



ELMADAĞ MESLEK YÜKSEK OKULU

Sistem Analizi ve Tasarımı II



NÖ 2-B Sistem Analizi ve Tasarımı II

MURATCAN ÇOBAN 18330014

DOĞAN AKCAN 18330004

İçindekiler

1. Gün Lambası.....	3
1.2. Hangi Probleme Çare Olacak?.....	3
1.3. Proje Dili	3
1.4. Proje Ortakları Arasında Yapılan Görev Paylaşımı	3
1.5. Yazılım Geliştirme Süreci	3
1.6. Proje Aşaması.....	3
2. LDR Nedir?.....	4
Şekil 1. Seri İletişim	4
3. Ekranlar	5
Şekil 2. Ana Ekran	5
3.1. Ekran Çalışırken Neler Olacak?.....	5
Şekil 3. Çalışma Aşaması	5
3.2. Port Seçilmezse:	6
Şekil 4. Hata.....	6
Şekil 5.Veritabanı	6
4. Arduino.....	7
4.1. Ardunio İno	7
Şekil 6. Ardunio Tanımlamalar	7
Şekil 7. Hoşgeldiniz Ekranı.....	7
Şekil 8. Sıcaklık Ve Nem.....	8
Şekil 9. Ldr ve Led.....	8
Şekil 10. Yüksek Işık Değeri	9
5. Ardunio Çalışması.....	9
Şekil 11. Hava Aydınlik	9
Şekil 12. Hava Karanlık	10
Şekil 13. Sıcaklık ve Nem	10
6. Projenin Kattıkları.....	10

1. Gün Lambası

1.1. Projenin Tanımı

Ardunio ve C# kullanarak LDR' den gelen veri ile kaydetme ve okuma yapılması. Üzerinde bulunan ekranda sıcaklık ve nem bilgisinin ardından ldr' den gelen verinin gösterilmesi ve ledin yanma veya sönme işlemini takip etmesi.

1.2. Hangi Probleme Çare Olacak?

Gün Lambası, yoğun iş yükü ve piyasa pahalılığından yakının seracılarımızın iş yükünü olabildiğince azaltan ve seralarından uzakta olsalar bile anlık olarak bilgi sahibi olabilecekleri bir Ardunio “elektronik beyin” sistemi yapmış bulunmaktayız.

Sera bilgilerini Ardunio üzerinde bulunan lcd ekran ve visual studio üzerinde geliştirdiğimiz program üzerinde görüntülenecektir.

1.3. Proje Dili

Proje kararı sonrası uygulanacak yazılım geliştirme teknik ve yöntemlerine karar verilmiştir. Yazılım geliştirme süreci Visual Studio ve Ardunio programları kullanılacaktır ve C# programlama dili kullanarak yazılmasına karar verilmiştir.

Proje Ardunio'dan gelen verileri istedeğe bağlı olarak metin belgesi halinde kaydedebiliyor.

1.4. Proje Ortakları Arasında Yapılan Görev Paylaşımı

Doğan Akcan: Muratcan Çoban ile Visual Studio' da Gün Lambası programın kodlarını ve tasarımı birlikte yapmışlardır.

Muratcan Çoban: Ardunio bağlantılarını ve kodlarını yapmıştır. Ayrıca Doğan Akcan ile Visual Studio' da programın kodlarını birlikte yapmışlardır.

1.5 Yazılım Geliştirme Süreci

- ❖ Planlanan Gün Lambası projesine ait iş planı ve geliştirme süreci aşağıdaki gibidir.
 - Projenin Belirlenmesi (1. Hafta)
- ❖ Problemin ayrıntılı olarak belirlenmesi
 - Proje ortaklarının katılımının planlanması
- ❖ Programın Tasarlanması (2. Hafta)
 - Visual Studio'da geliştirilen programın ekranının tasarlanması
 - Ardunio bağlantılarının yapılması
- ❖ Programın Kodlanması (1Ay)
 - Ardunio'nun Kodlanması
- ❖ Programın Test edilmesi (7.Hafta)

1.6. Proje Aşaması

Proje Tamamlanmıştır.

2. LDR Nedir?

LDR (Light Dependent Resistor), Türkçede “Işığa Bağımlı Direnç” anlamına gelmektedir. Bir diğer adı da foto dirençtir. LDR her ne kadar bir direnç çeşidi olsa da aynı zamanda pasif bir sensördür. LDR’ler bulundukları devrelerde değişen direnç değerleri ile bir çıkış sağlarlar fakat bu işlemi dış ortamdan aldıkları fiziksel bir değişim ile gerçekleştirdiklerinden dolayı bir sensör görevi görmüş olurlar.

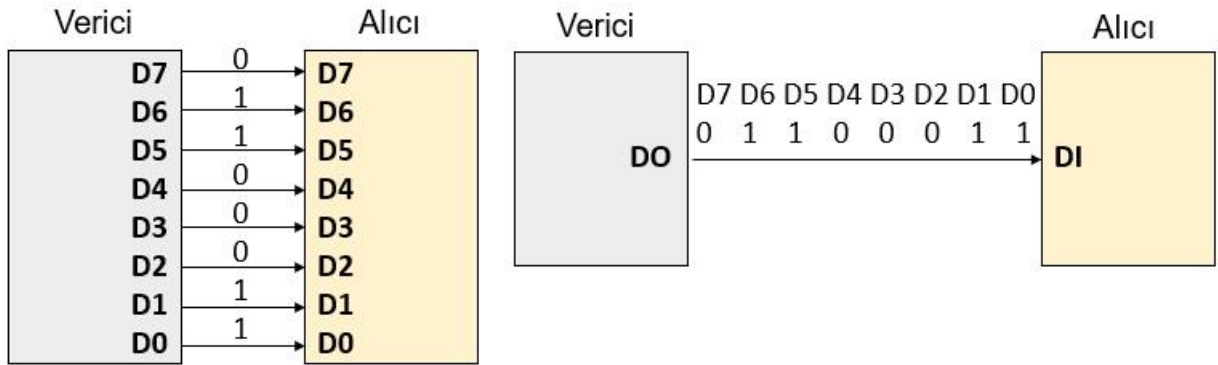
2.1.LDR Çalışma Prensibi : Üzerine düşen ışık şiddeti ile ters orantılı bir çalışma prensibine sahiptir. Yani üzerine düşen ışık şiddeti arttıkça sahip olduğu direnç değeri azalır, ışık şiddeti azaldıkça sahip olduğu direnç değeri artar. LDR’ler sahip oldukları direnç değerlerinin değişmesi ile bir anahtarlama görevi görürler. Başka bir açıdan bakacak olursak bir optik sensör görevi de görmüş olurlar.

2.3.LDR Kullanım Alanları – LDR Kontrollü Devreler Nerelede Kullanılır?

LDR günlük hayatımızda en yaygın olarak aydınlatma sistemlerinde kullanılmaktadır. Sokak lambaları, gece lambaları, kumanda sistemleri günlük kullanımlarımıza örnek olarak verilebilirler.

2.4.C# İle Seri İletişim Nedir?

En basit tanımıyla seri iletişim dijital bilginin tek bir hat üzerinden ardı sıra taşınmasıdır, yani seri iletişimde bir kerede sadece bir bit iletilir. Paralel iletişimde ise tüm bitler aynı anda transfer edilir. Bu yüzden seri iletişimi tek bir kablo hattı ile sağlayabilirken paralel iletişimde her bit için ayrı bir kablo hattı gerekir. Bunun sonucu olarak da seri iletişim, maliyeti bakımından daha avantajlıdır.



Şekil 1. Seri İletişim

Seri iletişimde alıcı doğru haberleşme sağlamak için iletim hızını, karakter uzunluğunu ve başlangıç bitiş bitlerini bilmek zorundadır.

Arduino’da seri iletişim, üzerindeki seri iletişim birimi (UART) vasıtasıyla sağlanır. Arduino’ya her kod yükleyişimizde seri iletişimi en az bir kere kullanmış oluruz. Bilgisayarla devamlı iletişim sağlamak için de `Serial.begin(9600)` komutuyla seri iletişimi başlatırız; `Serial.print()`, `Serial.println()` komutları ile bilgisayarımıza veri göndeririz. `Serial.read()` komutu ile de bilgisayarımızdan veri okuruz.

3. Ekranlar

The screenshot shows a Windows form titled 'Form1'. It contains a 'Port' dropdown menu, four buttons labeled 'Basla', 'Durdur', 'Sıfırla', and 'Kaydet', a large empty rectangular area for data display, and two radio buttons labeled 'Led Açık!' and 'Led Kapalı!'.

Şekil 2. Ana Ekran

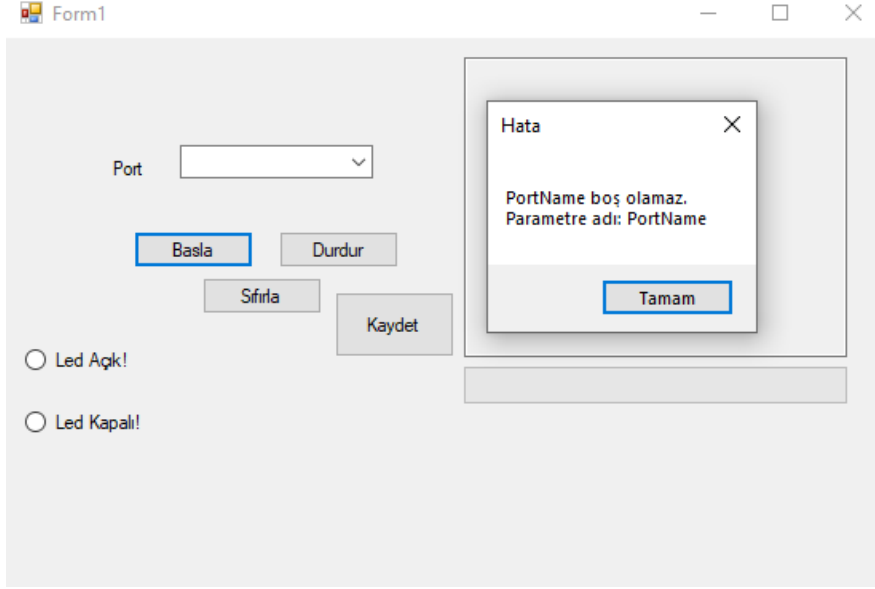
3.1. Ekran Çalışırken Neler Olacak?

- Port seçimini yapıp başla düğmesine basıldığı andan itibaren Ldr'den gelen veriler ekranda tarih saat ve değer olarak yazılmaya başlayacaktır.

The screenshot shows the same form as in Şekil 2, but now it is in operation. The 'Port' dropdown is set to 'COM4'. The 'Basla' button is highlighted. The data display area shows two lines of text: '30.04.2020 04:23:24 4' and '30.04.2020 04:23:39 6'. The 'Led Açık!' radio button is selected. Below the data display area, a label reads 'Işık Değeri = 6'.

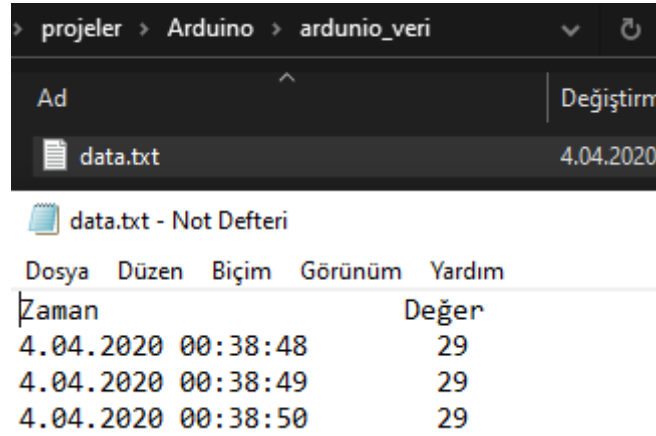
Şekil 3. Çalışma Aşaması

3.2. Port Seçilmezse: Görseldeki hata ekranda belirecektir.



Şekil 4. Hata

- Durdur düğmesine basıldığında Arduino'ya yüklenen kodlar çalışmayı durduracaktır.
 - Sıfırla düğmesine basıldığında gelen veriler ekrandan silinecektir.
- Kaydet düğmesine basıldığında Arduino' dan gelen veriler daha önceden ayarlanmış olan klasöre, metin belgesi olarak kaydedilecektir.



Şekil 5.Veritabanı

- Ldr'den gelen veriler aynı zamanda Progressbar'da gösterilecektir.
- Ldr'den gelen veriler belirlenen ışık değerine göre Checkbox'da led açık veya led kapalı düğmelerini aktif hale getirecektir.

4. Arduino

4.1. Ardunio İno

Ardunio’da kullanılacak olan bütün malzemelerin bağlantıları yapıldıktan sonra programda belirtiyoruz.

```
sistem_analizi
#include <Wire.h>
#include <FaBoLCD_PCF8574.h>
#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN 2
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);
FaBoLCD_PCF8574 lcd;

int buzzerPin = 13;
int ldr = 0;

void setup() {
    // put your setup code here, to run once:
    pinMode(11, OUTPUT);
    pinMode(9, OUTPUT);
    pinMode(10, OUTPUT);
    pinMode(buzzerPin, OUTPUT);
    Serial.begin(9600);
```

Şekil 6. Ardunio Tanımlamalar

- Arduino üzerinde bulunan ekranda Hoşgeldiniz yazısı bizleri karşılarken ses çıkaracaktır.

```
lcd.begin(16, 2);
lcd.print("Hoşgeldiniz!");
delay(1000);
digitalWrite(buzzerPin, HIGH);
delay(1000);
digitalWrite(buzzerPin, LOW);
dht.begin();
```

Şekil 7. Hoşgeldiniz Ekranı

- Sonra sıcaklık ve nemi ölçerek ekranda gösterecektir.

```
void loop() {
    int temp = dht.readTemperature();
    int hum = dht.readHumidity();
    lcd.clear();

    lcd.setCursor(0, 0);
    lcd.print("Sicaklik: ");
    lcd.print(temp);
    lcd.print(" C");

    lcd.setCursor(0, 1);
    lcd.print("Nem: %");
    lcd.print(hum);
    delay(5000);
    lcd.clear();
}
```

Şekil 8. Sıcaklık Ve Nem

- Ardından Ldr'den gelen Işık değerini ölçükten sonra gelen veri belirtilen ışık değerinden az ise bir kere ses çıkardıktan sonra ledleri yakacaktır.

```
// put your main code here, to run
int deger = analogRead(ldr);
Serial.println(deger);
delay(2000);

lcd.setCursor(0, 0);
lcd.print("Isik Degeri: ");
lcd.print(deger);
delay(5000);
lcd.clear();

if (deger < 15)
{
    lcd.setCursor(0, 0);
    lcd.print("Hava : Karanlik");
    lcd.setCursor(0, 1);
    lcd.print("Ledler Acildi! ");
    delay(1000);
    digitalWrite(buzzerPin, HIGH);
    delay(1000);
    digitalWrite(buzzerPin, LOW);

    digitalWrite(11, HIGH);
    digitalWrite(9, LOW);
    digitalWrite(10, LOW);
    delay(200);
}
```

Şekil 9. Ldr ve Led

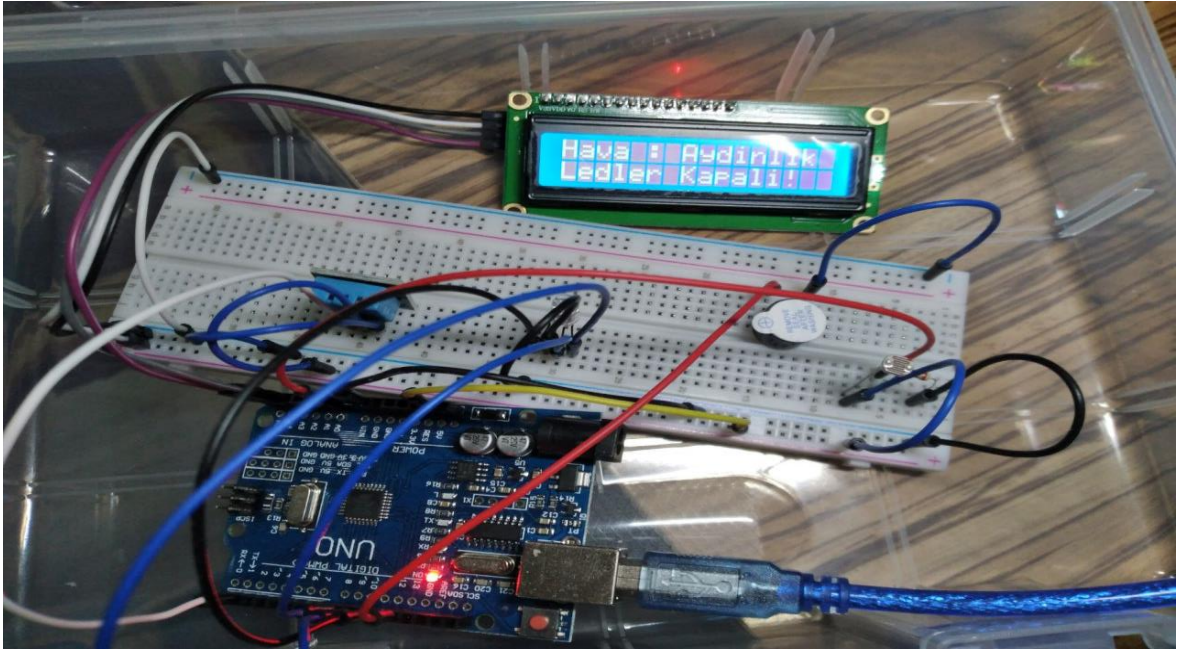
- Eğer belirtilen ışık değerinden yüksekse ledleri söndürecektir.

```
else if (deger > 20 )
{
  lcd.setCursor(0, 0);
  lcd.print("Hava : Aydinlik");
  lcd.setCursor(0, 1);
  lcd.print("Ledler Kapali! ");
  delay(2000);

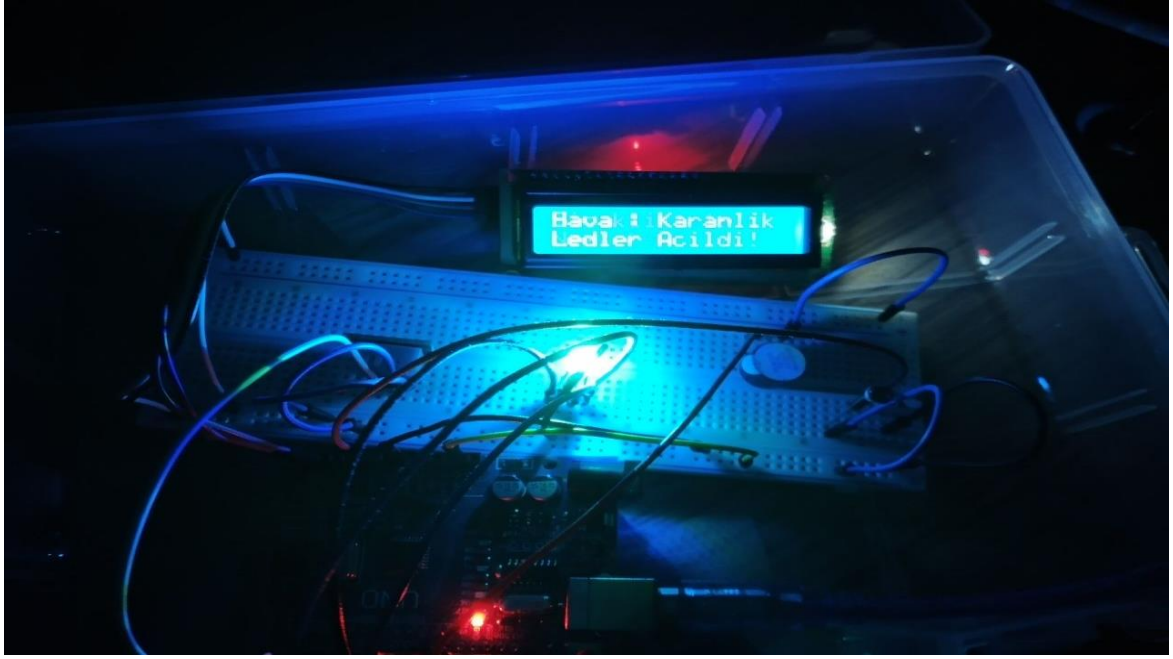
  digitalWrite(11, LOW);
  digitalWrite(9, LOW);
  digitalWrite(10, LOW);
}
```

Şekil 10. Yüksek Işık Değeri

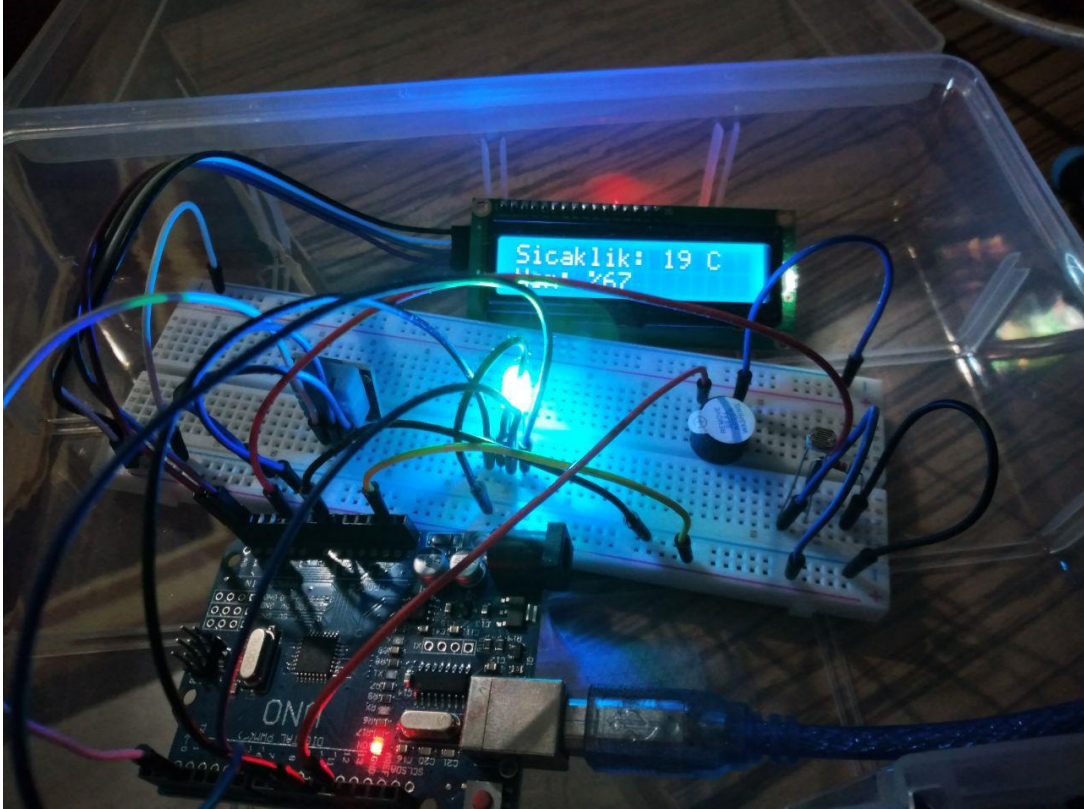
5. Ardunio Çalışması



Şekil 11. Hava Aydınlik



Şekil 12. Hava Karanlık



Şekil 13. Sıcaklık ve Nem

6. Projenin Kattıkları

Arduino adı gibi, yapacağımız projelerde en çok güveneceğimiz bileşen olacak. Arduino'ya bağlayacağınız bileşenlerle ona hemen hemen her şeyi yaptırabilirsiniz. Biz de Akıllı Sera projemizi hayata geçirdiğimizden oldukça mutluyuz. Ve projemizi geliştirmeye devam edeceğiz. Bir sonraki hedefimiz Gün Lambası mobil uygulamamızı geliştirerek uzaktan kontrollü bir sistem haline getirmektir.