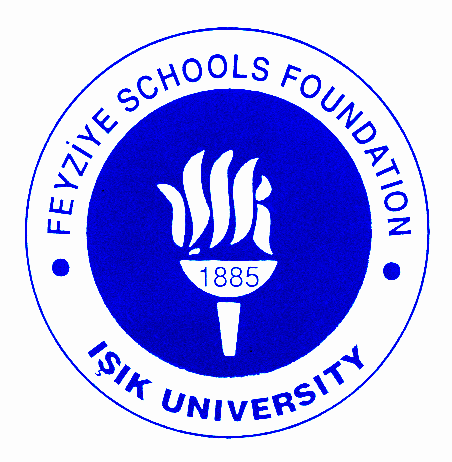
****

**IŞIK UNIVERSITY**

**Faculty of Engineering**

**Department of Electrical and Electronics Engineering**

**Image Processing Unit**

**Project Report**

**by**

**Name / Surname: Şuayb Talha Özçelik**

**Student ID: 215CS2018**

**Name / Surname: Metin Ozan Hazar**

**Student ID: 213MC2256**

**Name / Surname: Anıl Aydıner**

**Student ID: 213CS2226**

**Lecturer : Ast. Prof. Ramazan KÖPRÜ**

**2018-19, spring**

**Abstract**

The ultimate aim in a large number of image processing applications is to extract important features from image data, from which a description, interpratation or understanding of the scene can be provided by the machine. Image processing can be defined as, the processing or altering an existing image in a desired manner. This system allows user to recognize the texts and voicing them from a live stream that is being captured from Raspberry Pi 3 B+. The other features for the mobility that such as is that a WD Car as it carries three different microprocessors and a breadboard. Car also senses gas leak as it has a MQ-9 Gas Sensor.

This project aims to optimize the microprocessors to work in an efficent way and giving them kinds of different works to make it diversified.

## Contents

[Abstract ii](#_Toc447097068)

[Contents ııı](#_Toc447097070)

[Chapter 1 Introduction 1](#_Toc447097073)

[1.1 Overview 1](#_Toc447097074)

[1.2 Motivation 1](#_Toc447097075)

[Chapter 2 Project Title 1](#_Toc447097077)

[2.1 Project Description 1](#_Toc447097080)

[2.1.1 Hardware 1](#_Toc447097083)

[2.1.2 Software 1](#_Toc447097083)

[2.2 Design Details 1](#_Toc447097080)

[Chapter 3 Conclusions and Discussions 1](#_Toc447097090)

[References 1](#_Toc447097094)

[Acknowledgments 1](#_Toc447097094)

**1.Chapter 1 - Introduction**

**1.1 Overview**

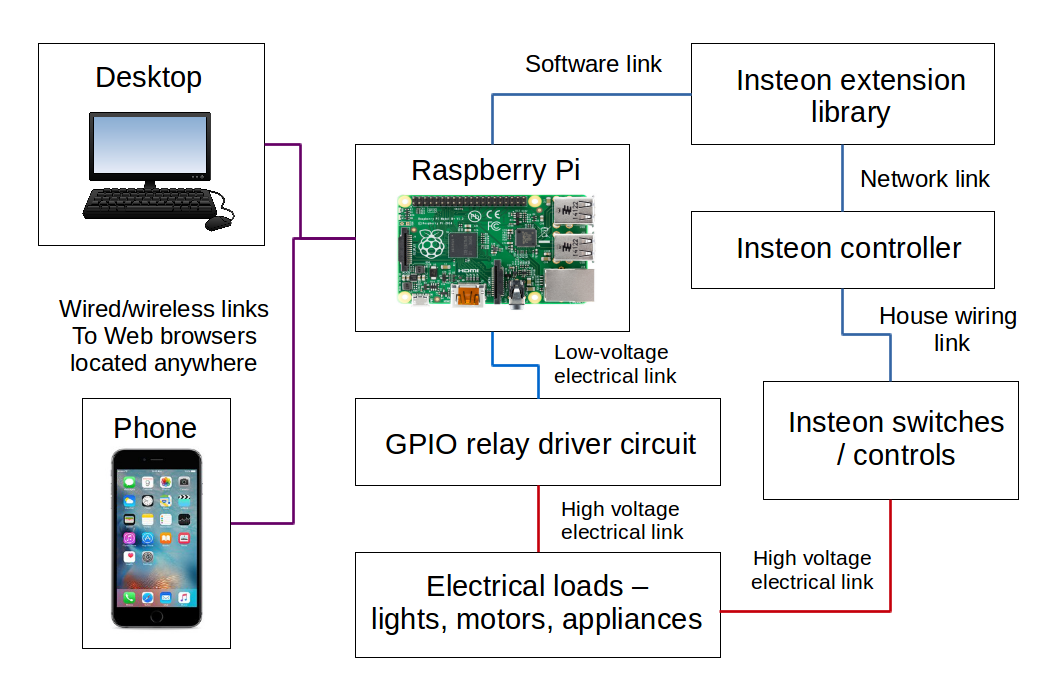
In the 21st century, Internet has become a worldwide mean of communication and a place to express ideas. Using internet in electrical engineering project is also become a phenomenon. People use mobile phones in every where we think that if we combine electrical engineering and android devices maybe it become more useful. In our group we have 2 Computer Engineers we have a code knowledge so we try to make text recognition in android devices with raspberry pi camera . We try to transfer the live stream video to android devices. In this way we see our raspberry camera image in every where with internet. We build the raspberry camera in car devices to make it portable. So our document is about car device(motor control), gas sensor, Raspberry Pi 3 B+, Raspberry Cam, Android Studio coding to image process.

**1.2 Motivation**

In this project our main idea is image processing unit and gas sensor on the car for understand this mechanism place’s condition. With camera we can see the environment and recognize the texts, with gas sensor we can learn the carbon monoxide level of environment ,with car we can move the mechanism to forward,back, left or right direction. If an environment has a risky to enter, we can send the our mechanism to understand the risk level or see the environment. In the future we can add a arm mechanism to do a job in this environment.We try to do this mechanism with a small budget but in the future we can develop this mechanism to more quality equipment to use this kind of area.

**2.Chapter 2 - Image Processing Unit**

We use raspberry like this to connect phone raspberry and sensors:



**2.1.1 Hardware**

**Raspberry Pi 3 Model B+**

Size(cm): 68.63 x 94.09 x 26.63

Weight: 70 gr

5V / 2.5A DC Power input



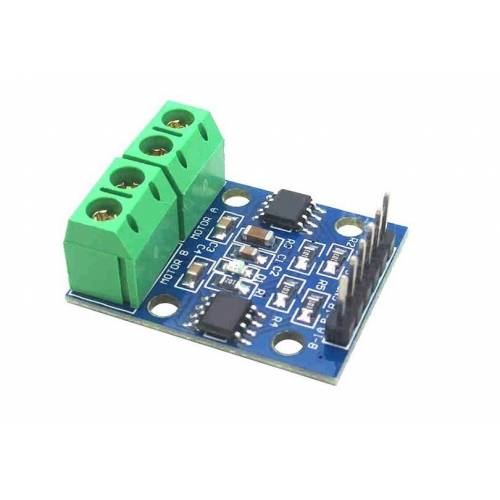
## L9110 Dual Motor Driver Board

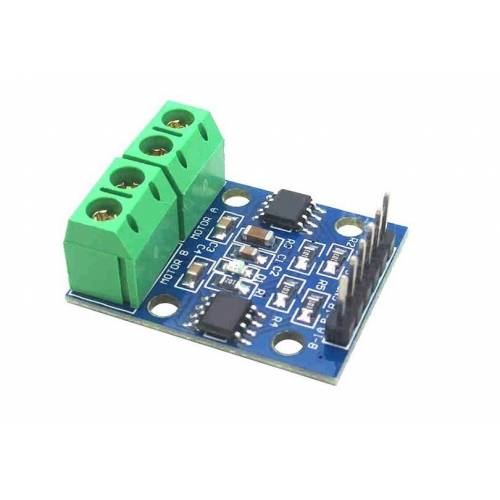
Sizes : 31 mm x 22 mm

Weight : 7 gr

2.5V-12V/2A DC power output

Each channel 800 mA





**Carbon monoxide and Combustible Gas Sensor Board - MQ-9**

Sizes: 32mm x 20mm x 22mm

5V power output

Operating voltage : 5.0V

Combustible Gas Concentration: 10-10000ppm

Concentration of Carbon Monoxide (CO): 10-1000ppm



**Arduino UNO R3**

Operating Voltage : 5V

Input Voltage (recommended): 7-12V

Input Voltage (limit):6-20V

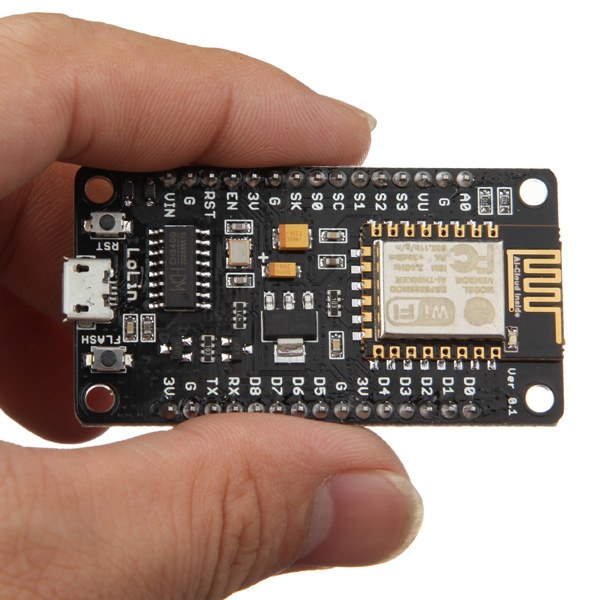
Digital I / O Pins: 14 (6 products PWM output)



**ESP8266 NodeMCU Cp2102 V2 Development Card**

Power - 5V via micro usb port

Sizes – 49 x 24.5 x 13mm



**4wd Smart Car**

Top and bottom transparent plexi body

4 Pcs Engine

4 Piece Wheels

Metal Spacers

4-way Battery Holder

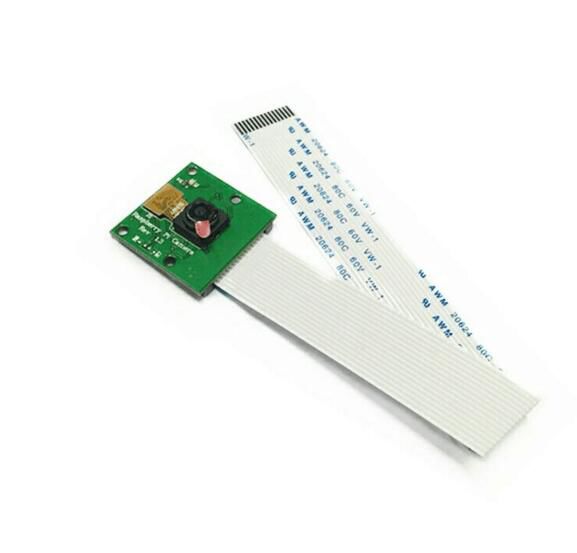
2 Piece Encoder Disc

Various screws and nuts



**Raspberry Pi Camera**

Sizes: 25x20x9mm  
Resolution: 5 MP (2592x1944 piksel)  
Video Shooting : 1080p, 720p and 640x480p

****

**Gear Motor**

**Features:**

**Motor:**

1. Voltage: DC 3V-6V

2. Now: 100 MA-120MA

3. Reduction ratio:: 48: 1

4. Cycle (elastic): 100-240

5. Rubber Diameter:65mm

6. Car Speed (M / minutes): 20-48

7. Motor Weight (g): 29 / her

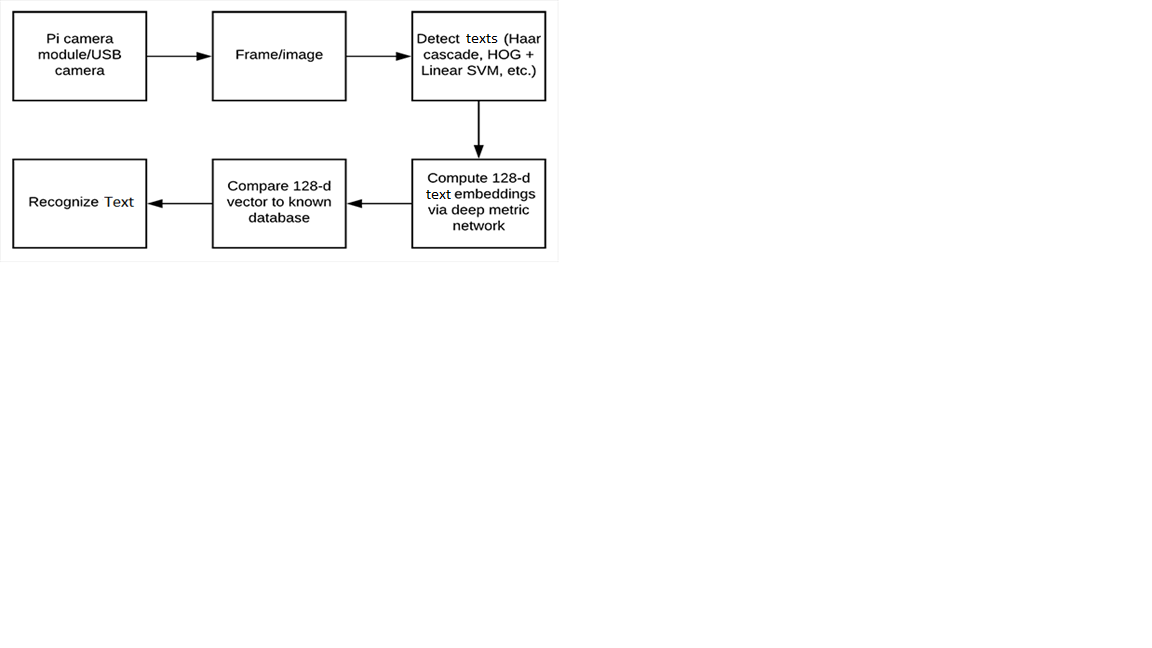
8. Engine Size: 70 mm x 22 mm x 18 mm

9. Desibel: <65dB



**2.1.1 Software**

**Text Recognize:**



Text Recognize Process Diagram

Text Recognize Android Studio Code:

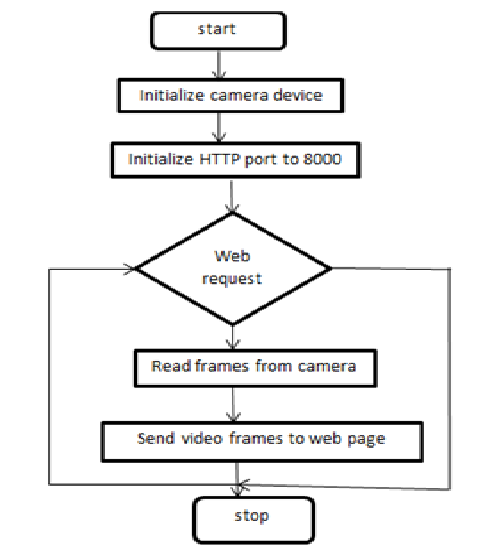
**public void** textRecognize(Bitmap photo){  
 FirebaseVisionImage image = FirebaseVisionImage.*fromBitmap*(photo);  
 FirebaseVisionTextRecognizer detector = FirebaseVision.*getInstance*()  
 .getOnDeviceTextRecognizer();  
 Task<FirebaseVisionText> result =  
 detector.processImage(image)  
 .addOnSuccessListener(**new** OnSuccessListener<FirebaseVisionText>() {  
 @Override  
 **public void** onSuccess(FirebaseVisionText firebaseVisionText) {  
 **for**(FirebaseVisionText.TextBlock block:firebaseVisionText.getTextBlocks()){  
 Rect rect=block.getBoundingBox();  
 Point[] corner=block.getCornerPoints();  
 **text**=block.getText();  
 Toast.*makeText*(Image.**this**,**"t: "**+**text**,Toast.***LENGTH\_SHORT***).show();  
 AlertDialog.Builder builder1=**new** AlertDialog.Builder(Image.**this**);  
 builder1.setTitle(**"Your Text"**);  
 builder1.setMessage(**"Recognized Text: "**+ **text**);  
 builder1.setCancelable(**false**);  
 builder1.setPositiveButton(**"Repeat"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialogInterface, **int** i) {  
  
 }  
 });  
 builder1.setNegativeButton(**"Listen"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 speak();  
 }  
 });  
 builder1.show();}  
 }  
 })  
 .addOnFailureListener(  
 **new** OnFailureListener() {  
 @Override  
 **public void** onFailure(@NonNull Exception e) {  
 Toast.*makeText*(Image.**this**,**"Failed to Recognize"**,Toast.***LENGTH\_SHORT***).show();  
 }  
 });  
}

**Android Studio Manifest Document:**

Maniferest document is so important because we want user to allow their camera or network resource to use in our application. We have to determine it and publish it.

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.company.rashminpc.mqtttest"**>  
  
 <**uses-permission android:name="android.permission.WAKE\_LOCK"** />  
 <**uses-permission android:name="android.permission.INTERNET"** />  
 <**uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"** />  
 <**uses-permission android:name="android.permission.READ\_PHONE\_STATE"** />  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@drawable/logo3"  
 android:label="Image Processing Unit"  
 android:roundIcon="@drawable/logo3"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 <**activity android:name=".Main2Activity"** />  
  
 <**service android:name="org.eclipse.paho.android.service.MqttService"** />  
  
 <**activity android:name=".MainActivity"** />  
 <**activity android:name=".Main3Activity"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 <**activity android:name=".CarControl"** />  
 <**activity android:name=".Image"** />  
 <**activity android:name=".MainPage"**></**activity**>  
 </**application**>  
  
</**manifest**>

Flow Chart:



Car Control Ardunio Code:

#include <SoftwareSerial.h>

SoftwareSerial ArduinoUno(2,3);

#define MotorA1 8

#define MotorA2 9

#define MotorB1 6

#define MotorB2 7

void setup() {

pinMode(MotorA1, OUTPUT);

pinMode(MotorA2, OUTPUT);

pinMode(MotorB1, OUTPUT);

pinMode(MotorB2, OUTPUT);

Serial.begin(9600);

ArduinoUno.begin(4800);

}

void loop() {

while(ArduinoUno.available()>0){

float val= ArduinoUno.parseFloat();

if(ArduinoUno.read()=='\n'){

Serial.println(val);// this is the value coming from nodemcu

if(val==1){//if val is equal to 1 is going to move forward

digitalWrite(MotorA1, HIGH);

digitalWrite(MotorA2, LOW);

digitalWrite(MotorB1, HIGH);

digitalWrite(MotorB2, LOW);

delay(2000);

digitalWrite(MotorA1, LOW);

digitalWrite(MotorA2, LOW);

digitalWrite(MotorB1, LOW);

digitalWrite(MotorB2, LOW);

}

if(val==2){//if val is equal to 2 is going to move backward

digitalWrite(MotorA1, LOW);

digitalWrite(MotorA2, HIGH);

digitalWrite(MotorB1, LOW);

digitalWrite(MotorB2, HIGH);

delay(2000);

digitalWrite(MotorA1, LOW);

digitalWrite(MotorA2, LOW);

digitalWrite(MotorB1, LOW);

digitalWrite(MotorB2, LOW);

}

if(val==3){//if val is equal to 3 is going to move right

digitalWrite(MotorA1, HIGH);

digitalWrite(MotorA2, LOW);

digitalWrite(MotorB1, LOW);

digitalWrite(MotorB2, HIGH);

delay(2000);

digitalWrite(MotorA1, LOW);

digitalWrite(MotorA2, LOW);

digitalWrite(MotorB1, LOW);

digitalWrite(MotorB2, LOW);

}

if(val==4){//if val is equal to 4 is going to move left

digitalWrite(MotorA1, LOW);

digitalWrite(MotorA2, HIGH);

digitalWrite(MotorB1, HIGH);

digitalWrite(MotorB2, LOW);

delay(2000);

digitalWrite(MotorA1, LOW);

digitalWrite(MotorA2, LOW);

digitalWrite(MotorB1, LOW);

digitalWrite(MotorB2, LOW);

}

}

}

}

Gas Sensor NodeMcu Code

void setup() {

pinMode(BUILTIN\_LED, OUTPUT);

pinMode(D5, INPUT);

pinMode(D7, OUTPUT);

Serial.begin(9600);

ESP8266WiFi.begin(4800);

setup\_wifi();

client.setServer(mqtt\_server, 10876);

client.setCallback(callback);

reconnect();

server.on("/sensorvalue.txt", [](){

text = (String)sensorValue;

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

});

server.on("/", [](){

page = "<h1>Sensor to Node MCU Web Server</h1><h1>Sensor Value:</h1> <h1 id=\"sensorvalue\">""</h1>\r\n";

page += "<script>\r\n";

page += "var x = setInterval(function() {loadData(\"sensorvalue.txt\",updateData)}, 1000);\r\n";

page += "function loadData(url, callback){\r\n";

page += "var xhttp = new XMLHttpRequest();\r\n";

page += "xhttp.onreadystatechange = function(){\r\n";

page += " if(this.readyState == 4 && this.status == 200){\r\n";

page += " callback.apply(xhttp);\r\n";

page += " }\r\n";

page += "};\r\n";

page += "xhttp.open(\"GET\", url, true);\r\n";

page += "xhttp.send();\r\n";

page += "}\r\n";

page += "function updateData(){\r\n";

page += " document.getElementById(\"sensorvalue\").innerHTML = this.responseText;\r\n";

page += "}\r\n";

page += "</script>\r\n";

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

});

//server.begin();

//Serial.println("Web Server Started");

server2.begin();

}

void loop() {

WiFiClient client2= server2.available();

if (!client.connected()){

reconnect();

}

client.loop();

http.get("http://api.ipify.org");

String reqmotor = client2.readStringUntil('\r');

Serial.println(reqmotor);

client2.flush();

if(reqmotor.indexOf("/forward")!= -1){

ESP8266WiFi.print(1);

ESP8266WiFi.println("\n");

Serial.println("Moved forward");

}

if(reqmotor.indexOf("/backwards")!= -1){

ESP8266WiFi.print(2);

ESP8266WiFi.println("\n");

}

if(reqmotor.indexOf("/left")!= -1){

ESP8266WiFi.print(3);

ESP8266WiFi.println("\n");

}

if(reqmotor.indexOf("/right")!= -1){

ESP8266WiFi.print(4);

ESP8266WiFi.println("\n");

}

client2.flush();

String s = "HTTP/1.1 200 OK\r\n";

s += "Content-Type: application/json\r\n\r\n";

s += "{\"data\":{\"message\":\"success\",\"value\":\"";

s += value;

s += "\"}}\r\n";

s += "\n";

// Send the response to the client

client2.print(s);

delay(1);

Serial.println("Client disconnected");

sensorValue = analogRead(A0);

Serial.println("Current gas value "+String(sensorValue));

server.handleClient();

String gas= String(sensorValue);

//ESP8266WiFi.print(sensorValue);

//ESP8266WiFi.println("\n");

Serial.println("Publish message: ");

Serial.println(gas);

int numgas= sensorValue;

char csgas[16];

itoa(numgas, csgas, 10);

delay(1500);

client.publish("Gas", csgas);

}

**2.2 Design Details**

The system specification shows the description of the function and the performance of system and the user. The scope of our project ”Image Processing Unit” is immense. The future implications of the project are very great considering the amount of time and possible hazards it saves. The project we try to make text recognition in android devices with raspberry pi camera . This system is aimed to see the environment and recognize the texts, with gas sensor we can learn the carbon monoxide level of environment ,with car we can move the mechanism to forward,back, left or right direction.

Along the course of project completion we encountered various problems and obstacles. Not everything that we had planned went smoothly during the project development span. Also we had a limited amount of time for its completion so we were under a certain amount of pressure as well. We had to start from the research phase at the beginning and needed to gain knowledge on all the devices and components that we had intended to use for our project. Other phases of the project included coding, debugging, testing, documentation and implementation and it needed certain time for completion so we really had to manage the limited time available to us and work accordingly to finish the project within the schedule.

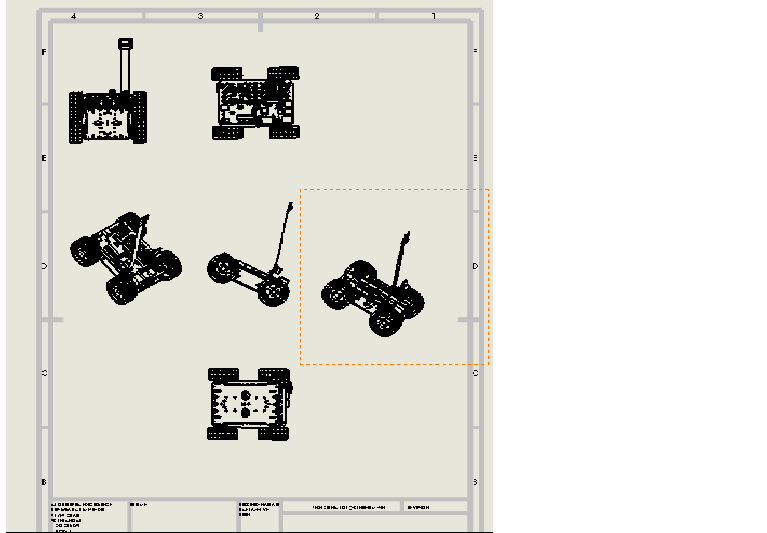
**Constraints Considerations**

The following is a list of constraint considerations:

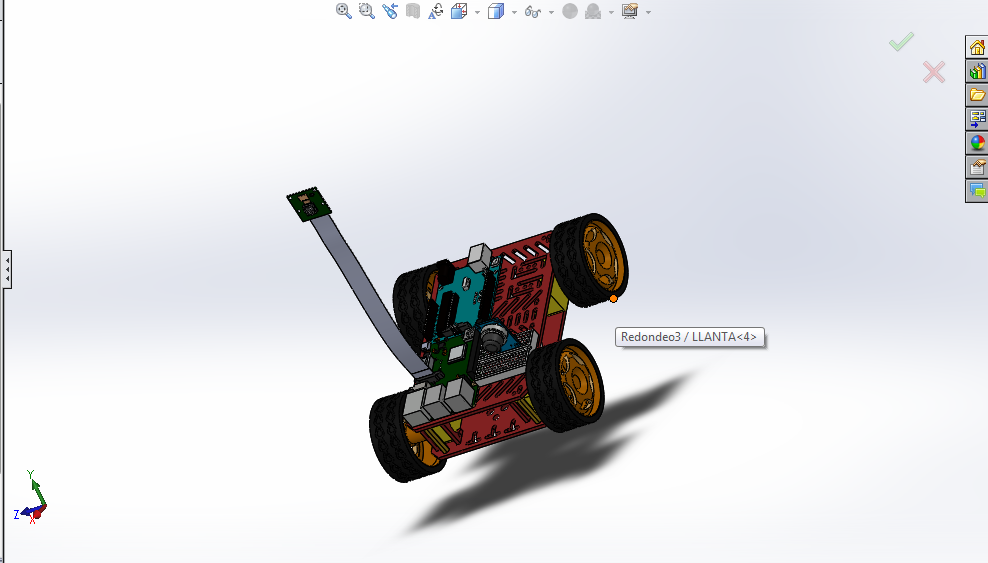
1- The PCB hardware will need an seriall control interface. This system is only capable of controlling electrical devices.

2- The hardware module will need to be shielded against electrical fluctuations. This will help in increasing the reliability of the syste

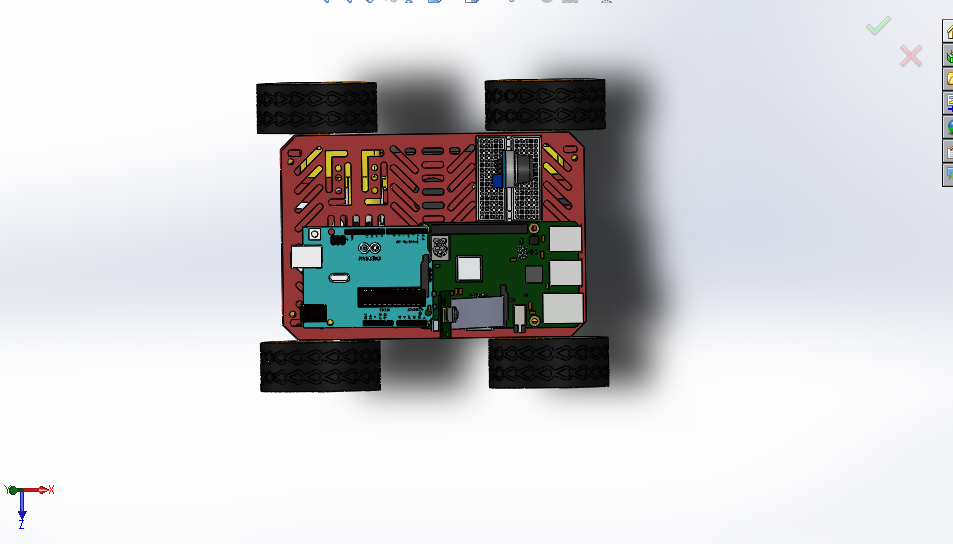
**Our Project Designs in Solid:**

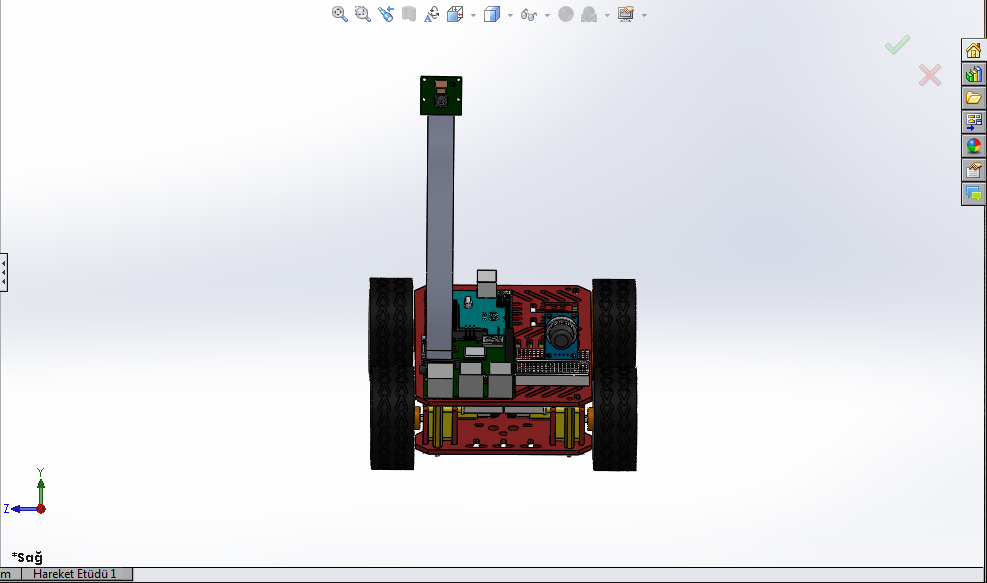


-Technical Image-



-Solid Works Designs-



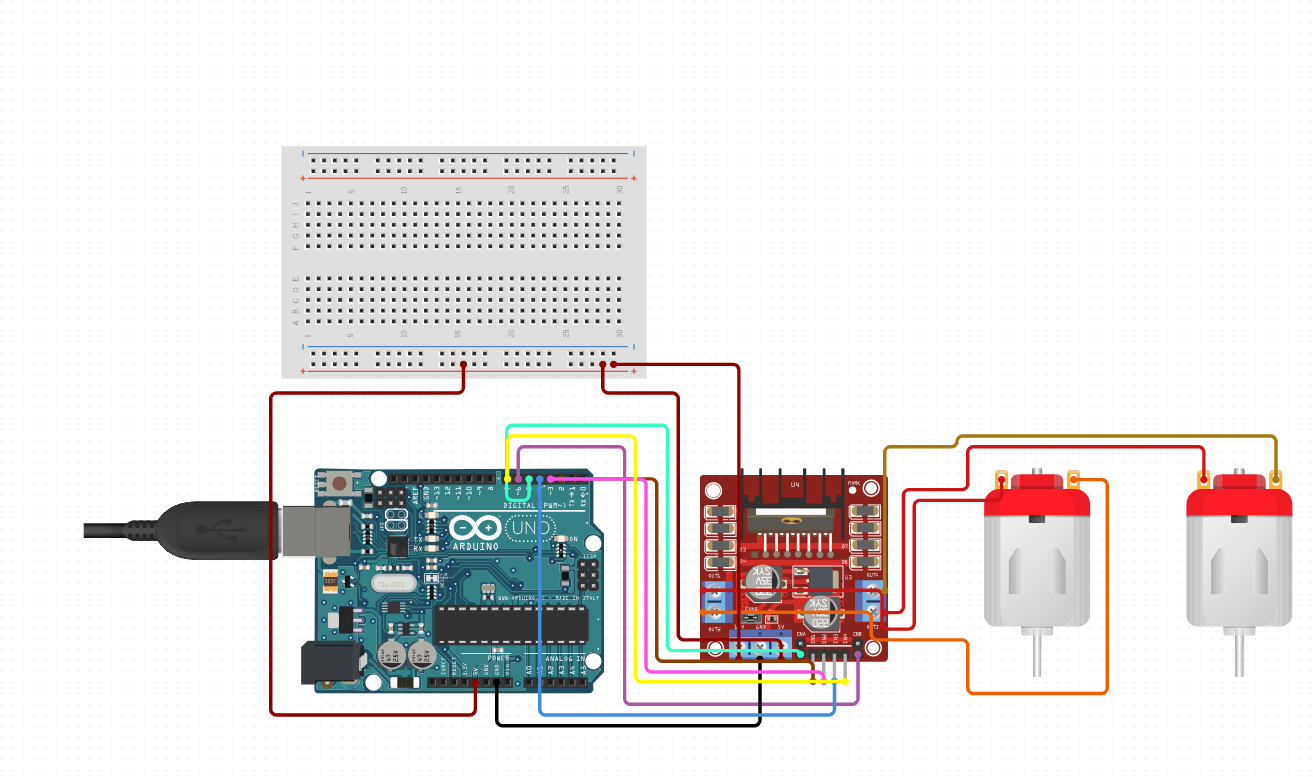


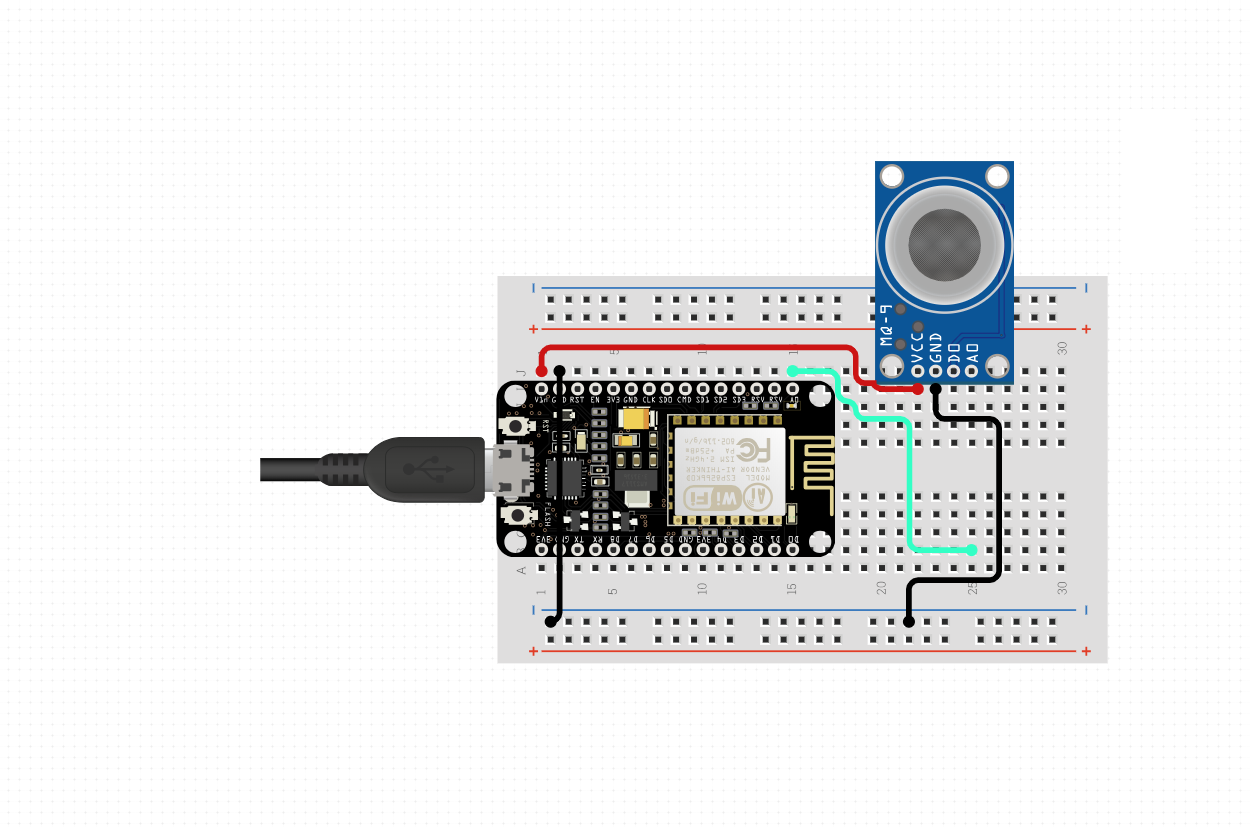
Ardunio has 2 motor driver connection. The terminals of the IN 1-2-3-4 pin from the motor driver are connected to Ardunio's pin 2-3-4-7. Pin3 is connecting to Pwm(Pulse Width Modulation) because it is used to create a square wave, a signal switched between on and off.

What is Pwm?

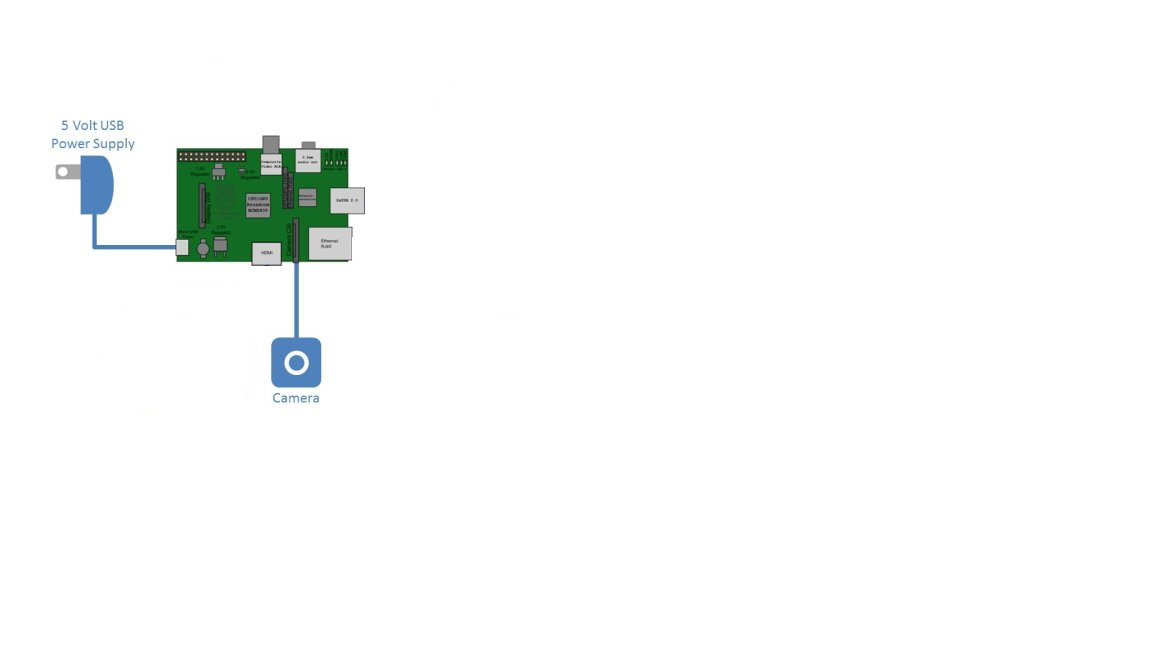
Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width.

Motor Driver’s ENB is connected to the 3rd Pin of the Arduino. Motor Driver outputs are connected to 2 motors. Arduino takes its power from the computer USB port.





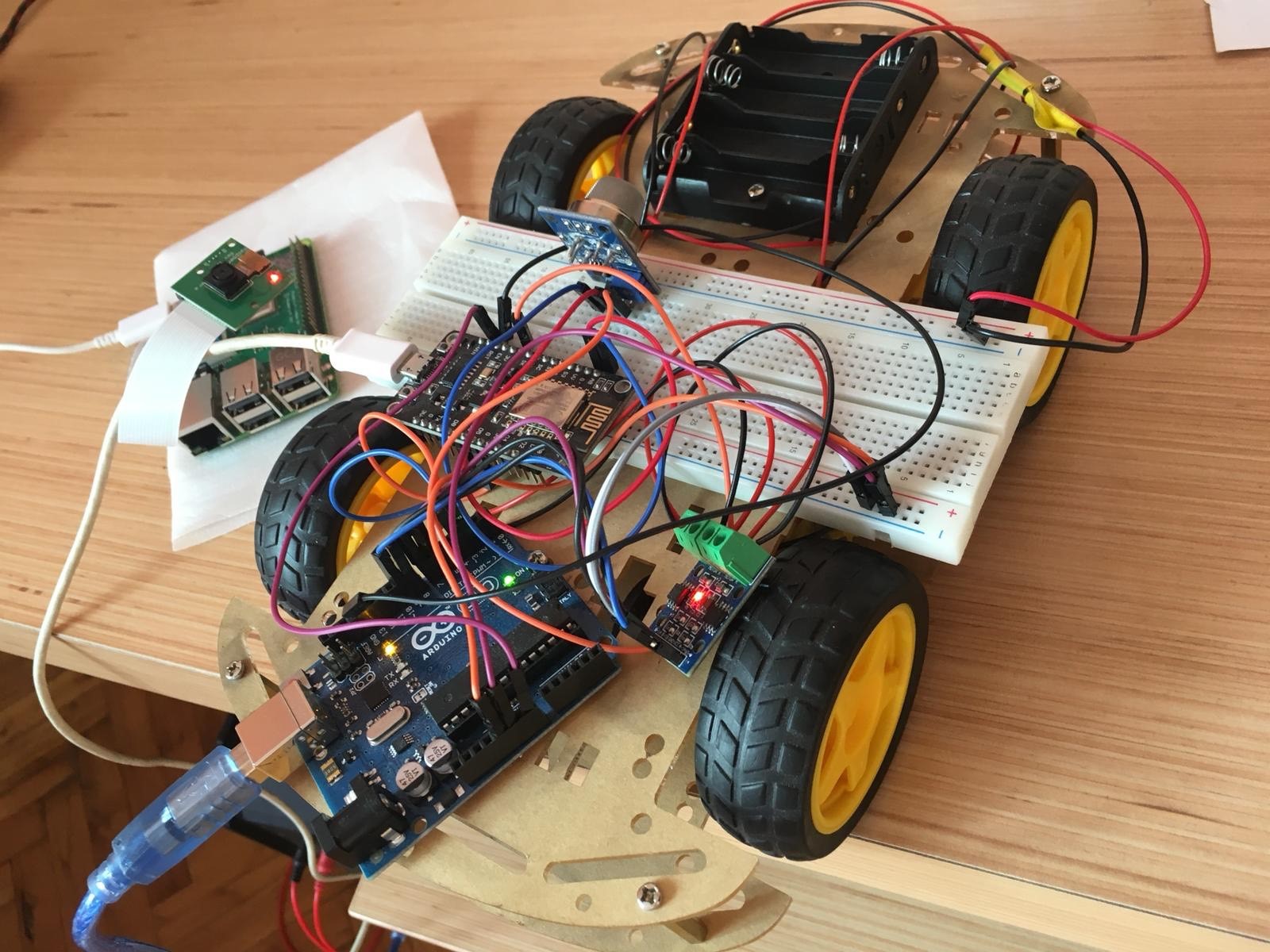
NodeMcu’s VIN is connected to Gas Sensor’s VCC pin. Gas Sensor’s ground is connected to NodeMcu’s ground. Gas Sensor’s D0 pin is connected to NodeMcu ‘s A0 pin.



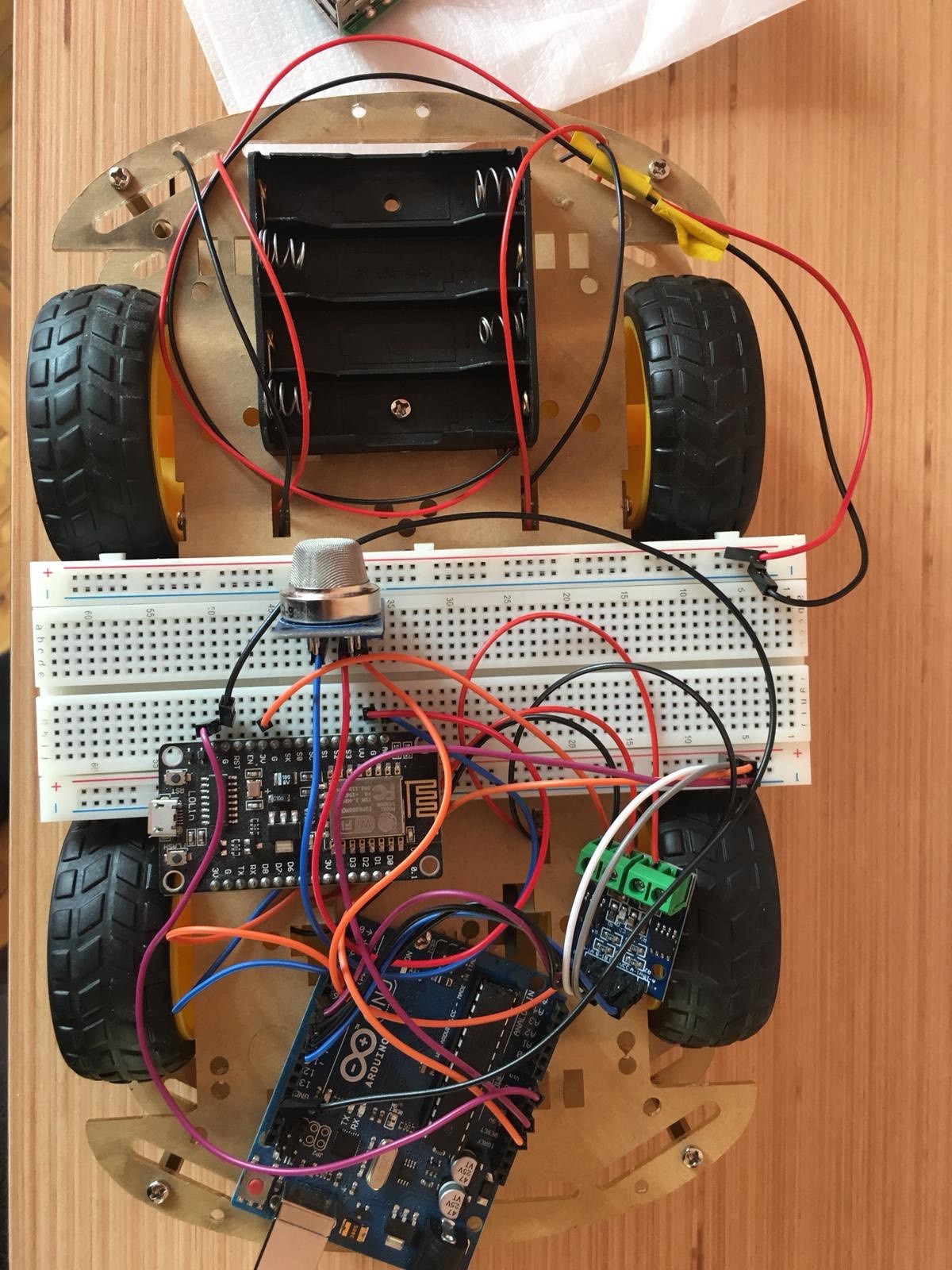
Camera connector is connected to Raspberry’s camera port.

Ardunio and NodeMcu is connected with serial.Gas Sensor’s value is sended by NodeMcu to Web Server. NodeMcu and Arduino takes its power from the computer USB port .Raspberry Pi takes its power from adapter(5V 2.5A ).

**Pictures From Our Project**







**3.Chapter Discussion and Conclusion**

**3.1 Discussion**

As this project is doing multiple works, not just image processing, we have worked with three different microprocessors which are NodeMcu ESP-8266, Arduino UNO and Raspberry Pi 3 B+ to give each of them fair workload. It would have been better if we had a higher budget for this project since we have used not-so-high quality materials, i.e motors. Current motors that we use don’t have enough power to move the whole part of the car. We should have managed to lower the delay of the packets that are sent for running the motors as they sometimes run delayed when the commands are sent through the Android application.

**3.2 Conclusion**

In the final, being able to finish this project has shifted our motivation upwards for the bigger projects in the future. It was a big move for us to optimize an android application with microprocessors which helped us to have a better understanding of IOT. We successfully managed to control the three different microprocessor and made them communicate with each other which enabled us to reduce their workloads to run in underload. This project helped us to understand the techniques of working with multiple microprocessors and have taught us the fundamentals of image processing.

**References**

[1] Ramazan Köprü, “On PWM generation via Raspberry platform”, April 1, 2019, *Radioengineering*, pp. 12-34.

[2] Isaac Newton, “Planatery motion equations”, April 1, 1717, *Physics*, pp. 212-134.

[3] <http://helloraspberrypi.blogspot.com/2016/07/stream-video-from-raspberry-pi-3-camera.html>

[4] <https://www.sitepoint.com/a-step-by-step-guide-to-building-an-android-audio-player-app/>

**Acknowledgments**

We would like to extend our sincere thanks to Ali Hür for his invaluable contribution in developing the hardware

I am also grateful to Prof. Mehmet Öz for his guidance in motor control theory and PWM generation.