Abstract

Remote Controlled Robot using Wi-Fi Interface is a wireless system, where robot is controlled by the control buttons in the web page. The robot also sends the data obtained from the sensors and audio-video device to the same web page. This system demonstrates the faster, convenient, efficient and cheaper way of controlling and monitoring the wireless robotic system.

The robotic system is using a smoke sensor to determine the amount of smoke detected by the robot. With the microcontroller connected to Wi-Fi, sensor data and audio video are automatically uploaded to the web site. Thus it helps in finding the real time situation of far distance with minimum effort.

List of Figures

Figure 1 Client Server Model 4
Figure 2 Arduino Uno 5
Figure 3 NodeMCU 5
Figure 4 Motor Driver and DC Motor 6
Figure 5 MQ-2 Smoke Sensor 7
Figure 6 IP Webcam Control Interface 8
Figure 7 Overall System Flowchart 10
Figure 8 Block Diagram 11
Figure 9 Circuit for Movement Control 14
Figure 10 Circuit for MQ-2 Sensor 15
Figure 11 Total Circuit 16
Figure 12 User Interface 17
Figure 13 Concentration vs Voltage graph of MQ-2 Sensor 18
Figure 14 Camera Output 18
Figure 15 Mechanical Setup 19
Figure 16 Final Design 20

Symbols and Abbreviations

WLAN: Wireless Local Area Network

MCU: Microcontroller Unit

HTML: Hypertext Markup Language

CSS: Cascaded Style Sheets

JS: Java Script

I/O: Input/ Output

SoC: System On a Chip

SDK: Software Development Kit

GPIO: General Purpose Input Output

AV: Audio Video

FPV: First Person View

HTTP: Hypertext Transfer Protocol

URI: Uniform Resource Identifier

Table of Contents

Abstract i

List of Figures ii

Symbols and Abbreviations ii

Chapter 1 Introduction 1

1.1. Background 1

1.2. Objectives 1

1.3. Innovation and Benefits 2

Chapter 2 Literature Review and Existing Knowledge 3

2.1. Client/Server Architecture 3

2.2. Components Description 4

2.2.1. Arduino Uno 4

2.2.2. NodeMCU 5

2.2.3. L293D Motor Driver Module and DC motors 5

2.2.4. MQ-2 Smoke Sensor 6

2.2.5. Wireless AV Camera 7

2.3. IP Webcam 7

2.4. Web Page Development 8

2.5. Router in Networking 9

2.6. ESP-12 as a webserver 9

Chapter 3 Methodology 10

Chapter 4 System Description 11

4.1. Block Diagram 11

4.2. Circuit for the System 14

4.2.1. Circuit for Movement Control 14

4.2.2. Circuit for MQ-2 Sensor interface 15

4.2.3. Total Circuit 16

4.3. User Interface Development 16

\4.4. Result and Analysi 17

4.4.1. Smoke Sensor Output 17

4.4.2. Camera Output 18

4.4.3 Mechanical Setup 19
4.4.4. Final Design 20
Chapter 5 Problem Encountered 21
Chapter 6 Gantt Chart and Cost Estimation 22
6.1. Gantt Chart 22
6.2. Cost Estimation 23
Chapter 7 Conclusion 24
7.1 Future work (Developments) 24
Chapter 8 References 25
Appendix I: Code 26
Appendix II: MQ-2 Sensor datasheet 34

Chapter 1 Introduction

1.1. Background

Today Internet is a leading technology in data communication between different devices in entire world. Every electronic system today is desired to be controlled by internet. Hence, most of the technology mega companies around the world are investing huge amount of resources in Internet controlled things. The easier and most convenient technology used in connecting the internet today is Wi-Fi. Use of Wi-Fi allows us to send data wirelessly in a wide range of location ranging from a small room to hundreds of kilometers.

Robotic field is one of the most benefited field by the use of Internet of internet in today's world. Robotic system requires continuous controlling and real time data monitoring system. Controlling the robotic system and monitoring of its data becomes easy when done wirelessly. If wireless controlling and monitoring is done through the internet then it would meet the requirement of today's world of connecting everything on internet. Wireless technology will also make it easier to control and monitor the robotic system hundreds of kilometers far. And use of Wi-Fi as the wireless technology will make system configuration easier and convenient. Today radio frequency and infrared technology are used in controlling a robot and most of them involve manual operated remote. But the range of transmission in the mentioned case is very less and the use of manually operated remote control doesn't meet the requirement of connecting everything on internet of today's world.

However, the use of Wi-Fi technology will make the robotic system easier to interact on internet from a large distance in very easy and efficient manner.

1.2. Objectives

The objectives of this project are:

1. To design a robotic system controlled through web page control buttons.
2. To analyze and monitor the data from the robot on the web page.

1.3. Innovation and Benefits

a) Web Control System

This project is mainly focused in making a wireless robotic system where control and monitor of robotic system is done in a more efficient, faster, convenient and reliable method. This system requiring minimal human effort.

b) Wide range of communication

Due to the use of Wi-Fi module, the range of our system will vary from a small distance to 100m distance. Use of more powerful module can vary the range to 336 km. This would provide a large distance controlling and monitoring easier.

Chapter 2 Literature Review and Existing Knowledge

In the recent years internet communication and software interference to most of hardware is increasing rapidly to make the life of people (consumer) more convenient and comfortable. Wireless Communication like WLAN has become more popular in home networking. Different technologies has been made to make every day work of humans easier and comfortable

In this project the microcontroller used is Particle photon which has an inbuilt ESP-12 Wi-Fi chip which lets us connect to Internet through Wi-Fi. The NodeMCU can be connected to Internet through a website. The NodeMCU acts as a server in our system and it deals with three functions: Sending data from client to server, sending data from server to client and external device connected to the server.

These stated functions will be performed by controlling movement of our robotic system from web, displaying the sensor data output from our robotic system on web and displaying external camera output on output on our web page.

2.1. Client/Server Architecture

Client/server architecture is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client. This type of architecture has one or more client computers connected to a central server over a network or internet connection. This system shares computing resources.

Client/server architecture is also known as a networking computing model or client/server network because all the requests and services are delivered over a network. Client/server architecture is a producer/consumer computing architecture where the server acts as the producer and the client as a consumer. The server houses and provides high-end, computing-intensive services to the client on demand. These services can include application access, storage, file sharing, printer access and/or direct access to the server's raw computing power.

Client/server architecture works when the client computer sends a resource or process request to the server over the network connection, which is then processed and delivered to the client. A

server computer can manage several clients simultaneously, whereas one client can be connected to several servers at a time, each providing a different set of services. In its simplest form, the internet is also based on client/server architecture where web servers serve many simultaneous users with website data.

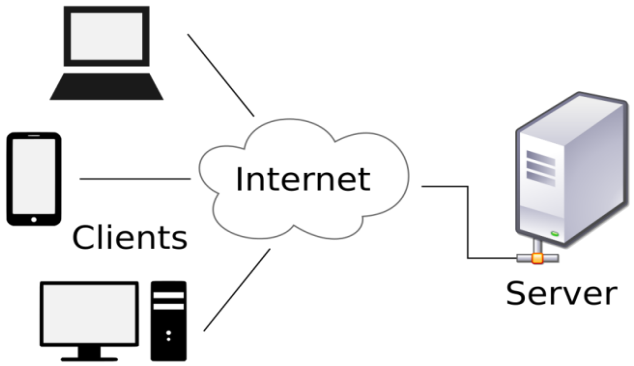


Figure 1 Client Server Model

2.2. Components Description

2.2.1. Arduino Uno

An Arduino Uno is a computer present in a single integrated circuit which is dedicated to perform one task and execute one specific application. It contains memory, programmable input/output peripherals as well a processor. Arduino Uno is mostly designed for embedded applications and is heavily used in automatically controlled electronic devices. They are far more economical to control electronic devices and processes as the size and cost involved is comparatively less than other methods. It operates at a low clock rate frequency, usually use four bit words and are designed for low power consumption. Its architecture varies greatly with respect to purpose from general to specific, and with respect to microprocessor, ROM, RAM or I/O functions. It has a dedicated input device and often has a display for output. It is usually embedded in other equipment and are used to control features or actions of the equipment. It is used in situations where limited computing functions are needed.

Figure 2 Arduino Uno

2.2.2. NodeMCU

NodeMCU is an open-source firmware and development kit that helps us to prototype or build IoT product. 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 firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It is a system that can be considered as the combination of ESP 8266 Wi-Fi module and Arduino. NodeMCU acts as a server while connecting to a network. Its data can be fetched and modified in the client system.

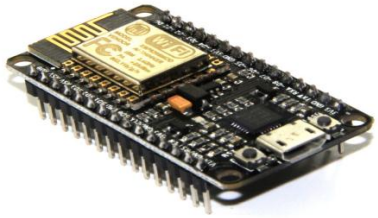


Figure 3 NodeMCU

2.2.3. L293D Motor Driver Module and DC motors

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 works on the concept of H-bridge. H-bridge is a circuit which allows the voltage to be flown in either direction. In a single L293D chip there are two h-Bridge circuit inside the IC

which can rotate two dc motor independently. Due its size it is very much used in robotic application for controlling DC motors.

The DC motor or direct current motor works on principal of motoring action which states “When a current carrying conductor is placed in a magnetic field, it experiences a torque and has a tendency to move. In other words, when a magnetic field and an electric field interact, a mechanical force is produced.”

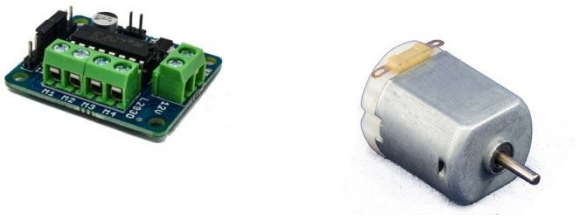


Figure 4 Motor Driver and DC Motor

2.2.4. MQ-2 Smoke Sensor

Smoke Sensor(MQ2) module is useful for smoke leakage detection (home and industry). It is suitable for detecting H2, LPG, CH4, CO, Alcohol, Smoke or Propane. Due to its high sensitivity and fast response time, measurement can be taken as soon as possible. The output voltage from the Gas sensor increases when the concentration of gas increases.

Item	Parameter	Min	Typical	Max	Unit
VCC	Working Voltage	4.9	5	5.1	V
PH	Heating consumption	0.5	-	800	mW
RL	Load resistance		adjustable		
RH	Heater resistance	-	33	-	Ω
Rs	Sensing Resistance	3	-	30	kΩ

Figure 5 MQ-2 Smoke Sensor

2.2.5. Wireless AV Camera

It is a small device having good performance with high-quality picture and sound transmitting and receiving device. It supports minimum of 100m transmission distance without block and can be used on TV, monitor, LCD, etc. including adaptive bracket and supports easy installation. The AV signal from the camera is sent to the TV tuner which is connected to the local system through the easy cap The Audio-Video streaming is done at the local and also at the remote system.

Similarly, android phones can also be used as an audio video camera. It can be done with the help of an application called IP Webcam. IP webcam application allows us to send the captured audio and video to the web which can be retrieved and streamed in our site.

2.3. IP Webcam

IP Webcam is an app that allows us to convert our Android device into an internet camera that we can see on any platform using internet browser. IP Webcam allows us to record videos in Webm, MOV and MPEG4 . The audio transmission, on the other hand, can be as a Wav, Opus or

AC file. It is a tool that allows us to convert our Android device in to a video surveillance tool. Whenever the android phone is connected to the Wi-Fi network, the router assigns a static IP address to the phone. Through this static IP the application streams the recorded live audio video in the network. This audio and video can further be retrieved in our website by using javascript commands.

Figure 6 IP Webcam Control Interface

Similarly, if we use a FPV camera instead of our Android device then we use desktop application called TimCam. TimCam has the similar features as IP webcam. It takes input via a TV tuner card, which is assigned a static IP by a Wi-Fi router and the AV streaming is made to that IP address. In comparison with IP webcam it requires more resources that make the system expensive.

2.4. Web Page Development

Web development is the coding or programming that enables website functionality, per the owner's requirements. It mainly deals with the non-design aspect of building websites, which includes coding and writing markup. Web development ranges from creating plain text pages to complex web-based applications, social network applications and electronic business applications. The web development hierarchy is as follows:

- Client-side coding
- Server-side coding
- Database technology

Web page development is done for the front end development of any system. It involves HTML for making layout. CSS for providing graphics to make page look attractive and javascript for making page responsive in order to perform specific task. We can also use the platforms of javascript like JQuery in order to make web page do whatever we want it to do like making specific part of the page refresh automatically after a certain interval of time.

2.5. Router in Networking

The Internet is formed by networks throughout the world interconnecting and passing on data to each other. Routers make the Internet work by forwarding data using a unified addressing system. They can send information to anywhere in the world as long as that location has an IP address. Routers rely on the Internet Protocol to provide a common addressing system. The computer sending information over the Internet has to package that data into a packet. The data packet contains a header which includes the IP address of the destination computer. Routers read this address and then forward the packet in the direction of its destination.

2.6. ESP-12 as a webserver

Esp-12 NodeMCU can be used as a webserver. It works as a server for the front end of the client side of network. To make ESP-12 a web server first a webserver library <ESP8266webvserver.h> should be included. Then instance called ESP8266WiFiMulticlass should be added. Then a function prototypes for HTTP handlers are made and serial communication inorder to send computer the message are started. Then we add a Wi-Fi network to which our ESP-12 will be connected and wait for it to connect that is scan and connect to the strongest of all. Then it tells the computer that it is connected to the network, on receiving the message computer send IP address of the ESP8266 to the computer.

Then for backend handle by the server it calls a inbuilt function called handleroot to handle clientURI. Then, it finally starts the server and listens for the HTTP requests. On receiving HTTP request, it processes it and send back the request to the client.

Chapter 3 Methodology

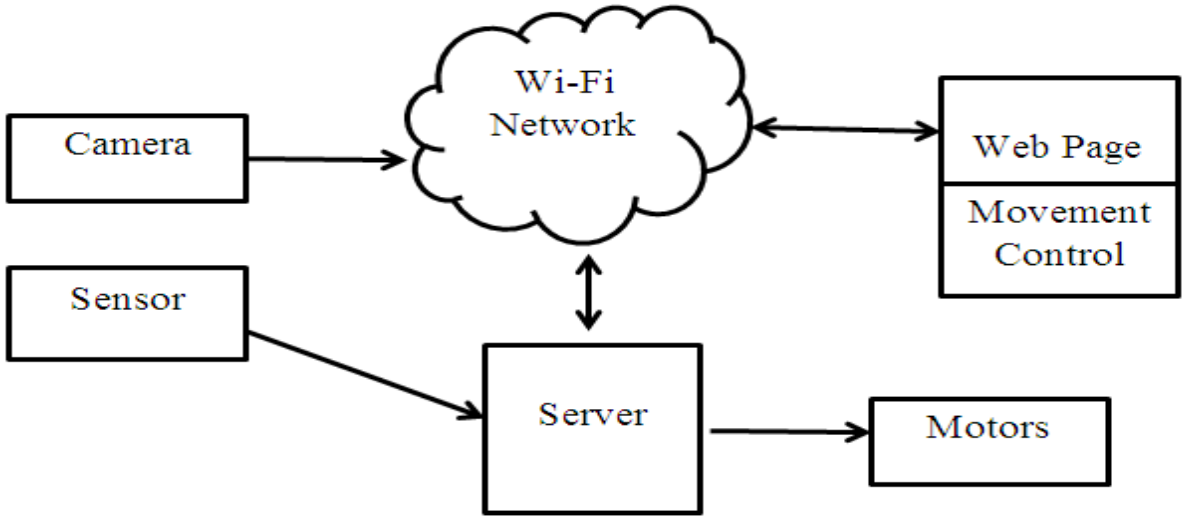


Figure 7 Overall System Overview

There are basically three major task of our system.

1. The first one is to **control the server side through the client side** in our Wi-Fi network. It is done by controlling the movement of robotic system. Certain code was given for the specified direction pressed by the client and the back end server side is coded in such a way to respond according to the instructions from the client.
2. The second task is to **display the changes made in the server side to our client side**. It is done by continuously updating the data from the smoke sensor (which is connected to our server) to the client side.
3. The third task is to **display the data of external device connected to our network on our client side**. It is done with the help of displaying audio video from the AV device. Here, AV device is not connected in our server network. Router assigns a IP address for our AV device which is manipulated in client side coding to make the audio video stream possible in our client side.
4. After individual tasks are performed then finally **the system is integrated** with all three features under a common system.

Chapter 4 System Description

The system design used a microcontroller with integrated Wi-Fi module. This microcontroller is integrated with movement control system and sensor system. These data are collected in the microcontroller and are frequently uploaded on the user interface where all the data are visible through internet anytime and anywhere connected within the nwtwork.

The following block diagram describes the working of the system.

4.1. Block Diagram

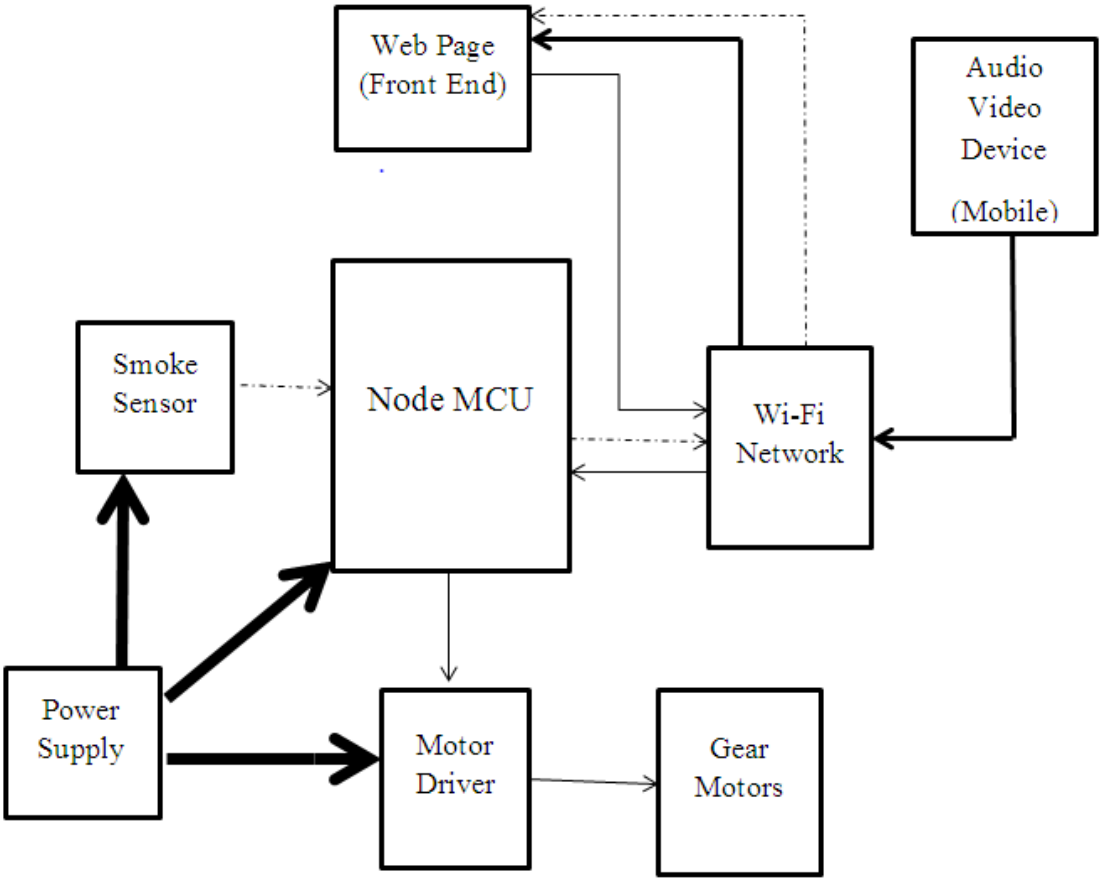


Figure 8 Block Diagram

Movement Control

The movement control section of our system deals with the movement of robotic system in the direction specified by the user. The front end design of the system in client side has five buttons for Front, Back, Left, Right and Back direction. Each button consist of a link, which when clicked will be opened in a new window made through frameset providing message fore, back, left, right and stop for the direction mentioned above respectively. A frameset is made in same window with very small height and width which shows the direction clicked.

The generated message will be transmitted to our server side via Wi-Fi module attached in NodeMCU. On detecting 'fore' message at front end it will produce logic HIGH, LOW, HIGH, LOW. Thus making both motors move in forward direction. Similarly on detecting 'back' message NodeMCU will produce LOW, HIGH, LOW, HIGH. Rotating both motors anti-clockwise. Similarly for left logic will be HIGH, LOW, LOW, LOW, thus rotating right motor in forward and stopping the left side motor. For moving robot in right opposite logic of left is applied and to stop the motor all logic are made LOW

I 1	I 2	Result
0	0	Stop
0	1	Reverse
1	0	Forward
1	1	Stop

Audio Video Stream

The audio video device, which is android smartphone for our case is placed in the robotic system. App called IP Webcam is installed in it and the device is connected to a Wi-Fi Network. The Wi-Fi network assigns a static IP address to the camera system. The audio and video captured by the camera can be viewed when visiting the IP assigned by the Network.

In that address other functionalities are also present which will control the various parameters of the captured audio and video. From that IP address we only need the Audio and Video part. So, in our frontend design, we used a textbox where the IP address of the camera will be placed. On submitting the IP address we made a javascript code which will only retrieve the audio and video part from complex site consisting of other different functionalities. Then the audio and video is streamed in our Wi-Fi module server.

Sensor Output

The third section of our system is the section for displaying the output of a smoke sensor. In the client side front end design a displaying window is created. It consists a word @@Smoke@@ which on backend processing will be replaced by the value we get from smoke sensor. The analog value we get from smoke sensor is uncalibrated value. We must calibrate it through the concentration voltage curve indicated in the datasheet. After calibration the actual value of the concentration of smoke will be detected in the client window. In order to make sensor take the data continuously the sensor output division will be made automatically refresh after certain interval of time.

4.2. Circuit for the System

4.2.1. Circuit for Movement Control

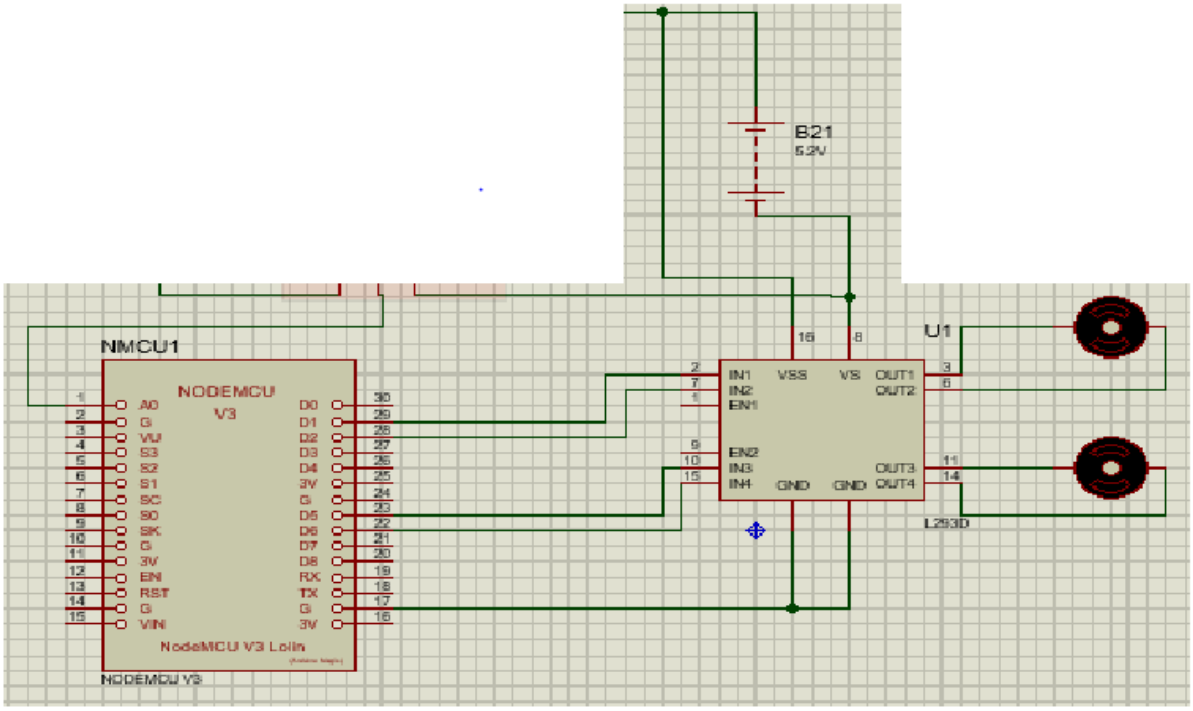


Figure 9 Circuit for Movement Control

Here, The NodeMCU is given a constant voltage of 5 volts via. a USB cable. For controlling the movement of our robotic system we need four logics, these logics are digital signals which are about 1.7 volts. A NodeMCU consists 16 GPIO pins which are used for taking digital inputs and outputs. Out of these 16 pins we take GPIO 13, GPIO 5,GPIO 4 and GPIO 14 for generating the four digital logics required for driving the DC motors. These generated digital logics are then given to L293D motor driver IC. L293D IC works for voltage range 5V-36V. This value can be adjusted according to speed requirements, 5V meaning the slowest and 36V meaning the highest. Then from the four outputs of motor driver module two motors of 300rpm are simultaneously connected. When the required logic is generated by pressing the control buttons from front end window, it is provided to motor driver module, which acts as an interface to the DC motors and control the movement.

4.2.2. Circuit for MQ-2 Sensor interface

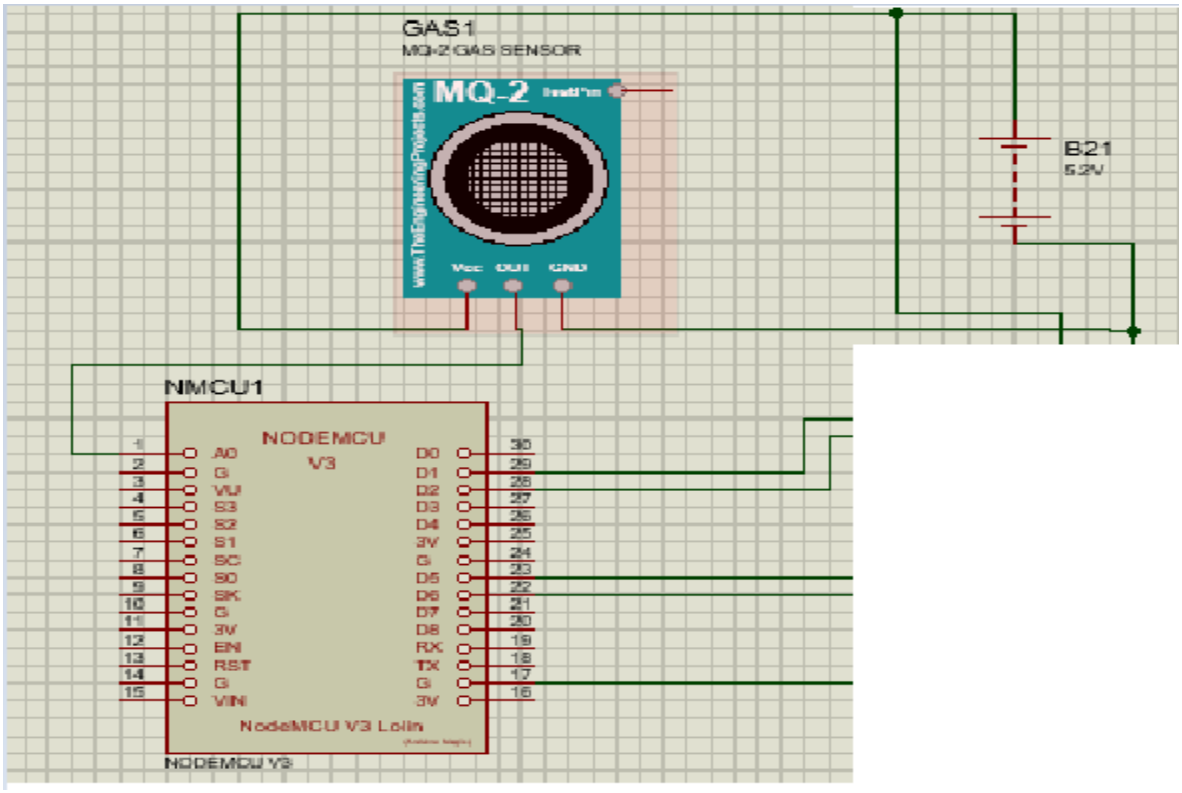


Figure 10 Circuit for MQ-2 Sensor

Here, we take the MQ-2 smoke sensor for detecting the smoke. The MQ-2 module consists of four pins VCC, GND, D0 and A0. VCC is the power supply pin which is given a constant 5 Volt. GND is the ground pin which is grounded, D0 is for taking the digital value and A0 is for taking analog value from the sensor. Here, we need to take the analog data and display in the client front end screen. The analog data from sensor is thus supplied to the NodeMCU analog pin. Then the required calibration is performed via programming and data is displayed in the web site.

4.2.3. Total Circuit

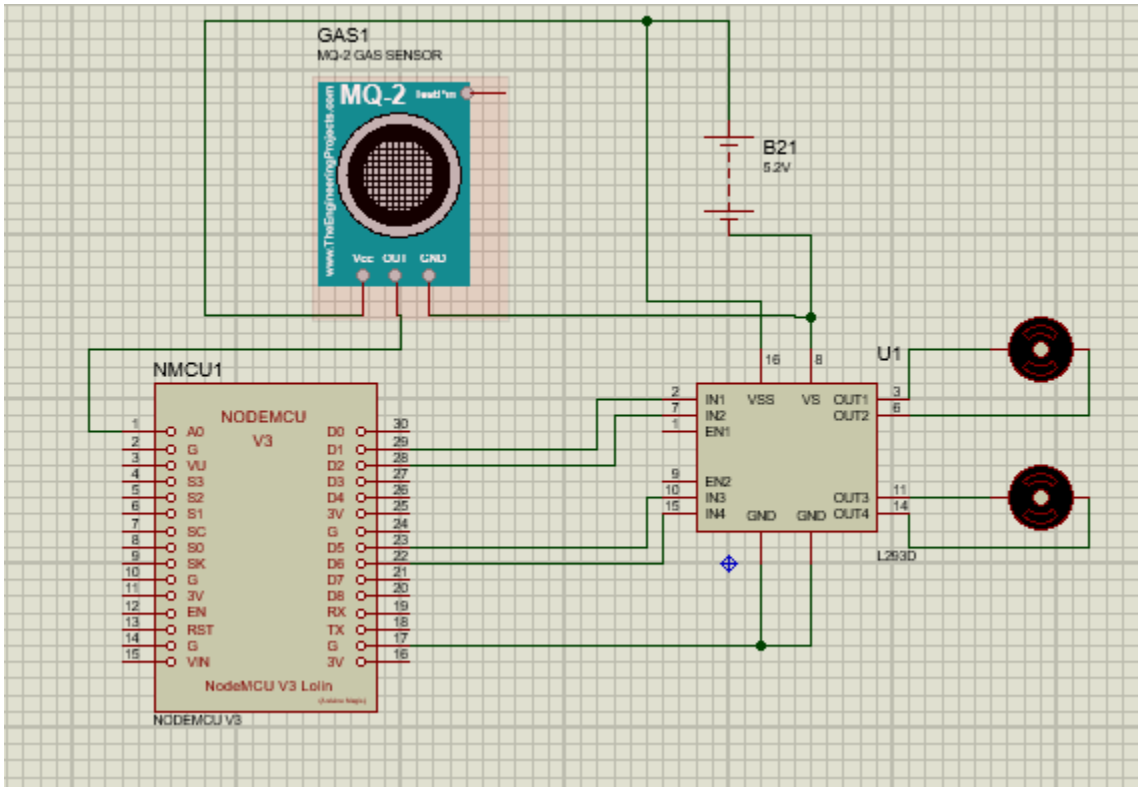


Figure 11 Total Circuit

4.3. User Interface Development

The front end in client side is developed for controlling the robot, displaying the sensor data and displaying the audio vidio. HTML, CSS and jvascript are involved in the user interface design. HTML is used for structuring the page, CSS for adding the attractive effects and javascript for making page responsive.

Here three divisions are made first of width 35% and height 470px, second division od width 65% and height 470px and third dividion of width 100% and height 150px. In the first division Control buttons are added for the movement control of our robotic system, it consist of 5 buttons for 5 state if the robot. Each button is a link which opens in a samll frameset of height 2px, created in the side of ‘Robot state’ in the figure below. The buttons are made attarctive by the use of CSS effects.

Similarly, the second division is created to display the audio and video, it consist of a text box where the IP address of AV device is added and on clicking the submit button javascript effect displays the audio and video by retriving it form its parent site. The height and width of audio and Video sections are mentioned by the use of CSS.

The third section is the smoke sensor data displaying division. In this division keyword '@@Smoke@@' is replaced by sensor concentration value depending upon the code in back end. This division is to be made refresh automatically after some interval in order take data continiously.This is done with the help of JQuery.

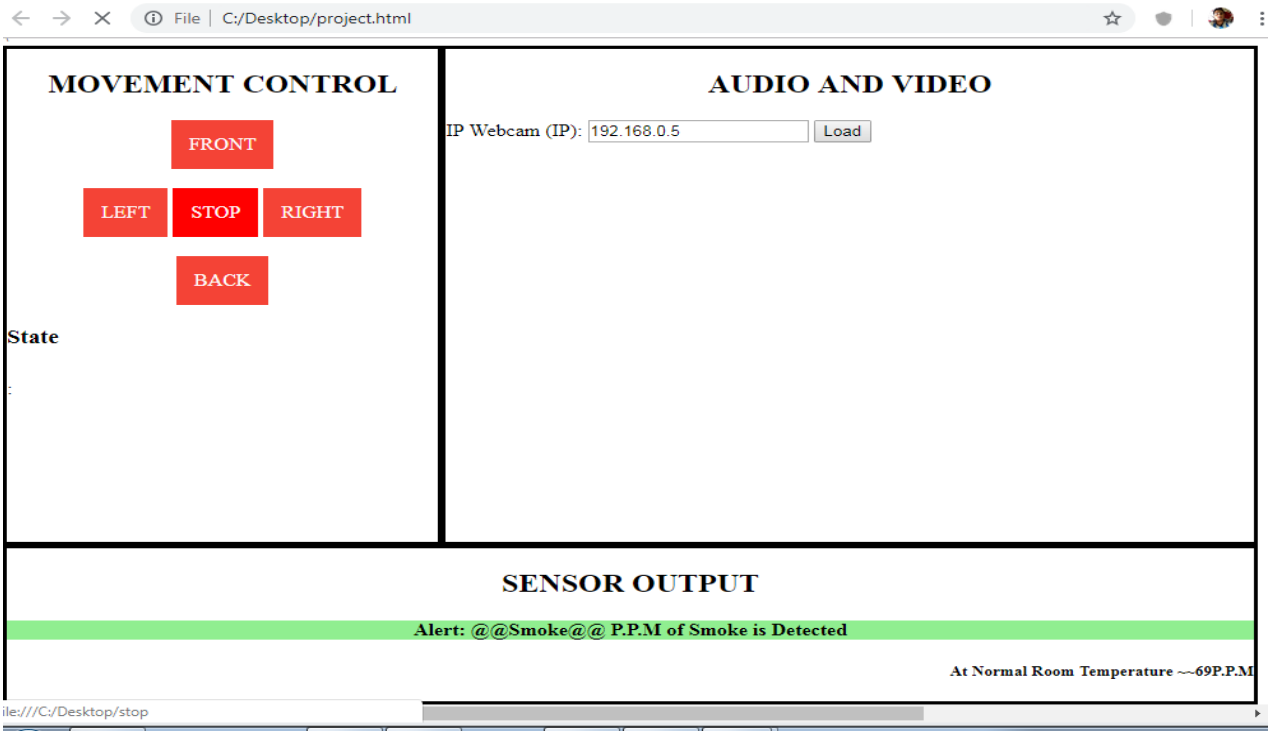


Figure 12 Control Interface

\4.4. Result and Analysis

4.4.1. Smoke Sensor Output

Following voltage measurements were done in lab with a help of multi meter.

The relationship between voltage and concentration is as follows

Concentration	Voltage
0	-2.4
500	-1.9
1000	-1.7
1500	-1.65
2000	-1.6
2500	-1.41
3000	-1.21
3500	-1.12
4000	-0.75
4500	-0.71
5000	-0.4
5500	-0.19
6000	0
6500	0.22
7000	2

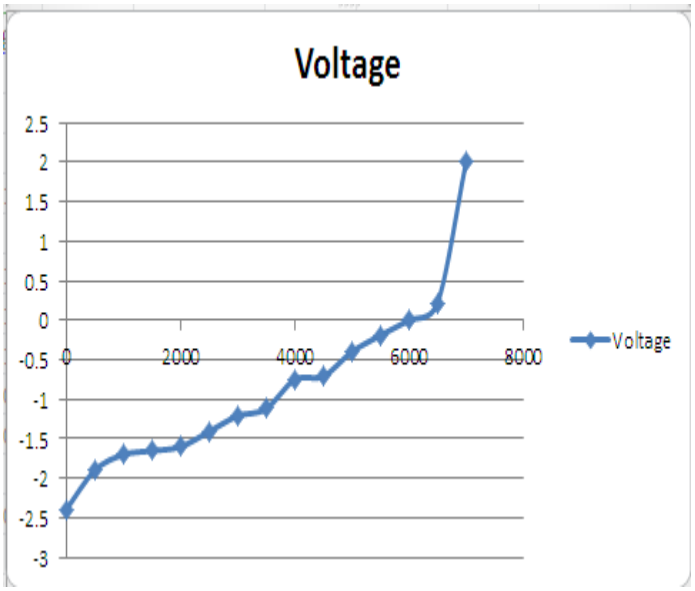


Figure 13 Concentration vs Voltage graph of MQ-2 Sensor

Hence the voltage of the sensor varies according to the concentration of gas.

4.4.2. Camera Output

The output obtained from the camera is shown in figures below.

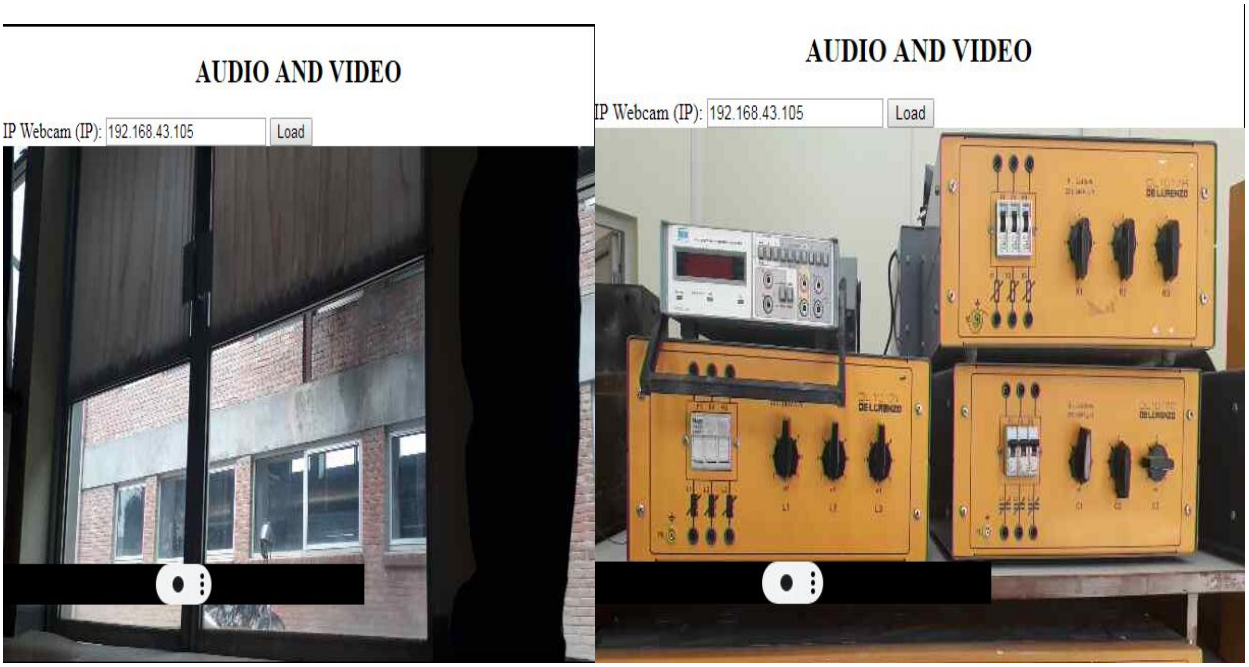


Figure 14 Camera Output

Hence, the output of the camera is taken and it is found that the delay or lag affects the video quality with higher resolution. The quality is improved in decreasing the resolution.

4.4.3 Mechanical Setup

Figure 15 Mechanical Setup

4.4.4. Final Design

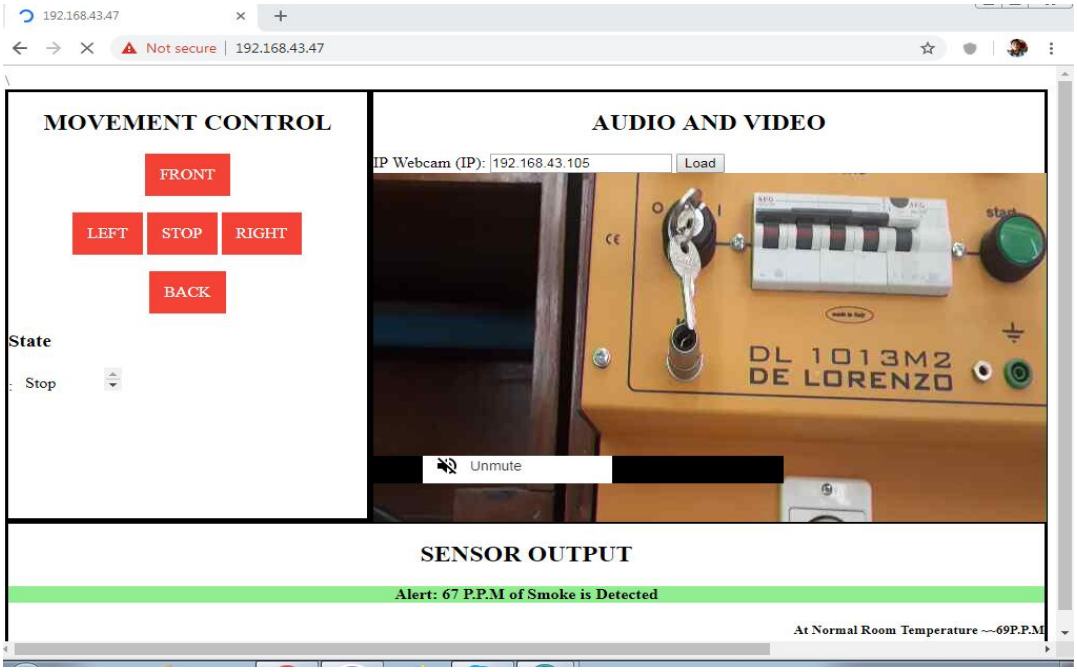


Figure 16 Final Design

Chapter 5 Problem Encountered

1. Difficulty of connecting ESP8266 and Arduino

   Problem

   In our proposed system we planned to use an ESP8266 Wi-Fi module and connect it with the arduino to take input and output. But while doing so we were unable to connect the Wi-Fi module with the arduino. The voltage and current requirement of ESP-8266 module is very specific, it required 3.3V and 500mA current but our arduino was able to supply 3.3V to it but not 500mA current. Use of external power source was also not enough to supply the required current.

   Solution

   As connecting Wi-Fi module with arduino was not possible, we replaced the ESP8266 Wi-Fi module with ESP-12 NodeMCU module. ESP-12 is a module which consist both Wi-Fi module and arduino chip in a single system. On doing so we didn't have to bother about the power requirement of the Wi-Fi module chip. The nodeMCU module consists all the voltage and current regulating circuit capable of providing enough voltage and current to the Chip.

2. Lagging of Video and Audio

   Problem

   The default resolution of our camera was 1920*1080, in this resolution we faced a lag of about five seconds in our audio and video.

   Solution

   Higher resolution results in higher frame rate so it requires faster transmission speed. The video streaming app IP webcam and TimCam both have feature of setting the different resolution. So to make transmission compatible with our system 960*720 or 960*540.

Chapter 6 Gantt Chart and Cost Estimation

6.1. Gantt Chart

WP Number and Title	Duration in Weeks										
	1	2	3	4	5	6	7	8	9	10	11
Literature Review											
NodeMCU and Sensor Integration											
Software integration and website											
Monitoring output											
Circuit and model integration											
Report writing											

6.2. Cost Estimation

Components	Quantity	Cost per unit(NRs)	Total Cost
NodeMCU	1	800	800
Gear Motors	2	150	300
Motor Driver Module	1	600	600
Structure	1	300	300
Casing (Steel)	1	200	200
Wires	1m	50	50
Others		500	500
Total			**2750**

Chapter 7 Conclusion

The design and product of Remote Controlled Robot using Wi-Fi interface is completed with all the proposed work. In this system, control buttons in web page control the movement of the robot. The usage sensors and audio video cameras help in monitoring the system from large distance. This helps in getting real time data for continuous monitoring of robot. More integration and upgrading can be done to make the product more efficient and error free.

7.1 Future work (Developments)

Following work can be done for the betterment this project.

1. For making this device commercial, the finishing of the case of the device must be made durable, reliable and easy-to-install.
2. To make the device more durable, a PCB of the circuit should be developed.
3. To make device accessible from anywhere, server must be connected to internet network.
4. More sensors and control parts can be added to make system automatic.

Chapter 8 References

1. V. Ramya, B. Palaniappan, and Subash Prasad, " Embedded Controller for Radar based Robotic Security Monitoring and Alerting System", International Journal of Computer Applications (IJCA), Volume 47-No. 23, June 2012.

2. V.Ramya, B.Palaniappan, K.Karthick and Subash Prasad "Embedded System for vehicle cabin toxic gas detection and alerting", Journal of Elsevier Procedia Engineering, 30(2012).

3. V. Ramya, B. Palaniappan , "Embedded Technology for Vehicle cabin safety Monitoring and Alerting System", International Journal of Computer Science Engineering and Applications, Volume 2-No.2, April 2012.

4. V. Ramya, B. Palaniappan , "Embedded system for Hazardous Gas detection and Alerting", International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.3, May 2012

Appendix I: Code

```
 #include <ESP8266WiFi.h>

#include <WiFiClient.h>


//ESP Web Server Library to host a web page

#include <ESP8266WebServer.h>


// HTML webpage contents in program memory

const char MAIN_page[] PROGMEM = R"=====(<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<style>

* {

  box-sizing: border-box;

}

.column1 {

  float: left;

  width:35%;

  padding: 0px;

  height: 470px;
```

```
}
.column2 {
  float: left;
  width:65%;
  padding:0px;
  height: 470px;
}
.column3 {
  float: left;
  width:100%;
  padding:0px;
  height: 150px;
}
.row:after {
  content: "";
  display: table;
  clear: both;
}
h4 {
  background-color: lightgreen;
}
a{
  background-color: #f44336;
  color: white;
  padding: 14px 14px;
```

```
  text-align: center;
  text-decoration: none;
  display: inline-block;
}
a:hover, a:active {
  background-color: red;
}
</style>
<script>
function readUrlAV (form) {
    TextVar = form.inputbox.value;
 VideoVar = "http://"+TextVar+":8080/video";
  AudioVar = "http://"+TextVar+":8080/audio.opus";
  document.getElementById("video").setAttribute('data', VideoVar);
  document.getElementById("audio").setAttribute('data', AudioVar);
}
</script>
</head>
<body>
<div class="row">
   <div class="column1" style="border:solid;">
  <h2 align="center">MOVEMENT CONTROL</h2>
   <div style="text-align:center;">
     <a href="fore" target="myIframe">FRONT</a><br> <br>
     <a href="left" target="myIframe">LEFT</a>
```

```
    <a href="stop" target="myIframe">STOP</a>
    <a href="right" target="myIframe">RIGHT </a><br> <br>
    <a href="reve" target="myIframe">BACK </a>
  </div>
<h3>State</h3>: <iframe name="myIframe" width="100" height="25" frameBorder="0"></iframe><br>
  </div>
  <div class="column2" style="border:solid;">
<h2 align="center"> AUDIO AND VIDEO</h2>
    <form name="myform" action="" method="GET">
      IP Webcam (IP):
      <input type="text" name="inputbox" value="192.168.0.5">
      <input type="button" name="button1" value="Load" onClick="readUrlAV(this.form)">
    </form>
    <object id="video" type="text/html" data="http://192.168.0.5:8080/video"
      style="position: absolute; width:100%; height:380px;">
    </object>
    <object id="audio" type="text/html" data="http://192.168.0.5:8080/audio.opus"
      style="position: absolute; width:390px; height:30px; top:425px;">
    </object>
   </div>
  </div>
  <div class="column3" style="border:solid;" >
 <h2 align="center">SENSOR OUTPUT</h2>
   <h4 align="center">Alert: @@Smoke@@ P.P.M of Smoke is Detected</h4>
   <h5 align="right">At Normal Room Temperature ~~69P.P.M</h5>
```

```
 </div>
</div>
</body>
</html>
)=====";
#define log1  13
#define log2  5
#define log3  4
#define log4  14
//SSID and Password of our WiFi router
const char* ssid = "Tenda_0CF358";
const char* password = "12345678";
ESP8266WebServer server(80); //Server on port 80
// This routine is executed when we open its IP in browser
void handleRoot() {
 Serial.println("Called root page");
 String s = MAIN_page; //Read HTML contents
 int smoke;
 smoke=analogRead(A0);
 s.replace("@@Smoke@@", String(smoke));
   Serial.println(smoke);
 server.send(200, "text/html", s); //Send web page
}
void handleForeward() {
 Serial.println("Foreward");
```

```
 digitalWrite(log1,HIGH);
 digitalWrite(log2,LOW);
digitalWrite(log3,HIGH);
digitalWrite(log4,LOW);
 server.send(200, "text/html", "Foreward"); //Send ADC value only to client ajax request
}
void handleReverse() {
 Serial.println("Reverse");
 digitalWrite(log1,LOW);
digitalWrite(log2,HIGH);
digitalWrite(log3,LOW);
digitalWrite(log4,HIGH);
 server.send(200, "text/html", "Reverse"); //Send ADC value only to client ajax request
}
void handleLeft() {
 Serial.println("Left");
 digitalWrite(log1,LOW);
digitalWrite(log2,LOW);
digitalWrite(log3,HIGH);
digitalWrite(log4,LOW);
 server.send(200, "text/html", "Left"); //Send ADC value only to client ajax request
}
void handleRight() {
 Serial.println("Right");
 digitalWrite(log1,HIGH);
```

```
digitalWrite(log2,LOW);
digitalWrite(log3,LOW);
digitalWrite(log4,LOW);
 server.send(200, "text/html", "Right"); //Send ADC value only to client ajax request
}
void handleStop() {
 Serial.println("Stop");
 digitalWrite(log1,LOW);
digitalWrite(log2,LOW);
digitalWrite(log3,LOW);
digitalWrite(log4,LOW);
 server.send(200, "text/html", "Stop"); //Send ADC value only to client ajax request
}
void setup(void){
  Serial.begin(115200);
  WiFi.begin(ssid, password);     //Connect to your WiFi router
  Serial.println("");

  pinMode(log1,OUTPUT);
pinMode(log2,OUTPUT);
pinMode(log3,OUTPUT);
pinMode(log4,OUTPUT);
  // Wait for connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
```

```
   Serial.print(".");
  }
  //If connection successful show IP address in serial monitor
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());  //IP address assigned to our ESP
  server.on("/", handleRoot);      //Which routine to handle at root location. This is display page
  server.on("/fore", handleForeward);
  server.on("/reve", handleReverse);
server.on("/left", handleLeft);
server.on("/right", handleRight);
server.on("/stop", handleStop);
  server.begin();                  //Start server
  Serial.println("HTTP server started");
}
void loop(void){
  server.handleClient();          //Handle client requests
}
```

Appendix II: MQ-2 Sensor datasheet

MQ-2 Semiconductor Sensor for Combustible Gas

Sensitive material of MQ-2 gas sensor is SnO_2, which with lower conductivity in clean air. When the target combustible gas exist, The sensor's conductivity is more higher along with the gas concentration rising. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration.

MQ-2 gas sensor has high sensitity to LPG, Propane and Hydrogen, also could be used to Methane and other combustible steam, it is with low cost and suitable for different application.

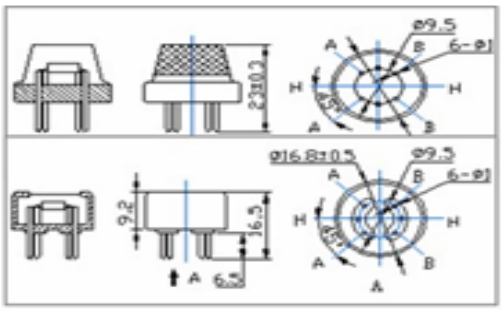
Character

*Good sensitivity to Combustible gas in wide range
* High sensitivity to LPG, Propane and Hydrogen
* Long life and low cost
* Simple drive circuit

Application

* Domestic gas leakage detector
* Industrial Combustible gas detector
* Portable gas detector

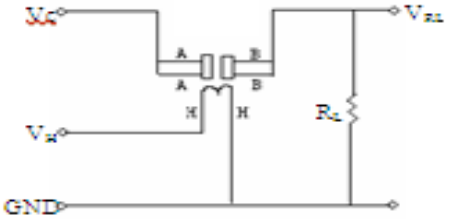
Configuration



Technical Data

Model No.			MQ-2
Sensor Type			Semiconductor
Standard Encapsulation			Bakelite (Black Bakelite)
Detection Gas			Combustible gas and smoke
Concentration			300-10000ppm (Combustible gas)
Circuit	Loop Voltage	V_c	≤24V DC
	Heater Voltage	V_H	5.0V±0.2V ACorDC
	Load Resistance	R_L	Adjustable
Character	Heater Resistance	R_H	31Ω±3Ω (Room Tem.)
	Heater consumption	P_H	≤900mW
	Sensing Resistance	R_s	2KΩ-20KΩ(in 2000ppm C_4H_{10})
	Sensitivity	S	Rs(in air)/Rs(1000ppm isobutane)≥5
	Slope	α	≤0.6($R_{2000ppm}/R_{1000ppm}$ CH_4)
Condition	Tem. Humidity		20℃±2℃ ; 65%±5%RH
	Standard test circuit		Vc:5.0V±0.1V ; V_H: 5.0V±0.1V
	Preheat time		Over 48 hours

Basic test loop



The above is basic test circuit of the sensor. The sensor need to be put 2 voltage, heater voltage (VH), and test voltage (VC). VH used to supply certified working temperature to the sensor, while VC used to detect voltage (VRL) on load resistance (RL), whom is in series with sensor. The sensor has light polarity, Vc need DC power. VC and VH could use same power circuit with precondition to assure performance of sensor. In order to make the sensor with better performance, suitable RL value is needed:
Power of Sensitivity body(Ps):
$Ps=Vc^2\times Rs/(Rs+RL)^2$

Resistance of sensor(Rs): Rs=(Vc/VRL-1)×RL

Sensitivity Characteristics

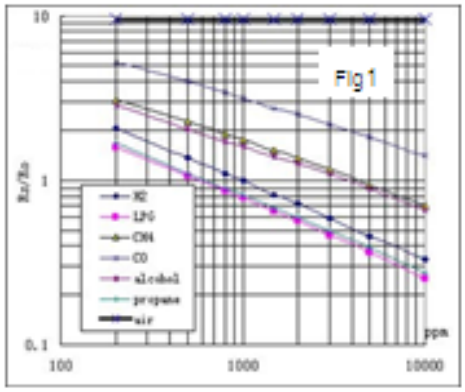


Fig.1 shows the typical sensitivity characteristics of the MQ-2, ordinate means resistance ratio of the sensor (Rs/Ro), abscissa is concentration of gases. Rs means resistance in different gases, Ro means resistance of sensor in 1000ppm Hydrogen. All test are under standard test conditions.

Influence of Temperature/Humidity

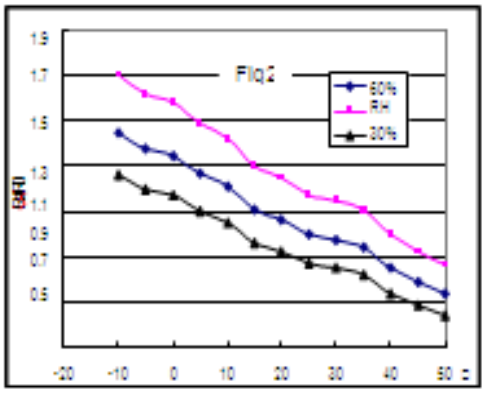
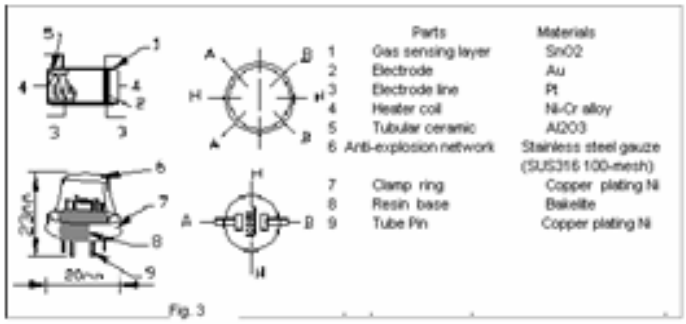


Fig.2 shows the typical temperature and humidity characteristics. Ordinate means resistance ratio of the sensor (Rs/Ro), Rs means resistance of sensor in 1000ppm Butane under different tem. and humidity. Ro means resistance of the sensor in environment of 1000ppm Methane, 20℃/65%RH

Structure and configuration



Structure and configuration of MQ-2 gas sensor is shown as Fig. 3, sensor composed by micro AL2O3 ceramic tube, Tin Dioxide (SnO2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-2 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.

Notification

1 Following conditions must be prohibited

1.1 Exposed to organic silicon steam

Organic silicon steam cause sensors invalid, sensors must be avoid exposing to silicon bond, fixature, silicon latex, putty or plastic contain silicon environment

1.2 High Corrosive gas

If the sensors exposed to high concentration corrosive gas (such as H_2Sz, SO_x, CL_2, HCl etc), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation.

1.3 Alkali, Alkali metals salt, halogen pollution

The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt especially brine, or be exposed to halogen such as fluorin.

1.4 Touch water

Sensitivity of the sensors will be reduced when spattered or dipped in water.

1.5 Freezing

Do avoid icing on sensor surface, otherwise sensor would lose sensitivity.

1.6 Applied voltage higher

Applied voltage on sensor should not be higher than stipulated value, otherwise it cause down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly.

1.7 Voltage on wrong pins

For 6 pins sensor, if apply voltage on 1、3 pins or 4、6 pins, it will make lead broken, and without signal when apply on 2、4 pins

2 Following conditions must be avoided

2.1 Water Condensation

Indoor conditions, slight water condensation will effect sensors performance lightly. However, if water condensation on sensors surface and keep a certain period, sensor' sensitivity will be decreased.

2.2 Used in high gas concentration

No matter the sensor is electrified or not, if long time placed in high gas concentration, if will affect sensors characteristic.

2.3 Long time storage

The sensors resistance produce reversible drift if it's stored for long time without electrify, this drift is related with storage conditions. Sensors should be stored in airproof without silicon gel bag with clean air. For the sensors with long time storage but no electrify, they need long aging time for stbility before using.

2.4 Long time exposed to adverse environment

No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc, it will effect the sensors performance badly.

2.5 Vibration

Continual vibration will result in sensors down-lead response then recture. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration.

2.6 Concussion

If sensors meet strong concussion, it may lead its lead wire disconnected.

2.7 Usage

For sensor, handmade welding is optimal way. If use wave crest welding should meet the following conditions:

2.7.1 Soldering flux: Rosin soldering flux contains least chlorine
2.7.2 Speed: 1-2 Meter/ Minute
2.7.3 Warm-up temperature : 100±20℃
2.7.4 Welding temperature : 250±10℃
2.7.5 1 time pass wave crest welding machine

If disobey the above using terms, sensors sensitivity will be reduced.