



PARK SENSÖRÜ UYGULAMASI (PARKING SENSORS APPLICATION)

Mustafa GÜNEŞ*, İbrahim ÇAYIROĞLU**

*Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği, 78050, Karabük, mustafa.327@hotmail.com

**Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği, 78050, Karabük, icayiroglu@yahoo.com

Anahtar
Kelimeler:
Park Sensörü

Özet: Bu makalede Park Sensörünün genel bir tanıtımı yapıldıktan sonra uygulamalı iki örneği verilmiştir. Park Sensörü otomobillerin arka tamponuna yerleştirilen ultrasonik dedektörler aracılığı ile geri manevra sırasında araçla arkasındaki engel arasındaki mesafeyi ölçer. Bu ölçüm engeli yaklaştıkça frekansı yükselen bip sesleri ile sürücü tarafından algılanır. Engel ile araç arasındaki mesafe 30 cm'ye kadar düştüğünde bip sesi sürekli hale gelir. Park sensörüne sahip bir araç kullanırken, park sensörü yokmuş gibi arkayı kontrol etmeniz gerekir. Çünkü bu sensörler küçük veya ince çubuk yada direk şeklindeki engelleri algılayamaz. Ayrıca geri manevra sırasında aracın arkasına hareketli bir çocuk veya hayvan geçebileceğini de unutmamalıyız.

Keywords:
Parking Sensors

Abstract: In this article, given an practical example of Parking Sensors after two general introduction. Parking Sensor during reversing the car through the ultrasonic detector placed on the rear bumper measure the distance between obstacles behind the vehicle. These measurements closer to the obstacle is detected by the driver with rising frequency beeps. Means the distance between the obstacle and up to 30 cm fell beep becomes continuous. When using a car with parking sensors , parking sensors like there is no need to check back . Because these sensors can detect obstacles in the form of small or thin rod or pole. We should also not forget the back of a moving child or pet can pass the vehicle while reversing .

©2015 ibrahimcayiroglu.com, All rights reserved. Bu makale hakem kontrolünden geçmeden bilgi paylaşımı amacıyla yayınlanan bir dökümandır. Olabilecek hata ve yanlışlıklardan dolayı sorumluluk kabul edilmez. Makaledeki bilgiler referans gösterilip yayınlanabilir. {These articles are published documents for the purpose of information sharing without checked by the referee. Not accepted responsibility for errors or inaccuracies that may occur. The information in the article can be published by referred. }

1. GİRİŞ

Araç kullanmaya başlayanların en büyük sıkıntılarından biri, aracı kazasız belasız park etmektir. Park sensörleri işte tam bu noktada imdadımıza yetişerek kolaylık sağlayan bir teknolojidir. Yeni model birçok araç için standart haline gelen park sensörleri, eski model araçlar için montaj ve maliyet açısından kolaylıkla araçlara entegre edilebilir haldedirler. Park sensörü sayesinde çarpma korkusu olmadan rahat bir şekilde aracınızı park edebilirsiniz. Bu sensörler genellikle araçların arka kısmında olmasına rağmen bazı araçlarda hem arka hem de ön tarafta bulunabilir. Aracınız çarpmaya yaklaşık 1 metre kala park sensörü sizi modeline göre sesli ve görsel olarak uarmaya başlar. Park sensörleri hem ses dalgaları üretebilen hem de ürettikleri bu ses dalgalarını alabilen aygıtlardır. Üretilen ses dalgaları boşluğa gönderilir. Otomobil çevresinde bir nesne varsa bu ses dalgaları nesneye çarparak geri döner. Geri dönen bu ses bir yankı olarak düşünülebilir. Otomobildeki alıcılar geri dönen bu ses dalgalarını algırlar. Sensör, ses dalgalarının ne kadar sürede geri döndüğünü hesaplayarak, nesnenin

otomobilden ne kadar uzakta olduğu belirlemiş olur. Ses dalgalarının geri dönüş süresine bağlı olarak sistem bir uyarı sinyali üretmeye başlar. Nesnenin uzaklığına bağlı olarak üretilen bu uyarı sinyali “Dıt, dıt” şeklinde olur. Otomobil nesneye yaklaştıkça sinyalin sıklığı artarak düz bir sese dönüşür. Modeline göre bazı sensörler sürücüye otomobilin nesneye uzaklığının ölçüsünü de metre cinsinden gösterir veya sözlü olarak ifade eder. Aşağıdaki yapılmış park sensörü uygulamasında kullanıcıya hem sesli hemde görsel olarak uyarı vermektedir.

2. MALZEMELERİN TANITIMI

2.1. Gerekli Malzemeler

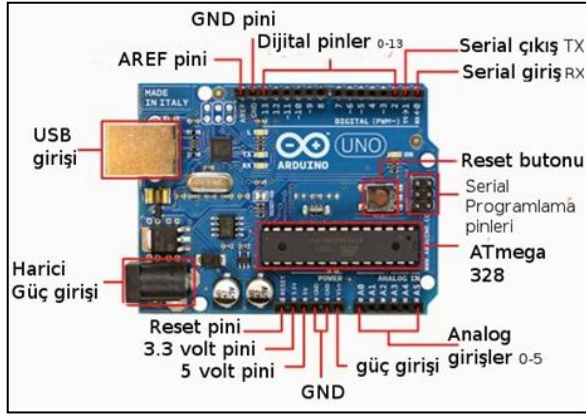
1. Arduino Uno (www.gittigidiyor.com)
2. Ultrasonik Mesafe Sensörü (www.gittigidiyor.com)
3. Breadboard(www.gittigidiyor.com)
4. Jumper Kablolar(www.robotistan.com)
5. 5.2 Adet 56Ω Direnç(www.gittigidiyor.com)
6. Buzzer , Yeşil ve Kırmızı Led(www.robotistan.com)
7. Lcd Ekran (16x2)(www.robotistan.com)

2.2. Arduino Uno

Arduino Uno normal USB girişine ve adaptör girişine sahiptir. USB girişi sayesinde bilgisayara bağlayıp kolay bir şekilde kod atabilir yada bilgisayar ile haberleşmesini sağlarız. Adaptör girişi ise adaptör yada pil sayesinde arduino için gereken gücü almamızı sağlar ancak usb ile bilgisayara bağliırken yada başka kaynaktan güç alırken bu

girişi kullanmak zorunda değiliz. Arduino Uno 3 adet led barındırır bunlardan ikisi RX ve TX ledleridir ve seri haberleşme akışı sırasında çalışırlar diğer led ise 13. pine bağlıdır bu sayede led uyarısı istediğimiz durumlarda harici led bağlamadan bu ledi kullanabiliriz.

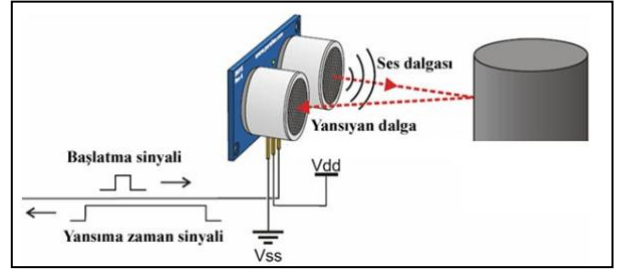
Arduino Uno barındırdığı ATmega 328 sayesinde 32 Kb lık bir hafızaya sahiptir ancak bunun 0.5 Kb lık kısmı Arduino bootloaderi tarafından kullanılmaktadır. Ayrıca Arduino Uno 2 Kb RAM ve 1 Kb EEPROM barındırır.



Şekil 1. Arduino Uno

2.3. HC-SR04 Ultrasonik Sensör

Ses dalgaları sınıflandırılmasında 20Khz-1Ghz aralığındaki ses sinyalleri ultrasonik ses olarak tanımlanmıştır. Bizim sensörümüz ve bir çok ultrasonik sensör 40Khz frekansında ultrasonik ses üretmektedir. Burada sesin yüksekliğinde belirleyici olan etken frekanstır. Ses yüksekse frekansta yüksektir. Ultrasonik ses sinyallerini insan kulağı algılayamaz. Ultrasonik sensörlerin yapısını incelediğimizde, bir tane özel hoparlör ve bir tane özel mikrofön bulunmaktadır. Özel hoparlör sayesinde 40kHz seviyesinde ses dalgaları üretilmektedir. Özel mikrofön sayesinde de bu ses dalgaları algılanmaktadır. Sensörü kullanmak için trig pininden yaklaşık 10us'lık bir pals gönderiyoruz. Bu değeri datasheetten biliyoruz. Sensör kendi içerisinde 40kHz frekansında bir sinyal üretilip 8 pals verici transdüserine gönderiyor. Bu ses dalgası havada, deniz seviyesinde ve 15 °C sıcaklıkta 340 m/s bir hızla yol alır. Bir cisme çarpar ve geri sensöre yansır. Cismin sensörden uzaklığı ile doğru orantılı olarak echo pini bir süre lojik-1 seviyesinde kalır ve tekrar lojik- 0 olur. Bizim bu mesafeyi ölçmek için tek yapmamız gereken echo pininin ne kadar lojik-1 olduğunun süresini bulmaktır.



Şekil 2. HC-SR04 Ultrasonik Sensor

2.4. Buzzer

Elektronik cihazlarda üzerine gelen voltajın büyüklüğü ile orantılı "bip" sesi çıkaran minik hoparlörlerdir. Devreli ve devresiz buzzer olmak üzere iki çeşidi var ve devreli olanlardan daha güçlü bir ses çıkıyor. Arabaların gösterge panelinde bulunan, 2 tonlu sese sahip parça buzzerdir. Işıklar açıkken kapı açıldığında ya da bildiğimiz sinyal verildiğinde çıkan ses bu parçadan gelmektedir. Yine multimetre, park sensörleri, doğal gaz kaçak alarm cihazları, hırsız alarm devreleri dahil bir çok elektronik parçada buzzer kullanılmaktadır.

2.5. Lcd Ekran (16x2)

VEE= Lcd ekranın parlaklık ayarının yapılmasında kullanılır.

RS= Lcd ye komutmu yoksa verimi gönderileceğini belirlemede kullanılır.

RW=Lcd ye okumamı yoksa yazma mı yapılacağını belirler.

EN(enable)=Etkinleştirme ucudur. Lcd ve pinler arasında gerçek veri akışını sağlar.

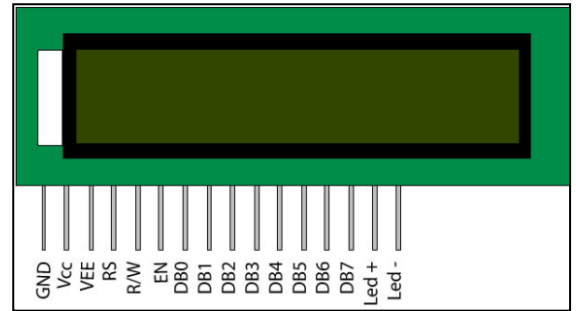
DB0-DB7=Data pinleridir. Bu pinlerle 4 veya 8 bitlik veri gönderilebilir.

Led+= Lcd panel ışığının (+) ucudur.

Led -= Lcd panel ışığının (-) ucudur.

Gnd = Toprak bağlantı ucudur.

Vcc =5V luk gerilim ucudur.



Şekil 3. Lcd Ekran ve Pinleri

3. DONANIMIN KURULMASI

3.1. Arduino' nun Pc'ye Tanıtılması

Arduino kartımızı bilgisayarımıza tanıtmak için ilk önce "Arduino Resmi Sitesi"nden Arduino IDE programını indirip bilgisayarımıza kurmamız gerekiyor. Arduino UNO için basit bir yazıcı kablosuyla, Leonardo için Micro USB kablosuyla doğrudan bilgisayarımızın USB portuna bağlıyoruz. Bilgisayarım(Sağ tık) -> Özellikler -> Aygıt Yöneticisi(Win XP'de Donanım sekmesinde, Win

7'de sol tarafta) -> Açılan Listedan Arduino UNO'ya çift tıklayın. Ardından:Sürücüyü güncelleştir->Belirtilen konumu dahil et(isim değişik olabilir, son seçeneği seçiyoruz) -> Sihirbazın sıradaki ekranında C:\... şeklinde bir yer olacak. Gözet butonuna tıklayıp Arduino klasörü -> Drivers klasörünü seçip İleri'ye basıyoruz. Artık sürücülerimiz de yüklenmiş oluyor.

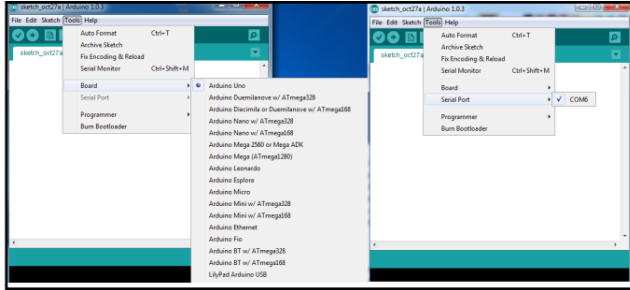
<https://www.arduino.cc/en/Main/Software>

3.2. Arduino Uno Haberleşmesi ve Bağlantının Kurulması

Arduino Uno sahip olduğu ATmega 328 sayesinde 0.(RX) ve 1.(TX) pinleri sayesinde seri haberleşme yapabilmektedir, ayrıca sahip olduğu USB bağlantı sayesinde doğrudan USB bağlantısında bulunabilir. Bu bağlantılar ARduino Uno kartımıza kod atmak dışında sensörlerden ölçtüğümüz verileri Arduino IDE içinde bulunan "serial monitor" sayesinde görebilirken yine bu araç sayesinde Arduino Unoya anlık komutlar verebiliriz.

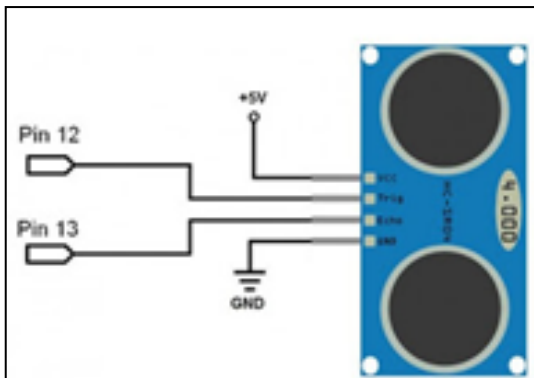
3.3. Arduino Programının Kullanımı ve Programın Yüklemesi

Arduino'ya program atmak için ilk olarak bilgisayarımıza bağladığımız Arduino'yu ve bağlantı portunu Arduino program menüsünden seçmemiz gerekiyor. Bunun için sırayla şu adımları izleyebiliriz. Hemen ardından, Tools menüsü altında Serial Port seçeneğine gelip bağladığımız portun yanında tik işareti yoksa tıklayıp işaretliyoruz. İkinci aşama olarak, kullandığımız Arduino modelini tanıtıyoruz. İlk önce aşağıdaki ekranı görmeliyiz.

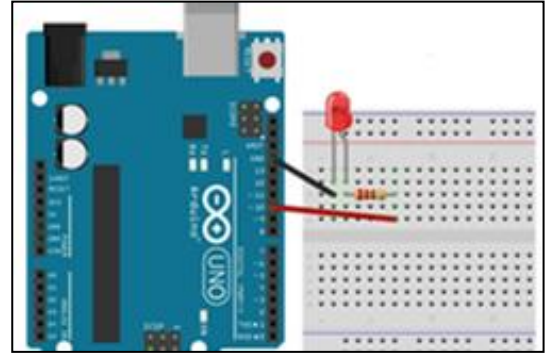


Şekil 4. Arduino'nun Pc'ye Tanıtımı

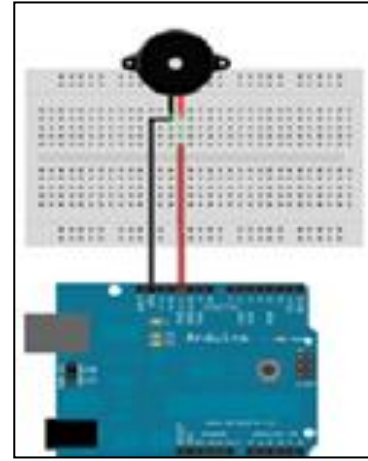
3.4. Devre Bağlantıları



Şekil 5. Sensör Bağlantısı



Şekil 6. Led Bağlantısı

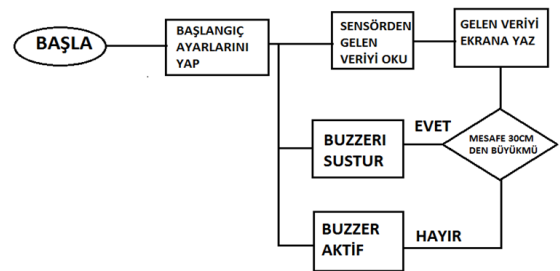


Şekil 7. Buzzer Bağlantısı

4. PROGRAMLAMA

4.1. Uygulama 1: Bilgisayar üzerinden sensörün kullanılması

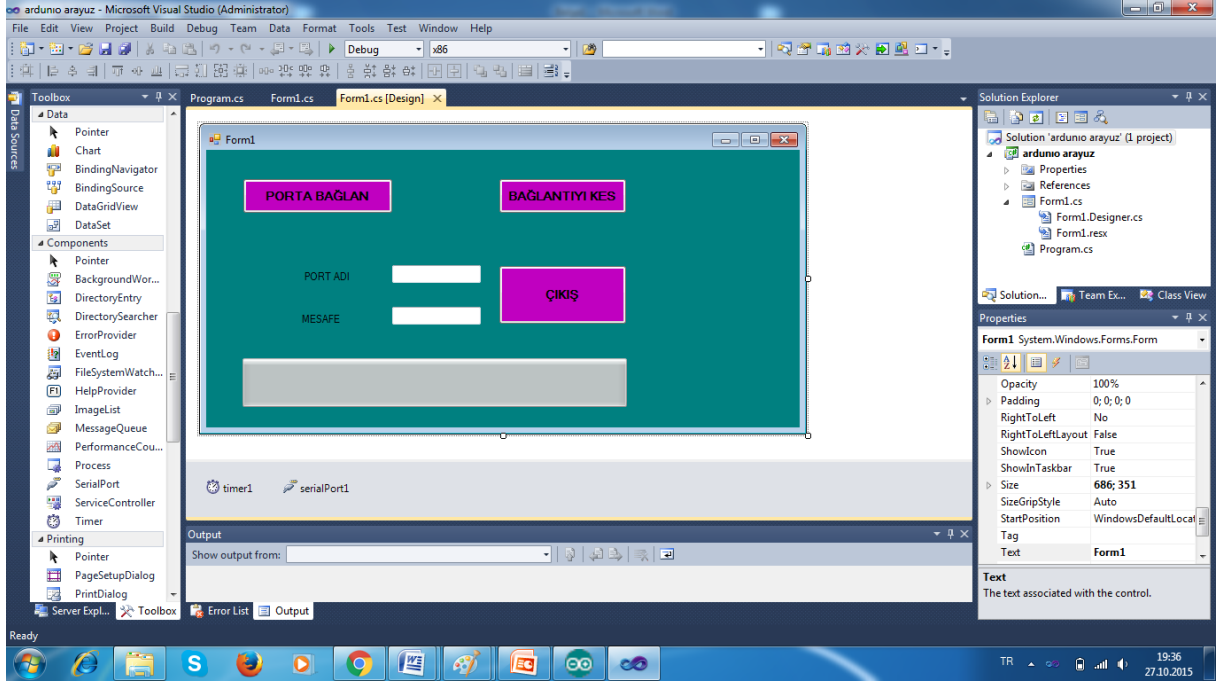
Programlamada kullanılacak olan akış diyagramı aşağıdaki gibidir.



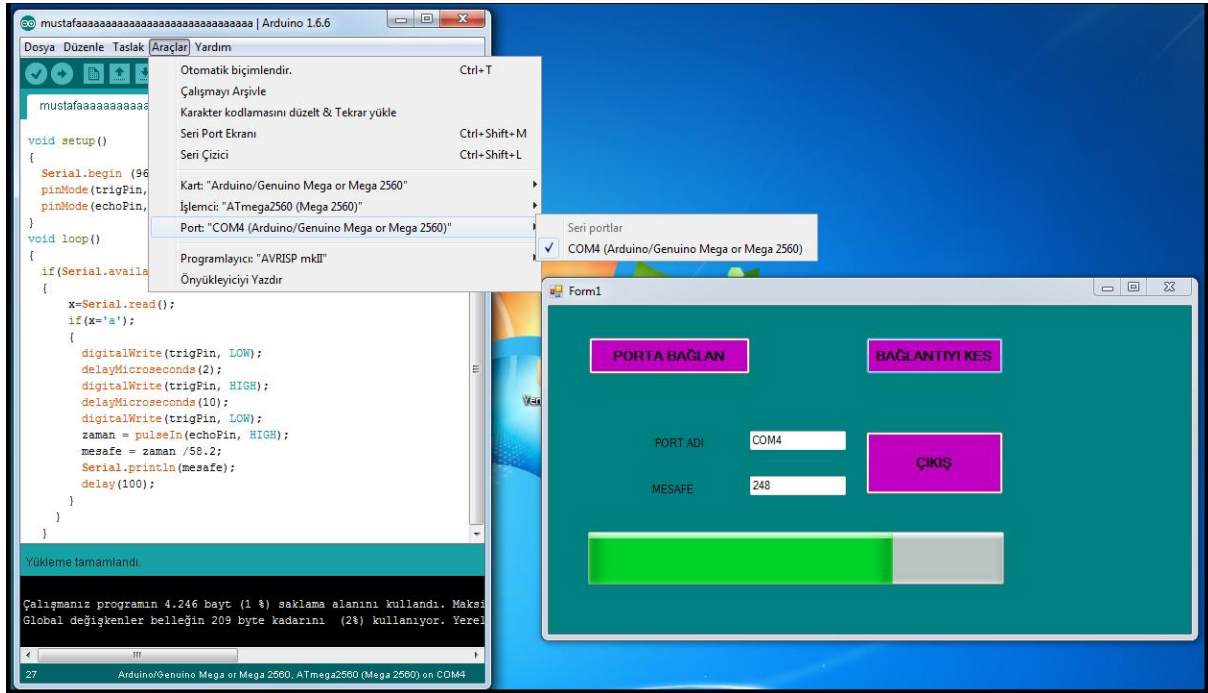
Şekil 8. Programın algoritması

4.2. Visual Studio ile Yazılım Arayüzünün Geliştirilmesi

Arduino ile visual studio haberleşmesi sağlanarak aşağıdaki gibi bir arayüz programı hazırlanmıştır. Bu uygulamada sadece arduino ve ultrasonik sensör kullanılmıştır. Sensör üzerindeki echo pini arduino üzerindeki 12 nolu dijital çıkış pinine, trig pini ise 13 nolu dijital çıkışa bağlanmıştır.



Şekil 9. Visual Studio İle Hazırlanan Arayüz



Şekil 10. Çalışan Uygulaması

4.3. Visual Studio Kodları

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

```
using System.Media;
namespace arduino_arayuz
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            serialPort1 = new SerialPort();
            serialPort1.BaudRate = 9600;
        }
    }
}
```



```

private void Form1_Load(object sender,
EventArgs e)
{
    textBox1.Text = "COM4";
}

private void button1_Click(object sender,
EventArgs e)
{
    timer1.Start();
    try
    {
        serialPort1.PortName = textBox1.Text;
        if (!serialPort1.IsOpen)
            serialPort1.Open();
        MessageBox.Show("bağlandı");
    }
    catch
    {
        MessageBox.Show("bağlantı hatası");
    }
}

private void timer1_Tick(object sender,
EventArgs e)
{
    try
    {
        serialPort1.Write("a");
        int alınanveri =
Convert.ToInt16(serialPort1.ReadExisting());
        textBox2.Text = alınanveri.ToString();
        progressBar1.Value = alınanveri;
        System.Threading.Thread.Sleep(100);
    }
    catch (Exception ex)
    {
        int deger =
Convert.ToInt16(textBox2.Text);
        if (deger <= 20)
        {
            SoundPlayer müzik= new
SoundPlayer();
            string path =
"C:\\Users\\Hp\\Desktop\\Klasik-eski-telefon-
sesi.wav";
            müzik.SoundLocation = path;
            müzik.Play();
        }
    }
}

private void button2_Click(object sender,
EventArgs e)
{
    timer1.Stop();

```

```

serialPort1.Close();
}

```

```

private void button3_Click(object sender,
EventArgs e)
{
    this.Close();
}
}
}

```

4.4. Arduino Kodları

long sure, mesafe; //mesafe hesabında
kullanılacak olan değişkenler

char x; //seri iletişimde kullanılacak
olan değişken

int val=0;

#define echoPin 12 //sensör üzerindeki echo
pininin arduino üzerindeki bağlandığı pin numarası

#define trigPin 13 //sensör üzerindeki trig pininin
arduino üzerindeki bağlandığı pin numarası

void setup()

{

Serial.begin (9600); //seri iletişimi başlat

pinMode(trigPin, OUTPUT); //trig pinini
çıkış olarak ayarla

pinMode(echoPin, INPUT); //echo pinini
giriş olarak ayarla

}
void loop()

{
if(Serial.available()>0) //visual studio ile
seri iletişimi başlatma şartı

{
x=Serial.read();
if(x='a');

{
digitalWrite(trigPin, LOW); //trig pinini
2 mikrosaniye kapat

delayMicroseconds(2);
digitalWrite(trigPin, HIGH); //trig pinini
10 mikrosaniye aç

delayMicroseconds(10);
digitalWrite(trigPin, LOW); //trig pini
kapat

sure = pulseIn(echoPin, HIGH); //süre yi
echo pinden al

mesafe = sure/58.2; //
mesafeyi cm olarak hesapla

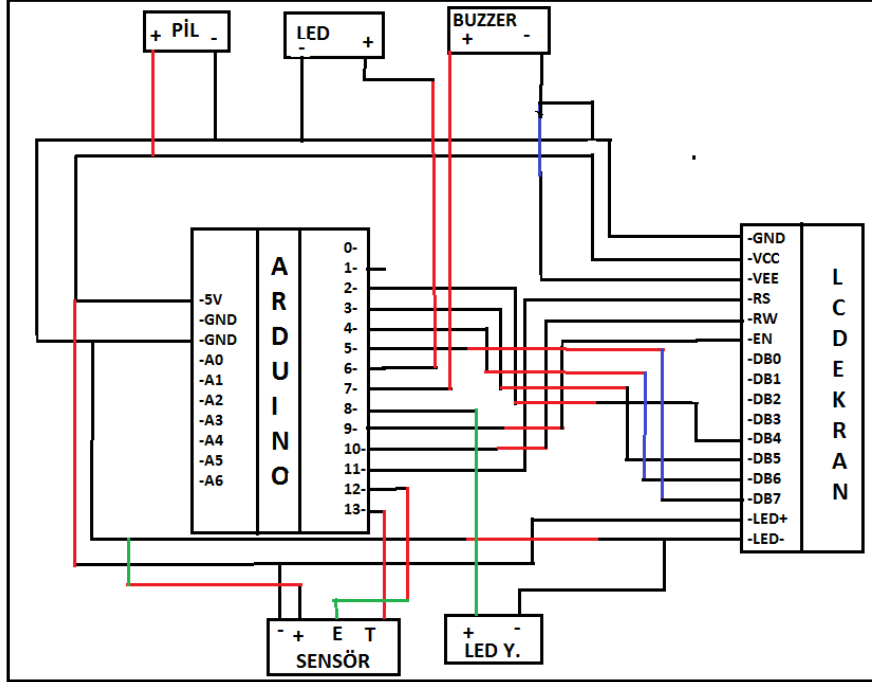
Serial.println(mesafe); // serial
monitor ekranına mesafeyi yaz

delay(100);
}
}
}

4.5. Uygulama 2: Bilgisayar olmadan sensörün kullanılması

Bu uygulamada yukardaki bahsi geçen tüm malzemeler kullanılmıştır. Bilgisayarımızdan yazmış olduğumuz arduino kodlarını yükledikten sonra bilgisayara gerek olmadan bir pil yada adaptör takarak aracımıza monte edebiliriz. Burada buzzerın (-) pini ile lcd ekranın üçüncü pinini

birleştirip board üzerinden (-) ye bağlamaya dikkat etmeliyiz. Aksi takdirde buzzerdan ses çıkışı alınamayacaktır. Aşağıdaki arduino programından mesafe değerlerini değiştirip istediğimiz mesafe aralıklarında bize uyarı vermesini sağlayabiliriz. Lcd olmadan uygulamayı yapabilmek için arduino programındaki lcd ekrana ait olan tüm komutlar silinip lcd siz uygulaması da yapılabilir. Uygulamanın devre şeması aşağıdaki gibi olmalıdır.



Şekil 11. Devre Şeması

4.6. Arduino Kodları

```
#define trigPin 13 // Trigger Pini (Ses gönderme)
#define echoPin 12 // Echo Pini (Yansıyan ses algılama)
#include <LiquidCrystal.h>
LiquidCrystal lcd(11,10,9,2,3,4,5); //lcd nin
bağlandığı pinler
int led1 = 6; //kırmızı ledin bağlandığı pin
int bzt =7; //buzer pini
int led2=8; //yeşil ledin bağlandığı pin
long microSaniyeSure, cmMesafe; // Mesafe
hesabında kullanılan değişkenler
int ton =7;
//-----//

void setup() {
  Serial.begin(9600); // Seri iletişimi başlat
  pinMode(trigPin, OUTPUT); // Ses gönderme pin
  türünü çıkış olarak ayarla
  pinMode(echoPin, INPUT); // Yansıyan ses
  algılama pin türünü giriş olarak ayarla
  lcd.begin(16, 2); // lcd ayarla
```

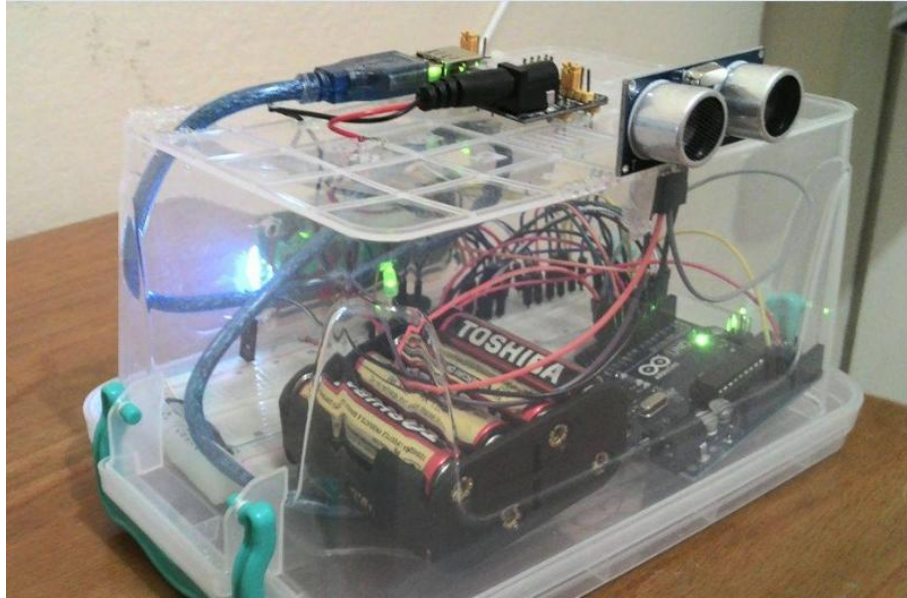
```
pinMode(led1,OUTPUT); //kırmızı ledi çıkış olarak
ayarla
pinMode(led2,OUTPUT); //yeşil ledi çıkış olarak
ayarla
pinMode(bzt, OUTPUT); //buzzerı çıkış olarak
ayarla
pinMode(ton, OUTPUT); //sesi çıkış olarak ayarla
}
//-----//

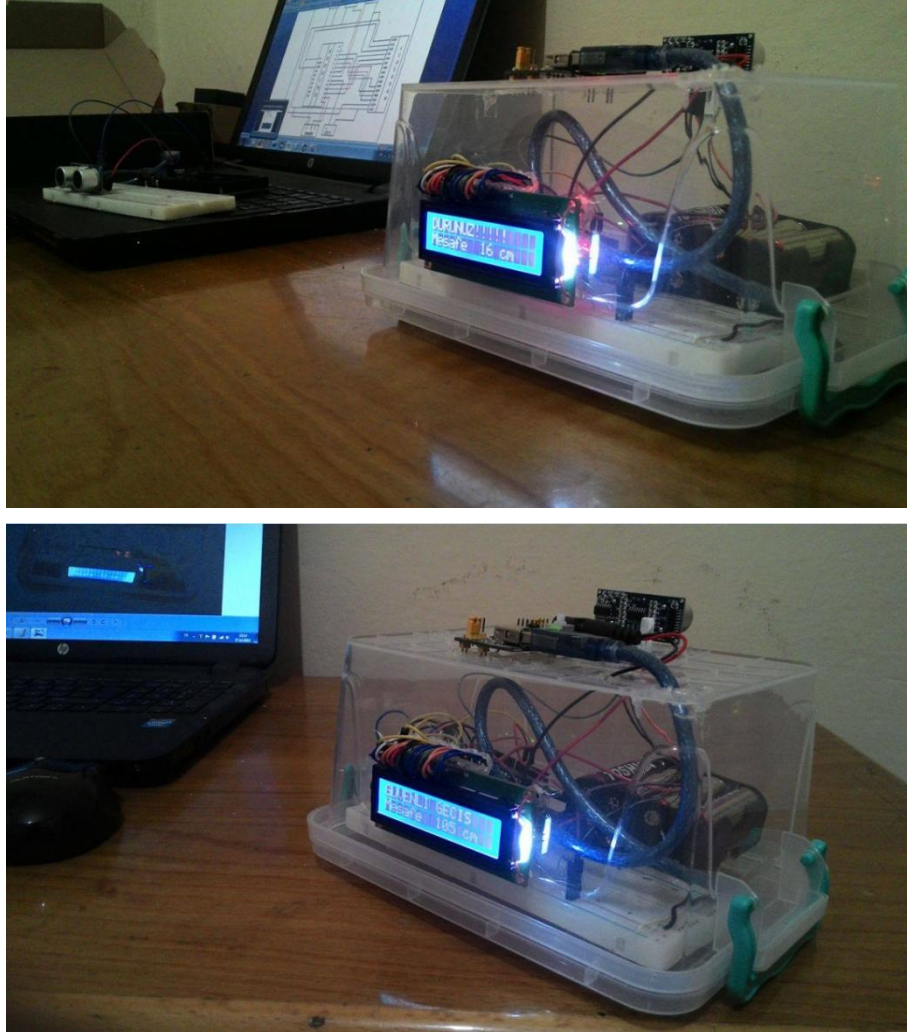
void loop() {
  digitalWrite(trigPin, LOW); // Hoparlör 2ms
  sessiz
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH); // Hoparlör 10ms
  sesli
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW); // Hoparlör sessiz
  microSaniyeSure = pulseIn(echoPin, HIGH); //
  Mikrofonu dinle, yansıma süresini microSaniyeSure
  değişkenine al
  cmMesafe = microSaniyeSure/58.3; //
  microSaniyeSure'dan cm olarak mesafeyi hesapla
  lcd.clear();
  if (cmMesafe >60){
    digitalWrite(led2,HIGH); //yeşil led aktif
```

```
digitalWrite(led1,LOW); //kırmızı ledi söndür
lcd.setCursor(0,0); // imleci 1.satıra getir
lcd.print("GUVENLI GECİS"); // 1. satıra yaz
}
if (cmMesafe >= 40 && cmMesafe <=60){
digitalWrite(led1,LOW); //kırmızı ledi söndür
digitalWrite(led2,HIGH); //yeşil ledi 1 sn yak
delay(1000);
digitalWrite(led2,LOW); //yeşil ledi 1 sn söndür
delay(1000);
tone(ton,200000, 150); // ses tonu ayarı
lcd.setCursor(0,0); // imleci 1.satıra getir
lcd.print("YAVASLAYINIZ");// 1. satıra yaz
delay(300);
}
if (cmMesafe >=20&& cmMesafe <40 ){
digitalWrite(led1,HIGH); //kırmızı ledi 0,1 sn yak
delay(100);
digitalWrite(led1,LOW); //kırmızı ledi 0,1 sn
söndür
digitalWrite(led2,LOW); //yeşil ledi söndür
tone(ton, 200000,250); // ses tonu ayarı

lcd.setCursor(0,0); // imleci 1.satıra getir
lcd.print("DURUNUZ");// 1. satıra yaz
delay(100);
}
if (cmMesafe <20 ){
digitalWrite(led1,HIGH); //kırmızı led aktif
digitalWrite(led2,LOW); //yeşil ledi söndür
tone(ton, 200000); // ses tonu ayarı
lcd.setCursor(0,0); // imleci 1.satıra getir
lcd.print("GECMIS OLSUN");// 1. satıra yaz
}
/* Seri iletişimden ölçülen mesafeyi lcd ye gönder
*/
lcd.setCursor(0,2);
lcd.print("Mesafe ");
lcd.print(cmMesafe); //Mesafe bilgisi
lcd.print(" cm");// Birim değeri
Serial.println(cmMesafe);

delay(1000); //Sonraki ölçümden önce bekleme
süresi.
}
```





Şekil 12. Çalışmanın Donanım Yapısı

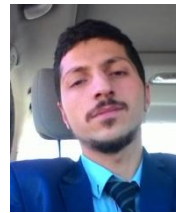
5. SONUÇ

Yapılan deneyler ve ölçümler sonucu uygulamanın başarılı bir şekilde çalıştığı görülmüştür. Lcd ekranındaki mesafe değerlerinin doğruluğunu kontrol etmek amaçlı metre ile ölçüm yapılmıştır ve doğruluğu gözlemlenmiştir. Buda bize yapılan uygulamanın aracımıza gönül rahatlığıyla monte edilebileceğini göstermektedir. Bu sayede kaza yapma riskini en aza indirmiş olup, maliyet açısından büyük kazanç sağlamış oluruz.

Bu uygulamaya ek olarak kablosuz alıcı ve kablosuz verici modülü eklenerek kablosuz

uygulaması yapılabilir. Böylelikle araca monte ederken ekstra kablolar çekmekten kurtulmuş oluruz. Yine uygulamada kullanılmış olan malzemelere ek olarak dc yada servo motor eklenerek engelden kaçan robot uygulaması da yapılabilir. Benzer uygulamaları aşağıdaki linklerde bulabilirsiniz. <http://www.instructables.com/id/Wireless-Car-Parking-Alarm/> <http://www.bajdi.com/obstacle-avoiding-robot-made-from-cheap-parts/>

The Authors



Mustafa Güneş is a student in Mechatronic Engineering at Karabuk University, Turkey. He is born in Sarıkamış/KARS. Autocad, Ansys, Visual Studio, Matlab, Plc, Arduino and is working on robotic systems. Excavators, differential system, brake system, freight elevator, has been found in several studies , such as tower cranes, with ANSYS.



İbrahim Çayıroğlu is an insructor in Mechatronic Engineering at Karabuk University, Turkey. He received his B.Sc. in Mechanical Engineering from Istanbul Technical University in 1991.

He received his M.Sc. and Ph.D. in Computer Aided Design and Manufacturing from Kirikkale University, in 1996 and 2002, respectively. His research interests include CAD-CAM, Software and Mechatronic Systems.