EE-222: Microprocessor Systems

Project Proposal

Web-Controlled Multifunctional Car

Group Members

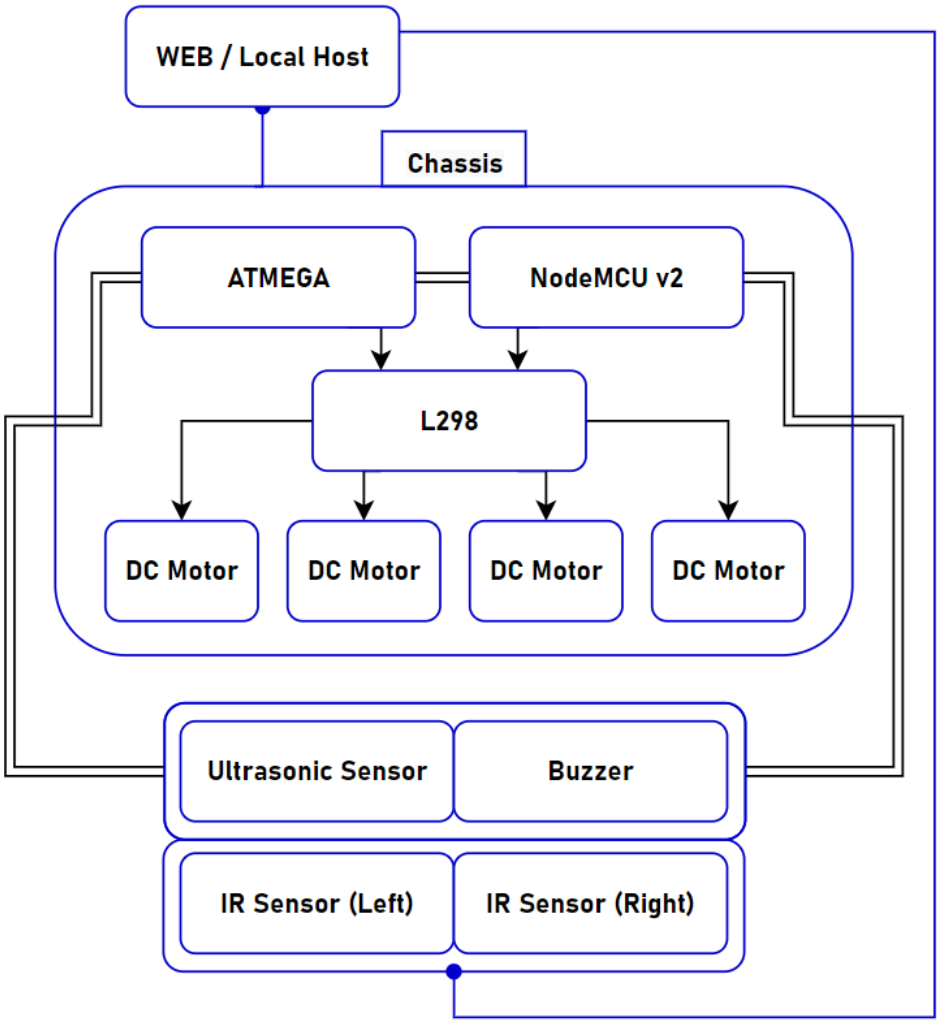
Name	Reg. No
Muhammad Ahmed Mohsin	333060
Muhammad Umer	345834
Tariq Umar	334943

Abstract

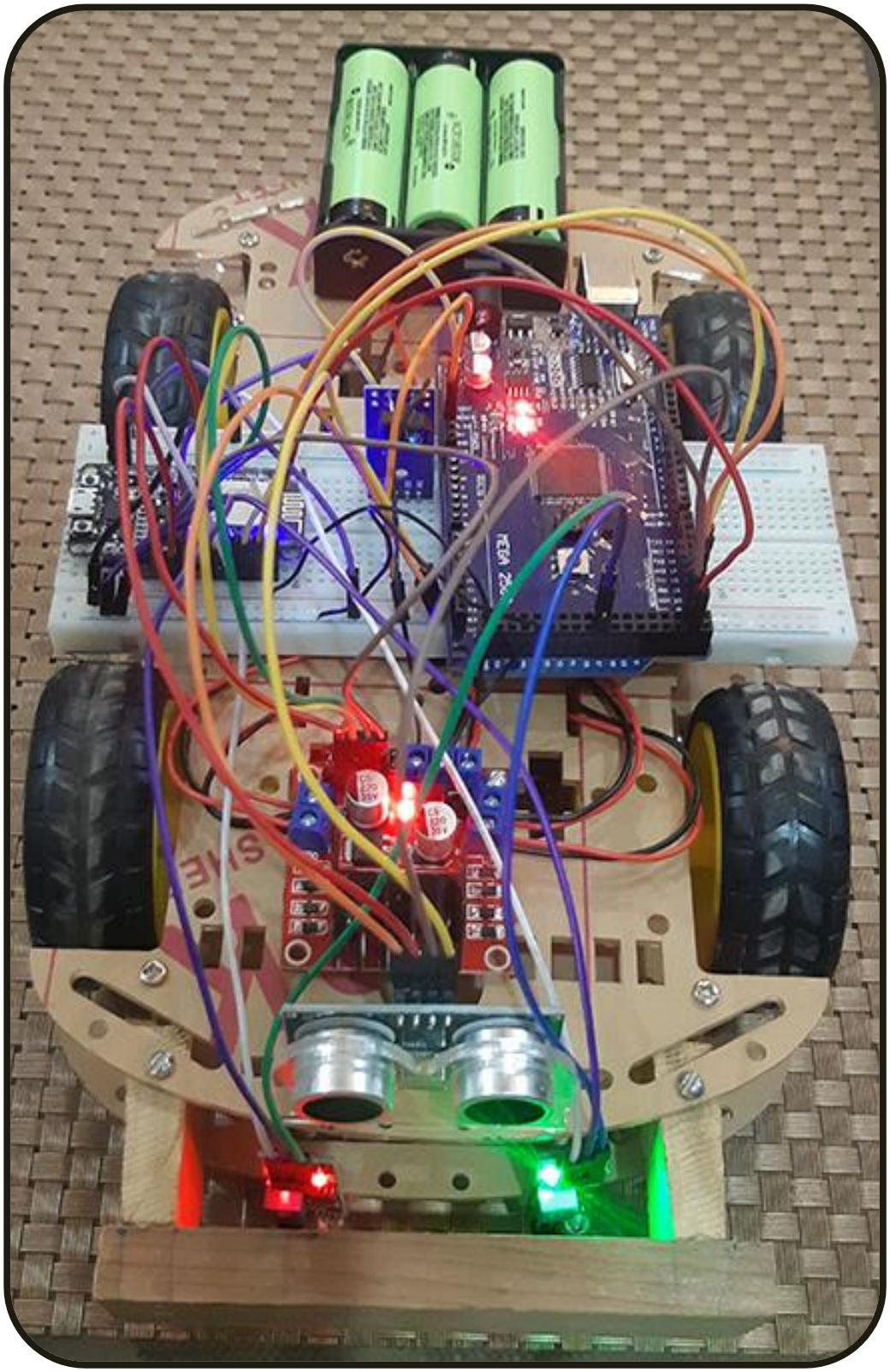
In this project, we aim to develop a Web-Controlled Multifunctional Car. We hope to implement features such as line following, obstacle detection etc. along with some specialized features such as light sensors and buzzers. All of this will be implemented with an ATMEL microcontroller as the heart of our project.

Other hardware components will include NodeMCU (ESP8266 Wi-Fi-Module) along with L298 Motor Driver. Moreover, we also aim to develop a mobile app or a website to control the car. Our objective is to develop skills to use microprocessor to implement advanced functions on hardware. This will give us an insight on how to implement the theoretical knowledge on hardware and what are the difficulties faced during the process. We can further extend our functionality from detection to identification in order to make our car as an information gathering device over a long-ranged Wi-Fi network.

Block Diagram

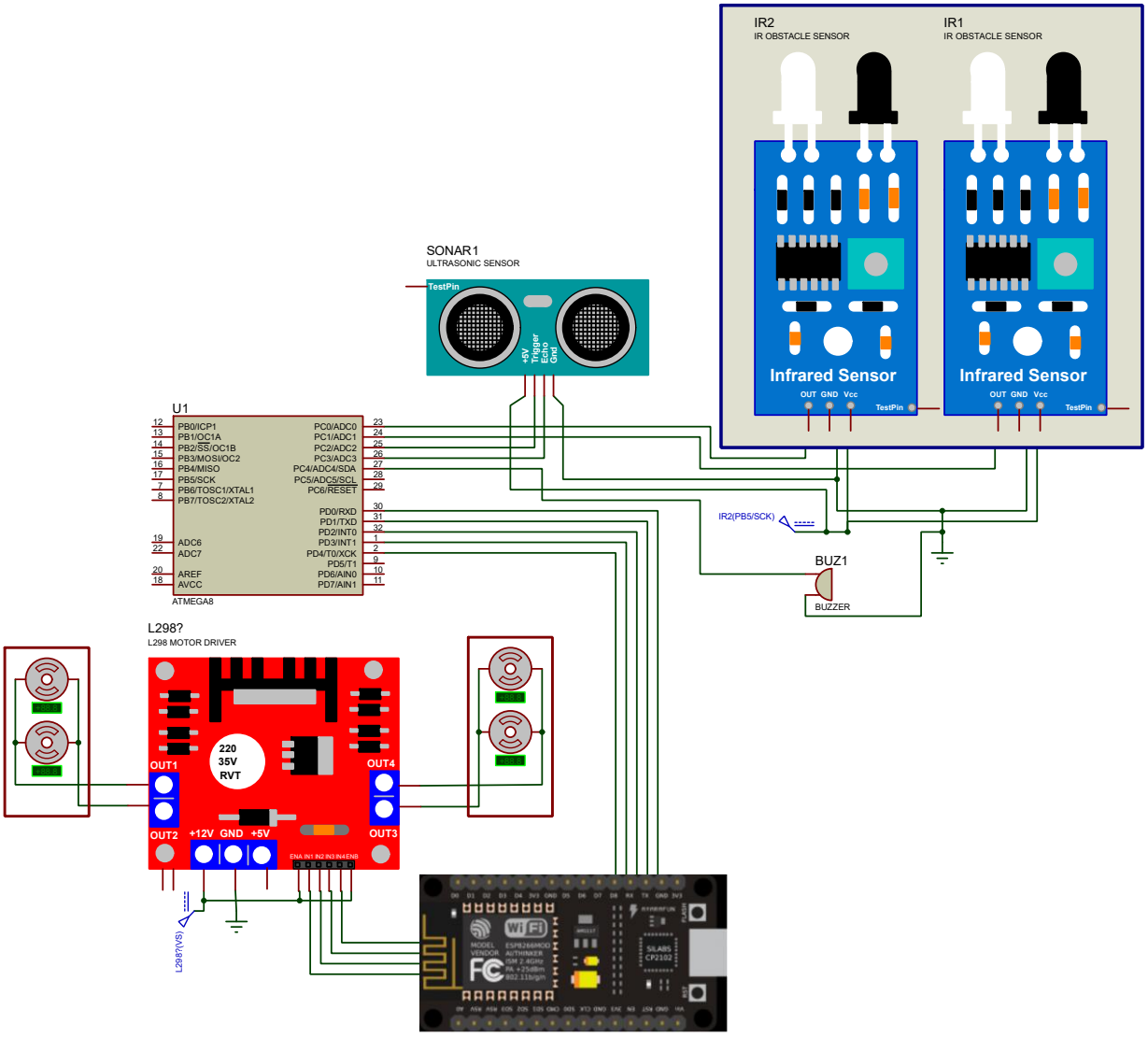


Diagram



Practical Model

Schematic



Design Methodology

- **Web Sever / Local Host**

NodeMCU is a low-cost open source IoT platform. It initially included firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which was based on the ESP-12 module. Later, support for the ESP32 32-bit MCU was added. We used the Web Server / Local host and used its in built Wi-Fi module to connect with the APK installed in our phones and generate a local IP address which then helped in performing distinct functions built through the cellphone application.

- **L298 Motor Driver**

This L298N Motor Driver Module is a high power motor driver module for driving DC and Stepper Motors. This module consists of an L298 motor driver IC and a 78M05 5V regulator. L298N Module can control up to 4 DC motors, or 2 DC motors with directional and speed control. We used this module to drive our motors and to perform basic functions like move forward, backward, etc. It is used to control the motors and the movement of the car to perform different web controlled functions. The L298N Motor Driver module consists of an L298 Motor Driver IC, 78M05 Voltage Regulator, resistors, capacitor, Power LED, 5V jumper in an integrated circuit.

Module Description

- **ATMega**

The ATmega16A is a low-power CMOS 8-bit microcontroller based on the Atmel AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16A achieves throughputs approaching 1MIPS per MHz allowing the system designer to optimize power consumption versus processing speed. It was basically used to send signals to the sensors in our project. The Atmega16 was coded and burnt and then was used to send signals and instructions to the sensors.

- IR Sensors

Infrared Sensor Module has built-in IR transmitter and IR receiver that sends out IR rays and looks for reflected IR rays to detect presence of any obstacle in front of the sensor module.IR sensors were used to implement the function of line following. Both the sensors were place with an equal distance from the ground and with the same sensitivity and with the help of both of them we implemented line tracking system. This line tracking system was in built in the application as well.

- **Ultrasonic Sensor**

SR-04 Ultrasonic Sensor Module is a 4 - pin module, pin names are Vcc, Trigger, Echo and Ground respectively.

$$\text{Distance} = \text{Speed} \times \text{Time}$$

HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver.

We used this sensor for obstacle avoidance. The code was built such that at a distance of about 15cm from the obstacle and then rotates 90 degrees to the left and starts moving. When it detects the object it stops and then a beep from buzzer is played then the car rotates 90 degrees and then starts moving.

- **DC Motors**

This Micro DC Gear Motor with Shaft is ideal for DIY enthusiasts and is designed to easily incorporate the encoder. These motors are inexpensive, small, easy to install, and ideally suited for use in a mobile robot car. These motors are controlled by L298 motor driver where two motors each are connected in common and the functionality and operation of these motors are controlled by the motor driver.

Functionality
Line Following
Web – Controlled Movement
Obstacle Detection & Avoidance

Components
ATMEGA 8/16
NodeMCU V2 (ESP8266)
L298N Motor Driver
4 x DC Motors
3 x Li-Ion Batteries (3.3V)
Breadboard
Car Chassis
Buzzer
2 x IR Sensor
Ultrasonic Sensor

Relevant Industries
LRRC Robotics
IoT (Internet of Things)
Information Nodes

.INO Codes

NodeMCU

```
#define IN_1  D8          // L298N in1 motors Rightx          GPIO15(D8)
#define IN_2  D7          // L298N in2 motors Right           GPIO13(D7)
#define IN_3  D4           // L298N in3 motors Left            GPIO2(D4)
#define IN_4  D3           // L298N in4 motors Left            GPIO0(D3)

#define TRIG_SIGNAL D5
#define ECHO D6

#define LT1 D2
#define LT2 D1

#define LT_SIGNAL D0

#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>

/* data received from application */
String  data = "";

int speedCar = 400;         // 400 - 1023.
int speed_Coeff = 3;

long duration;
int distance;

/* define port */
ESP8266WebServer server(80);

/* WIFI settings */
const char* ssid = "Car";

void setup() {
 Serial.begin(115200);
 pinMode(TRIG_SIGNAL, OUTPUT);
 pinMode(ECHO, INPUT);
 pinMode(IN_1, OUTPUT);
```

```
 pinMode(IN_2, OUTPUT);
 pinMode(IN_3, OUTPUT);
 pinMode(IN_4, OUTPUT);

 pinMode(LT1, INPUT);
 pinMode(LT2, INPUT);

  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid);
//
  IPAddress myIP = WiFi.softAPIP();
  Serial.print("AP IP address: ");
  Serial.println(myIP);

 // Starting WEB-server
     server.on ( "/", HTTP_handleRoot );
     server.onNotFound ( HTTP_handleRoot );
     server.begin();
}

void goBack(){

      digitalWrite(IN_1, LOW);
      digitalWrite(IN_2, HIGH);

      digitalWrite(IN_3, LOW);
      digitalWrite(IN_4, HIGH);
  }

void goAhead(){

      digitalWrite(IN_1, HIGH);
      digitalWrite(IN_2, LOW);

      digitalWrite(IN_3, HIGH);
      digitalWrite(IN_4, LOW);
  }

void goRight(){

      digitalWrite(IN_1, HIGH);
      digitalWrite(IN_2, LOW);

      digitalWrite(IN_3, LOW);
      digitalWrite(IN_4, HIGH);
  }

void goLeft(){

      digitalWrite(IN_1, LOW);
      digitalWrite(IN_2, HIGH);
```

```
      digitalWrite(IN_3, HIGH);
      digitalWrite(IN_4, LOW);
  }

void stopRobot(){

      digitalWrite(IN_1, LOW);
      digitalWrite(IN_2, LOW);

      digitalWrite(IN_3, LOW);
      digitalWrite(IN_4, LOW);
 }

void lineTracker(){
      digitalWrite(LT_SIGNAL, HIGH);
      //line detected by both
      if(digitalRead(LT1)==0 && digitalRead(LT2)==0){
        //Forward
        goAhead();
      }
      //line detected by left sensor
      else if(digitalRead(LT1)==0 && !digitalRead(LT2)==0){
        //turn left
        goLeft();
      }
      //line detected by right sensor
      else if(!digitalRead(LT1)==0 && digitalRead(LT2)==0){
        //turn right
        goRight();
      }
      //line detected by none
      else if(!digitalRead(LT1)==0 && !digitalRead(LT2)==0){
        //stop
        stopRobot();
      }
      digitalWrite(LT_SIGNAL, LOW);
 }

 void objectAvoid(){
      digitalWrite(TRIG_SIGNAL, HIGH);
      delayMicroseconds(13);

      duration = pulseIn(ECHO, HIGH);
      distance = duration * 0.034 / 2;
      Serial.print(distance);

      if (distance <= 15){
        stopRobot();
        delay(500);
        goLeft();
        delay(500);
      }
```

```
      else {
        goAhead();
      }
      digitalWrite(TRIG_SIGNAL, LOW);
 }

 void loop()
{
  /* If the server available, run the "checkClient" function */
  server.handleClient();
  data = server.arg("State");

  if (data == "up") goAhead();
  else if (data == "down") goBack();
  else if (data == "left") goLeft();
  else if (data == "right") goRight();
  else if (data == "stop") stopRobot();

  else if (data == "linetracker") lineTracker();
  else if (data == "objectavoid") objectAvoid();
  else stopRobot();
}

void HTTP_handleRoot(void) {

if( server.hasArg("State") ){
       Serial.println(server.arg("State"));
  }
  server.send ( 200, "text/html", "" );
  delay(1);
}
```

ATMega16A

```
#define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <avr/sfr_defs.h>

void objectAvoid();
void lineTracker();

int main()
{
       DDRA = 0b11111100;
       DDRC = 0x00;
       DDRD = 0xFF;

       while (1) {
              if ((PINA & (0b00000001))) //  && !(PINA & (0b00000010))
              {
```

```
                        objectAvoid();
                }
                if ((PINA & (0b00000010))) // && !(PINA & (0b00000001))
                {
                        lineTracker();
                }
        }
}

void objectAvoid(){

        PORTD = 0b00000000;
        _delay_us(2);

        PORTD = 0b00000001;
        _delay_us(10);
        PORTD = 0b00000000;

}

void lineTracker(){
        //line detected by both
        if(!(PINC & (0b00000001)) && !(PINC & (0b00000010))){
                PORTD = 0b00000000;
        }
        //line detected by left sensor
        else if(!(PINC & (0b00000001)) && (PINC & (0b00000010))){
                //turn left
                PORTD = 0b00000010;
        }
        //line detected by right sensor
        else if((PINC & (0b00000001)) && !(PINC & (0b00000010))){
                //turn right
                PORTD = 0b00000100;
        }
        //line detected by none
        else if((PINC & (0b00000001)) && (PINC & (0b00000010))){
                PORTD = 0b00000110;
        }
}
```

Conclusion

In conclusion,

- In this project we learnt the basics of microprocessor systems and the skills of implementing different microprocessors at the hardware level to build something practical out of it.
- Moreover, we also learnt the basics of different hardware components and learnt their practical implementation.
- The project gave us an insight of how to bring into use the theoretical knowledge and use it along with the hardware.

Future Prospects:

- It developed our interest in this course.
- We learnt about the microprocessor industry and its vast scale.
- Gave us an intuition about the future courses regarding this domain and the research area of the field.
