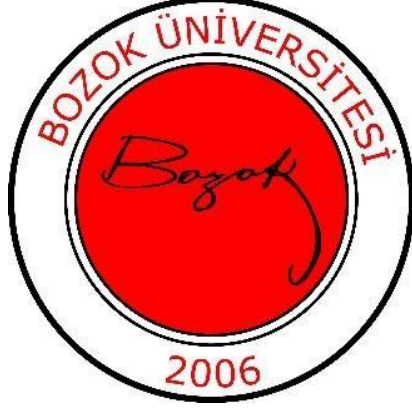


**BOZOK ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**EV OTOMASYON SİSTEMİ**

**20/06/2020**

**YAZILIM MÜHENDİSLİĞİ PROJESİ**

**TEKNİK DOKÜMAN**

**Proje Danışmanı: Arş. Gör. Hasan ULUTAŞ**

**Grup 9**

**Esra YÜCE**

**Feyza YILMAZ**

**Halil SEÇİLMİŞ**

**Özlem ÖZKAYA**

<https://github.com/Ev-Otomasyon-Sistemi>

## İÇİNDEKİLER

TEKNİK DOKÜMAN .....	1
1. Mobil Uygulama .....	3
1.1. Mobil Uygulamanın İşleyiş Mantığı .....	3
1.2. Mobil Uygulama Kodları ve Açıklamaları .....	3
2. Arduino Sistemi .....	7
2.1. Arduino Sisteminin İşleyiş Mantığı .....	7
2.2. Arduino Kodları ve Açıklamaları .....	9
3. Veri Günlüğü .....	20
3.1. Veri Günlüğü İşleyiş Mantığı .....	20
3.2. Veri Günlüğü Kodları .....	20
4. Birim Testi .....	28
4.1. Mobil Uygulama Test Kodları ve Açıklamaları .....	28
4.2. Arduino Test Kodları .....	31
5. Entegrasyon Testi .....	34
6. Veri Toplama .....	36
6.1. Verilerin Toplanması ve Kontrol Edilmesi .....	39

# TEKNİK DOKÜMAN

## 1. Mobil Uygulama

Ev otomasyon sisteminde kullanılacak sensörlerin uzaktan kontrolünü sağlamak amaçlı tasarlanmıştır. Kullanıcı evde olsun veya olmasın telefonda Ev Otomasyon Sistemi dâhilindeki bütün sistemleri kontrol edebilir.

### 1.1. Mobil Uygulamanın İşleyiş Mantığı

Gelişmiş özellikleri sayesinde yaşam konforunuzun üst seviyelere çıkmasına yardımcı olan Ev Otomasyon Sistemi'nde aydınlatma, kapı-perde kontrolü ve oda sıcaklığını görüntüleme komutu gibi özellikleri bulunur. Bu gibi birçok özelliğin bulunduğu sistemi kontrol etmede mobil uygulama kullanılır.

Mobil uygulama, kullanışlı ara yüzü sayesinde her yaşa hitap eden kullanıcı dostu kolay bir kullanım sağlar. Arduino ünitesine bağlı sensörlerin kontrol bağlantısı bluetooth cihazı aracılığıyla gerçekleşir. Bağlantı sağlanan mobil uygulamadan Arduino ünitesine bağlı olan bütün sensörler kontrol edilebilir.

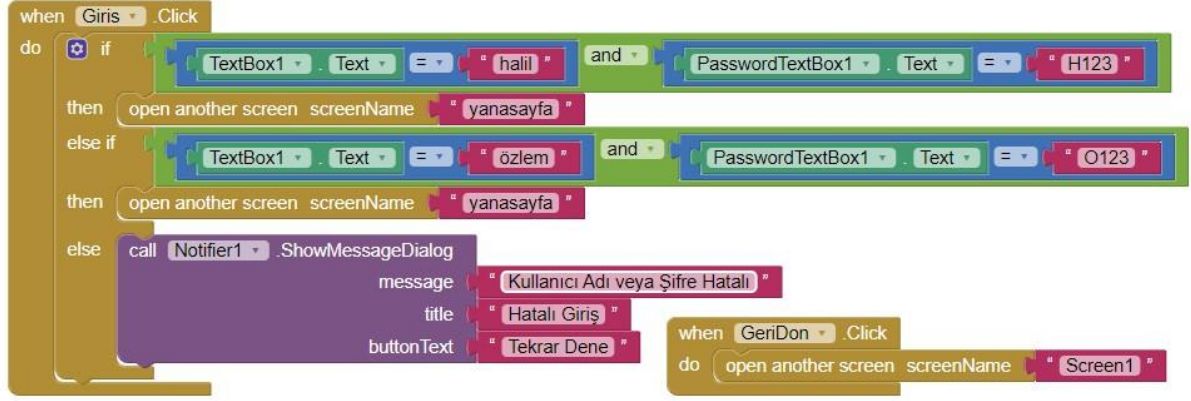
Kullanıcı hesaplarında güvenlik amaçlı bazı kısıtlamalar yer almaktadır. Bu kısıtlamalar daha çok küçük yaştaki kullanıcılar için geçerlidir. Bu sayede birçok sorunu önleyecek güvenlik sağlanır.

### 1.2. Mobil Uygulama Kodları ve Açıklamaları

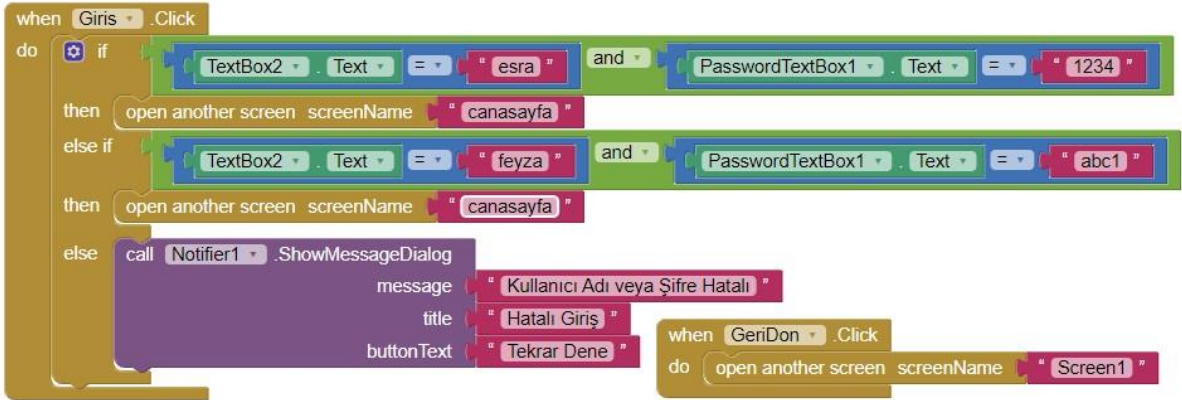
#### Kullanıcı Tercih Sayfası:



### Yetişkin Giriş Ekranı



### Çocuk Giriş Ekranı



### Anasayfa:

```
when cikisy .Click
do open another screen screenName "Screen1 "

when ListPicker1 .BeforePicking
do if BluetoothClient1 . Available
then set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when baglantiyiKes .Click
do call BluetoothClient1 .Disconnect

when ListPicker1 .AfterPicking
do set ListPicker1 . Selection to call BluetoothClient1 .Connect
address ListPicker1 . Selection
```

```
when salonAc .Click
do call BluetoothClient1 .SendText
text " A "
```

```
when cocukAc .Click
do call BluetoothClient1 .SendText
text " C "
```

```
when yatakOdaAc .Click
do call BluetoothClient1 .SendText
text " E "
```

```
when mutfakAc .Click
do call BluetoothClient1 .SendText
text " G "
```

```
when koridorAc .Click
do call BluetoothClient1 .SendText
text " I "
```

```
when bahceAc .Click
do call BluetoothClient1 .SendText
text " K "
```

```
when kapiAc .Click
do call BluetoothClient1 .SendText
text " M "
```

```
when perdeAc .Click
do call BluetoothClient1 .SendText
text " P "
```

```
when sicaklikKontrol .Click
do call BluetoothClient1 .SendText
text " Z "
```

```
when MesafeKontrol .Click
do call BluetoothClient1 .SendText
text " O "
```

```
when salonKapa .Click
do call BluetoothClient1 .SendText
text " B "
```

```
when cocukKapa .Click
do call BluetoothClient1 .SendText
text " D "
```

```
when yatakOdaKapa .Click
do call BluetoothClient1 .SendText
text " F "
```

```
when mutfakKapa .Click
do call BluetoothClient1 .SendText
text " H "
```

```
when koridorKapa .Click
do call BluetoothClient1 .SendText
text " J "
```

```
when bahceKapa .Click
do call BluetoothClient1 .SendText
text " L "
```

```
when kapiKapa .Click
do call BluetoothClient1 .SendText
text " N "
```

```
when perdeKapa .Click
do call BluetoothClient1 .SendText
text " R "
```

```
when GazKontrol .Click
do call BluetoothClient1 .SendText
text " S "
```

```
when ToprakNemKontrol .Click
do call BluetoothClient1 .SendText
text " W "
```

Uygulama 5 ayrı sayfadan oluşmaktadır. Sayfa düzeni için dikey sıralamada **VerticalArrangement**(Birbirinin altında görüntülenmesi gereken bileşenlerin yerleştirileceği bir biçimlendirme ögesi.), yatay sıralamada ise **HorizontalArrangement**(Soldan sağa görüntülenmesi gereken bileşenlerin yerleştirileceği bir biçimlendirme ögesi.) kullanılmıştır. Bu öğelerin içerisine gerekli Button, TextBox ve Label bileşenleri yerleştirilmiştir.

İlk sayfada iki tane buton yer almaktadır. Butonlara tıklanarak **open another screen** komutu ile butona atanan sayfaya yönlendirme yapılır. Bu iki butonun her birine atanan farklı sayfalar dolayısıyla farklı sayfalara geçiş sağlanır. Açılan sayfalar Çocuk Girişi ve Yetişkin Girişi sayfalarıdır. Bu sayfalarda kullanıcı, kullanıcı adını ve şifresini girer. Girilen bilgiler giriş kontrolüne bağlı olan **matematik** ve **lojik** yapılar sayesinde kayıtlı kullanıcı bilgileri ile karşılaştırılır. Bilgiler doğru ise 'Giriş Yap' butonuna atanan sayfaya geçiş yapılır, bilgiler yanlış ise '**Kullanıcı adı veya şifre hatalı.**' uyarısı verip '**Tekrar dene**' seçeneği sunar. Bu hata mesajı **Notifier**(uyarı iletişim kutularını, iletileri ve geçici uyarıları görüntüler) ögesi ile sağlanır. Giriş sağlandıktan sonra son olarak Ev otomasyon sistemi maketinde bulunan sensörlerin kontrolünün sağlandığı ana sayfa açılır. Bluetooth bağlantısını sağlamak için ilk olarak **.BeforePickink** komutu ile bluetooth adresleri aranır ve **.AfterPicking** kontrolünde **.Connect address** ile çıkan ve bluetooth cihazımız ile eşleşen adres seçilir. Aynı şekilde bağlantıyı kesmek için **.Disconnect** kullanılır. Sensör kontrolünü sağlayan her bir buton bluetooth bağlantısı sayesinde **.SendText** komutu ile veri göndererek sensör kontrolünü sağlar. Kontrolü sağlanan belirli sensörlerin işlem değeri LDC ekrana yansıtılır.

## 2. Arduino Sistemi

Ev otomasyon sistemi özelliklerinin gerçekleştirildiği tüm sensörleri kapsayan sistemdir. Mobil uygulamadan verilen komutları işleyerek kullanıcının istediği bir şekilde geri dönüş sağlanmak amaçlanır.

### 2.1.Arduino Sisteminin İşleyiş Mantiğı

Ev otomasyon sisteminde aşağıda tanımları verilen sensör ve modüller yer almaktadır.

**Arduino Uno:** çeşitli sensörlerden fiziksel bilgi almak, bu bilgiler ile çeşitli mekanizmalar tasarlamak için kullanılır.

Ayrıca motor, LED, buzzer gibi uyarıcılardan bir çıktı elde etmek için kullanılabilir.

**LED:** elektrik enerjisini ışıığa dönüştüren yarı iletken bir devre elemanıdır.

**Servo Motor:** bir mekanizmanın performansını etkileyecek hataları; geri bildirim sinyalleri yardımıyla denetler. Kısa zaman aralığında hataları kontrol edecek ve bu hataları engelleyecek, çoğunlukla bir DC tür motor çeşididir.

Servo motorlar, dönüş yönünün belirli açılarda dönmesi istenilen uygulama alanlarında çok tercih edilmektedir.

**Dalgıç Su Pompası:** genellikle basınç ile su temini sağlar.

**Çift Kanallı Motor Sürücü:** iki yönde de iki motoru birbirlerinden bağımsız olarak kontrol edebileceğimiz bir motor sürücü kartıdır.

**Bluetooth Modülü:** kablosuz seri haberleşme uygulamaları için tasarlanmıştır.

**Isı-Nem Sensörü:** sıcaklık ve nem algılayıcı kalibre edilmiş dijital sinyal çıkışı veren gelişmiş bir sensördür.

**Doğal Gaz ve Metan Gazı Ölçümleme Modülü:** 300ppm ve 10000ppm arasında gaz kaçağı tespiti için uygun bir dizi konsantrasyonlarda metan (CNG) doğal gaz varlığını algılar.

**Toprak-Nem Sensörü:** toprağın içerisindeki nem miktarını veya ufak ölçekte bir sıvının seviyesini ölçmek için kullanılan bir sensördür.

**Mesafe Sensörü:** mesafe ölçmek amacıyla kullanılan araçtır. Sensörün bulunduğu yerden saptadığı cisim ile aradaki mesafeyi elektriksel çıkış olarak verir.

**Hareket Sensörü:** nesnelerin veya canlıların hareketlerini kızılötesi, ultrasonik, mikrodalga, titreşim yöntemleriyle algılanmasını ve bu algılama sonucu ilgili mekanizmaları tetiklemesini sağlayan aktif ve pasif özellikteki sensörlerdir.

**Buzzer:** verilen voltaja göre farklı ses sinyalleri sağlayan bir cihazdır.

Ev otomasyon sistemi maketinde, belli başlı sensörler ve modüller bir arada kullanılıp, gerçekleştirilmek istenen işlevi yerine getirecek şekilde devre sistemi tasarlanmıştır. Bunun yanında tüm sensör ve modüller Arduino uno ile bağlantılıdır.

Sistemin bütün bu bileşenleri kapsayan ve haberleşerek bir arada etkileşimini sağlayan elektronik devresi bulunmaktadır. Devre karmaşıklığı, baskı devre oluşturularak önlenmiş olup daha rahat çalışma ortamı sağlanmıştır.

Kapı sisteminde, mobil uygulama kontrolü dâhilinde kapıyı açıp kapatmak için servo motor kullanılmıştır. Servo motor ile etkileşim içerisinde olan hareket sensörü ve buzzer ile yabancı girişleri önlemek amaçlanmıştır. Hareketi algılayan hareket sensörü buzzer sistemini tetikler ve alarm uyarısı verilir.

Çiçek sulama sisteminde ise toprağın nem değerini ölçmek için toprak-nem sensörü kullanılmıştır. Toprak-nem sensörü ile etkileşimde olan dalgıç su pompası, toprağın kuruluğu algılanması durumunda çiçeklere su pompalar.



## 2.2.Arduino Kodları ve Açıklamaları

```
kod §
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

#include "DHT.h"
#define DHTPIN 2
#define DHTTYPE DHT11
DHT evotomasyonu(DHTPIN, DHTTYPE);

#include <Servo.h>
Servo kapiservo;
Servo pencereservo;

const int mileri=9;
const int mgeri=10;

const int pirPin = 8;
const int gazsensor=A0;
const int trig = 3;
const int echo = 4;

const byte latchPin = 7;
const byte dataPin = 6;
const byte clockPin = 5;
int x = 0;
int sayac=0,btgelen,btgelenveri,gazdegeri,nemdegeri,sicaklikdegeri,mesafedegeri,skontrol;
```

#include <Wire.h> bağlantı fonksiyonlarını bulunduran kütüphane

#include <LiquidCrystal\_I2C.h> :LCD ekran çalışma ve bağlantısını bulunduran kütüphane

#include "DHT.h" sıcaklık-nem sensörü bağlantısını bulunduran kütüphane

#include <Servo.h> servo motor bağlantısını bulunduran kütüphane

Burada sisteme gerekli kütüphane ekleme işlemleri yapılır.

LiquidCrystal\_I2C lcd (0x27, 16, 2);

LCD ekranın adresi ve ekran boyutu hakkında bilgi verildi. #define

DHTTYPE DHT11

**#define** (sabit değişkenler için kullanılır) sıcaklık-nem sensörü için kullanılacak değişken tanımlandı Servo kapiservo;

Servo pencereservo;

**Servo** kütüphanesindeki nesneler üretildi ve kapı ve pencere için isimlendirildi. **const**(sabit değerler için kullanılır) Burada sistemde kullanılacak değişkenlerin pinleri belirlendi. Yani arduinodaki konumları belirlendi. Çoklu led kontrolü için **Shift Register**(74HC595) kullanıldı hafızada tutma işlemleri için **.int** değişebilen yani sabit olmayan değişkenler tanımlandı.

```
int buzzerpin=13;

byte led1,led2,led3,led4,led5,led6;

byte kapi,penc;
void setup()
{
  Serial.begin(9600);
  lcd.begin(); //lcd hazırlandı
  lcd.backlight(); // arka plan ışığı ayarlandı
  kapiservo.attach(A3); // attaches the servo on pin 9 to the servo object
  pencereservo.attach(A1 );
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(trig, OUTPUT);
  pinMode(echo, INPUT);
  pinMode(pirPin, INPUT);
  pinMode(mileri,OUTPUT);
  pinMode(mgeri,OUTPUT);
  pinMode(buzzerpin,OUTPUT);
}
```

**Void setup()** metottu içinde genel ayarlamalar yapıldı ve bu metot sadece 1 kez çalışır. **Serial.begin()** ile seri haberleşme hızı belirlendi. **lcd.begin()**, **lcd.backlight()** LCD ekranın genel olarak ayarladı arka ışığı. **kapiservo.attach()**, **pencereservo.attach()** servo kütüphanesinde oluşturduğumuz nesnenin Arduinodaki pini belirlendi. **pinMode()** ile ledler için tanımlanan pinleri çıkış pini olarak ayarlandı.

```

    lcd.setCursor(7,0); //yazının lcd ekrandaki konumu belirlendi
    lcd.print("EV");
    lcd.setCursor(3,1);
    lcd.print("OTOMASYONU");
    delay(3000);
    lcd.clear();

}

void loop()
{
    if(Serial.available() > 0){
        btgelenveri = Serial.read();
    }
    //Serial.println(btgelenveri);

    bluetoothkontrol(btgelenveri);
    gazdegeri=gaz();
    nemdegeri=topraknem();
    sicaklikdegeri=sicaklik();
    mesafedegeeri=mesafeolc();
    lcdyaz(gazdegeri,nemdegeri,mesafedegeeri,sicaklikdegeri,btgelenveri);
}

```

lcd. **setCursor()** ile LCD konumu, lcd.**print()** LCD ekrana yazılacak yazı, lcd.**clear()** LCD ekranı temizlemesi belirlendi.

**void loop()** metodu arduinonun yapması gereken işlemleri yazıldı. Bu metot birde fazla kez çalışabilir. İlk kullanılan **if** işlemi ile gelen veri kontrol edildi.

**Serial.read()** ile gelen veri aktarıldı. Sonrasında gelen değeri yukarıda tanımlanan değişkenlere atama işlemi yapıldı.

**if(gazdegeri>300)** koşulu ile gelen gaz değeri belirtilenden yüksek ise alarm çalması sağlandı.

**if(mesafedegeeri<4)** koşulu ile arada bırakılan mesafe değeri belirtilenden az olduğu zaman alarm çalması sağlandı.

**if(nemdegeri<400)** koşulu ile nem değeri belirtilenin altında ise sulama işlemi yapılması sağlandı.

```

if(gazdegeri>300){
    buzzer(50);
}
if(mesafedegeri<4){
    buzzer(50);
}
if(nemdegeri>400){
    digitalWrite(mileri,1);
    digitalWrite(mgeri,0);
}
else{
    digitalWrite(mileri,0);
    digitalWrite(mgeri,0);
}

Serial.print(gazdegeri);//çoklu veri için
Serial.print("*");
Serial.print(nemdegeri);
Serial.print("*");
Serial.print(sicaklikdegeri);
Serial.print("*");
Serial.println(mesafedegeri);
Serial.print("*");

```

```

int bluetoothkontrol(int btgelen){

    if(btgelen==65)
    led1=1;
    if(btgelen==66)
    led1=0;

    if(btgelen==67)
    led2=1;
    if(btgelen==68)
    led2=0;

    if(btgelen==69)
    led3=1;
    if(btgelen==70)
    led3=0;

    if(btgelen==71)
    led4=1;
    if(btgelen==72)
    led4=0;

    if(btgelen==73)
    led5=1;
    if(btgelen==74)
    led5=0;
}

```

**int bluetoothkontrol(int btgelen)** burada arduino ile mobil uygulama bağlantısı kontrol ediliyor bunun doğrultusunda diğer işlemler gerçekleşiyor.

Mobil uygulamadaki kapat-aç butonlarına verilen harfler acii tablosundan çevrilerek if koşup yaapısı içine yazıldı. **btnglen**'nin değeri hangi sayıya eşit ise ona göre açma-kapama işlemi yapıldı. (0: kapat,1: aç)

```
if(btngelen==77)
kapi=0;
if(btngelen==78)
kapi=45;

if(btngelen==80)
penc=45;
if(btngelen==82)
penc=0;

if(hareket()==1 && kapi==45 )
buzzer(50);

led(led1,led2,led3,led4,led5,led6);
kapiservo.write(kapi);
delay(20);
pencereservo.write(penc);

delay(150 );
}
```

```
int led(byte a,byte b,byte c,byte d,byte e,byte f){
boolean dizi[] = {0, 0, f, e, d, c, b, a}; //nokta,g,f,e,d,c,b,a = 1 açık 0 kapalı
digitalWrite(latchPin, 0);

for (int i = 0 ; i < 8 ; i++)
{
x = dizi[i]; //for döngü boyunca dizinin i. değeri x'e atanır

digitalWrite(dataPin, x); //datapin'e x değeri verilir
digitalWrite(clockPin, 1); //saat darbesi ile yazma ve kaydırma işlemi yapılır
digitalWrite(clockPin, 0);
}
digitalWrite(latchPin, 1); //8 bitli dizi çıkışa verilir
}
```

Kapı için servo motor uygulamada butona tıklandığı zaman **delay(20)** komutu ile 0.02 saniye sonra işlevini yapar.

Pencere için servo motor uygulamada butona tıklandığı zaman **delay(150)** komutu ile 0.15 saniye sonra işlevini yapar.

Kullanılan ledlerin hepsi fonksiyonun içine parametre olarak aktarıldı. Aşağıda tekrar çağırılarak isimlendirildi. **int led(byte, ). boolean dizi[]** şeklinde dizi oluşturuldu. Çoklu led

kullanılacağı için bir diziye kayıt edip işlemleri gerçekleştirildi. Sonrasında oluşturulan **for (int i = 0 ; i < 8 ; i++)** döngüsü ile ledlerin durumları hafızaya kayıt edildi.

```
void lcdyaz(int gazveri,int nemverisi,int mesafeverisi,int sicaklikverisi,int bluetoothveri){  
  
    //Serial.println(blueoothveri);  
  
    if(blueoothveri==90){  
  
        lcd.setCursor(7,0);  
        lcd.print("SIC");  
        lcd.setCursor(9,1);  
        lcd.print(" ");  
        lcd.setCursor(7,1);  
        lcd.print(sicaklikverisi);  
    }  
}
```

Lcd ekranına yazı yazmak için **lcdyaz** adında bir fonksiyon tanımlandı.

**Lcdyaz** fonksiyonun **gazveri, nemverisi, mesafeverisi, sicaklikverisi, bluetoothveri** olmak üzere 5 tane parametresi vardır ve gelen değerleri lcd ekranına yazdırır.

If komutu bluetooth verisini kontrol eder. Eğer bluetooth verisi 90(sıcaklığı göster anlamına gelir) değerine eşit ise; **setCursor(7,0)** komutu yazının hangi satırda hangi sutunda olacağını belirtir.

7. sutun, 0.satıra ekran imlecini ayarlar. 7.sutun, 0.satır itibariyle ekrana “SIC” yazar.

9. sutun, 1.satıra ekran imlecini ayarlar. 9.sutun, 1.satır itibariyle ekrana “ ” yazar.

7. sutun, 1.satıra ekran imlecini ayarlar. 7.sutun, 1.satır itibariyle ekrana DHT11 sensöründen aldığı **sicaklikverisi**’ni yazar.

```

if(blueetoothveri==79) {
    if(mesafeverisi<100) {
        lcd.setCursor(9,1);
        lcd.print(" ");
        lcd.setCursor(10,1);
        lcd.print(" ");
    }
    if(mesafeverisi<10) {
        lcd.setCursor(8,1);
        lcd.print(" ");
        lcd.setCursor(9,1);
        lcd.print(" ");
    }
    lcd.setCursor(7,0);
    lcd.print("MES");
    lcd.setCursor(7,1);
    lcd.print(mesafeverisi);
}

```

İf komutu bluetooth verisini kontrol eder. Eğer bluetooth verisi 79 değerine eşit ise; Mesafeverisine bakar.

Eğer mesafeverisi 100 değerinden küçük ise;

9. sütun, 1.satıra ekran imlecini ayarlar. 9.sutun, 1.satır itibariyle ekrana " " yazar.

10. sütun, 1.satıra ekran imlecini ayarlar. 10.sutun, 1.satır itibariyle ekrana " " yazar.

Eğer mesafeverisi 10 değerinden küçük ise;

8. sütun, 1.satıra ekran imlecini ayarlar. 8.sutun, 1.satır itibariyle ekrana " " yazar.

9. sütun, 1.satıra ekran imlecini ayarlar. 9.sutun, 1.satır itibariyle ekrana " " yazar.

Eğer hiçbirisi değil ise;

7. sütun, 0.satıra ekran imlecini ayarlar. 7.sutun, 0.satır itibariyle ekrana " MES" yazar.

7. sütun, 1.satıra ekran imlecini ayarlar. 7.sutun, 1.satır itibariyle ekrana HC-SR04 sensöründen aldığı **mesafeverisi** yazar.

```

if(blueetoothveri==83){
    if(gazveri<100){
        lcd.setCursor(9,1);
        lcd.print(" ");
        lcd.setCursor(10,1);
        lcd.print(" ");
    }
    lcd.setCursor(7,0);
    lcd.print("GAZ");
    lcd.setCursor(7,1);
    lcd.print(gazveri);
}

```

İf komutu bluetooth verisini kontrol eder. Eğer bluetooth verisi 83 değerine eşit ise;

**Gazverisine** bakar. Eğer **gazverisi** 100 değerinden küçük ise;

9. sutun, 1.satıra ekran imlecini ayarlar. 9.sutun, 1.satır itibariyle ekrana “ ” yazar.

10. sutun, 1.satıra ekran imlecini ayarlar. 10.sutun, 1.satır itibariyle ekrana “ ” yazar.

Eğer **gazverisi** 100 değerinden büyük ise;

7. sutun, 0.satıra ekran imlecini ayarlar. 7.sutun, 0.satır itibariyle ekrana “ GAZ” yazar.

7. sutun, 1.satıra ekran imlecini ayarlar. 7.sutun, 1.satır itibariyle ekrana MQ-2 sensöründen aldığı **gazverisi** yazar

```

if(blueetoothveri==87){
    if(nemverisi<1000){
        lcd.setCursor(10,1);
        lcd.print(" ");
        lcd.setCursor(11,1);
        lcd.print(" ");
    }
    lcd.setCursor(7,0);
    lcd.print("NEM");
    lcd.setCursor(7,1);
    lcd.print(nemverisi);
}

```

İf komutu bluetooth verisini kontrol eder. Eğer bluetooth verisi 87 değerine eşit ise;

**Nemverisine** bakar. Eğer **nemverisi** 1000 değerinden küçük ise;

10. sutun, 1.satıra ekran imlecini ayarlar. 10.sutun, 1.satır itibariyle ekrana “ ” yazar.

11. sutun, 1.satıra ekran imlecini ayarlar. 11.sutun, 1.satır itibariyle ekrana “ ” yazar.



Eğer **nemverisi** 1000 değerinden büyük ise;

7. sutun, 0.satıra ekran imlecini ayarlar. 7.sutun, 0.satır itibariyle ekrana “NEM” yazar.

7. sutun, 1.satıra ekran imlecini ayarlar. 7.sutun, 1.satır itibariyle ekrana DHT11 sensöründen aldığı **nemverisi** yazar.

```
int gaz() {  
    int gazdeger=analogRead(gazsensor);  
    return gazdeger;  
}  
  
int mesafeolc() {  
    int sure,mesafe;  
    digitalWrite(trig, HIGH);  
    delayMicroseconds(1000);  
    digitalWrite(trig, LOW);  
    sure = pulseIn(echo, HIGH);  
    mesafe = (sure/2) / 29.1;  
    return mesafe;  
}  
  
float sicaklik() {  
    float t = evotomasyonu.readTemperature();  
    return t;  
}
```

Gaz ın çıkış pininden bir değer okunur ve o değer return olarak ana fonksiyona döndürülür.

Gaz isminde int tipinde değer döndüren bir fonksiyon oluşturuldu.

MQ-2 sensöründen okunan analog değer **gazdeger** isimli değişkene aktarılır.

Sonuç olarak sensörden aldığı **gazdegeri**’ni return olarak ana fonksiyona döndürür ve ekrana yazdırılır.

**Mesafeolc** isminde int tipinde değer döndüren bir fonksiyon oluşturuldu. Sure ve mesafe isminde 2 tane değişken oluşturuldu.

İlk olarak trig pinini HIGH durumunda başlatıyoruz.

1 saniye aralıklarla ses dalgası yayılır.

Yeni ses dalgası üretebilmesi için trig pini pasif duruma getirilir.

Mikrofona herhangi bir ses dalgası çarparsa bunun için bir süre tutar. Bu süreye çarpıp geri gelmesinde iki kere iş yapmış olur. Bu süreyi 2 ye bölüp, 29.1(sesin havada yayılma hızı)’e bölünür ve mesafe adlı değişkene atanır.

Sonuç olarak geriye mesafe değişkeni return olarak ana fonksiyona döndürülür ve ekrana yazdırılır.

**Sicaklik** isminde **float** tipinde değer döndüren bir fonksiyon oluşturuldu.

**readTemperature** metodu sayesinde DHT11 sensöründen sıcaklık değeri okunur ve t değişkenine aktarılır.

Sonuç olarak sensörden aldığı sıcaklık değeri return olarak ana fonksiyona döndürülür ve ekrana yazdırılır.

```
int hareket() {  
    int hdeger;  
    int pirDeger = digitalRead(pirPin);  
  
    if (pirDeger == HIGH)  
        hdeger=1;  
    else  
        hdeger=0;  
  
    return hdeger;  
}  
  
int servokapi(int kapiderece) {  
    kapiservo.write(kapiderece);  
}  
  
int servopencere(int pencerederece) {  
    pencereservo.write(pencerederece);  
}
```

Hareketi algılamak için **hareket** isminde int tipinde değer döndüren bir fonksiyon oluşturuldu.

**digitalRead** komutuyla PIR sensöründen okuma yapılır ve **pirDegerine** aktarılır.

Eğer alınan **pirDegerinde** hareketlilik var ise **hdeger** yani hareket değeri 1 olur.

Eğer alınan **pirDegerinde** hareketlilik yok ise **hdeger** yani hareket değeri 0 olur.

Sonuç olarak sensörden aldığı hareket değeri döndürülür.

Kapının kontrolü için **servokapi** isminde int tipinde değer döndüren ve kapı derecesini alan bir fonksiyon oluşturuldu.

**kapiservo.write(kapiderece);** kodu ile servo motorun **kapiderecesi** açılış değerine dönmesini sağlar.

Pencerenin kontrolü için **servopencere** isminde int tipinde değer döndüren ve pencere derecesini alan bir fonksiyon oluşturuldu.

**pencereservo.write(kapiderece);** kodu ile servo motorun **pencerederece** açılış değerine dönmesini sağlar.

```
void motor(int msure) {  
    delay(1000);  
    sayac=sayac+1;  
    if(sayac>msure) {  
        digitalWrite(mileri,0);  
        digitalWrite(mgeri,0);  
    }  
    else{  
        digitalWrite(mileri,0);  
        digitalWrite(mgeri,1);  
    }  
    //Serial.println(sayac);  
}  
  
int topraknem() {  
    int nemdeger=analogRead(A2);  
    return nemdeger;  
}  
void buzzer(int ton) {  
    digitalWrite(buzzerpin,1);  
    delay(ton);  
    digitalWrite(buzzerpin,0);  
    delay(ton);  
}  
}
```

**Motor** isminde, motor süresini alan ve değer döndürmeyen bir fonksiyon oluşturuldu.

1000 milisaniye bekletilir. Sayaç oluşturulur ve her seferinde sayaç 1 arttırılır.

Eğer sayaç değeri motor süresinden büyükse, **digitalWrite** fonksiyonu sayesinde motor ileri pini logic 0 ve motor geri pini logic 0 yapılır.

Sayaç değeri motor süresinden küçükse, motor ileri pini logic 0 ve motor geri pini logic 1 yapılır.

Toprak ve nem ölçümü yapmak için **topraknem** isimli int tipinde değer döndüren ve değer almayan bir fonksiyon oluşturuldu.

Toprak nem sensöründen okunan analog değer **nemdeger** isimli değişkene aktarılır.

Sonuç olarak sensörden aldığı **nemdegeri** döndürür.

Buzzer kontrolünü sağlamak için **buzzer** isminde değer döndürmeyen ve int tipinde değer alan bir fonksiyon oluşturuldu.

**digitalWrite** fonksiyonu sayesinde buzzerpin pini logic 1 yapılır. **Delay(ton)** fonksiyonu sayesinde buzzer ses çıkartır.

**digitalWrite** fonksiyonu sayesinde buzzerpin pini logic 0 yapılır. **Delay(ton)** fonksiyonu sayesinde buzzer ses çıkartır.

### 3. Veri Günlüğü

#### 3.1. Veri Günlüğü İşleyiş Mantığı

Seri port haberleşmesi ile ekranda toprak nem, mesafe, yabancı gaz ve sıcaklık değerlerinin görüntülenmesi sağlanır. Görüntülenen değerler veri tabanına kaydedilir. Herhangi bir ihtiyaç durumunda veri tabanında kayıtlı veriler görüntülenir ve çıktısı alınabilir.

Evin güvenliğini sağlayan kamera görüntülerine erişilir. Herhangi bir kısımdan kesitler alıp kaydedilebilir.

#### 3.2. Veri Günlüğü Kodları

##### Kullanıcı Kimlik Doğrulama(Form1)

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.Data.SqlClient;
11 namespace Güvenlik_Veri
12 {
13     //kullanici_giris
14     public partial class Form1 : Form
15     {
16         SqlConnection Baglanti = new SqlConnection("Data Source=LAPTOP-ISLQI3BT\\SQLEXPRESS;Initial Catalog=Arduino_Veri;Integrated Security=True");
17         // SqlConnection Baglanti = new SqlConnection(@"Data Source=.;Initial Catalog=kullanici;Integrated Security=True");
18
19         public Form1()
20         {
21             InitializeComponent();
22         }
23
24         private void button2_Click(object sender, EventArgs e)
25         {
26             Application.Exit();
27         }
28
29         private void button1_Click(object sender, EventArgs e)
30         {
31             try
32             {
33                 Baglanti.Open();
34             }
35         }
36     }
37 }
```

```

35         Baglanti.Open();
36         string sql = "Select * From kullanici_giris where k_adi=@adi And k_sifre=@sifre ";
37         SqlParameter prm1 = new SqlParameter("adi", textBox3.Text.Trim());
38         SqlParameter prm2 = new SqlParameter("sifre", textBox4.Text.Trim());
39         SqlCommand komut = new SqlCommand(sql, Baglanti);
40         komut.Parameters.Add(prm1);
41         komut.Parameters.Add(prm2);
42         DataTable dt = new DataTable();
43         SqlDataAdapter da = new SqlDataAdapter(komut);
44         da.Fill(dt);
45
46         if (dt.Rows.Count > 0)
47         {
48             secme f2 = new secme();
49             f2.Show();
50         }
51     }
52 }
53 catch (Exception)
54 {
55     MessageBox.Show("Kullanıcı Adı veya Şifre Hatalı.");
56 }
57 }
58 }
59
60
61
62 private void checkBox1_CheckedChanged(object sender, EventArgs e)
63 {
64     if (checkBox1.Checked)
65     {
66         textBox4.PasswordChar = '\0';
67     }
68     else
69     {
70         textBox4.PasswordChar = '*';
71     }
72 }

```

## Kamera ve Veri Kaydı Seçme Ekranı(secme)

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace Güvenlik_Veri
12 {
13     public partial class secme : Form
14     {
15         public secme()
16         {
17             InitializeComponent();
18         }
19
20         private void button1_Click(object sender, EventArgs e)
21         {
22             Kamera k = new Kamera();
23             k.Show();
24         }
25
26         private void button2_Click(object sender, EventArgs e)
27         {
28             Veri v = new Veri();
29             v.Show();
30         }
31
32         private void btn_cikis_Click(object sender, EventArgs e)
33         {
34             Application.Exit();
35         }
36     }

```

## Kamera Güvenlik Sistemi(kamera)

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using AForge.Video; /*
11 using AForge.Video.DirectShow; /*
12 using System.IO;
13
14
15 namespace Güvenlik_Veri
16 {
17     public partial class Kamera : Form
18     {
19         public Kamera()
20         {
21             InitializeComponent();
22         }
23         private FilterInfoCollection webcamsayisi; //Kamera nesnelerini yaratıyoruz
24         private VideoCaptureDevice kamera;
25         private void Kamera_Load(object sender, EventArgs e)
26         {
27             webcamsayisi = new FilterInfoCollection(FilterCategory.VideoInputDevice);
28             foreach (FilterInfo videocapturedevice in webcamsayisi)
29             {
30                 comboBox1.Items.Add(videocapturedevice.Name); //Kayıtlı kameralar
31             }
32             // comboBox1.SelectedIndex = 0;
33         }
34         private void kullanilacakcihaz_NewFrame(object sender, NewFrameEventArgs eventArgs)
35         {
36             Bitmap bit = (Bitmap)eventArgs.Frame.Clone();
37             pictureBox1.Image = bit;
38         }
39
40         private void btn_basla_Click(object sender, EventArgs e)
41         {
42             kamera = new VideoCaptureDevice(webcamsayisi[comboBox1.SelectedIndex].MonikerString);
43             kamera.NewFrame += new NewFrameEventHandler(kullanilacakcihaz_NewFrame);
44             kamera.Start();
45         }
46
47         private void btn_dur_Click(object sender, EventArgs e)
48         {
49             if (kamera.IsRunning) //kamera açıksa kapatıyoruz.
50             {
51                 kamera.Stop();
52             }
53         }
54
55         private void btn_kyit_Click(object sender, EventArgs e)
56         {
57             SaveFileDialog saveFileDialog1 = new SaveFileDialog();
58             saveFileDialog1.Filter = "Jpeg Resmi|*.jpg|Bitmap Resmi|*.bmp|Gif Resmi|*.gif";
59             saveFileDialog1.Title = "Resmi Kaydet";
60             saveFileDialog1.ShowDialog();
61
62             if (saveFileDialog1.FileName != "")
63             {
64                 FileStream DosyaAkisi = (FileStream)saveFileDialog1.OpenFile();
65                 switch (saveFileDialog1.FilterIndex)
66                 {
67                     case 1:
68                         pictureBox1.Image.Save(DosyaAkisi, System.Drawing.Imaging.ImageFormat.Jpeg);
69                         break;
70
71                     case 2:
72                         pictureBox1.Image.Save(DosyaAkisi, System.Drawing.Imaging.ImageFormat.Bmp);
73                         break;
74
75                     case 3:
76                         pictureBox1.Image.Save(DosyaAkisi, System.Drawing.Imaging.ImageFormat.Gif);
77
78                     case 3:
79                         pictureBox1.Image.Save(DosyaAkisi, System.Drawing.Imaging.ImageFormat.Gif);
80                         break;
81                 }
82                 DosyaAkisi.Close();
83             }
84
85         private void btn_cikis_Click(object sender, EventArgs e)
86         {
87             this.Close();
88             Application.Exit(); //çıkış
89         }
90
91         private void pictureBox1_Click(object sender, EventArgs e)
92         {
93         }
94
95         private void button1_Click(object sender, EventArgs e)
96         {
97             secme s = new secme();
98             s.Show();
99         }
100     }

```

## Veri Kaydı(Veri)

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Windows.Forms;
9 using System.IO.Ports;
10 using System.Data.SqlClient;
11 using System.IO;
12
13
14 namespace Güvenlik_Veri
15 {
16     public partial class Veri : Form
17     {
18         string[] ports = SerialPort.GetPortNames(); //Port Numaralarını ports isimli diziye atıyoruz.
19         SqlConnection baglanti = new SqlConnection("Data Source=LAPTOP-1SLQ1387\\SQLEXPRESS;Initial Catalog=Arduino_Veri;Integrated Security=True");
20         DataTable tablo = new DataTable();
21
22         public Veri()
23         {
24             InitializeComponent();
25         }
26
27         private void button1_Click(object sender, EventArgs e)
28         {
29             timer1.Start(); //250 ms olarak ayarladım timer'ı.
30             if (serialPort1.IsOpen == false)
31             {
32                 if (comboBox1.Text == "")
33                     return;
34                 serialPort1.PortName = comboBox1.Text; // comboBox1'e zaten port isimlerini aktarmıştık.
35                 serialPort1.BaudRate = Convert.ToInt16(comboBox2.Text); //Seri Haberleşme baudrate'i comboBox2 'de seçilene göre belirliyoruz.
```

```
36         try
37         {
38             serialPort1.Open(); //Haberleşme için port açılıyor
39             label3.ForeColor = Color.Green;
40             label3.Text = "Bağlantı Açık";
41
42         }
43         catch (Exception hata)
44         {
45             MessageBox.Show("Hata:" + hata.Message);
46         }
47     }
48     else
49     {
50         label3.Text = "Bağlantı kurulu !!!";
51     }
52 }
53
54
55 private void button2_Click(object sender, EventArgs e)
56 {
57     //BAĞLANTIYI KES BUTONU
58     timer1.Stop();
59     if (serialPort1.IsOpen == true)
60     {
61         serialPort1.Close();
62         label3.ForeColor = Color.Red;
63         label3.Text = "Bağlantı Kapalı";
64     }
65 }
66
67 private void button3_Click(object sender, EventArgs e)
68 {
69     listBox1.Visible = true;
70     listBox1.Items.Add(label1.Text); //Okunan veri listBox'a atılıyor
71
72 }
73 }
```

```

75 private void button4_Click(object sender, EventArgs e)
76 {
77     listBox1.Items.Clear(); // listbox temizleniyor.
78 }
79
80
81 private void button5_Click(object sender, EventArgs e)
82 {
83     baglanti.Close();
84     string m = Convert.ToString(listBox1.SelectedItem).Substring(0, 2);
85
86     // string m = listBox1.Items.ToString();
87     string s = Convert.ToString(listBox1.SelectedItem).Substring(3, 3);
88     string h = Convert.ToString(listBox1.SelectedItem).Substring(7, 2);
89     string t = "";
90
91
92     if (Convert.ToString(listBox1.SelectedItem).Count() - 2 == 14)
93     {
94         t = Convert.ToString(listBox1.SelectedItem).Substring(10, 3);
95     }
96
97     else if (Convert.ToString(listBox1.SelectedItem).Count() - 2 == 13)
98     {
99         t = Convert.ToString(listBox1.SelectedItem).Substring(10, 2);
100
101     }
102     else if (Convert.ToString(listBox1.SelectedItem).Count() - 2 == 12)
103     {
104         t = Convert.ToString(listBox1.SelectedItem).Substring(10, 1);
105     }
106

```

```

106
107     string komut = "insert into Veri_gunlugu2 (Sicaklik,Mesafe,Yabanci_gaz,Toprak_nem) values('" + m + "','" + s + "','" + h + "','" + t + "')";
108     SqlCommand sorgu = new SqlCommand(komut, baglanti);
109     try
110     {
111         baglanti.Open();
112         sorgu.ExecuteNonQuery();
113         MessageBox.Show("Kayıt Eklendi");
114         baglanti.Close();
115     }
116     catch (Exception ex)
117     {
118
119         MessageBox.Show("hata var", ex.Message);
120     }
121 }
122
123 private void timer1_Tick(object sender, EventArgs e)
124 {
125     try
126     {
127         string sonuc = serialPort1.ReadExisting();//Serial.print kodu ile gelen analog veriyi alıyoruz,string formatında sonuc'a atıyoruz
128         label1.Text = sonuc + " "; //Labele yazdırıyoruz.
129     }
130     catch (Exception ex)
131     {
132         MessageBox.Show(ex.Message); // başarısız olursa hata verecek.
133         timer1.Stop();
134     }
135 }
136

```

```

136
137 private void Veri_Load(object sender, EventArgs e)
138 {
139     foreach (string port in ports)
140     {
141         comboBox1.Items.Add(port); // Port isimlerini combobox1'de gösteriyoruz.
142         comboBox1.SelectedIndex = 0;
143     }
144     comboBox2.Items.Add("2400"); // Baudrate'leri kendimiz combobox2'ye giriyoruz.
145     comboBox2.Items.Add("4800");
146     comboBox2.Items.Add("9600");
147     comboBox2.Items.Add("19200");
148     comboBox2.Items.Add("115200");
149     comboBox2.SelectedIndex = 2;
150
151     label3.Text = "Bağlantı Kapalı"; //Bu esnada bağlantı yok.
152 }
153
154 private void Veri_FormClosed(object sender, FormClosedEventArgs e)
155 {
156     // Form kapandığında Seri Port Kapatılmış Olacak.
157     if (serialPort1.IsOpen == true)
158     {
159         serialPort1.Close();
160     }
161 }
162
163 private void button8_Click(object sender, EventArgs e)
164 {
165     this.Close();
166     Application.Exit();
167 }
168
169 private void button7_Click(object sender, EventArgs e)
170 {
171     secme s = new secme();
172     s.Show();
173 }
174

```



```

175 private void btn_cikis_Click(object sender, EventArgs e)
176 {
177     this.Close();
178     Application.Exit(); ;
179 }
180
181 private void button9_Click(object sender, EventArgs e)
182 {
183     secme s = new secme();
184     s.Show();
185 }
186
187 private void button6_Click(object sender, EventArgs e)
188 {
189     kayıtlar k = new kayıtlar();
190     k.Show();
191 }
192
193
194
195

```

## Veri Tabanı Kayıt Çıktıları (kayıtlar)

```

1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10 using System.Data.SqlClient;
11 using System.Collections;
12
13 namespace Güvenlik_Veri
14 {
15     public partial class kayıtlar : Form
16     {
17         public kayıtlar()
18         {
19             InitializeComponent();
20         }
21
22
23         StringFormat strFormat;
24         ArrayList arrColumnLefts = new ArrayList();
25         ArrayList arrColumnWidths = new ArrayList();
26         int icellHeight = 0;
27         int itotalWidth = 0;
28         int iRow = 0;
29         bool bFirstPage = false;
30         bool bNewPage = false;
31         int iHeaderHeight = 0;
32
33

```

```

36
37 SqlConnection baglanti = new SqlConnection("Data Source=LAPTOP-15LQI3BT\\SQLEXPRESS;Initial Catalog=Arduino_Veri;Integrated Security=True");
38 DataTable tablo = new DataTable();
39 private void kayıtlar_Load(object sender, EventArgs e)
40 {
41
42     baglanti.Open();
43     string kayit = "Select *from Veri_gunlugu2";
44     SqlCommand komut = new SqlCommand(kayit, baglanti);
45     SqlDataAdapter da = new SqlDataAdapter(komut);
46     // DataTable dt = new DataTable();
47     da.Fill(tablo);
48     dataGridView1.DataSource = tablo;
49     baglanti.Close();
50 }
51
52 private void btn_cikis_Click(object sender, EventArgs e)
53 {
54     this.Close();
55     Application.Exit();
56 }
57
58 private void button9_Click(object sender, EventArgs e)
59 {
60     Veri v = new Veri();
61     v.Show();
62 }
63

```

```

63 private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
64 {
65     try
66     {
67         int iLeftMargin = e.MarginBounds.Left;
68         int iTopMargin = e.MarginBounds.Top;
69         bool bMorePagesToPrint = false;
70         int iTmpWidth = 0;
71         bFirstPage = true;
72
73         if (bFirstPage)
74         {
75             foreach (DataGridViewColumn GridCol in dataGridView1.Columns)
76             {
77                 iTmpWidth = (int)(Math.Floor(((double)GridCol.Width /
78                     (double)iTotalWidth * (double)iTotalWidth *
79                     ((double)e.MarginBounds.Width / (double)iTotalWidth))));
80
81                 iHeaderHeight = (int)(e.Graphics.MeasureString(GridCol.HeaderText,
82                     GridCol.InheritedStyle.Font, iTmpWidth).Height) + 11;
83
84                 arrColumnLefts.Add(iLeftMargin);
85                 arrColumnWidths.Add(iTmpWidth);
86                 iLeftMargin += iTmpWidth;
87             }
88         }
89     }
90 }

```

```

92 while (iRow <= dataGridView1.Rows.Count - 1)
93 {
94     DataGridViewRow GridRow = dataGridView1.Rows[iRow];
95
96     iCellHeight = GridRow.Height + 5;
97     int iCount = 0;
98
99     if (iTopMargin + iCellHeight >= e.MarginBounds.Height + e.MarginBounds.Top)
100     {
101         bNewPage = true;
102         bFirstPage = false;
103         bMorePagesToPrint = true;
104         break;
105     }

```

```

108 if (bNewPage)
109 {
110     e.Graphics.DrawString("Metin Ev Otomasyonu Sistemi", new Font(dataGridView1.Font, FontStyle.Bold),
111         Brushes.Black, e.MarginBounds.Left, e.MarginBounds.Top -
112         e.Graphics.MeasureString("Metin Ev Otomasyonu Sistemi", new Font(dataGridView1.Font,
113             FontStyle.Bold), e.MarginBounds.Width).Height - 13);
114
115     String strDate = DateTime.Now.ToString("dd/MM/yyyy HH:mm:ss");
116
117     e.Graphics.DrawString(strDate, new Font(dataGridView1.Font, FontStyle.Bold),
118         Brushes.Black, e.MarginBounds.Left + (e.MarginBounds.Width -
119         e.Graphics.MeasureString(strDate, new Font(dataGridView1.Font,
120             FontStyle.Bold), e.MarginBounds.Width).Width), e.MarginBounds.Top -
121         e.Graphics.MeasureString("Metin Ev Otomasyonu Sistemi", new Font(new Font(dataGridView1.Font,
122             FontStyle.Bold), FontStyle.Bold), e.MarginBounds.Width).Height - 13);
123
124     iTopMargin = e.MarginBounds.Top;
125     foreach (DataGridViewColumn GridCol in dataGridView1.Columns)
126     {
127         e.Graphics.FillRectangle(new SolidBrush(color.LightGray),
128             new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
129             (int)arrColumnWidths[iCount], iHeaderHeight));
130
131         e.Graphics.DrawRectangle(Pens.Black,
132             new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
133             (int)arrColumnWidths[iCount], iHeaderHeight));
134
135         e.Graphics.DrawString(GridCol.HeaderText, GridCol.InheritedStyle.Font,
136             new SolidBrush(GridCol.InheritedStyle.ForeColor),
137             new Rectangle((int)arrColumnLefts[iCount], iTopMargin,
138             (int)arrColumnWidths[iCount], iHeaderHeight), strFormat);
139         iCount++;
140     }
141     bNewPage = false;
142     iTopMargin += iHeaderHeight;
143 }
144 iCount = 0;

```

```

147         foreach (DataGridViewCell Cel in GridRow.Cells)
148         {
149             if (Cel.Value != null)
150             {
151                 e.Graphics.DrawString(Cel.Value.ToString(), Cel.InheritedStyle.Font,
152                     new SolidBrush(Cel.InheritedStyle.ForeColor),
153                     new Rectangle((int)arrColumnLefts[iCount], (float)iTopMargin,
154                         (int)arrColumnWidths[iCount], (float)iCellHeight), strFormat);
155             }
156
157             e.Graphics.DrawRectangle(Pens.Black, new Rectangle((int)arrColumnLefts[iCount],
158                 iTopMargin, (int)arrColumnWidths[iCount], iCellHeight));
159
160             iCount++;
161         }
162         iRow++;
163         iTopMargin += iCellHeight;
164     }
165 }
166
167 if (bMorePagesToPrint)
168     e.HasMorePages = true;
169 else
170     e.HasMorePages = false;
171 }
172 catch (Exception exc)
173 {
174     MessageBox.Show(exc.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
175 }
176 }
177 }

```

```

168     if (bMorePagesToPrint)
169         e.HasMorePages = true;
170     else
171         e.HasMorePages = false;
172 }
173 catch (Exception exc)
174 {
175     MessageBox.Show(exc.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
176 }
177 }
178
179 private void printDocument1_BeginPrint(object sender, System.Drawing.Printing.PrintEventArgs e)
180 {
181     try
182     {
183         strFormat = new StringFormat();
184         strFormat.Alignment = StringAlignment.Near;
185         strFormat.LineAlignment = StringAlignment.Center;
186         strFormat.Trimming = StringTrimming.EllipsisCharacter;
187
188         arrColumnLefts.Clear();
189         arrColumnWidths.Clear();
190         iCellHeight = 0;
191         iRow = 0;
192         bFirstPage = true;
193         bNewPage = true;
194
195         iTotalWidth = 0;
196         foreach (DataGridViewColumn dgvGridCol in dataGridView1.Columns)
197         {
198             iTotalWidth += dgvGridCol.Width;
199         }
200     }
201     catch (Exception ex)
202     {
203         MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
204     }
205 }

```

```

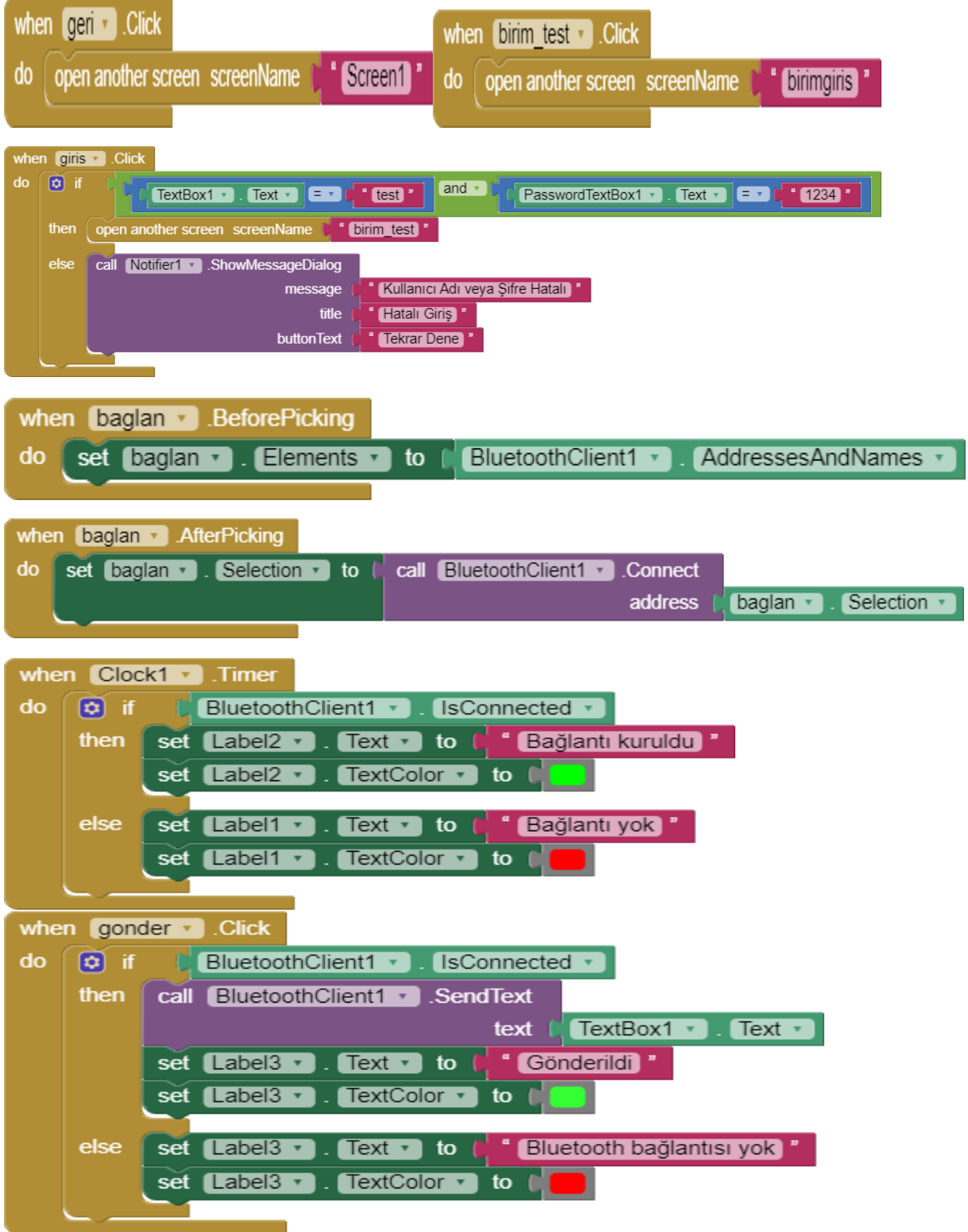
207 private void button1_Click(object sender, EventArgs e)
208 {
209     PrintPreviewDialog onizleme = new PrintPreviewDialog();
210     onizleme.Document = printDocument1;
211     onizleme.ShowDialog();
212 }
213
214 private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
215 {
216 }
217 }
218
219 }

```

## 4. Birim Testi

Yazılım programlamasında bir tasarım ve geliştirme yöntemidir.

### 4.1. Mobil Uygulama Test Kodları ve Açıklamaları



```

when x .Click
do
  set info . Visible to true
  set info . Visible to false
  set info . Visible to true

```

```

when kes .AfterPicking
do
  call BluetoothClient1 .Disconnect

```

```

when geri .Click
do
  open another screen screenName "birimgiris"

```

```

when info .Click
do
  set info . Visible to false
  set info . Visible to true
  set info . Visible to false

```

```

when s .Click
do
  call BluetoothClient1 .SendText
  text "Z"
  if BluetoothClient1 . IsConnected
  then
    if
    call BluetoothClient1 .ReceiveText
    numberOfBytes 1
    then
      set Label4 . Text to call BluetoothClient1 .ReceiveText
      numberOfBytes 6
    else
      open another screen screenName "sicaksorun"

```

```

when geri .Click
do
  open another screen screenName "birim_test"

```

Uygulama bu aşamada asıl uygulama hariç sadece birim testi için tasarlanmış 3 ayrı sayfadan oluşmaktadır. Sayfa düzeni için dikey sıralamada **VerticalArrangement**(Birbirinin altında görüntülenmesi gereken bileşenlerin yerleştirileceği bir biçimlendirme ögesi.), yatay sıralamada ise **HorizontalArrangement**(Soldan sağa görüntülenmesi gereken bileşenlerin yerleştirileceği bir biçimlendirme ögesi.) kullanılmıştır. Bu öğelerin içerisine gerekli Button, TextBox ve Label bileşenleri yerleştirilmiştir.

İlk sayfada 1 adet buton yer almaktadır. Butonlara tıklanarak '**open another screen**' komutu ile butona atanan sayfaya yönlendirme yapılır. Açılan sayfada kullanıcı, kullanıcı adını ve şifresini girer. Girilen bilgiler giriş kontrolüne bağlı olan **matematik** ve **lojik** yapılar sayesinde kayıtlı kullanıcı bilgileri ile karşılaştırılır. Bilgiler doğru ise giriş butonuna atanan sayfaya geçiş yapılır, bilgiler yanlış ise '**Kullanıcı adı veya şifre hatalı.**' uyarısı verip '**Tekrar dene**' seçeneği sunar. Bu hata mesajı **Notifier**(uyarı iletişim kutularını, iletileri ve geçici uyarıları görüntüler) ögesi ile sağlanır.

Testin ilk aşamasında **BluetoothClient.Disconnect**(bağlantı kesme) işlemi gerçekleştirilir ve **ListPicker** (kullanıcının aralarından seçim yapabileceği metinlerin bir listesini görüntüleyen bir buton) ile listelenir. Geri bağlanmak için **BluetoothClient.AddressesAndNames**(bağlantı) yapılır ve şifre girilmesi beklenir. Burada **if-then-else** komutu kullanılır şifre doğruluğuna göre yönlendirme yapılır. Label'ler yardımı ile mesaj verilir fakat butona basıldığında aktif hale gelir. Onun dışında görünmezler. Ayrıca yardımcı **info** sayesinde bağlantının nasıl olacağı hakkında bilgi verilir. Kullanıcı ev sıcaklığını görüntülemek için gerekli butona tıkladığında bağlantı olup olmadığı iç **if-then-else** ve **if-then** komutları ile kontrol yapılır. Eğer bağlantı var ise evin sıcaklık değerini gerekli LCD ekranda görüntüler, bağlantı yok ise '**bağlantı sorunu**' hatta mesajını verir. Bu aşamaya kadar testlerimiz bu şekilde yapılmıştır. İlerleyen zamanlarda projenin test aşamaları detaylandırılacaktır.

## 4.2. Arduino Test Kodları

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.IO.Ports;
7  using System.Threading;
8  namespace Arduino_Test
9  {
10     class Program
11     {
12     }
13 }
14
15 public class PortChat
16 {
17
18     static bool _continue;
19     static SerialPort _serialPort;
20
21     public static void Main()
22     {
23         string name;
24         string message;
25         StringComparer stringComparer = StringComparer.OrdinalIgnoreCase;
26         Thread readThread = new Thread(Read);
27
28         _serialPort = new SerialPort();
29
30         _serialPort.PortName = SetPortName(_serialPort.PortName);
31         _serialPort.BaudRate = SetPortBaudRate(_serialPort.BaudRate);
32
33         _serialPort.Parity = Parity.None; // Gönderilen verinin doğruluğunu kontrol etmek
34
35         _serialPort.DataBits = 8; //göndereceğimiz bilginin kaç bitten olduğu
36
37         _serialPort.ReadTimeout = 500;
38         _serialPort.WriteTimeout = 1;
39
40         _serialPort.Open();
41         _continue = true;
42         readThread.Start();
43
44         Console.Write("Name: ");
45         name = Console.ReadLine();
46
47         Console.WriteLine("Çıkış için");
48
49         while (_continue)
50         {
51             message = Console.ReadLine();
52
53             if (stringComparer.Equals("Çıkış", message))
54             {
55                 _continue = false;
56             }
57             else
58             {
59                 _serialPort.WriteLine(
60                     String.Format("<{0}>: {1}", name, message));
61             }
62         }
63
64         readThread.Join();
65         _serialPort.Close();
66     }
67 }
68
69
70
71
72
```

```

73     public static void Read()
74     {
75         while (!_continue)
76         {
77             try
78             {
79                 string message = _serialPort.ReadLine();
80                 Console.WriteLine(message);
81             }
82             catch (TimeoutException) { }
83         }
84     }
85
86     public static string SetPortName(string defaultPortName)
87     {
88         string portName;
89
90         Console.WriteLine("Available Ports:");
91         foreach (string s in SerialPort.GetPortNames())
92         {
93             Console.WriteLine("    {0}", s);
94         }
95
96         Console.Write("COM port({0}): ", defaultPortName);
97         portName = Console.ReadLine();
98
99         if (portName == "")
100         {
101             portName = defaultPortName;
102         }
103         return portName;
104     }
105

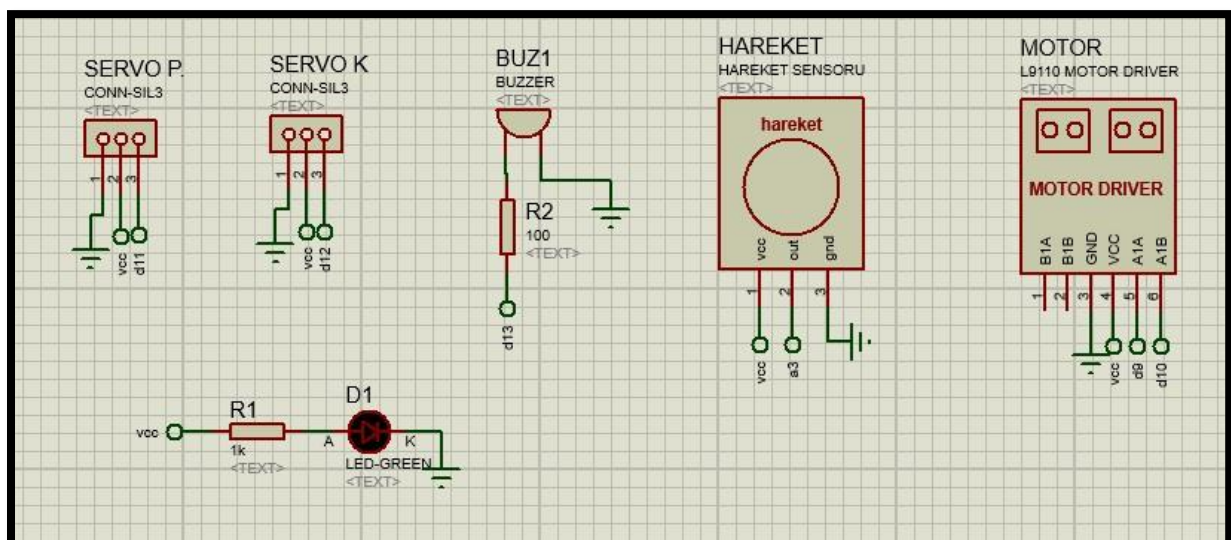
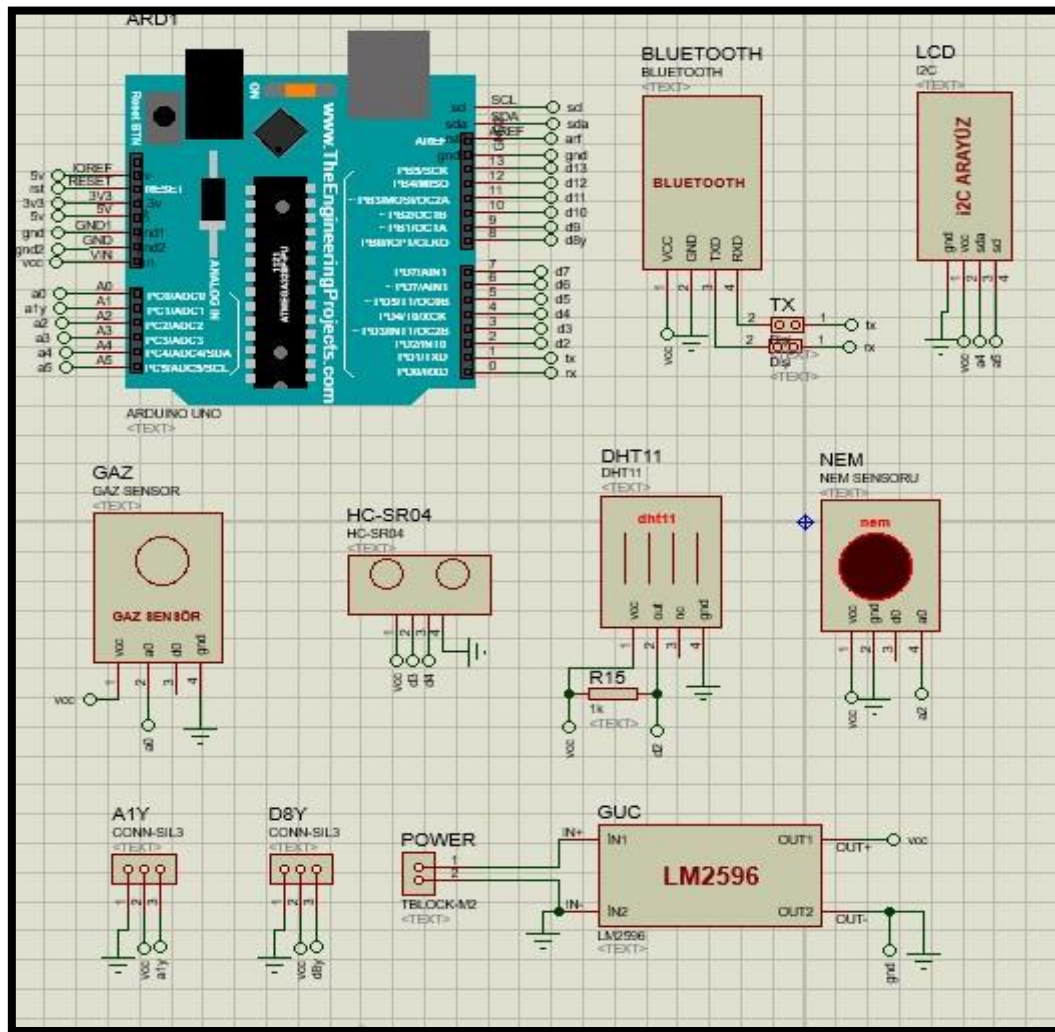
```

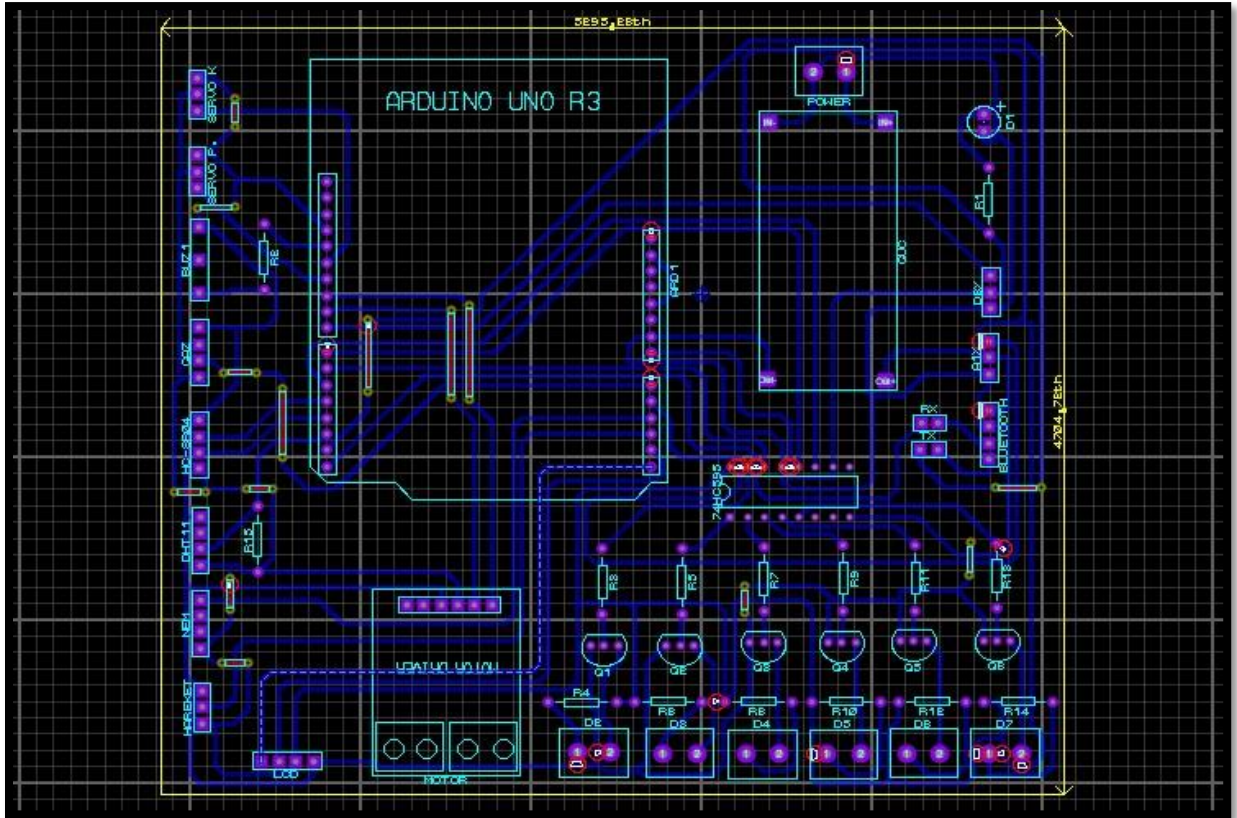
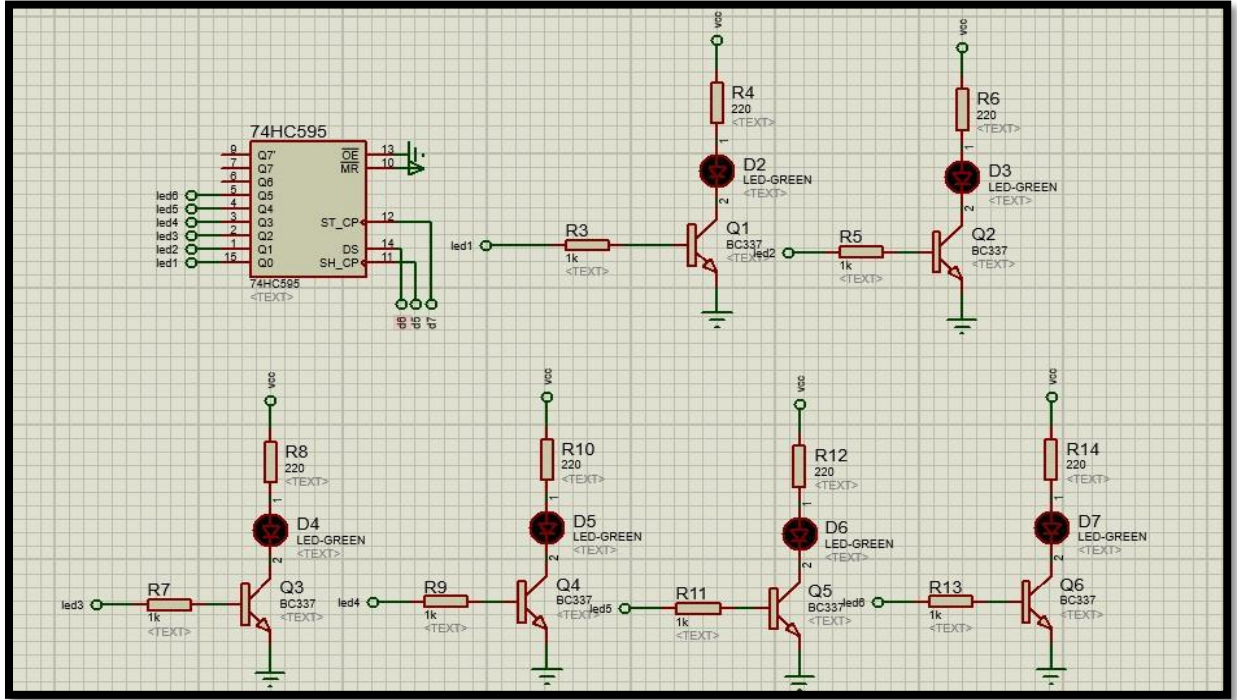
```

100     {
101         portName = defaultPortName;
102     }
103     return portName;
104 }
105
106 public static int SetPortBaudRate(int defaultPortBaudRate)
107 {
108     string baudRate;
109
110     Console.Write("Uzaktan iletişim hızı({0}): ", defaultPortBaudRate);
111     baudRate = Console.ReadLine();
112
113     if (baudRate == "")
114     {
115         baudRate = defaultPortBaudRate.ToString();
116     }
117
118     return int.Parse(baudRate);
119 }
120
121 }

```







Sistemin test aşaması için Proteus programında gerekli devre çizimi yapıldı. Bağlantı kontrolleri gerçekleştirildi. Devrede girdi ve çıktı kontrolü sağlandı.

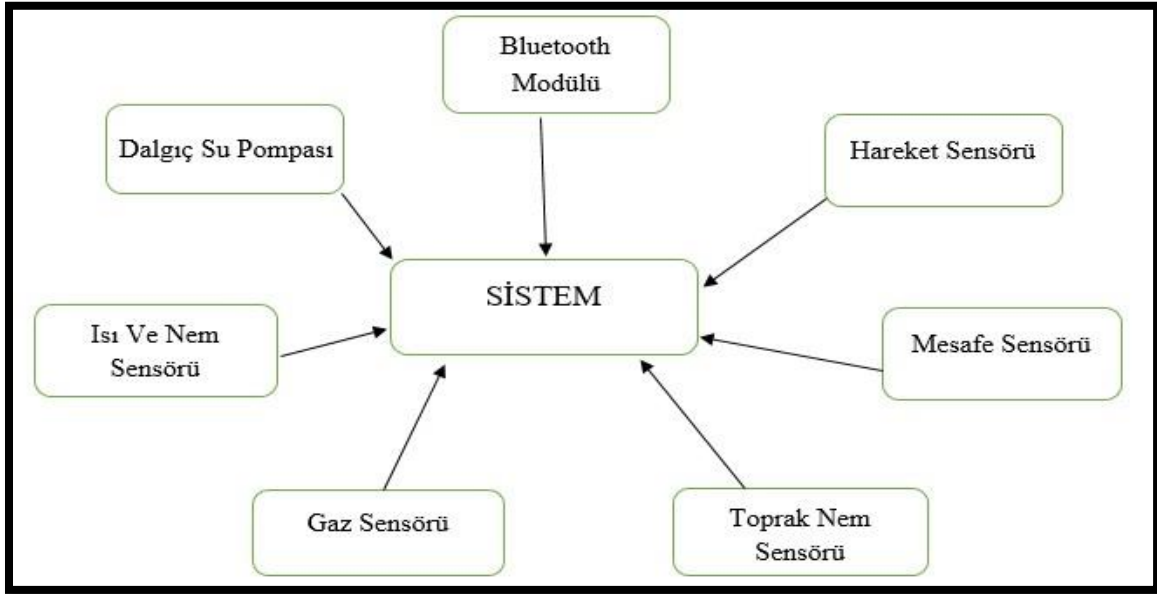
## 5. Entegrasyon Testi

Oluşturulan yazılım modüllerinin, bir araya getirilerek doğruluğunu sağlamaktır. Yazılım ürünü için oluşturulan tüm modüller bir araya getirilir ve bu şekilde test edilir. Burada ki amaç:

metotlar birim başına testten geçerken, modüller halinde bir araya geldiğinde bazı hatalara sebep oluyor olabilirler. Entegrasyon testleri ile ise bu tarz yazılım ürünü problemlerinin henüz canlı (prod) ortama çıkmadan veya geliştirdiğimiz yeni bir modülün de sorunsuz çalışabileceğinden hızlı bir şekilde emin olabilmemizi sağlamaktadır.

Entegrasyon birleştirme demektir. Entegrasyon testi bir yazılımın bileşenlerinin birbirine entegre edilmesi sırasında yapılabileceği gibi iki farklı yazılımın birbirine entegre edilmesi sırasında da yapılabilir. Bu yüzden entegrasyon testi, birim entegrasyon testi ve sistem entegrasyon testi olarak farklı test seviyelerinde yapılabilir. Yazılım geliştirme uzmanlarının birim test sırasında ayrı ayrı test ettikleri bileşenler birbirine entegre edildikleri zaman hataya sebep olabilirler. Entegrasyon testi, sistemin bu farklı bileşenlerinin(birimlerinin) birlikte doğru çalışıp çalışmadıklarını test etmeyi amaçlar.

Bu testler ile kullanıcının yaşaması muhtemel olası sorunların ortadan kalkması hedeflenir. Daha önceden testler sonucu sorunlar belirlenir ve çözülür. Bu sayede yaşanabilecek aksaklıklar en aza indirgenmiş olur. Kullanıcının tamamen kolay ve kullanışlı bir tasarım görmesini sağlamış oluruz. Ayrıca modüllerinde birbiri ile uyumu test edilir. Gerekli kod çakışmaları belirlenip çözülür. Bu modül ve sensörler tek tek rahat şekilde çalışır gibi bir arada sorunsuz çalışır hale getirilir. Bu testlerin uygulanıp eksiklerin giderilmesinin ardından yapılan işlemler kullanıcı karşına çıkmaya hazır hale gelmiş olur.



Her bir modül tek sistem üzerinde bir birinden bağımsız şekilde çalışmaktadır. Bu çalışma kullanılan sensörler ve mobil uygulama ile tetiklenip çalışmaktadır.

Sisteme gerekli sensörler teker teker aşamalı bir şekilde dahil edilip her sensörden sonra işlemin başarılı olup olmadığı test edilmiştir. Her adımda bir başka sensör eklenerek hepsinin çalışma durumu kontrol edilip onaylanmıştır.



No	Test	Fonksiyon	Prosedür	Sonuç
1	Bluetooth bağlantısı.	Mobil uygulama ile Arduino sistemi haberleşmesi.	Bağlantı adresi seçilir.	Bağlantı kuruldu. Başarılı.
2	Aydınlatma kontrolü.	Mobil uygulama kontrolünde aydınlatma açıp kapatma.	Aydınlatma butonlarına basılır.	Aydınlatma açıp kapatıldı. Başarılı.
3	Servo motor.	Kapı-perde açıp kapatma.	Kapı-perde butonlarına basılır.	Kapı-perde açıp kapatıldı. Başarılı.
4	DHT11 Isı-sıcaklık sensörü.	Isı-sıcaklık ölçümü.	Sıcaklık butonuna basılır.	Sıcaklık ölçülüp ekrana yansıtıldı. Başarılı.
5	Doğal gaz metan gazı sensörü.	Ortamdaki yabancı gaz değeri ölçümü.	Gaz değeri butonuna basılır.	Gaz değeri ölçülüp ekrana yansıtıldı. Başarılı.
6	HC-SR04 Mesafe sensörü	Cisimler arasındaki mesafe değeri ölçümü.	Mesafe değeri butonuna basılır.	Mesafe değeri ölçülüp ekrana yansıtılır. Başarılı.
7	Toprak-nem sensörü.	Toprağın nem değeri ölçümü.	Toprağın nemsiz olması durumunda sulama yapılır. Nem değeri butonuna basılır.	Nem değeri ölçülüp ekrana yansıtılır. Başarılı.
8	PIR hareket sensörü	Hareketi algılaması durumunda alarm sistemini tetikler.	Kapı zorlanarak açıldığında alarm sistemi devreye girer.	Alarm çalar. Başarılı.

## 6. Veri Toplama

Sensör kullanılarak yapılan uygulama veya deneysel çalışmalarda alınan ölçümlerin mobile aktarılması için veri işleme kartlarına ihtiyaç duyulmaktadır. Bu tür çalışmalarda kullanım kolaylığı açısından en çok tercih edilen platformlardan birisi de Arduino'dur. Açık kaynaklı bir geliştirme platformu olan Arduino'nun dijital ve analog girişleri sayesinde veriler okunup anlık olarak işlenebilmektedir. Ev otomasyon sistemi çalışmasında, mobil ortamda App Inventor kullanarak geliştirilen uygulama sayesinde Arduino'ya bağlanan farklı sensörler içerisinde istenilen sensörler seçilebilmektedir. Seçilen bu sensörleri kullanmak için gerekli Arduino kodu oluşturulabilmektedir. Arduino üzerindeki sensörlerden gelen veriler belirli zaman aralığında

veya anlık olarak kaydedilebilmektedir. Kaydedilen veriler, istenilen periyotlarda ve özelliklerde elde edilebilmekte ve kullanılabilir.

Sensörler, fiziksel veya kimyasal büyüklükleri elektriksel büyüklüklere çevirerek kullanılabilir formata dönüştürmektedirler. Sensörler kablosuz iletişim, sinyal algılama, alınan sinyali işleme ve yayma gibi yeteneklere sahiptir. Sensörler kullanılarak yapılan uygulama veya deneysel çalışmalarda alınan ölçümlerin bilgisayara aktarılması için veri toplama kartlarına ihtiyaç duyulmaktadır.

```
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;

namespace _veri
{
    public partial class Form1 : Form
    {
        private SerialPort _port; // Seri port nesnesi oluşturuyoruz

        private DateTime datetime;
        private string in_data;
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            _port = new SerialPort();
            _port.BaudRate = 9600; //Kullanacağımız seri port hızını
            _port.PortName = textBox1.Text; //Kullanacağımız seri port adını
            _port.Parity = Parity.None; // Gönderilen verinin doğruluğunu kontrol etmek
            _port.DataBits = 8; //göndereceğimiz bilginin kaç bitten olduğu
            _port.StopBits = StopBits.One; //Stop bitinin kaç bit olacağını belirtir.
```

```

port.DataBits = 8; //göndereceğimiz bilginin kaç bitten olduğu
Project: _veri (Ctrl+F2) e; //Stop bitinin kaç bit olacağını belirtir.
Use the dropdown to view and switch to other projects this file may belong to. aTecevied;

try
{
    _port.Open();
    textBox2.Text = "";
    // listBox1.Text = "";
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Hata Oluştı");
}
}

void my_DataTecevied(object sender, SerialDataReceivedEventArgs e)
{
    in_data = _port.ReadLine();
    //Veriler alınır

    this.Invoke(new EventHandler(displaydata_event));
}

private void displaydata_event(object sender, EventArgs e)
{
    //Gelener verileri güncel zaman ile ekle
    datetime = DateTime.Now;
    string time = datetime.Hour + ":" + datetime.Minute + ":" + datetime.Second;

    textBox2.AppendText(time + "\t\t\t" + in_data + "\n");
}

```

```
private void button3_Click(object sender, EventArgs e)
{
    try
    {
        string pathfile = @"C:\Users\USER\Desktop";
        string filename = "veri.txt";
        System.IO.File.WriteAllText(pathfile + filename, textBox2.Text);
        MessageBox.Show("Kayıt Edildi " + pathfile, "Bilgi");
    }
    catch (Exception ex3)
    {
        MessageBox.Show(ex3.Message, "Hata Oluştı");
    }
}

private void button2_Click(object sender, EventArgs e)
{
    _port.Close(); //seri portu kapat
}
}
```

## 6.1. Verilerin Toplanması ve Kontrol Edilmesi

### *Hareket Sensörü*

Projemizde hareket sensörü kullanarak herhangi bir harekette sensör durumu algılayacak. Sensör sistemi uyararak alarm devreye girecektir. Bu sensör oldukça kullanışlıdır. Artık evler bu sensör sayesinde daha güvenlidir.

Genel olarak akşam-gece saatlerinde hırsızlık olaylarının daha yoğun olduğu zamanlarda sistem tarafından veriler alınmakta olup daha çok güvenliği ön planda tutmak amaçlanmaktadır.

### *Işıklar*

Evdeki ışıklar, kullanıcı isteği ile etkinleşir. Kullanıcı evden uzakta olsa bile ışıklara müdahale edip kapatıp açabilir.

Veriler, gündüz güneşli saatlerde ışıkların kapalı, akşam saatlerinde ışıkların açık olduğu şeklinde alınmaktadır. Aynı zamanda evden uzaktayken veya tatildeyken evin ışıklarını kontrol edip, hırsız girmemesi için akşam saatlerinde evde biri varmış hissi uyandırmak için kullanılır. Sabahleyin tekrardan ışıklar kapatılır.

### *Alarm*

Zorlama ile eve giriş sağlandığı anda etkinleşir. Hareket sensörüyle birlikte çalışır. Genel olarak hırsızlık olaylarının yoğun olduğu zamanlarda çalışır. Eğer sistem bir hareket algılasa buzzer devreye girer ve alarm çalar.

### ***Mesafe Sensörü***

Park işlemlerini kolaylaştırır. Burada minimum mesafe 10cm olarak ayarladığı zaman arabanın kolay bir şekilde park edilmesini sağlamaktadır.

Veriler genellikle, akşam iş dönüşünde alınmaktadır. Çünkü genel olarak akşam arabanın park edilmesi üzerine veriler akşam saatlerinde çok alınır. ***Sulama***

Toprağın kurduğunu sistem anladığı zaman su vermektedir.

Veriler toprağın kuruluşuna bakıp alınmaktadır. Eğer topraktaki su miktarı az ise sistem tarafından toprağa su verilmektedir. Genel olarak tatil modu durumunda iken evdeki çiçekler sulanmaktadır.

### ***Perde Sistemi***

Bu sistem ev ortamında oldukça işe yarayan bir sistemdir. Kullanıcı, perdeyi mobil uygulama ile tek bir tuşla istediği saatlerde açar veya kapatır. Genel olarak perde sisteminden alınan veriler şu yöndedir; sabah saatlerinde perde açık, akşam saatlerinde perde kapalı durumda olduğu bilgisi alınmaktadır.

### ***Isı-Nem Sensörü***

Bu sensör sayesinde çevrenin ısı-nemi algılanır ve ona göre sisteme veri gönderir ve veri tabanında güncellenmiş olur.