

# HEDEF UZAY

Enes Malik Tok - 180202042 — Yekta Ahmet Gör - 180202075

Kocaeli Üniversitesi  
Bilgisayar Mühendisliği

**Abstract**—Geliştirdiğimiz oyunda oyuncu bir uzay gemisi yönetmektedir. Joystick yardımıyla uzay gemisini sağa sola hareket ettirerek üstüne gelen engellerden kaçınmakta ve ateş butonu yardımıyla da üzerine gelen engellere ateş ederek onları yok edebilmektedir. Oyuncuya zorluk seçenekleri sunulmakta ve bunları menüden seçebilmektedir. Ayrıca oyuncunun oyunu sadece ekrandan değil elektrik devresinden de takip edebilmesi için görsel, işitsel bir çok cihaz bulunmaktadır devrede. Aynı zamanda oyunun içinde bulunduğu ortama göre oyunun renkleri değişmektedir.

**Index Terms**—Arduinu Uno, SSD1306 OLED Ekran, Shift Register, LED, LDR, Buzzer, 7 Segment Display, Servo Motor, Potansiyometre, Direnc, Transistor, Buton

## I. GİRİŞ

Oyunumuzun ismi Hedef Uzay. Bu oyunda 2 boyutlu 8 genişliğe ve 16 uzunluğa sahip olan bir matriste, oyuncu kullandığı uzay gemisini üstüne gelen meteorlardan koruması gerekmektedir. Cihaz çalıştırıldıktan sonra oyuncunun karşısına bir menü çıkmakta ve bu menüde zorluk seviyesi seçmektedir. Oyunda iki adet zorluk seviyesi bulunmaktadır. Easy ve Hard seviyeleri. Easy, yani kolay seviyeyi seçmek için S1 seçeneğini seçmeli ve ateş butonuna basarak oyunu başlatması gerekmektedir. Aynı şekilde Hard seviyeyi seçmek için de S2 seçeneğini seçmeli ve oyunu başlatmak için ateş butonuna basmalıdır. Easy seviyede engeller oyuncunun üzerine her bir saniyede bir satır atlayarak gelmektedir. Oyuncu tüm can haklarını kaybedene kadar aynı şekilde oyun devam eder. Oyun başladığında oyuncuya 3 can hakkı ve 3 silah hakkı verilmektedir. Oyuncunun can hakları devrede mavi 3 led ile temsil edilmektedir. Oyuncu her bir engele çarptığında bir can hakkı kaybetmekte ve ledlerden biri sönmektedir. Aynı şekilde oyuncunun, oyun başında 3 silah bulunmaktadır. Bu silah hakları kırmızı 3 led ile temsil edilmektedir. Oyuncu her ateş ettiğinde bir silah hakkı eksilmekte ve kırmızı ledlerden biri sönmektedir. Oyuncu her 14 engel atladığında bir can ve bir silah hakkı kazanmaktadır. Ancak oyuncu maksimum 3 can ve 3 silah hakkı bulundurabilmektedir. Oyuncu silah hakkı ve can hakkı kazandıkça sönmüş olan ledler tekrar yanmaktadır. Oyuncu S2 seçeneğini seçtiğinde yani Hard seviye zorluk seçtiği zaman başlangıçtaki gibi engeller başta her saniye bir satır atlamaktadır. Ancak her 10 saniye geçtiğinde engeller yüzde 20 hızlanmaktadır. Örneğin başta engellerin hızı 1000 ms yani 1 saniyedir. İlk 10 saniyeyi atladıktan sonra oyuncu engeller  $1000 \cdot 0,8 = 800$  ms hızına çıkmaktadır. Yani her 0,8 saniyede bir satır inmektedir. Bu hız 640 ms, 512 ms, 410 ms, 328 ms şeklinde artmaktadır. Engeller saniyede 100 ms'den daha hızlı hareket edemez. Oyuncu bir engele çarptığı zaman devreye bağlı olan buzzer ötmektedir. Oyuncu ışığı

düşük bir ortamda oyun oynuyorsa devreye bağlı olan ldr cihazı bunu algılar ve normalde siyah ekran üzerine olan beyaz çizim, tam tersine dönerek beyaz ekran üzerine siyah çizim yapmaktadır. Oyuncu her bir satır atladığında 2 zorluk seviyesi için de bir puan kazanmaktadır ve can hakları bittiği zaman ekrana skoru yansıtılmaktadır. Sonrasında oyuncu ateş butonuna basarak tekrar menüye dönebilmektedir.

Kullanılan Malzemeler:

- 1) Arduino Uno Mikrokontrolcü
- 2) SSD1306 OLED Ekran
- 3) 3 adet Kırmızı LED
- 4) 3 adet Mavi LED
- 5) 6 adet 220 ohm Direnç
- 6) 3 adet 10k ohm Direnç
- 7) 2 adet 1k ohm Direnç
- 8) 1 adet BC547 Transistör
- 9) 1 adet Buzzer
- 10) 1 adet Buton
- 11) 1 adet Potansiyometre
- 12) 1 adet Servo Motor
- 13) 3 adet 74HC595 Shift Register
- 14) 3 adet 7 Segment Kırmızı Display

## II. YÖNTEM

Kod içerisinde Adafruit-GFX kütüphanesi ve ssd1306 oled ekran için gerekli kütüphaneler eklemiştir. Kodun başlangıcında elektrik devremizde kullandığımız çeşitli cihazlar için gerekli bağlantı noktaları atanmış ve bu cihazlar setup() fonksiyonu içerisinde kullanım amaçlarıyla tanımlanmıştır. 7 segment display ekranında gösterilecek olan rakamlar için bir byte listesi oluşturulmuştur ve bu listenin içerisine 0. index'i 0 ve 9. index'i 9 olacak şekilde gerekli değerler yazılmıştır. Oyun içi can, silah, süre takibi, atlanan engel takibi için skor, oyuncunun atladığı satır takibi için skor, menüde olup olmadığını, oyunu kaybedip kaybetmediğini takip etmek gibi gerekli değişkenler hazırlanmış ve default halleri tanımlanmıştır. Kod içerisinde setup ve loop fonksiyonları hariç 11 adet fonksiyon tanımlanmıştır. Roketi çizmek ve silmek, engelleri çizmek ve silmek roketi sağa ve sola hareket ettirmek, menüyü çizmek, can kaybetme, oyun bitti, oyun başladı ve skor yazdırma fonksiyonlarıdır.

Roket ve engel çizim fonksiyonları gerekli koordinat değerlerini alır ve ekranın boyutlarına göre oluşturulmuş olan Blok büyüklüğünde birer dikdörtgen çizer. Ekran 128\*64 boyutlarındadır. Ancak biz ekranı 90 derece yatırarak kullandığımız için geniş olan kenar 16 uzunluk satırına dar olan kenar ise 8 genişlik sütununa bölünmüştür. Çizim fonksiyonu da bunu göz önünde bulundurarak bir hesaplama ile çizim yapmakta ve eğer ortam ışığı yüksek ise beyaz çizim yapmaktadır. Ortam ışığı düşük ise siyah çizim yapmaktadır.

Roket ve Engel silme fonksiyonları da aslında aynı roket ve engel çizme fonksiyonları gibi çalışmaktadır. Tek farkı çizim sırasında örneğin yüksek ışıktaki bulunan ortamda beyaz iken silme fonksiyonlarında siyah olmaktadır. Böylece örneğin engel bir satır alta geçmek istediğinde önce engelin bulunduğu koordinatlar silme fonksiyonuna gönderilir ve engelin bulundupu konum doğru boyutlarda siyaha boyanır. Sonrasında çizim fonksiyonu bir sonraki koordinatlara ters renkte çizim yapmaktadır.

Roketi sağa ve sola hareket ettirme fonksiyonlarında ise loop fonksiyonu içerisinde çalıştırılan oyun başla fonksiyonu içerisinde servo motor değeri okunur bu değer eğer 90'nın üstünde ise yani potansiyometre 50'nin üzerinde ise sağa servo motor değeri 90'nın altında ise yani potansiyometre 50'nin altında ise sola hareket etme fonksiyonları çağırır. Bu fonksiyonların işlevi basittir. Önce roket bulunduğu konumdan silinir. Roket tek satırda hareket edebildiği için sadece sütun değeri sağa hareket ediyorsa 1 arttırılır. Sola hareket ediyorsa 1 azaltılır. Roket pozisyonu değiştirildiği zaman roket çizme fonksiyonu çağırılmaktadır. //Menü çizim fonksiyonu eğer menüde miyiz değeri doğru ise loop fonksiyonu içerisinde çağırılır. Bu fonksiyon içerisinde zorluk seviyesi ile ilgili seçim için servo motor değeri okunarak aynı roket hareketinde olduğu gibi eğer servo motor değeri 90'nın üstündeyse yani potansiyometre 50'nin üstündeyse üst seçeneğe geçme eğer servo motor değeri 90'nın altındaysa yani potansiyometre 50'nin altındaysa alt seçeneğe geçme sağlanır ve ateş butonuna basıldığı zaman bulunulan seçenek ile oyuna başlanır ve menüde miyiz değeri yanlış olarak atanarak loop fonksiyonu içerisinde oyuna başla fonksiyonu çağırılarak oyuna başlanır. Can kaybetme fonksiyonu, oyuna başla fonksiyonu içerisinde oyuncu bir engelle çarpıştığı zaman çağırılır. İşlevinin bir kısmı oyuna başla kısmında çalıştırılır. Oyuncunun can haklarının bulunduğu değeri 0 olana kadar indirir. Her can hakkı kaybında buzzer bir defa çalıştırılır ve öter. Eğer oyuncu son canını da kaybettiyse oyun bitti fonksiyonunu çağırır.

Oyun bitti fonksiyonunda sadece oyun bitti değerine True atanır ve ekran tamamen temizlenerek Game Over Skor ve tekrar başlamak için fire butonuna basın yazılarını yazdırır. Oyuncu ateş butonuna bastığı zaman menüye geri döner.

Skor yazdırma fonksiyonu içerisinde skor değeri olarak alınır ve birler, onlar ve yüzler basamağına ayrılır. Sonrasında Shift Registerlara gerekli değerler atanarak, 7 Segment Display'ler üzerinde bu rakamların yansıtılması sağlanır.

Oyun başladı fonksiyonu, oyunun neredeyse tamamının döndüğü fonksiyondur. Öncelikle engelin çıkış zamanı ve koordinatları tanımlanır. Roket ve engel çizilerek devam edilir.

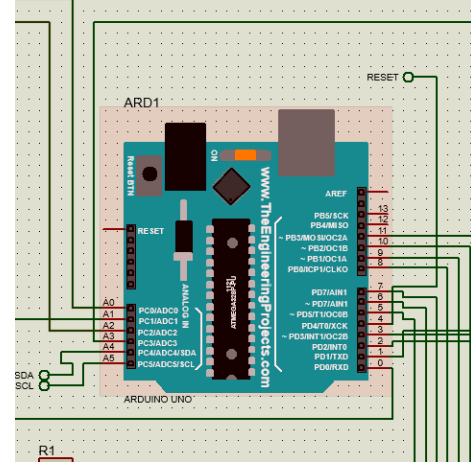


Fig. 1. Kullanılan mikrokontrolcü Arduino Uno Proteus 8 programında

Oyuncunun her bir Can hakkı için 3 ayrı şekilde Mavi LEDler için tanımlamalar yapılır ve kontrol sonucu olması gereken LEDler ışık verir. Aynı şekilde oyuncunun silah hakları için 3 ayrı LED için 3 ayrı ışıklandırma tanımı yapılır ve olması gereken şekilde seçilir. Sonrasında oyuncunun seçtiği zorluk seçeneği kontrol edilir. Eğer oyuncu Easy zorluk seviyesini seçtiyse Oyunda geçen zamandan son engelin oluşturulduğu zamanın çıkartıldığı ve oyunun bitip bitmediği kontrol etmediği bir döngü içerisine girer. Bu döngü içerisinde sürekli engel oluşturulur; çizilir ve silinir; çarpıp çarpmadığı kontrol edilir; roketin hareket yönü kontrol edilir ve ona göre roket çizimi ve silinme işlemleri yapılır; ateş butonu kontrol edilir ve ona göre silah ledleri ve engelin yok olup olmadığı kontrol edilir. Oyuncu eğer Hard zorluk seviyesini seçtiyse tek fark oyunda geçen her 10 saniyede bir engellerin hızının yüzde 20 artması için hız değeri her 10 saniyede bir 0.8 ile çarpılarak kendine atanır. Böylece daha önce de belirttiğimiz gibi 1000 ms – 800 ms – 640 ms – 512 ms gibi değişerek ilerler. Ancak asla 100 ms'nin altına düşmez.

### III. SONUÇ

Arduino IDE ortamında kod geliştirmeyi ve bu kodu simüle edebilmek için Proteus 8 programını kullanmayı öğrendik. Bu kodumuzu kullanarak ve Proteus 8 programında simüle ettiğimiz gibi gerekli devrelerimizi sağlayıp programdaki gibi doğru bağlantıları sağladığımız zaman cep boyutunda kullanabileceğimiz bir arcade oyunu hazırlamış olduk. Bu kodlama süreci boyunca bir çok farklı konuda bilgiler öğrendik. Kodumuzun algoritmasını geliştirirken bunu ekrana nasıl yansıtacağımız, kullanıcıyla nasıl interaktif bir şekilde iletişim kurabileceğimiz ve görsel olarak kullanıcının ilgisini çekecek bir oyun hazırlamış olduk. Böyle bir oyun hazırlayıp gerekli devreyi bir konsol tarzında hazırlandığı takdirde ilgi çekebilecek bir ürün olarak piyasaya sunulabilir.

### IV. DENEYSEL SONUÇLAR VE AKIŞ DİYAGRAMI

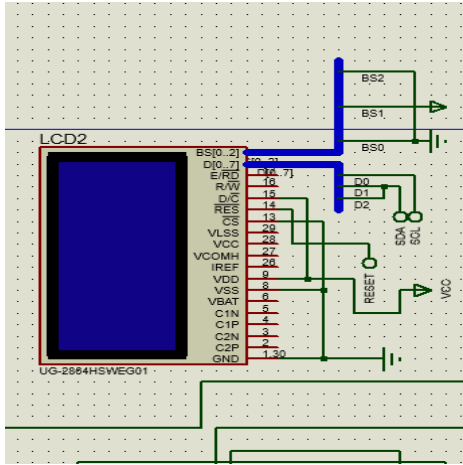


Fig. 2. SSD1306 OLED LCD monokrom ekran

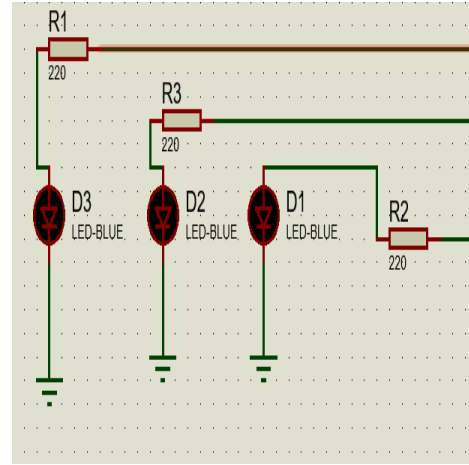


Fig. 5. Can hakları için kullanılan Mavi LEDler

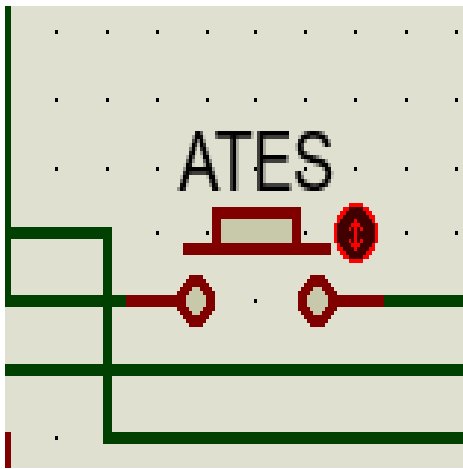


Fig. 3. Ateş etme butonu

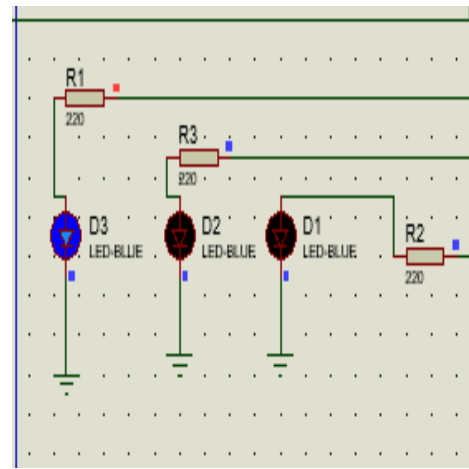


Fig. 6. Can hakkı 1 kalmış oyuncunun LEDleri

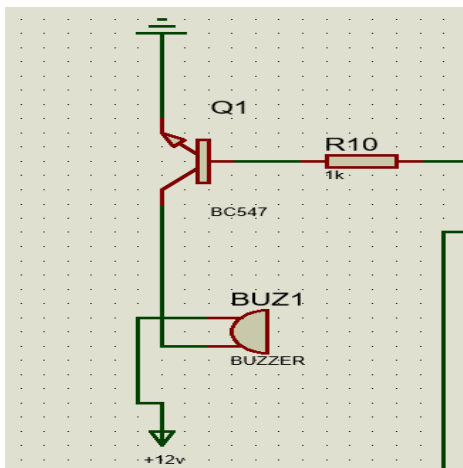


Fig. 4. Kullanılan buzzer

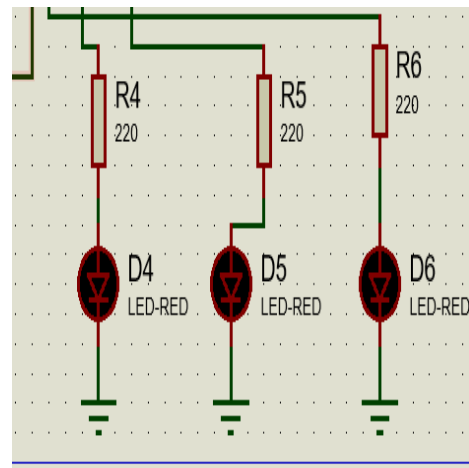


Fig. 7. Silah hakları için kullanılan kırmızı LEDler

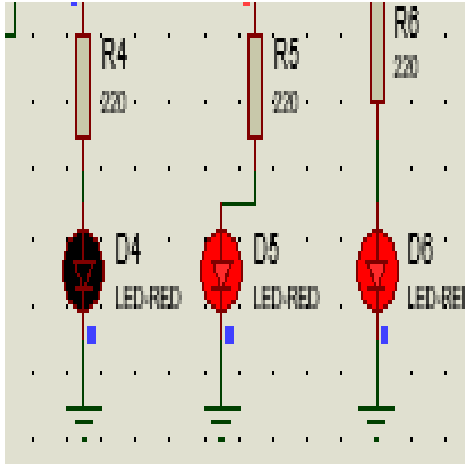


Fig. 8. Silah hakkı 2 kalmış oyuncunun LEDleri

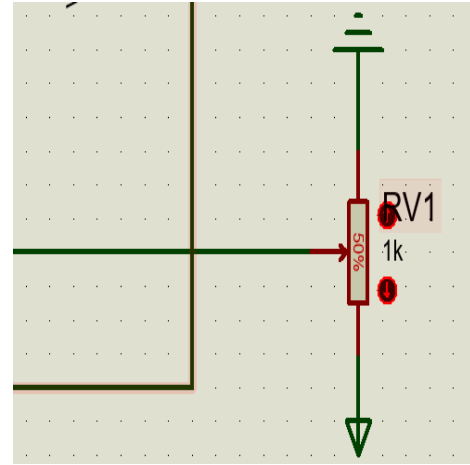


Fig. 11. Servo motor için kullanılan ölçüm cihazı potansiyometre

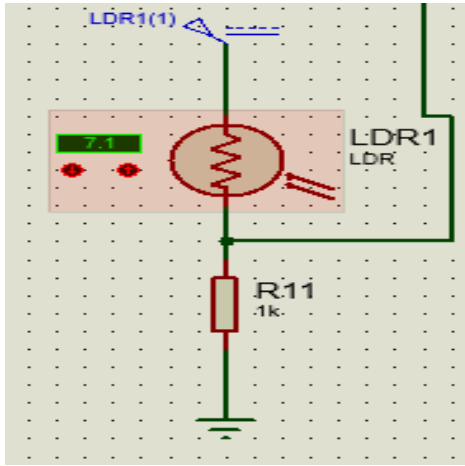


Fig. 9. Oyuncunun bulunduğu ortamın ışık seviyesini algılayan cihaz

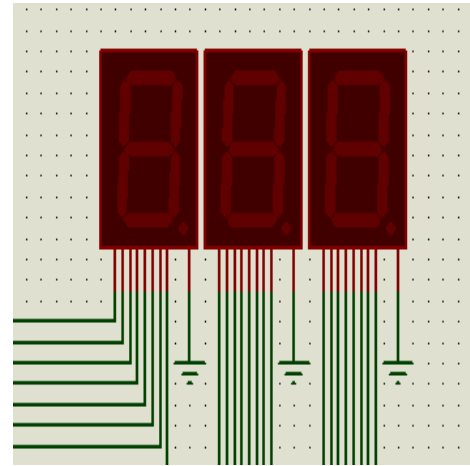


Fig. 12. Skor tablosunu oyuncuya yansıtmak için kullanılan tablo

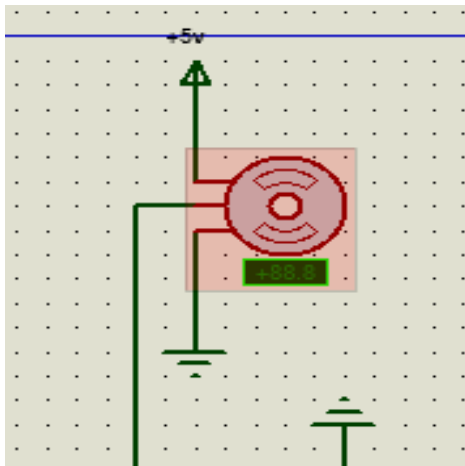


Fig. 10. Joystick için kullanılan servo motor

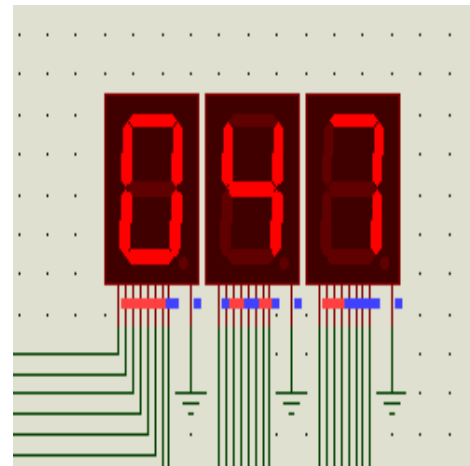


Fig. 13. Oyun içerisinde 47 puan alan oyununun skor tablosu

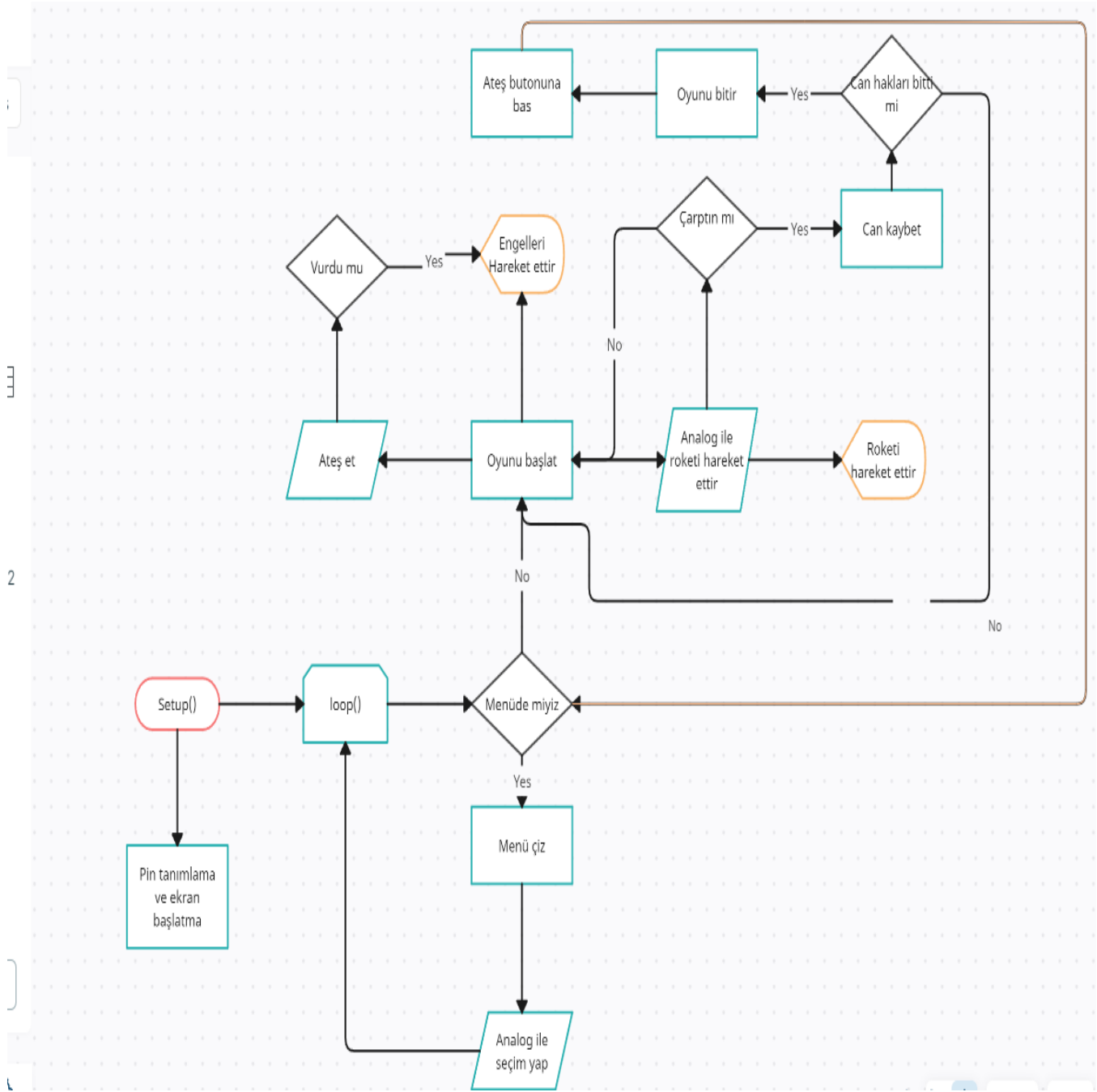


Fig. 14. Kod Akış Diyagramı

## V. KAYNAKÇA

- 1-<https://www.youtube.com/watch?v=jTt24W7zbpg>
- 2-<https://dev.to/projectiot123/ssd1306-oled-using-arduino-in-proteus-13nm>
- 3-<https://320volt.com/proteus-arduino-library/>
- 4-<https://microdigisoft.com/arduino-ds1307rtc-and-ssd1306-oled-display-in-proteus/>
- 5-<https://www.mekinfo.net/arduino-ve-proteus-kullanimi-i/>
- 6-[http://matematik.fef.duzce.edu.tr/Dokumanlar/matematik\\_fef/latex\\_otlarr.pdf](http://matematik.fef.duzce.edu.tr/Dokumanlar/matematik_fef/latex_otlarr.pdf)