Computer Interface Course Project Report
4th Year Computer Engineering

Project Title: [AC-1 Autonomous Car]

Team ID: [**2B**] [**AC-1**]

Team Members:

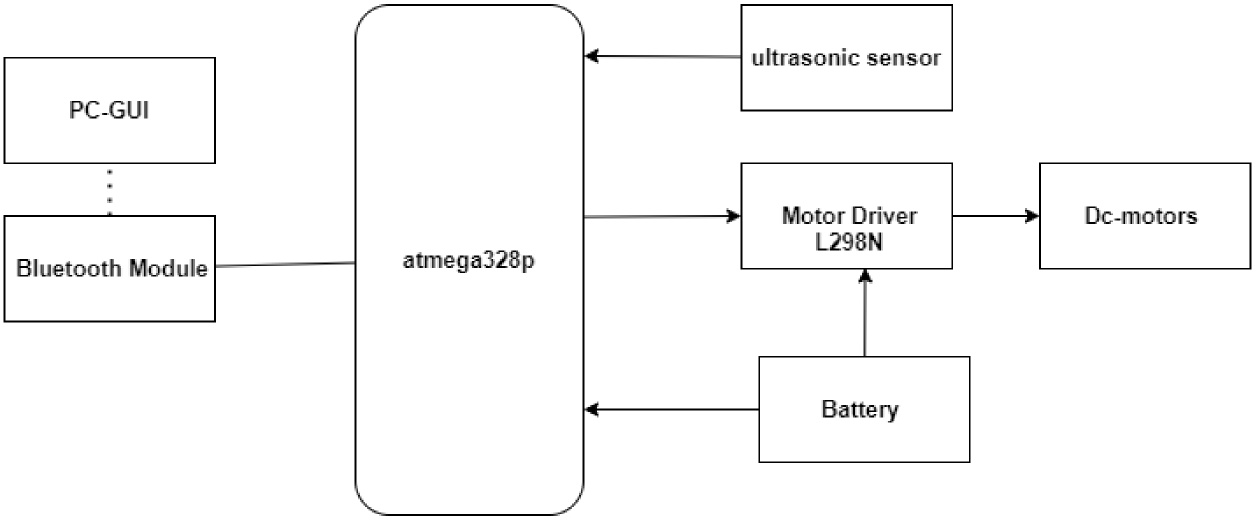
SN	Student Name	Section Number
1	Aliaa Mohamed Abdeldaym	2
2	Rana Ahmed Mahmoud	2
3	Rania Mostafa Ebrahiem	2
4	Sara Salah Ahmed	2
5		

1. Project Objective:

A simple car connected to a PC through Bluetooth module, having 2 modes of operation,
that the user can select through the GUI. First mode is manual mode, where the user can control car movements through the GUI using buttons in GUI. Second mode is autonomous mode, when selected through GUI by user, in which the car keeps moving while avoiding collisions using its on-board sensors (proximity sensors, e.g. ultrasonic).
In all modes, the car will send sensor readings to the PC to display it on the GUI, while showing a status of (Obstacle in range or No Obstacles) and the distance between the car and the available obstacle if it exists.

2. System Block Diagram:

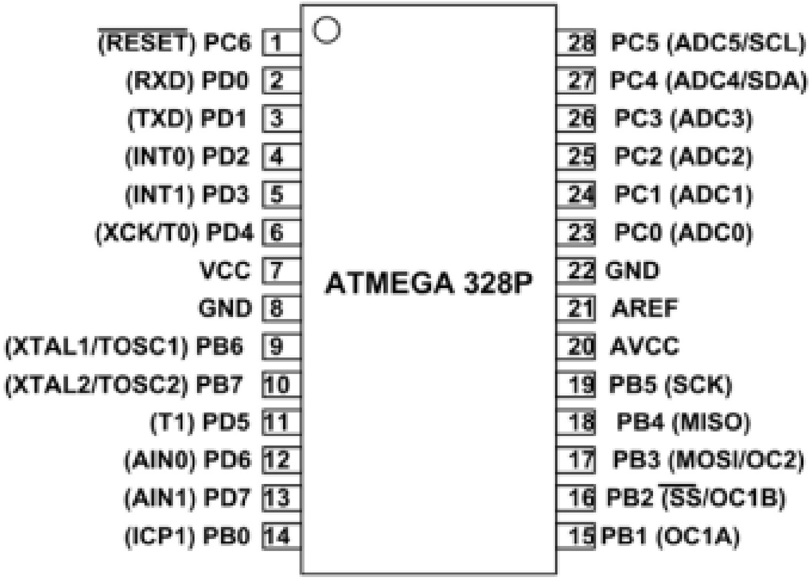
2.1. Block Diagram:



2.2. Block Diagram Description:

Atmega 328P Module:

ATmega328P is a very advance and feature-rich microcontroller. It is one of a famous microcontrollers of Atmel because of its use in Arduino UNO board. It is a microcontroller from Atmel's megaMVR microcontrollers family (Later in 2016 the Atmel is obtained by Microchip Technology Inc, the microcontrollers manufactured in megaMVR family are designed for handling larger program memories and each microcontroller in this family contains different amount of ROM, and RAM, I/O pins and other features and also, they are manufactured in different output pins which are from 8 pins to hundreds of pins. The internal circuitry of ATmega328P is designed with low current consumption features. The chip contains 32 kilobytes of internal flash memory, 1 kilobyte of EEPROM, and 2 kilobytes of SRAM. The EEPROM and the flash memory are the memories that save information and that information still exits the power is disconnected or off but the SRAM is a memory that only saves the information until the power is supplied and when the power is disconnected all the information saved in SRAM will be erased.

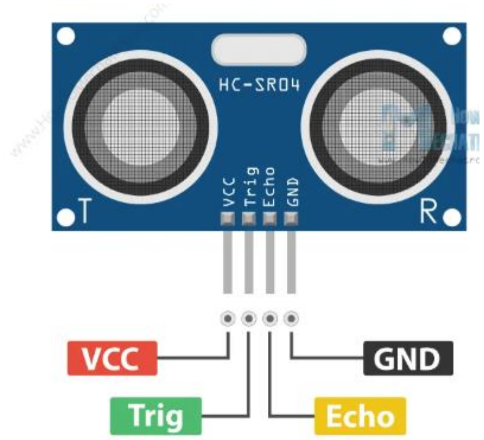


Ultrasonic Sensor:

The HC-SR04 is an affordable and easy-to-use distance-measuring sensor that has a range from 2cm to 400cm (about an inch to 13 feet).

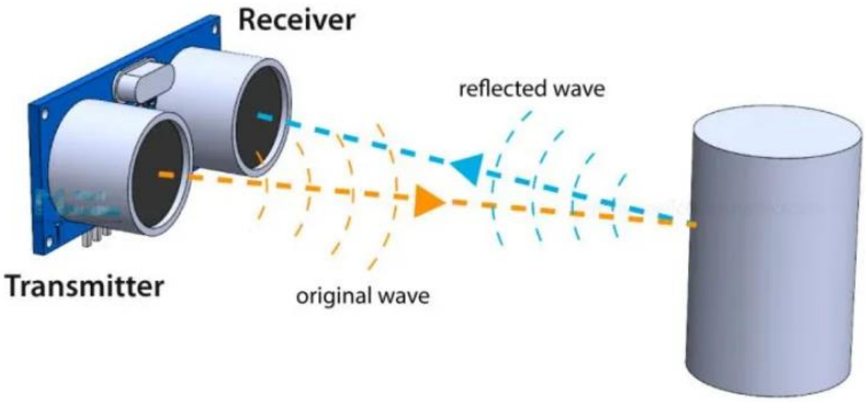
The sensor is composed of two ultrasonic transducers. One is a transmitter that outputs ultrasonic sound pulses and the other is a receiver that listens for reflected waves

The sensor has 4 pins. *VCC* and *GND* go to *5V* and *GND* pins on the Arduino, and the **Trig** and *Echo* go to any digital Arduino pin. Using the *Trig* pin we send the ultrasound wave from the transmitter, and with the *Echo* pin we listen for the reflected signal.

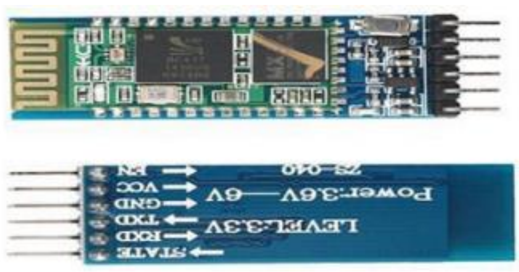


How the HC-SR04 Ultrasonic Distance Sensor Works?

It emits an ultrasound at 40 000 Hz which travels through the air and if there is an object or obstacle on its path It will bounce back to the module. Considering the travel time and the speed of the sound you can calculate the distance.

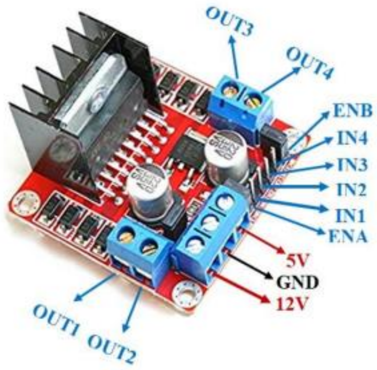


Bluetooth Module:
This component is a Serial Port Protocol Bluetooth module. It is mainly designed for serial Wi-Fi connection arrangement. It can provide switching between slave and master mode that means it can be used for transmitting or receiving Serial port data Bluetooth module which is a fully qualified Bluetooth V2.0+EDR 3Mbps Modulation having 2.4GHz frequency and base band. There are two work roles those are slave and master at the programmed connection work mode. This HC-05 is used to communicate between android phone or PC and microcontroller. This Bluetooth module is used to communicate by the help of USART at 9600 baud rates. This is very easy to interface with the microcontroller which can supports USAR



L298 Dual Channel H-BRIDGE:

Since the Motor Driver can handle only two motors, we connect two motors in parallel on each side. To power up the Motor Driver, we connect the power pins to Arduino UNO as Arduino is connected to a 12V battery. Enable pins of motor driver are connected to PWM pins of Arduino UNO



L298 Dual H-Bridge Motor Driver is ideal for robotic applications and well-suited for connection to a microcontroller just a couple of control lines per motor. It can also be interfaced with simple manual switches, TTL logic gates, relays, etc.

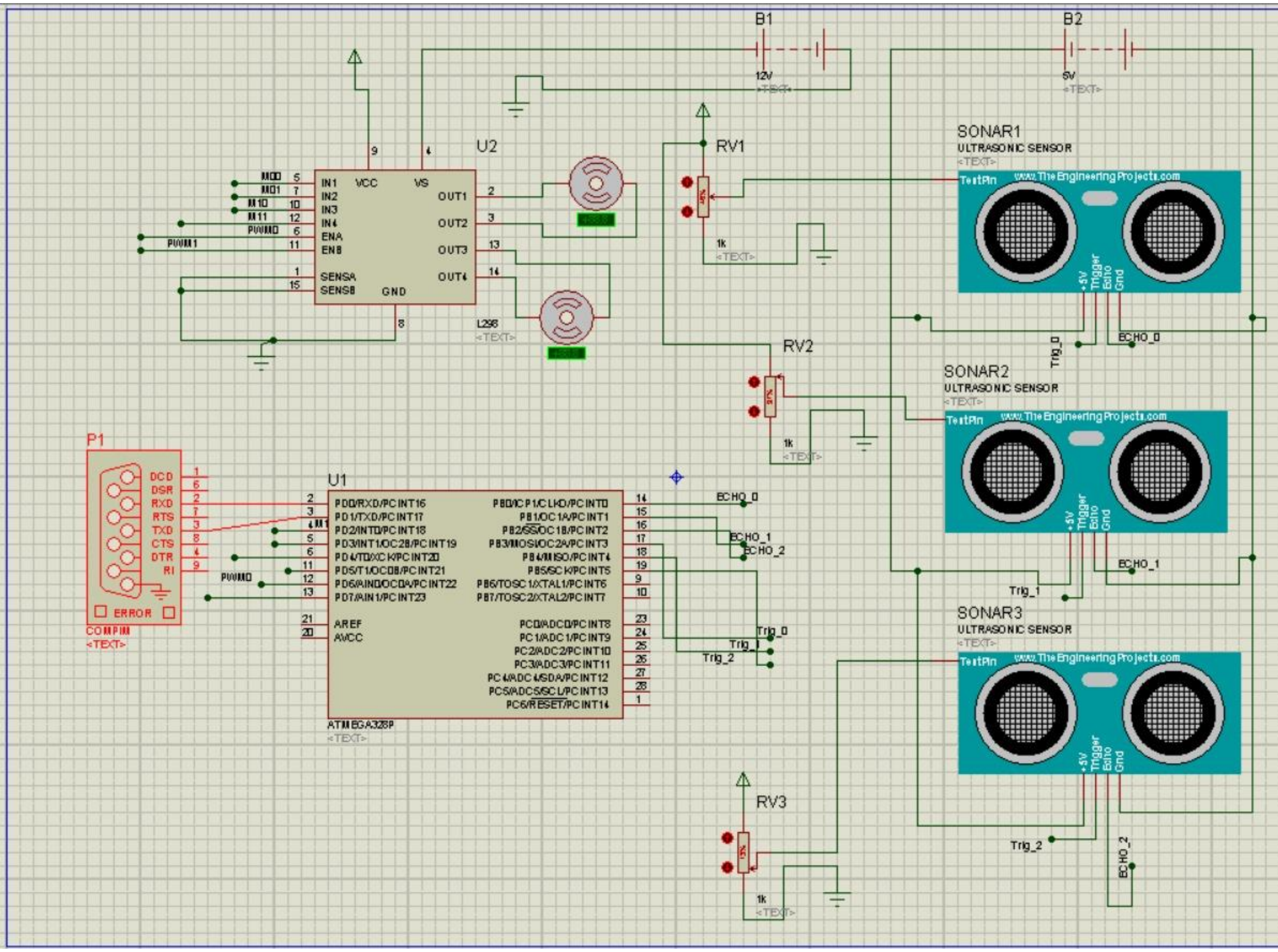
Dc-motor Wheels:

The 100 RPM Duel Shaft BO Motor gives good torque and rpm at lower operating voltages, which is the biggest advantage of these motors. It is an alternative to our metal gear DC motors. It comes with an operating voltage of 3-6V and is perfect for building small and medium robots.

A small shaft with matching wheels gives an optimized design for your application or robot. Mounting holes on the body & lightweight makes it suitable for in-circuit placement. The motor is ideal for DIY enthusiasts. This motor set is inexpensive, small, easy to install, and ideally suited for use in our robot car.

3. Schematic Diagram (Circuit Diagram):

[Provide here your full hardware schematic diagram]



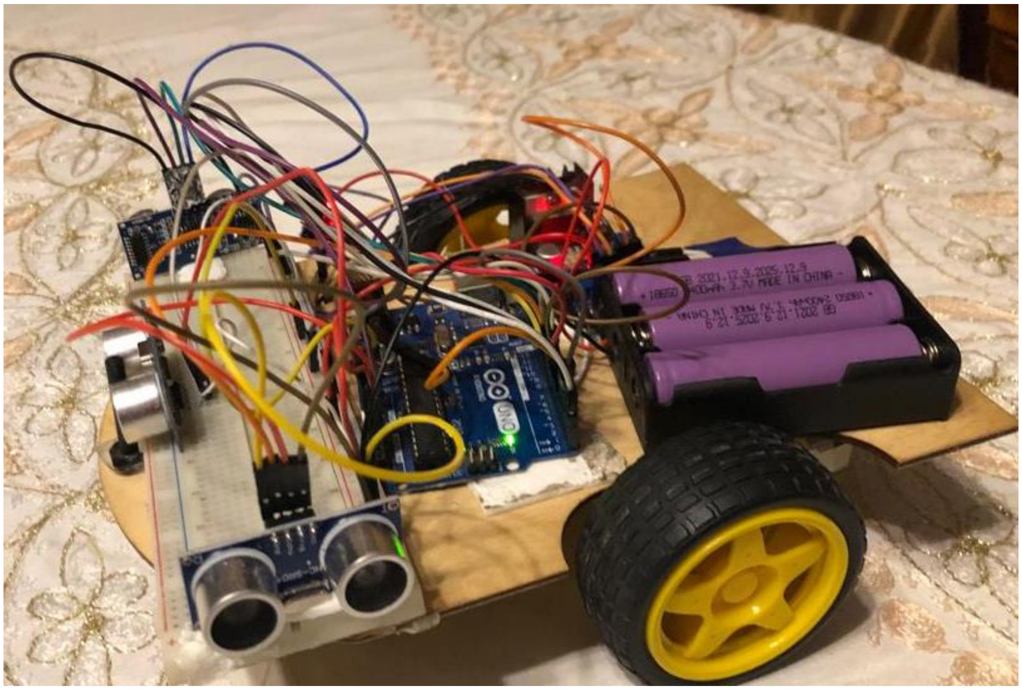
4. List Of Components:

in the project. For example, Item Type: Temperature sensor - Item Code Name: LM35]

SN	Item Type	Item Code Name	Purpose	Quantity
1	Microcontroller	Atmega 328P		1
2	Ultrasonic Sensor	HC-SR04	object or obstacle detect	3
3	Motor Driver	L298 Dual Channel H-BRIDGE:	control two motors	1
4	Bluetooth Module	HC-05	communicate between an android phone or PC and a microcontroller	1
5	Dc-motor Wheels	100RPM DC 3-12V Duel Shaft		2
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				

5. Real-Time Hardware Photo:

[Place a photo of your real-time hardware]



6. Source Code:

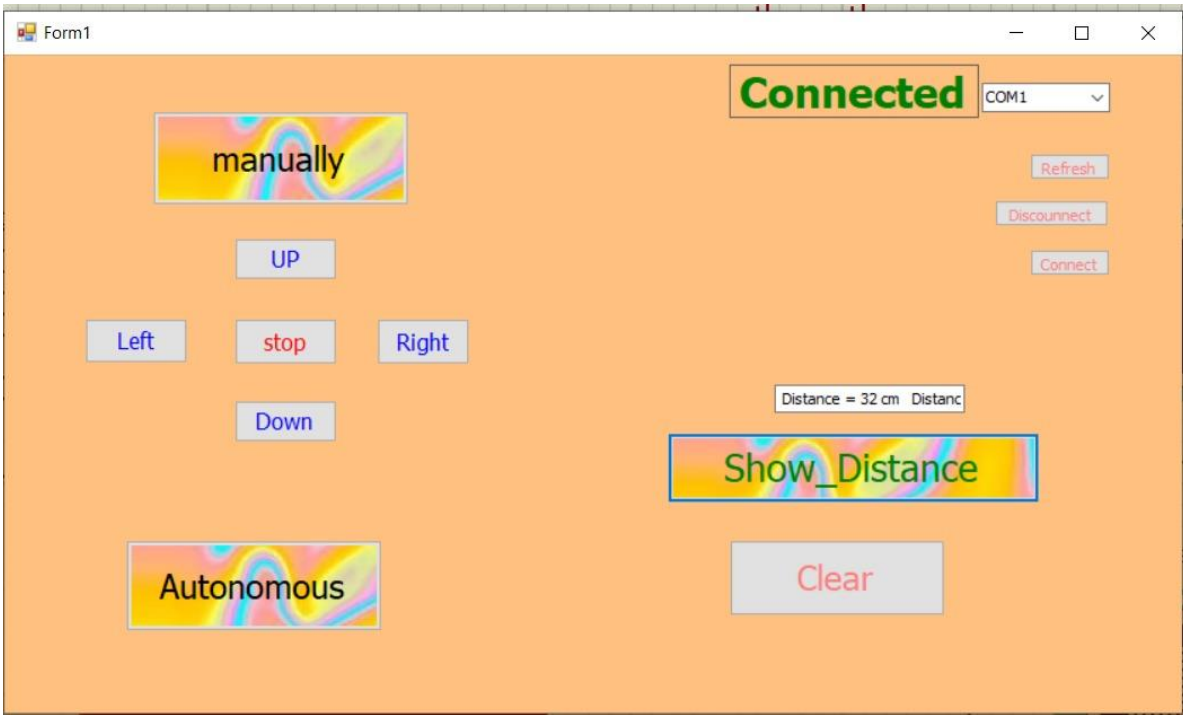
6.1. Hardware-side source code:

```
#define F_CPU 16000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include "Motors_Driver.h"
#include "UltraSonic_Driver.h"
#include "MACROS.h"
#include "Bluetooth.h"
#include "distance.h"
// static int R,L,M;
void AUTO_MODE();
int stop = 0 ;

int main(void)
{
USART_init();
Ultrasonic_init();
unsigned char _REC;
sei();
while (1)
{
        //continue; //TO BE DEleted
        _REC = USART_receive();

        switch(_REC)
                {
                        /*case 'A':
```

```
                        {
                                stop =1;
                                AUTO_MODE();
                                break;
                        }*/
                        case 'A':
                        {
                                        stop = 1;
                                        AUTO_MODE();

                                break;

                        }
                        case '2':
                        {
                                stop = 0;
                                Forward_M();
                                break;
                        }
                        case '4':
                        {
                                stop =0;
                                Backward_M();
                                break;
                        }
                        case '6':
                        {
                                stop =0;
                                Right_M();
                                break;
                        }
                        case '8':
                        {
                                stop =0;
                                Left_M();
                                break;
                        }
                        case'D':
                        {
                                calc_dis();
                                break;
                        }
                        default:
                        {
                                stop = 0;
                                Stop_M();
                        }

                }
}
        }


void AUTO_MODE()
{
        while(1 && stop){ //
        int R,L,M;
        M = ReadUS(_ULTRASONIC_MIDDLE_TRIGGER);
        L = ReadUS(_ULTRASONIC_LEFT_TRIGGER);
        R =ReadUS(_ULTRASONIC_RIGHT_TRIGGER);

        Forward_M();

        if (M<=20)
        {
                Backward_M();
                _delay_ms(500);
                Left_M();
                if (L<=20)
                {
                        Right_M();
                }
                _delay_ms(500);
                Forward_M();
        }
        else if (R<=15)
        {
                Left_M();
                _delay_ms(200);
                Forward_M();
        }
        else if(L<=15)
        {
                Right_M();
                _delay_ms(200);
                Forward_M();
        }
        else if(L<=20 && M<=20&& R<=20)
        {
                Backward_M();
                _delay_ms(2000);
                Right_M();
                _delay_ms(200);
                Forward_M();
        }

        }
}
int calc_dis()
{
        char numberString[10];
        uint16_t r =0;
        int distance =0;
        while(1){
                HCSR04Init();
                //_delay_ms(40);
                HCSR04Trigger();
                //_delay_ms(40);
                break;
        }
        r=GetPulseWidth();
        //distance=( r * 0.0344 / 2);
        //distance=((r/2.0)/150);
        distance = ReadUS(1);
        USART_putstring("Distance = ");
        itoa(distance, numberString,10);
        USART_putstring(numberString);
        USART_putstring(" cm");
        if (distance <= 15 ){
        USART_putstring("OBSTACLE!!!  ");}
        else {
                USART_putstring(" No obstacle     ");
        }
}

ISR(USART_RX_vect)
{

        switch(UDR0)
        {
                case 'M':
                {
                        stop = 0;
                        break;
                }
        }
}
```

6.2. PC-side source code:

[Provide your source code and images of the GUI application running on the PC side]



```
namespace WindowsFormsApp2
{
    public partial class Form1 : Form
    {
        SerialPort port = null;
        string distanse;
        string dataRX = " ";
        delegate void serialCalback(string val);
        public Form1()
        {
            InitializeComponent();
            refreshCom();
            label1.Text = "Disconnected";
            label1.ForeColor = Color.Red;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Connect();
        }

        private void button2_Click(object sender, EventArgs e)
        {
```

```
    Disconnect();
}

private void button3_Click(object sender, EventArgs e)
{
    refreshCom();
}
private void refreshCom()
{
    comboBox1.DataSource = SerialPort.GetPortNames();
}
private void Connect()
{
    port = new SerialPort(comboBox1.SelectedItem.ToString());
    port.BaudRate = 9600;
    port.DataBits = 8;
    port.StopBits = StopBits.One;
    try
    {
        if (!port.IsOpen)
        {
            port.Open();
            label1.Text = "Connected";
            label1.ForeColor = Color.Green;
                }
    }catch(Exception ex) {
        MessageBox.Show(ex.Message);
    }
    port.DataReceived += new System.IO.Ports.SerialDataReceivedEventHandler(DataReceived);


}
private void Disconnect()
{
    try
    {
        if (port.IsOpen)
        {
            port.Close();
            label1.Text = "Disconnected";
            label1.ForeColor = Color.Red;
        }
    }catch(Exception ex)
    {

    }
}

private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    Disconnect();
}



private void button8_Click(object sender, EventArgs e)
{

    try
    {
        port.Write("A");
    }
    catch (Exception ex) { }
}



private void button9_Click(object sender, EventArgs e)
{

    try
    {
        port.Write("M");
    }
    catch (Exception ex) { }

}

private void button10_Click(object sender, EventArgs e)
{
    try
    {
        port.Write("S");
    }
    catch
    (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void Form1_Load(object sender, EventArgs e)
{
    String[] ports = SerialPort.GetPortNames();

 }

private void up()
{
    try
    {
        port.Write("4");
    }
    catch
    (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
private void back()
{
    try
    {
        port.Write("2");
    }
    catch
    (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
```

```
}
private void right()
{
    try
    {

        port.Write("8");
    }
    catch
    (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
private void left()
{
    try
    {
        port.Write("6");
    }
    catch
    (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
private void button4_Click(object sender, EventArgs e)
{up();}

private void button7_Click(object sender, EventArgs e)
{left();}

private void button6_Click(object sender, EventArgs e)
{back();}

private void button5_Click(object sender, EventArgs e)
{right();}


private void button11_Click(object sender, EventArgs e)
{textBox1.Text = dataRX;}

private void button12_Click(object sender, EventArgs e)
{
    right();
}

private void DataReceived(object sender, SerialDataReceivedEventArgs e)
{ try{
        SerialPort sp = (SerialPort)sender;
        string temp = sp.ReadExisting();
        //setText(temp);
        dataRX += temp;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);

    }
}
```

```
    private void distance()
    {
        try{
            port.Write("D");
            // distanse = port.ReadExisting();
        }
        catch (Exception ex){
            MessageBox.Show(ex.Message);
        }
    }

    private void button13_Click(object sender, EventArgs e)
    {
        textBox1.Clear();
        dataRX = " ";
    }
}
}
```