



**HİTİT**  
ÜNİVERSİTESİ

Mühendislik Fakültesi  
Bilgisayar Mühendisliği

2023-2024 Bahar Dönemi

**BIL 306 – Mikrodenetleyiciler Laboratuvarı**

# Uygulama Ödevi

(Ağırlık %20)

## Mikrodenetleyici Tabanlı Sistem Tasarımı

	1. Öğrenci	2. Öğrenci
Okul No	214210043	
Ad	Melisa İrem	
Soyad	Çık	
Ödev Linki	<a href="https://www.youtube.com/watch?v=gWP80YdLFGA">https://www.youtube.com/watch?v=gWP80YdLFGA</a>	

Değerlendirme	
Tarih	Puan

Dr. Öğr. Üyesi Serkan DİŞLİTAŞ  
Ders Sorumlusu

2024  
Çorum / Türkiye

## Uygulama Ödevi (%20)

### Mikrodnetleyici Tabanlı SistemTasarımı

Uygulama ödevi kapsamında her öğrenci mikrodnetleyici kart ile birlikte farklı ekran, sensör, LED, motor, buzzer, buton vb. elektronik devre elemanları kullanarak mikrodnetleyici tabanlı sistem tasarımı yapacaktır.

### Değerlendirme

Uygulama ödevi, ders başarısı değerlendirmesinde %20 not ağırlığına sahip olacaktır.

### Teslimat Bilgileri

Uygulama ödevinin aşağıdaki hususlara uygun olarak teslim edilmesi gerekmektedir:

- 1) UBYS – Ödev sekmesi üzerinden Uygulama Ödevi dosyaları (\*.docx) temin edilecektir.
- 2) Uygulama ödevinde sistem tasarımına yönelik yazılım ve donanım tabanlı bilgiler verilecektir. Bu kapsamda ödevde tasarlanan sistemin fotoğraflarına, devre şemalarına, kullanılan devre elemanlarına, geliştirilen programlara, sistemin çalışmasına ve simülasyonlara yer verilecektir.
- 3) Ödev için tanıtım videosu çekilecek (Gurup varsa tüm öğrencilerin katılımı sağlanacaktır) ve Youtube platformunda yayınlanacaktır. İlgili link ödev dosyası kapağına iliştilirilecektir.
- 4) Ödev dosyası istenen özelliklere uygun olarak doldurulacaktır.
- 5) UBYS – Ödev sekmesinde belirtilen teslim tarihi ve saatine kadar tekrardan (\*.docx) dosya formatında sisteme yüklenecektir.
- 6) Sistemde her öğrenci için 1 defa yükleme imkânı vardır. Gurup öğrencilerinden birinin ödevi sisteme yüklemesi yeterlidir.

## DİKKAT

Uygulama Ödevine yönelik anlatımlar sonraki sayfadan itibaren yapılacaktır.

## İçindekiler

1. Giriş.....	4
1.1. Projenin Amacı.....	4
1.2. Projede Kullanılan Malzemeler.....	4
2. Yapım Aşaması.....	5
2.1. Devre Bağlantılarının Yapılması.....	5
2.2. Arduino Yazılımının Yazılması.....	6
2.3. Bluetooth Programının Ayarlanması.....	6
2.4. Bluetooth Programının Ayarlanması Görselleri.....	7
3. Kod Kısmı.....	10
3.1. Arduino Kodun Görselleri.....	10
3.2. Arduino Kodun Açıklaması.....	11
4. Görseller.....	13
4.1. Proteus Görseli.....	13
4.2. Devre Görseli.....	13

# 1.Giriş

## 1.1.Projenin Amacı

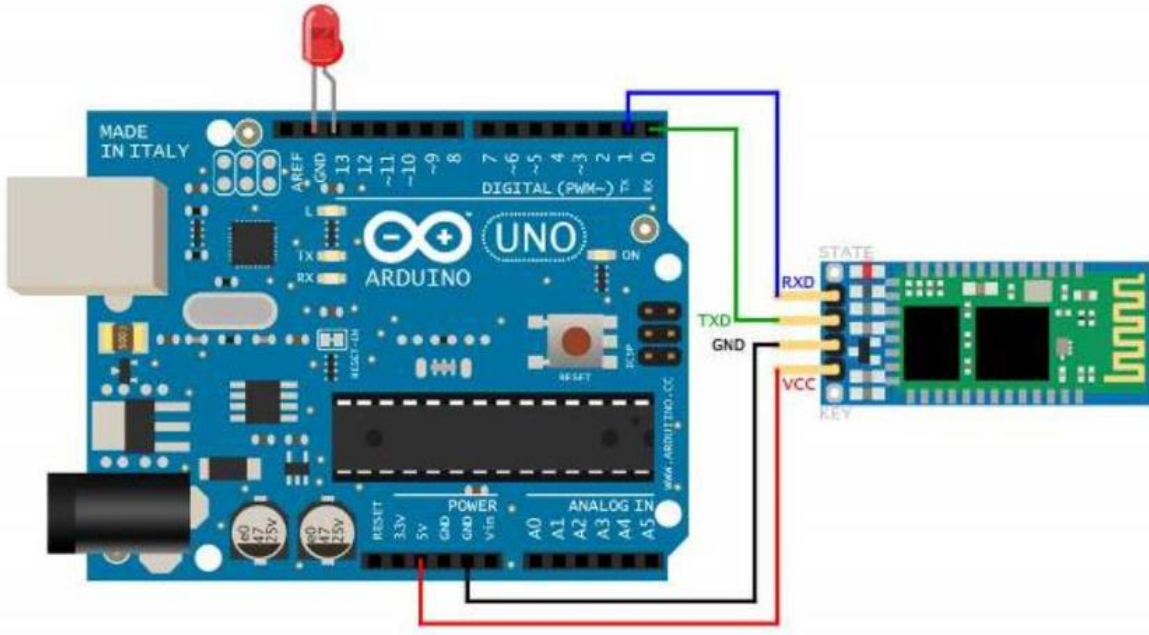
Bluetooth teknolojisi kullanılarak LED lambaların kontrol edilmesi, kullanıcılara bu lambaları kolayca açma ve kapatma imkanı sağlamaktadır. Bluetooth üzerinden gerçekleştirilen kablosuz bağlantı sayesinde, kullanıcılar LED lambaları uzaktan kontrol edebilmekte ve istedikleri ayarları yapabilmektedirler. Bu teknoloji, kullanıcıların lambaların parlaklığını ve renklerini de uzaktan ayarlayabilmelerine olanak tanımakta, böylece kullanım kolaylığı ve esneklik sunmaktadır. Bu sistem, hem ev otomasyonu hem de enerji verimliliği açısından önemli avantajlar sağlamaktadır.

## 1.2.Projede Kullanılan Malzemeler

- **LED Lambalar:** Düşük enerji tüketimi, uzun ömür ve geniş renk seçenekleri gibi avantajlarıyla dikkat çeken aydınlatma elemanlarıdır. LED lambalar, yüksek verimlilikleri ve dayanıklılıkları nedeniyle modern aydınlatma çözümlerinde yaygın olarak kullanılmaktadır.
- **Bluetooth modülü:** Bluetooth modülü, Bluetooth kablosuz iletişimini sağlayan bir donanım bileşenidir. Bu modül, mikrodenetleyici veya diğer kontrol devreleriyle etkileşim kurarak, LED lambalarına kablosuz olarak komut iletimini gerçekleştirir. Böylece, uzaktan kontrol işlemlerini mümkün kılar.
- **Mikrodenetleyici:** Mikrodenetleyici, uygulamanın genel kontrolünü sağlayan entegre bir bilgisayar çipidir. Bluetooth modülü ile haberleşerek LED lambalarını kontrol eden komutları oluşturur. Ayrıca, kullanıcı ile etkileşim sağlamak amacıyla bir kullanıcı arabirimi sunabilir. Mikrodenetleyici, sistemin merkezi kontrol ünitesi olarak işlev görmektedir.
- **Kablo ve Bağlantı Elemanları:** Mikrodenetleyici, Bluetooth modülü ve LED lambaları arasındaki bağlantıları sağlamak için kablolar ve bağlantı elemanları kullanılır.

## 2.Yapım Aşaması

### 2.1.Devre Bağlantılarının Yapılması



LED'in uzun bacağı olan anot (+) ucu, Arduino'nun 13. giriş pinine, kısa bacağı ise GND'ye bağlanmıştır. Bu bağlantı, LED'in doğru polarite ile çalışmasını sağlamaktadır.

Bluetooth modülü ile Arduino arasındaki bağlantılar şu şekilde gerçekleştirilmiştir: Modülün RXD pini Arduino'nun TX pinine, TXD pini Arduino'nun RX pinine, GND pini Arduino'nun GND pinine ve Vcc pini ise Arduino'nun 5V pinine bağlanmıştır. Bu bağlantılar, veri iletişiminin doğru bir şekilde gerçekleşmesini ve modülün gerekli gücü almasını sağlamaktadır.

## 2.Yapım Aşaması

### 2.2.Arduino Yazılımının Yazılması

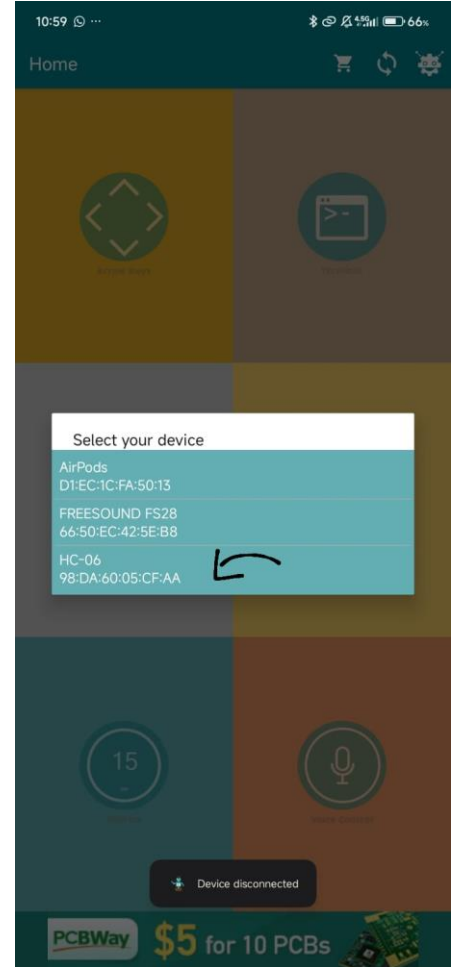
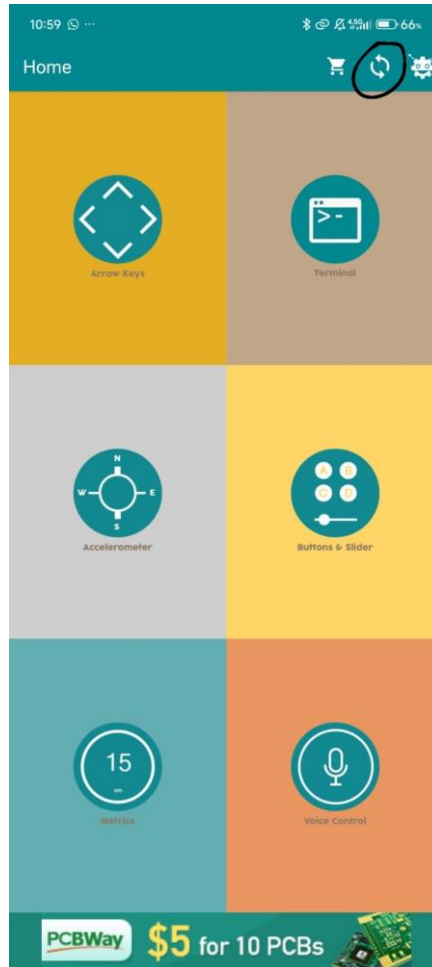
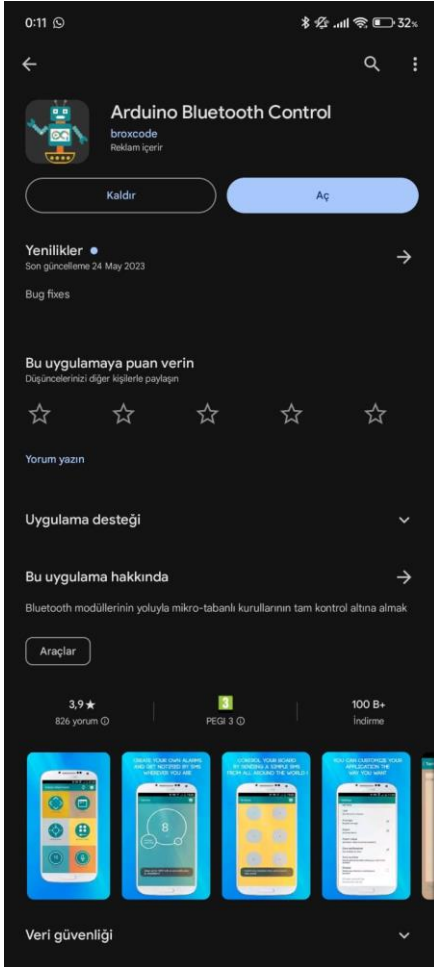
- Arduino IDE ortamı açılarak, yazılım geliştirme süreci başlatılır.
- İlgili proje için gerekli kod yazılır ve doğruluğu kontrol edilir.
- "Araçlar" menüsü altında, kullanılan Arduino kartı seçilir. Ayrıca, "Port" menüsünden Arduino'nun bağlı olduğu port (örneğin, COM5) seçilerek doğru bağlantı noktası belirlenir.
- Tüm ayarlar yapıldıktan sonra, yazılım Arduino'ya yüklenir. Bu işlem, yazılımın kart üzerindeki mikrodenetleyiciye aktarılmasını ve çalıştırılmasını sağlar.

### 2.3.Bluetooth Programının Ayarlanması

- Arduino BlueControl programı google playdan indiriniz.
- Sağ üst menüden yenilenme ikonuna tıklayıp bluetooth modülünü (HC-06) seçiniz.
- Ana menüden Buttons & Slider içerisine giriniz.
- Sağ üstten ayarlara girin, Command Buttons Configuration seçeneğine tıklayın.
- Button A seçeneğine tıklayıp 1 değerini giriniz.
- Button B seçeneğine tıklayıp 0 değerini giriniz.
- Button C seçeneğine tıklayıp 2 değerini giriniz.
- Daha sonra 2 kez geriye gelip butonların olduğu menüye gelin.
- 1(A) butonuna bastığınız zaman lamba yanacaktır, 0(B) butonuna bastığınız zaman lamba sönecektir.
- 2(C) butonuna bastığınız zaman lamba belirli aralıklar ile yanıp sönecektir.

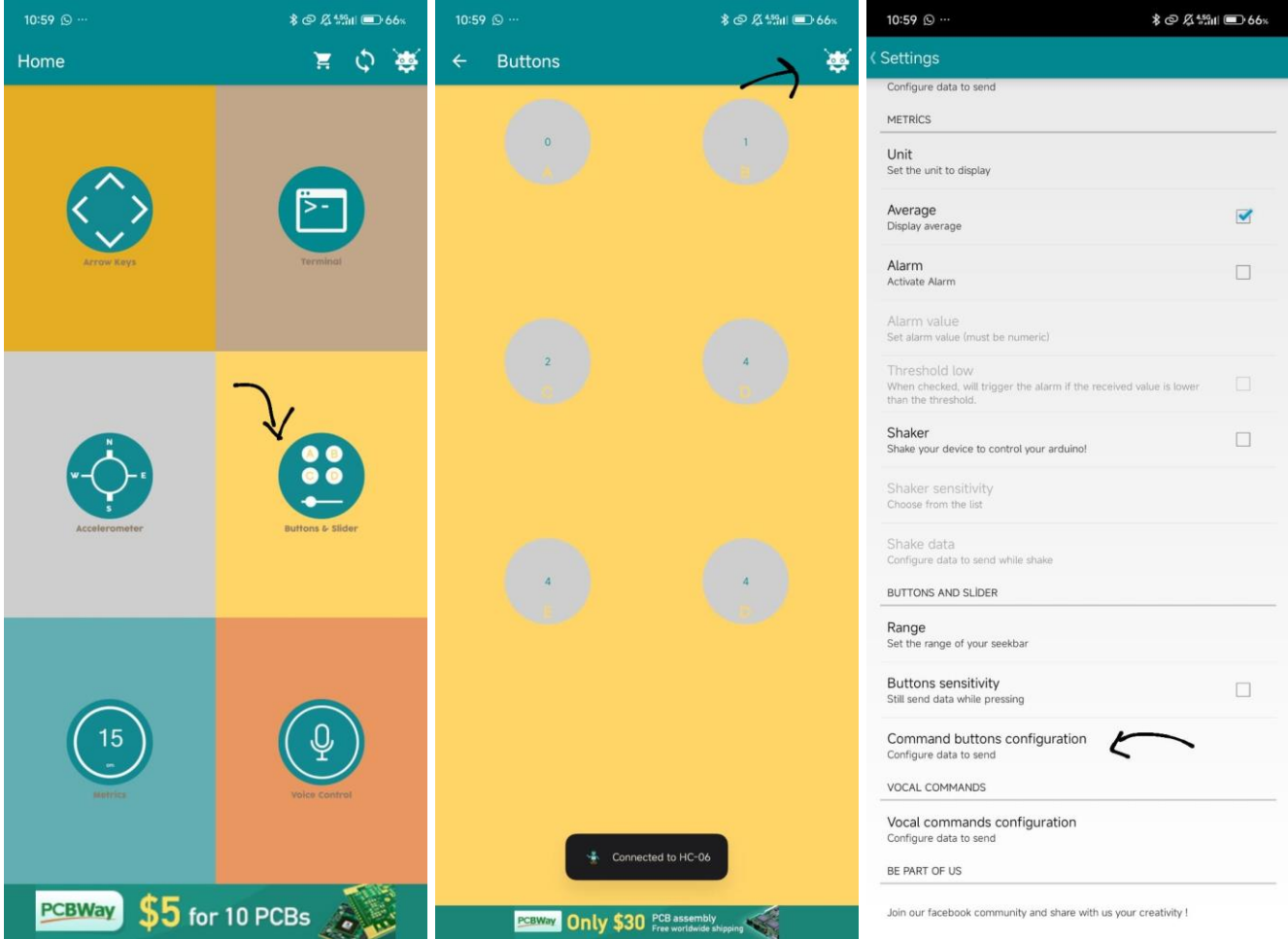
## 2.Yapım Aşaması

### 2.4.Bluetooth Programının Ayarlanması Görselleri



## 2.Yapım Aşaması

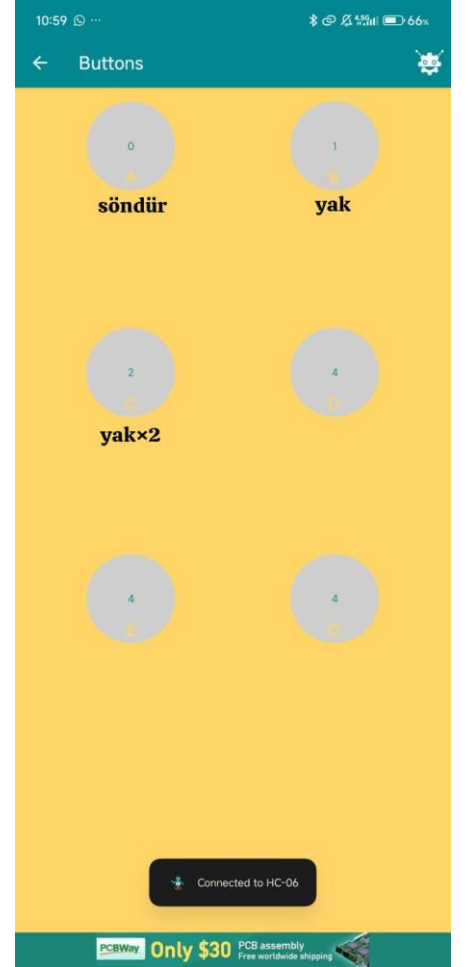
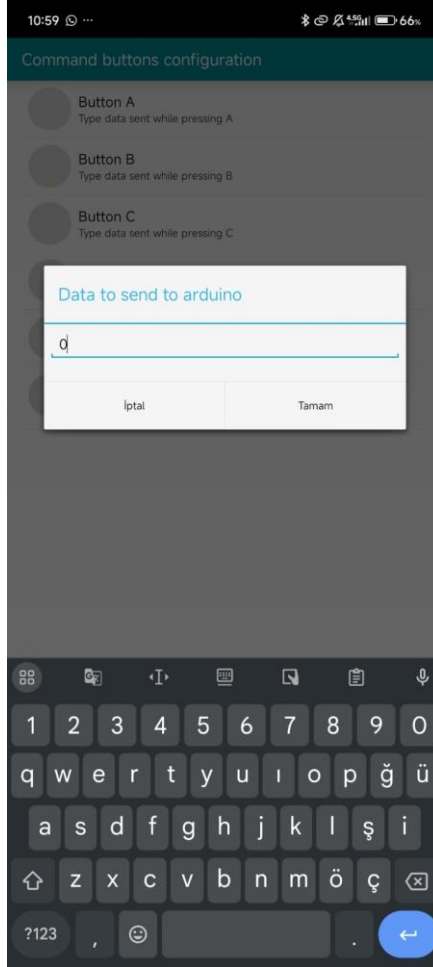
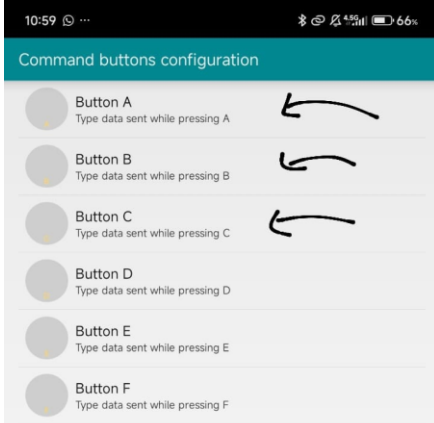
### 2.4. Bluetooth Programının Ayarlanması Görselleri





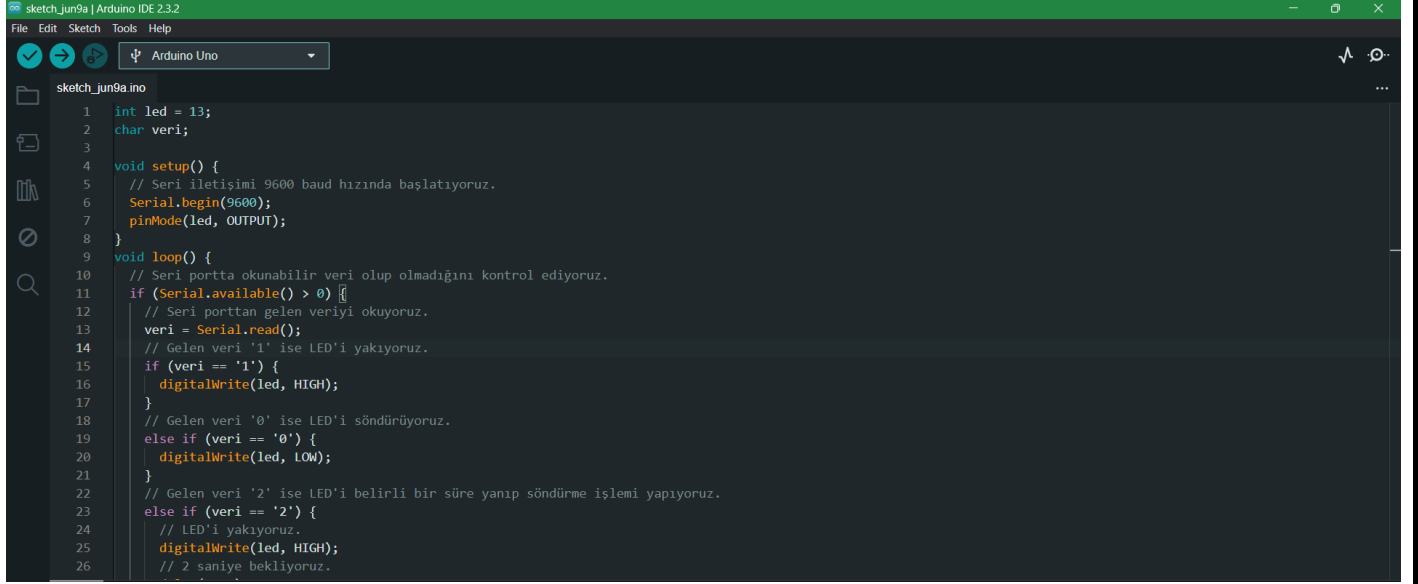
## 2.Yapım Aşaması

### 2.4. Bluetooth Programının Ayarlanması Görselleri

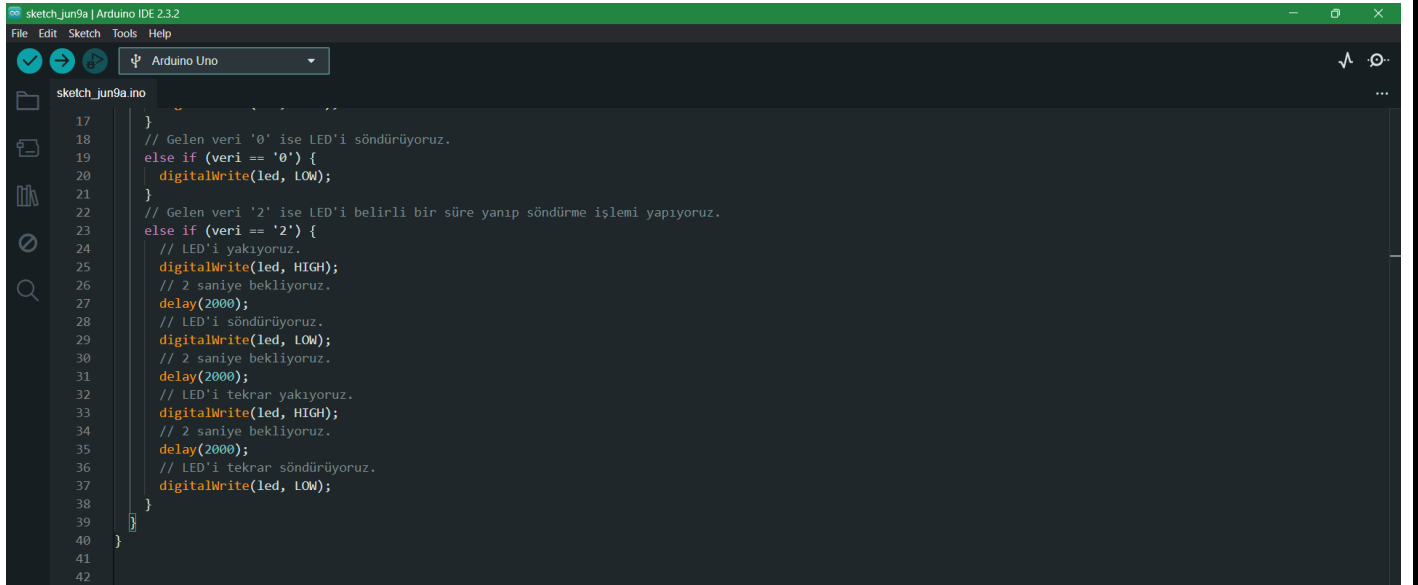


## 3.Kod Kısmı

### 3.1.Arduino Kodun Görselleri



```
sketch_jun9a.ino
1  int led = 13;
2  char veri;
3
4  void setup() {
5      // Seri iletişimi 9600 baud hızında başlatıyoruz.
6      Serial.begin(9600);
7      pinMode(led, OUTPUT);
8  }
9
10 void loop() {
11     // Seri portta okunabilir veri olup olmadığını kontrol ediyoruz.
12     if (Serial.available() > 0) {
13         // Seri porttan gelen veriyi okuyoruz.
14         veri = Serial.read();
15         // Gelen veri '1' ise LED'i yakıyoruz.
16         if (veri == '1') {
17             digitalWrite(led, HIGH);
18         }
19         // Gelen veri '0' ise LED'i söndürüyoruz.
20         else if (veri == '0') {
21             digitalWrite(led, LOW);
22         }
23         // Gelen veri '2' ise LED'i belirli bir süre yanıp söndürme işlemi yapıyoruz.
24         else if (veri == '2') {
25             // LED'i yakıyoruz.
26             digitalWrite(led, HIGH);
27             // 2 saniye bekliyoruz.
28             delay(2000);
29             // LED'i söndürüyoruz.
30             digitalWrite(led, LOW);
31             // 2 saniye bekliyoruz.
32             delay(2000);
33             // LED'i tekrar yakıyoruz.
34             digitalWrite(led, HIGH);
35             // 2 saniye bekliyoruz.
36             delay(2000);
37             // LED'i tekrar söndürüyoruz.
38             digitalWrite(led, LOW);
39         }
40     }
41 }
```



```
sketch_jun9a.ino
17 }
18 // Gelen veri '0' ise LED'i söndürüyoruz.
19 else if (veri == '0') {
20     digitalWrite(led, LOW);
21 }
22 // Gelen veri '2' ise LED'i belirli bir süre yanıp söndürme işlemi yapıyoruz.
23 else if (veri == '2') {
24     // LED'i yakıyoruz.
25     digitalWrite(led, HIGH);
26     // 2 saniye bekliyoruz.
27     delay(2000);
28     // LED'i söndürüyoruz.
29     digitalWrite(led, LOW);
30     // 2 saniye bekliyoruz.
31     delay(2000);
32     // LED'i tekrar yakıyoruz.
33     digitalWrite(led, HIGH);
34     // 2 saniye bekliyoruz.
35     delay(2000);
36     // LED'i tekrar söndürüyoruz.
37     digitalWrite(led, LOW);
38 }
39 }
40
41
42 }
```

## 3.Kod Kısmı

### 3.2. Arduino Kodun Açıklaması

```
1 int led = 13;  
2 char veri;
```

**led:** 13 numaralı dijital pini temsil eder. Bu pin, LED'in bağlı olduğu pin olarak tanımlanmıştır.

**veri:** Seri port üzerinden gelen karakter verisini depolamak için kullanılan bir değişkendir.

```
4 void setup() {  
5     // Seri iletişimi 9600 baud hızında başlatıyoruz.  
6     Serial.begin(9600);  
7     pinMode(led, OUTPUT);  
8 }
```

**setup fonksiyonu:** Arduino programının başlangıcında bir kez çalıştırılır. Bu fonksiyonun amacı, programın çalışması için gerekli olan tüm başlangıç ayarlarını yapmaktır.

**Serial.begin(9600):** Seri iletişimi 9600 baud hızında başlatır. Bu, Arduino'nun bilgisayar veya başka bir cihazla seri port üzerinden iletişim kurmasını sağlar.

**pinMode(led, OUTPUT):** 13 numaralı pini çıkış olarak ayarlar. Bu, LED'in bu pin üzerinden kontrol edilebileceği anlamına gelir.

## 3.Kod Kısmı

### 3.2. Arduino Kodun Açıklaması

```
9 void loop() {  
10 // Seri portta okunabilir veri olup olmadığını kontrol ediyoruz.  
11 if (Serial.available() > 0) {  
12 // Seri porttan gelen veriyi okuyoruz.  
13 veri = Serial.read();  
14 // Gelen veri '1' ise LED'i yakıyoruz.  
15 if (veri == '1') {  
16 digitalWrite(led, HIGH);  
17 }  
18 // Gelen veri '0' ise LED'i söndürüyoruz.  
19 else if (veri == '0') {  
20 digitalWrite(led, LOW);  
21 }  
22 // Gelen veri '2' ise LED'i belirli bir süre yanıp söndürme işlemi yapıyoruz.  
23 else if (veri == '2') {  
24 // LED'i yakıyoruz.  
25 digitalWrite(led, HIGH);  
26 // 2 saniye bekliyoruz.  
27 delay(2000);  
28 // LED'i söndürüyoruz.  
29 digitalWrite(led, LOW);  
30 // 2 saniye bekliyoruz.  
31 delay(2000);  
32 // LED'i tekrar yakıyoruz.  
33 digitalWrite(led, HIGH);  
34 // 2 saniye bekliyoruz.  
35 delay(2000);  
36 // LED'i tekrar söndürüyoruz.  
37 digitalWrite(led, LOW);  
38 }  
39 }  
40 }
```

**Serial.available() > 0:** Seri porttan okunabilir veri olup olmadığını kontrol eder. Eğer veri varsa, if bloğu içine girer.

**veri = Serial.read():** Seri porttan gelen veriyi okur ve veri değişkenine atar.

**if (veri == '1'):** Eğer gelen veri '1' karakteri ise, LED'i yakmak için digitalWrite(led, HIGH) komutunu kullanır.

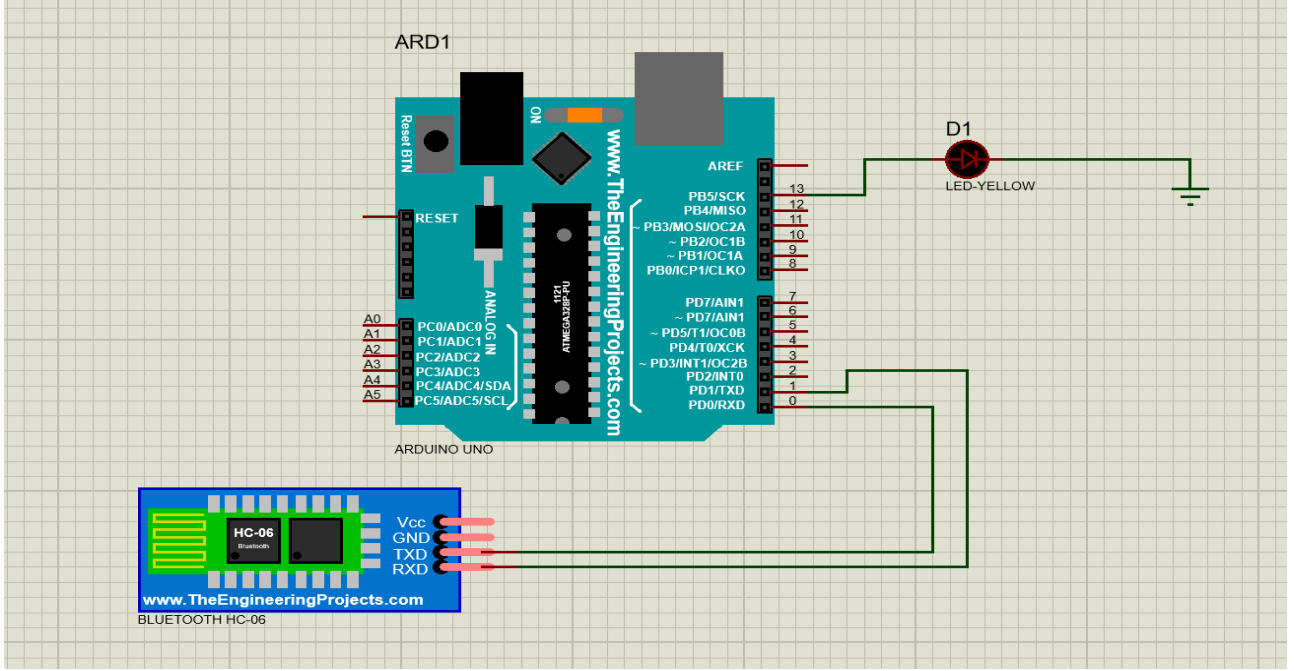
**else if (veri == '0'):** Eğer gelen veri '0' karakteri ise, LED'i söndürmek için digitalWrite(led, LOW) komutunu kullanır.

**else if (veri == '2'):** Eğer gelen veri '2' karakteri ise, LED'i belirli bir süre yanıp söndürme işlemini gerçekleştirir:

- digitalWrite(led, HIGH): LED'i yakar.
- delay(2000): 2 saniye bekler.
- digitalWrite(led, LOW): LED'i söndürür.
- delay(2000): 2 saniye bekler.
- Bu işlemi iki kez daha tekrarlar.

## 4.Görseller

### 4.1. Proteus Görseli



### 4.2. Devre Görseli

