

**KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



ARDUİNO İLE PONG OYUNU

TASARIM PROJESİ

Furat AÇIK

2015-2016 GÜZ DÖNEMİ

**KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

ARDUİNO İLE PONG OYUNU

TASARIM PROJESİ

Furat AÇIK

Bu projenin teslim edilmesi ve sunulması tarafımda uygundur.

Danışman :Yrd. Doç.Dr. Hüseyin PEHLİVAN

2015-2016 GÜZ DÖNEMİ



IEEE Etik Kuralları IEEE Code of Ethics



Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriye kabul etmek ve eleştiriye yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayırimcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

ÖNSÖZ

Karadeniz Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü'nde Tasarım Projesi olarak belirlenen temaya uygun olarak Arduino ile Pong Oyunu tasarlanmıştır.

Projenin seçilme amacı günümüz teknolojilerini göz önüne aldığımızda oyunlar insanların hayatında çok önemli bir yer edinmiştir. İnsanlar genelde karşılıklı olarak oynanabilen oyunları tercih etmektedir ki günümüzde yapılan mobil oyunların bir çoğu da online karşılıklı oynanabilmektedir. Bu projede karşılıklı olarak oynanabilecek şekilde tasarlanıp, oyuncuların potansiyometreler ile karşılıklı olarak birbirleriyle oynayabilme fırsatı sunmaktadır.

Eğitim ve öğretim hayatımız boyunca bana destek veren aileme sonsuz teşekkür ederim.

Çalışmam boyunca bana yardımını esirgemeyen arkadaşım Murat HAS ve danışman hocam YRD. DOÇ. DR Hüseyin PEHLİVAN'a teşekkürü bir borç bilirim.

Furat AÇIK
Trabzon-2015

İÇİNDEKİLER

	Sayfa No
IEEE ETİK KURALLARI	II
ÖNSÖZ	III
İÇİNDEKİLER	IV
ÖZET	V
KISALTMALAR VE SEMBOLLER	VI
1. GENEL BİLGİLER	1
1.1. Giriş	1
1.2. Projede kullanılan araçlar	1
1.2. Arduino Uno nedir ?	2
1.2.1 ATmega328 PIC	3
1.2.2 Haberleşme	6
1.2.3 USB çok fazla akım koruması	7
1.3. Potansiyometre nedir?	7
1.3.1 Doğrusal (Lineer) Potansiyometre	7
1.3.2 Dairesel (Rotational) Potansiyometre	7
1.3.3 Potansiyometrenin Avantajları	8
1.3.4 Potansiyometre Dezavantajları	9
1.3.5 Potansiyometre kullanım alanları	9
1.4 Led	9
1.5 Dot Matris	9
2. YAPILAN ÇALIŞMALAR	11
2.1 Devre şeması	11
2.2. Proje Programlanması	12
2.2.1 Koda Gececek olursak ;	12
3. SONUÇLAR	20
4. ÖNERİLER	20
5. KAYNAKLAR	21
STANDARTLAR ve KISITLAR FORMU	22

ÖZET

Gerçekleştirilen projede Arduino Uno'nun programlanmasıyla 8x8 LED MATRIX üzerinde PONG oyununun oynanması planlanmıştır. İki tane kullanıcı sağlanıp kullanıcılar potansiyometre kontrolü ile led matrix üzerinde dolaşan led'in kendi duvarına geçmemesini sağlamaya çalışmaktadır.

Arduino IDE içindeki kodu çalıştırarak öncelikle ekrana gelecek olan gülücük simgesinden sonra sağ duvarda 2 led sol duvarda da 2 led yanmış olacak. Ayrıca led matrix'in ortasında 1 tane daha led yanmış olacak ve led hareket edecektir sol veya sağ duvara doğru gidecektir. Potansiyometreler ile duvarlardaki ledleri aşağı yukarı hareket ettirerek ortada dolaşan led'in duvara çarpmaması kullanıcılar tarafından sağlanacaktır. Ortadaki led sağ veya sol duvara çarptığında oyun başa dönecektir.

KISALTMALAR VE SEMBOLLER

DC	Direct Current (Dođru Akım)
PIC	Peripheral Interface Controller
RAM	Random Access Memory
RISC	Reduced Instruction Set Computing
ADC	Analog to Digital Converter
UART	Universal Asynchronous Receiver/Transmitter
ICSP	In-Circuit Serial Programming
V	Gerilim
EEPROM	Electrically Erasable Programmable Read-Only Memory

1. GENEL BİLGİLER

1.1. Giriş

Arduino günümüzde bütün dünyada oldukça popüler durumdadır. İnternette yapacağımız bir aramada onlarca hatta yüzlerce Arduino projesi ile karşılaşabiliriz. Ayrıca geliştiricilerin katkısı ile Arduino kod tabanı oldukça gelişmiş durumdadır. Bir çok donanım için kütüphaneler bulunmaktadır. Arduino'nun birçok çeşidi bulunmaktadır. Hala bu kartlar geliştirilmeye devam edilmektedir.

Arduino üzerinde bulunan donanımlar ve pinler, Arduino'ya yüklenen kodlar tarafından kolaylıkla kontrol edilebilmektedir. Programcı tarafından yazılan bu kodların işlenmesi için Arduino üzerinde Atmel marka mikroşlemciler bulunmaktadır. Bu mikroşlemcilerin türüne göre de Arduino türleri belli olmaktadır. Arduino'nun bir türü için yazılmış bir kod, eğer o türe has özel donanımlar kullanmıyorsa diğer Arduino türleri üzerinde de sorunsuz çalışmaktadır. Bu yüzden çoğu Arduino projesi hemen hemen her Arduino türünde çalışmaktadır.

C bilgisi ile USB üzerinden mikrodenetleyicimizin programlamasını gerçekledebiliyoruz ve gerçek zamanlı uygulamaları çalıştırabiliyoruz. Birçok kütüphane mevcuttur. Editör gün geçtikçe daha kullanışlı ve fonksiyonel hale getirilmektedir.

Arduino IDE ile çok kolay bir şekilde programlar yazılıp derlenmesi sağlanmaktadır. Kullanıcıya birçok kolaylık sağlanıyor.

Arduino projeleri tek başına geliştirilebilir ya da bilgisayar üzerinde çalışan yazılımlara bağlanabilir. Arduino bir mikroşlemci değildir. Arduino mikroşlemciler için kolaylaştırıcı bir geliştirme ortamı sunar,

- Programlama için USB bağlantısı
- Girdi/Çıktı pinleri
- Entegre Led
- Güç girişi

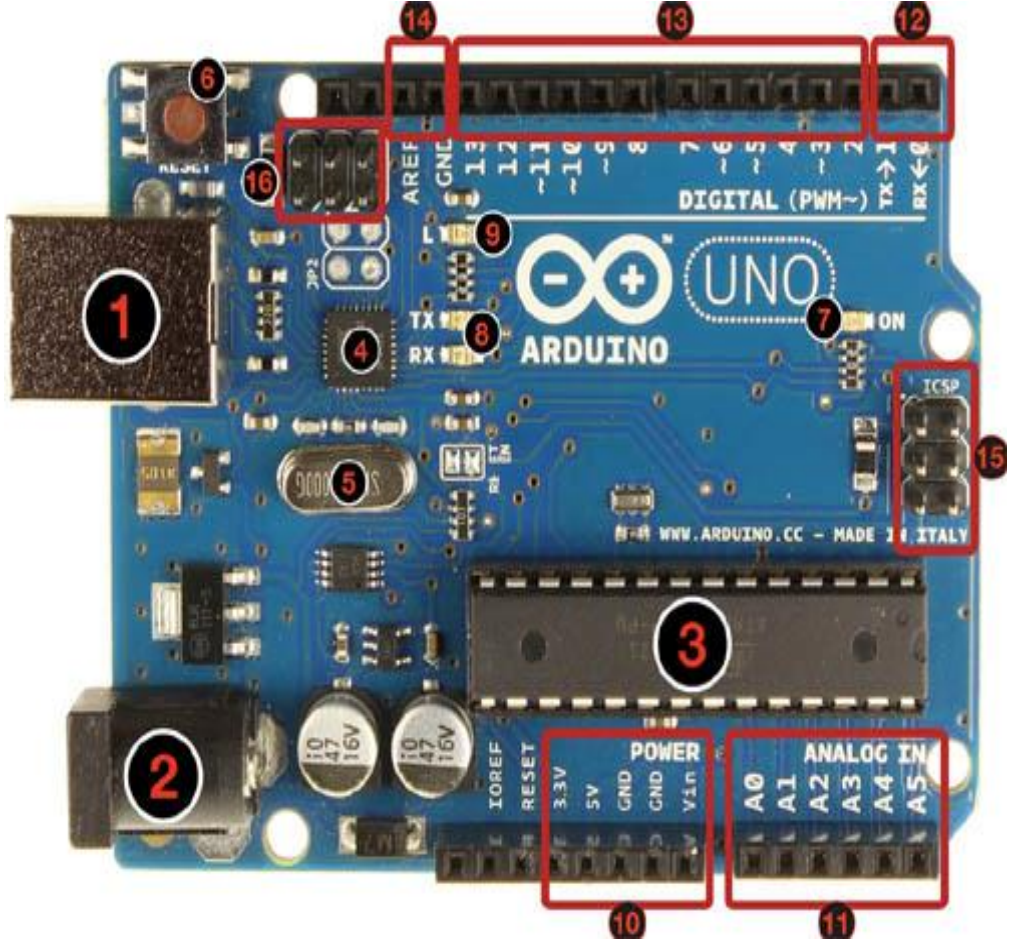
Fiziksel ortam için birçok mikrokontrolör ve mikrokontrolör platformu mevcuttur. Birçoğu aynı fonksiyona sahiptir. Fakat hepsinin programlanması zordur. Arduino ise programlamayı oldukça kolay hale getirir.

1.1. Projede kullanılan araçlar

- Arduino Uno R3
- Potansiyometre
- 8x8 Dot Matris
- Jumper Kablolar
- Board

1.2.Arduino Uno nedir ?

Arduino uno ATmega328 mikrodenetleyici kullanan bir Arduino kartıdır. Arduino 'nun en yaygın kullanılan kartı olduğu söylenebilir. Arduino Uno 'nun ilk modelinden sonra Arduino Uno R2, Arduino Uno SMD ve son olarak Arduino Uno R3 çıkmıştır.



Şekil.1.1 Arduino Uno R3 kısımları

Arduino Uno 'nun 14 tane dijital giriş / çıkış pini vardır. Bunlardan 6 tanesi PWM çıkışı olarak kullanılabilir. Ayrıca 6 adet analog girişi, bir adet 16 MHz kristal osilatörü, USB bağlantısı, power jakı (2.1mm), ICSP başlığı ve reset butonu bulunmaktadır. Arduino Uno bir mikrodenetleyiciyi desteklemek için gerekli bileşenlerin hepsini içerir. Arduino Uno 'yu bir bilgisayara bağlayarak, bir adaptör ile ya da pil ile çalıştırabilirsiniz. Yukarıdaki resimde Arduino Uno R3 'ün kısımları gösterilmektedir.

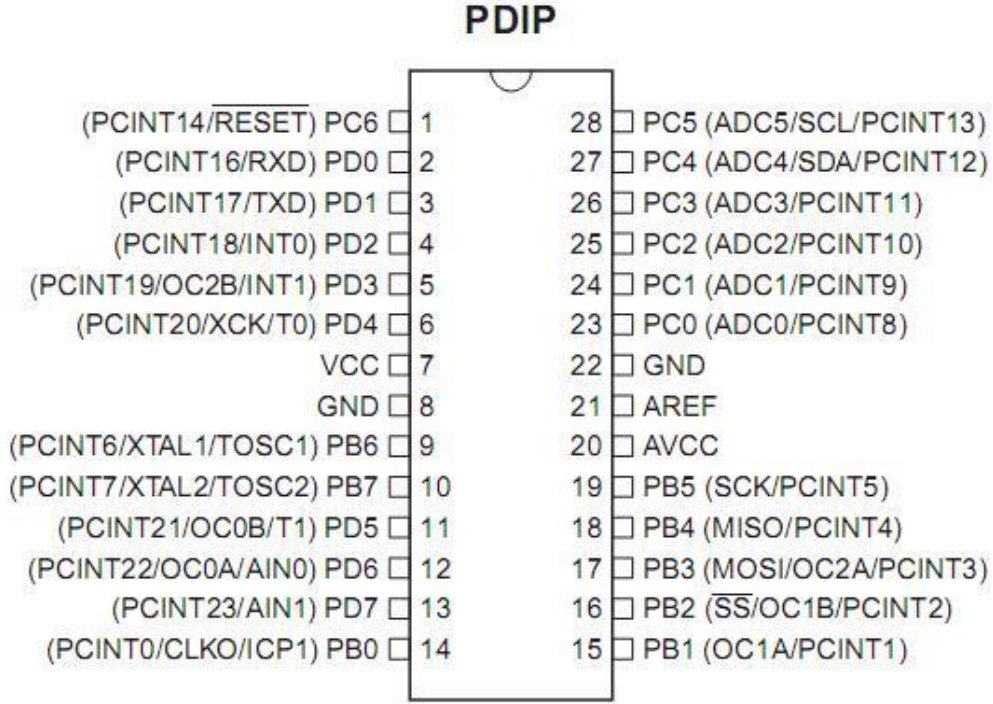
- 1 - USB jakı
- 2 - Power jakı (7-12 V DC)
- 3 - Mikrodenetleyici ATmega328
- 4 - Haberleşme çipi
- 5 - 16 MHz kristal
- 6 - Reset butonu
- 7 - Power ledi
- 8 - TX / NX ledleri
- 9 - Led
- 10 - Power pinleri
- 11 - Analog girişler
- 12 - TX / RX pinleri
- 13 - Dijital giriş / çıkış pinleri (yanında ~ işareti olan pinler PWM çıkışı olarak kullanılabilir.)
- 14 - Ground ve AREF pinleri
- 15 - ATmega328 için ICSP
- 16 - USB arayüzü için ICSP

Arduino uno teknik özelliklerini şöyle sıralayabiliriz.

- 1-Mikrodenetleyici : ATmega328
- 2-Çalışma gerilimi : +5 V DC
- 3-Tavsiye edilen besleme gerilimi : 7 - 12 V DC
- 4-Besleme gerilimi limitleri : 6 - 20 V
- 5-Dijital giriş / çıkış pinleri : 14 tane (6 tanesi PWM çıkışını destekler)
- 6-Analog giriş pinleri : 6 tane
- 7-Giriş / çıkış pini başına düşen DC akım : 40 mA
- 8-3,3 V pini için akım : 50 mA
- 9-Flash hafıza : 32 KB
- 10-SRAM : 2 KB
- 11-EEPROM : 1 KB
- 12- Saat frekansı : 16 MHz

1.2.1 ATmega328 PIC

Arduino kartları üzerinde Atmel firmasının çeşitli mikrodenetleyicileri bulunuyor. Arduino Uno, Mini gibi modellerde Atmega328p'yi görüyoruz. Yüksek performanslı Atmel 8-bit AVR RISC tabanlı mikroişlemci okuma-yazma yeteneklerine sahiptir. 32KB ISS flash bellek, 1KB EEPROM, 2KB SRAM, 23 genel amaçlı I / O hatlarını, 32 genel amaçlı çalışma kayıtlarını, üç esnek zamanlayıcı / sayaçları modlarını, iç ve dış kesmeleri, seri programlanabilir USART, bir bayt odaklı 2-tel seri arabirimi, SPI seri portu, 6-kanal 10-bit A / D dönüştürücüsü (TQFP ve QFN / MLF paketlerinde 8-kanal) dahili osilatör ile programlanabilir watchdog zamanlayıcı ve beş tane seçilebilir güç tasarruf modlarını içermektedir. Cihaz 1,8-5,5 volt arasında çalışır.



Şekil.1.2 ATMega 328 PIC

VCC – Dijital Besleme Gerilimi

GND – Toprak

Port C (PC5:0) – Port C iç pull-up dirençleri (her bit için seçilen) ile 7-bit çift yönlü I / O portudur. PC5 .. 0 çıkış tamponları kaynak yetenekleri olan simetrik sürücü özelliklerine sahiptir. Pull-up dirençleri aktif olduğunda, Port C pinleri low'a çekilir. Saat çalışmasa bile reset durumu aktif olduğunda Port C pinleri 3 durumlu olmaktadır.

PC6 / RESET – Eğer RSTDISBL programlanmış ise, PC6 I / O pini olarak kullanılabilir. PC6'nın diğer Port C pinlerine göre elektriksel özellikleri farklıdır. Eğer RSTDISBL programlanmamış ise, PC6 RESET girişi olarak kullanılır. Saat çalışmasa dahi minimum darbe uzunluğundan daha uzun süre low da kalacak olan pin RESET üretecektir. Kısa darbelerin reset üreteceği garanti edilemez.

Port D (PD7:0) – Port D C iç pull-up dirençleri (her bit için seçilen) çift yönlü bir I / O portudur. Port D çıkış tamponları kaynak yetenekleri olan simetrik sürücü özelliklerine

sahiptir. Pull-up dirençleri aktif olduğunda, Port C pinleri low'a çekilir. Saat çalışmasa bile reset durumu aktif olduğunda Port C pinleri 3 durumlu olmaktadır.

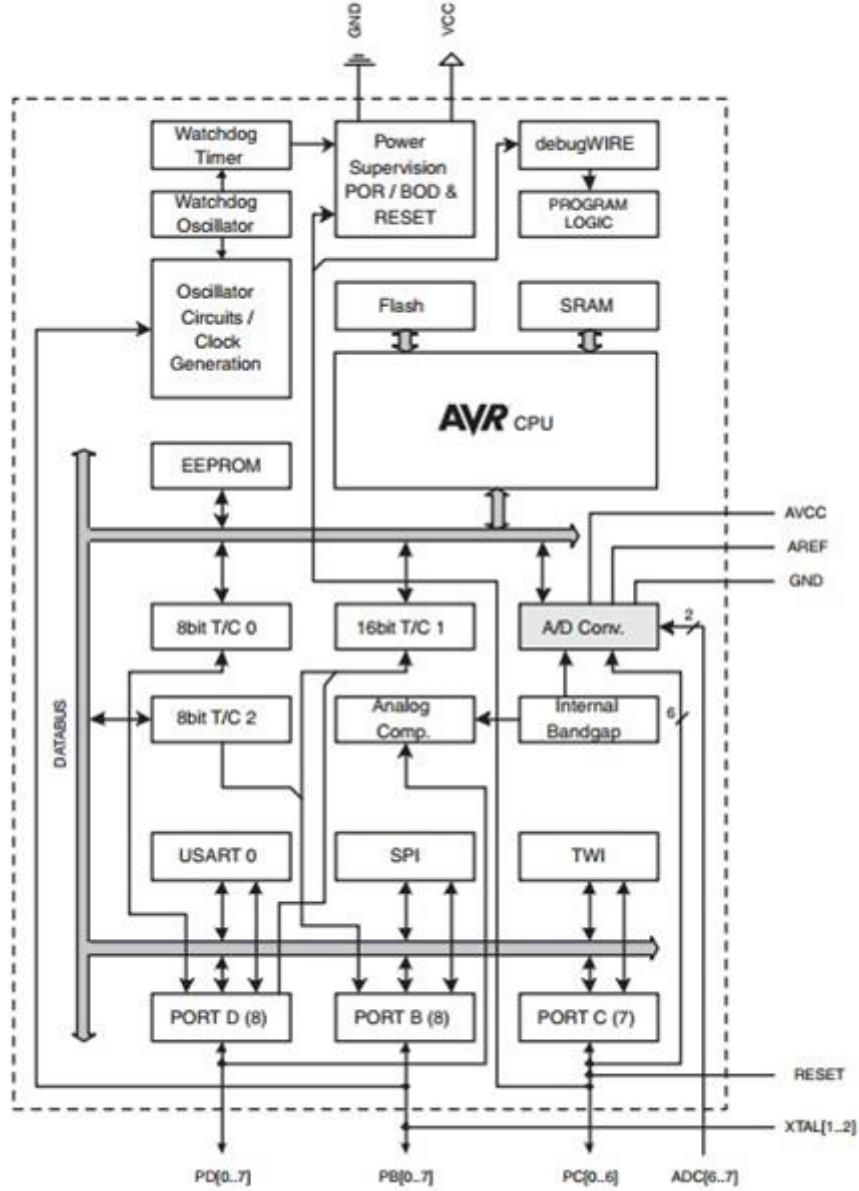
AVcc - A/D Dönüştürücü, PC3:0, ve ADC7:6 için besleme gerilimidir. ADC kullanılmasa bile VCC dışarıdan bağlanmalıdır. ADC kullanıldığında düşük bir geçiş filtresi içerisinde VCC bağlı olmalıdır. PC6...4 VCC besleme gerilimini kullanır.

AREF – AREF, A / D Dönüştürücü için analog referans pinidir.

ADC7:6 (TQFP ve QFN/MLF) - TQFP ve QFN/MLF paketinde, ADC7:6 , A / D Dönüştürücünün analog girişlerine hizmet sunar. Bu pinler analog kaynaktan güç alır ve 10 bit olan ADC kanallarına hizmet sunar.

ATmega328 PIC Özellikleri :

Parametreler	Değerler
Flash:	32 KBytes
Pin Sayısı:	32
Max frekans:	20
MHz CPU:	8-bit
AVR Dokunmatik kanal:	16
EEPROM:	1 KBytes
RAM:	2 KBytes
Max I/O Pin Sayısı:	23
Dış Kesme:	24
USB Hızı:	No
USB Arayüzü:	No



Şekil.1.3 ATmega328 Blok Diyagram

1.2.2 Haberleşme

Arduino Uno bir bilgisayarla, başka bir Arduino'yla ya da diğer mikrodenetleyiciler ile haberleşme için imkanlar sunar. ATmega328 mikrodenetleyici, RX ve TX pinlerinden erişilebilen UART TTL (5V) seri haberleşmeyi destekler. Kart üzerindeki bir ATmega16U2 seri haberleşmeyi USB üzerinden kanalize eder ve bilgisayardaki yazılıma sanal bir com portu olarak görünür. 16U2 standart USB com sürücülerini kullanır ve harici sürücü gerektirmez. Ancak, Windows 'ta bir .inf dosyası gereklidir. Kart üzerindeki RX ve TX ledleri USB den seri çipe ve USB den bilgisayara veri giderken yanıp söner.

SoftwareSerial kütüphanesi Arduino Uno 'nun digital pinlerinden herhangi biri üzerinden seri haberleşmeye imkan sağlar.

1.2.3 USB çok fazla akım koruması

Arduino Uno, bilgisayarımızın USB portunu fazla akım ve kısa devreden koruyan resetlenebilir bir çoklu sigortası vardır. Çoğu bilgisayarın portlar için kendi korumaları olmasına rağmen bu sigorta ekstra bir koruma katmanı sağlar. Eğer USB portuna 450 veya 500 mA den fazla bir yük binerse, sigorta otomatik olarak bağlantıyı kısa devre veya fazla akım durumu ortadan kalkana dek keser.

2.Potansiyometre nedir?

Potansiyometre bir direnç (reosta) türüdür. Fakat diğer direnç türlerinden ayrılan en büyük özelliği ise direnç değeri değiştirilebilir olmasıdır.

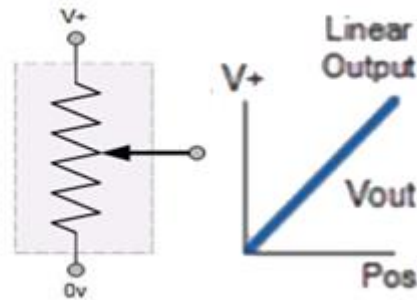
Potansiyometre genel olarak 3 bacaklı olmaktadır. Bu bacakların ikisi iç yapısında sabit fakat üçüncü bacak ise iç yapısında hareketli bir yapıya sabittir. İşte bu hareketli yapı sayesinde sabit diğer iki baktan sürekli değişen bir voltaj çıkışı alabilmek mümkün hale gelmektedir.

Başlıca potansiyometre çeşitleri şunlardır;

- Doğrusal (Lineer) potansiyometre
- Dairesel (Rotational) potansiyometre

1.3.1 Doğrusal (Lineer) Potansiyometre

Doğrusal bir düzlem üzerine montaj edilmiş direnç türüdür. Direnç değeri 0'dan itibaren doğrusal bir şekilde artar. Bu artım sayesinde elektronik bir devrede doğrusal bir şekilde voltaj kontrolü sağlanabilir.

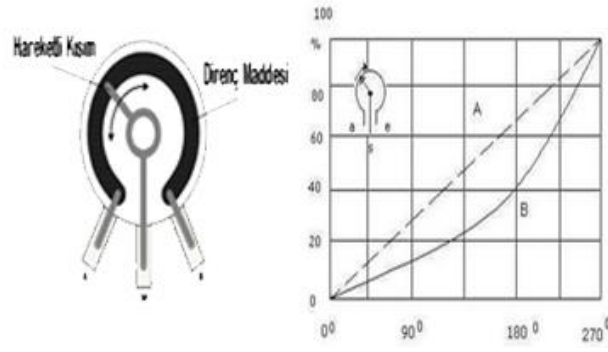


Şekil 1.2.1 Doğrusal Sistemli Bir Potansiyometrenin İç Yapısı

Soldaki resim doğrusal sistemli bir potansiyometrenin iç yapısını göstermektedir. Sağdaki resim de ise bu potansiyometrenin hareketli milinin pozisyonuna göre verdiği çıkışı sembolize eden bir grafik yer almaktadır.

1.3.2 Dairesel(Rotational) Potansiyometre

Dairesel bir düzlem üzerine montaj edilmiş bir direnç türüdür. Doğrusal Potansiyometre türünden ayrılan en önemli özelliği ise Direnç değeri logaritmik bir düzen de artma göstermektedir.



Şekil.1.2.2 Dairesel Potansiyometre

Soldaki resim dairesel sistemli bir potansiyometrenin iç yapısını göstermektedir. Sağdaki resim de ise bu potansiyometrenin hareketli milinin hareketine göre verdiği çıkışı sembol eden bir grafik yer almaktadır.



Şekil.2.3 Potansiyometre

1.3.3 Potansiyometrenin Avantajları

Bu kısımda maddeler halinde potansiyometreyi ona benzer özellik taşıyan diğer elektronik parçalardan ayıran önemli özellikleri hakkında bilgi vermeye çalışacağım.

Düşük maliyet: Diğer voltaj ayarlama için kullanılacak elemanlara kıyasla daha ucuza bulunabilmektedir.

Düşük Teknoloji: Kullanılan teknolojinin basitliğinden dolayı kolayca nasıl kullanılacağı anlaşılabilir.

Kolay kullanım: Kullanımı kolay ve çabuk öğrenilebilir.

1.3.4 Potansiyometre Dezavantajları

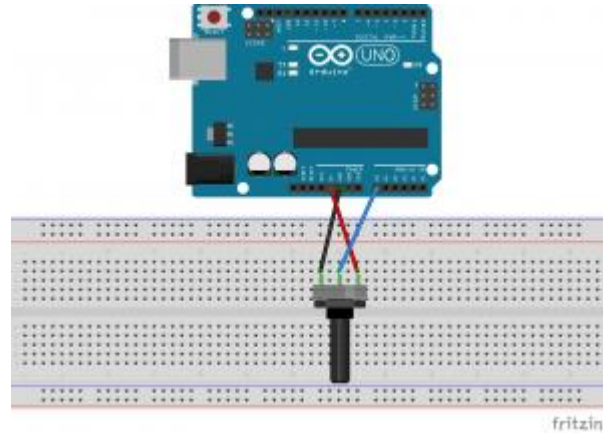
Potansiyometreyi olumsuz etkileyen sebepler ise kısaca şunlardır;

İç Parçalarda Aşınma: İç parçalarda zamanla aşınmalar meydana gelebilmektedir.
Düşük Hassasiyetlik: Yüksek hassasiyetliği yakalamak zordur.

1.3.5 Potansiyometrenin Kullanım Alanları

Voltaj Kontrolü: Potansiyometre genel anlamda voltaj kontrolü amacı ile kullanılmaktadır. Çoğunlukla ise gerilim bölücü olarak kullanılmaktadır. Bu alana örnek olarak dimmer devreleri ve basit motor kontrol devreleri örnek gösterilebilir.

Pozisyon Algılama: Pozisyon algılama sensörü olarak ta kullanılmaktadırlar. Fakat bu durumda hassas bir ölçüm için ayrıca bir elektronik devre ile kontrol edilmesi gerekir.



Şekil.1.3. Arduino ile potansiyometre bağlantısı

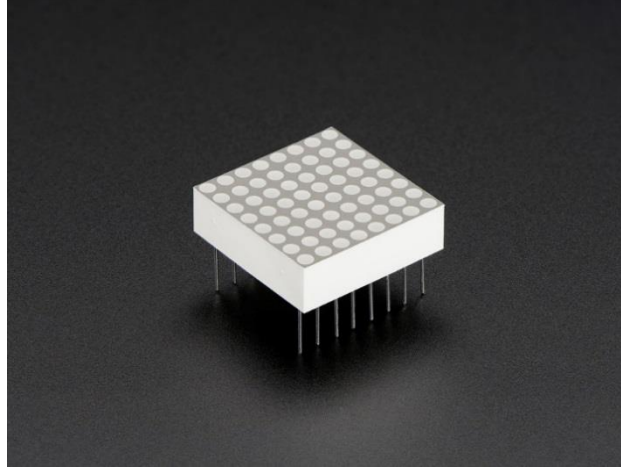
3.Led

LED ("Light Emitting Diode", Işık Yayan Diyot), yarı-iletken, diyot temelli, ışık yayan bir elektronik devre elemanıdır.

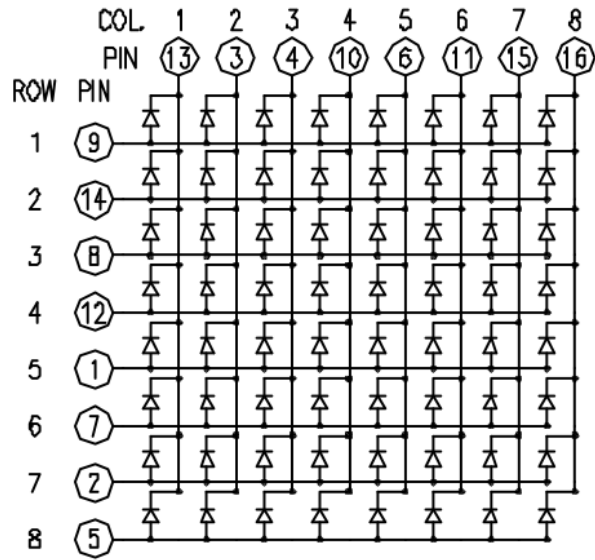
3.1 Dot Matris

LED ekranlar sıklıkla matrisler olarak paketlenir LED'ler ortak anot ve katot ortak sütunlarında ya da ters sıralar halinde düzenlenmiştir.

Şemadan da anlaşılacağı üzere sütun bacakları ANOT (+), satır bacakları ise KATOT(-) .Diyelim ki 3. satırdaki 4. LED i yakıp diğerlerini söndürmek istiyoruz, yapmamız gereken 4. sütundan lojik 1 verip diğer sütun bacaklarına lojik sıfır uygulamak ve 3. satır bacağına lojik 0 verip diğer satır bacaklarına lojik 1 vermek. Eğer herhangi iki satır bacağına lojik sıfır verecek olursak, sıfır verilen her bir satırın 4. LED ini aktif etmiş oluruz.

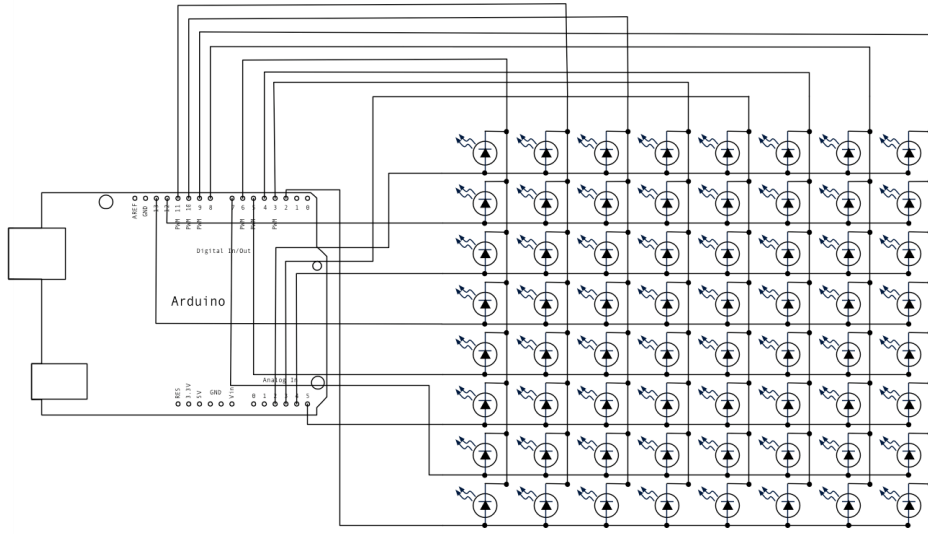


Şekil.3.1 8x8 Dot Matris



Şekil.3.2 8x8 Dot Matrix Şeması

8x8 dot matrisler 64 adet LED den oluşan, sistemler üzerinde bir çıkış birimi olarak kullanılan bileşenlerdir. Dışarı çıkartılan bacaklar yardımıyla her bir satırdaki her bir LED in durumunu kontrol edebilirsiniz.

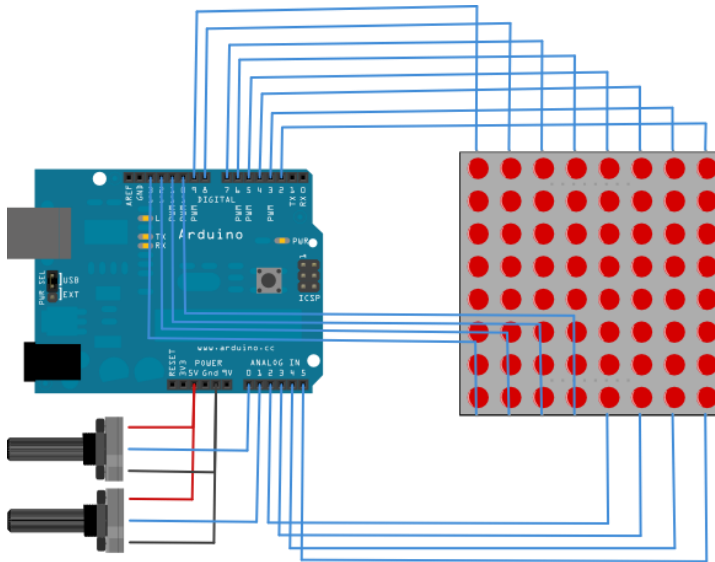


Şekil.3.3 Dot Matris'in Arduino ile bağlantısı

YAPILAN ÇALIŞMALAR

2.1 Devre şeması

Devre şemasını kurarken ilk olarak potansiyometre'nin (5K potansiyometre) bir bacağına GND ye bir bacağına da 5 V verdim. Orta bacaklarını da arduino'nun A4 ve A5 analog girişlerine bağladım. Bu bacaklar benim için sağ ve sol duvarlardaki yanar 2 led'i kontrol etmemi sağlamak için bağlandı. Daha sonra 2 tane board üzerine Dot matrix i yerleştirdim. 8 bacağına bir board'a verdim. 8 bacağına da diğer board'a verdim. Dot matrix'in birinci bacağından itibaren verilen dizi içeriğine göre arduino'nun analog ve digital pinlerine bağlantısını gerçekleştirilir.



Şekil.2.1 Devre Şeması

2.2 Proje Programlanması

Yazdığımız Arduino programının içeriğinden bahsetmeden önce. Arduino'ya ait kullandığımız komutlardan bahsetmem faydalı olacaktır. Arduino'da *void setup()* ve *void loop()* fonksiyonları vardır. *void setup()* fonksiyonu, program başladığında ilk çalışan ve gerekli setlemeleri yapan fonksiyonlardır.

2.2.1 Koda Gececek olursak ;

```
#include "TimerOne.h" // timer kütüphanesini koduma ekledim.
```

```
#define PIN_LEFT 4 // sol potansiyometre analog A4 girişinden alınır.
```

```
#define PIN_RIGHT 5 // sağ Potansiyometre analog A5 girişinden alınır.
```

```
unsigned int left = 0; // potansiyometre LEFT değeri default olarak ayarlanır .
```

```
unsigned int right = 0; // potansiyometre RIGHT değeri default olarak ayarlanır.
```

Burada ise Dot Matris in satır ve sütun değerleri atanır . Dot Matris pin girişlerinin hangilerinin hangi uçtan Arduino'ya bağlanacağını gösterilir.

```
byte rows[8] = {9, 14, 8, 12, 1, 7, 2, 5};
```

```
byte cols[8] = {13, 3, 4, 10, 6, 11, 15, 16};
```

```
byte pins[16] = {5, 4, 3, 2, 14, 15, 16, 17, 13, 12, 11, 10, 9, 8, 7, 6};
```

```
byte screen[8] = {0, 0, 0, 0, 0, 0, 0, 0}; // Dot Matris 0'a setlenir .
```

```
volatile byte screenRow = 0; // satır tüm değerleri 0'a setlenir.
```

```
volatile byte screenCol = 0; // sütun tüm değerleri 0'a setlenir.
```

Setup fonksiyonu başlangıçta sadece bir defa çalışır. Timer1 arduino kesmeleri için kullanılır. Kontrol içindir. PinMode fonksiyonu arduinonun fonksiyonun pinlerine çıkış veriliyor yani o pinlerden. 2 tane buffer dan oluşan kesme havuzu gösterilir. Tüm kesmeler havuza atılıyor. Giriş çıkış yapılır. Test işlemleri burada görülür. Ekran yazı yazdırılabilir gerektiğinde çalışacaktır.

```
void setup()
{
  Timer1.initialize(100);
  for (int i = 2; i <= 17; i++)
    pinMode(i, OUTPUT);

  Timer1.attachInterrupt(doubleBuffer);
  Serial.begin(9600);

  face();
  // yüz fonksiyonuna
  reset();
  // sıfırlama fonksiyonu
}
```

2 bufferlı kesme havuzu fonksiyonu tanımlanır. Digital Write ile pin'i yetkilendiririz. Dizi içinde verilen pinler 0 volt'u gösterecektir. Buradaki dizide yani 2.satırda verilen pinler 5 Volt a setlenir. Durmadan artış yapıyoruz. ScreenCol 8 i aştığında screencol 0'a setlenecektir screenRow ++ ; ile.

```
void doubleBuffer()
{
  digitalWrite(translatePin(rows[screenRow]), LOW);

  digitalWrite(translatePin(cols[screenCol]), HIGH);

  screenCol++;
  if (screenCol >= 8) {
    screenCol = 0; // 0' a setlenir
    screenRow++;
  }
  if (screenRow >= 8) {
    screenRow = 0; // 0'a setlenir
  }
}
```

Burada >> bitleri sağa kaydırıp hex B1 (177) . B1 binary sayısındaki bir olan bitler screen[screenRow] da 1 ise if'in içindeki yetkilendirmeyi yapacak. değilse else çalışacaktır.

```
if((screen[screenRow] >> screenCol) & B1 == B1)
{
    digitalWrite(translatePin(rows[screenRow]), HIGH);
    digitalWrite(translatePin(cols[screenCol]), LOW);
}
else {
    digitalWrite(translatePin(rows[screenRow]), LOW);
    digitalWrite(translatePin(cols[screenCol]), HIGH);
}
}
```

Bu translatePin fonksiyonu byte tipinde bir değer döndürmüş byte alıyor gelen bayttan 1 çıkarıyor pins dizine bak.

```
byte translatePin(byte original)
{
    return pins[original - 1];
}
```

Projemde tüm ledlerin kapanıp açılması gibi bir uygulama gerçekleştirdim. Oyun bittiğinde falan tüm ledlerin bir defa olması koşulu ile tüm ledleri kapattım alloFF fonksiyonu ile gerçekleştirdim.

```
void alloFF() {
    for (int i = 0; i < 8; i++)
        screen[i] = 0;
}
```

Gelen satır ve sütundaki led setleniyor. Aynı şekilde off fonksiyonu ile gelen satır sütunu kapatıyor. Yukarıdaki fonksiyonda açmıştık led'i.

```
void on(byte row, byte column)
{
    screen[column-1] |= (B1 << (row - 1));
}
```

```
void off(byte row, byte column)
{
    screen[column-1] &= ~(B1 << (row - 1));
}
```

Burada duvarı hesaplıyoruz. Çarpınca yansımaları sağlıyoruz. Sol duvar potansiyometre yeri hesaplanması yapılıyor. Sağ duvar potansiyometre yeri hesaplanması yapılıyor. Potansiyometrelerin kullanacağı led sayıları belirleniyor . 2 Şer tane setlendi.

```
void calcWall()
{
  left = analogRead(PIN_LEFT); // sol potansiyometre
  right = analogRead(PIN_RIGHT); // sağ potansiyometre
  left = constrain(map(left, 223, 800, 0, 6), 0, 6);
  right = constrain(map(right, 223, 800, 6, 0), 0, 6);

  clearWall();

  on(1, left + 1);
  on(1, left + 2);
  on(8, right + 1);
  on(8, right + 2);

  _wall[0] = left;
  _wall[1] = right;
  show();
}
```

Kenarları temizlemek clearWall fonksiyonunu kullandım.

```
void clearWall()
{
  for (int i = 0; i < 8; i++)
    screen[i] &= B01111110;
}
```

Oyunu temizliyoruz clearGame fonksiyonu ile .

```
void clearGame()
{
  for (int i = 0; i < 8; i++)
    screen[i] &= B10000001;
}
```

100 delay gecikmesiyle önce duvarı hesaplıyoruz . sonra çarpışmaları hesaplayan fonksiyon çağırılıyor.

```
void loop()
{
  calcWall();
  enterFrameHandler();
  delay(100);
}
```

Bu fonksiyon count speedten büyükse fonksiyon çalışsın. Hız alt limiti koyduk. Count her defasında 0'lanır. Çarpışma olup olmadığına bakılıyor . sonra açısı hesaplanır devam ediliyor.

```
void enterFrameHandler()
{
  if (_count++ < _speed)
    return;

  _count = 0; // sıfırlanır her defasında.
  checkCollision();
  calcAngleIncrement();
  show();
}
```

Bu fonksiyonda Led in geriye dönüşünü açıyı alarak hesaplıyor. Retorted fonksiyonu ile çarpışmaları alıp geri döndüreceğiz.

```
void retorted(int angle)
{
  Serial.println(angle);
  _angle = angle;

  if (++_countPoints % 5 == 0 && _speed > 1)
    _speed--;
}
```

Çarpışmaları resetlemek için resetAnim fonksiyonunu forlarla gerçekleştirdim.

void resetAnim() çarpışmaları resetliyoruz.

```
{
  for (int i = 0; i < 8; i++){
    screen[i] = B11111111;
    delay(25);
  }
  for (int i = 0; i < 8; i++){
    screen[i] = B00000000;
    delay(25);
  }
}
```

void face() // Gülen ifademizi Dot matris üzerinde ayarladık .

```
{
  on(1, 1);
  on(1, 2);
  on(2, 1);
  on(2, 2);
  on(7, 1);
  on(7, 2);
  on(8, 1);
}
```

```

on(8, 2);
on(1, 1);
on(1, 2);
on(4, 4);
on(4, 5);
on(5, 4);
on(5, 5);
on(2, 7);
on(7, 7);
on(3, 8);
on(4, 8);
on(5, 8);
on(6, 8);
delay(5000);
}

```

Resetlendiğinde random olarak hız ve açı koordinatları belirlendi. Oyunumuzda kullanıcılar durmadan maçı kaybedebileceklerdir. Bundan dolayı her oyun bittiğinde başka kullanıcıya avantaj sağlamamak için tekrar başlarken oyun random olarak başlaması en doğru karar olacaktır.

```

void reset()
{
  resetAnim();
  _px = random(3, 5);
  _py = random(3, 5);
  _angle = random(0, 2) == 0 ? 0 : 180;
  _speed = 5;
  _countPoints = 0;
  show();
  delay(500);
}

```

Belirlenen pixelde yanan led yanıyor.

```

void show()
{
  clearGame();
  on(_px + 1, _py + 1);
}

```


Çarpıştıktan sonra geriye dönen değeri hesaplanan kod aşağıda verilmiştir. Her bir potansiyometre ledlerine çarptığı andaki çarpımları hesaplıyoruz.

```
void checkCollision()
{
  if (_px == _w - 1)
  {
    if (_angle == 315 || _angle == 0 || _angle == 45)
    {
      if (_py == _wall[1] || _py == _wall[1] + 1)
      {
        if (_angle == 0 && _py == _wall[1])
          retorted(225);
        else if (_angle == 0 && _py == _wall[1] + 1)
          retorted(135);
        else if (_angle == 45 && _py == _wall[1])
          retorted(135);
        else if (_angle == 45 && _py == _wall[1] + 1)
          retorted(180);
        else if (_angle == 315 && _py == _wall[1])
          retorted(180);
        else if (_angle == 315 && _py == _wall[1] + 1)
          retorted(225);
      }
    }
  }
  else if (_px == 1)
  {
    if (_angle == 225 || _angle == 180 || _angle == 135)
    {
      if (_py == _wall[0] || _py == _wall[0] + 1)
      {
        if (_angle == 180 && _py == _wall[0])
          retorted(315);
        else if (_angle == 180 && _py == _wall[0] + 1)
          retorted(45);
        else if (_angle == 135 && _py == _wall[0])
          retorted(45);
        else if (_angle == 135 && _py == _wall[0] + 1)
          retorted(0);
        else if (_angle == 225 && _py == _wall[0])
          retorted(0);
        else if (_angle == 225 && _py == _wall[0] + 1)
          retorted(315);
      }
    }
  }
  if (_px == _w)
  {
    reset();
  }
}
```

```

}
else if (_px == 0)
{
    reset();
}
else if (_py == _h)
{
    if (_angle == 45)
        _angle = 315;
    else if (_angle == 135)
        _angle = 225;
}
else if (_py == 0)
{
    if (_angle == 225)
        _angle = 135;
    else if (_angle == 315)
        _angle = 45;
}
}

```

Çarpışma ile birlikte açının hesaplanıp geriye led'in hangi açı ile yansımaları gerektiğini hesaplıyoruz. Oyunun devamlılığı için bu gereklidir.

```

void calcAngleIncrement()
{
    if (_angle == 0 || _angle == 360)
    {
        _px += 1;
    }
    else if (_angle == 45)
    {
        _px += 1;
        _py += 1;
    }
    else if (_angle == 135)
    {
        _px -= 1;
        _py += 1;
    }
    else if (_angle == 180)
    {
        _px -= 1;
    }
    else if (_angle == 225)
    {
        _px -= 1;
        _py -= 1;
    }
    else if (_angle == 315){
        _px += 1;
        _py -= 1; }}

```

2. SONUÇLAR

Sonuç olarak projemi başarıyla gerçekleştirdim. Şu an için 2 tane ayrı kullanıcının birbirleriyle karşılıklı olarak oynayabilecekleri güzel bir PONG oyununu gerçekleştirmiş bulunuyorum. Bu projeyi gerçekleştirirken Potansiyometre , Dot matris Ve arduino hakkında çok fazla bilgi ve birikime sahip oldum. Bu benim için çok büyük bir artı olmuş oldu . Çünkü günümüzde arduino oldukça hızlı bir gelişime sahip bir teknoloji ilerde tekrar karşıma çıktığında kolaylıkla bir şeyler yapabilme fırsatını ve kolaylığını bulacağıma inanıyorum.

3. ÖNERİLER

Bu proje şöyle geliştirilebilir. Örneğin her kazanan tarafın kazanma durumunu 1 artırılıp gösterilebilir. Bunun için de ekstra bir decoder entegre devresine ihtiyacımız olacaktır. Skorboard eklemeyi gerçekleştirebiliriz.

KAYNAKLAR

<https://www.youtube.com/watch?v=Mf6buDkbyAQ> 19 mart 2009

<https://www.youtube.com/watch?v=qOttojWD1oQ> 19 ekim 2012

<http://arduinoturkiye.com/arduino-nedir-ve-ne-degildir/> 4 şubat 2013

https://en.wikipedia.org/wiki/Dot_matrix 15 haziran 2015

<http://www.elektrikrehberiniz.com/elektronik/potansiyometre-nedir-10639/>
18 şubat 2015

STANDARTLAR ve KISITLAR FORMU

Projenin hazırlanmasında uyulan standart ve kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. Projenizin tasarım boyutu nedir? (Yeni bir proje midir? Var olan bir projenin tekrarı mıdır? Bir projenin parçası mıdır? Sizin tasarımınız proje toplamının yüzde olarak ne kadarını oluşturmaktadır?)

Çarpışma sonucu oluşan açılarda daha önce kullanılmış tanımlı teknik kullandığım için tasarım projenin %70'ini kapsamaktadır.

2. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü? Açıklayınız.

Çarpışmalardan sonra yansımaları hesapladım.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Evet kullandım. C bilgisi ve elektronik laboratuvar bilgileri projeyi yapmamda yardımcı oldu.

4. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir? (Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız).

Elektronik devre standartları ve programlamada açıl standartları kullandım.

5. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen boşlukları uygun yanıtlarla doldurunuz.

- a) Ekonomi

Herhangi bir ekonomik kısıtlama ile karşılaşmadım.

- b) Çevre sorunları:

Herhangi bir çevre sorunu ile karşılaşmadım.

c) Sürdürülebilirlik:

Daha fazla geliştirilebilir ve sürdürülebilirliği mevcuttur.

d) Üretilebilirlik:

Üretilebilirlik açısından sıkıntı oluşmamaktadır. Çok fazla artırılabilir.

e) Etik:

Proje etik açıdan herhangi bir sorun içermemektedir.

f) Sağlık:

Proje sağlık açısından herhangi bir sorun içermemektedir.

g) Güvenlik:

Proje güvenlik açısından herhangi bir sorun içermemektedir.

h) Sosyal ve politik sorunlar:

Proje sosyal ve politik açıdan herhangi bir sorun içermemektedir.