

KOCAELI UNIVERSITY *FACULTY OF ENGINEERING

ADVANCED ROBOT CAR WITH C# PROGRAMMING

C# PROGRAMMING PROJECT

Tuana DALGIÇ

170208084

Department: Elektronik and Communication Engineering

Advisor: Dr. Oğuzhan Karahan

KOCAELİ, 2021

CONTENTS

1.ABSTRACT	3
2.INTRODUCTION.....	4
3.MATERIALS	5
3.1.Arduino UNO	5
3.2..Buzzer.....	6
3.3.Led.....	7
3.4..DC Motor.....	8
3.5.Ultrasonic Distance Sensor.....	9
3.6.DHT11 Temperature Sensor.....	9
3.7.Battery	10
3.8.Blutetooth Module	11
3.9.Car Kit	12
3.10.Rain Sensor.....	12
3.11.Motor Driver Board.....	13
3.12.Sound Sensor	13
4.CONCLUSION	14
4.1.CODES	15
4.1.1.C# Codes.....	15
4.1.2.Arduino Codes	28
5.REFERENCES.....	35

1.ABSTRACT

Machines are highly affects our lifes, nowadays it is the fourth and maximum dimension of it. Robot machines make our lifes comfortable and more easy .Our robot car performs multiple tasks when conditions are met. Advanced robot car checks the enviroment conditions via sensors. These sensors connected to Arduino-Uno Micro-processor which is “heart” for our car. We can say like that because everything perform on it . Arduino-Uno also connects to our PC with Bluetooth module . After reacting sensors , interface warns for options. Our robot car performs different actions , depending on the user selection.

INTRODUCTION

In this Project , we create various interactives with the help of sensors. After meeting our pre-determined conditions , car should performs something that user selects. This happens on interface , controlled with C# (some sort of remote control) . We load our codes through serial ports. Therefore , we load necessary libraries too .

Here is the list of interactives :

1)When Advanced Robot Car detects object , user should select one of the following options :

- Turn right

- Turn left

- Stop

2)Advanced Robot Car receives high sound , warns user. After that options given :

- Keep moving

- Ring the buzzer

3)Advanced Robot Car calculates the temperature of room. When it goes above 25 degrees warns user and gives the following options :

- Turn LED on

- Do nothing

4)When Advanced Robot Car contact with water, user gets warning and following options:

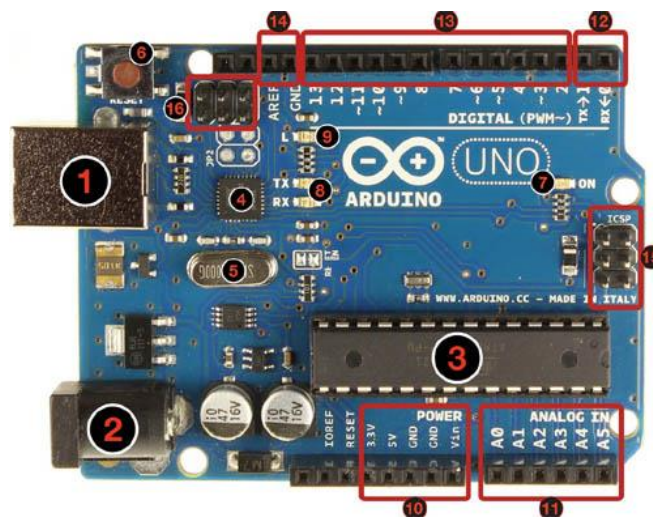
- Go away from water and give feed-back as “ There is no water contact anymore.”

- Stop right there.

3.MATERIALS

3.1.Arduino UNO

Arduino Uno has 14 digital input / output pins. 6 of these can be used as PWM outputs. There are also 6 analog inputs, one 16 MHz crystal oscillator, USB connection, power jack (2.1mm), ICSP header and reset button. Arduino Uno contains all the necessary components to support a microcontroller. You can power the Arduino Uno by connecting it to a computer, using an adapter or a battery. The picture below shows the parts of the Arduino UNO.



- 1: USB jack
- 2: Power jack (7-12 V DC)
- 3: Microcontroller ATmega328
- 4: Communication chip
- 5: 16 MHz crystal
- 6: Reset button
- 7: Power led
- 8: TX / NX leds
- 9: Led
- 10: Power pins
- 11: Analog inputs
- 12: TX / RX pins
- 13: Digital input / output pins (pins with ~ sign can be used as PWM outputs.)
- 14: Ground and AREF pins

15 : ICSP for ATmega328

16: ICSP for USB interface

3.2.Buzzer

Buzzer; It is a type of auditory warning device that works based on mechanical, electromechanical or piezoelectric principles. Buzzers can be used in functions such as alarm, timer, confirmation response alert, depending on their area of use. As a matter of fact, as we mentioned in the definition, buzzers are auditory warning device types. They have types such as illuminated buzzer, non-light buzzer, passive buzzer and active buzzer..



3.3.LED

LED is an abbreviation of Light Emitting Diode (Turkish for light emitting diode). As the name suggests, LED is a diode. As we know, a diode is a two-pin semiconductor circuit element that allows current to flow in only one direction.



3.4.DC Motor

DC motor is the machine that converts direct current electrical energy into mechanical energy. It is based on the principle of moving with the effect of the magnetic force generated in the opposite direction to the permanent magnets in the motor when electric current is applied to the windings inside the motor. It is necessary to change the direction of this current to continuously generate a reverse magnetic field to the permanent magnet.



3.5. Ultrasonic Distance Sensor

Ultrasonic Distance Sensors are sensors that detect the distance between the nearest object and between them by means of sound waves.



3.6. DHT11 Temperature Sensor

DHT11 temperature and humidity sensor is an advanced sensor unit that outputs a calibrated digital signal.



3.7.Battery

The battery is a self-contained, chemical power source that can generate a limited amount of electrical energy wherever it is needed. Unlike the electrical energy that is transmitted through cables from a power plant to your homes and needs to be consumed instantly, the battery transforms the chemicals stored in its structure into electrical energy that can be used for days, weeks and months.



3.8. Bluetooth Module

Bluetooth modules are released as HC series. Under this category, it takes place in the market with different variations with codes such as HC-03, 04, 05,06. However, HC-05 module and HC-06 module are generally used in our country. The biggest distinguishing feature of these two models is that the HC-05 module works actively in both Slave and Master modes. On the other hand, HC-06 only works in Slave mode. The Bluetooth module is interconnected with Arduino via TX / RX protocol. RX comes from the words "Receive", and TX comes from the words "Transmit". In short, it is a receiver / transmitter protocol. The important point in the connection is that the TX pin of Arduino must be connected to the RX pin of the Bluetooth module, and the Arduino's RX pin must be cross-connected to the TX pin of the Bluetooth module.



3.9.Car Kit



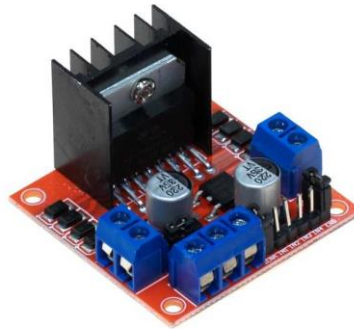
3.10.Rain Sensor

The rain sensor is a set of circuits with liquid sensing plates on it, which detects rain and warns with an alarm, programmed - audible or visual. The rain sensor, which can be prepared for various purposes such as using water efficiently or protecting water resources, is used in many different areas such as automobiles, telecommunications and agriculture. The working principle of the rain sensor is based on the resistance of the nickel plated plates on it. The sensor measures the humidity via analog output pins and gives digital output after exceeding the specified humidity threshold. In other words, when raindrops fall on the board, parallel resistance occurs and the sensor gives a warning. The sensor has a bipolar resistance and resists depending on moisture. The sensor shows more resistance when dry, but less resistance when wet.



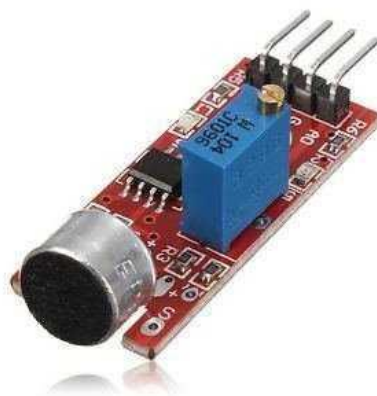
3.11. Motor Driver Board

This motor driver board, which is prepared to drive motors up to 24V, has two channels and supplies 2A current per channel. L298N motor driver IC is used on the board. It can be used in robots and a wide variety of motor control applications.



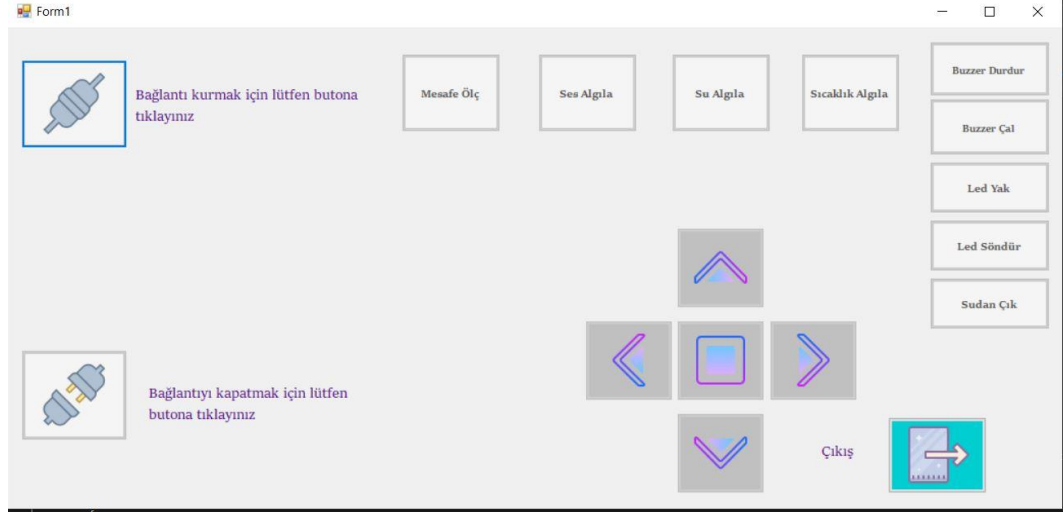
3.12. Sound Sensor

Sounder sensor card receive the sounds via microphone on it . It has digital input and outputs .

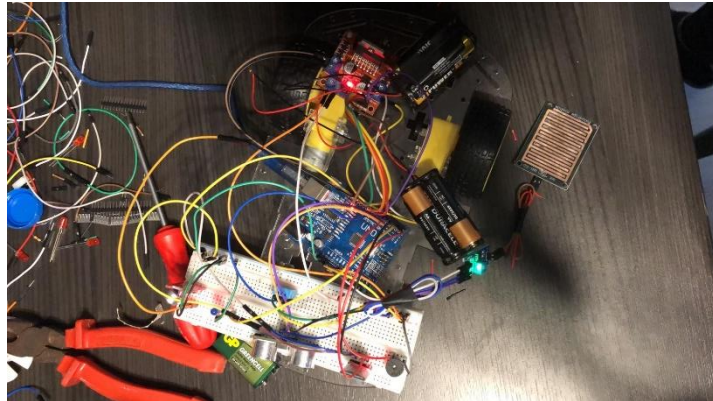


4.CONCLUSION

We got data through sensors. According to the data we received, we adjusted the threshold values and made measurements with sensors in this way. We have adjusted the timings and waiting times. The robot car performs all interactions with C# and Arduino codes we uploaded..



User Interface



4.1.CODES

4.1.1. C# Codes

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Globalization;
using System.IO.Ports;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace araba
```

```
{
    public partial class Form1 : Form
    {
```

```
        NumberStyles sayiStili;
```

```
        CultureInfo culture;
```

```
        SerialPort serialPort; //iletişim kurmak için SerialPort nesnemizi tanımladık
```

```
        List<Button> yonBuzzerLedSudanCikButonlari; //işlemlere göre butonlarımızın
        özellikleri ile oynadığımızdan bu işlemleri kolayca yapmak için button tipinde liste
        tanımladık
```

```
        List<Button> sensorButonlari;
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
            serialPort = new SerialPort();
```

```
            sayiStili = NumberStyles.Number | NumberStyles.AllowDecimalPoint;
```

```

        culture = CultureInfo.CreateSpecificCulture("en-GB");
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        serialPort.PortName = "COM9"; //port adımızı girdik
        yonBuzzerLedSudanCikButonlari = new List<Button>(); //listemizi
        oluşturduk
        sensorButonlari = new List<Button>();
        yonBuzzerLedSudanCikButonlari.Add(buzzerDurdurButton); //yön
        butonlarımızı ve buzzerDurdur butonumuzu listeye ekledik
        yonBuzzerLedSudanCikButonlari.Add(buzzerCalButton);
        yonBuzzerLedSudanCikButonlari.Add(sagaButton);
        yonBuzzerLedSudanCikButonlari.Add(solaButton);
        yonBuzzerLedSudanCikButonlari.Add(geriButton);
        yonBuzzerLedSudanCikButonlari.Add(sudanCikButton);
        yonBuzzerLedSudanCikButonlari.Add(ledYakButton);
        yonBuzzerLedSudanCikButonlari.Add(ledSondurButton);
        sensorButonlari.Add(mesafeOlcButton);
        sensorButonlari.Add(sesAlgilaButton);
        sensorButonlari.Add(yagmurAlgilaButton);
        sensorButonlari.Add(sicaklikAlgilaButton);
        baglantiKapatButton.Enabled = false; //uygulama ilk açıldığında
        baglantiKapat butonu pasif olacak şekilde ayarladık

    }

    private void baglanButton_Click(object sender, EventArgs e)
    {
        try
        {
            if (!serialPort.IsOpen) // eğer bağlantı kurulmamışsa
            {

```



```
serialPort.Open(); // bağlantıyı kuruyoruz
AraclariSifirla(); //Yön butonları,buzzer durdurma butonumuzu ve
labellarımızı pasifleştiriyoruz
```

```
baglantiLabel.Text = "Bağlantı kuruldu.Yapabileceğiniz işlemlerin ilgili
butonları aktifleştirildi."; //bağlantının kurulduğunu bildiriyoruz
```

```
baglantiKapatLabel.Text = "Bağlantıyı kapatmak için lütfen butona
tıklayınız"; //bağlantı kurulunca, baglantiKapat labelına baştaki gibi açıklama
bildirimini yazıyoruz
```

```
buttonDurumRenkAyari(baglanButton, Color.PaleGreen, false);
//bağlantı kurulunca bağlan butonumuz pasif oluyor,bağlan butonumuzun arkaplan
rengi yeşil oluyor
```

```
buttonDurumRenkAyari(baglantiKapatButton, Color.WhiteSmoke,
true);//bağlantı kapat butonumuz aktif oluyor,bağlantı kapat butonumuzun rengi
baştaki gibi gri oluyor
```

```
foreach (Button btn in sensorButonlari)
{
    btn.Enabled = true;
    btn.BackColor = Color.Gold;
}
buttonDurumRenkAyari(ileriButton, Color.Gold, true);
buttonDurumRenkAyari(durButton, Color.Gold, true);

}
}
catch
{
    baglantiLabel.Text = "Seri port hatası!";
}
```

```

    }

    private void baglantiKapatButton_Click(object sender, EventArgs e)
    {
        serialPort.Write("s"); //Bağlantıyı kapatmadan önce arabayı durduruyoruz
        serialPort.Close(); //Baglantiyi kapatıyoruz

        buttonDurumRenkAyari(baglantiKapatButton, Color.Tomato, false); //bağlantı
        kapat butonumuzu pasif yapıyoruz butonumuzun arkaplanını kırmızı tonunda
        yapıyoruz

        buttonDurumRenkAyari(baglanButton, Color.WhiteSmoke, true); //bağlan
        butonumuzu aktifleştiriyoruz bağlan butonumuzun arkaplan rengini başlangıçtaki gri
        renk yapıyoruz

        baglantiKapatLabel.Text = "Bağlantı kapatıldı."; //bağlantıyı kapattığımızı
        bildiriyoruz

        baglantiLabel.Text = "Bağlantı kurmak için lütfen butona tıklayınız";
        //bağlantı kapatılınca, baglanti labelına baştaki gibi açıklama bildirimini yazıyoruz
        AraclariSifirla();
        foreach (Button btn in sensorButonlari)
        {
            btn.Enabled = false; //sensör pasif yapıyoruz ve arkaplan renklerini
            değiştiriyoruz

            btn.BackColor = Color.Silver;
        }

        buttonDurumRenkAyari(ileriButton, Color.Silver, false);
        buttonDurumRenkAyari(durButton, Color.Silver, false);
    }

    private void buttonDurumRenkAyari(Button buton, Color renk, bool durum)
    {
        buton.Enabled = durum;
        buton.BackColor = renk;
    }

```

```
}
```

```
private void AraclariSifirla() //Her sensörde kullanıcıya farklı seçenekler  
sunduğumuzdan, her veri algıladığımızda labellerı saklıyoruz ve yön ve buzzer  
butonlarımızı pasifleştiriyoruz
```

```
{ //sonrasında istediğimiz araçları aktifleştiriyoruz  
    bildirimLabel.Text = "";  
    mesafeDegerLabel.Text = "";  
    yagmurDegerLabel.Text = "";  
    sicaklikDegerLabel.Text = "";  
    foreach (Button btn in yonBuzzerLedSudanCikButonlari)  
    {  
        btn.Enabled = false; //yön butonlarımızı ve buzzer butonumuzu pasif  
yapıyoruz ve arkaplan renklerini değiştiriyoruz  
        btn.BackColor = Color.Silver;  
    }  
}
```

```
mesafeOlcTimer.Stop();  
sesAlgilaTimer.Stop();  
yagmurAlgilaTimer.Stop();  
sicaklikAlgilaTimer.Stop();
```

```
}
```

```
private void ileriButton_Click(object sender, EventArgs e)  
{  
    serialPort.Write("w");  
}
```

```
private void sagaButton_Click(object sender, EventArgs e)  
{  
    serialPort.Write("d");
```

```
}
```

```
private void solaButton_Click(object sender, EventArgs e)
{
    serialPort.Write("a");
}
```

```
private void geriButton_Click(object sender, EventArgs e)
{
    serialPort.Write("x");
}
```

```
private void durButton_Click(object sender, EventArgs e)
{
    serialPort.Write("s");
}
```

```
private void buzzerDurdurButton_Click(object sender, EventArgs e)
{
    serialPort.Write("b");
}
```

```
private void buzzerCalButton_Click(object sender, EventArgs e)
{
    serialPort.Write("g");
}
```

```
private void mesafeOlcButton_Click(object sender, EventArgs e)
{
    serialPort.Write("m");
    AraclariSifirla();
    mesafeOlcTimer.Start();
}
```

```

private void mesafeOlcTimer_Tick(object sender, EventArgs e)
{

    double mesafe;
    string mesafeString = serialPort.ReadExisting();
    bool mesafeDurum = Double.TryParse(mesafeString, sayiStili, culture, out
mesafe);
    if (mesafeDurum)
    {

        mesafeDegerLabel.Text = "Mesafe: " + mesafe + " cm";
        if (mesafe <= 50 && mesafe != 0)
        {
            bildirimLabel.Text = "DİKKAT! Bir cisme 50 cm veya daha az mesafede
yakınsınız! Yapabileceğiniz işlemlerin ilgili butonları aktifleştirildi.";
            serialPort.Write("s");
            buttonDurumRenkAyari(sagaButton, Color.Gold, true);
            buttonDurumRenkAyari(solaButton, Color.Gold, true);
            mesafeOlcTimer.Stop();
        }
        else
        {
            serialPort.Write("m");
        }

    }
    else
    {
        serialPort.Write("m");
    }

}

```

```

private void sesAlgilaButton_Click(object sender, EventArgs e)
{
    serialPort.Write("v");
    AraclariSifirla();
    sesAlgilaTimer.Start();
}

private void sesAlgilaTimer_Tick(object sender, EventArgs e)
{
    int sesDeger;
    string sesString = serialPort.ReadExisting();
    bool sesDurum = Int32.TryParse(sesString, out sesDeger);
    if (sesDurum)
    {
        if (sesDeger==111)
        {
            bildirimLabel.Text = "Ses Algılandı! Yapabileceğiniz işlemlerin ilgili
butonları aktifleştirildi.";
            serialPort.Write("s");
            foreach (Button btn in yonBuzzerLedSudanCikButonlari)
            {
                btn.Enabled = true;
                btn.BackColor = Color.Gold;
            }
            sesAlgilaTimer.Stop();
        }
    }
}

```

```

private void yagmurAlgilaButton_Click(object sender, EventArgs e)
{
    serialPort.Write("r");
    AraclariSifirla();
    yagmurAlgilaTimer.Start();
}

private void yagmurAlgilaTimer_Tick(object sender, EventArgs e)
{
    int suDegeri;
    string suString = serialPort.ReadExisting();
    bool suDurum = Int32.TryParse(suString, out suDegeri);
    if (suDurum)
    {
        yagmurDegerLabel.Text = "Okunan su değeri: " + suDegeri;
        if (suDegeri < 400)
        {
            serialPort.Write("s");
            bildirimLabel.Text = "Su algılandı! Yapabileceğiniz işlemlerin ilgili
butonları aktifleştirildi.";
            buttonDurumRenkAyari(ileriButton, Color.Silver, false);
            buttonDurumRenkAyari(sudanCikButton, Color.Gold, true);
            yagmurAlgilaTimer.Stop();
        }
        else
        {
            bildirimLabel.Text = "Su ile temas edilmiyor.";
            serialPort.Write("r");
        }
    }
    else
    {

```

```

        serialPort.Write("r");
    }

}

private void sicaklikAlgilaButton_Click(object sender, EventArgs e)
{
    serialPort.Write("c");
    AraclariSifirla();
    sicaklikAlgilaTimer.Start();
}

private void sicaklikAlgilaTimer_Tick(object sender, EventArgs e)
{
    double sicaklikDegeri;
    string sicaklikString = serialPort.ReadExisting();
    bool sicaklikDurum = Double.TryParse(sicaklikString, sayiStili, culture, out
sicaklikDegeri);
    if (sicaklikDurum)
    {

        sicaklikDegerLabel.Text = "Sicaklik: " + sicaklikDegeri + "°C";
        if (sicaklikDegeri >= 25)
        {
            serialPort.Write("s");
            bildirimLabel.Text = "Sıcaklık 25 °C üstünde! Yapabileceğiniz işlemlerin
ilgili butonları aktifleştirildi.";
            buttonDurumRenkAyari(sagaButton, Color.Gold, true);
            buttonDurumRenkAyari(solaButton, Color.Gold, true);
            buttonDurumRenkAyari(geriButton, Color.Gold, true);
            buttonDurumRenkAyari(ledYakButton, Color.Gold, true);
            buttonDurumRenkAyari(ledSondurButton, Color.Gold, true);
            sicaklikAlgilaTimer.Stop();

```



```

    }
    else
    {
        serialPort.Write("c");
    }

}
else
{
    serialPort.Write("c");
}

}

private void cikisButton_Click(object sender, EventArgs e)
{
    if (serialPort.IsOpen)
    {
        serialPort.Write("s"); //Çıkmadan önce arabayı durduruyoruz
    }
    this.Close(); //Uygulamayı kapatıyoruz
}

private void sudanCik_Click(object sender, EventArgs e)
{
    serialPort.Write("w");
    AraclariSifirla();
    sudanCikTimer.Start();
}

private void sudanCikTimer_Tick(object sender, EventArgs e)
{
    int suDegeri;

```

```

string suString = serialPort.ReadExisting();
bool suDurum = Int32.TryParse(suString, out suDegeri);
if (suDurum)
{
    yagmurDegerLabel.Text = "Okunan su değeri: " + suDegeri;
    if (suDegeri > 400)
    {
        bildirimLabel.Text = "Su ile temas edilmiyor.";
        buttonDurumRenkAyari(ileriButton, Color.Gold, true);
        sudanCikTimer.Stop();

    }
    else
    {
        bildirimLabel.Text = "Hala su ile temas halindesiniz!";
        serialPort.Write("r");
    }

}
else
{
    serialPort.Write("r");
}
}

private void ledYakButton_Click(object sender, EventArgs e)
{
    serialPort.Write("y");
}

private void ledSondurButton_Click(object sender, EventArgs e)
{
    serialPort.Write("I");
}

```

}
}
}

4.1.2.Arduino Codes

```
#include <SoftwareSerial.h>
#include <dht11.h>
#define sagmotorhiz 11
#define solmotorhiz 6
#define sagmotoron 12
#define solmotoron 8
#define sagmotorarka 10
#define solmotorarka 9
SoftwareSerial bt(0,1);

const int trigPin = 4; //mesafe sensörü pin tanımlaması
const int echoPin = 3;
const int suSensorPin=A0; //yağmur sensörü pin ve eşik değeri tanımlaması
const int sesSensorPin=13;
dht11 DHT11; //sıcaklık sensörümüz için dht11 nesnemizi oluşturduk

const int buzzerPin = 5; //buzzer pin tanımlaması
const int ledPin = 2;
double uzaklik=0;
double sure=0;
int suDeger=0;

void setup(){
  DHT11.attach(7); // digital pinlerimizde 2 numaralı olanı dht11 sensörümüze
  bağıladığımızı belirtiyoruz
  bt.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(buzzerPin, OUTPUT);
  pinMode(ledPin, OUTPUT);
  pinMode(sesSensorPin,INPUT);
  pinMode(sagmotorhiz,OUTPUT);
```

```
pinMode(solmotorhiz,OUTPUT);
pinMode(sagmotoron,OUTPUT);
pinMode(sagmotorarka,OUTPUT);
pinMode(solmotoron,OUTPUT);
pinMode(solmotorarka,OUTPUT);

}
```

char okunanDeger; //formdan gönderilecek karakterlere göre işlem yapacağımızdan
char tipinde değişken tanımladık

```
void loop(){
if(bt.available()>0){
    okunanDeger=bt.read();
}
if(okunanDeger=='w'){ //eger ki bilgisayardan gönderilen deger 'w' ise araba ileri
gider.
    ileri();
}
if(okunanDeger=='a'){ //eger ki bilgisayardan gönderilen deger 'a' ise araba sola
gider.
    sola();
}
if(okunanDeger=='d'){ //eger ki bilgisayardan gönderilen deger 'd' ise araba saga
gider.
    saga();
}
if(okunanDeger=='x'){ //eger ki bilgisayardan gönderilen deger 'x' ise araba geri
gider.
    geri();
}
if(okunanDeger=='s'){ //eger ki bilgisayardan gönderilen deger 's' ise araba durur.
dur();
}
```

```

    }
    if(okunanDeger=='m')
    {
        mesafeHesapla();
    }
    if(okunanDeger=='v')
    {
        sesAlgila();

    }
    if(okunanDeger=='r')
    {
        suAlgila();
    }
    if(okunanDeger=='c')
    {
        sicaklikHesapla();
    }
    if(okunanDeger=='b')
    {
        buzzerDurdur();
    }
    if(okunanDeger=='g')
    {
        buzzerCal();
    }
    if(okunanDeger=='y')
    {
        ledYak();
    }
    if(okunanDeger=='I')
    {
        ledSondur();
    }

```

```

    }

    delay(500);
}

void mesafeHesapla(){

    digitalWrite(trigPin, LOW);
    delayMicroseconds(5);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    sure = pulseIn(echoPin, HIGH);
    uzaklik = sure /29.1/2;
    bt.println(uzaklik,2);

}

void sesAlgila(){
    while(true){
        int deger=digitalRead(sesSensorPin);
        if(deger==1){
            bt.print("111");
            break;
        }

    }

}

void suAlgila(){
    suDeger=analogRead(suSensorPin);
    bt.println(suDeger);
}

```

```

}

void sicaklikHesapla(){
    int chk = DHT11.read();
    bt.println((float)DHT11.temperature, 2);
}

void buzzerCal(){
    digitalWrite(buzzerPin, HIGH);
}

void buzzerDurdur()
{
    digitalWrite(buzzerPin, LOW);
}

void ledYak(){
    digitalWrite(ledPin, HIGH);
}

void ledSondur(){
    digitalWrite(ledPin, LOW);
}

void ileri()
{
    analogWrite(sagmotorhiz,100); //motorhız
    digitalWrite(sagmotoron,1); //motor ön
    digitalWrite(sagmotorarka,0); //arkaya dönmesine engel

    analogWrite(solmotorhiz,100); //motorhız
    digitalWrite(solmotoron,1); //motor ön
    digitalWrite(solmotorarka,0); //arkaya dönmesine engel

}

void geri()
{
    analogWrite(sagmotorhiz,100); //motorhız

```



```
digitalWrite(sagmotoron,0); //motor ön  
digitalWrite(sagmotorarka,1); //arkaya dönmesine engel
```

```
analogWrite(solmotorhiz,100); //motorhız  
digitalWrite(solmotoron,0); //motor ön  
digitalWrite(solmotorarka,1); //arkaya dönmesine engel
```

```
}  
void saga()  
{  
analogWrite(sagmotorhiz,100); //motorhız  
digitalWrite(sagmotoron,0); //motor ön  
digitalWrite(sagmotorarka,1); //arkaya dönmesine engel
```

```
analogWrite(solmotorhiz,100); //motorhız  
digitalWrite(solmotoron,1); //motor ön  
digitalWrite(solmotorarka,0); //arkaya dönmesine engel
```

```
}  
  
void sola()  
{  
analogWrite(sagmotorhiz,100); //motorhız  
digitalWrite(sagmotoron,1); //motor ön  
digitalWrite(sagmotorarka,0); //arkaya dönmesine engel
```

```
analogWrite(solmotorhiz,100); //motorhız  
digitalWrite(solmotoron,0); //motor ön  
digitalWrite(solmotorarka,1); //arkaya dönmesine engel
```

```
}  
void dur()  
{
```

```
analogWrite(sagmotorhiz,0); //motorhız  
digitalWrite(sagmotoron,0); //motor ön  
digitalWrite(sagmotorarka,0); //arkaya dönmesine engel
```

```
analogWrite(solmotorhiz,0); //motorhız  
digitalWrite(solmotoron,0); //motor ön  
digitalWrite(solmotorarka,0); //arkaya dönmesine engel
```

```
}
```

REFERENCES

[1] Kerem İzgöl,2015

(<https://maker.robotistan.com/arduino-dersleri-14-dc-motor-hiz-ve-yon-kontrolu/>)

[2] Kerem İzgöl,2015

(<https://maker.robotistan.com/arduino-ile-ses-sensor-modulu/>)

[3] (Arduino ile Ultrasonik Mesafe,2018)

(<https://hayaletveyap.com/arduino-ile-ultrasonik-mesafe-sensoru-hc-sr04-kullanimi/>)

[4] Alperen Kopuz,2019

(<https://maker.robotistan.com/arduino-c-sharp-bluetooth-servo/>)