

**BIL 395 PROJE I**

**Ball on Plate**

**Final Demosu**

**Danışman: Prof. Dr. Erkan ZERGEROĞLU**

**Asistan: Mahmud Rasih ÇELENLİOĞLU**

**Grup 3**

**Deniz BABAT**

**Ersin CEYLAN**

**Melek Nurten YAVUZ**

**Muhammed Tasin Neciyullah YAĞAN**

**Murat Salih TUNALI**

**Safa Emre DULUNDU**

**Yasin AÇIKGÖZ**

**Yasin TULUMEN**

**Aralık-2016**

**Gebze – KOCAELİ**

İçindekiler

[**1. Giriş** 4](#_Toc470244689)

[**1.1 Doküman İçeriği** 4](#_Toc470244690)

[**1.2 Hedef Kitle** 4](#_Toc470244691)

[**2.Donanım Modülü** 4](#_Toc470244692)

[**2.1 Genel Tanım** 4](#_Toc470244693)

[**2.2 Kullanılacak Donanım Birimleri** 4](#_Toc470244694)

[**2.3 Platformun Kurulması** 5](#_Toc470244695)

[**2.3.1 Tablanın Monte Edilmesi** 5](#_Toc470244696)

[**2.3.2 Motorların Monte Edilmesi:** 5](#_Toc470244697)

[**2.3.3 Motorlar ile Tabla Arasındaki Bağlantının Sağlanması** 6](#_Toc470244698)

[**2.3.4 Dokunmatik Panelin Tabla Üzerine Sabitlenmesi** 7](#_Toc470244699)

[**2.4 Donanım Elemanları Arasındaki Elektronik Bağlantıların Sağlanması** 8](#_Toc470244700)

[**2.4.1 Dokunmatik Ekranın Arduino’ya Bağlanması** 8](#_Toc470244701)

[**2.4.2 5mm 3v Kırmızı Mantar LED’lerin Sisteme Dahil Edilmesi** 9](#_Toc470244702)

[**2.4.3 Kızılötesi Alıcı Sensörün Sisteme Dahil Edilmesi** 9](#_Toc470244703)

[**3. Yazılım Modülü** 10](#_Toc470244704)

[**3.1 Genel Tanım** 10](#_Toc470244705)

[**3.2. Modül Özellikleri** 10](#_Toc470244706)

[**3.2.1 Modül Temelleri** 10](#_Toc470244707)

[**I. PID** 10](#_Toc470244708)

[**II. PID Tuning** 10](#_Toc470244709)

[**III. Mapping** 11](#_Toc470244710)

[**3.3 MODLAR** 11](#_Toc470244711)

[**3.3.1 Dikdörtgen Çizme Modu** 11](#_Toc470244712)

[**3.3.2 Daire Çizme Modu** 11](#_Toc470244713)

[**3.3.3 Şekil Çizme Modu** 12](#_Toc470244714)

[**3.3.4 Oyun Modu** 12](#_Toc470244715)

[**3.4 AKIŞ DİYAGRAMI** 12](#_Toc470244716)

[**I. BalancePlate** 13](#_Toc470244717)

[**II. ModeInput** 13](#_Toc470244718)

[**III-I Circle** 13](#_Toc470244719)

[**III-II Rectangle** 13](#_Toc470244720)

[**III-III Draw Shape** 13](#_Toc470244721)

[**III-IV Game** 13](#_Toc470244722)

[**IV. CoordinateMapping** 13](#_Toc470244723)

[**V.PID** 13](#_Toc470244724)

[**VI.AngleMapping** 13](#_Toc470244725)

[**VII.Output** 13](#_Toc470244726)

[**VIII.ServeEngines** 14](#_Toc470244727)

[**IX.RemoteInput** 14](#_Toc470244728)

[**X.LedOptimization** 14](#_Toc470244729)

[**4. 3D Modülü** 14](#_Toc470244730)

[**4.1 Genel Tanım** 14](#_Toc470244731)

[**4.2 Modül Özellikleri** 14](#_Toc470244732)

[**4.2.1 PID Hata Grafiği** 14](#_Toc470244733)

[**4.2.2 Arayüz Tasarımı** 14](#_Toc470244734)

[**4.2.3 3D Tasarımı** 15](#_Toc470244735)

[**5. Donanımlar Arası Haberleşme** 17](#_Toc470244736)

[**5.1 Yazılım Modülünden 3D Modülüne Veri Gönderilmesi** 17](#_Toc470244737)

[**5.2 3D Modülünden Yazılım Modülüne Veri Gönderilmesi** 18](#_Toc470244738)

# **1. Giriş**

## **1.1 Doküman İçeriği**

Bu doküman BİL 395 Proje I dersi kapsamında 3. Grup tarafından yapılacak olan Ball on Plate projesi hakkında bilgi vermek amacıyla yazılmıştır. Proje 3 adet modülden oluşmaktadır. Bunlar; Donanım, Yazılım ve 3D modülleridir. Doküman projenin nasıl yapılacağına, kullanılacak yöntemlere ve gerekli araçlara dair bilgiler içermektedir**.** Ayrıca proje modüllerinin projenin gerçeklenmesi esnasında ne şekilde kullanılacağı konusunda bilgiler verilecektir.

## **1.2 Hedef Kitle**

* Prof. Dr. Erkan ZERGEROĞLU
* Arş. Gör. Mahmud Rasih ÇELENLİOĞLU
* Grup 3 üyeleri

**1.3 Proje Tanımı**

Proje I dersi kapsamında, Arduino mikro denetleyicisine bağlanan, dokunmatik panelden oluşan düz bir yüzeye konulan topun düşürülmeden belirlenen denge noktasına getirilmesini sağlamak ayrıca söz konusu dokunmatik paneli ve topu 3 boyutlu olarak farklı bilgisayarda görüntülemek ve topun, yazılan programı kullanarak kullanıcı isteğine göre tabla üzerinde hareket etmesini sağlamak.

# **2.Donanım Modülü**

## **2.1 Genel Tanım**

Bu modül “Ball on Plate” projesinin gerçeklenmesi için gerekli donanım birimlerinin senkron şekilde çalışmasını sağlamayı amaçlamaktadır. Platform iskeletinin kurulması, kullanılacak olan dokunmatik panel, servo motorlar ve diğer bileşenlerin platforma monte edilmesi ve arduinoyla olan elektronik bağlantılarının yapılması bu modül tarafından sağlanacaktır.

## **2.2 Kullanılacak Donanım Birimleri**

Donanım modülündeArduino Mega kullanılacaktır. Tabla üzerine konulan topun konum verisini alabilmek için dokunmatik panel ve bu paneli hareket ettirebilmek için servo motorlar kullanılacaktır. Platformu kurabilmek için gerekli tahtalar marangozdan önceden belirlenen ölçülerde alındı. Donanım birimleri ve platform arasındaki bağlantıları sağlamak için jumper kablo, balljoint, alüminyum çubuk, yeke, kızılötesi alıcı, mantar LED ve breadboard kullanılacaktır. Servo motorları çalıştırmak için AC/DC adaptör, Arduino’yu bilgisayar üzerinden çalıştırmak ve veri alışverişi yapabilmek için ise USB kablo kullanılacaktır. Ayrıca sistem tarafından dengeye getirilmek için kullanılacak top için 35 mm çapında 176 gram ağırlığında demir bilye kullanılacaktır.

## **2.3 Platformun Kurulması**

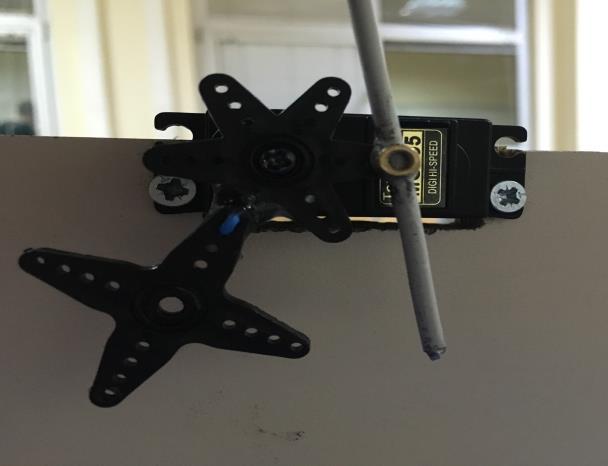


### **2.3.1 Tablanın Monte Edilmesi**

Motorlar yardımıyla tablanın istenilen yönde dönmesini sağlayacak olan mafsal, 20 cm uzunluğundaki tahta çubuğa açılan oyuk yardımıyla monte edildi. Ardından donanım modülünün denge konusunda sıkıntı yaşanmaması için tablayı taşıyacak olan tahta çubuk 40\*25 cm uzunluğuna sahip sunta üzerine sabitlendi. Dokunmatik paneli taşıyacak olan 36\*25 cm uzunluğundaki üst tabla, mafsal üzerine monte edildi.

*Şekil 1 : Tahta çubuk ile üst tabla arasındaki mafsal*

### **2.3.2 Motorların Monte Edilmesi:**



Donanım modülünde tabla hareketini sağlamak için 2 adet [Tower Pro MG995 RC Servo Motor](http://www.robotkutusu.com/Tower-Pro-MG995-RC-Servo-Motor,PR-14.html) kullanıldı. Öncelikle üst tablanın birbirine dik iki kenarının orta noktaları belirlendi. Belirlenen noktalara motorlara bağlanacak çubukların denk gelmesi için gerekli ölçümler yapıldı. Bu noktalara “yeke” isimli parça vidalandı. Alt tablaya motorları sabitlemek için kullanılacak olan 8 cm yüksekliğindeki tahta parçasının her birine 4\*2 cm’lik oyuk açıldı. Servo motorlar bu oyuklara yerleştirilip vidalandı.

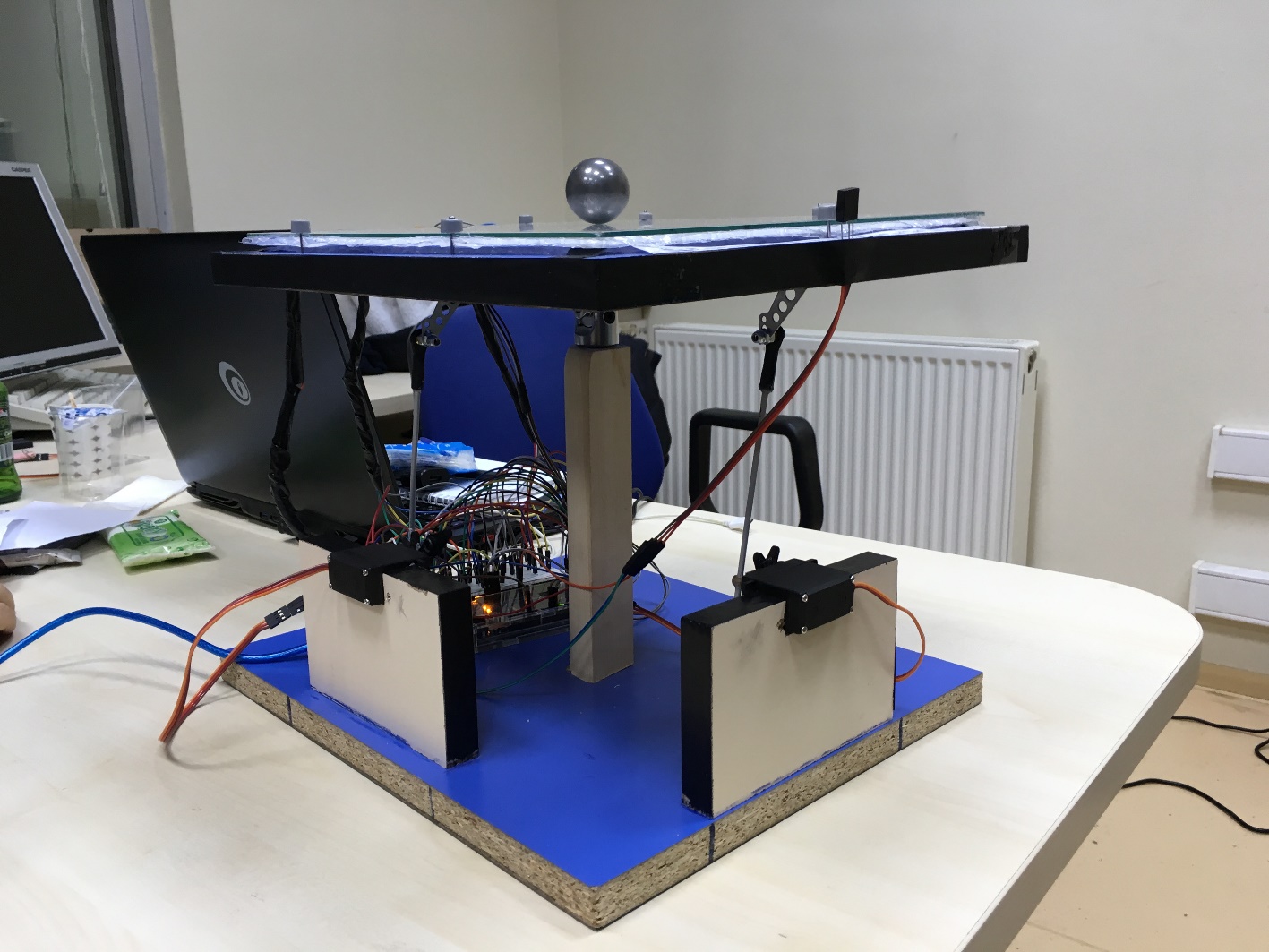
*Şekil 2 : Servo motor ve sabitlenmiş alüminyum çubuk*

### **2.3.3 Motorlar ile Tabla Arasındaki Bağlantının Sağlanması**



Üst tablanın motor hareketlerine uygun olarak her yönde hareket edebilmesini sağlamak amacıyla dana gözü diye tabir edilen “balljoint” isimli parça satın alındı. Bu parça, üst tablanın orta noktalarına sabitlenen yekeye takıldı. Ardından daha önce motorlara bağlanan 3 mm kalınlıktaki alüminyum çubuklar balljoint’e yapıştırılarak motorlar ile tabla bağlantı sağlandı.

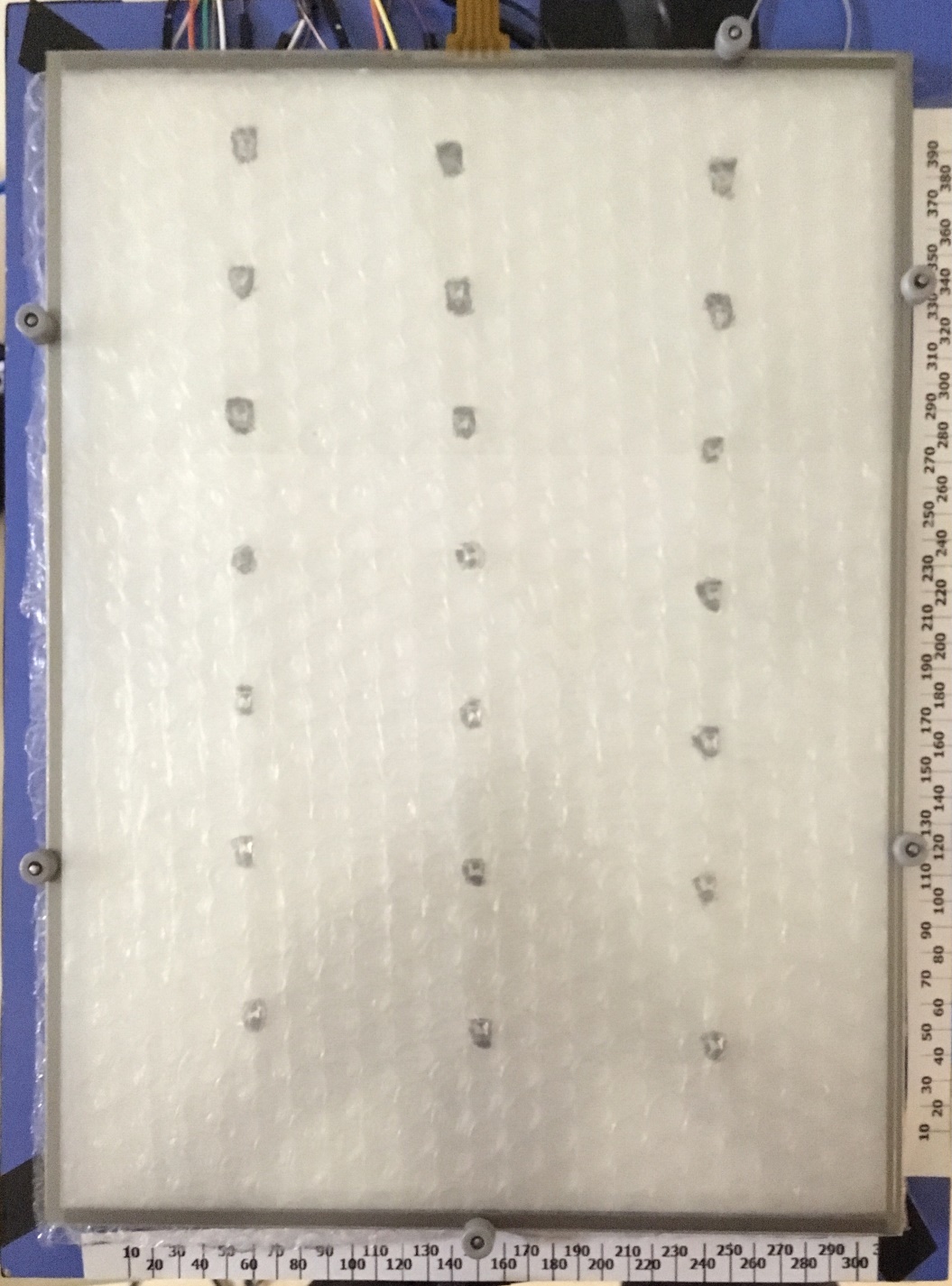
*Şekil 3: Üst tablaya sabitlenmiş balljoint'in görünümü*



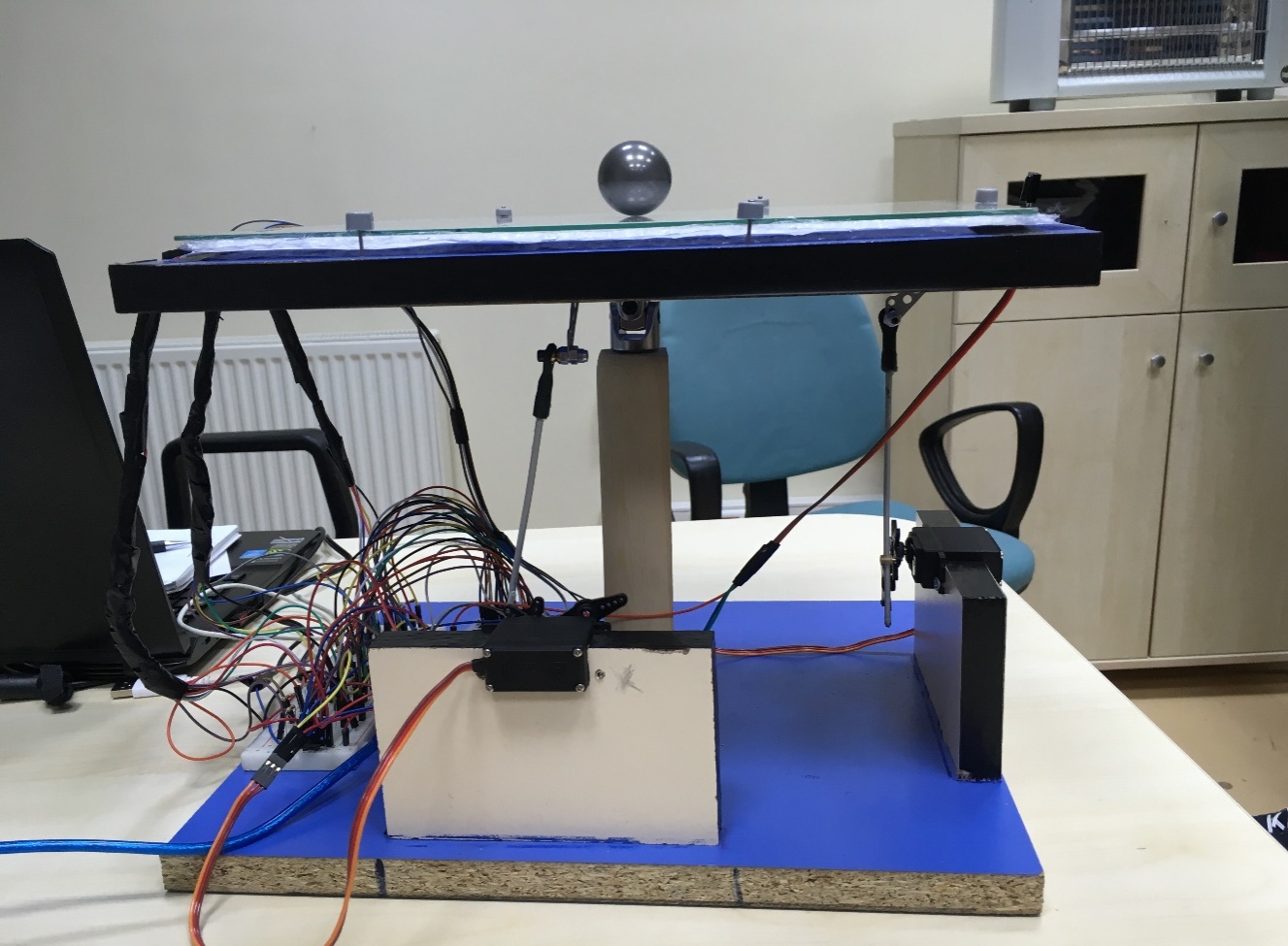
*Şekil 4: Donanım modülününgenel görünümü*

### **2.3.4 Dokunmatik Panelin Tabla Üzerine Sabitlenmesi**

Bu işlem için plastik raptiyeli çiviler kullanıldı. Bu çiviler yardımıyla üst tablaya dokunmatik panel boyutlarında 7 adet delik açıldı ve bu noktalardaki çiviler yardımıyla dokunmatik panelin tabla üzerinde hareketsiz şekilde sabit kalması sağlandı. Tabla üzerinde mafsalı sabitlemek amacıyla kullanılan yapıştırıcıların, dokunmatik ekranın yatay konumda düz şekilde durmasını engellediği için dokunmatik panelin tabla ile arasındaki kısmına balonlu naylon koyuldu. Ardından dokunmatik panelin bağlantı kablolarının zarar görmemesi için pin girişleri üst tablanın altına bant yardımıyla yapıştırıldı.Panelin etrafına konum bilgisinin bilinebilmesi için konum gösteren cetvel eklendi.Dokunmatik panelin kısa kenarı(x) yaklaşık olarak 869 piksel ve uzun kenarı(y) 1209 piksel olduğu için bu değerler biraz daha küçük değerler olan 300 ve 400’e mapedildi.Dolayısıyla x kenarının birimi piksel/2,897 ve y kenarının birimi piksel/3,0236 olarak belirlendi.



*Şekil 5: Sabitlenmiş dokunmatik panelin görünümü*



*Şekil 6: Donanım modülünün genel görüntü**sü*

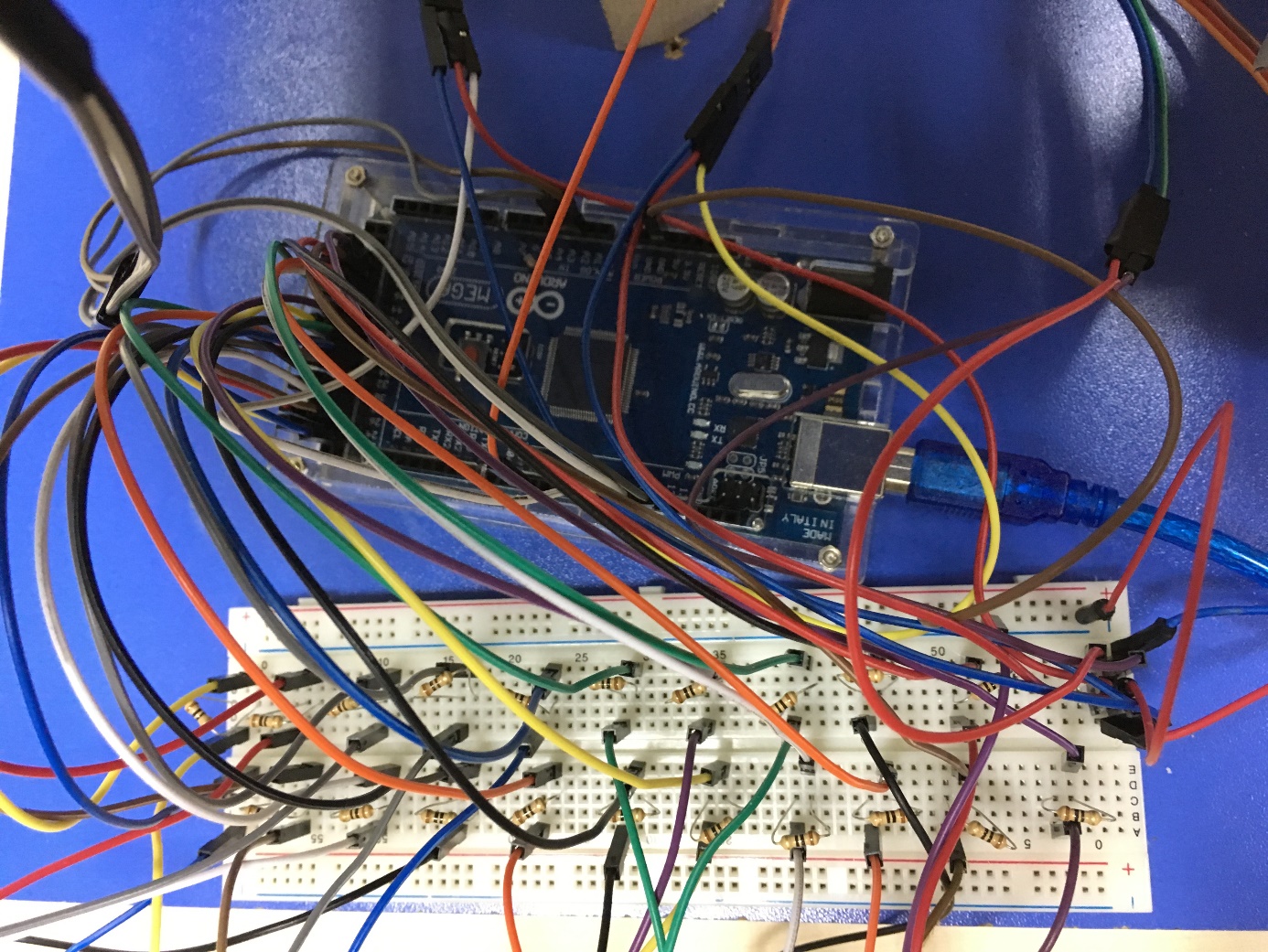
## **2.4 Donanım Elemanları Arasındaki Elektronik Bağlantıların Sağlanması**

### **2.4.1 Dokunmatik Ekranın Arduino’ya Bağlanması**

Dokunmatik ekranın sırasıyla Y+ X- Y- X+ koordinatlarını belirten 4 adet çıkışı vardır. Bu çıkışlardan Y+ ve X- Arduino üzerinde analog pinlere, Y- ve X+ çıkışları ise dijital pinlere bağlanır.

**2.4.2 Servo Motorların Bağlanması**

Söz konusu servo motorları çalıştırmak üzere şebekeden gelen alternatif akımı 5 volt 5 amperlik bir doğru akıma dönüştüren bir adaptör satın alındı. Bu akım kaynağı breadboard’a bağlanarak 2 motora da güç verilmesi sağlandı. Servo motorlar üzerindeki turuncu renkli pwm çıkışı arduino üzerindeki dijital pinlere bağlandı. Bu bağlantı sayesinde servo motorların arduino üzerinden hareket ettirilmesi sağlandı.

 *Şekil 7: Platformun tabanında bulunan Arduino ve Breadboard*

### **2.4.2 5mm 3v Kırmızı Mantar LED’lerin Sisteme Dahil Edilmesi**

Projeye spaceimpact oyunundan esinlenilerek bir “oyun modu” eklendi. Söz konusu oyundaki mermi efektini vermek için donanım modülündeki dokunmatik panelin alt kısmına LED’ler eklendi.3 sıra halinde 7’şer olmak üzere toplam 21 LED kullanıldı. LED’lerin topraklama uçları dokunmatik ekranın altında tek bir hatta toplanarak dışarı verildi.5V çıkışları ise Arduino yardımıyla LED’leri ayrı ayrı yakıp söndürebilmek için teker teker dokunmatik ekranın altından dışarıya breadboarda verildi. Breadboard’da bu uçlara ek seri 10’arohmluk dirençler bağlandı. Bu dirençlerden alınan çıkışlar yönetilebilmek için Arduino’nun 5 voltluk girişlerine bağlandı**.**

### **2.4.3 Kızılötesi Alıcı Sensörün Sisteme Dahil Edilmesi**

Kızılötesi sensörolarak TK19 kullanıldı. Bu sensörüArduino ile programlamak için IRremote.h kütüphanesi kullanıldı. Kızılötesi sensördokunmatik panel üzerinde topun istenilen konuma getirilebilmesi için kullanıldı. Bu sayede kumanda ile hem oyun modu kullanılabilir, hem de serbest pozisyonda denge noktası değiştirilebilir. Sensörün 3 çıkışı bulunur. Bir çıkış veri okumak amacıyla Arduino’ya, diğer iki çıkıştan biri topraklama diğeri ise 5V girişine bağlanır. Kızılötesi ışın yayan herhangi bir kumandanın tuşları kullanılarak topundenge noktasıdeğiştirilerek topun panel üzerinde hareket etmesi sağlandı.

# **3. Yazılım Modülü**

## **3.1 Genel Tanım**

Bu modül tablanın orta noktasında ve aynı zamanda verilen X ve Y koordinatlarında topu dengeye getiren servo motorlarının ihtiyacı olan gerekli değerleri hesaplayarak servo motorlarına açı vermektir.

## **3.2. Modül Özellikleri**

Dokunmatik panelden gelen X ve Y koordinatları alınarak bu değerler map edilir. Map edilmiş yeni değerler PID algoritmasına gönderilir ve istenilen X ve Y koordinatlarına getirilmek için PID’den gelen servooutputları alınır. Bu outputlarservo motorların alacağı açıya göre map edilir. Map edilmiş açılar servolara gönderilerek işlem tamamlanır.

## **3.2.1 Modül Temelleri**

Bu kısım modülü oluştururken kullanılan algoritma için kullanılan PID kontrol ve işlemleri açıklanacak ve PID Tuning sorununun nasıl çözüldüğünden bahsedilecektir.

### **I. PID**

PID oransal-integral-türevsel denetleyici PID kontrol döngüsü yöntemi, yaygın olarak kontrol sistemlerinde kullanılan genel bir kontrol döngüsü geribildirim mekanizmasıdır. Bir PID denetleyici ölçülü bir süreç içinde değişen ve istenilen ayar noktası ile arasındaki farkı olarak bir "hata" değerini hesaplar. Controller process kontrol girişini ayarlayarak hatayı en aza indirerek istenilen ayar değerine ulaşmak için çalışır.

PID algoritması üç ayrı sabit parametreyi içerir ve buna göre bazen üç aşamalı kontrol denir: oransal, P ile gösterilir integral I ile gösterilir türev değerleri, D ile gösterilir.

Sezgisel olarak, bu değerler mevcut değişim göz önüne alınarak zaman açısından şu şekilde yorumlanabilir; P mevcut hataya bağlıdır, I geçmiş hataların toplamı ve D gelecekteki hataların bir tahminidir.

### **II. PID Tuning**

Pidtuning için kullanılan farklı yöntemler vardır. Bu katsayılar sistemden sisteme göre çok farklılık gösterir. Bizim kullandığımız yöntem şu şekildedir: Üç katsayı da sıfırlanır. Kd değeri sistemin topun hızına karşı gösterdiği tepkidir. Diğer değerler sıfır iken Kd değeri yavaş yavaş arttırılır. Bu değer fazla düşükse yeterli açı sağlanmadığı topun hızını sıfırlanamaz ve top sistemden düşer. Eğer bu değer fazla yüksek ise sistem topun hızına gereğinden fazla tepki göstererek git gide topun hızını ters yönde arttrırız ve topun sistemden dışarı çıkmasına neden olur. Bu faktörler göz önünde bulundurularak Kd değeri arttırılır ve sistem gözlemlenir. Kd 'nin doğru değeri bulunur. Uygun Kd değeri bulununca bu değer sabit bırakılır ve Ki değeri sıfır iken Kp değeri arttırılır. Kp değeri topun denge noktasından uzaklığına göre vereceği tepkiyi belirler. Bu değer fazla düşük ise sistem yeterli açıda kalkmaz ve top ters yönden sistemden dışarı çıkmasına neden olur. Bu değer fazla yüksek ise belirli bir uzaklıkta olan topu gereğinden fazla kaldırdığı için denge konumunu çok fazla geçerek sistemden dışarı çıkmasına sebep olur. Sıfırdan başlanarak bu değer arttırılır ve sistem gözlemlenir. Bir noktaya konulan top denge konumuna doğru yeterli miktarda hız kazanarak hareket ettiği durum bu değerin doğru olduğu değerdir. Kp değeri de bulunduktan sonra Kp ve Kd değerleri sabit tutularak Ki değeri sıfırdan başlanarak arttırlır. Ki dengeye getirilmeye çalışılan noktadaki sapma olarak nitelendirilebilir. Örneğin bu değer düşükse top denge noktasının gerisinde yüksek ise ilerisinde dengelenir. Bu değer de sıfırdan başlanarak arttırılır. Topun tam denge konumunda dengelenmesi durumunda bu değer de doğru bulunmuş olur ve pidtuning işlemi tamamlanmış olur. Tuning işlemi yaplırken iki eksende çalışmak gözlemlemeyi zorlaştırdığı için iki eksen ayrı ayrı bulunur.

### **III. Mapping**

Map genel olarak, verilen değerleri, istenilen değerlere oranlanmasıdır. Sistemimizde 2 yerde map işlemi kullanılmaktadır. Dokunmatik panelden gelen koordinatlar ve servo motorlara verilecek açılardır.

**Dokunmatik panelden gelen X ve Y koordinatlarının maplenmesi**

* Dokunmatik panelden X ve Y’nin alt ve üst sınır değerleri, istenilen alt ve üst sınırlarınaoranlanır.

**PID Compute'den gelen hata değerinin maplenmesi**

* Donanımın tam düz şekilde dengede olduğu durum x ekseninde 75, y ekseninde 95derecedir. PID’de hesaplanan hata değeri map edilirken bu değerler göz önünde bulundurulmuştur. Map işlemini etkileyen bir diğer faktör de x eksenindeki servonun denge konumunun altında ve üstündeki açı değişimlerinin sistemde aynı oranda aşı değişimi oluşturmamasıydı. Hesaplanan hata değerleri y ekseni 10,180 şeklinde x ekseni 0’dan küçük ise 0 - 75 değerleri arasına büyük ise 75 - 115 değerleri arasında map edildi.

## **3.3 MODLAR**

### **3.3.1 Dikdörtgen Çizme Modu**

Kare çizilmesi için 3D Modülünde bulunan buton aracılığıyla gelen input ile yazılımda bulunan kare çizme algoritması çalışmaktadır.

### **3.3.2 Daire Çizme Modu**

Daire çizilmesi için 3D Modülünde bulunan buton aracılığıyla gelen input ile yazılımda bulunan daire çizme algoritması çalışmaktadır.

### **3.3.3 Şekil Çizme Modu**

Şekil çizilmesi için 3D Modülünde bulunan buton aracılığıyla gelen input ile yazılımda bulunan şekil çizme algoritması çalışmaktadır.

### **3.3.4 Oyun Modu**

Oyun moduna geçilmesi için 3D Modülünde bulunan buton aracılığıyla gelen input ile yazılımda bulunan oyun modu algoritması çalışmaktadır.

Touchscreen’in altına 3 şerit halinde 21 adet ledyerleştirilmiştir. Her şerit hareket eden bir kurşunu temsil etmektedir. Şerit halinde yanan led’ler topa doğru gelmektedir. Kumandayı kullanan oyuncu rastgele olarak yanarak gelen kurşundan topu kaçırmaktır. Eğer ki top ile gelen mermi(led) denk gelirse bütün ledler yanmakta ve oyuncuya yandığını haber etmektedir. Oyunun tekrarlığı için bütün ledler söner ve şerit halinde random olarak mermiler gelmeye devam etmektedir.

## **C:\Users\deniz\AppData\Local\Microsoft\Windows\INetCache\Content.Word\akışdiagramı.gif****3.4 AKIŞ DİYAGRAMI**

Şekil 8: Akış Diagramı

### **I. BalancePlate**

Sistem çalıştığında platformun tam dengeye gelmesini sağlanır.

### **II. ModeInput**

Donanımın çalıştıracağı modun seçimi yapılır.

### **III-I Circle**

Top touchscreen’i üzerinde çember çizer.

### **III-II Rectangle**

Top touchscreen’i üzerinde dikdörtgen çizer.

### **III-III Draw Shape**

Önceden touchscreeen’e çizilen şekli, top tekrar çizer.

### **III-IV Game**

Donanım oyun moduna geçer. Top kumanda ile yönlendirilerek mermiyi temsil eden ledlerden kaçırılması sağlanır.

### **IV. CoordinateMapping**

Dokunmatik panel üzerinden alınan X ve Y koordinatlarının sistemimize uygun olarak map edilir.

### **V.PID**

Uygun hale getirilmiş X ve Y koordinatlarını PID Compute fonksiyonuna vererek output değerleri hesaplanır.

### **VI.AngleMapping**

Gelen hata değerini map edilir ve uygun açı değerleri elde edilir.

### **VII.Output**

Map edilerek elde edilen uygun açı değerleri output olarak oluşturulur ve servo motorlarına input olarak verilir.

### **VIII.ServeEngines**

Gelen açı değeri ile servomotorlar gerekli açıyı yapar ve döngü içinde denge konumuna getirilene kadar tekrar edilir.

### **IX.RemoteInput**

Topun kaçması için yönlendiren topun kumandadan alınan inputudur.

### **X.LedOptimization**

Mermi şeritleri random seçilir ve ledlerin yanma sırası ayarlanır.

# **4. 3D Modülü**

## **4.1 Genel Tanım**

Bu modül tasarlanmış ve çalışır duruma getirilmiş projenin sanal ortamda 3 boyutlu bilgisayar ortamına aktarmak ve gerçek zamanlı olarak donanımsal hareketi bilgisayar ortamında göstermeyi içerir. Sistemden gelen topun konumu ve platformun oryantasyonu ile ilgili veriler kullanılarak 3D görüntü oluşturulur. Bunun yanında, gelen x ve y değerlerinin hata payları zamana göre grafiğe aktarılır.

## **4.2 Modül Özellikleri**

Sistemden gelen topun konumu ve platformun oryantasyonu ile ilgili veriler kullanılarak 3D görüntü oluşturulur.

Görüntüyü sanal ortamda oluşturmak için Irrlicht oyun motorundan yararlanılmıştır. Sanal ortamda oluşturulan görüntü kullanıcıya gösterilmek üzere ara yüze aktarılır.

### **4.2.1 PID Hata Grafiği**

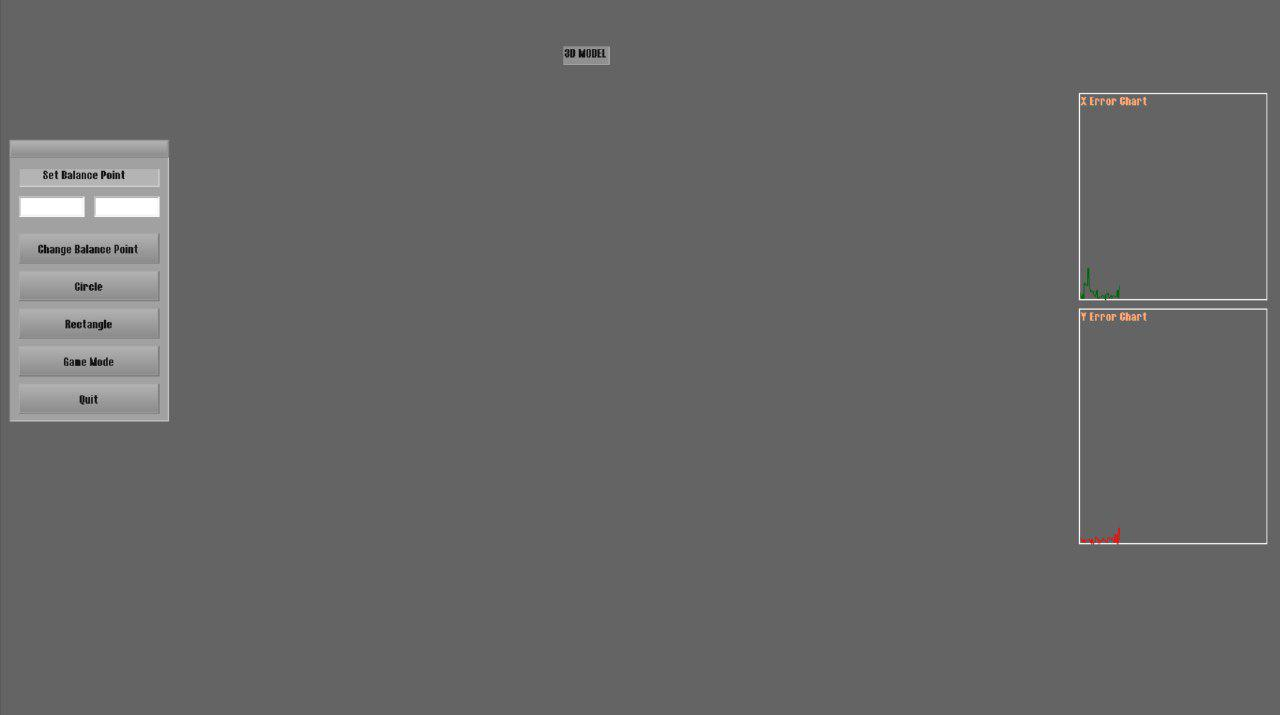
Topun anlık konumundan, o anda olması gereken denge konumunun farkının değerinin mutlak değeri alınarak x ve y koordinatları için ayrı ayrı çizgi grafiği ile arayüzde grafikler göstrilmektedir.

### **4.2.2 Arayüz Tasarımı**

Ara yüz kısmı 3D modülünün içerisinde aynı oyun motorunun ara yüz desteği kullanılarak implement edilmiştir.

Şekilde ekran görüntüsü verilen ara yüzde, sol panel sistemin denge noktasını değiştirebilmek için gerekli olan x ve y koordinatlarını alır. Change Center butonuna basılarak string olarak alınan koordinatlar sistemde kullanılmak üzere integer’açevirilir ve veriyi göndermek için hazır hale getirilir.

*Şekil 9: Arayüz tasarımının görünümü*



Topun denge noktasına getirilirken x ve y koordinatlarında yaşanılan hata paylarının zamana göre grafiği ise gerçek zamanlı bir şekilde ekranda yer alır.

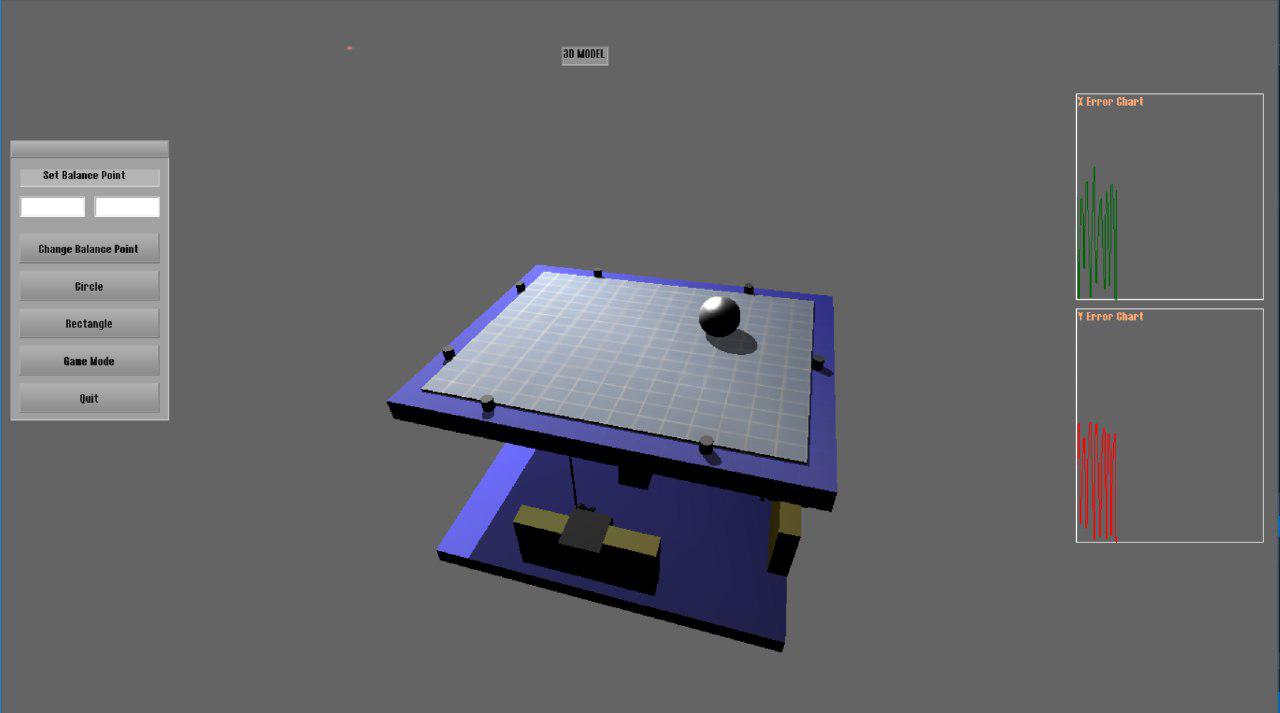
Sağ panel ise topun güncel koordinatlarını gerçek zamanlı olarak ekranda listelenir.

### **4.2.3 3D Tasarımı**

3DMAX programında objeler. mtlve. obj dosyaları halinde oluşturulur. Bu dosyalar oyun motoru içerisine import edilerek 3D modelleme gerçekleştirilir. Donanımdan alınan koordinatlar ile top ve tablanın yeni konumları çizdirilir. Arduino’dan gelen değerlere göre donanımın hareketi 3D modelleme ile bilgisayar ortamında gerçek zamanlı olarak gösterilir.

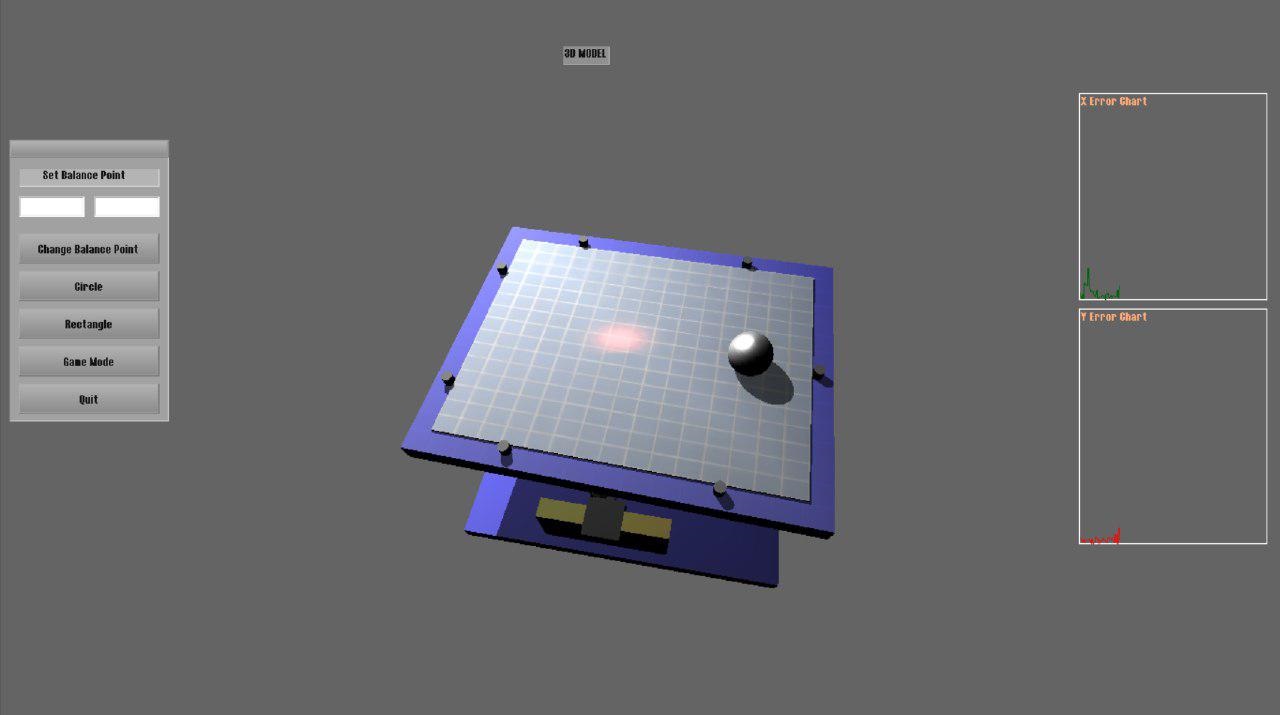
Sistemin bire bir 3D görüntüsü yapılmaya çalışıldı. Tablanın üst kısmında dokunmatik ekran raptiyeli çivilerle tutturulduğu için 3D görüntüsü de o şekilde tasarlandı. Ball on Plate oyunu için gerekli ledler ise dokunmatik panelin hemen alt kısmında konumlandırıldı.

Bu alınan veriler topun yüzey üzerindeki x ve z konumlarını ve tablanın o anki açısını içerir. Her framede tabla ve top çizilir fakat konum ve tablanın açısı 5 framede bir güncellenir. Bunun sebebi daha gerçekçi bir hızda görüntüyü elde etmektir. Tabla (0,0) konumuna çizilir ve konumu değişmeyecektir. Sadece x ve z eksenlerinde rotate işlemi yapılacaktır. Irrlicht kütüphanesinin **rotate()** fonksiyonu localrotation yaptığı için bizim için yeterli değildir. Çünkü bir eksende döndürme işlemi yapıldıktan sonra tablanın eksenleri değiştiği için diğer eksendeki döndürme işlemi istediğimiz şekilde gerçekleşmeyecektir. Bizim tablayı global eksenlere göre döndürmemiz gerekmektedir. Bunun için globalrotation yapan **rotateNodeInWorldSpace()** yardımcı fonksiyonu yazılmıştır. Tabla ve sistemin diğer bileşenlerinin çiziminden sonraki işlem topun doğru konuma çizilmesi işlemi vardır. 3D modülüne gelen konum verisi topun yüzeydeki konumudur. Fakat Irrlicht kütüphanesinde global koordinat sistemine göre çizdirme işlemi yapıyoruz. Topu doğru konuma çizmek için öncelikle top gelen verideki konuma çizilir. Bundan sonra top tablanın açı değerlerine göre orijin noktası etrafında döndürülür. Bu yaptığımız işlem için de yardımcı bir fonksiyon yazıldı. Çünkü bu döndürme işleminin global eksende belirli bir pivot noktası etrafında yapılması gerekmekteydi. Bu işlem için **revolveNodeInWorldSpace()** fonksiyonu yazıldı. Topun ve tablanın 3 boyutta doğru bir şekilde çizimi gerçekleştirilmiş oldu. 3D algısını veren en önemli etkenlerden biri ışık ve dolayısıyla gölgedir. Projeye directionallight eklendi ve görüntünün daha gerçekçi olması sağlandı. Sistem sürekli hareketli olduğu için sabit bir kamera görüntünün istenilen şekilde görülmemesine sebep olduğu için kamera hareketli hale getirildi. Kamera orijin etrafında mouse hareketiyle döndürülerek istenilen açıda bakılabilmesi sağlandı.



Şekil 10: 3D modülün yandan görünümü

Ledlerin çalışma prensibi ise şöyledir, oyun modu algoritması merminin gelebilmesi için rastgele bir koordinat belirler. Bu koordinattaki led yakılır ve aynı eksen boyunca diğer ledler de sıra ile yanar. Birisi yandığında bir önceki led sönmelidir. Oyuncu kumanda ile topu hareket ettirir ve mermiden kaçmaya çalışır. Topun koordinat verisi Irrlicht oyun motoruna verilerek 3D ekranda da topun aynı şekilde hareket ettirilmesi sağlanır. Eğer top mermiden kaçamazsa oyun başarısızlıkla sonuçlanır ve tüm ledler(21 tane) aynı anda yanar. Aynı şekilde 3D ekranda da ledler, gelen konum bilgisine göre yanar.



*Şekil 11: 3D Modülününled yakarken üstten görünümü*

# **5. Donanımlar Arası Haberleşme**

3D modülü ile yazılım modülü arasındaki haberleşmeyi USB bağlantısı kullanarak sağlamak için RS232.h kütüphanesini kullanıldı. Söz konusu kütüphane cross-platform çalışmayı desteklemektedir.

**RS232\_OpenComport()** fonksiyonu Arduino’ya bağlanmak için port açmaktadır.

**RS232\_PollComport()**fonksiyonu Arduino tarafından Serialport’a yazılan buffer’ı okumaya yaramaktadır.

**RS232\_cputs()** fonksiyonu 3D modülündenArduino’ya veri göndermeye yaramaktadır.

## **5.1 Yazılım Modülünden 3D Modülüne Veri Gönderilmesi**

Dokunmatik ekrandan gelen veriler, PID algoritması kullanılarak motorlara gönderilen açı değerleri, donanım modülünde hangi LED’in yakıldığı ve güncel denge noktası Serial Port kullanılarak 3D modülüne gönderildi. Bu modül yukarıda belirtilen RS232\_PollComport() fonksiyonunu kullanarak bu değerleri okur ardından gerekli parse ve mapping işlemlerini yapar. Bu değerler kullanılarak 3D modülünde donanım modülüne senkronize şekilde modelleme yapılır.

## **5.2 3D Modülünden Yazılım Modülüne Veri Gönderilmesi**

Kullanıcı, 3D modülündeki butonları kullanarak yazılım modülüne şekil çizme, oyun modunu açma ve denge noktası değiştirme gibi istekler gönderebilir. Bu istekler yukarıda belirtilen RS232\_cputs() fonksiyonu kullanılarak gönderilir. Bu işlemi gerçekleyebilmek için yazılım modülünde sürekli olarak 3D modülünden veri gelip gelmediği kontrol edilmektedir. Eğer ki söz konusu veri gelirse istenilen işlem yapılır.