

# **KENDİ KENDİNE YOL BULAN ARAÇ**

**Projeyi Yapan : Selim Göksu**

**Proje Yöneticisi : Prof. Dr. Tülay Yıldırım**

## **GİRİŞ**

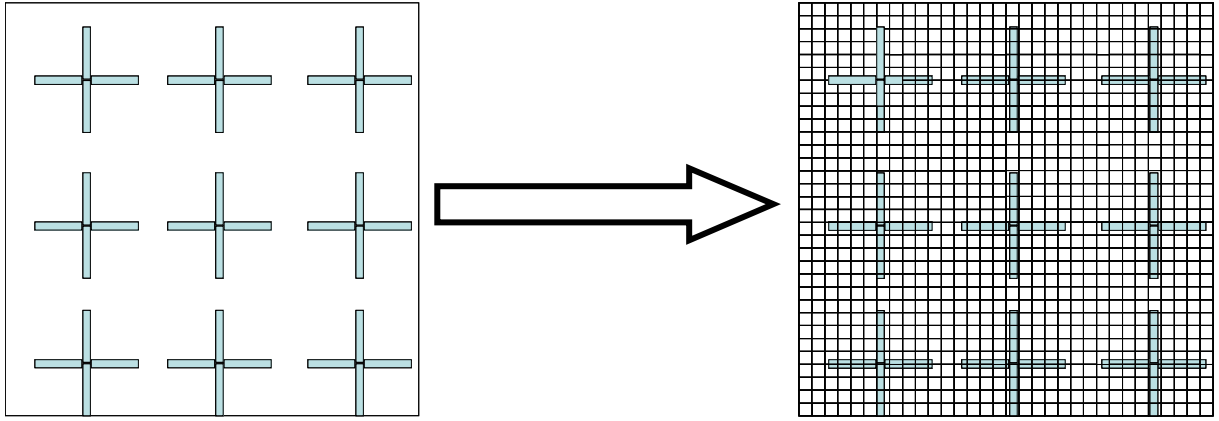
Günümüzde, kullanılan bir takım araçların (evdeki robotlardan fabrikalardaki forkliftlere, sokaktaki arabalardan askeri alanlarda kullanılan diğer araçlara), tehlike de içerebilen idare ve denetimlerini uzaktan yapılabilir hale getirerek yada bu araçları tamamen kendi kendilerine iş yapabilir kılarak, insanların üzerine düşen idare ve denetim yükünün ve de tüm bunların getirdiği tehlikelerin azaltılması için yapılan çalışmalar teknoloji dünyasının başlıca uğraşları arasında yer almaktadır.

Kendi Kendine Yol Bulan Araç projesi ile, herhangi bir harita üzerinde, bir noktadan diğerine, engelleri de göz önünde bulundurarak en kısa yoldan gidebilen araç, yazılım ve bir model araç yardımıyla donanım olarak gerçekleştirilmiştir. Amaç, araçların hareketi söz konusu olduğunda, insanın üzerine düşen kontrol yükünün hafifletilmesidir.

Bitirme projesi adı altında yapılan bu çalışma ile, önce herhangi bir düzlem, boyutları belirli hücreler ile kaplanarak, o düzlemi temsil edebilecek bir harita meydana getirilmiş, bu harita üzerinde herhangi bir konumdan diğerine giden en kısa yol bulma algoritması geliştirilmiştir. Çalışırken daima haritanın neresinde olduğunu bilen araç, gitmesi istenen hedef bilgisini aldıktan sonra belleğinde bulunan harita üzerinde o hedefe varan en kısa yolu bulur ve o yol üzerinden harekete başlar. Araç, hareketi sırasında bulundurduğu engel algılayıcılar yardımı ile çevresindeki engellere çarpmadan hedefe varır. Hareket halinde, engel algılayıcılardan elde edilen bilgiler bellekteki harita ile çelişirse, algılayıcılardan elde edilen bilgiye göre bellekteki harita değiştirilerek, yeniden o konumdan hedef noktaya giden en kısa yol hesaplanır ve bu yol üzerinden harekete devam edilir.

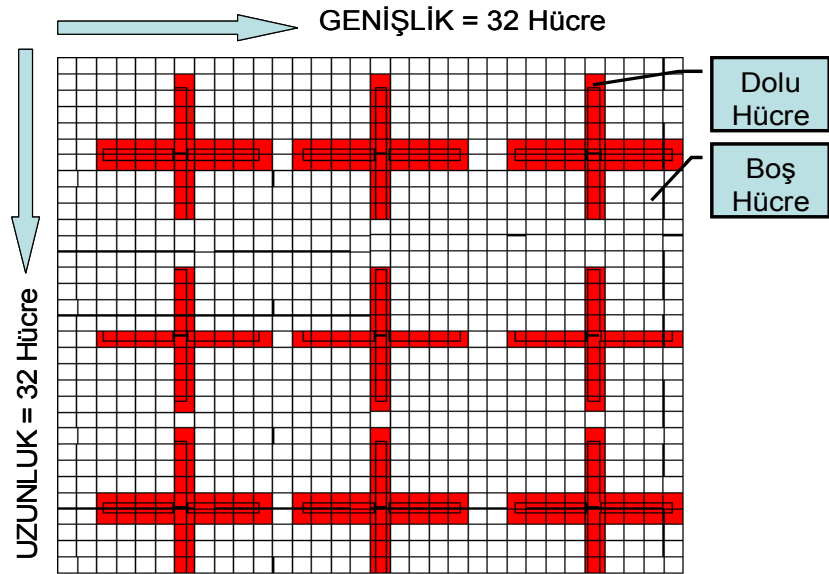
## HARİTA OLUŞTURMA

Aşağıdaki şekillerden, bu projede çalışılan haritanın nasıl oluşturulduğu görülebilmektedir. Şekil 1 ‘in sol tarafında, mavi renkli engellere sahip bir düzlem görülmektedir. Bu, bir işyerindeki çalışma ortamının üstten görünüşü olarak düşünülebilir. Düzlem, boyutları belirli karelere (hücre) bölünürse, Şekil 1 ‘in sağ tarafındaki gibi bir görüntü oluşur. Düzlemsel alan ne kadar çok hücreden oluşursa aracın hareket kabiliyeti de o kadar hassas olmaktadır.



Şekil 1 Herhangi Bir Alanın Hücrelere Bölünmesi

Engellere denk gelen hücreler kırmızı ile gösterilirse Şekil 2 ‘deki gibi sayısal bir harita ortaya çıkar ve bu da projede kullanılan haritadır.

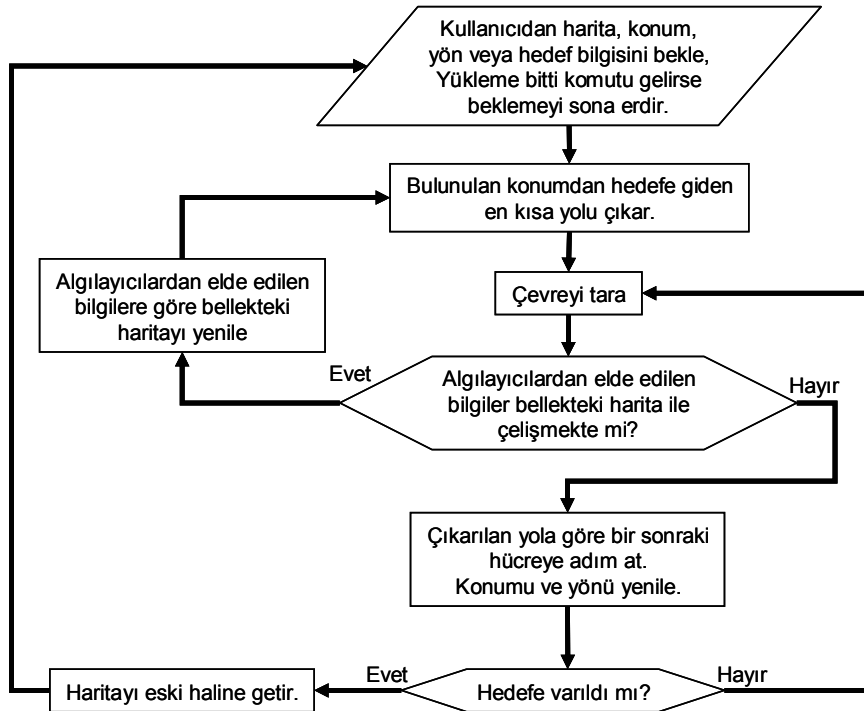


Şekil 2 Oluşturulan Harita

Bu şekildeki bir harita araç belleğinde  $32 * 32 = 1024$  elemanlı bir dizi şeklinde tutulmaktadır. Hücrelerin adresi dizideki indislere, içerikleri ise dizideki değerlerine denk gelmektedir. Genişlik ve uzunluk büyüklükleri haritanın boyutunu belirler. Araç boş hücreler üzerinden adım adım ilerleyerek hedefe varmaktadır.

## HAREKET

Araç harekete başlamadan önce kullanıcıdan, gezindiği haritanın şekli, sahip olduğu yön bilgisi, bulunduğu konum hücre bilgisi ve gideceği hedef hücre bilgisi gibi bir takım komut veya veriler bekler. Kullanıcının yüklemeyi bitirdim anlamındaki komutuyla birlikte araç bulunduğu konumdan hedef giden en kısa yolu çıkarır. Sonra bulundurduğu sensörler aracılığıyla çevresindeki engelleri tarar. Eğer sensörlerden elde ettiği engel bilgileri belleğindeki harita ile çelişirse belleğindeki haritada geçici olmak üzere yenileme yaparak tekrar bulunduğu konumdan hedefe yeni bir yol çıkarır, çelişki olmadığında ise daha önce çıkardığı yola göre sıradaki hücreye adım atarak, belleğinde konum ve yön bilgilerini yeniler. Her adımda hedefe varıp varmadığını kontrol eden araç, hedefe vardığında belleğindeki harita üzerinde yapmış olduğu geçici değişiklikleri iptal ederek haritayı eski haline getirir. Aracın hareket algoritması aşağıdaki gibidir.

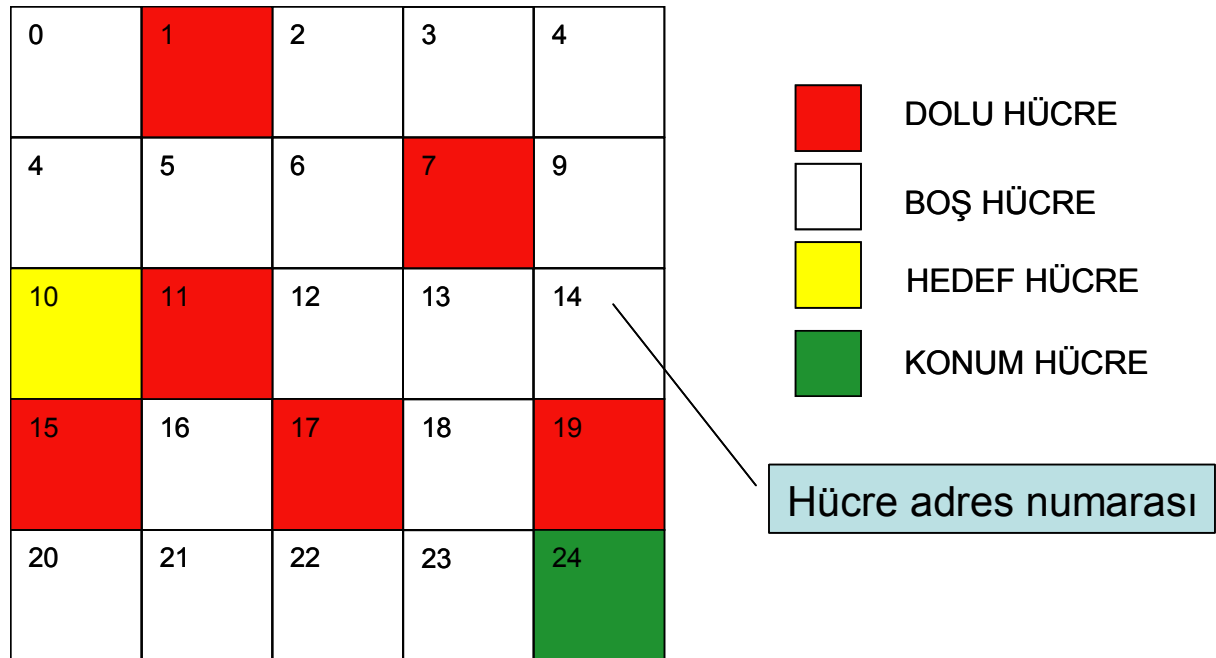


Şekil 3 Hareket Algoritması

## EN KISA YOL BULMA

En kısa yol bulma problemlerinin en bilinen çözümleri Dijkstra, Bellman-Ford ve Floyd algoritmalarıdır. Bu algoritmalar graf veri modelinde yer alan komşuluk veya kenar matrisleri üzerinden çözüme varan algoritmalarlardır. Bu projede en kısa yolu bulmak için farklı bir yöntem denenmiş, haritadaki her bir hücrenin taranması yolu ile, harita üzerinde konum ile hedef arasında adımlar oluşturularak en kısa yol çıkarılmıştır.

Aşağıdaki örnek harita 25 hücreden oluşmaktadır. Böyle bir harita üzerinde yol bulma şu şekilde gerçekleştirilmektedir:

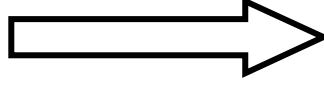


Şekil 4 Örnek Harita

1. Aracın belleğinde kullanıcı tarafından gönderilen harita yapısı mevcuttur. Araç konumunun ve gideceği hedefin adresini bilmektedir. (KONUM = 24, HEDEF = 10)
2. Hedef hücre merkez alınarak, konum hücreye ulaşılan kadar haritaya adımlar yerleştirilir (Şekil 5).
3. Konum hücreden itibaren, hedefe varılana kadar adım numarası bir eksik hücreye adım atılarak hedefe en kısa yoldan varılır (Şekil 6 Mavi Renk).

|    |    |    |    |    |
|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  |
| 2  |    | 4  | 5  | 6  |
| 4  | 5  | 6  | 7  | 9  |
| 1  | 2  | 3  |    | 7  |
| 10 | 11 | 12 | 13 | 14 |
| 0  |    | 4  | 5  | 6  |
| 15 | 16 | 17 | 18 | 19 |
|    |    | 6  |    |    |
| 20 | 21 | 22 | 23 | 24 |
|    |    | 8  | 7  | 8  |

Şekil 5 En Kısa Yol Bulma-1



|    |    |    |    |    |
|----|----|----|----|----|
| 0  | 1  | 2  | 3  | 4  |
| 2  |    | 4  | 5  | 6  |
| 4  | 5  | 6  | 7  | 9  |
| 1  | 2  | 3  |    | 7  |
| 10 | 11 | 12 | 13 | 14 |
| 0  |    | 4  | 5  | 6  |
| 15 | 16 | 17 | 18 | 19 |
|    |    | 6  |    |    |
| 20 | 21 | 22 | 23 | 24 |
|    |    | 8  | 7  | 8  |

Şekil 6 En Kısa Yol Bulma-2

## KULLANICI ARAYÜZÜ

Kullanıcının araca harita bilgisini, konum, yön ve hedef bilgilerini, bilgisayar ekranında görsel olarak takip edip araca gönderebilmesi için visual basic kullanılarak bir arayüz ekranı oluşturulmuştur.

KULLANICI PROGRAMI

HEDEF GİRİNİZ

0 1 2 3 4

HEDEF GÖNDER

KONUM YENİLE

5 6 7 8 9

KONUM GÖNDER

HARİTA YENİLE

10 11 12 13 14

HARİTA GÖNDER

20 21 22 23 24

HARİTA TUŞ TAKIMI

YÖN YENİLE

YÖN GÖNDER

DOĞU (0)

KUZAY (1)

BATİ (2)

GÜNEY (3)

YÖN TUŞLARI

YÜKLEME BASLAT

YOL BELİRLE

YÜKLEME BİTİR

KUZAY

BATİ

DOĞU

GÜNEY

GİRİŞ

ÇIKIŞ

ARACIN GİTMESİNİ İSTEDİĞİNİZ HEDEF HÜCREYİ

HEDEF SEÇİLDİ ----- HEDEF HÜCRE :1 ---

HEDEFİ ARACA GÖNDERMEK İÇİN - HEDEF GÖNDER - BUTONUNA TIKLAYINIZ

Y.T.Ü. ELEKTRİK ELEKTRONİK FAKÜLTESİ

ELEKTRONİK ve HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ

BİTİRME PROJESİ

KENDİ KENDİNE YOL BULAN ARAÇ ARAYÜZ PROGRAMI

SELİM GÖKSU

DOLU HÜCRE

HEDEF HÜCRE

BOŞ HÜCRE

KONUM HÜCRE

EN KISA YOL

GEÇİCİ DOLU HÜCRE

B

0

B

B

B

2

1

D

B

B

3

D

D

B

B

4

D

B

D

B

5

6

7

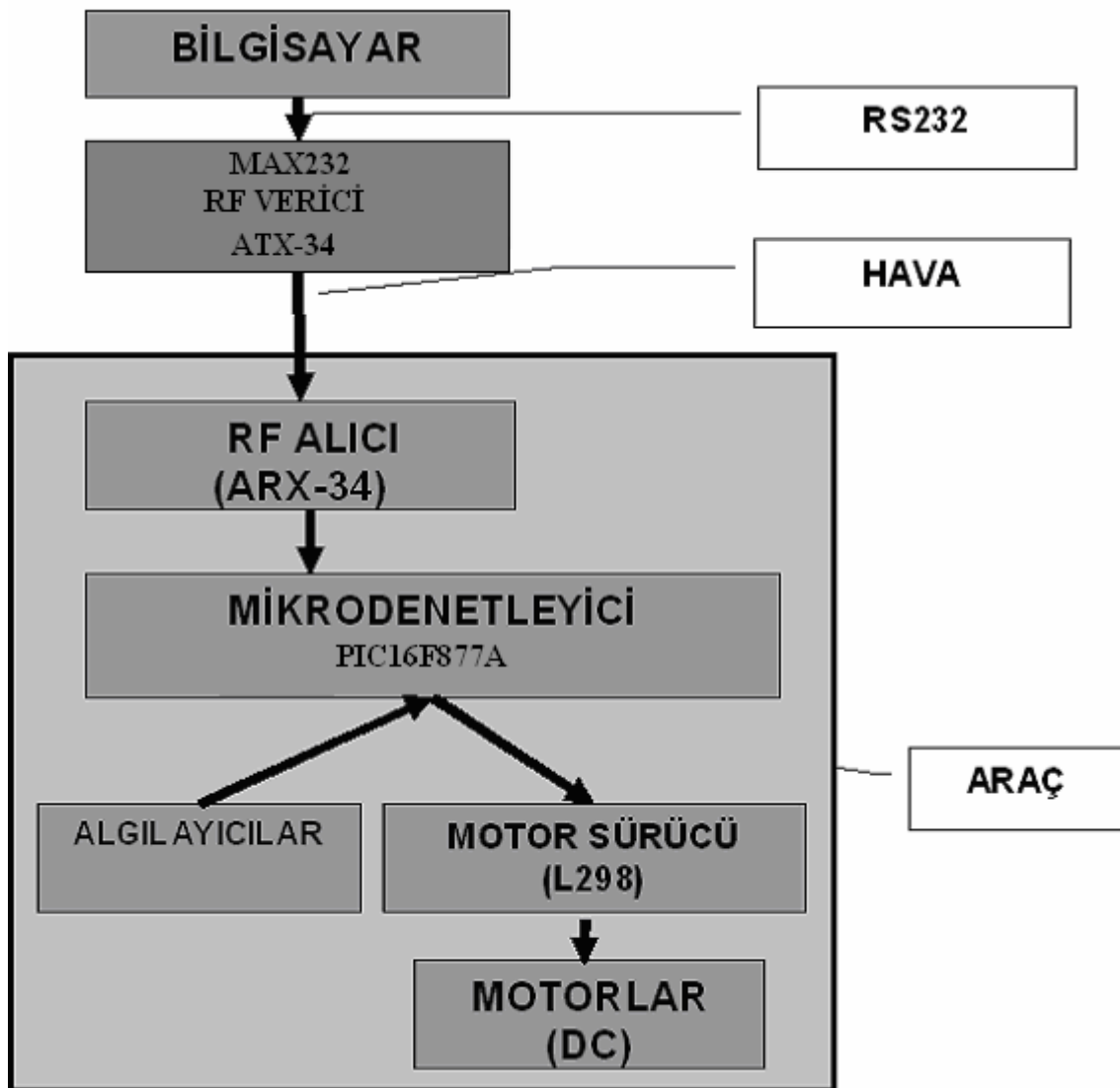
B

B

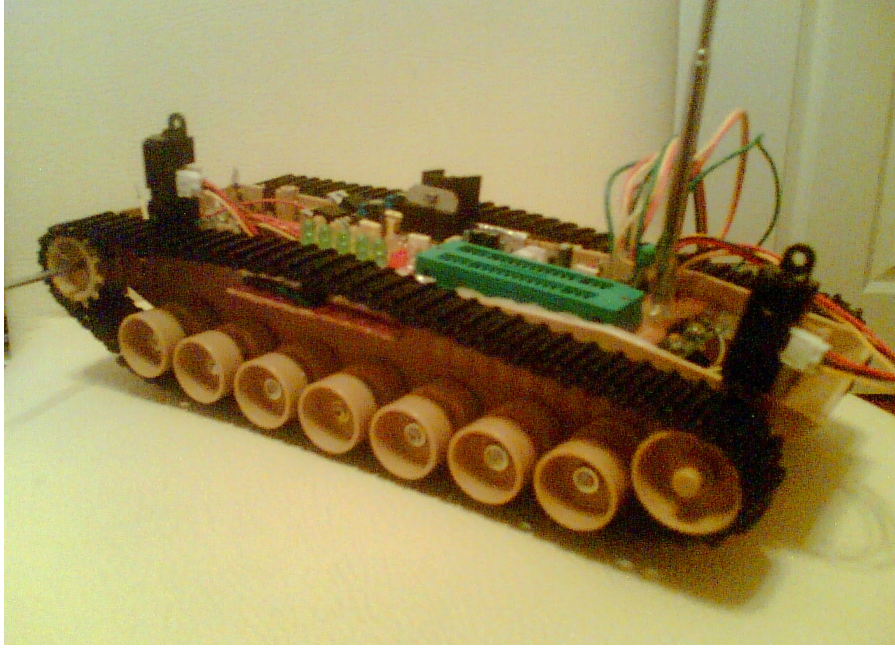
Şekil 7 Kullanıcı Arayüz Ekranı

## DONANIM

Kullanıcı tarafında, bilgisayarın RS232 portu çıkışına, MAX232 tümdevresi ile bağlanan ATX-34 RF verici birimi, bilgisayardan gelen verileri modüle ederek ortama yayar. Araç tarafında ise ARX-34 RF alıcı birim ile alınan veriler demodüle edilerek, mikrodenetleyici olarak kullanılan PIC16F877A 'nın Usart girişine gönderilir. Mikrodenetleyici aldığı verilere göre hesaplamalar yaptıktan sonra DC motor sürücü tümdevre, L298 aracılığı ile hareketi başlatır ve yönetir. Aynı zamanda aracın çevresine yerleştirilen SHARP 2D120X sensörleri, aracın etrafındaki engel bilgisini mikrodenetleyiciye iletir.



Şekil 8 Donanım



**Şekil 9 Tank Benzeri Model Araç**

## **YAZILIM**

Proje gerçekleştirilirken iki farklı yazılım dilinden yararlanılmıştır. Kullanıcı tarafında arayüz programı oluşturulurken, visual basic kullanılmıştır. Mikrodenetleyiciyi çalıştırmak için de MikroC dilinden yararlanılmıştır. Öğrenim ve kullanım kolaylığı, PIC16F877A ile uyumlu geniş bir kütüphaneye sahip olması MikroC ‘nin tercih edilme nedenidir.

## **SONUÇLAR**

Sonuç olarak, bir takım kısıtlamalarla da olsa boyutları bilinen bir harita üzerinde noktalar arası kendi kendine gezinebilen robot araç tasarlanmıştır. Bundan sonraki aşamalarda aracın çevresindeki engel bilgisini daha detaylı algılaması için algılayıcılar artırılabilir veya değiştirilebilir. RF iletişim çift yönlü yapılarak aracın her adımda çevresindeki engel bilgisini kullanıcıya göndermesi sağlanabilir. Aracın gezindiği harita bilgisi, haritaya hakim bir kamera yardımıyla belirli anlarda fotoğraflanıp, görüntü işleme teknikleri ile bu projede ele alınan hücresel harita görünümü ve bilgisi elde edilebilir. Harita karesel değil de altıgen hücrelerle kaplanabilir böylece bir hücreden daha fazla hücreye bir adımda geçiş olanağı sağlanabilir. Araçların hareket matematiği elde edilip projeye yüklendikten sonra herhangi bir araç için kullanılabilir halde proje genişletilebilir.