**ARDUINO BASED WIFI CONTROLLED INTELLIGENT ROBOT**

##### A PROJECT REPORT

##### *Submitted by*

##### B.SANDHYA REDDY (18113127)

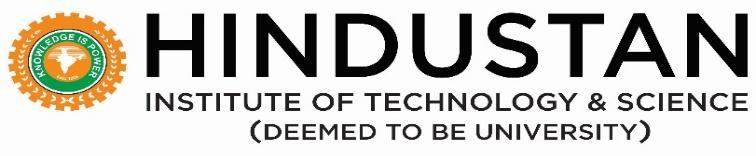
ASHWIN KUMAR (181130067)

***In partial fulfillment for the award of the course***

**IOT BASIC PROGRAMMING- ARDUINO, RASPBERRYAND EMBEDDED SYSTEM**

***in***

# DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING (NON DEPARTMENT ELECTIVE)

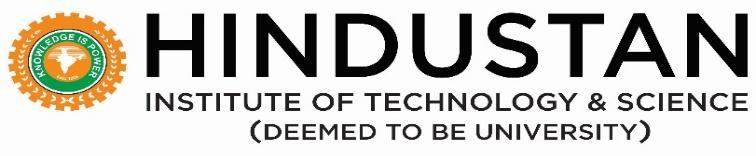


**DEPARTMENT OF COMPUTING SCIENCES**

**HINDUSTAN INSTITUTE OF TECHNOLOGY AND SCIENCE**

**PADUR, CHENNAI - 603 103**

**MAY 2020**

****

**BONAFIDE CERTIFICATE**

This is to certify that this project report **“ARDUINO BASED WIFI CONTROLLED INTELLIGENT ROBOT”** is the bonafide work of **“B.SANDHYA REDDY (18113127), ASHWIN KUMAR (18113067)”** who carried out the project work as a part of the subject **“EED 4298 452 – IoT Basic programming – Arduino, Raspberry Pi and Embedded systems”** under my supervision during the academic year 2019-2020.

**SIGNATURE**

**Dr. G. Muthukumaran**

SUPERVISOR

Dept. of EEE,

Hindustan Institute of Technology & Science,

No. 1, Rajiv Gandhi Salai (OMR),

Padur,

Chennai - 603103

**INTERNAL EXAMINER EXTERNAL EXAMINER**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGMENT**

We are pleased to acknowledge our sincere thanks to Board of Management of Hindustan Institute of Science and Technology for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We would like to express our sincere and deep sense of gratitude to our Project Guide Dr. G. Muthukumaran for his valuable guidance, suggestions and constant encouragement which paved the way for the successful completion of our project work.

We wish to express our thanks to all Teaching and Non-teaching staff members of the Department of Electrical and Electronics Engineering who were helpful in many ways for the completion of the project.

**B.SANDHYA (18113127)**

**ASHWIN KUMAR (18113067)**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **LIST OF TABLES** | **04** |
|  | **LIST OF FIGURES** | **05** |
|  | **ABSTRACT** | **05** |
| **1** | **INTRODUCTION** | **06** |
| **2** | **LITERATURE SURVEY** | **07** |
| **3** | **PROBLEM STATEMENT** | **08** |
| **4** | **COMPONENTS BEING USED** | **09-13** |
| 4.1 | NODE MCU |  |
| 4.2 | ULTRASONIC SENSOR |  |
| 4.3 | MOTOR DRIVER |  |
| 4.4 | WIFI MODULE |  |
| 4.5 | ISD1820 VOICE RECORDING |  |
| **5** | **METHODOLOGY AND IMPLEMENTATION** | **14-18** |
| 5.1 | CIRCUIT ARRANGEMENT |  |
| 5.2 | ANDROID APPLICATION |  |
| **6** | **RESULTS** | **19-20** |
| **7** | **CONCLUSION** | **21** |
| 7.1 | ADVANTAGES |  |
| 7.2 | LIMITATIONS |  |
| **8** | **ACKNOWLEDGEMENT** | **22** |
| **9** | **REFERENCES** | **23-24** |
| **10** | **CODE** | **25-32** |
| 10.1 | CODE FOR CHECKING IP |  |
| 10.2 | 9.2 PROJECT SOURCE CODE |  |

**LIST OF TABLES**

Table 1: Components with specifications

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **NUMBER** | **NAME** | **PAGE NUMBER** |
| FIG 1 | NODE MCU | 6 |
| FIG 2 | L9110S DC MOTOR DRIVER | 7 |
| FIG 3 | ULTRA SONIC SENSOR | 8 |
| FIG 4 | JUMPER WIRES | 9 |
| FIG 5 | ISD1820 VOICE RECORDING | 10 |
| FIG 6 | FLOW CHART OF A WI-FI CONTROLLED ROBOT | 11 |
| FIG 7 | IMPLEMENTATION | 13 |
| FIG 8 | IMPLEMENTATION (CHASSIS) | 15 |
| FIG 9 | CIRCUIT DIAGRAM | 16 |
| FIG 10 | FLOW CHART OF A WI-FI CONTROLLED ROBOT | 17 |
| FIG 11 | APP SCREEN SHOT | 19 |
| FIG 12 | SCREEN SHOT OF SOURCE CODE | 20 |
| FIG 13 | SCREEN SHOT OF IP CHECKING CODE | 21 |

**ABSTRACT**

It is 2020; the world has technically grown in a very drastic way in the past few decades. Many countries have implemented robots in various fields robots have replaced humans in performing repetitive and dangerous tasks .Object transportation robot has become a very important part of modern science and exploration. This types of robots are used in various sectors including astronomy, mining and industries. In this report the design and construction of an object transportation robot have been described elaborately. This particular robot can be controlled from a remote distance using radio frequency. There are basically three major parts of the robot: receiver- transmitter couple, chassis mounted on the wheels and the robotic arm. Two ATmega32 microcontrollers have been used for the purpose, one in the transmitter and the other in the receiver. In the receiver this microcontroller receives the RF signal and controls the robot according to the instruction. The movement of robot is controlled using 2 DC motors coupled with each other wheel. So, we as students and the future engineers, our team came to an idea of creating a robot, which can be controlled through phone or any device .in addition, this paper gives a broad idea about designing a remotely controlled two-wheeled intelligent robotic rover over a Wi-Fi network by using an Arduino-Uno connected to an ESP8266-wifi module.

**CHAPTER-1**

**INTRODUCTION**

One of the major uses of robot is their flexibility and ease with which they can be used in all places. The project particularly aims for those places where human involvement is difficult and dangerous such as places where a building is collapsed or fire accidents etc. For controlling mobile robots, low-cost Bluetooth communication was applied in most cases [2], but the bandwidth of Bluetooth wireless transfer is not good enough and its range is only some meters either. There have been such ICs capable of Wi-Fi communication available for some years, which support Wi-Fi 802.11n transfer standard with appropriate encryption contrary to their relatively low prices, and their ranges are higher as well.

Our motive is to implement and design an intelligentArduino board based robot that can be driven remotely using ESP8266 WIFI module over a Wi-Fi network. The design of the robot is in such a way that it can be controlled remotely by any android device through an app, which can control the speed of the moving rover and the direction in which it is being headed to. This paper is a proposal to the idea of installing the audio receiver, which records the sounds in the surroundings of the rover and also has the capability to transmit the message that the controller wants to send. It is basically the implementation of two – way radio transceiver in the intelligent rover. In section 1, there is a detailed description of the hardware required for the project and their working. The basic hardware required are Arduino UNO board, ESP8266 Wi-Fi module, L298 motor driver module and male-female wires. This section consists 8 different modules in the code. In section 2, there is a detailed explanation of the Arduino code required to run the robot. In section 3, complete details of the android app to control the Wi-Fi controlled robot is given. In this project, we create an android app through MIT app creator, which can be integrated to the Arduino set up and run. Section 4 consists of the future implementations in the project and better ways to improve the quality and efficiency of the project.

**CHAPTER-2**

**LITERATURE SURVEY**

**2.1 GENERAL REVIEW**

The proposed system is implemented using ultra-sonic sensor, which detects the obstacles and changes its direction without the help of user (controller). The ultrasonic sensor can detect the obstacle at a range set by the controller. An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity. The robot car also contains a sound recorder sensor to record the voices or sounds in the surroundings and the information is stored in a remote cloud. This voice Record Module is based on ISD1820, capable of multiple‐message record/playback. It can offers true single‐chip voice recording, no‐volatile storage, and playback capability of about 10 seconds.

**CHAPTER-3**

**PROBLEM STATEMENT**

Many countries have implemented robots in various fields robots have replaced humans in performing repetitive and dangerous tasks .Object transportation robot has become a very important part of modern science and exploration. This types of robots are used in various sectors including astronomy, mining and industries. In this report the design and construction of an object transportation robot have been described elaborately.

Robots produce more accurate and high quality work. Robot**s** rarely make mistakes and are more precise than human workers. They can produce a greater quantity in a short amount of time. They can work at a constant speed with no breaks, days off, or holiday time.

Object transportation robots have become a very important part of modern science and exploration. These types of robots are used in various sectors including astronomy, mining

and industries. In this report the design and construction of an object transportation robot have been described elaborately. This particular robot can be controlled from a remote distance using radio frequency. There are basically three major parts of this robot, receiver-transmitter couple, chassis mounted on the wheels and the robotic arm. Two ATmega32 microcontrollers have been used for this purpose, one in the transmitter and the other in the receiver. In the receiver, this microcontroller receives the RF signal and controls the robot according to the instruction. The movement of this robot is controlled using four DC motors coupled with each wheel. The speeds of these DC motors determine whether the

robot will go forward, reverse or take a turn. And finally three servo motors have been used to grab any object precisely, located in the shoulder, arm and claw. The accurate operation of all these three parts ensures the operation of the complete RF controlled robotic system for object transportation.

**CHAPTER-4**

**COMPONENTS BEING USED**

The following components are being used for the implementation of this project:

|  |  |
| --- | --- |
| Robot car chassis |  |
| Wi-Fi-module | ESP8266 |
| Motor Driver | L298 DC Motor |
| Voice Recorder | ISD1820 |
| Node MCU |  |
| Jumper Wires |  |

Table 1: Components with specifications

**4.1 NodeMCU**

It is a low-cost open source IoT platform. It initially included firmware which runs on the [ESP8266](https://en.wikipedia.org/wiki/ESP8266) [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) from Express Systems, and hardware which was based on the ESP-12 module. Later, support for the [ESP32](https://en.wikipedia.org/wiki/ESP32) 32-bit MCU was added.

NodeMCU is an open-source IoT platform. It includes firmware that runs on the ESP8266 Wi-Fi SoC from Expressive Systems, and hardware which is based on the ESP-12 module. It is based on the eLua project and built on the Espressif Non-OS SDK for ESP8266.

  It is a development board which runs on the ESP8266 with the Espressif Non-OS SDK, and hardware based on the ESP-12 module. The device features 4MB of flash memory, 80MHz of system clock, around 50k of usable RAM and an on chip Wi-Fi Transceiver.



Figure 1: Node MCU

**4.2 L298 DRIVER**

The L298 Driver is a high voltage, high current dual ful bridge driver designed to accept standard TTL logic levels and drive inductive loads such relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals.

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. This depends on the voltage used at the motors VCC.

Features of motor driver are:

* High working voltage – can reach up to 46v.
* Large output current.
* 25W Rated Power.
* High-Voltage and Current full-bridge driver with 2 H-bridges used to drive inductive loads like DC and Stepper Motors.
* Controlled with standard logic level signals.



Figure2:L298 DC Motor

**4.3 JUMPER WIRES:**

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.



Figure 3:Jumper Wires

**4.4 ULTRASONIC SENSOR**

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves. An ultrasonic sensor uses a transducer to send and receive ultrasonic pulses that relay back information about an object's proximity. The robot car also contains a sound recorder sensor to record the voices or sounds in the surroundings and the information is stored in a remote cloud. This voice Record Module is based on ISD1820, capable of multiple‐message record/playback. It can offers true single‐chip voice recording, no‐volatile storage, and playback capability of about 10 seconds.

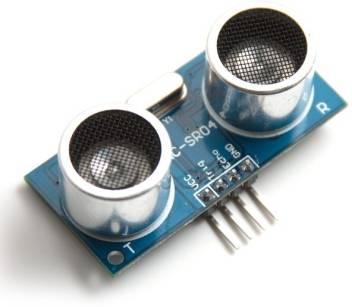


Figure 4: Ultra Sonic Sensor

**4.5 ISD1820 VOICE RECORDING**

Voice Record Module is based on ISD1820, which a multiple‐message record/playback device. It can offers true single‐chip voice recording, no‐volatile storage, and playback capability around 10 seconds. This module is easy to use which you could direct control by push button on board or by Microcontroller such as Arduino, STM32, Chip Kit etc. From these, you can easy control record, playback and repeat and so on.

**SPECIFICATION**

1. Push‐button interface, playback can be edge or level activated

2. Automatic power‐down mode

3. On‐chip 8Ω speaker driver

4. Signal 3.3V Power Supply

5. Can be controlled both manually and by MCU

6. Record up to around 10 seconds of audio

7. Dimensions: 37 x 54 mm

8. Push‐button interface, playback can be edge or level activated

9. Auto power‐down mode

10. Size: 38 x 42 mm



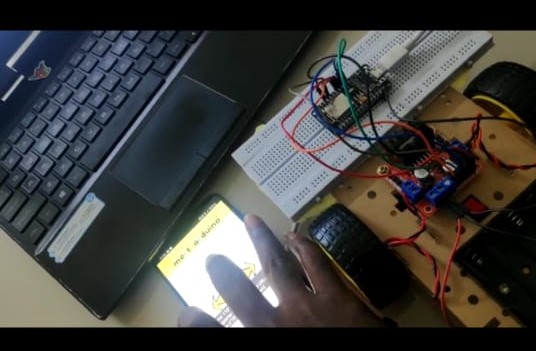
Figure 5:ISD1820 Voice Recording

**CHAPTER-5**

**METHODOLOGY AND IMPLEMENTATION**

The code for the Wi-Fi Controlled Robot project. One code is for the Node MCU ESP8266

* This Wi-Fi Controlled Robot is controlled with the help of an app
* There are many projects out there which have implemented a similar concept but using Apps like MIT app developer.
* The reason for which I have not gone with that App is that I felt that you really do not know what is going on actually as everything is done by the library files.
* So, if you want to know exactly what is going on with the project, then only you should try this. If not, you can simply implement the project using the Blynk App.
* If you go through both the codes carefully, you can understand how the communication happens between the Browser and the ESP Module.

Figure 6: Implementation

**5.1 CIRCUITARRANGEMENT**

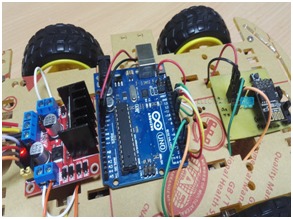


Figure 7: Implementation (Chasis)

Circuit diagrams are pictures with symbols that have differed from country to country and have changed over time, but are now to a large extent internationally standardized. Simple components often had symbols intended to represent some feature of the physical construction of the device. For example, the symbol for a resistor shown here dates back to the days when that component was made from a long piece of wire wrapped in such a manner as to not produce inductance, which would have made it a [coil](https://en.wikipedia.org/wiki/Inductor). These wire wound resistors are now used only in high-power applications, smaller resistors being cast from carbon composition (a mixture of [carbon](https://en.wikipedia.org/wiki/Carbon) and [filler](https://en.wikipedia.org/wiki/Filler_(materials))) or fabricated as an insulating tube or chip coated with a metal film. The internationally standardized symbol for a resistor is therefore now simplified to an oblong, sometimes with the value in [ohms](https://en.wikipedia.org/wiki/Ohm) written inside, instead of the zig-zag symbol. A less common symbol is simply a series of peaks on one side of the line representing the conductor.

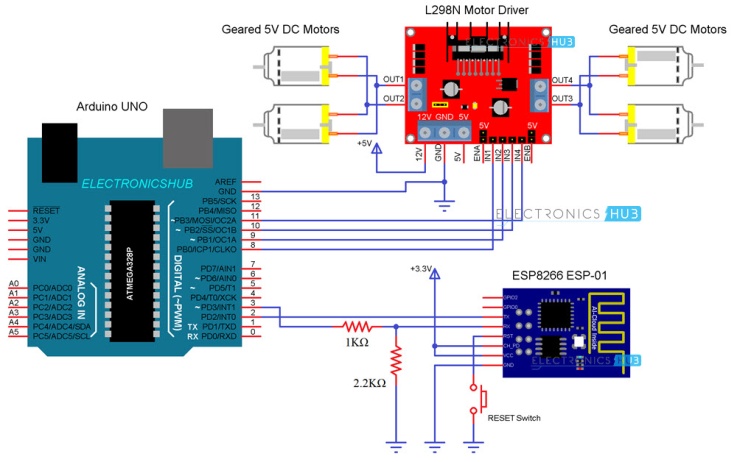


Figure 8: Circuit Diagram

Arduino is responsible for configuring the ESP8266 Module through Serial Communication and also controlling the L298n Motor Driver Module. The Digital Pins 2 and 3 of Arduino are configured as RX and TX using Software Serial function. These pins are connected to the TX and RX pins of the ESP8266 Module. Then, the Inputs of the L298n Motor Driver Module i.e. IN1, IN2, IN3 and IN4 are connected to Digital Pins 8, 9, 10 and 11 of Arduino UNO.As far as the robot chassis is concerned, it has 2 geared motors. So, the right motor is connected in parallel and connected to OUT1 and OUT2 terminals of the Motor Driver. Similarly, the left two motor is connected to OUT3 and OUT4.

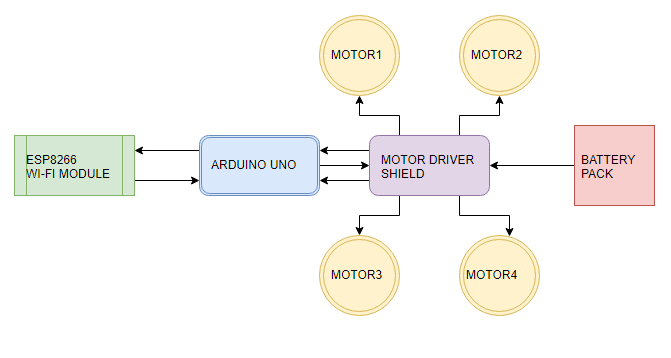
****

Figure 9: Flow chart of a Wi-Fi controlled robot

**5.2 ANDROID APPLICATION**

An android application is used as an interface between Node MCU ESP8266 module and the mobile. The app initially asks for the IP address. Once, the IP address is entered, it tracks the device with that IP address and acts like an interface to control the device. Here, the device is the robot. A screen with direction is displayed. The center button is the “STOP” button, which can be used to halt the robot. There are four arrows pointing towards four different directions, each arrow represents the movement of the robot in that direction.

To know the IP address of the Wi-Fi module, there is a code. When the code is compiled and executed an IP address is displayed. The same IP address must be entered in the arduino application. MIT app developer is used here to develop the application which ends up interfacing between them.

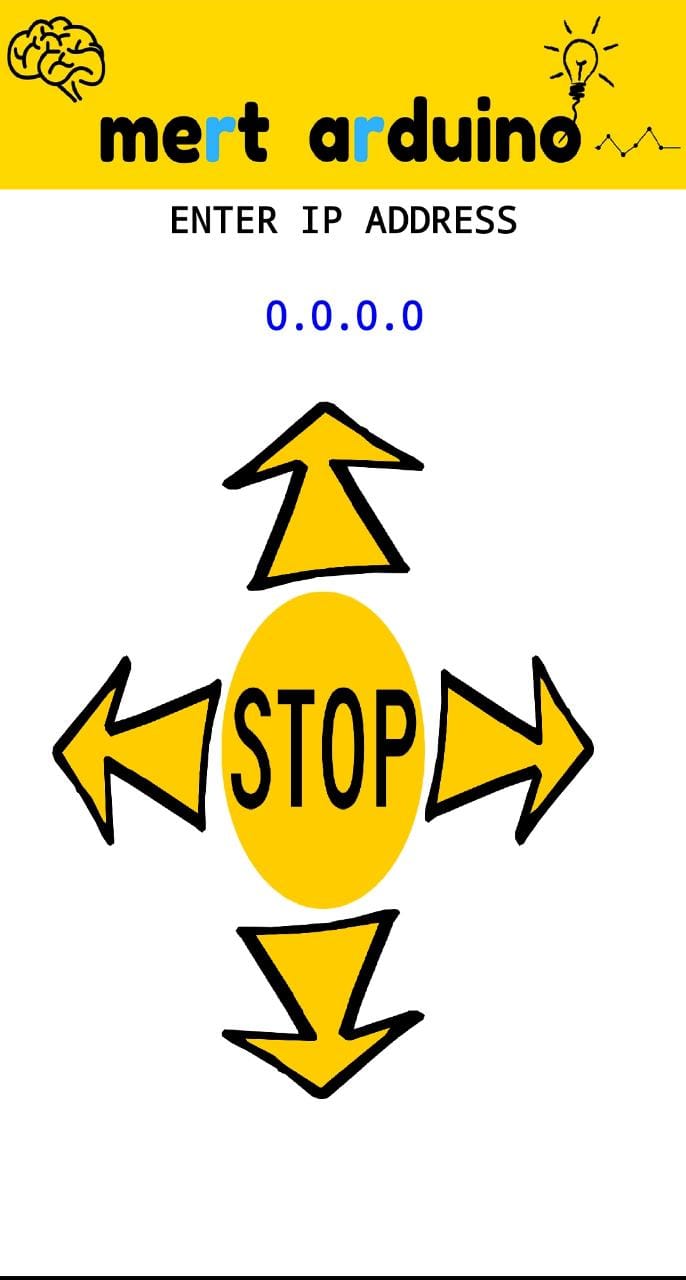


Figure 10: App Screen Shot

**CHAPTER-6**

**RESULTS**

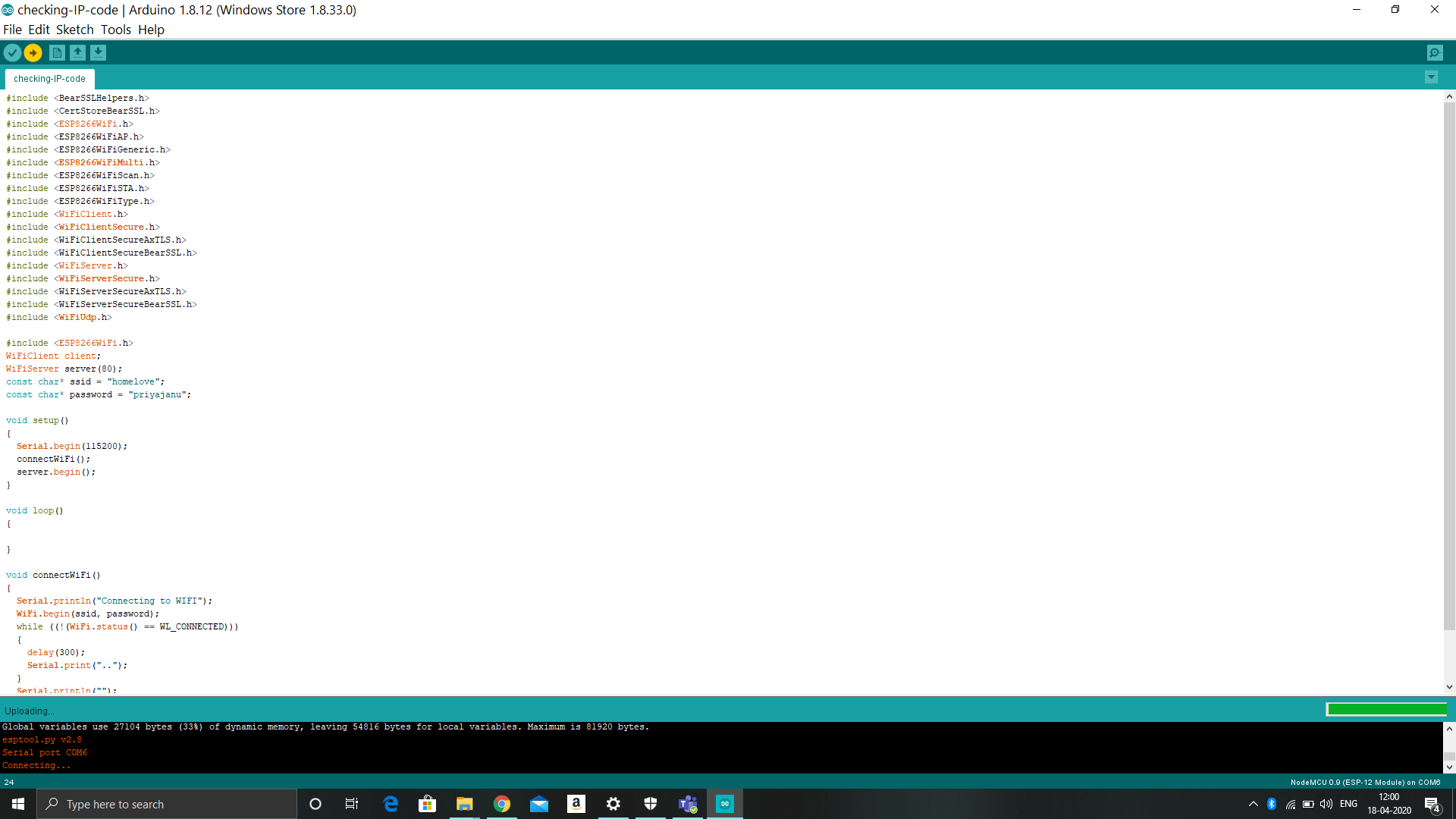


Figure 11.Screen shot of IP checking code

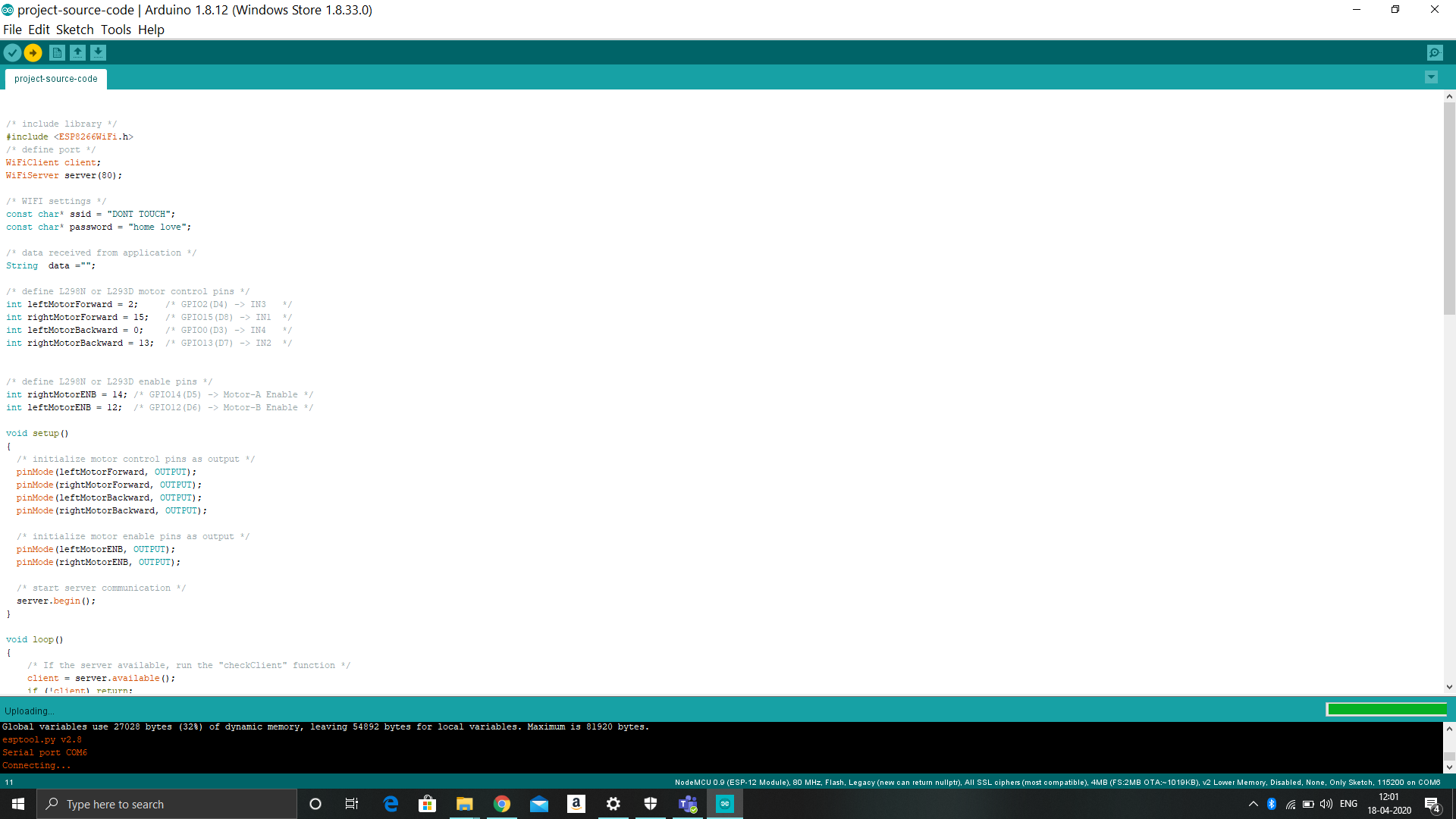


Figure 12: Screen shot of Source code

**CHAPTER-7**

**CONCLUSION**

* A simple Wi-Fi Controlled Robot is implemented in this project where a robotic car is controlled using a web page over Wi-Fi Network.
* You can make this project with advanced features like integrating a camera and accessing the feed lie on the browser.
* Industrial applications- Industrial remote controls, alarm systems and wireless transmission for various types of low-rate digital signals, industrial data acquisition systems.
* Monitoring applications- The applications include environmental monitoring, biomedical applications, habitat monitoring, battlefield management, wireless fire protection systems.
* Household applications: Alarm systems and wireless transmission for various types of low-rate digital signal, remote controls for various types of household appliances and electronics projects and several other applications related to RF wireless controlling.

**7.1 ADVANTAGES**

* Offers productivity, convenience and cost
* Advantage over traditional wired technology.
* No need for manual retrieval of data.
* Availability of real time data.

**7.2 LIMITATIONS**

The superior nature of this scheme depends on many environmental factors, such as operation scenarios, specific data types etc.

* More research work needs to be done in future to find the respective application scenarios for this scheme with all the related factors taken into consideration.
* This technique needs to be implemented in a wireless sensor network with mobile nodes, since mobility was not taken into account in this work

**CHAPTER-8**

**REFERENCES**

[1] Namita shinde, Shreya Srivastav, Vineet Sharma, Samrat Kumar

[2] Bharat Kumar, Rashmi Avinash, Sourav Sinha and K. Saravan kumar.

[3] K.M. Merlin Ruby, F. Anne Jenefer, D. Vidya

[4] Maheshbabu P, Chandra Shekar

[5] Chirag Jain, ReemaKadechkar, Sachin Chaturvedi, Susmitha Pradhan, Department of Electronics

[6]Muhammad Gulfam and Mirza Waleed Iftikhar Baig, “WG11 Android Based surveillance robot control system using socket programming with implementation” Internatonal journal of Multidisciplinary Science and Engineering, vol 6, no. 3, 2015.

[7]Karthikeyan K1 , Mutthu K2 , Nanthish N 3 , Rahesh R 4 , Vineeth V V5U.G. Student, Department of Electrical and Electronics Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India1,2,3,4 Assistant Professor, Department of Electrical and Electronics Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India 5Simulation of Robotic Car Controlled using Wi-Fi Module ESP8266

[8]Wi – Fi Control Robot Using Node MCU Raj Kumar Mistri Assistant Professor Dept. of ECE, RTC Institute of Technology, Ranchi

[9]Robot Controlled Car Using Wi-Fi Module S R Madkar (Assistant Professor), Vipul Mehta, Nitin Bhuwania, Maitri Parida Electronics Department, BVDU College of Engineering, Pune-Satara Road, Pune, Maharashtra, India

[10]Wireless Car Using WIFI – IoT - Bluetooth Dhruv Ketan Kotecha1 1Department of Electronics Engineering, K. J. Somaiya College of Engineering (Autonomous), (Affiliated to University of Mumbai), Mumbai, India

# [11] Wireless Gesture controlled Smart Robot using ESP8266

[Harshal Rade](https://www.semanticscholar.org/author/Harshal-Rade/1392150540), [Amol Bonde](https://www.semanticscholar.org/author/Amol-Bonde/1387021489), [Abhishek Deshpande](https://www.semanticscholar.org/author/Abhishek-Deshpande/49332064), [Sunny Pawar](https://www.semanticscholar.org/author/Sunny-Pawar/88971593), [M. T. Jagtap](https://www.semanticscholar.org/author/M.-T.-Jagtap/98810242)

# [12] Cost-effective WiFi controlled mobile robot J. Katona\* , A. Kovari\*\* \* University of Dunaujvaros/Institute of Computer Engineering, Dunaujvaros, Hungary \*\* University of Dunaujvaros/Institute of Engineering, Dunaujvaros, Hungary katonaj@uniduna.hu, [kovari@uniduna.hu](mailto:kovari@uniduna.hu)

[13] PrithvirajShetti et al, “On New Approach in Using

433MHz Radio Modules ”International Journal of

COMPUTER SCIENCE AND MOBILE COMPUTING,

VOL.3, PP.3405-345, APRIL-2014.

[14] Larry Mathies, Alonzo Kelly, “Obstacle Detection for

Unmanned Ground Vehicles: A Progress Report,PP.475-

[15] MichaelMcRoberts”BeginningArduino”,USA

ISBN-13 (electronic): 978-1-4302-3241-4, pp.81-96.

[16] M. Daily et al. Autonomous cross-country navigation

**CHAPTER-9**

**APPENDIX - CODE SNIPPETS**

There are two codes for the Wi-Fi Controlled Robot project. One code is for the Arduino UNO and the other is an HTML Code for creating a Web Page.

* This Wi-Fi Controlled Robot is controlled with the help of an HTML Web Page (which can be accessed using any web browser on a computer that is connected to the same WiFi Network as ESP8266).
* There are many projects out there which have implemented a similar concept but using Apps like MIT app developer.
* The reason for which I have not gone with that App is that I felt that you really do not know what is going on actually as everything is done by the library files.
* So, if you want to know exactly what is going on with the project, then only you should try this. If not, you can simply implement the project using the Blynk App.
* If you go through both the codes carefully, you can understand how the communication happens between the Browser and the ESP Module.

**9.1 CODE FOR CHECKING IP**

#include <BearSSLHelpers.h>

#include <CertStoreBearSSL.h>

#include <ESP8266WiFi.h>

#include <ESP8266WiFiAP.h>

#include <ESP8266WiFiGeneric.h>

#include <ESP8266WiFiMulti.h>

#include <ESP8266WiFiScan.h>

#include <ESP8266WiFiSTA.h>

#include <ESP8266WiFiType.h>

#include <WiFiClient.h>

#include <WiFiClientSecure.h>

#include <WiFiClientSecureAxTLS.h>

#include <WiFiClientSecureBearSSL.h>

#include <WiFiServer.h>

#include <WiFiServerSecure.h>

#include <WiFiServerSecureAxTLS.h>

#include <WiFiServerSecureBearSSL.h>

#include <WiFiUdp.h>

#include <ESP8266WiFi.h>

WiFiClient client;

WiFiServer server(80);

const char\* ssid = "Donedone";

const char\* password = "nooooooo4";

void setup()

{

Serial.begin(115200);

connectWiFi();

server.begin();

}

void loop()

{

}

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()));

}

**9.2 PROJECT SOURCE CODE**

/\* include library \*/

#include <ESP8266WiFi.h>

/\* define port \*/

WiFiClient client;

WiFiServer server(80);

/\* WIFI settings \*/

const char\* ssid = "Donedone";

const char\* password = "nooooooo";

/\* data received from application \*/

String data ="";

/\* define L298N or L293D motor control pins \*/

int leftMotorForward = 2; /\* GPIO2(D4) -> IN3 \*/

int rightMotorForward = 15; /\* GPIO15(D8) -> IN1 \*/

int leftMotorBackward = 0; /\* GPIO0(D3) -> IN4 \*/

int rightMotorBackward = 13; /\* GPIO13(D7) -> IN2 \*/

/\* define L298N or L293D enable pins \*/

int rightMotorENB = 14; /\* GPIO14(D5) -> Motor-A Enable \*/

int leftMotorENB = 12; /\* GPIO12(D6) -> Motor-B Enable \*/

void setup()

{

/\* initialize motor control pins as output \*/

pinMode(leftMotorForward, OUTPUT);

pinMode(rightMotorForward, OUTPUT);

pinMode(leftMotorBackward, OUTPUT);

pinMode(rightMotorBackward, OUTPUT);

/\* initialize motor enable pins as output \*/

pinMode(leftMotorENB, OUTPUT);

pinMode(rightMotorENB, OUTPUT);

/\* start server communication \*/

server.begin();

}

void loop()

{

/\* If the server available, run the "checkClient" function \*/

client = server.available();

if (!client) return;

data = checkClient ();

/\*\*\*\*\*\*\*\* Run function according to incoming data from application \*\*\*\*\*\*\*\*\*/

/\* If the incoming data is "forward", run the "MotorForward" function \*/

if (data == "forward") MotorForward();

/\* If the incoming data is "backward", run the "MotorBackward" function \*/

else if (data == "backward") MotorBackward();

/\* If the incoming data is "left", run the "TurnLeft" function \*/

else if (data == "left") TurnLeft();

/\* If the incoming data is "right", run the "TurnRight" function \*/

else if (data == "right") TurnRight();

/\* If the incoming data is "stop", run the "MotorStop" function \*/

else if (data == "stop") MotorStop();

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* FORWARD \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void MotorForward(void)

{

digitalWrite(leftMotorENB,HIGH);

digitalWrite(rightMotorENB,HIGH);

digitalWrite(leftMotorForward,HIGH);

digitalWrite(rightMotorForward,HIGH);

digitalWrite(leftMotorBackward,LOW);

digitalWrite(rightMotorBackward,LOW);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* BACKWARD \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void MotorBackward(void)

{

digitalWrite(leftMotorENB,HIGH);

digitalWrite(rightMotorENB,HIGH);

digitalWrite(leftMotorBackward,HIGH);

digitalWrite(rightMotorBackward,HIGH);

digitalWrite(leftMotorForward,LOW);

digitalWrite(rightMotorForward,LOW);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TURN LEFT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void TurnLeft(void)

{

digitalWrite(leftMotorENB,HIGH);

digitalWrite(rightMotorENB,HIGH);

digitalWrite(leftMotorForward,LOW);

digitalWrite(rightMotorForward,HIGH);

digitalWrite(rightMotorBackward,LOW);

digitalWrite(leftMotorBackward,HIGH);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TURN RIGHT \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void TurnRight(void)

{

digitalWrite(leftMotorENB,HIGH);

digitalWrite(rightMotorENB,HIGH);

digitalWrite(leftMotorForward,HIGH);

digitalWrite(rightMotorForward,LOW);

digitalWrite(rightMotorBackward,HIGH);

digitalWrite(leftMotorBackward,LOW);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* STOP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void MotorStop(void)

{

digitalWrite(leftMotorENB,LOW);

digitalWrite(rightMotorENB,LOW);

digitalWrite(leftMotorForward,LOW);

digitalWrite(leftMotorBackward,LOW);

digitalWrite(rightMotorForward,LOW);

digitalWrite(rightMotorBackward,LOW);

}

/\*\*\*\*\*\*\*\*\*\*\*\* RECEIVE DATA FROM the APP \*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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;

}