



**BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**MKT0311 MİKROİŞLEMCİ TABANLI SİSTEM TASARIMI
(BAHAR 2021)**

Guide for 8×8 Dot Matrix MAX7219 + Pong Game

Takım Üyesinin

Adı Soyadı: Dilara Kanalcı	Numarası: 18360859013
Adı Soyadı: Ebru Yaşar	Numarası: 18360859008
Adı Soyadı: Fatih Ateş	Numarası: 19360859074

İÇİNDEKİLER:

1.	İÇİNDEKİLER:	2
2.	Proje Adı	3
3.	Proje Amacı	3
4.	Kullanılan Malzemelerin Tanıtımı	3
4.1.	18+ Kitabında Yer Alan Projedeki Malzemeler	3
4.2.	Ek Takılan Sensör veya Komponentler	4
5.	Projenin Farklı Açılardan Fiziksel Görünümü	6
5.1.	Proje Başlatılmamış Durumdayken Devrenin Üstten Görünümü	6
5.2.	Proje Çalışır Haldeyken Devrenin Yandan Görünümü	6
5.3.	Proje Çalışır Haldeyken Android Uygulaması ve Devrenin Görünümü	7
6.	Breadboard - Arduino Bağlantı Şeması ve Açıklaması	8
7.	Proteus - Arduino Bağlantı Şeması ve Açıklaması	9
8.	Donanım Sisteminin Çalışma Mantığı	9
9.	Arduino Yazılım Programının Akış Diyagramı	11
10.	Proje Arduino Yazılım Programı ve Açıklamaları	12
11.	Proje Mobil Uygulaması Ekran Görüntüsü	20
12.	Proje Mobil Yazılım Programı ve Açıklamaları	21
13.	Grup Üyeleri İş Bölümü	26
14.	Kaynakça	27

1. Proje Adı

Guide for 8x8 Dot Matrix MAX7219 + Pong Game

2. Proje Amacı

18+ kitabında bulunan projenin gerçekleştirimi ve Bluetooth modülü ile buzzer eklenerek kullanıcıların oyunu oynama deneyimlerinin geliştirilmesi amaçlanmaktadır.

3. Kullanılan Malzemelerin Tanıtımı

3.1. 18+ Kitabında Yer Alan Projedeki Malzemeler

- **MAX7219 ile Lehimli 8x8 Led Matris:**

8x8'den oluşan 64 adet led bulunmaktadır. Ledlerin katot ve anotları ortaktır. Böylece ortak bir şekilde yanarlar.



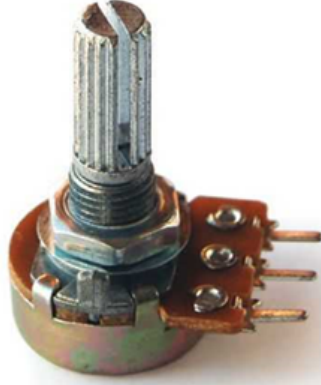
- **Arduino Uno:**

Arduino projeler yapmak için geliştirilmiş yazılım ve donanıma dayalı bir platformdur. Bir çok çeşidi günümüzde aktif olarak kullanılmaktadır. En yaygın modellerinden biri Arduino Uno'dur.



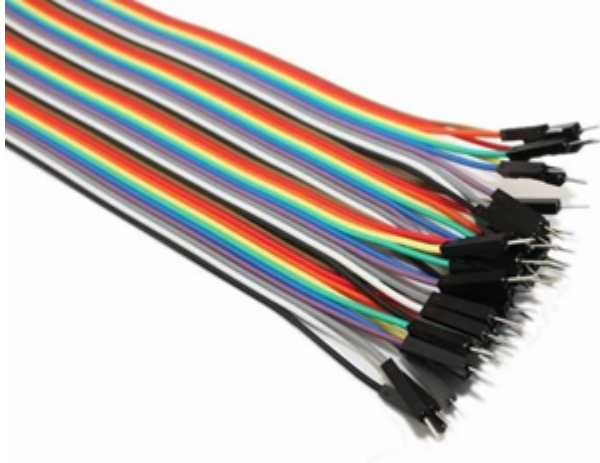
- **1K Ohm’luk Potansiyometre:**

Potansiyometreler kontrol edilebilir dirençlerdir. Arduino ile kullanımı uygundur.



- **Bağlantı Kabloları:**

Bağlantı kabloları elektronik elemanları bağlamak için bakırdan yapılmış kablolardır. F-M, M-M, F-F çeşitleri bulunur.



3.2. Ek Takılan Sensör veya Komponentler

- **HC-06 Bluetooth Modülü:**

Bluetooth modülü Arduino ile TX/RX protokolü ile birbirine bağlanır. RX “Receive” yani almak, TX ise “Transmit” yani vermek kelimelerinden gelmektedir. Arduino ile bluetooth protokolü yardımıyla bilgi alışverişi yapar.



- **Buzzer:**

Buzzer direnç ve transistör yardımıyla pine gelen voltu salınım sinyaline çevirir. Piezo seramik diske yüksek gerilim uygulanır. Bu mekanik genişleme ve daralmaya sebep olur. Böylece içerideki metal plaka ters yönde bükülür. Sürekli bükülen plaka yardımıyla buzzer havada ses dalgası üretir. Buzzerlar pil ile de çalışabilir.



- **330 Ohm'luk 2 Adet Direnç:**

Dirençler akım ve voltajı ayarlamamıza yardımcı olan devre elemanlarıdır. Çok fazla çeşidi vardır ve neredeyse tüm devrelerde kullanılan bir elemandır.



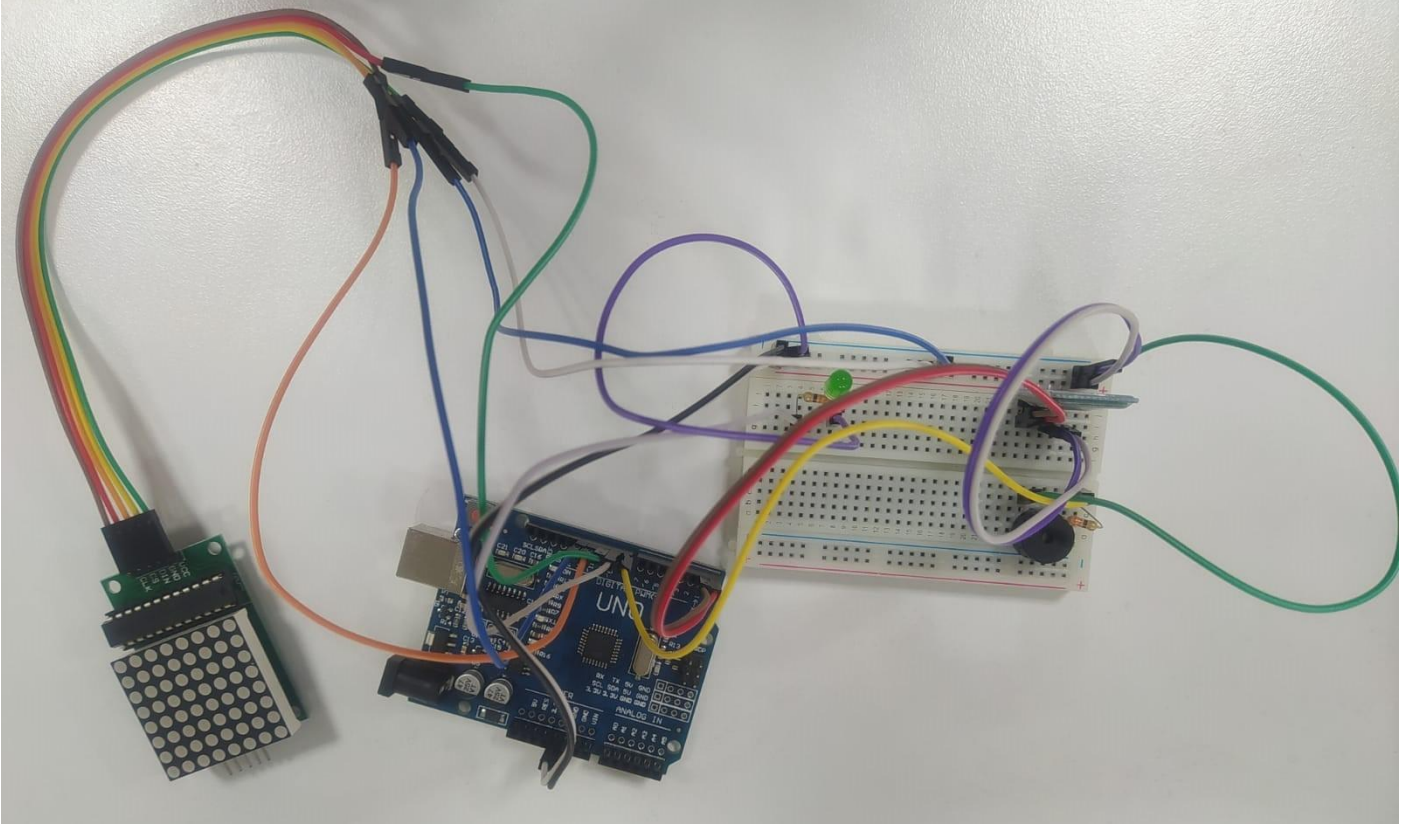
- **Led Lamba:**

Ledler, diyot kullanarak yapılmış elemanlardır. Az akım çekerler ve korunması için direnç kullanımı gereklidir.

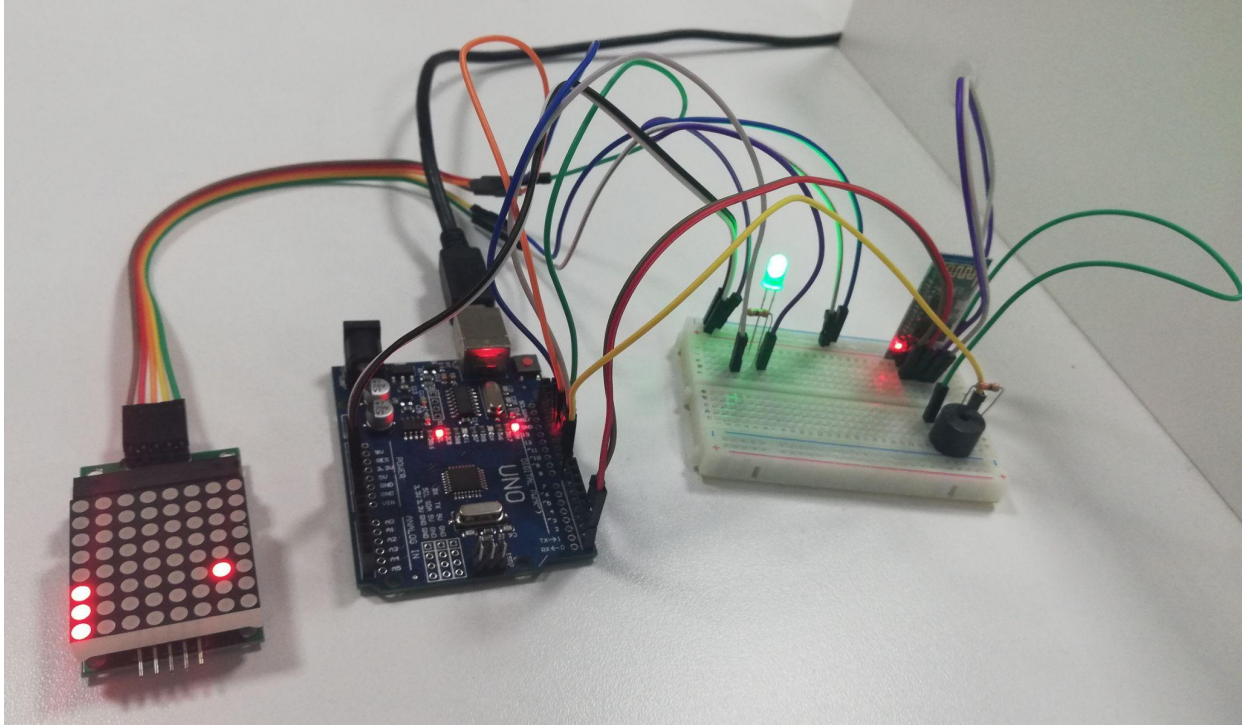


4. Projenin Farklı Açılardan Fiziksel Görünümü

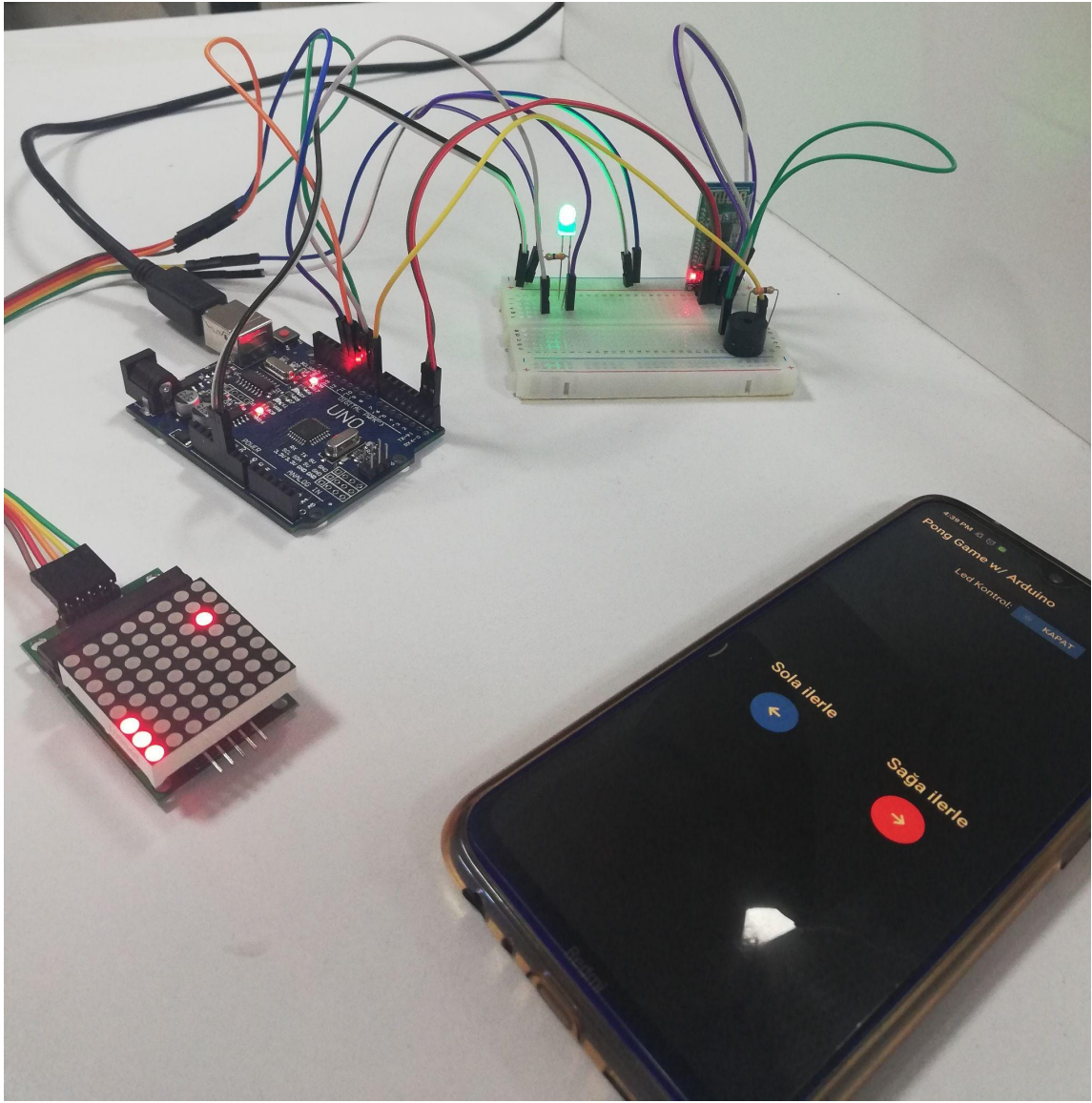
4.1. Proje Başlatılmamış Durumdayken Devrenin Üstten Görünümü



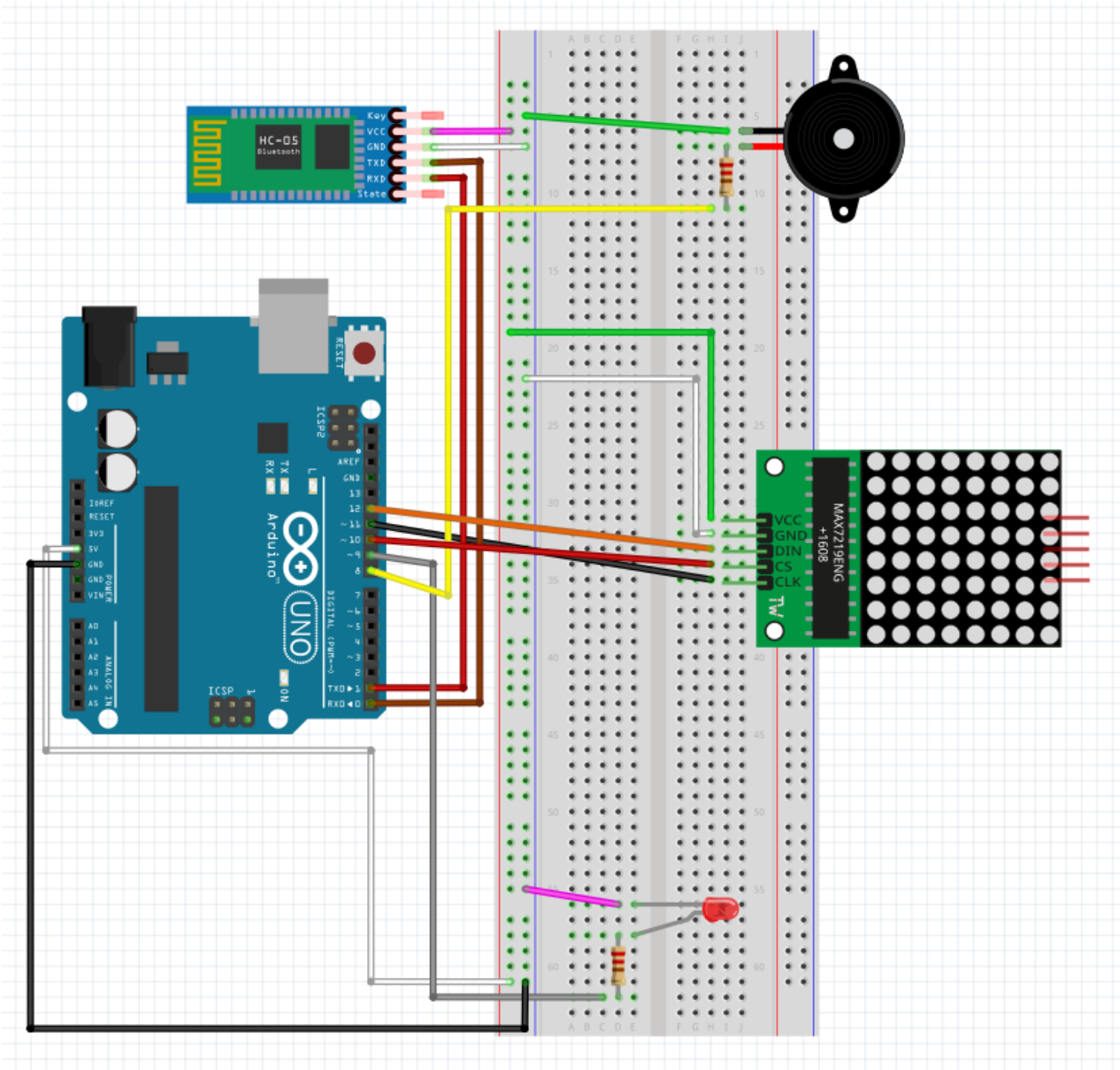
4.2. Proje Çalışır Haldeyken Devrenin Yandan Görünümü



4.3. Proje Çalışır Haldeyken Android Uygulaması ve Devrenin Görünümü

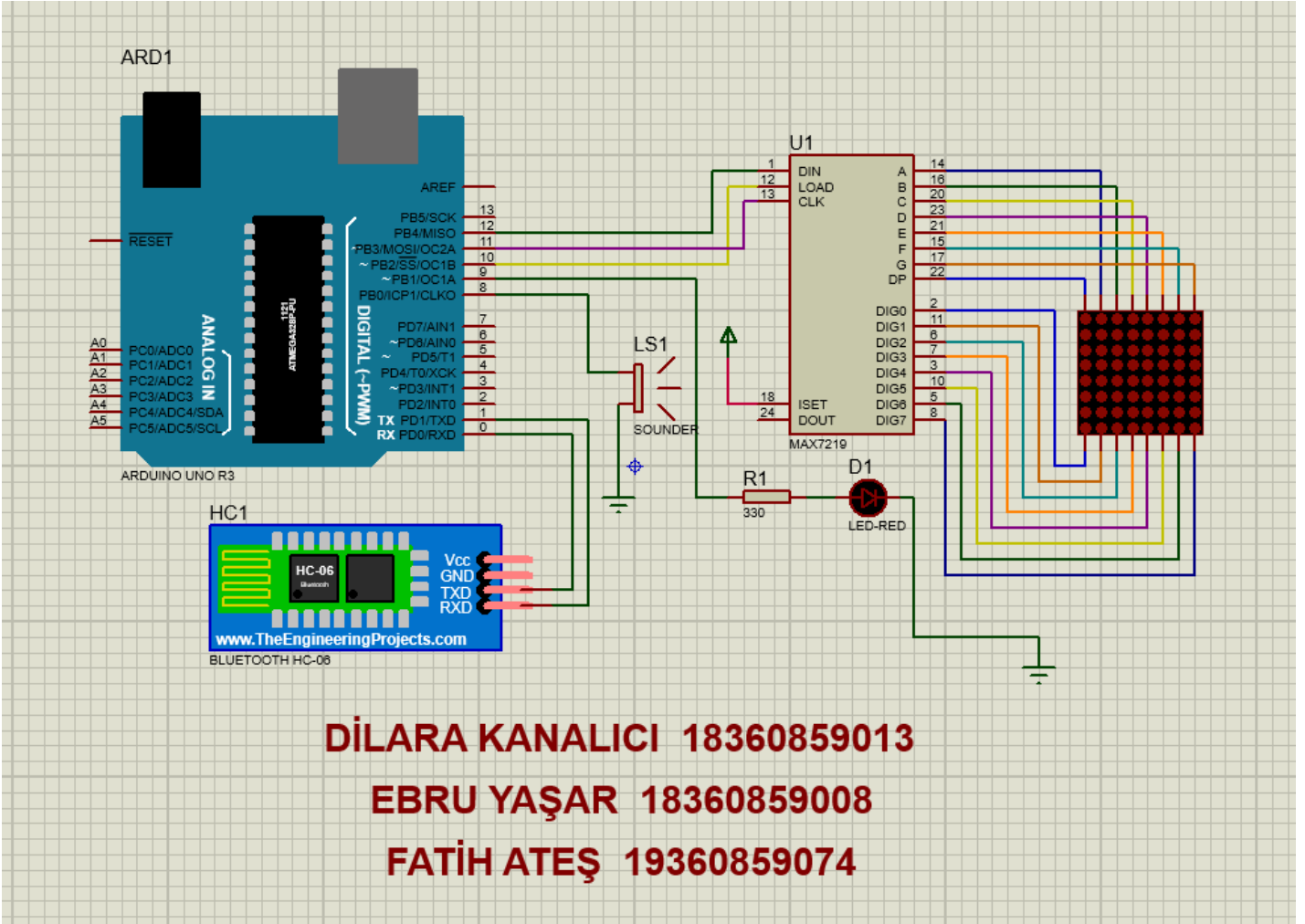


5. Breadboard - Arduino Bağlantı Şeması ve Açıklaması



- İlk olarak Hc-06 modülünün VCC pini breadboard'un 5V pinine, GND pini de breadboard'un toprak pinine bağlanır. TXD pini arduinonun RXD pinine, RXD pini arduinonun TXD pinine bağlanır.
- MAX7219 ve 8x8 led matrisinden oluşan entegrenin GND ve VCC pinlerinin breadboard üzerinde bağlantısı yapıldıktan sonra sırasıyla DIN pini arduinonun 12 numaralı pinine, CS pini 10 numaralı pine ve CLK pini ise 11 numaralı arduino pinine bağlanır.
- Ledin kısa bacağı toprağa, uzun bacağı 330 ohm'luk dirence bağlanır. Direncin boşa kalan diğer bacağı arduinonun 9 numaralı pinine bağlanır.
- Buzzer'ın uzun bacağı 330 ohm'luk dirence, kısa bacağı ise toprağa bağlanır.
- En son olarak arduinonun 5V pini breadboardun 5V pinine, GND pini de toprağa bağlanarak devre tamamlanmış olur.

6. Proteus - Arduino Bağlantı Şeması ve Açıklaması



- Bölüm 5'te verilen breadboard bağlantı şemasına göre Proteus programı üzerinde devrenin simülasyonu gerçekleştirilmiştir.
- Fritzing'de çizilen bağlantı şeması ve canlı devre bağlantısından farklı olarak, Proteus programında MAX7219 ve 8x8 led matris entegre halde bulunmadığı için ikisinin arasındaki bağlantılar arduino koduna da uygun olacak şekilde bağlanarak devrede yer aldı. Yine Fritzing'deki bağlantı şeması ve canlı devreden farklı olarak Proteus'ta buzzer devre elemanı bulunmadığı için yerine sounder kullanıldı.
- Bluetooth modülünün bağlantısının sağlanıp sağlanmadığını kontrol etmek amacıyla devreye led eklendi.
- Arduino ve bluetooth modülü ile birlikte bütün bağlantılar sağlandıktan sonra bluetooth modülüne port bilgisi, arduinoya da hex dosyasının yolu belirtilir ve devre tamamlanmış olur.

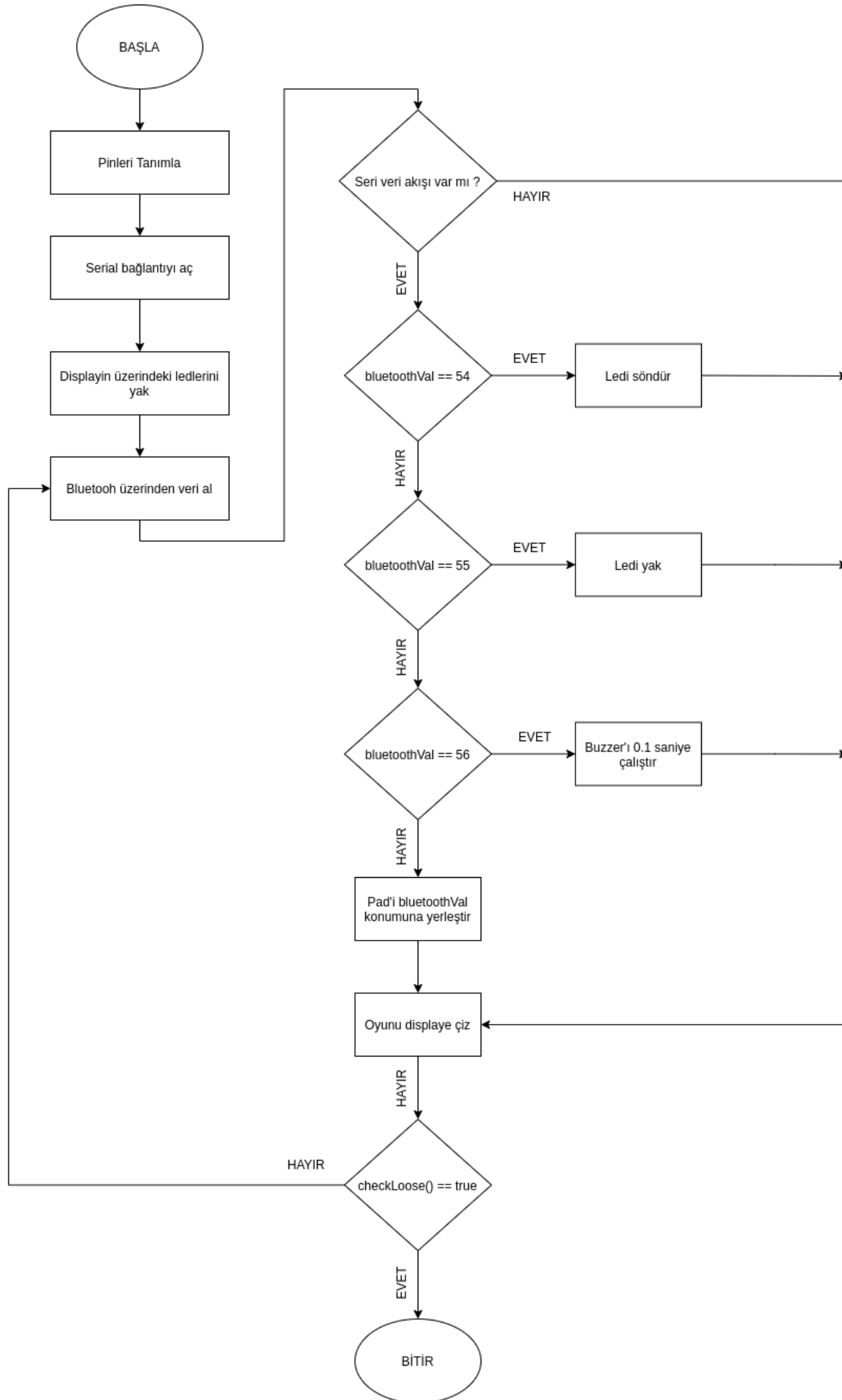
7. Donanım Sisteminin Çalışma Mantığı

Proje içerisinde HC-06 Bluetooth modülü ve Android üzerinde çalışan bir mobil uygulama aracılığıyla 8x8 dot matrix üzerinde Pong oyunu canlandırılmaktadır. Uygulama ve Arduino arasındaki iletişimi HC06 Bluetooth modülü sağlamaktadır. HC06 modülü cihaz ile Arduino arasında 10 metre mesafeye kadar çalışmaktadır.

Breadboard üzerinde bulunan Bluetooth modülü ile pong oyununun oynanması için ergonomik anlamda iyileştirme sunan bir sistem ortaya atılmıştır. Ayrıca eklenmiş olan Yeşil Led ile Android cihaz ile Arduino arasındaki bağlantının sağlanmasını yapma imkanı da kullanıcıya tanınmıştır.

- HC06 modülü ile Arduino ve Cihaz arasında bağlantı kurulur ve bağlantı kurulduğunda HC06 modülü üzerinde yanıp sönen ışık artık sabit yanmaya başlar.
- Cihaz Arduinoya bağlandığında gerçek zamanlı olarak LED açıp-kapatılarak bağlantı kontrolü manuel olarak sağlanabilir.
- Oyun yazılım üzerinden yönetilir ve oyun başladığında gülen surat 8x8 display üzerine bastırılır ve daha sonrasında pad ve top ekranda görünür ve kullanıcı artık oyunu oynayabilir hale gelir.
- Cihaz üzerinden sağa ilerle veya sola ilerle butonuna tıklandığında Arduinoya tamsayı türünde veriler gelir ve Arduino bunları işleyerek 8x8 display üzerindeki Pad'i hareket ettirir.
- Eğer ki gelen tamsayı sınırlardan taşıyor ise 8 numaralı pin üzerinden buzzer ile sınırı geldiğini belirtmek amacıyla uyarı verilir.
- Oyun sirkülasyon halinde devam ederken eğer ki top Padin bulunduğu LED noktalarından farklı bir alt noktaya değerse oyunu kaybeder ve üzgün surat displaye bastırılır ardından geliştiricilerin isimlerinin baş harfleri ekrana bastırılır ve daha sonrasında sirkülasyon devam eder.

8. Arduino Yazılım Programının Akış Diyagramı



9. Proje Arduino Yazılım Programı ve Açıklamaları

```
#include "LedControl.h"// max 7219 ile led kontrol kütüphanesi
#include "Timer.h"
```

```
#define LEDPIN 9 // led pini
#define BUZZERPIN 8 // buzzer pini
#define PADSIZ 3 //oynanacak çubuğun uzunluğu
#define BALL_DELAY 400
#define GAME_DELAY 10
#define BOUNCE_VERTICAL 1
#define BOUNCE_HORIZONTAL -1
#define NEW_GAME_ANIMATION_SPEED 50 //define ile yapılan değişiklikler ilk yazılan veriyle yanına
yazılan value için derlerken kodda yer değiştirmesini sağlar
#define HIT_NONE 0
#define HIT_NONE 0
#define HIT_CENTER 1
#define HIT_LEFT 2
#define HIT_RIGHT 3
```

```
//#define DEBUG 1
```

```
byte sad[] = {
B00000000,
B01000100, //dot matrix için 8x8 lik matriks tanımlandı ve her bir lede HIGH konumu için 1, LOW konumu için
0 verildi
B00010000,
B00010000,
B00000000,
B00111000,
B01000100,
B00000000
};
```

```
byte smile[] = {
B00000000,
B01000100,
B00010000, //dot matrix için 8x8 lik matriks tanımlandı ve her bir lede HIGH konumu için 1, LOW konumu için
0 verildi
B00010000,
B00010000,
B01000100,
B00111000,
B00000000
};
```

Timer timer; // timer.h kütüphanesi yardımıyla kullanılmak üzere zamanlayıcı tanımlandı.

LedControl lc = LedControl(12,11,10,1); //led control kütüphanesine MAX7219 driver için kullanılacak pinleri tanımladık. son rakam ise kullanılacak driver sayısıdır.

byte direction; //Rüzgar gülü, 0 kuzeydir

```
int xball;  
int yball;  
int yball_prev;  
byte xpad;  
int ball_timer;
```

```
int bluetoothVal; // bluetooth veri akışı için kullanılacak string tanımlandı
```

```
void setSprite(byte *sprite){  
    for(int r = 0; r < 8; r++){  
        lc.setRow(0, r, sprite[r]); //ilk değer matrix adresidir 1 matriks olduğundan 0 değeri verilir. ikinci parametre  
        satırlar arasında gezmemizi sağlar. üçüncü parametrede satırdaki yanacak ledleri byte cinsinden verir  
        (dizideki değerler alınacak  
    }  
}
```

```
void newGame() {  
    lc.clearDisplay(0); //tüm ledler adresteki matriks için söner  
    // x ve y pozisyonlarına değer verme  
    xball = random(1, 7); // 1ve 7 arasında random sayı üretir.  
    yball = 1;  
    direction = random(3, 6); // Güneye gider  
    for(int r = 0; r < 8; r++){  
        for(int c = 0; c < 8; c++){  
            lc.setLed(0, r, c, HIGH); // satır ve sütunlardaki ledi row coulumn numarasına göre yakar.  
            delay(NEW_GAME_ANIMATION_SPEED); //define ile tanımlanmış süre kadar bekletir  
        }  
    }  
    setSprite(smile); //gülen yüz çalıştırılır.  
    delay(1500); //1.5 sn bekler  
    lc.clearDisplay(0); //matriks ledleri söndürülür.  
}
```

```
void setPad(int x) {  
  
    xpad = x; // çubuğa konum verilmiştir  
  
}
```

```
void debug(const char* desc){ // konumları kontrol etmek için yazılmış fonksiyondur  
#ifdef DEBUG  
    Serial.print(desc);  
    Serial.print(" XY: ");  
    Serial.print(xball);  
    Serial.print(", "); //topun x ve y koordinatlarını basar  
    Serial.print(yball);  
    Serial.print(" XPAD: "); //çubuğun konumunu basar  
    Serial.print(xpad);  
    Serial.print(" DIR: "); //direction bastırır  
    Serial.println(direction);  
#endif  
}
```



```

int checkBounce() {
    if(!xball || !yball || xball == 7 || yball == 6){//xball veya yball null değilse veya xball 7 ise veya yball 6 ise
        int bounce = (yball == 0 || yball == 6) ? BOUNCE_HORIZONTAL : BOUNCE_VERTICAL;// yball 0 veya
yball 6 ise 1 horizontal veya -1 vertical değerini ata
#ifdef DEBUG
        debug(bounce == BOUNCE_HORIZONTAL ? "HORIZONTAL" : "VERTICAL");//
BOUNCE_HORIZONTAL bounce değerindeyse horizontal kelimesini değilse vertical kelimesini debug
fonksiyonuna gönderir.
#endif
        return bounce;//bounce değerini döndürür
    }
    return 0;
}

```

```

int getHit() { // çarpma var mı var ise koordinatını alır
    if(yball != 6 || xball < xpad || xball > xpad + PADSIZ){
        return HIT_NONE;
    }
    if(xball == xpad + PADSIZ / 2){
        return HIT_CENTER; //oynayan çubuğun ortasına çarpıp çarpmadığına bakar
    }
    return xball < xpad + PADSIZ / 2 ? HIT_LEFT : HIT_RIGHT; //ne tarafa çarptığını döndürür
}

```

```

bool checkLoose() {
    return yball == 6 && getHit() == HIT_NONE; // oyuncunun kaybedip kaybetmediğini kontrol eder.
}

```

```

void moveBall() {
    debug("MOVE"); // debug fonksiyonuna move gönderir
    int bounce = checkBounce(); // fonksiyondan değeri alır.
    if(bounce) { // sektiği yere göre konumu değişecek
        switch(direction){
            case 0:
                direction = 4; //yukarıdaysa aşağı inecektir
                break;
            case 1:
                direction = (bounce == BOUNCE_VERTICAL) ? 7 : 3; //geldiği tarafa göre 1 veya 5 yönüne gider.
saat 4 ve 10 arası
                break;
            case 2:
                direction = 6;//saat 3 yönünden saat 9 yönüne yönleneceğini belirtiyor
                break;
            case 6:
                direction = 2; //saat 9 yönünden saat 3 yönüne yönleneceğini belirtiyor
                break;
            case 7:
                direction = (bounce == BOUNCE_VERTICAL) ? 1 : 5; // geldiği tarafa göre 1 veya 5 yönüne gider.
saat 8 ve 2 yönü arası
                break;
            case 5:
                direction = (bounce == BOUNCE_VERTICAL) ? 3 : 7; //geldiği tarafa göre 1 veya 5 yönüne gider.
saat 4 ve 10 arası

```

```

        break;
        case 3:
            direction = (bounce == BOUNCE_VERTICAL) ? 5 : 1; // geldiği tarafa göre 1 veya 5 yönüne
gider.saat 8 ve 2 yönü arası
            break;
        case 4:
            direction = 0; //aşağıdaysa yukarı çıkacaktır
            break;
    }
    debug("->");
}

switch(getHit()){
    case HIT_LEFT:
        if(direction == 0){//çubuğun sol tarafına vurduysa ve yönü yukarı doğruysa direction saat 10 yönüne
doğru ayarla
            direction = 7;
        } else if (direction == 1){ //çubuğun sol tarafına vurduysa ve yönü saat 2 yönüne doğruysa direction
saat yukarı doğru ayarla
            direction = 0;
        }
        break;
    case HIT_RIGHT:
        if(direction == 0){// çubuğun sağ tarafına vurulduysa ve yön yukarı doğruysa direction saat 2 yönüne
ayarla
            direction = 1;
        } else if(direction == 7) { // çubuğun sağ tarafına vurulduysa ve yön saat 10 yönüne doğruysa direction
yukarı yöne ayarla
            direction = 0;
        }
        break;
}

// kenar ve ortagonal kontrolleri
if((direction == 0 && xball == 0) || (direction == 4 && xball == 7)){ // direction yukarı doğru ve xball 0 ise
veya direction aşağı doğru ve xball 7 ise sağa döndür
    direction++;
}
if(direction == 0 && xball == 7){ // direction yukarı doğru ve xball 7 ise direction saat 10 yönüne ayarla
    direction = 7;
}
if(direction == 4 && xball == 0){ // direction aşağı doğru ve xball 0 ise direction saat 4 yönüne ayarla
    direction = 3;
}
if(direction == 2 && yball == 0){ // direction saat 3 yönüne doğru ve yball 0 ise direction saat 4 yönüne
ayarla
    direction = 3;
}
if(direction == 2 && yball == 6){ // direction saat 3 yönüne doğru ve yball 6 ise direction saat 2 yönüne
ayarla
    direction = 1;
}
}

```

```

    if(direction == 6 && yball == 0){ // direction saat 9 yönüne doğru ve yball 0 ise direction saat 8 yönüne
ayarla
    direction = 5;
}
    if(direction == 6 && yball == 6){ // direction saat 9 yönüne doğru ve yball 6 ise direction saat 10 yönüne
ayarla
    direction = 7;
}

// köşelere çarparsa
if(xball == 0 && yball == 0){ // sol üst köşeye çarparsa direction saat 4 yönüne alınır
    direction = 3;
}
if(xball == 0 && yball == 6){ // sol alt köşeye çarparsa direction saat 2 yönüne alınır
    direction = 1;
}
if(xball == 7 && yball == 6){ // sağ alt köşeye çarparsa direction saat 10 yönüne alınır
    direction = 7;
}
if(xball == 7 && yball == 0){
    direction = 5; // sağ üst köşeye çarparsa direction saat 8 yönüne alınır
}

yball_prev = yball;
if(2 < direction && direction < 6) { // yön saat 3 ile 9 arasındaysa y artar
    yball++;
} else if(direction != 6 && direction != 2) { // yön saat 3 ile 9 arasında değilse y azalır
    yball--;
}
if(0 < direction && direction < 4) { // yön saat 12 ile 6 arasındaysa x artar
    xball++;
} else if(direction != 0 && direction != 4) { // yön saat 12 ile 6 arasında değilse x azalır
    xball--;
}
xball = max(0, min(7, xball)); // xball değerine alabileceği en büyük değeri verir
yball = max(0, min(6, yball)); // yball değerine alabileceği en büyük değeri verir
debug("AFTER MOVE");
}

void gameOver() {
    setSprite(sad); // üzgün yüzü bastırır
    delay(1500); // 1.5 sn bekler
    lc.clearDisplay(0); // tüm ledleri söndürür
}

void drawGame() { // oyunu çizdirir
    if(yball_prev != yball){ // topun y koordinatı öncekinden farklıysa
        lc.setRow(0, yball_prev, 0); // matriksin yball_prev indeksli satırda tüm ledleri söndürür
    }
    lc.setRow(0, yball, byte(1 << (xball))); // matriksin yball indeksli satırındaki ledleri sola kaydırarak yakar.
    byte padmap = byte(0xFF >> (8 - PADSIZ) << xpad); // çubuğu gelen isteğe göre kaydırır.
#ifdef DEBUG
    Serial.println(padmap, BIN);
#endif
}

```

```

#endif
    lc.setRow(0, 7, padmap); // kaydırıldıktan sonra çubuğu çizdirir
}

void isimyazdir(){

    byte birinci = B11100110;
    byte ikinci = B11100101; //dot matrix için 8x8 lik matriks tanımlandı ve her bir lede HIGH konumu için 1,
    LOW konumu için 0 verildi
    byte ucuncu= B10000101;
    byte dorduncu= B10111110;
    byte besinci= B00100000;
    byte altinci= B00111000;
    byte yedinci= B00100000;
    byte sekizinci=B00111000;

    int i=0;
    while(i<=7){

        byte ad[] = {birinci,ikinci,ucuncu,dorduncu,besinci,altinci,yedinci,sekizinci};
        setSprite(ad);
        delay(500);
        birinci= byte(birinci >> 1);
        ikinci= byte(ikinci >> 1);
        ucuncu= byte(ucuncu >> 1);
        dorduncu= byte(dorduncu >> 1);
        besinci= byte(besinci >> 1);
        altinci= byte(altinci >> 1);
        yedinci= byte(yedinci >> 1);
        sekizinci= byte(sekizinci >> 1);
        i=i+1;

    }

}

void setup() {

    pinMode(LEDPIN, OUTPUT); // 9 numaralı pini çıkış pini olarak ayarlıyoruz, led için kullanılıyor
    pinMode(BUZZERPIN, OUTPUT); // 8 numaralı pini çıkış pini olarak ayarlıyoruz, buzzer için kullanılıyor

    Serial.begin(9600); // serial portu açar, saniyede 9600 bit yollar

    lc.shutdown(0,false); // max7219 driver güç tasarrufundan çıkarılır

    lc.setIntensity(0, 8); // 0 nolu led matriksin 8 seviyesinde led parlaklığını ayarlar (0 dan 15 e kadar
    ayarlanabilir

```

```
lc.clearDisplay(0);// tüm ledleri söndürür
randomSeed(analogRead(0)); /// A0 pininden değer okur o değerle rastgele bir dizi oluşturur ve bu dizidei
rastgele bir noktadan başlangıç seçer
#ifdef DEBUG

    Serial.println("Pong");
#endif
    newGame();// yeni oyunu başlatır
    ball_timer = timer.every(BALL_DELAY, moveBall);// her 200 msde moveball fonksiyonunu çalıştırır
}

void loop() {
    if (Serial.available())// tanımlanan pinlerle bağlantı kurulabiliyorsa(veri akışı yapılabilecekse)
    {

        bluetoothVal = Serial.read();// telefonda gelen string değeri bluetooth modülü okur

        if(bluetoothVal == 54){

            digitalWrite(LEDPIN, LOW); // led söner

        }
        else if(bluetoothVal == 55){

            digitalWrite(LEDPIN, HIGH); // led yanar

        }
        else if(bluetoothVal == 56){

            tone(BUZZERPIN, 1000); // buzzerın ses vermesini sağlar
            delay(100); // 0.1 saniye gecikme sağlar
            noTone(BUZZERPIN); // buzzerı durdudur

        }
        else {

            setPad(bluetoothVal % 6); // gelen değere göre( modunu alarak) çubuğu hareket ettirir.

        }

    }

    timer.update();// zamanlayıcıyı günceller

// setPad();// çubuğu hareket ettir

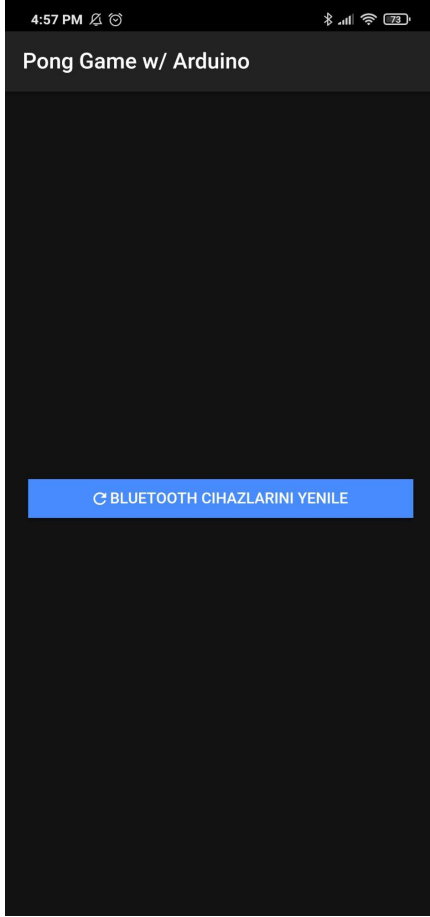
#ifdef DEBUG
    Serial.println(xpad);//debug modda konumunu basar
#endif
    // Ekranı günceller
    drawGame();// oyunu çizdirir
    if(checkLoose()){// kaybedip etmediğini kontrol eder.
        debug("LOOSE");
    }
}
```



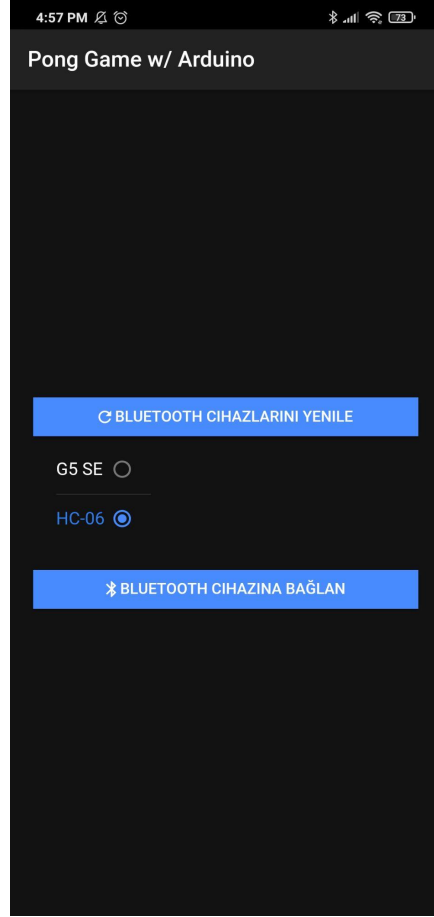
```
    gameOver();// game over fonksiyonu çalışır  
    isimyazdir();  
    newGame();// yeniden oyunu başlatır  
}  
delay(GAME_DELAY);// loop aralarında bekleme süresi  
}
```

10. Proje Mobil Uygulaması Ekran Görüntüsü

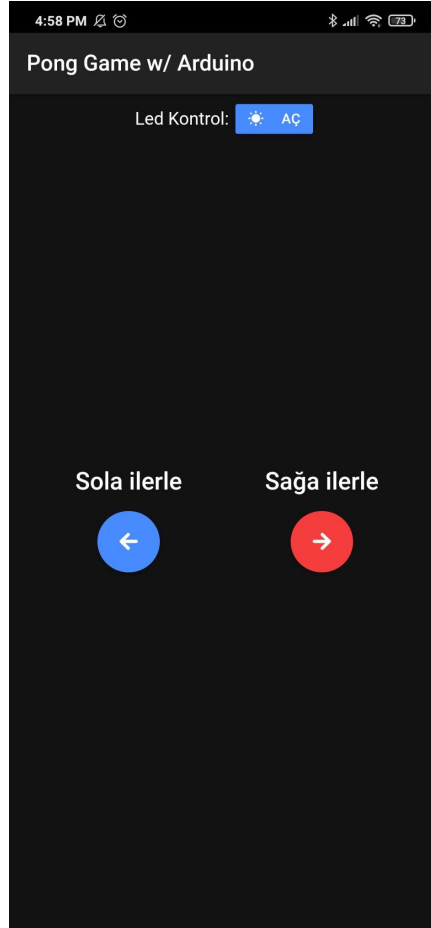
10.1. Başlangıç Ekranı



10.2. Cihazların Listelenmesi



10.3. Oyun Ekranı



11. Proje Mobil Yazılım Programı ve Açıklamaları

```
import { Component } from '@angular/core'
import { BluetoothSerial } from '@ionic-native/bluetooth-serial'
import { LoadingController, AlertController, ToastController, Loading } from 'ionic-angular'

// Ionic için component özellikleri tanımlanıyor
@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})

// İstekleri karşılayacak class oluşturuluyor
export class HomePage {

  // Public değişkenler tanımlanıyor.
  pairedList: pairedlist
  listToggle: boolean = false
  pairedDeviceID: number = 0
  isOpen: boolean = false
  dataSend: string = ""
  isConnected: boolean = false
  locationX: number = 0

  // Kurucu metod kendisine verilen parametreleri otomatik olarak nesneye aktarıyor
  // örn: this.loadingController olarak erişilebilir
  constructor(public loadingController: LoadingController, private alertCtrl: AlertController,
    private bluetoothSerial: BluetoothSerial, private toastCtrl: ToastController) {

  }

  // İşlemler gerçekleştirilirken gösterilen loader
  presentLoadingWithOptions(content = "İşleminiz gerçekleştiriliyor..."): Loading {

    // Yeni bir loadingController oluşturuluyor.
    const loading = this.loadingController.create({
      spinner: null,
      content: content,
      cssClass: 'custom-class custom-loading',
    })

    // Controller ekranda gösteriliyor
    loading.present()

    // Controller kapandığında gerçekleşecek olaylar tanımlanıyor.
    loading.onDidDismiss(function(data, role){
      console.log('Loading dismissed with role:', role)
    })

    return loading
  }
}
```

```
// Bluetooth'nin açık/kapalı durumunu kontrol eden metod  
checkBluetoothEnabled() {
```

```
    // Loader başlatılıyor  
    const loading = this.presentLoadingWithOptions()  
  
    // Durum kontrol ediliyor  
    this.bluetoothSerial.isEnabled()  
        .then(() => {  
        // Açıkysa loader kapatılıyor  
        loading.dismiss()  
    })  
    .catch(error => {  
        // Kapalıysa loader kapatılıyor ve açılması için uyarı veriyor  
        loading.dismiss()  
        this.showError("Lütfen Bluetooth aracınızı aktif hale getirin.")  
    })  
}
```

```
// Daha önce bluetooth eşleştirmesi yapılmış cihazları listeleyen metod  
listPairedDevices() {
```

```
    // Loader başlatılıyor  
    const loading = this.presentLoadingWithOptions()  
  
    // Listeleme gerçekleştiriliyor  
    this.bluetoothSerial.list()  
        .then(success => {  
        // Listeleme başarılıysa veriler pairedList değişkenine aktarılıyor  
        // Sayfada listeleme sağlanıyor  
        // Loader kapatılıyor  
        this.pairedList = success  
        this.listToggle = true  
        loading.dismiss()  
    })  
    .catch(error => {  
        // Listeleme başarısızsa loader kapatılıyor  
        // Hatalar ekrana yazılıyor  
        // Sayfadan listeleme kaldırılıyor  
        loading.dismiss()  
        this.showError(error + ".\n")  
        this.listToggle = false  
    })  
}
```

```
// Bir cihazı seçmeyi sağlıyor  
selectDevice() {
```

```
    // Seçilen cihazı diziden getiriyor  
    let connectedDevice = this.pairedList[this.pairedDeviceID]  
    // Eğer ki gelen nesne boş ise uyarı veriyor  
    if (!connectedDevice.address) {  
        this.showError('Eşleştirmek istediğiniz cihaz seçiniz.')
```

```

        return -1
    }
    // Cihazın adresini alıyor
    let address = connectedDevice.address

    // Cihaza bağlanıyor
    this.connect(address)
}

// Bir cihaza bağlanmayı sağlıyor
connect(address) {

    // Loader başlatılıyor
    const loading = this.presentLoadingWithOptions()

    // Cihaza bağlanılıyor
    this.bluetoothSerial.connect(address)
        .subscribe(success => {
            // Bağlantı başarılıysa bluetoothSerial değişkenine abone olunuyor
            // Loader kapatılıyor
            // Kullanıcıya mesaj veriliyor
            // isConnected değişkeni güncelleniyor
            this.deviceConnected()
            loading.dismiss()
            this.showToast("Bluetooth bağlantısı başarılı!")
            this.isConnected = true

        }, error => {
            // Bağlantı başarısızsa abone olunduğu için abonelik sonlanana kadar
            // Loader kapatılıyor
            // isConnected değişkeni güncelleniyor
            // Kullanıcıya mesaj veriliyor
            if(loading)
                loading.dismiss()
            this.isConnected = false
            this.showError("Cihaz ile telefon arasındaki bağlantı koparıldı.")
        })
}

// BluetoothSerial değişkenine abone olmayı sağlıyor
deviceConnected() {

    // Değişkenine abone olunuyor
    this.bluetoothSerial.subscribe('\n')
        .subscribe(success => {
            return 0
        }, error => {
            this.showError(error)
        })
}

```

çalışabilir


```
// Led kontrolü yapılmasını sağlıyor
```

```
ledChange(param: boolean) {
```

```
    // Led açılıyorsa 7 kapanıyorsa 6 değeri cihaza iletiliyor
```

```
    this.dataSend = param ? '7': '6'
```

```
    this.bluetoothSerial.write(this.dataSend)
```

```
        .then(success => {
```

```
            // İşlem başarılıysa mesaj veriliyor
```

```
            // isOpen değişkeni güncelleniyor
```

```
            this.showToast(param ? 'AÇILDI': 'KAPATILDI')
```

```
            this.isOpen = !this.isOpen
```

```
        })
```

```
        .catch(error => {
```

```
            // İşlem başarısızsa mesaj veriliyor
```

```
            this.showError(error)
```

```
        })
```

```
}
```

```
// Pad'i hareket ettirmeyi sağlıyor
```

```
moveTo(param: string) {
```

```
    // Değişken yerel bir değişkene aktarılıyor(0-5 arasında bir değere sahip olabilir)
```

```
    // Parametre Right ise değer 1 azaltılıyor
```

```
    // Parametre Left ise değer 1 artırılıyor
```

```
    var newLocation: number = this.locationX
```

```
    if(param == 'Right')
```

```
        newLocation -= 1
```

```
    else if(param == 'Left')
```

```
        newLocation += 1
```

```
    else
```

```
        return 0
```

```
    // Değer kontrolü yapıyor
```

```
    if(newLocation > 5 || newLocation < 0){
```

```
        this.showToast('Sınırdasınız')
```

```
        this.bluetoothSerial.write("8")
```

```
        return 0
```

```
    }
```

```
    // Değer cihaza gönderiliyor
```

```
    this.bluetoothSerial.write(newLocation.toString())
```

```
        .then(success => {
```

```
            // Veri başarıyla iletiliyse değer güncelleniyor
```

```
            this.locationX = newLocation
```

```
        })
```

```
        .catch(error => {
```

```
            // Veriler iletilemediyse hata mesajı gösteriliyor
```

```
            this.showError(error)
```

```
        })
```

```
}
```

```
// Hata mesajı göstermeyi sağlıyor
```

```
showError(error) {
```

```
// Gönderilen parametreye göre hata componenti üretiyor
// Component ekranda gösteriliyor
let alert = this.alertCtrl.create({
    title: 'Hata',
    subTitle: error,
    buttons: ['Kapat']
})
alert.present()
}

// Tost mesajları göstermeyi sağlıyor
showToast(msj) {
    // Gönderilen parametreye göre toast component üretiyor
    // Component ekranda gösteriliyor
    const toast = this.toastCtrl.create({
        message: msj,
        duration: 1000
    })
    toast.present()
}

}

// PairedList adında bir interface tanımlanıyor
// TypeScript dilinde interfaceler daha çok tip tanımlamada kullanılır
interface pairedlist {
    "class": number,
    "id": string,
    "address": string,
    "name": string
}
```

12. Grup Üyeleri İş Bölümü

Dilara Kanalcı:

- Devre elemanlarının ve çalışma mantıklarının açıklanması
- Projenin arduino kodunun gerçekleştirilmesi
- İş bölümünü doğrultusunda yapılanların rapora işlenmesi

Ebru Yaşar:

- Canlı devre şemasının Fritzing'de çiziminin gerçekleştirilmesi ve açıklamaları
- Canlı devre ve Fritzing'deki çizim doğrultusunda devrenin simülasyonunun Proteus'ta kurulması ve devrenin açıklanması
- Donanım sisteminin çalışma mantığının açıklanması
- İş bölümü doğrultusunda yapılanların rapora işlenmesi
- Raporun düzenlenmesi

Fatih Ateş:

- Proje mobil yazılım programının kodlanması ve kodun açıklanması
- Arduino yazılımının akış diyagramı
- Devrenin canlı halinin kurulması ve test işlemleri
- Donanım sisteminin çalışma mantığının açıklanması
- İş bölümü doğrultusunda yapılanların rapora işlenmesi
- Raporun düzenlenmesi

13. Kaynakça

- <https://ionicframework.com/>
- <https://forum.ionicframework.com/t/ionic-start-followed-by-ionic-serve-fails-today/170709>
- <https://forum.ionicframework.com/t/read-multiple-data-via-bluetooth-with-ionic-and-arduino/110886>
- <https://stackoverflow.com/questions/33946163/how-to-communicate-between-ionic-cordova-application-and-arduino>
- <https://create.arduino.cc/projecthub/dfornaciari/mobile-arduino-communication-ble-device-85f943>
- <https://ionicframework.com/docs/native/bluetooth-serial>
- <https://angular.io/docs>
- <https://www.best-microcontroller-projects.com/max7219.html>
- <https://microcontrollerslab.com/max7219-8-digit-led-display-driver/>
- <https://www.kaizen40.com/max7219-7-segmentli-display/>
- <https://yazcizboz.wordpress.com/2012/02/09/pic-ile-7-segment-display-srmek-max7219/>
- <https://www.makerguides.com/max7219-led-dot-matrix-display-arduino-tutorial/>
-
- <https://sinancanbayrak.com/buzzer-nedir-nasil-calisir-nicin-kullanilir-kac-cesit-buzzer-vardir/>
- <https://www.robodukkan.com/Erkek-Disi-Baglanti-Kablosu-1-Adet-20-Cm-Dupont-Kablo-PR-166.html>
-
- <https://maker.robotistan.com/arduino-dersleri-17-hc-05-bluetooth-modulu-kullanimi/>