

Arduino Uzay Oyunu

Umut YİĞİT
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi 2023
200201041

Betül BODUR
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi 2023
210201069

I. ÖZET

Bu dokümanda Programlama Laboratuvarı II dersimizin 2. projesi olan Arduino Uzay Oyunu Projesi için oluşturulan algoritmalar açıklanmıştır. Dokümanda problemin tanımı verilip bu problemler hakkında araştırmalar ve yöntemler sunulmuştur. Araştırmalar ve yöntemlerde bahsedilen fonksiyonların ayrıntılı açıklaması bulunmaktadır.

Index Terms—led,pin,fonksiyon,loop,segment,arduino

II. PROBLEM TANIMI

Bu projede, çeşitli sensörler kullanarak mikrodenetleyici tabanlı bir oyun makinesi geliştirmemiz istenmiştir. Oyun OLED ekran üzerinde tasarlanmalıdır. Oyunun bir başlangıç menüsü olması ve bu menü üzerinden zorluk seviyesi ve başlat butonlar yardımıyla seçilmelidir. Zorluk seviyesi 2 seviyeden oluşması gerekmektedir. 1. seviye platform her zaman saniyede 1 kare aşağı hareket etmelidir. 2. seviyede ise platform her 10 saniyede bir %20 hızlanarak hareket etmelidir. Oyun başladığında uzay aracı potansiyometre ile sağ ve sol olmak üzere kontrol edilmelidir. Uzay aracının, her biri 3 hak olmak üzere silah hakkı ve can hakkı bulunmaktadır ve bunlar led ile gösterilmektedir. Uzay aracı engele çaptığında; buzzer ile kullanıcıya uyarı vermelidir, bir can kaybetmektedir ve 3 saniye içinde dokunulmaz olmaktadır. Oyuncu can haklarının hepsini kaybettiğinde ana menüye geri dönmelidir. Oyuncu her satır atladığında skor puanı 7 segment display üzerinden gösterilmelidir. LDR ışık sensörü alınan veriler ile oyun ortam rengi değiştirilmelidir. Bu proje java programlama dili ile yazılmıştır

III. ARAŞTIRMALAR VE YÖNTEMLER

Bu projede öncelikle elektronik kısmının tasarlanması için simülasyon ortamı belirlenmesi gerekmektedir. Öncelikle çalışmaları Proteus üzerinden yapmaya karar verdik. Ancak burada **Simulation is not running in real time due to excessive CPU load. @00:00:02.105000 hatası** aldık. Bunun üzerine yeni kütüphane ekleme frame süresini kısma gibi işlemler uyguladık ve herhangi bir olumlu sonuca ulaşamadık.

Bu yüzden yeni simülasyon ortamı arayışına girdik. WOKWİ Online simülasyonu ile projeyi yapmaya karar verdik.

WOKWİ üzerinde projede gerekli olacak materyallerin bağlantısını yaptık. Projenin yazılım kısmı ile ilgili gerekli ön hazırlığı yaparak. Öncelikle ana menü tasarımını yaptık. Ancak ana menü'deki zorluk seviye ayarlama seçeneğinde sorun yaşadık. Arduinoda bulunan loop() fonksiyonu bir sonsuz bir while döngüsü gibi çalıştığı için sürekli ana menü'ye döndü. Buna çözüm olarak if yapısını kullanmaya ve parametre olarak boolean değerler olan true-false değerleri olmasına karar verdik. Bu sayede Ana Menü de bir seçenek seçildiğinde Ana Menü'yu oluşturan fonksiyon false değerini almakta ve o if bloğu çalışmadığından ikinci if bloğu olan zorluk seviyesi çalışmaktadır. Bu sayede ekranları tek tek sağlıklı bir şekilde görmeyi sağlamış olduk.

Gecikmeler için delay fonksiyonu kullanıldığında fazla bir şekilde geciktiği için delay fonksiyonlarını olabildiğince azalttık ve bu gecikme sorununu çözerek akıcı bir arduino uzay oyunu tasarladık.

IV. GELİŞTİRME ORTAMI

- Bu proje C programlama dili ile yazılmıştır.
- Wokwi - Online Arduino and ESP32 Simulator kullanıldı.
- Arduino IDE kullanıldı.
- Arduino Mega kullanıldı.

V. KOD BİLGİSİ

A. Fonksiyonlar

Programın fonksiyonları şu şekilde özetlenmiştir:

setup fonksiyonu:

Arduino başlatıldığında veya yeniden başlatıldığında ilk çalıştırılan kısımdır. Fonksiyonların çalışma ortamını hazırlar ve yeniden başlatmaya kadar tekrar çalışmamaktadır. Seri iletişimi başlatmak için **Serial.begin(9600)** fonksiyonu kullanılmıştır. Oyun oynanılıp, kaybedildiğinde tekrar ana menü açıldığı zaman oyunun sıfırdan başlaması

için değişkenlerin sıfırlanması için **restart()** fonksiyonu çağırılmıştır. Yukarı,aşağı,onay ve ateş butonları veriyi kullanıcıdan aldığı için pin tanımlanırken **INPUT** şeklinde tanımlanmıştır. Can için 3 adet, Silah için 3 adet led bulunmaktadır. Bu ledler de **setup()** fonksiyonu içerisinde tanımlanmıştır. Bu ledler bize çıkış vereceği için tanımlanırken **OUTPUT** şeklinde tanımlanmıştır. SSD 1306 OLED ekranımızda oluşabilecek sorun için voltaj gelip gelmediğini kontrol etmektedir. Herhangi bir hata sonucunda sanal terminal üzerinden kullanıcıya bildirmektedir. Ekran varsayılan yönü yataydır. Ancak yapılan projede yatay olması görsel açıdan küçük gözükmesinden dolayı ekranı dikey kullanmaya karar verilmiştir. Bu işlemi **display.setRotation(3)** ile yaparak ekranı 90 derece döndürülmüştür.

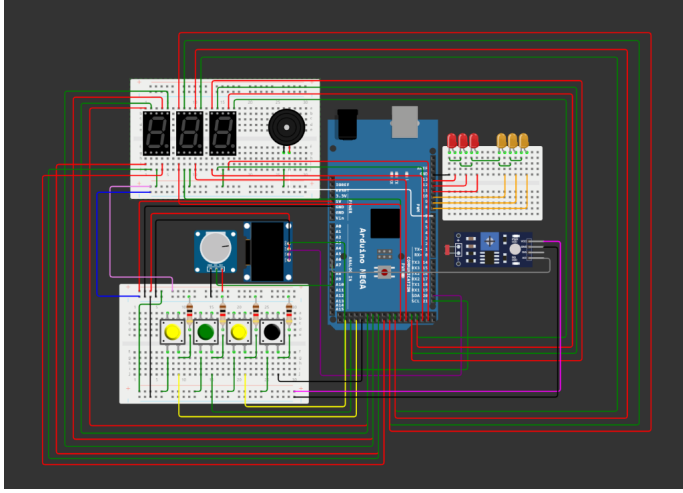


Fig. 1. Devre Şeması

loop fonksiyonu:

setup fonksiyonu çalıştırıldıktan sonra çalıştırılır ve sonsuz döngü olarak çalışmaktadır. Integer değer olarak ilk değeri 0 olan isikMiktari değişkeni **map()** fonksiyonunun değerini almaktadır. **map()** fonksiyonu LDR sensörden gelen verileri 0-100 arası sayılara çevirmektedir. **map fonksiyonun** belirlediği **isikMiktari** adlı değişkenin değeri 50'den küçük değil ise **invertDisplay()** fonksiyonu ile ekranın renklerini tersine çevirmektedir.

loop() fonksiyonu sonsuz bir döngü olduğu için burada yukarı,aşağı ve onay butonlarına ait fonksiyonlar tanımlanmıştır. Boolean tipinde olan **baslangic_ekrani**, **zorluk_ekrani** ve **oyun_ekrani** **restart()** fonksiyonunda atanan değerleri sırasıyla true, false ve false'tur. **loop()** fonksiyonun sonsuz bir döngü olarak çalışmasından dolayı sürekli ilgili fonksiyonlara girmemesi için **if yapısı** kullanılmıştır. Eğer değişkenlerin değeri true ise ilgili fonksiyon çalışmaktadır.

butonSifirla fonksiyonu:

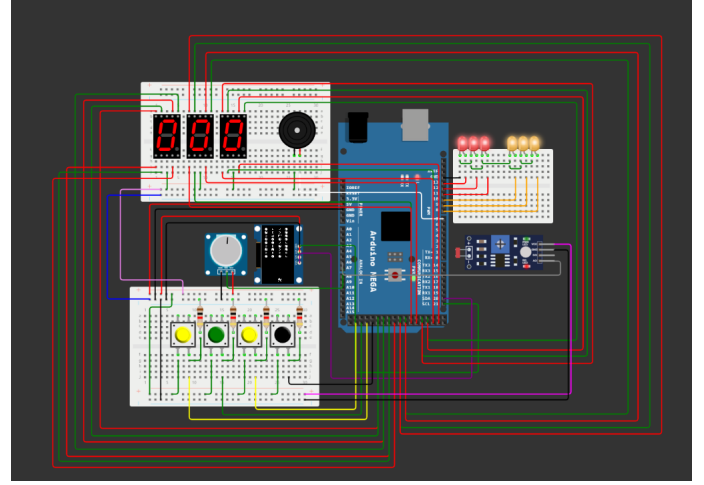


Fig. 2. Devre Şeması

Butonların kontrolü için tanımlanmıştır. Integer tipinde tanımlanan **sag_basildi** , **sag_cekildi** , **sol_basildi** , **sol_cekildi** , **ates_basildi** , **ates_cekildi** , **ok_basildi** ve **ok_cekildi** değişkenlerinin başlangıç değerleri 0 olarak atanmıştır. Ancak butonların kullanılması durumunda tekrar sıfırlanması gerekmektedir. Bu fonksiyon bu değişkenlerin sıfırlanmasını sağlamaktadır.

restart fonksiyonu:

Oyunun sıfırlanması için oluşturulmuştur. Eğer oyun ekranı aktifse **buzzer'dan 500ms aralıklarla tone()** fonksiyonu ile 2 çeşit olmak üzere 3 nota gelmektedir. Ve **noTone()** fonksiyonu ile buzzer'ın çıkış vermeyi kesecektir. Bu fonksiyon sıfırlama veya yeniden başlatma için kullanılacağından oyun başlatıldığı ilk değerleri olan varsayılan değerlerine döndürülmelidir. Sırasıyla şu işlemler gerçekleşmektedir : **butonSifirla()** fonksiyonu ile butonların değerleri sıfırlanmıştır, **baslangic_ekrani**, **zorluk_ekrani** ve **oyun_ekrani** değerleri sırasıyla true, false ve false değerleri atanmıştır, **dokunulmazlikAlindi** , **zorluk_secildi**, **seviye1**, **seviye2**, **skorBaslat** değerleri false olarak atanmıştır, **kursunSayisi**, **canSayisi**, **fram_suresi**, **platformSaniye** ve **dokunulmazlikSaniye** değerleri oyunun varsayılan değerlerine döndürülmüştür. Son olarak bir for döngüsü içerisinde **ekranKontrol** matrisinde bulunan bütün değerler 4 olarak atanmıştır. **items** matrisin içindeki değerler 4 olarak atanmıştır. Bu 4 ataması sayesinde engel özellik olmayıp her şey sıfırlanmış olmaktadır.

sec fonksiyonu:

51. pine bağılı olan butondur. Bu buton ile **onaylama işlemi** gerçekleşmektedir. Butona ilk basıldığında **ok_basildi değişkeni** 1 olmaktadır.

Basıp çekme durumu için butona basılmama durumunda ve **ok_basildi değişkeni** 1 ise **ok_cekildi değişkeni** 1 değerini almaktadır.

Eğer ki **ok_basildi ve ok_cekildi** değişkenlerinin değeri 1 ve y değeri 0 ise ve baslangic_ekrani değeri true olduğu durumda **butonSifirla() fonksiyonu** çağrılmaktadır. baslangic_ekrani değeri false olup,zorluk_ekrani değeri true olmaktadır.

Eğer ki **ok_basildi ve ok_cekildi** değişkenlerinin değeri 1 ve y değeri 1 ise ve baslangic_ekrani değeri true ve zorluk_secildi değeri true olduğu durumda **butonSifirla() fonksiyonu** çağrılmaktadır. baslangic_ekrani değeri false ve zorluk_ekrani değeri false olup oyun_ekrani değeri true olmaktadır.

Eğer ki **ok_basildi ve ok_cekildi** değişkenlerinin değeri 1 ve zorluk_ekrani değeri true olduğu durumda **butonSifirla() fonksiyonu** çağrılmaktadır. Eğer y değeri 0 ise seviye1,seviye2 ve zorluk_secildi boolean değişkenleri sırasıyla true,false ve true değerlerini almaktadır. Ve y değeri başlangıç ekranında üstte olması için 0 olmaktadır. Eğer y değeri 1 ise seviye1,seviye2 ve zorluk_secildi boolean değişkenleri sırasıyla false,true ve true değerlerini almaktadır. Ve y değeri 0 olmaktadır. y kontrolleri bittiğinde baslangic_ekrani değeri true ve zorluk_ekrani değeri false olmaktadır.

kursunSayisi değişkeninin değeri 0 değil ise **kursunAt() fonksiyonu** çağrılmaktadır. Ve sonrasında **butonSifirla() fonksiyonu** çağrılmaktadır.

sol fonksiyonu:

49. pine bağılı olan butondur.Bu buton ile **menüde yukarı yönlü hareket** gerçekleşmektedir. Butona ilk basıldığında **sol_basildi değişkeni** 1 olmaktadır. Basıp çekme durumu için butona basılmama durumunda ve **sol_basildi değişkeni** 1 ise **sol_cekildi değişkeni** 1 değerini almaktadır. Eğer ki **sol_basildi ve sol_cekildi** değişkenlerinin değeri 1 ise y değeri sıfırlanmaktadır ve **butonSifirla() fonksiyonu** çağrılmaktadır.

sag fonksiyonu:

53. pine bağılı olan butondur. Bu buton ile **menüde aşağı yönlü hareket** gerçekleşmektedir. Butona ilk basıldığında **sag_basildi değişkeni** 1 olmaktadır. Basıp çekme durumu için butona basılmama durumunda ve **sag_basildi değişkeni** 1 ise **sag_cekildi değişkeni** 1 değerini almaktadır. Eğer ki **sag_basildi ve sag_cekildi** değişkenlerinin değeri 1 ise y değeri 1 olmaktadır ve **butonSifirla() fonksiyonu** çağrılmaktadır.

ates fonksiyonu:

47. pine bağılı olan butondur. Bu buton ile **oyun içindeki ateş etme eylemi** gerçekleşmektedir. Butona ilk basıldığında **ates_basildi değişkeni** 1 olmaktadır. Basıp çekme durumu için butona basılmama durumunda ve **ates_basildi değişkeni** 1 ise **ates_cekildi değişkeni** 1 değerini almaktadır. Eğer ki **ates_basildi ve ates_cekildi** değişkenlerinin değeri 1 olduğu durumda kursunSayisi değişkeninin değeri 0 değil ise **kursunAt() fonksiyonu** çağrılmaktadır. Ve sonrasında **butonSifirla() fonksiyonu** çağrılmaktadır.



Fig. 3. Ateş Etme

kursunAt fonksiyonu:

Oyun içinde ateş etme butonuna basıldığında çalışan fonksiyondur. Ateşin gideceği x ve y değerleri belirlenmektedir. Başlangıç değeri 0 olan kursun_x ,kursun_y roket_x değerleri ile işlemler yapılmıştır. kursun_x değeri , **roket_x / 8** değerinin sonucu atanmıştır. kursun_y, 15 olarak atanmıştır.Her **kursunAt() fonksiyonu** çağrıldığında kursunSayisi 1 azaltılmaktadır.ekranKontrol matrisindeki kurşunun gittiği doğrultuda 3 değerine sahip meteor varsa o meteor artık vurulmuş meteor olur ve ekranKontrol matrisindeki değeri 6 olmaktadır. Eğer ki ekranKontrol matrisindeki kurşunun gittiği doğrultuda 6 değerine sahip vurulmuş meteor varsa o matristeki değer 4 olur yani o konumda bir boşluk oluşur.

upToDown fonksiyonu:

Bu fonksiyon ile oyun sırasında meteor,can, ateş hakkı vb. nesnelerin alt satıra sırayla inmesi için oluşturulmuştur. Bir for döngüsü içinde **ekranKontrol[i][j] = ekranKontrol[i-1][j]** işlemi kullanılarak her bir satırı alt satırına kopyalamaktadır.satir sayısı 16 göre modu 0 ise satir değışkeni 0 olmaktadır.16 ya 4'lük items dizisine 4 değerleri atanmaktadır.Eğer items dizisinde bulunan değeri 1 ise meteorSayisi 0 ile çarpılmaktadır değilse 1 ile çarpılmaktadır. Eğer items dizisindeki bir değeri 4 ise random bir şekilde 2 veya 3 değeri atanmaktadır.items dizisinde bulunan değeri 3 ise meteorSayisi 1 ile çarpılmaktadır değilse 0 ile çarpılmaktadır. Eğer meteorSayisi, 14'e ulaştıysa bir satır can hakkı,bir satır

silah hakkı gelmektedir.

oyunEkrani fonksiyonu:

Atış hakkı 3 tane olması sebebiyle 3 tane led bulunmaktadır.For döngüsü içerisinde kursunSayisi kadar led yanması için if bulunmaktadır.forda bulunan i değişkeni ; 0 ise 8.pin , 1 ise 9.pin , 2 ise 10.pin aktif olmaktadır.loop() fonksiyonundan dolayı sürekli kontrol edilmektedir.Eğer kursunSayisi 0 ise 8,9 ve 10. pine bağlı olan ledler LOW değerini almaktadır.For döngüsü içerisinde canSayisi kadar led yanması için if bulunmaktadır.forda bulunan i değişkeni ; 0 ise 11.pin , 1 ise 12.pin , 2 ise 12.pin aktif olmaktadır.loop() fonksiyonundan dolayı sürekli kontrol edilmektedir.Eğer canSayisi 0 ise 8,9 ,10,11,12 ve 13. pine bağlı olan ledler LOW değerini almaktadır.

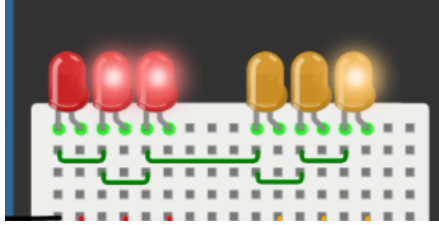


Fig. 4. Can ve Silah Hakkı

Bu durumda oyun bitmiş demektir bu yüzden oyunun ana menüye düşmesi için restart() fonksiyonu çağrılmaktadır.ates() fonksiyonu çağrılmaktadır.roket_x değeri map() fonksiyonu sonucunun 8 katı olarak atandı. roket_y değeri 120 olarak atanmıştır.Roketi görsel açıdan oluşturmak için piksel görsel drawBitmap() fonksiyonu ile roketin konumunu tutan roket_x ve roket_y değişkenleriyle belirtilen konuma yerleştirilmiştir.delay() fonksiyonunda drop sorunlarından dolayı millis() fonksiyonu kullanılmıştır.millis() fonksiyonu ile 3 saniye dokunulmaz olma isteri gerçekleştirilmiştir.Her bir milisaniyede dokunulmazlikSaniye değişkeni 1 artmaktadır.Eğer dokunulmazlikSaniye değeri 3 ise dokunulmazlikAlindi false değerini almaktadır.frame_suresi başlangıç değeri 1000 ms'dir. Eğer bir sonrrai loop'a kadar frame_suresi'ne eşit veya fazla zaman geçtiyse upToDown() fonksiyonu çağrılmaktadır.skorBaslat değeri; true ise score değeri 1 artmakta , false ise score değeri artmamaktadır.Birler,onlar ve yüzler basamağını 7 segment display'e yazdırmak için showScore() fonksiyonu her bir basamak için ayrı ayrı çağrılmıştır.

```
showScore(score % 10, 0); Birler Basamağı  
showScore((score/10) % 10, 1); Onlar Basamağı  
showScore(score/100, 2); Yüzler Basamağı
```

seviye1 ve seviye2'deki platform hızı farklı olduğu için bunu ayırt etmek için platformSaniye bulunmaktadır.platformSaniye her bir döngüde 1 artmaktadır.Eğer platformSaniye 10'a

eşitse ve seviye2 true ise frame_suresi %20 oranında azalmaktadır.platformSaniye, 0'a eşitlenmektedir.Bu sayede zorluk seviyesini 2.seviye seçen oyuncu her 10 saniyede %20 hızlanan bir platformda oynamaktadır.Ekrana nesneler koymak için iç içe for döngüsü kullanılmıştır.ekranKontrol dizisi içinde bulunan nesne tipi numarasına göre oyunIcons dizisinden nesneler alınıp drawBitmap() fonksiyonu ile ekrana eklenmektedir.Roketin nesnelere çarpma, alma gibi işlemleri bir switch-case içerisinde kontrol edilmiştir.ekranKontrol[15][roket_x] ' te bulunan değer:

0 ise canSayisi, 3'ten küçük ise canSayisi 1 artmaktadır.

1 ise kursunSayisi,3'ten küçük ise kursunSayisi 1 artmaktadır.

2 ise dokunulmazlikAlindi değeri true değeri alıp,dokunulmazlikSaniye değeri 0 olarak atanmıştır.

3 ise meteordur

6 ise hasar almış meteordur. dokunulmazlikAlindi değeri false ise dokunulmazlikAlindi değeri true olmaktadır.dokunulmazlikSaniye değeri 0 olarak atanmıştır ve canSayisi 1 azalmaktadır.Kullanıcıya can kaybettiğini veya meteora çarptığını bildirmek için tone() fonksiyonu çalışmaktadır.

4 ise noTone() fonksiyonu çalışmaktadır.

ekranKontrol[15][roket_x/8] = 4 ile loop içerisinde alınan item tekrar alınmasın diye tanımlanmıştır.Ekranda görüntülerin gözükmesi için **display()** fonksiyonu kullanılmaktadır.Ve **loop()** fonksiyonunun sonsuz döngüde çalışmasından dolayı görüntülerin üst üste gelmemesi için **clearDisplay()** fonksiyonu ile ekran temizlenmektedir.

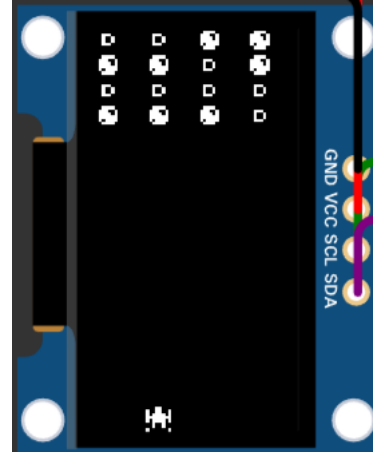


Fig. 5. Oyun Ekranı

zorlukEkrani fonksiyonu:

baslangicEkrani() fonksiyonunda olduğu gibi burada da zorluk ekranı tasarımı yapılmıştır ve yine aynı şekilde oluşturulan piksel görüntüler **drawBitmap()** fonksiyonu ile Oled ekranımızda gösterilmektedir.y değeri 0 olduğunda **1. seviyenin seçili** olduğu piksel görüntüler görüntülenmektedir.Eğer y değeri 1 ise yani **2. seviye seçili** olarak gözükmektedir.Ekranda görüntülerin

gözükmesi için **display()** fonksiyonu kullanılmaktadır. Ve **loop()** fonksiyonunun sonsuz döngüde çalışmasından dolayı görüntülerin üst üste gelmemesi için **clearDisplay()** fonksiyonu ile ekran temizlenmektedir.

baslangicEkrani fonksiyonu:

Ana menü tasarımı için oluşturulan piksel görüntüler **drawBitmap()** fonksiyonu ile Oled ekranımızda gösterilmektedir. Eğer y değeri 0 ise yani **Zorluk Seviyesi** seçili olarak gözükmektedir ve onun seçili olduğunu belirtecek piksel görüntüler görüntülenmektedir. Eğer y değeri 1 ise yani **Başlat** seçili olarak gözükmektedir ve onun seçili olduğunu belirtecek piksel görüntüler görüntülenmektedir. Ekran görüntülerin gözükmesi için **display()** fonksiyonu kullanılmaktadır. Ve **loop()** fonksiyonunun sonsuz döngüde çalışmasından dolayı görüntülerin üst üste gelmemesi için **clearDisplay()** fonksiyonu ile ekran temizlenmektedir.



Fig. 6. Zorluk Ekranı

showScore fonksiyonu:

Bu fonksiyonda skor tablosunun seven segment display üzerinde gösterilmesi için oluşturulmuştur. 2 parametre almaktadır. Integer tipinde rakam ve basamak adlı parametrelerdir. rakam ;
0 ise G segmenti hariç hepsi çalışmaktadır.
1 ise B ve C segmentleri çalışmaktadır.
2 ise C ve F segmentleri çalışmamaktadır.
3 ise E ve F segmentleri çalışmamaktadır.
4 ise A, D ve E segmentleri çalışmamaktadır.
5 ise B ve E segmentleri çalışmamaktadır.
6 ise B segmenti çalışmamaktadır.
7 ise A, B ve C segmentleri çalışmaktadır.
8 ise segmentlerin hepsi çalışmaktadır.
9 ise E segmenti çalışmamaktadır.

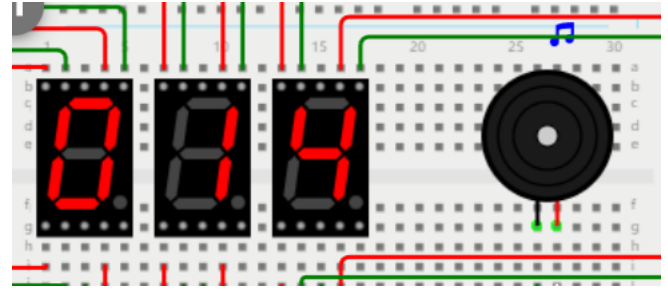


Fig. 7. Skor Ekranı ve Buzzer

B. Yalancı kod

setup() Fonksiyonu

Seri iletişim için saniyede gönderilen ve alınan bit sayısı 9600 olarak belirlenir ve seri iletişim başlar.

restart() fonksiyonu çağrılır. (Fonksiyon kendi bölümünde açıklanacaktır.)

Arduino üzerinde kullanılan pinler input ve output olarak ayarlanır.

Eğer ekran açılmaz ise
ekrana "SSD1306 allocation failed" yazısını yaz.
sonsuz döngüye gir (programın devamının derlenmemesi için)

ssd1306 OLED Ekranı dik konumda kullanabilmek için rotation 3 olarak ayarlanır.

Ekran temizlenir.

loop() Fonksiyonu

LDR ışık sensöründen algılanan ışık miktarı map() fonksiyonu ile 0 ile 100 arasında bir değere çevrilir ve isikMiktari değişkenine atanır.

Eğer isikMiktari < 50 ise
renkleri ters çevirme
Değilse
renkleri ters çevir

sol(), sag() ve sec() fonksiyonları çağrılır. (Butonlara basma olayları için)

Eğer baslangic_ekrani değeri true ise
baslangicEkrani() fonksiyonu çağrılır.

Eğer zorluk_ekrani değeri true ise
zorlukEkrani() fonksiyonu çağrılır.

Eğer oyun_ekrani değeri true ise
oyunEkrani() fonksiyonu çağırılır.

butonSifirla() Fonksiyonu

Butonların basılma ve çekilme durumlarını kontrol etmek için oluşturulan değişkenlere 0 (sıfır) değeri atanır. (Basıldığı zaman 1 değerleri atanmaktadır.)

restart() Fonksiyonu

Eğer oyun_ekrani true ise (yani oyun ekranı açık ise) Kaybetme müziği için ayarlanan şekillerde notalar Buzzer tarafından oynatılır.

butonSifirla() fonksiyonu çağırılır.

Oyunun tekrardan ana menüye dönmesi için baslangic_ekrani değişkenine true değeri atanır ve ekran için oluşturulan diğer değişkenlere false değeri atanır.

Oyunun baştan başlaması için gerekli değişkenler başlangıç değerlerine döndürülür.

Oyunun akıcı bir şekilde ekranda gösterilmesi için oluşturulan items[][] ve ekranKontrol[][] dizilerinin içleri 4 integer değeri ile doldurulur. (4 boşluk anlamına gelmektedir.) Diziler de bu sayede sıfırlanmış oldu.

sec() Fonksiyonu (51. pin)

Butona basılma ve çekilme durumları için oluşturulan değişkenler 1 değerini alır.

Eğer butona basıp çektiysem, y değeri 0 ve ana menüde isem Butonları sıfırla ve zorluk ekranını aç.

Eğer butona basıp çektiysem, y değeri 1, başlangıç ekranı açık ve zorluk seçildiyse Butonları sıfırla ve oyun ekranını aç.

Eğer butona basıp çektiysem ve zorluk ekranı açık ise Butonları sıfırla ve butona basıldığı andaki y değerine göre seviye1 veya seviye2 değişkenine true değerini ver.

sol() Fonksiyonu (49. pin)

Butona basılma ve çekilme durumları için oluşturulan değişkenler 1 değerini alır.

Eğer butona basıp çektiysem y değerine 1 değeri atanır ve butonlar sıfırlanır.

sag() Fonksiyonu (53. pin)

Butona basılma ve çekilme durumları için oluşturulan değişkenler 1 değerini alır.

Eğer butona basıp çektiysem y değerine 0 değeri atanır ve butonlar sıfırlanır.

ates() Fonksiyonu (47. pin)

Butona basılma ve çekilme durumları için oluşturulan değişkenler 1 değerini alır.

Eğer butona basıp çektiysem Eğer kursunSayisi 0 (sıfır) değilse kursunAt() fonksiyonunu çağır. Butonları sıfırla.

kursunAt() Fonksiyonu

Kurşunun başlangıç koordinatları roketin konumuna göre belirlenir. kursunSayisi bir azaltılır.

Eğer kurşun hiç vurulmamış meteora denk gelirse Meteor hasarlı meteora dönüşür. (ekranKontrol matrisindeki değeri 6 olur.)

Değilse Eğer kurşun hasarlı meteora denk gelirse Meteor kaybolur. (ekranKontrol matrisindeki değeri 4 olur.)

upToDown() Fonksiyonu

Ekranda gösterilen itemler bir alta kaydırılır.

Eğer satir == 16 ise satir değeri sıfırlanır. items matrisi temizlenir.

Ekranda gösterilecek itemler bir döngü içerisinde items matrisine kayıt edilir.

(Her 14 meteorda bir silah ve can matris içerisinde kayıt edilir.)

Ekranda gösterilen itemlerin kayıt edildiği ekranKontrol matrisinin en üstüne yeni itemler ve boşluklar gelir.

ekranKontrol matrisinden okunan itemler bir döngü içerisinde OLED ekranda gösterilir.

satir değeri bir artırılır.

Eğer satir == 16 ise (itemler roket ile aynı satıra geldiyse) skorBaslat değişkenini true yap ve skor sayımını başlat.

oyunEkrani() Fonksiyonu

Kurşun sayısı ve Can sayısını gösteren ledler (hak var ise) yanar.

Eğer canSayisi == 0 ise
Can haklarını gösteren ledler söner.
restart() fonksiyonu çağırılır. (Oyun yeniden başlar)

ates() fonksiyonu çağırılır.

Roketin x ve y konumuna göre OLED ekranda roket gösterilir.

Eğer 1 Saniye (1000 ms) geçtiyse
dokunulmazlikSaniye değeri 1 artırılır.

Eğer dokunulmazlikSaniye == 3 ise
dokunulmazlikAlindi değişkenine false değeri atanır. (3 sn geçti)

Eğer frame_suresi kadar zaman geçtiyse
upToDown() fonksiyonu çağırılır.

Eğer skorBaslat == true ise
score değeri 1 artırılır.

showScore() fonksiyonuna score değerinin birler, onlar ve yüzler basamağı
yollanarak fonksiyon çağırılır.

platformSaniye değeri 1 artırılır.

Eğer platformSaniye ve seviye2 değişkenleri true ise
frame_suresi platformSaniye değişkeni sıfırlandı.

ekranKontrol matrisinden okunan itemler bir döngü içerisinde OLED ekranda gösterilir.

Eğer roket can hakkı aldıysa ve canSayisi < 3 ise
canSayisi değeri 1 artırılır.

Eğer roket kursun hakkı aldıysa ve kursunSayisi < 3 ise
kursunSayisi değeri 1 artırılır.

Eğer roket dokunulmazlik aldıysa
dokunulmazlikAlindi değişkeni true yapılır.
dokunulmazlikSaniye değeri sıfırlanır. (3 sn baştan başlar)

Eğer roket meteora çarptıysa
Eğer dokunulmazlikAlindi false ise
canSayisi 1 azaltılır

dokunulmazlikAlindi değeri true yapılır.
dokunulmazlikSaniye değeri sıfırlanır. (3 sn baştan başlar)

Buzzer meteora çarptığı için ses çıkarır.

Eğer roket boşlukta ise
Buzzer susturulur.

Roketin bulunduğu konum boşluk yapılır. (loop içerisinde tekrar aynı item alınmasın diye)

OLED ekran yenilenir.

zorlukEkrani() Fonksiyonu

y değişkeninin değerine göre ekranda seçilen buton beyaz veya siyah (LDR değerine göre) rengini alır.

baslangicEkrani() Fonksiyonu

OLED ekranda alt ve üst tasarımlar gösterilir.

y değişkeninin değerine göre ekranda seçilen buton beyaz veya siyah (LDR değerine göre) rengini alır.

showScore() Fonksiyonu

Fonksiyona parametre olarak gelen rakam ve basamak değerine göre 3 adet seven segment display üzerinde score değeri kullanıcıya gösterilir.

VI. İSTATİSTİK

Kullanılan kütüphaneler ve kullanım amaçları:
stdlib.h

ArrayList oluşturmak,kullanıcıdan veri almak için kullanılmıştır.

SPI.h

Yazı tipi, rengi gibi olayları düzenlemek için kullanılmıştır.

Wire.h

Buton, görüntü , çerçeve,.. gibi arayüzü oluşturmak için gerekli nesneleri oluşturmak için kullanılmıştır.

Adafruit_GFX.h

URL adresindeki veriye erişmek için kullanılmıştır.

Adafruit_SSD1306.h

URL’de bulan veriyi okumak için kullanılmıştır..

- [30] <https://reference.arduino.cc/reference/en/language/functions/random-numbers/random/>
- [31] <https://rcinfofboard.wordpress.com/2014/08/07/the-arduino-program-initial-pseudocode-algorithm/>

VII. SONUÇLAR

Sonuç olarak, isterlerin tamamını yerine getirerek arduino gibi mikroişlemcilerin mantığını uygulamalı olarak kavramış olduk.Ve arduino’nun elektronik bağlantılarını nasıl yapacağımız öğrendik.Ayrıca C dilinde arduino kodlamamıza yardımcı olan Adafruit_GFX.h ve Adafruit_SSD1306.h kütüphanelerine hakim olduk.

REFERENCES

- [1] <https://devreyakan.com/arduino-i2c-ile-oled-ekran-kullanimi/>
- [2] <https://www.youtube.com/watch?v=UPupkt5z52A>
- [3] <https://www.youtube.com/watch?v=fZ3YhvOWwEk>
- [4] <https://microdigisoft.com/arduino-ds1307rtc-and-ssd1306-oled-display-in-proteus/>
- [5] <https://docs.wokwi.com/parts/wokwi-7segment>
- [6] <https://maker.robotistan.com/arduino-ldr-devresi/>
- [7] <https://gelecegiyazanlar.turkcell.com.tr/konu/egitim/arduino-201/milliss-fonksiyonu-nedir-ve-nerelerde-kullanilir>
- [8] <https://www.youtube.com/watch?v=8KXpvh48Go>
- [9] <https://www.youtube.com/watch?v=fcGXOmyjN2I>
- [10] <https://www.youtube.com/watch?v=iqYiptHhT6U>
- [11] <https://www.youtube.com/watch?v=s5vj6FkQKGg>
- [12] <https://www.youtube.com/watch?v=xAFnED4UZMM>
- [13] <https://www.youtube.com/watch?v=pyD9jdVeZGI>
- [14] <https://www.youtube.com/watch?v=cURh2-dTuI0>
- [15] <https://www.youtube.com/watch?v=SfvvKesO74Q>
- [16] <https://www.youtube.com/watch?v=0KGMFhFQ0YY>
- [17] https://www.youtube.com/watch?v=fgbMi1VT_qU
- [18] <https://www.youtube.com/watch?v=IQW6eNhCppA>
- [19] https://www.youtube.com/watch?v=_RaMTyD3UJA
- [20] <https://www.youtube.com/watch?v=i8FE4J1wvr8>
- [21] <https://www.instructables.com/Arduino-and-the-SSD1306-OLED-I2C-128x64-Display/>
- [22] <https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/>
- [23] <https://roboticsbackend.com/arduino-setup-loop-functions-explained/>
- [24] [https://github.com/ArduinoHocam/ARDUINO-OLED-MENU-DESIGN/blob/master/CODES\(KODLAR\)](https://github.com/ArduinoHocam/ARDUINO-OLED-MENU-DESIGN/blob/master/CODES(KODLAR))
- [25] <https://mongoose-os.com/docs/mongoose-os/api/arduino/arduino-adafruit-ssd1306.md>
- [26] <https://www.tutorialspoint.com/tone-and-notone-in-arduino>
- [27] <https://docs.wokwi.com/guides/serial-monitor>
- [28] <https://www.electricaltechnology.org/2020/05/7-seven-segment-display.html>
- [29] <https://arduino.stackexchange.com/questions/72229/is-there-an-adafruit-function-to-invert-the-colors-of-my-screen-for-arduino-lang>