



**T.C.**

**GEBZE TEKNİK ÜNİVERSİTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**ŞEKİL DEĞİŞTİREBİLEN DOKUNSAL NAVİGASYON**

**BİL 395 PROJE I - GRUP 4**

**PROJE ÇALIŞANLARI:**

Bayram YÜKSEL

Burhan OĞUR

Zehra Buse IŞIK

Hazel PALA

Ömer BAŞKURT

Sevgi BORAZAN

Yavuz Oğuz İPEK

Serap ASLAN

**DANIŞMAN : Prof. Dr. Erkan ZERGEROĞLU**

## İÇİNDEKİLER

1	ÖZET .....	4
2	GİRİŞ .....	5
2.1	Amaç.....	5
2.2	Hedef Kitle .....	5
2.3	Proje Kapsamı.....	5
3	PROJE DONANIMI .....	6
3.1	DONANIM .....	6
3.1.1	Küp .....	6
3.1.2	Modüller .....	8
3.1.3	Arduino Mega ADK.....	8
3.1.4	GSM/GPRS Shield V3 .....	8
3.1.5	GY-NEO6MV2 GPS Modülü .....	9
3.1.6	ESP8266 .....	9
3.1.7	28 BYJ-48 Redüktörlü Step Motor .....	10
3.1.8	HMC5883L 3 Eksenli Pusula Sensörü .....	10
3.1.9	Powerbank ( Güç Kaynağı ) .....	10
3.1.10	Kısaltma Listesi .....	10
3.2	YAZILIM .....	11
3.2.1	Donanım Yazılımı .....	11
3.2.2	Server Yazılımı .....	13
4	PROJE ARA YÜZÜ ( INTERFACE ).....	14
4.1	Harita Modelleme .....	14
4.1.1	Modellemede Kullanılan Programlar .....	14
4.1.2	Modelleme.....	14
4.2	Ara yüz Yazılımı (Interface Software) .....	16
4.2.1	Yazılımda Kullanılan Programlar .....	16
4.2.2	Arayüz (Interface) Ana Ekran .....	16
4.2.3	Interface Dış Mekan .....	18
4.2.4	Interface İç Mekan .....	19
5	PROJE SERVER'I.....	20
5.1	Shortest Path Sınıfı .....	20
5.2	Use Case Diyagramı .....	21
5.2.1	Kullanım Durum Diyagramı .....	21

5.2.2	Etkinlik Diyagramı .....	22
6	SONUÇ .....	22
6.1	Gerçekleřtirdiklerimiz .....	22
6.2	Gerçekleřtiremediklerimiz.....	22
6.3	Çalıřma Esnasında Oluřabilecek Aksilikler .....	23
7	KULLANIM KILAVUZU.....	23
8	KAYNAKLAR .....	25

# 1 ÖZET

Bu proje Gebze Teknik Üniversitesi Bilgisayar Mühendisliği BIL395 Projesi olarak hazırlanmıştır. Proje genel olarak kullanıcıdan gelen emirlere göre hareket özelliğine sahip dokunsal mekanizması ile kullanıcıyı iç ve dış mekanlarda yönlendirebilen bir el cihazının geliştirilmesidir.

Günümüzde navigasyon cihazlarının önemi giderek artmaktadır. Ancak gelişen ve önemi giderek artan navigasyon cihazları daha çok yazılımsal olarak kaldığı için hayatın her alanında ve her kullanıcıya hitap edememektedir. Bu projedeki en büyük amaç bir navigasyon cihazı geliştirirken hem donanımsal hem de yazılımsal modüller kullanarak , bütün kullanıcıların kullanabilmelerini sağlamaktır.

Kullanıcılar için farklı yönlerde hareket edebilen küp şeklinde bir cihaz geliştirilmiştir. Kullanıcılar bu cihazın ellerindeyken yaptığı ileri-geri ve sağ-sol hareketlerini dokunsal olarak algılayarak hangi yönde gitmeleri gerektiğini öğreneceklerdir. Gidecekleri yerin seçilmesi içinse gelişmiş ve sade bir arayüzde 3D harita kullanacaklardır. Harita üzerinde küpün nerede olduğunun gösterilmesinden sonra gideceği yerin seçilmesiyle beraber harita üzerinde gerçek zamanlı kişinin hareketleri yansıtılacaktır.

Tüm çalışmalar sonucunda bu projede hareket ederek kullanıcılara gidecekleri yönü belirten bir navigasyon cihazı oluşturulmuştur. Navigasyon cihazı iç mekanlarda ve dış mekanlarda kullanıcının bulunduğu yeri tanıyarak gitmek istediği yere göre gerekli yönlendirmeleri yapacak şekilde cihazı hareket ettirebilecektir.

## 2 GİRİŞ

### 2.1 Amaç

Server üzerinden aldığı hedef doğrultusunda kullanıcıları gidecekleri hedefe doğru yönlendirebilen, bağlı bulunduğu server ile kendi lokasyon bilgisini paylaşabilen bir avuç içi navigasyon cihazının donanımsal ve yazılımsal olarak geliştirilmesi amaçlanmıştır. Ayrıca kullanıcının elindeki cihazın bulunduğu yerin, geçmiş olduğu belirli nokta veya bölgelerin iç ve dış mekan olarak ayrı ayrı hazırlanmış olan harita modellemelerinde yansıtılabilmesi amaçlanmıştır.

Navigasyon cihazı asgari miktarda elektronik bileşen kullanılarak tasarlanmış olup, bir bütün cihaz halinde getirilmiştir. Cihazın yönlendirilmesini sağlayacak verimli ve sade bir server tasarlanmıştır. Son olarak da server ve cihaz haberleştirilmiş olup kullanıcıyı başlangıç noktasından varış noktasına doğru bir şekilde yönlendirerek en kısa yoldan hedefe varmasını sağlayan bir cihaz geliştirilmiştir.

### 2.2 Hedef Kitle

Geliştirilmiş olan avuç içi navigasyon cihazı, temel olarak görme engelli vatandaşların kullanabilmesi için tasarlanmış olup Gebze Teknik Üniversitesi'nde BIL 395 Proje 1 dersi kapsamında gerçekleştirilmiştir. Hedef kitle olarak, okul içerisinde bir yerden başka bir yere gitmek isteyen görme engelli vatandaşlar baz alınmıştır. Ayrıca bu projeyi değerlendirecek olan Prof. Dr. Erkan ZERGEROĞLU hedef kitle kapsamında yer almaktadır.

### 2.3 Proje Kapsamı

- Server kısmında sade bir ara yüz kullanılarak navigasyon cihazının lokasyon verisinin gelmesi ve bu verinin işlenerek anlamlandırılması sağlanmıştır.
- Kullanıcının gideceği varış noktasının seçilmesiyle cihazın izleyeceği yol en kısa yol algoritması kullanılarak hesaplanmıştır.
- Bu hesaplamalar doğrultusunda iç ve dış mekan olmak üzere iki farklı şekilde hazırlanmış olan modellemelerde cihazı kullanan kişinin bulunduğu yerin senkronizasyonu sağlanmıştır.

Tüm bu aşamalara geçilmeden önce,

- Server – Client haberleşmesinin nasıl sağlanacağı analiz edildi,
- En kısa yol algoritmasının implementasyonu yapıldı,
- Yazılımsal diğer gereksinimler çıkarıldı,
- Kullanıcı ve ara yüz için gerekli analizler yapıldı,
- Donanım için gerekli olan malzemelerin analizleri yapıldı.

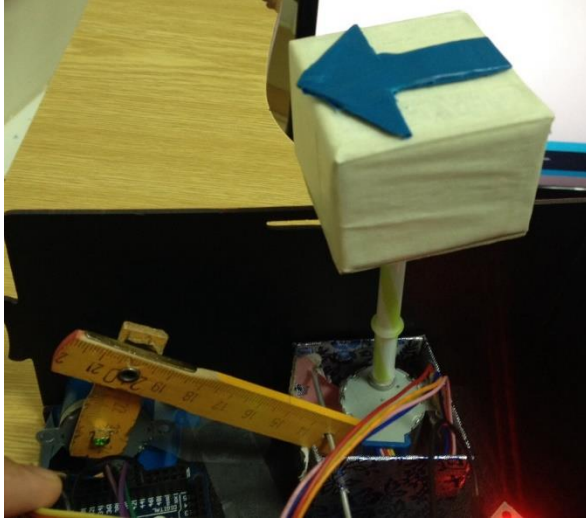
## 3 PROJE DONANIMI

### 3.1 DONANIM

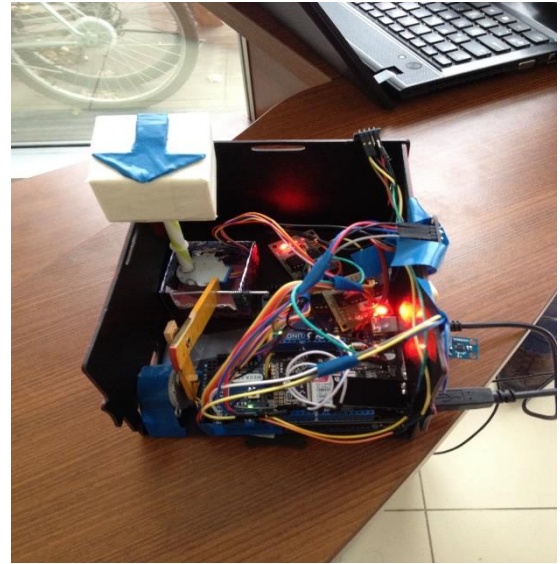
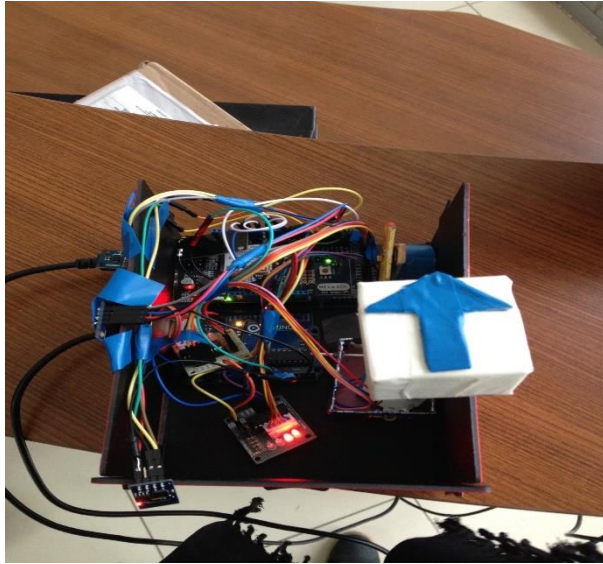
#### 3.1.1 Küp

Küp yapımında sert plastik kullanılmıştır. Yapıştırıcı madde yardımıyla Arduino'lar sabitlenmiş olup USB bağlantılarının sağlanabilmesi için gerekli düzenlemeler plastik küp üzerinde yapılmıştır. GPS modülü, Wi-Fi modülü ve compass modülü de daha iyi veri alabilmek için küpün dış kısmına sabitlenmiş olup daha iyi veri alabilmek için etrafı açık bırakılmıştır. Üst kısımda gidilecek yönü gösteren bir küçük küp mevcuttur ve bu küpün üzerinde doğru yönün belirlenmesinde kullanılan bir ok simgesi mevcuttur. Üst kısımdaki bu küp iki adet servo motor yardımıyla hareket ettirilmektedir. Bu küpün ileri hareketi için bir servo motor ve sağa-sola dönüş hareketi için ayrı bir servo motor kullanılmıştır.

Servo motorun hareketi dairesel bir hareket olduğu için bu dairesel hareketi düzlemsel harekete çevirmek gerekti. Dairesel hareketi düzlemsel harekete çevirmek için kullanmış olduğumuz yöntem ise **Slider Crank Mechanisms** olarak bilinen bir **Crankshaft Mechanism** methodu kullanılmıştır. Ana küpün içerisi bu mekanizmanın çalışabileceği şekilde tasarlanmıştır. Servo motorların konumları bu mekanizmaya göre belirlenmiş ve yardımcı olarak bir adet plastik silindir kullanılmıştır. Böylelikle servo motorun dairesel hareketi düzlemsel harekete başarılı bir şekilde çevrilmiştir.



Yandaki şekilde **Slider Crank Mechanisms** methodunun uygulanaşı gösterilmiştir. Sol alttaki step motorun dairesel hareketi, düzlemsel harekete çevrilmiştir. Böylece küpün ileri ve geri hareketlerini gerçekleştirmesi sağlanmıştır.



Küpün farklı açılardan görünümü

#### 3.1.1.1 Küpün Kullanımı

Varış noktasının seçiminin yapılmasının ardından gidilecek yol belirlenmiş olur. Daha sonra başlangıç noktasından hangi yöne döneleceği küp üzerindeki ok ile gösterilir. Doğru yöne döndükten sonra ok ileri hareket eder ve hareket etmeye başlanır. Dönüş yapılması gereken noktalarda ok ne kadar dönülmesi gerekiyorsa o kadarlık bir açı yapar ve gidilmesi gereken yönü gösterir. Ok un ileri çıkması düz gidileceği manasına gelmektedir ve zaten ileri çıkmış vaziyette olan ok un hareket etmemesi o doğrultuda gidilmesi gerektiği anlamına gelir. Varış noktasına gelindiğinde küp hareket etmez veya başlangıçta olduğu konuma geri döner. Böylelikle varış noktasına ulaşıldığı anlamına gelir.

### **3.1.2 Modüller**

Navigasyon cihazı Arduino geliştirme kartı üzerine projedeki gereksinimleri karşılayacak şekilde çalışabilecek modüllerin eklenmesiyle oluşturulmuştur. Navigasyon cihazının verimli olabilmesi adına farklı modüller bir arada kullanılmıştır.

### **3.1.3 Arduino Mega ADK**

Projede kullandığımız geliştirme kartı olan Arduino Mega ADK , Atmega2560 temelli bir mikrodenetleyici kartıdır. Bu projede Arduino kullanmamızın en büyük sebebi basit bir kullanılabilirliğe sahip olması, teknik destek ortamının fazlalığı ve kullanacağımız ekstra modüllerin büyük bir çoğunluğuna uyumlu olmasıdır. Arduino geliştirme kartları arasında Mega ADK seçmemizin sebebi ise Atmega2560 mikro denetleyicisi sayesinde , basit bir Atmega328 temelli UNO R32'den daha performanslı çalışması ve kompleks algoritmaların çalışabilirliği için üzerinde daha fazla memory'e sahip olmasıdır. Ayrıca Arduino Mega ADK kullanmamızın bir diğer sebebi de, üzerinde 54 tane dijital I/O pine sahip olmasıdır. Giriş seviyesi olan UNO R3'te sadece 14 dijital I/O pini bulunmaktadır. Kullanacağımız modüllerin pin sayılarını hesapladığımızda GSM/GPRS shield hariç 14 tane dijital pin gereklidir(ESP8266 2, Step Motor 4x2, GPS 2, Pusula 2). Pin kullanımı 14 tane gibi gözükse de GSM/GPRS Shield'in Arduino geliştirme kartının üzerine takıldıktan sonra 4 tane daha pin kullandığı için normal bir UNO R3 yetersiz kalmaktadır. Bu sebeplerden ötürü Arduino Mega ADK kullanarak güzel ve verimli bir geliştirme oluşturduk.

### **3.1.4 GSM/GPRS Shield V3**

Navigasyon cihazının dış mekanlarda çalışıp server'la iletişim kurabilmesi adına GSM kullanmayı tercih ettik. Yaptığımız araştırmalarda alternatif yöntemlerin verimsiz olabileceği kanaatine vardık. Mesela radyo sinyali kullanmanın getirdiği en büyük dezavantajlar belirli bir mesafede çalışması ve projeyi baz aldığımızda sıkça yaşanabilecek sinyal alanının dışına çıkılmasından kaynaklanabilecek bağlantı kopmalarıdır. Ayrıca üzerinde bulunan antenlerden ötürü, ortaya çıkacak ürünün navigasyon cihazından daha çok eski tip televizyon antenlerine benzemesine yol açacağı için tercih etmedik. Bir diğer alternatif olan wifi modülümüz SP8266'nın da çekim alanının bu proje için yetersiz kalmasından dolayı dışarıda haberleşmek için wifi modülü kullanamadık. Bir diğer yöntem olan navigasyon cihazının yakınında



telefon taşıyarak telefonun 3G'si üzerinden haberleşme tercihi de genel olarak verimli olmadığı için tercih etmedik ki zaten bu tip bir çözüm yönteminin daha çok Raspberry tabanlı kartlar için uyumlu olmaktadır. Bu sebeplerden ötürü 4 bant desteği olan ve üzerinde gelişmiş SIM900 bulunan SeeedStudio markalı bir GSM/GPRS shield kullanmayı tercih ettik. Shield'ın Arduino Mega ADK uyumlu olması, direkt olarak üzerine oturması , Arduino pinlerini yukarı taşıması ve yaklaşık 68x53mm'lik ergonomik yapısı sayesinde büyük bir kullanım kolaylığı sağlamaktadır. Shield'e takacağımız SIM kart ile Arduino internete erişebilir bir yapıya sahip olmaktadır. Statik bir IP üzerinden serverla iletişime geçerek donanımdan aldığı lokasyon ve yön verileri servera iletecek, aynı şekilde serverdan gelen hangi motorun nasıl çalışması gerektiği bilgisini de alarak step motorların dönebilmesi için gerekli ortamı oluşturmuştur.

### **3.1.5 GY-NEO6MV2 GPS Modülü**

GPS modülü ile dış mekanda koordinat belirlenerek gelen x, y değerleri işlenmek üzere GSM Shield üzerinden servera aktarılmaktadır. GY-NEO6MV2 üzerinde barındırdığı dahili seramik anteni sayesinde olabildiğince doğru koordinat değerleri üretebilmektedir. Hassasiyet değerinin yeterli olması ve teknik desteğinin fazlalığından ötürü kullanmayı tercih ettik. GPS modülü iç mekanda düzgün veri üretmediği için, sadece dış mekanda konum bulabilmesi için kullandık. İç mekanda bazı noktalarda veri alabilsek te gelen verinin tutarsız ve hata payının fazla olmasından ötürü kullanmayı tercih etmedik iç mekanda.

### **3.1.6 ESP8266**

ESP8266 wifi modülü 802.11 b/g/n desteği, 54 Mbit/s veri hızına sahip olması ve 2.4-2.5Ghz frekansta çalışabilir yapısından ötürü bu projede tercih ettik. Proje başlangıcında haberleşmeyi de Wifi üzerinden yapmayı planladığımızdan ötürü veri aktarma hızının ve çalışma frekans değerlerinin iyi olmasına dikkat ettik. ESP8266'nın kolay kullanımı ve teknik desteğinin fazla olması da bu ürünü kullanmamızdaki bir diğer etmen olmuştur. ESP8266'yı iç mekanda lokasyon bulunabilmesi adına kullandık. İç mekan bölgelerinde yer alan AP hâlindeki telefon, bilgisayar vesairelere erişerek sinyal değerlerini ve Mac adreslerini alıp işleyerek yer tespiti yaptık.

### 3.1.7 28 BYJ-48 Redüktörlü Step Motor

Bu projede navigasyon cihazının ileri-geri ve sağa-sola dönme hareketi yapabilmesi adına bipolar step motor kullanmayı tercih ettik. İki yönlü çalışabilir olması , adım açılı çalışması ve mekanik bakımının kolay olmasından ötürü step motor kullanmayı tercih ettik. Servo motorlara göre RPM değerinin düşük olması, açık çevrim yapısı ile daha kolay kontrol edilebilme olanağı sağlamaktadır. Bu sebeplerden ötürü servo motor yerine bipolar 4 fazlı 5.625°/64 adım açısına sahip redüktörlü step motor kullandık. Step motorun kontrol edilip kullanılabilmesi adına ULN2003A Step Motor Sürücü Kartı da kullandık.

### 3.1.8 HMC5883L 3 Eksenli Pusula Sensörü

Navigasyon cihazının temel özelliğinden biri olan olan yön verisinin elde edilmesi için I<sup>2</sup>C dijital haberleşme protokolünü kullanan ve  $\pm 1,3-8$  Gauss ölçüm değerine sahip HMC5883L Pusula Sensörü kullandık. Kullanıcıların başlangıç yönünün tespiti ve gideceği hedefe doğru ilerlerken yol üzerindeki noktalar arasındaki yön bilgilerinin elde edilebilmesi adına pusula sensörü kullandık. Kuzeyle yaptığı açıyı 0 alıp, gelen veriler üzerinden cihazın nereye baktığı bilgisine ulaştık.

### 3.1.9 Powerbank ( Güç Kaynağı )

Arduino Mega ADK geliştirme kartının 5V gerilimde çalışmasından ötürü , beslemeyi 5V değere sahip bir powerbank üzerinden sağladık.

### 3.1.10 Kısaltma Listesi

<b>ADK</b>	: Android Development Kit
<b>I/O</b>	: Input & Output
<b>GPS</b>	: Global Positioning System
<b>GSM</b>	: Global System for Mobile Communications
<b>GPRS</b>	: General Packet Radio Service
<b>AP</b>	: Access Point
<b>IP</b>	: Internet Protocol Address

## 3.2 YAZILIM

### 3.2.1 Donanım Yazılımı

Navigasyon cihazı iç ve dış mekanda çalışabilir yapıda olacağından ötürü, cihazın geliştirilme aşamasında yazılım kısmı iç ve dış olmak üzere iki parçaya ayırdık.

Bütün modüllerin iki kısımda da kullanılmamasından ötürü, geliştirilme aşaması 2 parçalı hâle getirilerek olası kompleks problemlerin oluşması engellenmiştir. Yazılım geliştirilme aşamasında öncelikle herbir modül kendi içinde bağımsız çalıştırılarak; çalışma yapısı öğrenilmiştir.

#### 3.2.1.1 İç Kısım

Bina içinde temel olarak server ile Arduino kartımızın haberleşebilmesi için gerekli server-client yazılımının oluşturduk. Okuldan aldığımız statik IP "193.140.134.42" ve kendi seçtiğimiz port üzerinden bağlantı kurulmasını sağladık. SIM900 ve GPRS\_Shield\_Arduino kütüphaneleri kullanarak oluşturduğumuz TCP bağlantısı sayesinde cihazımız ve serverımız arasında veri transferini sağladık. Bundan sonraki adım iç mekanda lokasyonun elde edilmesi aşamasıydı. Bu noktada GPS modülü yetersiz kaldığından ötürü ESP8266 wifi modülü ile yer tespiti yapmaya çalıştık. İç mekanda 11 bölge oluşturup , her bir bölgeye bir AP durumundaki telefon,pc yerleştirdik. ESP8266'nın bu cihazlara erişip veri alabilmesi adına temel AT komutlarını(AT+CWLAP , AT+CWJAP vs.) kullandık. Bir bölge içerisinde gelen AP verileri üzerinden en yüksek sinyal şiddetine sahip ve daha önceden kendi sistemimizde kayıtlı cihazı tanıdığımızda , o cihaz hangi bölgede ise navigasyon aletimizin konumunun o bölgede olduğunu anlamış olduk. İç mekanda ölçmemiz gereken son değer olan yön bilgisi için pusula sensörü üzerinden HMC5883L kütüphanesi fonksiyonlarını kullanarak pusulanın kuzeyle yapmış olduğu açı değerini hesapladık. Modüllerden gelen bölge numarası ve kuzeyle yapılmış olan açı değerini client-server üzerinden servera aktardık. Server kendi içinde bu değerleri işledikten sonra hangi motorun ne kadar döneceğini hesaplayarak gerekli verileri tekrardan gönderdiğinde dönme açısı ve motor bilgisinin gelen veriden çıkartılarak step motorların dönmesi sağlanmıştır. Step motorun doğru ve istenilen açıda döndürülebilmesi adına Stepper kütüphanesi üzerinden dönme adımları ve dönme hız değerleri belirlenmiştir.

Herhangi bir motor hareket ettiğinde önceden kaydetmiş olduğumuz cep telefonuna mesaj göndererek yaptığı hareketin bilgisini göndermektedir.

#### **3.2.1.2 Dış Kısım**

Bina dışında, iç mekanda kullanmış olduğumuz GSM bazlı client-server yapısını kullanarak başladık. İç kısımda kullandığımız statik IP üzerinden haberleşme sağlayarak dış lokasyon verilerini bu kanal üzerinden aktardık. Dış mekanda ESP8266 yerine GY-NEO6MV2 GPS Modülü üzerinden konum bilgisine erişmeye çalıştık. TinyGPS kütüphanesini kullanarak navigasyon cihazının x, y koordinatlarına eriştik. Koordinat değerlerinin virgülden sonra 6 basamaklık değerinde alarak hassasiyeti ve doğruluk payını arttırdık. İç mekanda olduğu üzere pusula sensörünü buradada kullanarak kuzeyle yapmış olduğu açı değerini hesapladık. Modüllerden aldığımız x, y koordinat verilerini ve cihazın yapmış olduğu açı değerini client-server üzerinden servera aktardık. Server kendi içinde bu değerleri işledikten sonra hangi motorun ne kadar döneceğini hesaplayarak gerekli verileri tekrardan gönderdiğinde dönme açısı ve motor bilgisinin gelen veriden çıkartılarak step motorların çalışmasını sağlanmıştır. Herhangi bir motor hareket ettiğinde önceden kaydetmiş olduğumuz cep telefonuna mesaj göndererek yaptığı hareketin bilgisini göndermektedir.

#### **3.2.1.3 Haberleşme**

GSM/GPRS Shield üzerinden iç ve dış mekanda haberleşme gerçekleştirebilmek adına SIM900 üzerinden kendi sim kartımızla internete açılarak client-server yapısının oluşturulması haberleşmenin temelini oluşturmaktadır. Bu yapı oluşturulurken öncelikle GPRS tanımlanıp gerekli ön çalışma verileri atanır. Sim kart verisi üzerinden internete erişip erişmediğinin kontrolünün sağlanmasından sonra sim kart kendi içinde bir IP alıp internete açılır. Daha sonra internet üzerinden statik IP numarasına ve portuna bağlanmayı deneyerek, serverla iletişime geçmeye başlar. Server ile veri transferi yaparken "gprs.send()" ile veri gönderme işlemi yaparken "gprs.recv()" ile de serverdan gelen verilerin alınmasını sağlar.

#### **3.2.1.4 Konum Bulma**

Lokasyon verilerinin elde edilmesinde en kritik noktalardan biride verinin hassasiyet oranının belirlenip ona göre oluşturulmasıdır. Okulun giriş kapısının koordinat değerleri 40.809205, 29.365657 iken bilgisayar mühendisliği binasının koordinatları 40.808027, 29.356291'dir. Bu veriler incelendiğinde virgül öncesi ve virgülden sonrasında ilk iki basamak değerinin

lokasyon da spesifik ve detaylı yerin bulunmasında yetersiz kaldığını görmekteyiz. Bu yüzden GY-NEO6MV2 GPS Modülünden gelen verileri kırpmadan virgülden sonraki 6 basamağını da hesaplayıp, veriyi o şekilde servera iletimini sağladık. Bu sayede hata payını minimuma indirgemiş olduk.

### **3.2.2 Server Yazılımı**

#### **3.2.2.1 İç Kısım**

Zemin katta her sınıf ayrı bir node olarak düşünülmüştür, birbirine komşu olan sınıflar yazılımsal olarak da komşu olarak tanımlanmıştır. Nodeların koordinatları, belirli noktalara yerleştirdiğimiz telefon ve bilgisayarların MAC adresleri aracılığı ile elde edildi. Bu noktaların her biri bir graph yapısında tutuldu. Bundan sonraki senaryo Dış Kısımın yazılımı ile aynıdır, detaylı olarak orada anlatılmaktadır.

#### **3.2.2.2 Dış Kısım**

Dış kısımda gidilebilecek bazı noktalar belirlenmiştir ayrıca bu noktalara komşu olan başka noktalarda yön bulmayı kolaylaştırmak adına eklenmiştir ve bu noktaların koordinatları google maps aracılığıyla alınmıştır. Bu noktaların her biri bir graph yapısında tutulmuştur. Ardından seçilen nokta ile bulunan nokta arasındaki en kısa yolu bulmak için dijkstra algoritması kullanılmıştır. Artık elimizde hedefe ulaşmak için hangi noktalardan geçileceği bilgisi vardı fakat konum bilgisi yönlendirmek için yetersizdi. Çünkü yönlendirmek mevcut yönümüzü ve bir sonraki noktaya ulaşabilmek için kaç derecelik açı ile döneceğimiz bilgisi lazımdı. Bunun için manyetik pusula kullanarak yönümüzü öğrendik ve bir sonraki noktanın koordinat bilgisini de kullanarak kaç derece döneceğimizi belirledik. Çıkan açının işaretine(negatif , pozitif) göre sağ veya sola dönülmesine karar verildi. Bu işlem yapılırken eğer açı 180 dereceden büyükse tümleyeni alınarak ters yönde tümleyeni kadar döndürülmüştür. Server kısmından gelen bilgi 3 tür olarak belirlenmiştir. Yapılan ilk kontrol kullanıcının baktığı yönün doğru olup olmadığı bilgisidir. Eğer kullanıcı doğru yöne bakıyorsa koordinat bilgileri ile belirlenen yoldaki gitmesi gereken ilk noktaya ulaşip ulaşılmadığı kontrolü yapılmıştır bu kontrol yapılırken belirli bir hata payına izin verilmiştir. Yapılan son kontrol ise hedef noktaya ulaşılıp ulaşılmadığı kontrolüdür. Bu bilgiler sürekli

kontrol edilerek hem arayüz modülüne hemde küpün çalışması için client'a bilgi gönderilmektedir.

## 4 PROJE ARA YÜZÜ ( INTERFACE )

### 4.1 Harita Modelleme

#### 4.1.1 Modellemede Kullanılan Programlar

- İç mekan modellemesi için:
  - Blender( 3d model tasarımı )
- Dış mekan modellemesi için :
  - Sketchup ( harita çizimi )
  - Blender ( haritayı 3d olarak görüntüleme )

#### 4.1.2 Modelleme

##### 4.1.2.1 Modelleme Dış Mekan

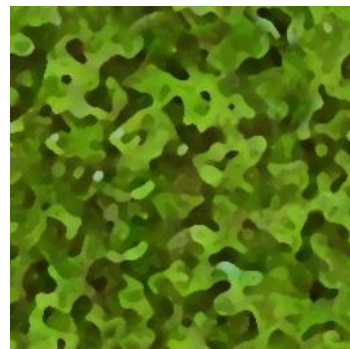
Dış mekanın modellemesi için öncelikle Sketchup'ta Bilgisayar binası, Elektronik binası, Kütüphane, Kimya ve Genetik binasını kapsayacak şekilde bir harita çizilmiştir. Programdaki bazı araçlar yardımıyla binalar yükseltip yollar belirginleştirilerek 3d bir görünüm kazanmıştır. Gerçekçi bir harita görüntüsü elde edebilmek amacıyla Sketcup'ın kendi texture'ları model için kullanılmıştır.



Çimen



Asfalt



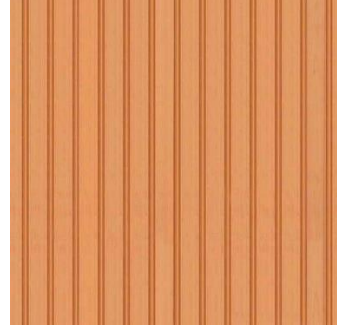
Ağaç



Dere



Köprü



Çardak

Haritada yollar için asfalt , zeminler için çimen , çardaklar için tahta , dere için su , köprü için çarpılı tahta desenleri kullanılmıştır. Ayrıca Bilgisayar bölümündeki otopark için beton deseni ve arabalar için renkli kareler kullanılmıştır.

#### 4.1.2.2 Modelleme İç Mekan

İç mekanın modellenmesi için “Blender” kullanılmıştır.Model bir plane objesi üzerinde tasarlanmıştır.Küp objelerinin scaling, rotation ve translation işlemlerinden geçirilmesi sonucunda zemin katın 3d görüntüsü oluşturulmuştur.Gerçeğe yakın bir görüntü elde etmek için programa dışarıdan 2d texture olarak tanımladığımız resimler eklenmiştir.



Asansör



Duyuru Panosu



Elektrik Panosu



Çift Kapılı Sınıflar



Tek Kapılı Sınıflar



Yangın Söndürücü



Kalorifer

Yukarıdaki resimleri de modelin üzerine yapıştırarak iç mekan için tasarladığımız 3d modelin son görüntüsü elde ettik.

## 4.2 Ara yüz Yazılımı (Interface Software)

### 4.2.1 Yazılımda Kullanılan Programlar

- Grafik ve animasyon için “Irrlicht” grafik kütüphanesi kullanılmıştır.
- Interface yazılımında kullanılan programlama dili C++ ‘dır.

### 4.2.2 Arayüz (Interface) Ana Ekran

- Program çalıştırıldığında dış mekan haritası ve kullanıcı için bir menü ekranı gelmektedir.





Açılış Ekranı

- Menü İçeriği:
  - ✓ Başlangıç ve bitiş konumları seçimi için iki adet liste bulunmaktadır.
  - ✓ Kullanıcı başlangıç ve bitiş konumlarını seçtikten sonra GO butonuna basarak seçimlerini servera iletir. Server bu konumları alarak bir path üretir. Path de yer alan konumlara ulaşıldığında ara yüz ekranında hangi konumda olduğu gösterilir.
  - ✓ Kullanıcı QUIT butonuna basarak programdan çıkabilir.

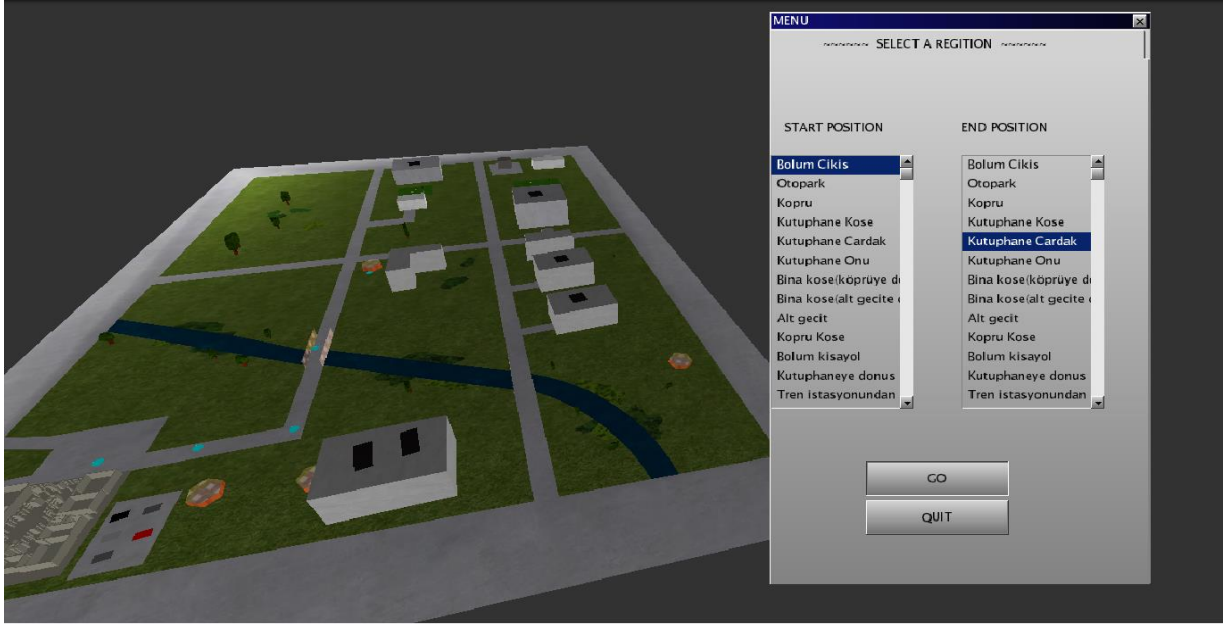
#### 4.2.3 Interface Dış Mekan

- Dış mekan seçiminde interface ekranı aşağıdaki gibi görünmektedir.



Dış Mekan Interface

- Kullanıcı başlangıç pozisyonu olarak “Bölüm Çıkış” ve bitiş pozisyonu olarak da “Kütüphane Çardak” seçerse hedef konuma ulaşıldığında interface ekranı aşağıdaki gibi olacaktır.



- Dış mekanda bulunduğu konumu temsilen küçük mavi-turkuaz noktalar kullanılmıştır.
- Başlangıçtan hedefe ulaşınca kadar var olan tüm noktalar (listede tanımlanmış) interfacede gösterilir.

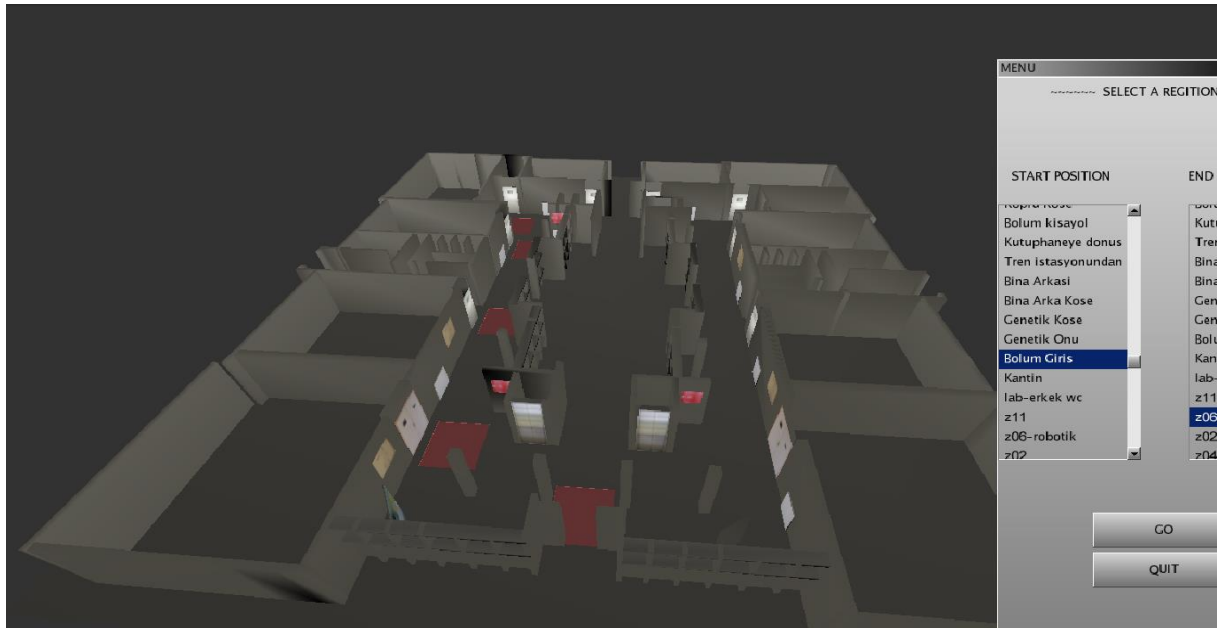
#### 4.2.4 Interface İç Mekan

- İç mekan seçildiğinde interface ekranı aşağıdaki gibi görünmektedir.



İç Mekan Interface

- Kullanıcı başlangıç pozisyonu olarak “Bölüm Giriş” ve bitiş pozisyonu olarak da “Z06”yı seçerse hedef konuma ulaşıldığında interface ekranı aşağıdaki gibi olacaktır.



## 5 PROJE SERVER'I

### 5.1 Shortest Path Sınıfı

En kısa yolu bulmak için dijkstra algoritması kullanılmıştır. Bu algorithmada graph için bazı nodelar tanımlanmıştır ve bulunan nokta ile hedef nokta arayüz tarafından alındıktan sonra kullanıcının gideceği yol belirlenmiştir. Alınan yol bilgisi ilk olarak bir listede tutulmaktadır. Bu listeden iteratör yardımı ile bir integer arraye alınmıştır. Bu array kullanılması üzere arayüze ardından stringe çevrilerek clienta yollanmıştır.

```
typedef int vertex_t;
```

```
typedef double weight_t;
```

```
const weight_t max_weight = std::numeric_limits<double>::infinity();
```

```
struct neighbor {
```

```
    vertex_t target;
```

```
    weight_t weight;
```

```
    neighbor(vertex_t arg_target, weight_t arg_weight)
```

```
        : target(arg_target), weight(arg_weight) { }    };
```

```

typedef std::vector<std::vector<neighbor> > adjacency_list_t;

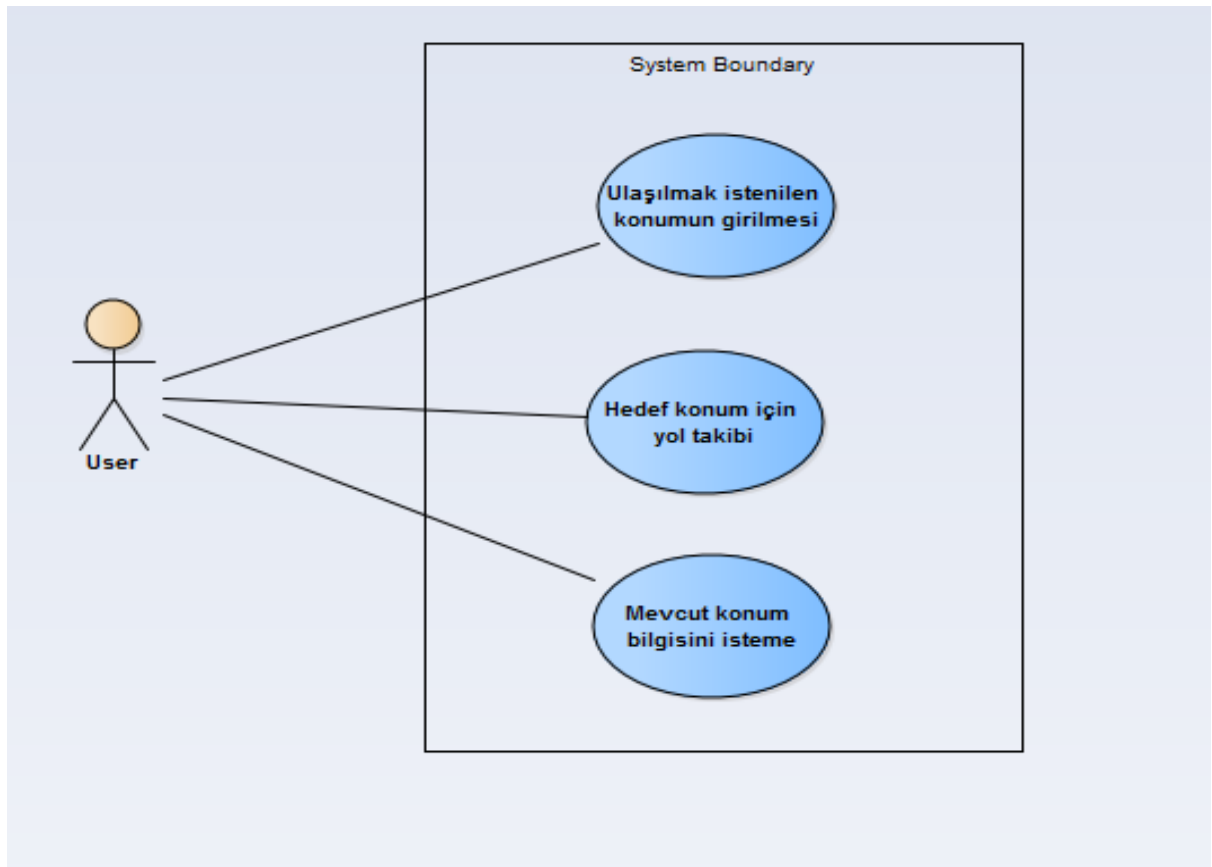
void DijkstraComputePaths(vertex_t source,
    const adjacency_list_t &adjacency_list,
    std::vector<weight_t> &min_distance,
    std::vector<vertex_t> &previous)

std::list<vertex_t> DijkstraGetShortestPathTo(
    vertex_t vertex, const std::vector<vertex_t> &previous)

```

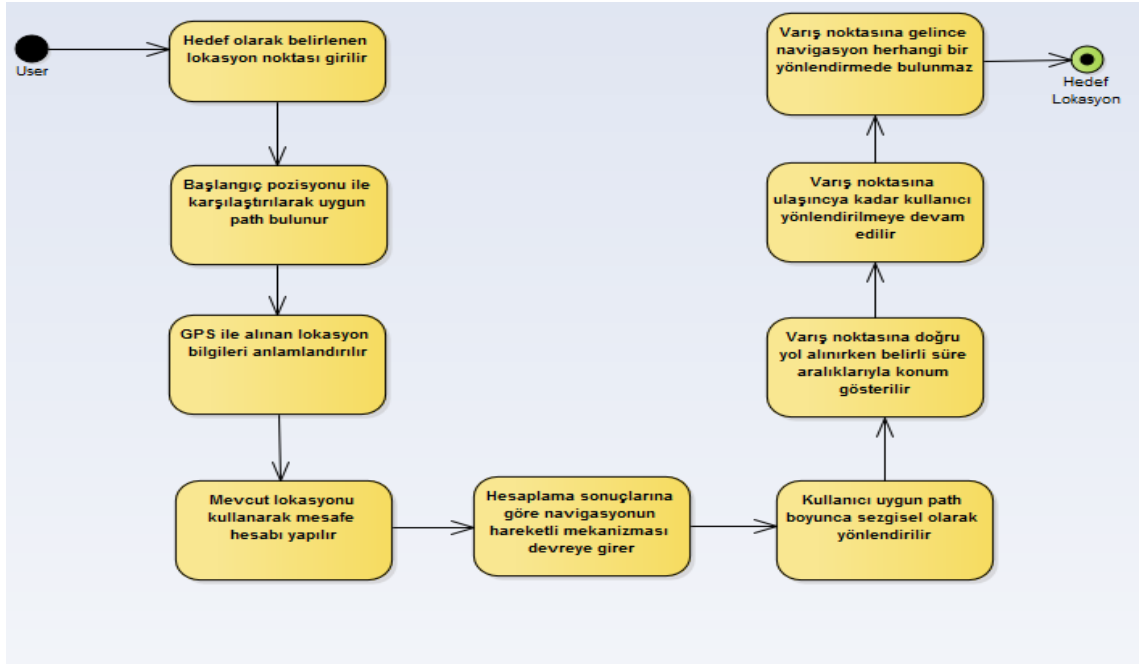
## 5.2 Use Case Diyagramı

### 5.2.1 Kullanım Durum Diyagramı



Proje kapsamında kullanıcının küp ile yapabilecekleri yukarıdaki diyagramda gösterilmiştir.

## 5.2.2 Etkinlik Diyagramı



Yukarıdaki diyagramda ise kullanıcı tarafından cihazın kullanımı esnasında, alt tarafta nasıl bir mekanizmanın çalıştığı gösterilmiştir.

## 6 SONUÇ

### 6.1 Gerçekleştirdiklerimiz

- ✓ Dijkstra algoritmasını kullanarak en kısa yol başarılı bir şekilde bulundu.
- ✓ İç ve dış mekan modellemesi ve simülasyonu ayrı ayrı başarılı bir şekilde gösterildi.
- ✓ Simülasyon üzerinde içeriden dışarıya ve dışarıdan içeriye geçiş başarıyla sağlandı.
- ✓ Donanımda kullanılan modüller ile server yazılımı ve modelleme yazılımı başarılı bir şekilde birleştirildi.

### 6.2 Gerçekleştiremediklerimiz

- ✗ Proje iç ve dış mekanda geçişli bir yapısı olacaktı. Ancak geliştirdiğimiz sistemde iç ve dış mekanlar arasında geçiş yapılamamaktadır. Özellikle iç

mekandan dışarıya geçişlerde yaşadığımız GPS modülünün geç aktif olması bu durumun oluşmasına sebep olmuştur.

- ✘ İç mekanda GSM/GPRS Shield kullanarak sim kartın alacağı IP üzerinden IP bazlı lokasyon bulmayı hedeflemiştik. Ancak sim kartın aldığı IP değerinin bu tarz bir lokasyon bulmaya uygun olmadığını öğrendik.

### 6.3 Çalışma Esnasında Oluşabilecek Aksilikler

- GPS modülü 5 metreye yakın sapma ve iç mekandan dışarıya geçişte konum almada yavaş çalışmaktadır.
- GSM/GPRS Shield'in elle çalıştırılması ve ısınma durumunda kendini kapatması.
- Pusulanın özellikle iç mekanlarda dış etmenlere bağlı olarak değerlerinde sapmalar oluşması.

## 7 KULLANIM KILAVUZU

Donanım: Öncelikle GSM/GPRS Shield in arkasındaki sim yuvasına sim kartımızı takıyoruz. Sonra Shieldin arkasındaki pinlerin isimlerine bakarak Android Mega ADK üzerinde denk gelecek şekilde takıyoruz. Bu işlemi düzgün yaptığımız takdirde MEGA ADK üzerindeki A6, A7 pinlerinin açıkta kaldığını göreceğiz. Daha sonra Shield üzerindeki dijital 7. pini 10.pine, 8.pine 11. pine taşıyoruz jumperla. Step motorların eklenmesi için ilk step motorun sürücü devresinden gelen 4 tane jumperı "22, 26, 24, 28" numaralı pinlere bağlıyoruz. Aynı şekilde ikinci motorun sürücü devresinden gelen pinleride jumper ile "38, 42, 40, 44" pinlerine bağlıyoruz. Arduino kit üzerinde sınırlı sayıda Vcc ve Gnd olduğu için 5V, 3.3V ve gnd çıkışlarını bir breadboarda aktarıp çoklayabiliriz. Bir diğer yöntemde ikinci bir arduino kullanarak Vcc ve Gnd bağlantılarını dağıtabiliriz. Motorların Vcc'lerini 5V'a , Gnd'lerininide Gnd'lere bağlıyoruz. Pusulamızın SCL ve SDA pinlerini MEGA ADK üzerindeki aynı isimdeki pinlere jumper ile bağlıyoruz. Vcc pinini 5V-3.3V'a Gnd'sini de yine Gnd ye bağlıyoruz. GPS'in RX,TX pinlerini MEGA ADK üzerinde digital 18 ve 19 numaralı pinlere jumper ile bağlıyoruz. Vcc'sini 3.3V'a , Gnd'sinide yine Gnd'ye bağlıyoruz. ESP8266'nın RX,TX pinlerini MEGA ADK üzerinde digital 14 ve 15 numaralı pinlere jumper ile bağlıyoruz. Vcc'sini 3.3V'a , Gnd'sinide yine Gnd'ye bağlıyoruz. CH\_PD pinini

jumper ile Vcc pinine bağlıyoruz. Bu işlemlerden sonra donanımımız hazır hâle geliyor. Arduinonun kendi IDE sini çalıştırıp kodumuzu açıyoruz. Araçlar kısmından kart sekmesini bularak Arduino Mega ADK'yı seçiyoruz. Yine araçlar kısmındaki port sekmesinden doğru portu seçili olup olmadığını kontrol ediyoruz. Kodumuzu derleyip, yüklemekten önce GSM/GPRS Shield üzerindeki güç tuşuna basıp Shield'ı çalıştırıyoruz. Ardından kodumuzu derleyip, yükleme işlemi yaptıktan sonra , cihaz kullanıma hazır hâle gelmiş oluyor.

#### Modelleme:

- Ubuntu için:

Öncelikle kendi çalıştığımız dosyayı indirmiş olduğumuz irrlicht klasörü içindeki examples' a taşıdık. Terminalden

```
$ cd irrlicht
```

```
$ cd examples
```

```
$ cd modelleme
```

```
$ make //burada kod compile edilir
```

```
$ cd ..
```

```
$ cd ..
```

```
$ cd bin
```

```
$ cd Linux
```

```
$ ./modelleme
```

Komutları girildiğinde programımız çalışmaya başlar ve ana ekranımız karşımıza gelir. Karşımıza gelen ara yüzde başlangıç ve bitiş noktalarının seçilebileceği bir menü ve dış mekanın haritası bulunmaktadır. Menüden başlangıç ve bitiş noktaları seçilir ve “GO” butonuna basılır. Seçilen noktalar server’a gönderilir. Server bir path döndürür. Donanım hareket ettikçe belirlenmiş nodeların üzerinden geçilirse, node’larda renklenme meydana gelir.

- Windows için :

-IDE olarak Visula Stduio kullandık.

-Öncelikle irrlicht librarylerinin bulunduğu dosya indirilir.

- Visual Studio’da c++ proje dosyası oluşturulur.

- Proje dosyasının üzerine sağ tıklayarak “Properties” seçeneğine tıklanır.



- Burada hem release hem de debug mod için <Include Directory> ve <Lib Directory> içine irrlichtin include ve lib dosyaları include edilir.
- Son olarak irrlicht dosyasındaki “irrlicht.dll” kopyalanarak proje dosyasında yer alan release ve debug dosyalarının içerisine kopyalanır.

#### Kodu Derleme ve Çalıştırma

- Proje dosyasına sağ tıklanarak “Build” seçeneği seçilir ve kod compile edilir.
- Build edildikten sonra yine proje dosyasına sağ tıklayarak “Run” seçilir ve program çalıştırılır.

## **8 KAYNAKLAR**

- ❖ Arduinio Documentation, <http://arduino.cc/en/Tutorial/HomePage>
- ❖ GPRS SHIELD [http://www.seeedstudio.com/wiki/GPRS\\_Shield\\_V3.0](http://www.seeedstudio.com/wiki/GPRS_Shield_V3.0)
- ❖ ESP8266 <http://www.esp8266.com/>
- ❖ <http://irrlicht.sourceforge.net/docu/example005.html>
- ❖ <http://irrlicht.sourceforge.net/docu/example012.html>