Jumpers | 1 |
| USB cable | 1 |

**5.2 DESCRIPTION OF COMPONENTS:**

* **NODE MCU :**

Node MCU is an open source firmware for which open source [prototyping](https://en.wikipedia.org/wiki/Prototyping) board designs are available. The name "Node MCU" combines "[node](https://en.wikipedia.org/wiki/Node_(computer_science))" and "MCU" ([micro-controller](https://en.wikipedia.org/wiki/Micro-controller) unit).[[8]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-developer.ibm.com-8) The term "Node MCU" strictly speaking refers to the firmware rather than the associated [development kits](https://en.wikipedia.org/wiki/Development_kits).

Both the firmware and prototyping board designs are [open source](https://en.wikipedia.org/wiki/Open_source).[[8]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-developer.ibm.com-8)

The firmware uses the [Lua](https://en.wikipedia.org/wiki/Lua_(programming_language)) scripting language. The firmware is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson[[9]](https://en.wikipedia.org/wiki/NodeMCU" \l "cite_note-9) and [SPIFFS](https://en.wikipedia.org/w/index.php?title=SPIFFS&action=edit&redlink=1).[[10]](https://en.wikipedia.org/wiki/NodeMCU#cite_note-spiffs-10) Due to resource constraints, users need to select the modules relevant for their project and build a firmware tailored to their needs. Support for the 32-bit [ESP32](https://en.wikipedia.org/wiki/ESP32) has also been implemented.

The prototyping hardware typically used is a circuit board functioning as a [dual in-line package](https://en.wikipedia.org/wiki/Dual_in-line_package) (DIP) which integrates a USB controller with a smaller surface-mounted board containing the MCU and antenna. The choice of the DIP format allows for easy prototyping on [breadboards](https://en.wikipedia.org/wiki/Breadboard). The design was initially based on the ESP-12 module of the [ESP8266](https://en.wikipedia.org/wiki/ESP8266), which is a Wi-Fi SoC integrated with a [Tensilica](https://en.wikipedia.org/wiki/Tensilica" \o "Tensilica) Xtensa LX106 core, widely used in IoT applications

### Types:

There are two available versions of NodeMCU as version 0.9 & 1.0 where the version 0.9 contains **ESP-12** and version 1.0 contains **ESP-12E** where E stands for "Enhanced".



## **ESP-12E Module**

The development board equips the ESP-12E module containing ESP8266 chip having **Tensilica Xtensa® 32-bit LX106 RISC microprocessor** which operates at **80 to 160 MHz** adjustable clock frequency and supports **RTOS**.

There’s also 128 KB RAM and 4MB of Flash memory (for program and data storage) just enough to cope with the large strings that make up web pages, JSON/XML data, and everything we throw at IoT devices nowadays.

The ESP8266 Integrates 802.11b/g/n HT40 Wi-Fi transceiver, so it can not only connect to a WiFi network and interact with the Internet, but it can also set up a network of its own, allowing other devices to connect directly to it. This makes the ESP8266 NodeMCU even more versatile.

As the operating voltage range of ESP8266 is **3V to 3.6V**, the board comes with a LDO voltage regulator to keep the voltage steady at 3.3V. It can reliably supply up to 600mA, which should be more than enough when ESP8266 pulls as much as **80mA during RF transmissions**. The output of the regulator is also broken out to one of the sides of the board and labeled as 3V3. This pin can be used to supply power to external components.

**Power to the ESP8266 NodeMCU**is supplied via the **on-board MicroB USB connector**. Alternatively, if you have a regulated 5V voltage source, the **VIN pin** can be used to directly supply the ESP8266 and its peripherals.

The ESP8266 NodeMCU features two buttons. One marked as **RST** located on the top left corner is the Reset button, used of course to reset the ESP8266 chip. The other **FLASH** button on the bottom left corner is the download button used while upgrading firmware.

### History:

NodeMCU was created shortly after the [ESP8266](https://en.m.wikipedia.org/wiki/ESP8266) came out. On December 30, 2013, [Espressif Systems](https://en.m.wikipedia.org/w/index.php?title=Espressif_Systems&action=edit&redlink=1" \o "Espressif Systems (page does not exist)) began production of the ESP8266. NodeMCU started on 13 Oct 2014, when Hong committed the first file of nodemcu-firmware to GitHub. Two months later, the project expanded to include an open-hardware platform when developer Huang R committed the [gerber](https://en.m.wikipedia.org/wiki/Gerber_format" \o "Gerber format) file of an ESP8266 board, named devkit v0.9. Later that month, Tuan PM ported [MQTT](https://en.m.wikipedia.org/wiki/MQTT) client library from [Contiki](https://en.m.wikipedia.org/wiki/Contiki) to the ESP8266 SoC platform, and committed to NodeMCU project, then NodeMCU was able to support the MQTT IoT protocol, using Lua to access the MQTT broker. Another important update was made on 30 Jan 2015, when Devsaurus ported the u8glib to the NodeMCU project, enabling NodeMCU to easily drive LCD, Screen, OLED, even VGA displays.

In the summer of 2015 the original creators abandoned the firmware project and a group of independent contributors took over. By the summer of 2016 the NodeMCU included more than 40 different modules.



### Pins:

NodeMCU provides access to the [GPIO](https://en.m.wikipedia.org/wiki/General-purpose_input/output) (General Purpose Input/Output) and a pin mapping table is part of the API documentation.[[19]](https://en.m.wikipedia.org/wiki/NodeMCU#cite_note-19)

|  |  |
| --- | --- |
| **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 |

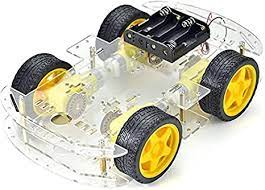
### ****Advantages of NodeMCU****

* Low cost
* Integrated support for WIFI network
* Reduced size of the board
* Low energy consumption

### ****Disadvantages****

* Need to learn a new language and IDE
* Reduced pinout
* Scarce documentation
* **4 Wheel chassis kit :**

4-Wheel Robot Chassis Kit, an easy to assemble and use robot chassis platform. The Chassis kit provides you with everything you need to give your robot a fast four wheel drive platform with plenty of room for expansion to add various sensors and controller.

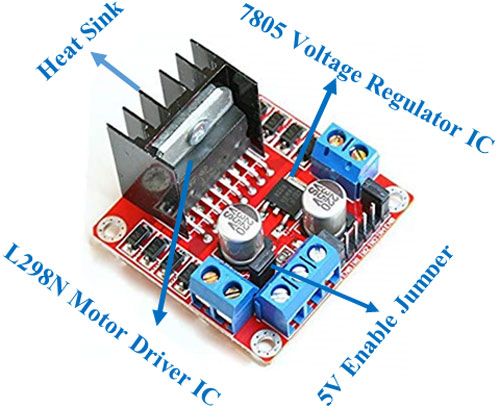


* **L298N motor driver IC:**

The L298N is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors. That means it can individually drive up to two motors making it ideal for building two-wheel robot platforms.

The L298N Motor Driver module consists of an L298 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 & 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.



At the heart of the module is the big, black chip with chunky heat sink is an L298N.

The L298N is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors. That means it can individually drive up to two motors making it ideal for building two-wheel robot platforms.

The L298N motor driver IC actually has two input power pins viz. ‘Vss’ and ‘Vs’.

From Vs pin the H-Bridge gets its power for driving the motors which can be 5 to 35V. Vss is used for driving the logic circuitry which can be 5 to 7V. And they both sink to a common ground named ‘GND’.

The module has an on-board 78M05 5V regulator from STMicroelectronics. It can be enabled or disabled through a jumper.

When this jumper is in place, the 5V regulator is enabled, supplying logic power supply(Vss) from the motor power supply(Vs). In this case, 5V input terminal acts as an output pin and delivers 5V 0.5A. You can use it to power up the Arduino or other circuitry that requires 5V power supply

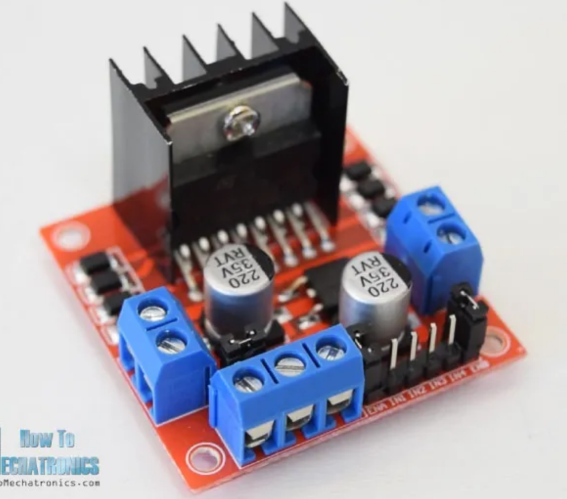
Let’s take a closer look at the pinout of L298N module and explain how it works. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output.

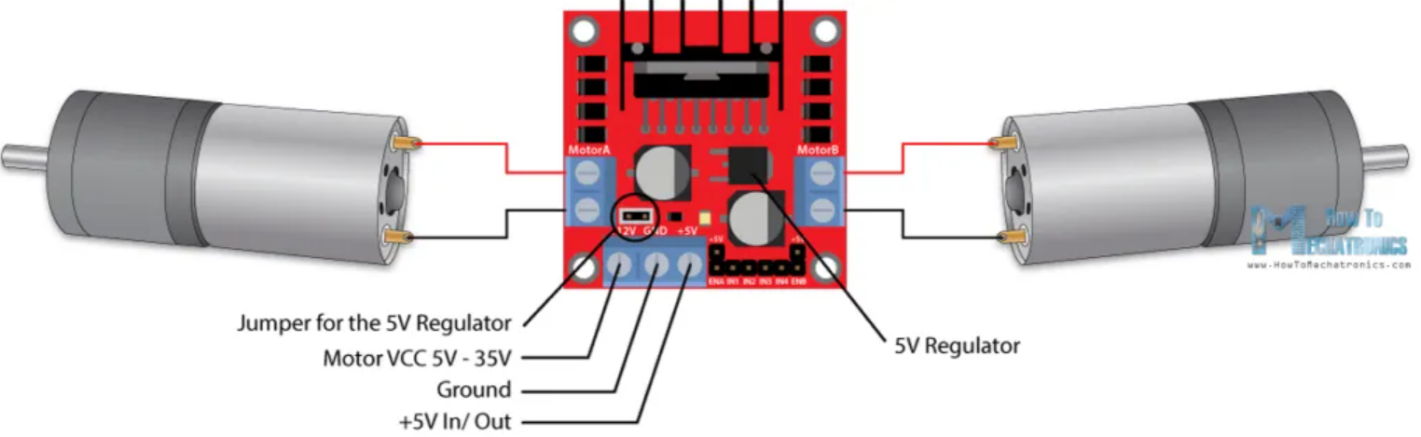
This depends on the voltage used at the motors VCC. The module have an onboard 5V regulator which is either enabled or disabled using a jumper. If the motor supply voltage is up to 12V we can enable the 5V regulator and the 5V pin can be used as output, for example for powering our Arduino board. But 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.

Next are the logic control inputs. The Enable A and Enable B pins are used for enabling and controlling the speed of the motor. If a jumper is present on this pin, the motor will be enabled and work at maximum speed, and if we remove the jumper we can connect a PWM input to this pin and in that way control the speed of the motor. If we connect this pin to a Ground the motor will be disabled.

Next, the Input 1 and Input 2 pins are used for controlling the rotation direction of the motor A, and the inputs 3 and 4 for the motor B. Using these pins we actually control the switches of the H-Bridge inside the L298N IC. If input 1 is LOW and input 2 is HIGH the motor will move forward, and vice versa, if input 1 is HIGH and input 2 is LOW the motor will move backward. In case both inputs are same, either LOW or HIGH the motor will stop. The same applies for the inputs 3 and 4 and the motor B.





**Pin Configuration:**

* ENA - D5
* ENB - D6
* IN1 - D8
* IN2 - D7
* IN3 - D4
* IN4 - D3

## **Features of the L298N motor driver Module**

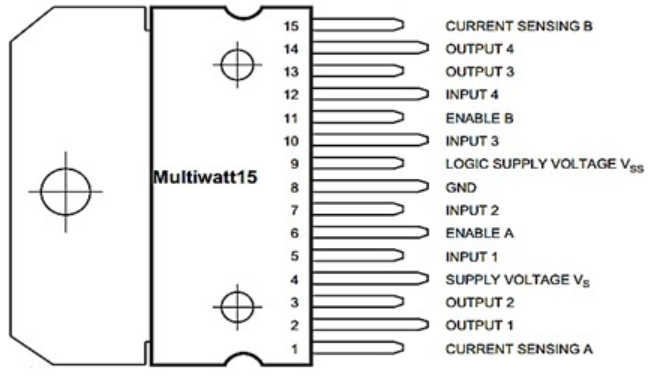
L298N is an integrated circuit multi watt 15 package and capable of giving high voltage. It is a high current dual [**full-bridge**](https://microcontrollerslab.com/how-to-make-h-bridge-using-ir2110/) driver that is designed to accept standard TTL logic levels. It can drive inductive loads e.g [**relays**](https://microcontrollerslab.com/electromechanical-relays-interfacing-with-microcontrollers/), solenoids, motors (DC and stepping motor), etc.

Its basic features are:

* Maximum supply voltage 46V
* Maximum output[**DC current**](https://microcontrollerslab.com/digital-ammeter-pic/) 4A
* Low saturation voltage
* Over-temperature protection
* Logical “0” Input Voltage up to 1.5 V

## **PIN DIAGRAM of L298N**

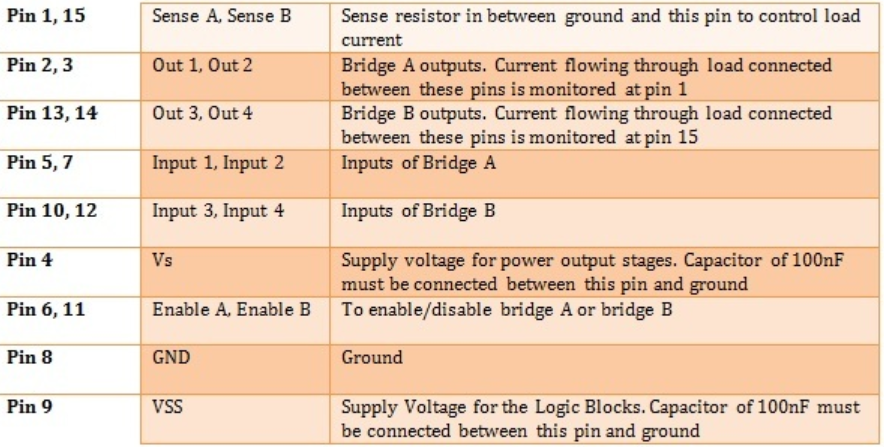
The pin diagram in top view for L298N is given below:



## **Motor Driver Internal Circuit**

## 

## **PIN DESCRIPTION of L298N module**



* **Switch:**

Switch is used control the battery.  Many specialized forms exist, such as the [toggle switch](https://en.wikipedia.org/wiki/Toggle_switch), [rotary switch](https://en.wikipedia.org/wiki/Rotary_switch), [mercury switch](https://en.wikipedia.org/wiki/Mercury_switch), [push-button](https://en.wikipedia.org/wiki/Push-button) switch, [reversing switch](https://en.wikipedia.org/wiki/Reversing_switch), [relay](https://en.wikipedia.org/wiki/Relay), and [circuit breaker](https://en.wikipedia.org/wiki/Circuit_breaker).In this we are using rocket type of switch.



* **Connecting wires :**

Connecting wires allows an electrical current to travel from one point on a circuit to another because electricity needs a medium through which it can move. Most of the connecting wires are made up of copper or aluminum.



* **Jumper wires :**

A jump 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.

**Types of Jumper Wires:**

Jumper wires typically come in three versions:

* male-to-male
* male-to-female and
* female -to-female.

**MALE-TO-MALE:**

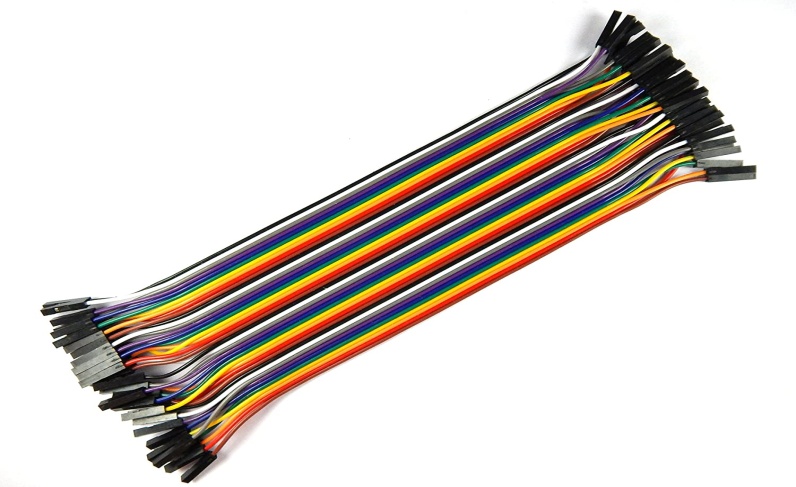


Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. Male-to-male jumper wires are the most common and what you likely will use most often. When connecting two ports on a breadboard, a male-to-male wire is what you'll need.

**MALE-FEMALE:**

The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into.

**FEMALE-FEMALE:**

****

female to female jumper wires used in connecting the female header pin of Arduino or any development board to plug in any other development board.

**BATTERIES:**

* A battery is a source of electric power consisting of one or more electrochemical cells with external connections for powering electrical devices such as flashlights, mobile phones, and electric cars.
* A lithium-ion battery or Li-ion battery is a type of [rechargeable battery](https://en.wikipedia.org/wiki/Rechargeable_battery) composed of cells in which [lithium](https://en.wikipedia.org/wiki/Lithium) [ions](https://en.wikipedia.org/wiki/Ion) move from the negative [electrode](https://en.wikipedia.org/wiki/Electrode) through an [electrolyte](https://en.wikipedia.org/wiki/Electrolyte) to the positive electrode during discharge and back when charging. Li-ion cells use an [intercalated](https://en.wikipedia.org/wiki/Intercalation_(chemistry)) lithium [compound](https://en.wikipedia.org/wiki/Chemical_compound) as the material at the positive electrode and typically [graphite](https://en.wikipedia.org/wiki/Graphite) at the negative electrode. Li-ion batteries have a high [energy density](https://en.wikipedia.org/wiki/Energy_density), no [memory effect](https://en.wikipedia.org/wiki/Memory_effect) (other than [LFP cells](https://en.wikipedia.org/wiki/Lithium_iron_phosphate_battery)) and low [self-discharge](https://en.wikipedia.org/wiki/Self-discharge). Cells can be manufactured to prioritize either energy or power density. They can however be a safety hazard since they contain flammable electrolytes and if damaged or incorrectly charged can lead to explosions and fires.
* A prototype Li-ion battery was developed by [Akira Yoshino](https://en.wikipedia.org/wiki/Akira_Yoshino) in 1985, based on earlier research by [John Goodenough](https://en.wikipedia.org/wiki/John_Goodenough), [M. Stanley Whittingham](https://en.wikipedia.org/wiki/M._Stanley_Whittingham), [Rachid Yazami](https://en.wikipedia.org/wiki/Rachid_Yazami) and [Koichi Mizushima](https://en.wikipedia.org/wiki/Koichi_Mizushima_(scientist)) during the 1970s–1980s,[[11]](https://en.wikipedia.org/wiki/Lithium-ion_battery#cite_note-ieee-11)[[12]](https://en.wikipedia.org/wiki/Lithium-ion_battery#cite_note-nobel-12)[[13]](https://en.wikipedia.org/wiki/Lithium-ion_battery#cite_note-NIMS-13) and then a commercial Li-ion battery was developed by a [Sony](https://en.wikipedia.org/wiki/Sony) and [Asahi Kasei](https://en.wikipedia.org/wiki/Asahi_Kasei) team led by Yoshio Nishi in 1991. Lithium-ion batteries are commonly used for [portable electronics](https://en.wikipedia.org/wiki/Portable_electronics) and [electric vehicles](https://en.wikipedia.org/wiki/Electric_vehicle) and are growing in popularity for military and [aerospace](https://en.wikipedia.org/wiki/Aerospace) applications

## **Uses:**

The vast majority of commercial Li-ion batteries are used in [consumer electronics](https://en.wikipedia.org/wiki/Consumer_electronics) and [electric vehicles](https://en.wikipedia.org/wiki/Electric_vehicle). Such devices include:

* *Portable devices*: these include [mobile phones](https://en.wikipedia.org/wiki/Mobile_phone) and [smartphones](https://en.wikipedia.org/wiki/Smartphones), [laptops](https://en.wikipedia.org/wiki/Laptop) and [tablets](https://en.wikipedia.org/wiki/Tablet_computer), [digital cameras](https://en.wikipedia.org/wiki/Digital_camera) and [camcorders](https://en.wikipedia.org/wiki/Camcorder), [electronic cigarettes](https://en.wikipedia.org/wiki/Electronic_cigarette), [handheld game consoles](https://en.wikipedia.org/wiki/Handheld_game_console) and [torches (flashlights)](https://en.wikipedia.org/wiki/Flashlight).
* [*Power tools*](https://en.wikipedia.org/wiki/Power_tools): Li-ion batteries are used in tools such as [cordless drills](https://en.wikipedia.org/wiki/Cordless_drill), [sanders](https://en.wikipedia.org/wiki/Sander), [saws](https://en.wikipedia.org/wiki/Saw), and a variety of garden equipment including [whipper-snippers](https://en.wikipedia.org/wiki/Whipper-snipper) and [hedge trimmers](https://en.wikipedia.org/wiki/Hedge_trimmer).
* [*Electric vehicles*](https://en.wikipedia.org/wiki/Electric_vehicle): [electric vehicle batteries](https://en.wikipedia.org/wiki/Electric_vehicle_battery) are used in [electric cars](https://en.wikipedia.org/wiki/Electric_car), [hybrid vehicles](https://en.wikipedia.org/wiki/Hybrid_vehicle), [electric motorcycles and scooters](https://en.wikipedia.org/wiki/Electric_motorcycles_and_scooters), [electric bicycles](https://en.wikipedia.org/wiki/Electric_bicycle), [personal transporters](https://en.wikipedia.org/wiki/Personal_transporter) and advanced [electric wheelchairs](https://en.wikipedia.org/wiki/Electric_wheelchair). Also [radio-controlled models](https://en.wikipedia.org/wiki/Radio-controlled_model), [model aircraft](https://en.wikipedia.org/wiki/Model_aircraft), [aircraft](https://en.wikipedia.org/wiki/Aircraft), and the [Mars *Curiosity* rover](https://en.wikipedia.org/wiki/Curiosity_(rover)).

More niche uses include backup power in telecommunications applications. Lithium-ion batteries are also frequently discussed as a potential option for [grid energy storage](https://en.wikipedia.org/wiki/Grid_energy_storage), although they are not yet cost-competitive at scale.

### Voltage limits:

Lithium-ion cells are susceptible to stress by voltage ranges outside of safe ones between 2.5 and 3.65/4.1/4.2 or 4.35V (depending on the components of the cell). Exceeding this voltage range results in premature aging and in safety risks due to the reactive components in the cells. When stored for long periods the small current draw of the protection circuitry may drain the battery below its shutoff voltage; normal chargers may then be useless since the [battery management system](https://en.wikipedia.org/wiki/Battery_management_system) (BMS) may retain a record of this battery (or charger) 'failure'. Many types of lithium-ion cells cannot be charged safely below 0 °C, as this can result in plating of lithium on the anode of the cell, which may cause complications such as internal short-circuit paths.

Other safety features are required in each cell:

* Shut-down separator (for overheating)
* Tear-away tab (for internal pressure relief)
* Vent (pressure relief in case of severe outgassing)
* Thermal interrupt (overcurrent/overcharging/environmental exposure)



### Charging and discharging[

During discharge, lithium ions (Li+) carry the [current](https://en.wikipedia.org/wiki/Electrical_current) within the battery cell from the negative to the positive electrode, through the non-[aqueous](https://en.wikipedia.org/wiki/Aqueous_solution) [electrolyte](https://en.wikipedia.org/wiki/Electrolyte) and separator diaphragm.

During charging, an external electrical power source (the charging circuit) applies an over-voltage (a higher voltage than the battery produces, of the same polarity), forcing a charging current to flow **within each cell** from the positive to the negative electrode, i.e., in the reverse direction of a discharge current under normal conditions. The lithium ions then migrate from the positive to the negative electrode, where they become embedded in the porous electrode material in a process known as [intercalation](https://en.wikipedia.org/wiki/Intercalation_(chemistry)).

Energy losses arising from electrical [contact resistance](https://en.wikipedia.org/wiki/Contact_resistance) at interfaces between [electrode](https://en.wikipedia.org/wiki/Electrode) layers and at contacts with current collectors can be as high as 20% of the entire energy flow of batteries under typical operating conditions.

The charging procedures for single Li-ion cells, and complete Li-ion batteries, are slightly different:

* A single Li-ion cell is charged in two stages:

1. [Constant current](https://en.wikipedia.org/wiki/Constant_current) (CC).
2. [Constant voltage](https://en.wikipedia.org/wiki/Voltage_source) (CV).

* A Li-ion battery (a set of Li-ion cells in series) is charged in three stages:

1. [Constant current](https://en.wikipedia.org/wiki/Constant_current).
2. Balance (not required once a battery is balanced).
3. [Constant voltage](https://en.wikipedia.org/wiki/Voltage_source).

**CHAPTER-5**

**5.1 Advantages:**

**The Advantages of Robots**

* They Increase Production. ...
* They're More Accurate than They Make Fewer Mistakes. ...
* Humans. ...
* They Reduce Wastage. ...
* They're More Reliable than Humans. ...
* They Can Work 24/7. ...
* They Save Time. ..

**5.2 APPLICATIONS:**

1. Assembling heavy parts.

2. Avoiding accidents.

3. Reducing human errors.

4. Increasing the speed of production.

**5.3 FUTURE EXPANSION:**

1. ELECTRONIC VEHICLES

2. AUTO PILOT

**CHAPTER-6**

**6.1 CONCLUSION:**

The hardware components are successfully assembled and interfacing the microcontroller with robot is achieved. Controlling the motion of robot via webpage as well as from android applet is successfully obtained. Hence the two modules of controlling the robot is successfully tested and demonstrated. Though controlling using Bluetooth limits the range of distance for communication, a smart and easy means to guide a robot is achieved. Controlling the motion of robot via internet is one of the easiest means as it requires the user to access the designated webpage to guide it. This system can be used in defiance applications for detecting landmines in war field and for bomb detections by mounting a metal detector sensor on it. Further, the size of device can be miniaturized based upon specific applications.

**REFERENCES:**

[1] ‘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

[2] [www.wifibot.com](http://www.wifibot.com)

[3] ‘Networking and Internet Applications’ by Sangay Yeshi

[4] <https://www.rakeshmondal.info/L293D-Motor-Driver>

[5] <https://youtu.be/hAOEkjjfaj8> (Behind the clutter – youtube channel)