

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



Kamera Kontrollü Otonom Yol Takip Eden Robot

TASARIM PROJESİ

**Adem KÖSE
Mahmut KARALI**

2015-2016 GÜZ DÖNEMİ

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

Kamera Kontrollü Otonom Yol Takip Eden Robot

TASARIM PROJESİ

**Adem KÖSE
Mahmut KARALI**

Bu projenin teslim edilmesi ve sunulması tarafımda uygundur.

Danışman :Prof. Dr. Cemal KÖSE

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

“Kamera Kontrollü Otonom Yol Takip Eden Robot“ konulu bu çalışma Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü’nde Tasarım Projesi Çalışması olarak ele alınmıştır.

Çalışmamızda popüler mikrodenetleyici Arduino ile bir robot tasarlanmış ve mobil işletim sistemi olarak Android kullanılmıştır. Mobil kamera olarak Android bir akıllı telefonun kamerası kullanılmıştır.

Eğitim ve öğretim hayatımız boyunca bize destek veren ailelerimize teşekkür ederiz. Çalışmalarımız boyunca değerli zamanını ayırıp, verdiği bilgileri ile bizi yönlendiren hocamız Prof. Dr. Cemal KÖSE’ye teşekkür ederiz.

Adem KÖSE
Mahmut KARALI

Trabzon 2015

İÇİNDEKİLER

	Sayfa No
IEEE ETİK KURALLARI	II
ÖNSÖZ	III
İÇİNDEKİLER	IV
ÖZET	V
1. GENEL BİLGİLER	1
1.1 Giriş	1
2. SİSTEM DONANIMI	2
2.1 Arduino Uno	2
2.1.1 Arduino Uno Board Özellikleri	2
2.1.2 Güç	3
2.1.3 Giriş ve Çıkış	4
2.1.4 Haberleşme	5
2.2 Motor Devresi	6
2.2.1 DC Motor	6
2.2.2 Sürücü Devre	6
2.3 Arduino Wifi Shield	8
2.4 Programlama	8
2.4.1 Client-Server	8
2.4.2 Robot Hareketleri	9
3. SİSTEM YAZILIMI	10
3.1 Android Ortamı ve Programlama Temelleri	10
3.2 Android Client-Server	11
3.2.1 Basit Bir Veri Transferi	11
3.2.2 Komutların Üretilmesi ve Komut Gönderimi	12
3.3 Android OpenCV	12
3.3.1 OpenCV'ye Giriş	12
3.3.2 Color Detection Hakkında Bilgi	13
3.3.3 Robotun Tespit Edilmesi	15
3.3.4 Yolun Tespit Edilmesi ve Rota Oluşturulması	16
3.3.5 Karar Mekanizması ve Kullanılan Matematiksel Yöntemler	18
4. SONUÇLAR	21
5. ÖNERİLER	22
6. KAYNAKLAR	23
STANDARTLAR ve KISITLAR FORMU	24

ÖZET

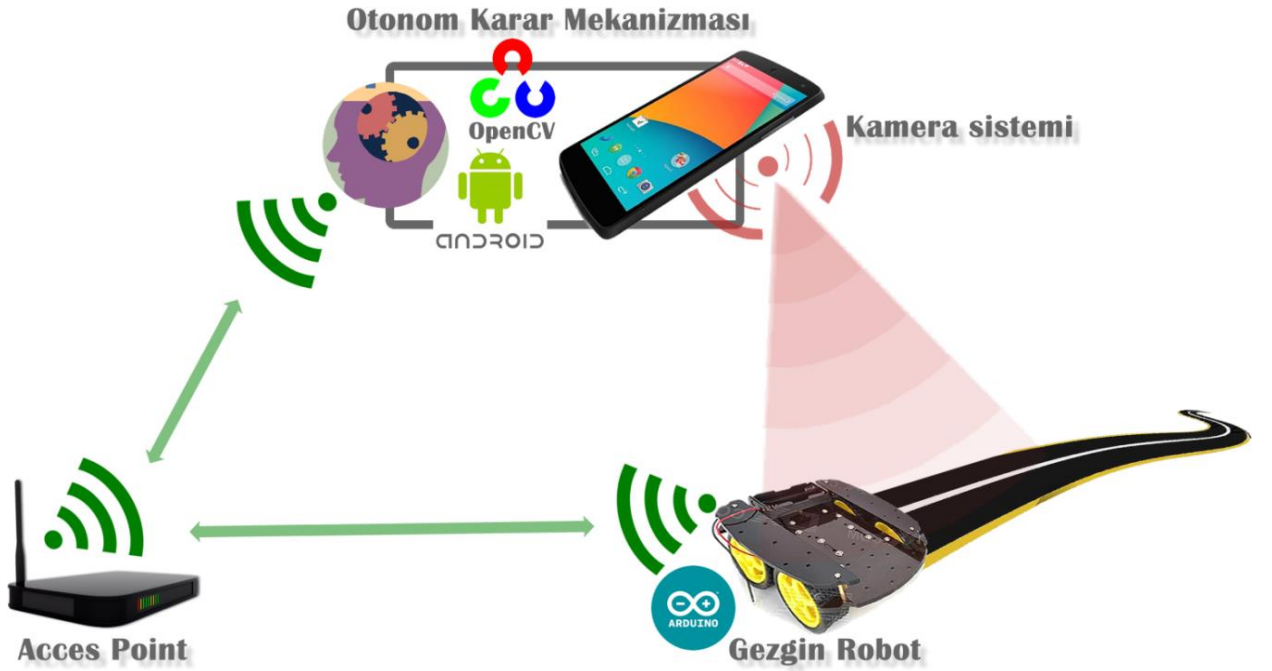
Otonom robot sistemleri alanında arařtırmalar her geen gn artmaktadır. Arařtırmalar robotların,askeri, geliřmiř retim sistemleri, saėlık sektr, dıř ortamlarda otonom aralar, uzay keřifleri, maden arama, tarım, su altı ve kurtarma sistemleri gibi geniř bir alanda uygulamalarda kullanılmaktadır. Birok alanda uygulama ortamı bulunan otonom robot sistemlerinde arařtırmacı olmak ve rn geliřtirmek cazip bir olaydır.Bu alanda bilgi sahibi olmak ve rn geliřtirmek iin bu alıřmayı yapmıř bulunmaktayız.

1. GENEL BİLGİLER

1.1. Giriş

Çalışmamızda kamera kontrollü otonom bir şekilde (insanz) yol takip eden bir robot tasarladık. Robot ile kamera arasında kablosuz haberleşme mevcut olduğu için bunun sağlamış olduğu bir avantajdan şöyle söz edebiliriz; bu avantaj robotun hareket alanını oldukça geniş tutması. Tepeden bir kamera ile izlenen arazi robot ,istenilen yolu takip edebilmesi için karar mekanizmasının oluşturulması, varolması gerekmektedir. Bu gereksinimden dolayı karar mekanizmasını robotun ve yolun durumlarına göre eş zamanlı aldığı görüntüleri işleyen ve robota komut üretebilen ufak bir yapay zekaya sahip karar verme mekanizması gerçekleştirdik. Örneğin veri toplamak için araziye gönderilen bir keşif robotu, önceden veya anlık oluşturulan bir yolun takip edilmesi istenildiğinde, o yolu başarılı bir şekilde takip edebilecek bir sistem tasarlamış olduk. Sistemin karar mekanizmasının karar vermesi ve ardından robota komutların gönderilmesi olayları eş zamanlı olduğu için bu çalışma bir çok sektör ve amaç için kullanılabilir,her uygulamaya kolayca adapte edilebilir. Projemizde en popüler ve en işlevsel bileşenleri kullandığımız için projemizin günlük teknolojilere uyumluluğu da çok yüksektir.

Projemizde kullandığımız bileşenler;



Android Akıllı Telefon ;

Karar mekanizmasını ve kamera sistemini içinde bulunduran sistem

Arduino ;

Robot beyni, gelen komutları işleyen, robotun mekanik hareketlerini üreten bileşen.

Arduino Wifi Shield ;

Karar mekanizmasından gönderilen komutu, kablosuz bir şekilde almaya yaran robot bileşeni.

Sürücü Devre;

Robotun teker motorlarını süren elektronik dc motor sürücü devresi bileşeni.

Access Point;

Robot ile karar mekanizmasının veri alışveriş yapacağı ortam.

2. SİSTEM DONANIMI

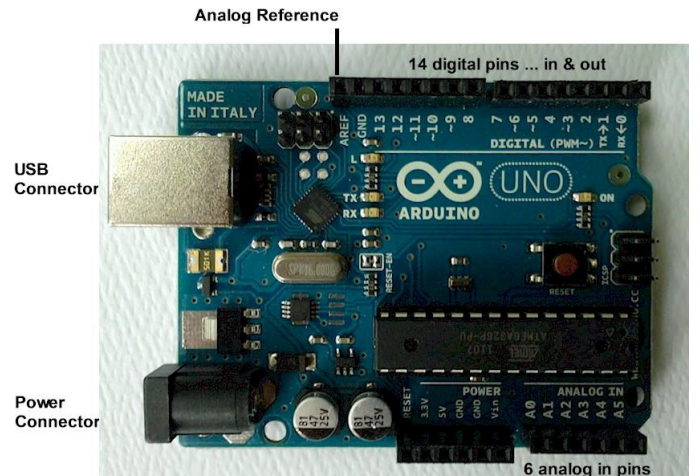
2.1 Arduino Uno

Arduino , açık kaynak kodlu geliştirilmiş , yazılım ve donanım tabanlı bir elektronik prototiptir. Arduino mikrodenetleyici programlamak ve bu mikrodenetleyiciyi giriş-çıkış pinleri sayesinde başka fiziksel cihazlarla haberleşmesini sağlamak için üretilmiş platformdur. Arduino ATMEL firmasının ürettiği ATmega serisi mikrodenetleyiciler kullanılmaktadır. Arduino ailesinin son versiyonu Arduino UNO'nun bir önceki versiyonundan farkı FTDI(future technology devices international) çipi yerine ATmega8U2 çipini kullanmasıdır. Bu çip daha hızlı transfer geçişine ve Linux ve Mac işletim sistemlerinde sürücüye ihtiyaç tanımadan direk tanınmasını sağlar. Bizde Arduino Uno kullanarak projemizi gerçekleştirdik.

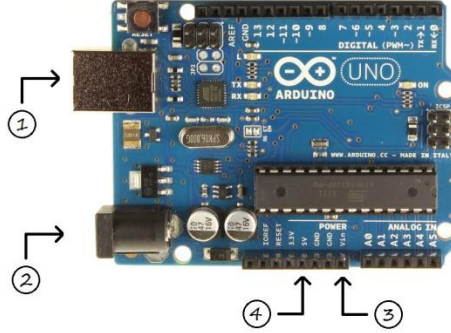
2.1.1 Arduino Uno Board Özellikleri

Arduino Uno ATmega328 tabanlı bir mikroişlemci kartıdır. Bu mikroişlemci kartında 14 dijital giriş / çıkış işareti (6 tanesi PWM çıkışları olarak kullanılabilir olan), 6 analog giriş, 16 MHz seramik rezonatör(belli bir frekansta titreşen ya da rezonans yapan, dolayısıyla belirli frekanslardaki dalgaları iletmeye ya da güçlendirmeye yarayan mekanik veya elektriksel sensör.), bir USB bağlantısı, bir güç girişi, bir ICSP başlık ve bir reset düğmesi vardır.

1. Çalışma gerilimi 5 Volt,
2. Besleme gerilim sınırı 6-20 Volt olup tavsiye edilen ara 7-12 Volt'tur,
3. Dijital giriş/çıkış pinleri 14 pin,
4. Analog giriş 6 pin,
5. Pin başına 40 mA,
6. 32 KB Flash,
7. 2 KB SRAM,
8. 1 KB EEPROM,
9. 16 MHz saat frekansı



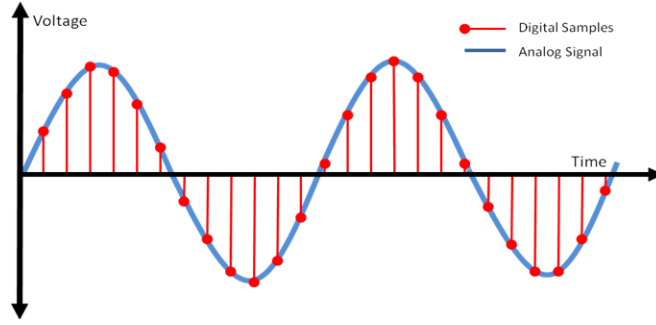
2.1.2 Güç



- Arduino'nun USB ile beslenmesi : Arduino'nun USB kablosunu bilgisayara bağladığımızda, Arduino çalışması için gerekli enerjiyi bilgisayardan almaktadır. Bu giriş, Arduino için gerekli enerjiyi sağlarken, aynı zamanda Arduino'nun bilgisayarla haberleşmesini, Arduino'ya yeni kod yüklenmesini de sağlar. Yukarıdaki görselde 1 numaralı giriş Arduino'nun USB girişidir. Görselde de görüldüğü gibi bu giriş, yazıcı kablosu olarak tarif edilen USB B girişidir. USB standartlarına uygun olarak tasarlanan bu girişe en fazla 5 Volt gerilim uygulanmalıdır. Eğer bu girişe 5 Volt üzeri bir gerilim uygulanırsa, Arduino zarar görebilir.
- Arduino'nun pille çalıştırılması : Arduino harici besleme kaynaklarıyla da çalıştırılabilir. Bunun için Arduino üzerinde birbirine bağlı iki farklı giriş bulunmaktadır. Bu girişlerden ilki, yukarıdaki görselde 2 numara ile gösterilen jack girişidir. Bu girişe 7 ile 12 Volt (önerilen) arasındaki gerilimler uygulanabilir. Bu girişe bağlı regülatör (gerilim düzenleyicisi) ile girişe uygulanan gerilim, Arduino'nun çalışma gerilimine düşürülür. Arduino üzerinde bulunan 'Vin' pini, Arduino'nun jack girişine bağlı bir pindir. Bu pine uygulanan gerilim, Arduino'ya ulaşmadan önce bu pine bağlı regülatör yardımıyla Arduino için uygun gerilime düşürülür. 'Vin' girişine 7 ile 12 Volt arasındaki gerilimler uygulanmalıdır. Pilin artı (+) ucu 'Vin' pinine bağlandıktan sonra, pilin eksi (-) ucu Arduino'nun 'GND' yani toprak ucuna bağlanmalıdır. 'Vin' pini yukarıdaki görselde 3 numara ile gösterilmiştir. Eğer bu girişlere uygulanması gereken gerilimden fazla bir gerilim uygulanırsa, Arduino zarar görebilir.
- Arduino'nun 5 Volt pininden beslenmesi : Arduino üzerinde bulunan 5 Volt pini de Arduino'yu beslemek için kullanılabilir. Arduino yaygın olarak bu pinden beslenmese bile, buraya 5 Volt gerilim uygulandığında, Arduino'nun çalıştığı görülmektedir. Bu pine 5 Volt geriliminden fazla bir gerilim uygulanması, Arduino'nun bozulmasına neden olacaktır. 5 Volt gerilim veren bir kaynak ile kaynağın artı (+) ucunu Arduino'nun 5 Volt, eksi (-) ucunu da Arduino'nun 'GND' yani toprak pinine bağlayarak kullanabiliriz. Bu pin yukarıdaki görselde 4 numara olarak gösterilmiştir.

2.1.3 Giriş Çıkış

Mikrokontrolcüler lojik 0 ve lojik 1 üzerine çalışan dijital cihazlar olsada analog sistem ve devrelerde yaygın kullanıma sahiptirler. Mikrokontrol örnekleme ile analog giriş voltajıda okuyabilirler ve onu kendi ADC (analog digital converter) ile digital değerlere çevirebilirler. Mikrokontrolcülerin birçoğunda dahili PWM modül ve ADC modül bulunması sayesinde analog voltaj girişi ve analog voltaj çıkışı sağlanabiliyor.



Arduino UNO kartımızdaki işlemcide, 10-bit çözünürlüğe sahip analogdan dijittale dönüştürücü (ADC – analog to digital converter) mevcuttur. Bu mikrokontrolcüde 10-bit ADC, 0V ile 5V arası gerilimleri $2^{10} = 1024$ adım hassasiyet ile okuyabilir. Yani analog input pinlerinden birine vereceğimiz 0V gerilim bize 0 değerini; aynı şekilde 5V gerilim ise 1023 değerine denk düşüyor

Eğer okuduğumuz analog degerin kaç volt olduğunu öğrenmek istersek :
(Deger * 5V)/1023 ile buluruz.

Arduino kartı üzerinde bulunan pinleri çıkış veya giriş olarak tanımlayabilirsiniz. Programın en başında (setup içerisinde), kullanılacak pinler ayarlanmalıdır. Bunun için aşağıdaki kod kullanılır:

```
pinMode(13,OUTPUT);
```

Böylece 13. Pin çıkış olarak tanımlanır. Giriş olarak tanımlanması için INPUT, çıkış için OUTPUT yazılması gerekir.

DigitalWrite : Çıkış olarak tanımlanmış pinlerden enerji vermek için veya enerji vermeyi durdurmak için kullanılır. Örneğin:

```
digitalWrite(13,HIGH);
```

Böylece, daha önce çıkış olarak tanımlanmış 13 numaralı pinden enerji çıkışı sağlanır. Enerji çıkışı durdurulmak istenildiğinde HIGH yerine LOW yazılması yeterlidir.

AnalogWrite : Seçilen pinde PWM(Pulse Width Modulation) sinyalinin üretilmesini sağlar. PWM sinyalinin görev zamanı (Duty cycle) fonksiyona verilen deęer ile belirlenir.

```
analogWrite(pin,175);
```

DigitalRead: Daha önce giriş olarak tanımlanmış bir pinin girişindeki gerilimin yüksek (5 volta yakın) veya düşük (0 volta yakın) olduğu anlaşılır. Örneğin daha önceden giriş olarak tanımlanmış 13 nolu pindeki durumu daha önceden tanımlanmış 'durum' değişkenine yazdıralım.

```
durum = digitalRead(13);
```

13 numaralı pindeki gerilim 5 volta yakın ise 'durum' değişkeni 1 (HIGH), 0 volta yakın ise 0 (LOW) olmuştur.

AnalogRead: 0 ve 5 volt arasındaki gerilimin tam değerinin ölçülmesi için analogRead fonksiyonu kullanılmalıdır. Fakat bu fonksiyon, tüm pinler için kullanılamaz. Bu fonksiyonun kullanılabilceği özel pinler vardır. Bunlara analog pinler denir ve genellikle A harfi ile başlarlar: A0, A1, A2... gibi. Kullanılan Arduino türüne göre bu pinlerin sayısı değişiklik gösterir. Arduino Uno'da 6, Arduino Mega'da 16 tane analog giriş bulunur. Kullanılacak analog pin, öncelikle giriş olarak tanımlanmalıdır:

```
pinMode(A0, INPUT);
```

Delay : İki kod arasında bir süre beklenmesi istenirse, delay fonksiyonu kullanılır. Örneğin yanıp sönen lamba uygulaması yapılmak istenirse, lamba yakıp söndürme kodları arasında bekleme amacıyla delay fonksiyonu kullanılır. Delay fonksiyonunun içerisine yazılan değer milisaniye düzeyindedir. Bir saniyelik bir bekleme için fonksiyona 1000 değeri verilmelidir. Örneğin yarım saniyelik bir bekleme fonksiyonu ;

```
delay(500);
```

2.1.4 Haberleşme

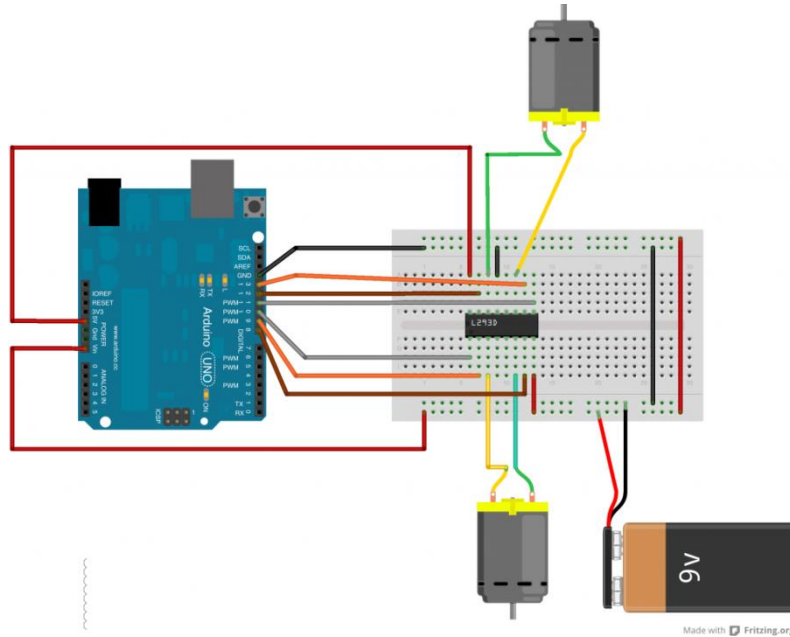
Arduino Uno bir bilgisayarla, diğer bir Arduino ile yada diğer mikrodenetleyiciler ile haberleşmek için bir takım özelliklere sahiptir. ATmega328, pin 0 (RX) ve pin 1 (TX) dijital pinleri üzerinde kullanılabilir olan UART TTL (5V) seri haberleşme sağlar. Board üzerindeki bir ATmega16U2 USB üzerinden seri haberleşme sağlar ve bilgisayarda sanal bir com port gibi gözükmektedir. '16U2' firmware, standart USB COM sürücülerini kullanır ve harici bir sürücüye gerek duymaz. Fakat Windows ortamında .inf uzantılı dosya gerekmektedir. Linux ve Mac işletim sistemlerinde harici sürücüye gerek duyulmaz. Arduino yazılımı, basit metinsel verileri Arduino boardına göndermek için seri bir monitör içerir. Board üzerindeki TX ve RX ledleri, veri seri USB yongası aracılığı ile iletilirken yanıp söner ve bilgisayarla USB bağlantısı sağlanır. Bu bağlantı pin 0 ve pin 1 üzerinde seri haberleşme için kullanılmaz. SoftwareSerial kütüphanesi Uno' nun herhangi bir pini üzerinde seri haberleşmeyi sağlar. ATmega328 sadece I2C (TWI) ve SPI haberleşmeyi destekler. I2C yollarını kolayca kullanabilmek için Arduino yazılımına bir Wire kütüphane eklenir. SPI haberleşmesi içinde SPI kütüphanesi kullanılmaktadır.

2.2 Motor Devresi

2.2.1 DC motorlar

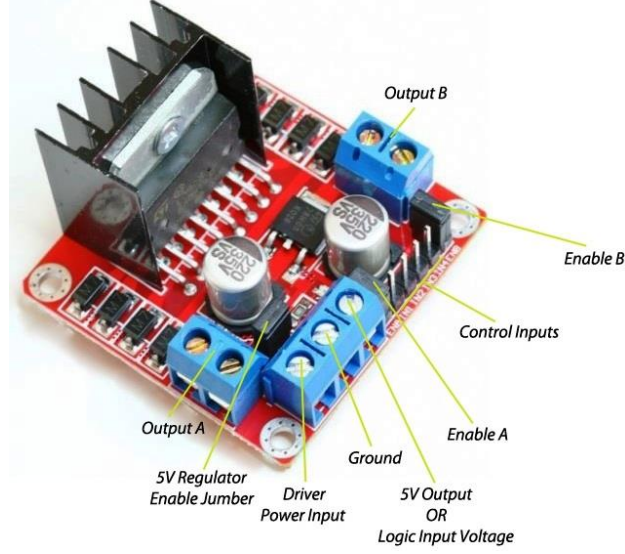
DC motorun ileri veya geri dönmesinin yanında, dönme hızını da Arduino üzerinden kontrol edilir. Arduino pinlerinden verilebilen akım motorları çalıştırmak için yeterli olmamaktadır. Bu yüzden DC motorlar, motor sürücülerle kullanılmalıdır. Motor sürücüsü kullanmadan doğrudan motoru Arduino'ya bağlamak Arduino'nun pinlerine zarar verebilir bu sebeple motor sürücü entegresi veya kartı kullanılmalıdır.

DC motorları kullanmak için motor sürücüsünü hazır devre kartı L298N Voltaj Regülatörlü Çift Motor Sürücü Kartı olarak veya kendiniz de devreyi L298 Entegresi kullanarak da tasarlayabiliriz. Aşağıdaki örnek şema L298 entegresi kullanılarak tasarlanmıştır. Arduino çıkış pinlerini yüksek seviyeye çekilerek ve 9V pil ile entegre beslenerek motorların çalıştırılması sağlanmıştır.



2.2.2 Sürücü Devre

L298N entegresi 4 giriş 4 çıkışa sahip H köprü motor sürücü entegresidir. L298 ile iki motoru birbirinden bağımsız olarak iki yöne sürmek mümkündür. Girişleri IN1-IN2-IN3-IN4, çıkışları OUT1-OUT2-OUT3-OUT4 ile belirtilmiştir. IN1-IN2 pini OUT1-OUT2 çıkışlarını, IN3-IN4 pinleri OUT3-OUT4 çıkışlarını kontrol etmektedir. İki yöne dönebilen bir motor için iki kanala ihtiyaç duyulduğundan L298N 'in 4 çıkışı ile 2 motor kontrol edilebilir. IN1den 5 volt uygulandığında OUT1 kanalında Vs pininden uygulanan gerilim görülür. Vs pininden uygulanan gerilimi ise enable pini kontrol eder.



Pin Özellikleri

ENA: Sol Motor Aktif Pini

IN1: Sol Motor 1. Giriş

IN2: Sol Motor 2. Giriş

IN3: Sağ Motor 1. Giriş

IN4: Sol Motor 2. Giriş

ENB: Sol Motor Aktif Pini

OUT1-2: Sol Motor Kanalı Çıkışı

OUT3-4: Sağ Motor Kanalı Çıkışı

L298N Özellikler

- Birbirinden bağımsız olarak iki ayrı motoru kontrol edebilir.
- Kanal başına 2A akım verebilmektedir.
- Üzerinde dahili regülatörü vardır.
- Yüksek sıcaklık ve kısa devre koruması vardır.
- Motor dönüş yönüne göre yanan ledler vardır.
- Kart üzerinde dahili soğutucu vardır.
- Akım okuma (current sense) pinleri dışa verilmiş haldedir.

2.3 Wifi Shield

Wifi modül sayesinde ortamdaki wireless modem aracılığı ile Arduino cihazı ile haberleşmeyi sağlar. IEEE802.11 seri port haberleşme protokolüne sahip cihaz IP alabilir ve bu sayede kablosuz haberleşme hatta internetten haberleşme imkanına ulaşmış olur. Bir ATMEGA 32UC3 bir ağ TCP hem de UDP yeteneğine sahiptir.



Wifi Shield



Arduino ile Wifi Shield

Üzerinde bulunan mikroSD kart girişi ile yapılan işlemleri kayıt altına alınabilir. Bu Wifi Shield Arduino ailesinden Uno ve Mega ile uyumlu bir şekilde çalışmaktadır. Kartın üzerindeki mikroSD kart okuyucu SD kütüphanesi aracılığıyla erişilebilir. Bu kütüphane ile çalışırken, SS Pin 4 üzerindedir.

Özellikleri:

- Bağlantı: 802.11b / g ağları
- Şifreleme tipleri: WEP ve WPA2 Kişisel
- Dahili mikro SD girişi
- WiFi shield seri hata ayıklama için FTDI bağlantı girişi
- WiFi shield yazılımını güncellemek için Mikro-USB girişi
- Shield üzerinde yeni protokol eklemek için açık kaynak yazılım özelliği.

2.4 Programlama

2.4.1 Client-Server

Android cihaz ile arduinoyu haberleştirmek için client server yöntemini kullandık. Arduinoyu client , arduinoyu server olarak tasarladık. Android cihazdan gönderilen veri arduino yani server tarafından sürekli olarak dinleniyor. Eğer ki arduinoya veri gelirse bu veri işlenerek istenen adımlar gerçekleştirilir.

Wifi shield üzerinden clientın gönderdiği verileri arduino işleyerek motorları çalıştıran kod bloğu aşağıdaki gibidir.

```
#define WLAN_SSID "ssid"
#define WLAN_PASS "password"
```

```

#define WLAN_SECURITY WLAN_SEC_WPA2
#define LISTEN_PORT 23

void loop(void)
{
  Adafruit_CC3000_ClientRef client = chatServer.available();

  if (client) {
    if (client.available() > 0) {
      uint8_t ch = client.read();
      count++;
      chatServer.write(ch);
      Serial.println(ch);
      digitalWrite(led,HIGH);
      if(ch=='w'){

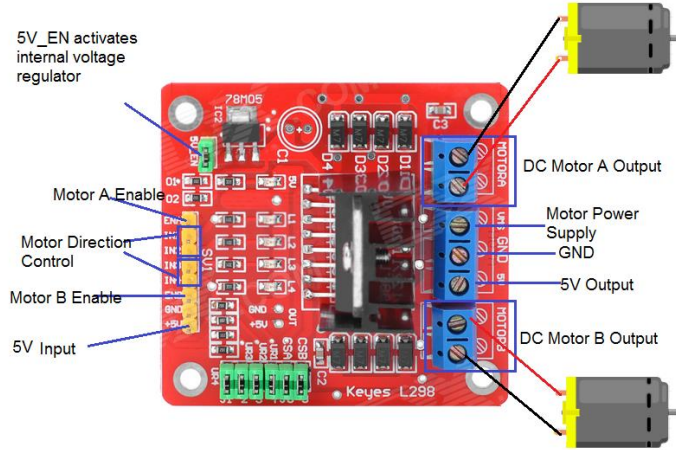
        for(int i=0;i<hiz_duz;i++){
          digitalWrite(in1,LOW);
          digitalWrite(in2,HIGH);
          digitalWrite(in3,LOW);
          digitalWrite(in4,HIGH);
        }

        digitalWrite(in1,LOW);
        digitalWrite(in2,LOW);
        digitalWrite(in3,LOW);
        digitalWrite(in4,LOW);
        Serial.println("ileri");
      }
    }
  }
}

```

2.4.2 Robot Hareketleri

Android cihazın kamerasından aldığımız verilere göre arabanın hangi yöne gideceğini arduinoya socket programlama ile gönderiyoruz. Cihazdan gönderilen veya arduinoya gelen veriler w,a,s,d yani ileri, ileri, sol,geri veya sağdır. Örnek olarak yol robotun ilerisinde olduğu bilgisi arduionaya gelirse motorun ikisinde çalışması için motor sürücü üzerindeki belirlenen pinler yüksek seviyeye çekilir ve robot ileriye hareket eder. Robotun diğer hareket kombinasyonlarında da bu mantık kullanılarak gerçekleştirilir.



3. SİSTEM YAZILIMI

3.1 Android Ortamı ve Programlama Temelleri

Google, Open Handset Alliance ve özgür yazılım topluluğu tarafından geliştirilen, Linux tabanlı, mobil cihaz ve cep telefonları için geliştirilmekte olan, açık kaynak kodlu bir mobil işletim sistemidir. Android, aygıtların fonksiyonelliğini genişleten uygulamalar yazan geniş bir geliştirici grubuna sahiptir. Geliştiriciler, ilk olarak aygıtı, Google'ın Java kütüphanesi aracılığıyla kontrol ederek Java dilinde yazmışlardır. Android, Linux çekirdeği üzerine inşa edilmiş bir mobil işletim sistemidir. Günümüzde akıllı telefonlar başta olmak üzere, dizüstü bilgisayarlar, tablet bilgisayarlar, E-kitap okuyucular, televizyon (Google TV), saat (I'm Watch) gibi birçok elektronik cihazda kullanılmaktadır. Biz de bu çalışmamızda Android Akıllı telefonları programladık. Programlama amacımız otonom karar mekanizması oluşturmak içindir. Android programlama hakkında ilk önce temel bilgiler edindik, basit uygulamalar gerçekleştirdik ardından OpenCv Android kütüphanesi ile çalışmaya başladık. Sistem yazılımının detaylarını ve aşamaları aşağıdaki gibidir.

Android bir program tasarlamak için Gerekli Programlar;

1. Java Development Kit (JDK)
2. Android SDK
3. Eclipse SDK

Gerekli programları indirdikten sonra öncelikle JDK'yı bilgisayarımıza kurduk.. JDK kurulumu tamamlandıktan sonra Eclipse'i zipten çıkarttık ve eclipse.exe'yi çalıştırdık. İlk çalıştırmada eclipse oluşturacağımız projeleri nereye kaydetmek istediğinizi soracaktır workspace için kayıt yeri belirledikten sonra karşımıza gelen arayüzden eclipse'e android Entegrasyonunu yaptık. Kurulum bittikten sonra Java programlama dilini kullanarak bir basit Android programı yazdık ve ardından bizim için gerekli programlama aşamalarına geçiş yaptık.

3.2 Android Client-Server



Otonom karar mekanizması Android bir işletim sistemine sahip olan akıllı telefon olduğu için android işletim sistemi üzerinden çalışan bir socket programı yazdık. Bu uygulama ile Otonom karar mekanizmasının ürettiği komutları bir network üzerinden Robota gönderimini sağladık. Android client(istemci) konumunda, robot(arduino) dinleyen server(sunucu) konumunda görev yapıyor. Arduino hep dinlemede iken bu yazdığımız Android programı hep komut üreten bir göreve sahip. Arduinonun 23 portunu tahsis ettik ve Arduinoya static bir IP adresi verdik. Bu sayede Android programından hangi IP'ye hangi porta veri göndereceğimizi biliyoruz. Aşağıda android client kısmının ana kod kısımları mevcuttur.

3.2.1 Basit Bir Veri Transferi

```

public void onClick(View v) {
    message ="ileri"; //örnek bir komutun üretilmesi
    SendMessage sendMessageTask = new SendMessage();
    sendMessageTask.execute(); // mesajın gönderilmesi
}
private class SendMessage extends AsyncTask<Void, Void, Void> {
    client = new Socket("192.168.43.219", 23); //server'a bağlantı
    kurulum
    printwriter = new PrintWriter(client.getOutputStream(), true);
    printwriter.write(message); // mesajın yazılımı
    client.close(); // bağlantıyı kapatma
}

```

3.2.2 Komutların Üretilmesi ve Komut Gönderimi



Kameradan aldığımız verilerin ardından otonom karar mekanizmasının oluşturduğu komutları robota gönderen, oluşan komutu Arduino makinenin anlayacağı bir biçime çeviren bir çevirici gerçekleştirdik. Örneğin ileri komutunu daha basit hale getirmek, transferini ve işleyişini hızlı yapabilmek için “ileri”->”w” komutuna çevirdik.



```
public void onClick(View v) {
    message =convert_mes("ileri"); // ileri-> w
    SendMessage sendMessageTask = new SendMessage();//Giden mesaj "w"
    sendMessageTask.execute(); // mesajın gönderilmesi
}
```

Robot üzerindeki Arduino "w" mesajını aldığı anda 2 arka tekerinide öne doğru sürecekle ve veri iletişimi ve komutun icrası yerine getirilmiş olacaktır.

3.3 Android OpenCV

3.3.1 OpenCV'ye Giriş

OpenCV (Open Source Computer Vision Library) Windows, Linux, Mac OS X, PSP işletim sistemleri üzerinde çalışabilen, C diliyle yazılmış, açık kaynak kodlu bir kütüphanedir. İlk sürümü Intel tarafından 1999'da çıkarılmıştır. Şu anda SourceForge tarafından geliştirilmektedir. Çoğu fonksiyonu platformlardan bağımsız olarak çalışmaktadır. C ve C++ dilleri ile geliştirilebilir. Özellikle yüz tanıma, işaret dili tanıma, hareket yakalama vb. gibi görüntü işleme algoritmaları ile bizlere kolaylık sağlamaktadır.

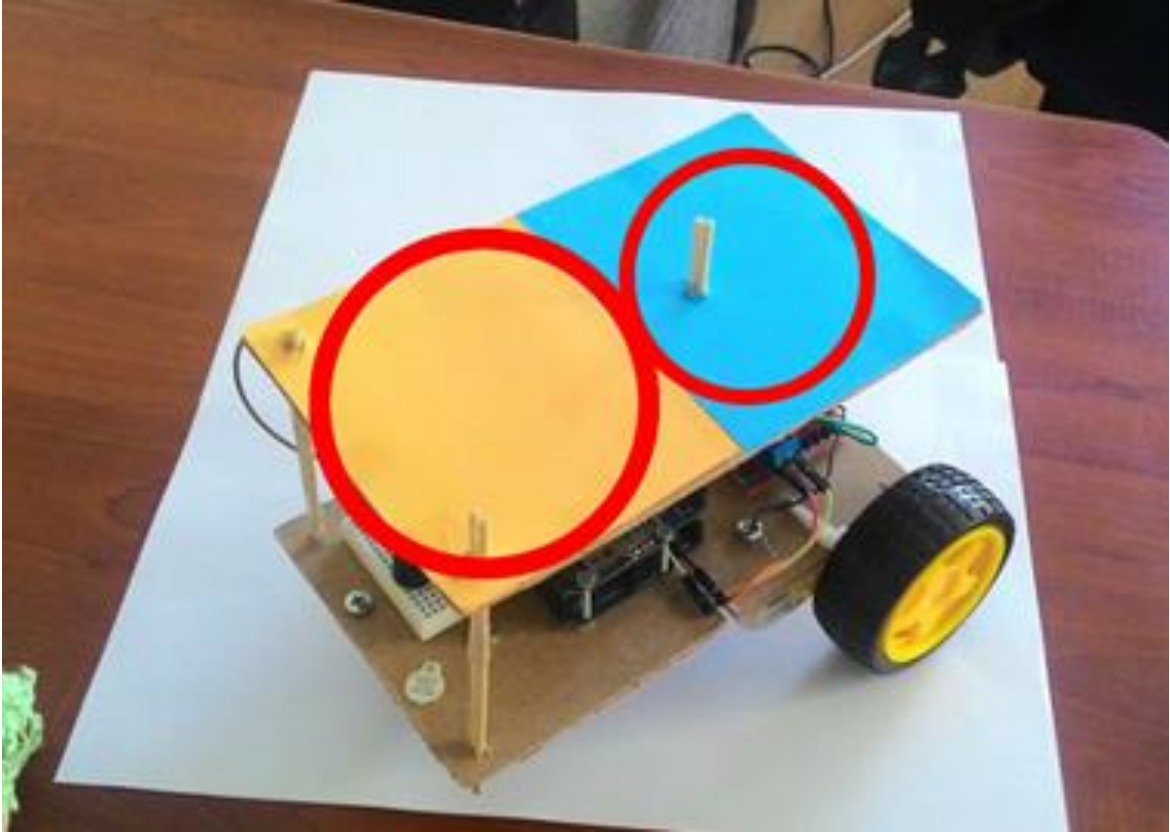
Bu kütüphane ile bir resmi veya video dosyası üzerinde kolaylıkla işlem yapılabilir. Farklı kameralardan alınmış görüntüleri yorumlanarak, herhangi bir nesnenin konumunu belirlenebilir.

OpenCV'nin -Machine Learning- adlı bölümünde yapay zeka ve kendiliğinden öğrenmeyle ilgili fonksiyonlar bulunmaktadır. Velhasıl OpenCV, daha doğrusu görüntü işleme, yeni ve gelişime açık bir alandır. Ayrıca robotik, tıbbi görüntüleme, güvenlik gibi pek çok alanda görüntü işleme için kullanılabilen 500'ün üzerinde yüksek seviyeli ve temel fonksiyon içerir.

Android üzerinde OpenCv ile ilk uygulamamızı gerçekleştirmemiz için OpenCv Sdk'sını OpenCv sitesinden indirip workspace'imize import etmemiz gerekir. Bu işlemle birlikte kütüphaneyi başarılı bir şekilde import etmiş oluyoruz.

3.3.2 Color Detection Hakkında Bilgi

Öncelikle renk tespiti uygulamasını gerçekleştirdik, renk tespiti ile robotun X, Y konumunu elde edeceğiz. Robotun önüne sarı, arkasına mavi rengini tahsis ettik. Anlık gelen görüntü karelerinden ilgili renkleri ayrıştıran bir OpenCV uygulaması tasarladık. İki renk kullanmamızın sebebi ise arabanın önü ile arkasını birbirinden ayırmaktı ve arabanın doğrultusunu tespit edebilmektir.



OpenCV tarafında arabanın doğrultusunu elde eden kod bloğu:

```

public class ColorBlobDetector {
public void process(Mat rgbaImage) {
    Imgproc.pyrDown(rgbaImage, mPyrDownMat);
    Imgproc.pyrDown(mPyrDownMat, mPyrDownMat);
    Imgproc.cvtColor(mPyrDownMat, mHsvMat, Imgproc.COLOR_RGB2HSV_FULL);
    Core.inRange(mHsvMat, mLowerBound, mUpperBound, mMask);
    Imgproc.dilate(mMask, mDilatedMask, new Mat());
    List<MatOfPoint> contours = new ArrayList<MatOfPoint>();
    Imgproc.findContours(mDilatedMask, contours, mHierarchy,
    Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE);
    // Find max contour area
    int i_max_contour=0;
    // Find max contour area
    double maxArea = 0;
    int i=0;
    Iterator<MatOfPoint> each = contours.iterator();
    while (each.hasNext()) {
        MatOfPoint wrapper = each.next();
        double area = Imgproc.contourArea(wrapper);
        if (area > maxArea){
            maxArea = area;
            i_max_contour=i;
        }
        i++;
    }
    mContours.clear();
    userPath.clear();
    each = contours.iterator();
    i=0;
    while (each.hasNext()) {
        MatOfPoint contour = each.next();
        if (Imgproc.contourArea(contour) > mMinContourArea*maxArea) {
            Core.multiply(contour, new Scalar(4,4), contour);

            if(i==i_max_contour && name!="maze")
            {
                int x=0;
                int y=0;
                Rect rect = Imgproc.boundingRect(contour);
                x=rect.x;
                y=rect.y;
                Point nokta=new Point();
                nokta.x=x;
                nokta.y=y;
                nokta.x=nokta.x+rect.width/2;
                nokta.y=nokta.y+rect.height/2;
                radius=rect.height/2;
                if(name!="maze"){
                    Imgproc.circle(rgbaImage,nokta,rect.height/2, new
    Scalar(255,0,0,0),4);
                }
                x_object=nokta.x;
                y_object=nokta.y;
                mContours.add(contour);
            }
        }
        i++;
    }
}
}

```

```

    }
}
private ColorBlobDetector mDetector_car_front,mDetector_car_back;
/**car front yellow-beyaz çizecek **
mBlobColorHsv_car_front.val[0]=33;
mBlobColorHsv_car_front.val[1]=134;
mBlobColorHsv_car_front.val[2]=253;
mBlobColorHsv_car_front.val[3]=0;
/**car front **

/**car back blue-yesil çizecek**
mBlobColorHsv_car_back.val[0]=136;
mBlobColorHsv_car_back.val[1]=255;
mBlobColorHsv_car_back.val[2]=255;
mBlobColorHsv_car_back.val[3]=0;
/**car back **
mDetector_car_back.name="car_back";
mDetector_car_front.name="car_front";
mDetector_car_back.process();//renk tespit işlemleri başlatılıyor
mDetector_car_front.process();//renk tespit işlemleri başlatılıyor

```

3.3.3 Robotun Tespit Edilmesi

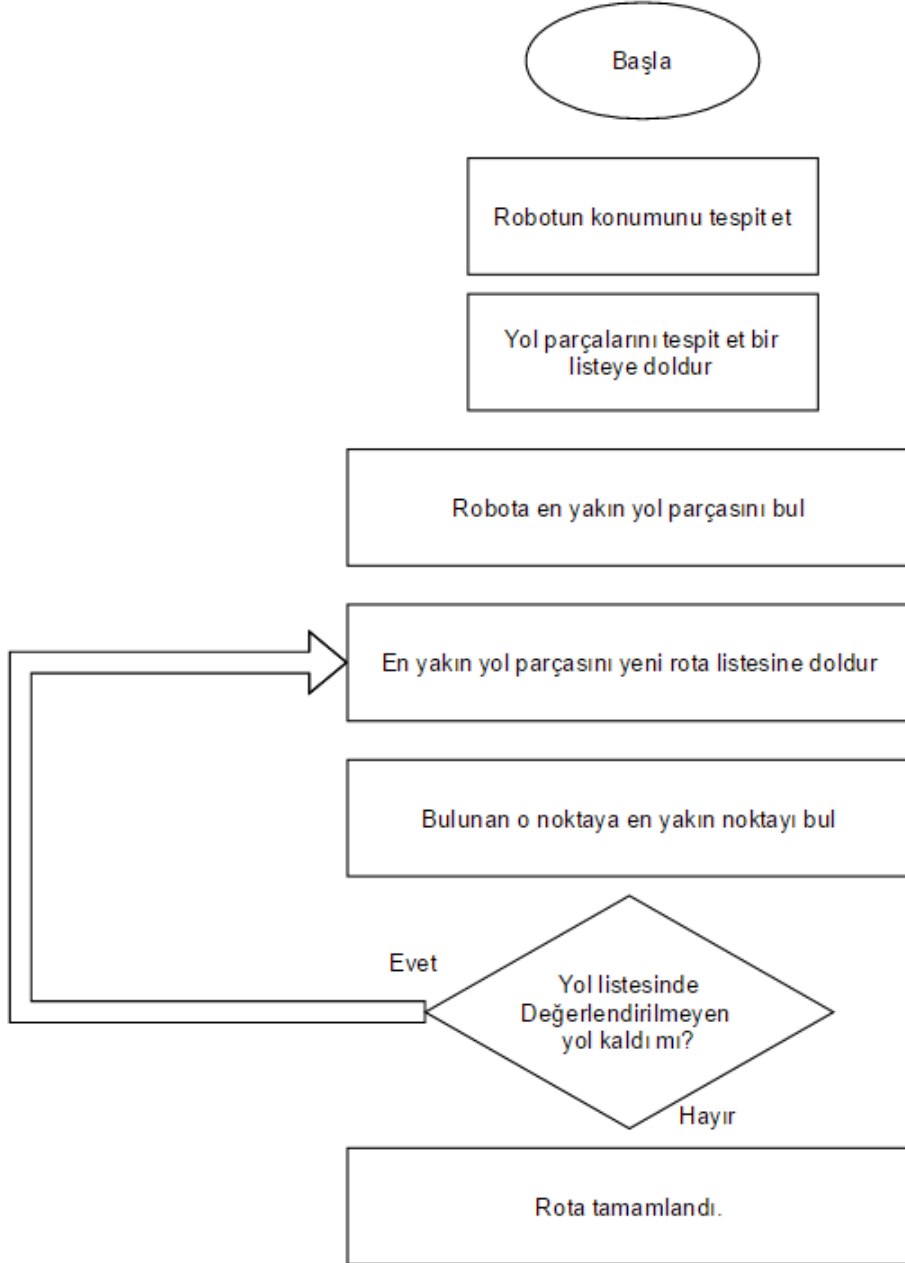
```

int X_car_front=(int)mDetector_car_front.getX();
int Y_car_front=(int)mDetector_car_front.getY();
int X_car_back=(int)mDetector_car_back.getX();
int Y_car_back=(int)mDetector_car_back.getY();
int line=(Y_car_back - Y_car_front)*(x - X_car_back) - (y -
Y_car_back)*(X_car_back - X_car_front);
//doğrultu hesaplanıyor

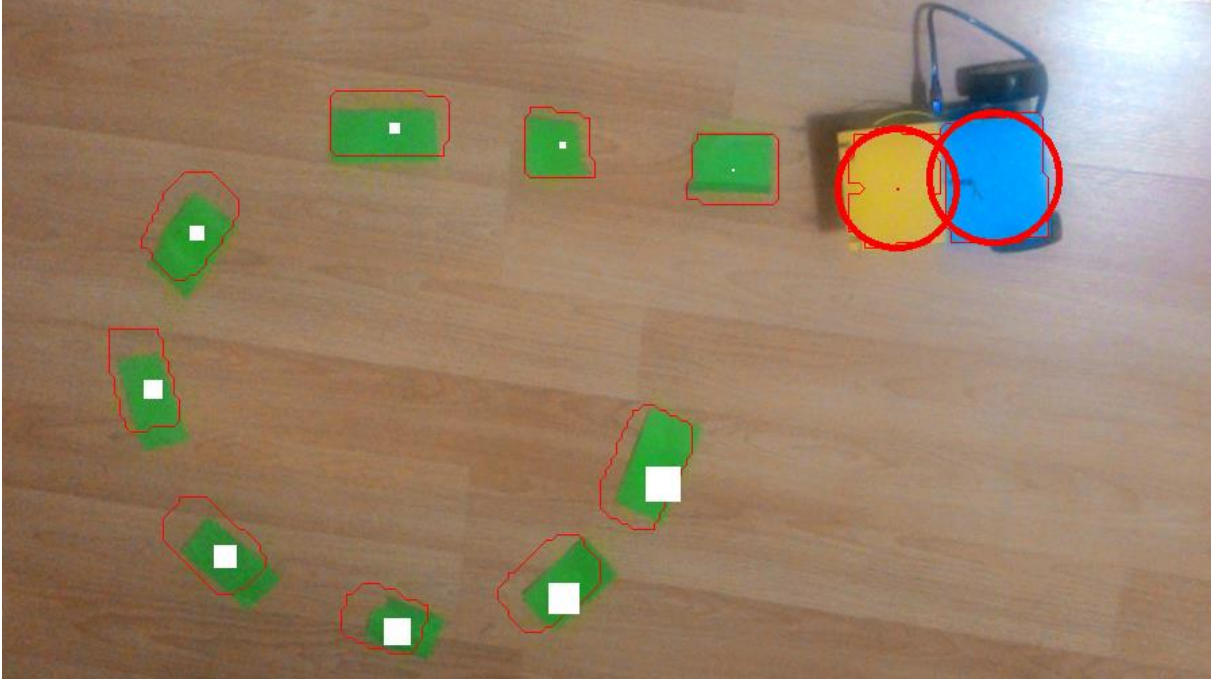
```

3.3.4 Yolun Tespit Edilmesi ve Rota Oluşturulması

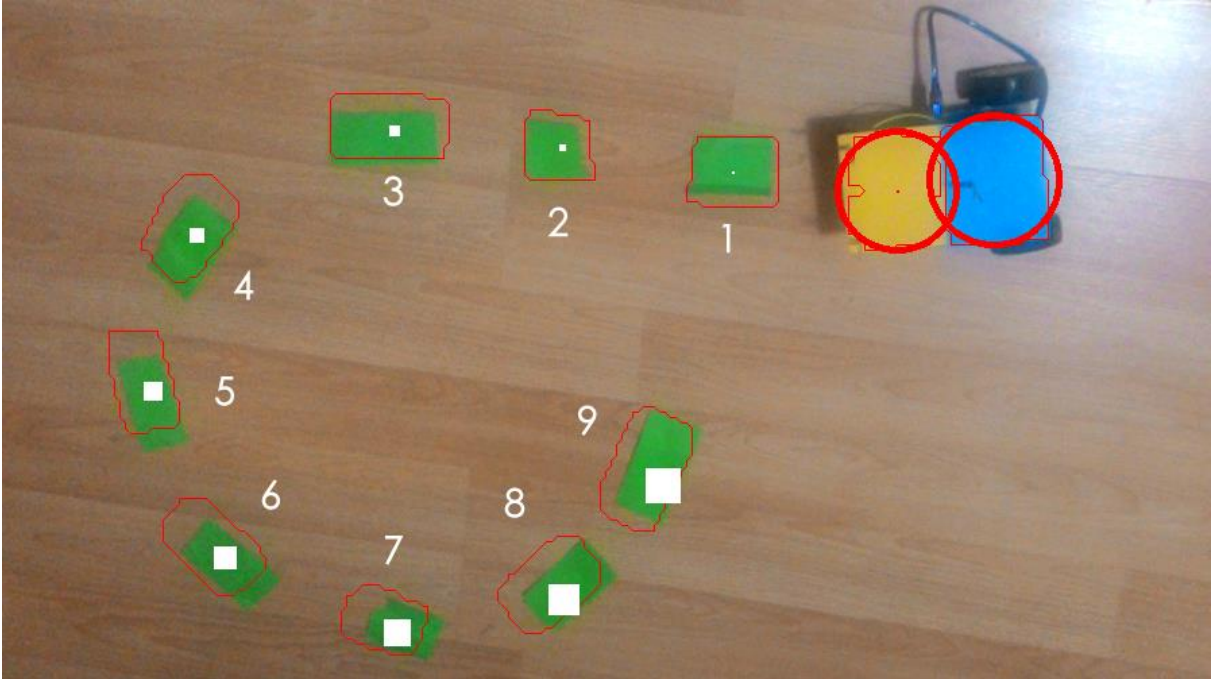
Kameradan aldığımız veriler ile yolun tespit edilmesi ve rotanın oluşturulması. Robota en yakın bir yol parçası bulunuyor sonra iteratif olarak o noktaya en yakın diğer bir yol parçası bulunuyor, bunlar bir listede bir rota olarak sıralanıyor. Aşağıdaki algoritma ile fotoğraflarda bir rotanın oluşturulmasının adımları gösterilmiştir.



Yol parçalarının tespit edilmesi.

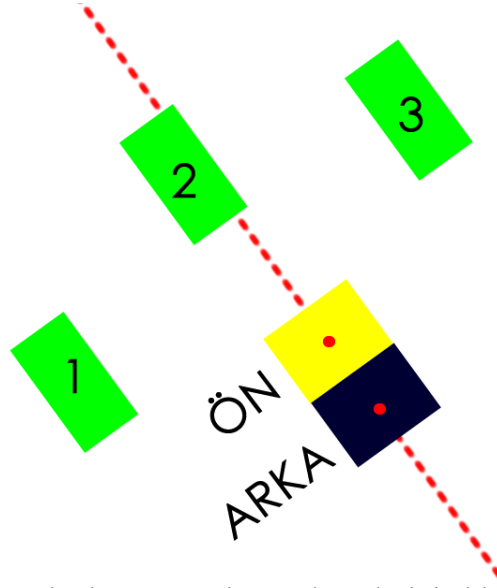


Rotanın oluşturulması.



Otonom karar mekanizmamız rotayı 1-2-3-4-5-6-7-8-9 şeklinde oluşturmuştur.

3.3.5 Karar Mekanizması ve Kullanılan Matematiksel Yöntemler



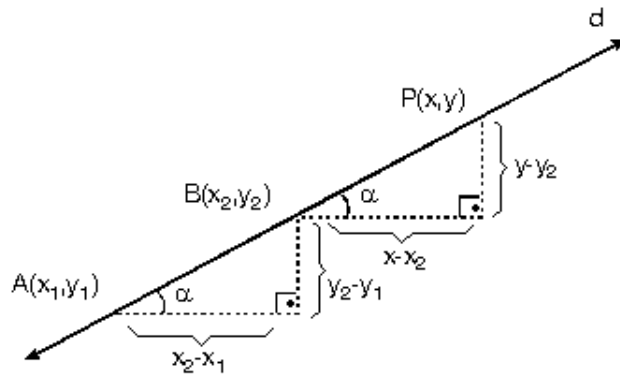
Robotun ön ve arka noktalarının pozisyon değerlerini aldıktan sonra robotun rotasını bir doğru ile temsil ediyoruz. Her durumları değerlendirdiğimizde yolun roboto göre 3 şekilde olması mümkün.

1.Durumda Yol, Robotun solunda duruyor ve otonom karar mekanizması robotun sola dönmesi gerektiğini söyleyen bir komut üretmeli.

2.Durumda Yol, Robotun önünde duruyor ve otonom karar mekanizması robotun ileri gitmesi gerektiğini söyleyen bir komut üretmeli.

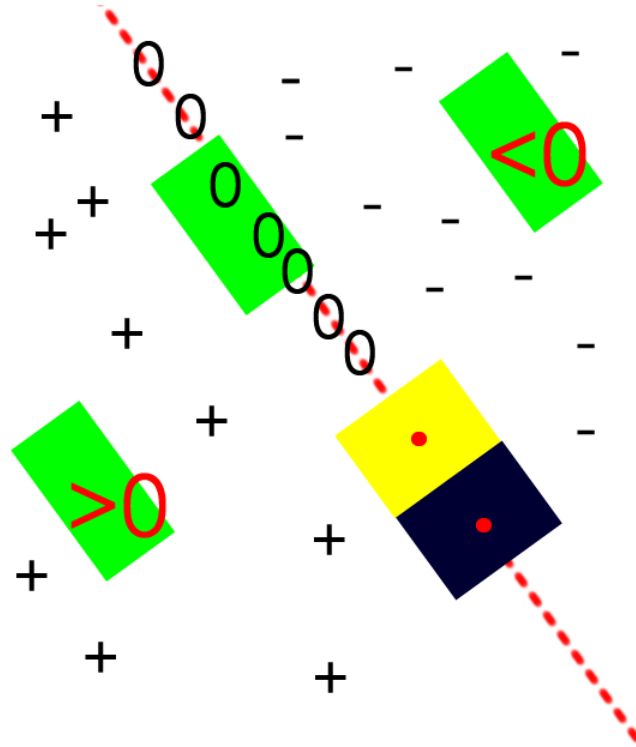
3.Durumda Yol Robotun önünde duruyor ve otonom karar mekanizması robotun sağa dönmesi gerektiğini söyleyen bir komut üretmeli.

Komutların üretilmesi için durumların değerlendirilmesi gerekir. Durumları değerlendirmek, otonom karar mekanizmasının algoritmasını oluşturmak için matematiksel yöntemler kullandık, kullandığımız matematiksel yöntemlerin amacı ve içeriği aşağıdaki gibidir.



$$\frac{Y_2 - Y_1}{X_2 - X_1} = \frac{Y - Y_2}{X - X_2}$$

2 noktası bilinen bir doğrunun denklemini bulan bir matematiksel yöntem, bu yöntemi kullanmamızın amacı robotun doğrultusunu bulmaktır. 2 nokta robotun önü ve arka noktalarıdır. Bu yöntem ile robotun doğrultusunu bulmuş oluyoruz. Robotun doğrultusunu bulduktan sonra 3 durum için komut oluşturacak algoritmamızın matematiksel yöntemi ve uygulandığı aşağıdaki gibidir.



Yolun pozisyon değerlerini Robotun bulduğumuz doğruya denklemine yerleştirdiğimizde;

1. Eğer değeri (>0) pozitif gelirse, otonom karar mekanizması ,yol robotun solunda kalıyor diyecek ve 1.durumu ifade etmiş olacak ardından komutunu üretmiş olacaktır.
2. Eğer değeri (<0) negatif gelirse, otonom karar mekanizması ,yol robotun sağında kalıyor diyecek ve 3.durumu ifade etmiş olacak ardından komutunu üretmiş olacaktır.
3. Eğer değeri ($=0$) 0 gelirse, Otonom karar mekanizması ,yol robotun ilerisinde kalıyor diyecek ve 2.durumu ifade etmiş olacak ardından komutunu üretmiş olacaktır.

```
int line=(Y_car_back - Y_car_front)*(x - X_car_back)- (y -  
Y_car_back)*(X_car_back - X_car_front);  
int threshold_value=(int)mDetector_car_front.getRadius(); //on line ?  
  
if(line>0 && distance>threshold_value){  
    Log.i(TAG, "1 birim saga dön");  
  
}else if(line<0 && distance>threshold_value){  
    Log.i(TAG, "1 birim sola dön :");  
}else{  
    Log.i(TAG, "1 birim ileri :");  
}
```

Üretilen komut socket programlama ile Robota gönderilecek ve Robot o komutu alarak icra edecektir.

4.SONUÇLAR

Genel olarak otonom araçlar ele alındığında; gerekli donanım ve yazılımla birlikte cihazların kendi kendini kontrol etmesini sağlamanın çokta zor olmadığı anlaşılmaktadır.Cihazın gerçekleştireceği işlemlere göre otonomluğun sağlanması çok üst düzey bir seviyeye çıkabilir.Projenin gerçekleşmesinde en önemli problem ortam değişkenlerine bağlı olarak sensörlerin yanlış değer üretmesidir.Proje genişletilebilirlik açısından değerlendirildiğinde ise haberleşme sistemi önem kazanmaktadır.Haberleşme sistemindeki en büyük kısıt maliyettir.Genel olarak sorunlar giderildiğinde ve daha ayrıntılı bir çalışma yapıldığında projenin geliştirilebilirlik oranı oldukça yüksektir.Kontrol cihazlarının ve kontrol edilecek cihazlarda göz önüne alındığında projenin sınırı olmadığı görülmektedir.

5. ÖNERİLER

Projeye başlamadan önce iyi arařtırmalar yapıp opencv, robot yapımı, android programlama, arduino, socket programlama gibi konular hakkında bilgi edinilir. Donanımsal olarak gerekli malzemeler tahsis edildikten sonra basit uygulamalarla yapılmalıdır. Android cihaz ile arduinoya socket programlama ile gönderilen veri arduino tarafından wifi shield aracılığı ile gelen veri işlenmesi için gerekli adımlar ve algoritmalar derinlemesine incelenmelidir. Bu aşamada yazılımsal olarak arařtırmalar yapılmalı ve bileşenlere hakim olunmalıdır.

Görüntü işleme olarak projemizde OpenCV hazır kütüphanesi kullanmamıza rağmen kendi algoritmalarımızı kullanarak birçok olayları gerçekleřtirdik. Bu sebeple ilk olarak OpenCv'nin open source olarak paylařtığı template projeler incelenerek kodun akışı hakkında bilgi edinilmelidir. Robotun hareketi için görüntü işleme ile elde edilen veriler oluşturulan algoritmaya göre arduinoya gönderilerek gerekli işlemler yapılır.

6. KAYNAKLAR

- [1]<https://evothings.com/doc/examples/arduino-led-onoff-tcp.html>
- [2]<https://www.arduino.cc/en/Reference/WiFi>
- [3]<http://android-er.blogspot.com.tr/2014/02/android-sercerclient-example-client.html>
- [4]<http://stackoverflow.com/questions/9443377/android-color-detection-using-opencv-how-to>
- [5]https://docs.google.com/presentation/d/1s0KvqZm0kmPYRdjQNgKFPan0RjNWBCjke m23UkXh2GE/edit#slide=id.g335951afa_012
- [6] <https://en.wikipedia.org/wiki/Arduino>
- [7] <http://www.robotiksystem.com/L298datasheet.pdf>

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?)

Projemiz yeni bir proje olup , var olan herhangi bir projenin tekrarı değildir.Tasarımımız projenin %100'nü oluşturmaktadır.

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

Durumları değerlendirmek,otonom karar mekanizmasının algoritmasını oluşturmak için matematiksel yöntemler kullandık.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Elektronik devreleri dersinde edindiğimiz temel elektronik bilgisi, NYP dersi ve görüntü işleme dersinden edindiğimiz temel bilgiler.

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

TCP/IP protokolü

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

- a) Ekonomi

400 ₺

b) Çevre sorunları:

Işık değişimine göre elde ettiğimiz görüntülerde renklerin birbirine karışımı olmaktadır. Uygun aydınlatmada sistemin başarılı bir şekilde çalıştığı görülmektedir.

c) Sürdürülebilirlik:

Sürdürülebilirlik açısından değerlendirildiğinde proje ileri götürülebilir, ihtiyaç ve istekler doğrultusunda yeni eklemeler yapılarak geliştirilmeye açıktır.

d) Üretilebilirlik:

Günümüz teknolojilerine uyarlanabilir.

e) Etik:

Projemizi etik açıdan değerlendirisek hiçbir sorun gözetmemektedir.

f) Sağlık:

Projenin insan ve doğaya hiçbir kötü etkisi yoktur.

g) Güvenlik:

Sistemi kullanan kullanıcılar güvenlik açısından hiçbir saldırı tehdidi altında bulunmazlar.

h) Sosyal ve politik sorunlar:

Projemizin sosyal ve politik açıdan sorun teşkil etmemektedir.