ENDÜSTRİYEL ÇİZGİ TAKİP EDEN ROBOT CİHAZI GELİŞTİRİLMESİ

IMPROVEMENT OF INDUSTRIAL PATH FOLLOWING ROBOT

KIVANÇ İREN

Hacettepe Üniversitesi

Lisansüstü Eğitim - Öğretim ve Sınav Yönetmeliğinin

MAKİNA MÜHENDİSLİĞİ Anabilim Dalı İçin Öngördüğü

YÜKSEK LİSANS TEZİ

olarak hazırlanmıştır.

2013

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından **MAKİNA MÜHENDİSLİĞİ ANABİLİM DALI 'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan :….....(İmza).....
(Unvanı, Adı ve Soyadı)

Üye (Danışman) :….....(İmza).....
(Unvanı, Adı ve Soyadı)

Üye (Varsa İkinci Danışman) :….....(İmza).....
(Unvanı, Adı ve Soyadı)

Üye :….....(İmza).....
(Unvanı, Adı ve Soyadı)

Üye :….....(İmza).....
(Unvanı, Adı ve Soyadı)

ONAY

Bu tez ...../...../..... tarihinde Enstitü Yönetim Kurulunca kabul edilmiştir.

Prof.Dr. ……..
Fen Bilimleri Enstitüsü Müdürü

Hayat Arkadaşıma . . .

ENDÜSTRİYEL ÇİZGİ TAKİP EDEN ROBOT CİHAZI GELİŞTİRİLMESİ

KIVANÇ İREN

ÖZ

Çizgi Takip Eden Robotlar günümüzde gelişen teknolojiyle birlikte endüstriyel alanda bir süredir kullanılmaktadır. Genellikle gelişen sanayinin oluşturduğu büyük sanayi kuruluşlarında lojistik ve otomasyon bölümleri içerisinde fazlaca gereksinim duyulan niteliksiz insan gücü ile yapılan taşıma işlemlerini bir süredir çizgi takip eden robotlar yapmaya başlamışlardır.

Bu tez çalışmasında; öncelikli olarak literatürde bulunan çizgi takip eden robot projeleri incelenmiş, daha sonra endüstriyel anlamda kullanılabilecek, dolum ve boşaltım istasyonlarını tanıma özelliğine sahip, istasyonlar ile kablosuz haberleşme sayesinde iletişim halinde olabilen ve temel görevi olan çizgi takip etme işlemini devam ettirebilecek bir robot tasarım üzerine çalışılmıştır. Bu noktada çizgi takip etme işleminin hem endüstriyel ortamlarda çözüm amaçlı fikirler verebilmesi, hem de yazılımsal anlamda ihtiyaçları karşılayabilmesi ve yeni sistemleri destekleyebilmesi amaçlanmıştır.

Tasarım ve programlama esnasında ortam koşullarından meydana gelebilecek iş güvenliği hassasiyetleri de analiz edilmiştir. Prototip çalışmasında vizyon olarak öngörülen çözümler büyük ölçekli robotlarda fikirsel ve yöntemsel olarak uygulanabilir şekilde geliştirilmiştir.

Bu tezde yapılan prototip robotta öğrenilen tasarım, yazılım ve kontrol sistemleri gibi tecrübeler büyük ölçekteki robotlarda ve endüstriyel uygulamalarda da kullanılabilmektedir. Çizgi Takip Eden Robotlar gelecekte sanayinin lojistik ve taşıma problemlerine çözüm getirerek endüstriyel anlamda otomasyon sistemlerinde kullanılmaya başlayacaktır.

Anahtar Kelimeler: Çizgi Takip Robot, Endüstriyel Çizgi takip , Path following

Danışman: Öğretim Görevlisi Dr. Özgür Ünver, Hacettepe Üniversitesi, Makina Mühendisliği Bölümü

DEVELOPMENT OF INDUSTRIAL LINE-FOLLOWING ROBOT DEVICES

KIVANÇ İREN

ABSTARCT

Today, line-following robots are being used in the industrial field in accordance with the emerging technology. For a while, line-following robots have begun conducting the transport operations generally carried out by unskilled manpower that is quite much needed within logistics and automation departments of major industrial enterprises of developing industries.

In this thesis, primarily the line-following robot projects in literature are studied, and then a robot design that can be industrially used, has the ability to recognize filling and discharge stations, can communicate with the stations through wireless communication and can maintain the line-following operation as its main task is worked on. At this point, the line-following operation is aimed both at generating ideas targeting solutions in industrial environments and meeting software needs, and also at supporting new systems.

In the course of design and programming, job security concerns that might arise from the conditions of the environment are analyzed as well. The solutions provided as visions in the prototype study are enhanced to be conceptually and methodologically applicable in large-scale robots.

The experiences such as the design, software and control systems studied in the prototype robot developed in this thesis can also be used in large-scale robotic systems and industrial implementations. In the future, line-following robots will start to be used industrially in automation systems by providing solutions to the logistics and transportation problems of industry.

Keywords: Line-following robot, Industrial line-following, Path-following

Advisor: Lecturer Dr. Özgür Ünver, Hacettepe University, Department of Mechanical Engineering

TEŞEKKÜR

Sabrı, tecrübesi ve ilgisiyle tez çalışmamı sürdürmem için desteğini, bilgisini ve yol göstericiliğini hiçbir zaman esirgemeyen danışmanım Sayın Dr. Özgür Ünver'e,

Bana olan sonsuz inancını, güvenini ve sevgisini herzaman hissettiğim hayat arkadaşım Özlem İren'e,

Yüksek Lisans Sürecim boyunca bana her türlü desteği gösteren departman müdürüm OYAK – RENAULT Eğitim Enstitüsü Departman Müdürü Erdoğan Güneş'e,

Sevgisiyle ve sabrıyla bana her zaman güç veren, beni bugünlere getiren ve en iyi şekilde yetiştiren anneme ve babama,

Yüksek Lisans sürecim boyunca ellerinden gelen her türlü yardımı koşulsuz yapan arkadaşlarım Ergül, Ahmet ve Merih'e,

Robit Teknoloji'den Nevzat Bey ve Uğur Bey'e sonsuz içtenliklerimle

Teşekkür ederim.

İÇİNDEKİLER DİZİNİ

ÖZ……..i

ABSTRACT……..ii

TEŞEKKÜR iii

İÇİNDEKİLER DİZİNİ iv

ŞEKİLLER DİZİNİ vi

ÇİZELGELER DİZİNİ viii

SİMGELER VE KISALTMALAR DİZİNİ ix

1. GİRİŞ 1

1.1. Problem Tanımı 1
1.2. Motivasyon 2
1.3. Karşılaşılan Zorluklar 3
1.4. Değerlendirme 4

2. SİSTEM TASARIMI 5

2.1. Yaygın Olarak Kullanılan Çizgi Takip Yöntemleri 5
2.1.1. CCD Kamera Yöntemi ile Ön Tanımlı Çizgi Takibi 6
2.1.2. Manyetik Algılama Yöntemi İle Ön Tanımlı Çizgi Takibi 8
2.1.3. Ultrasonik Algılama Yöntemi İle Ön Tanımlı Çizgi Takibi 11
2.1.4. Optik Algılama Yöntemi İle Ön Tanımlı Çizgi Takibi 11
2.1.5. Genel Olarak Kullanılan Yöntemlerin Değerlendirmesi 14
2.2. Robot Sürüş Yönteminin Belirlenmesi 18
2.3. Tekerlekli Robotlar için Sürüş Yöntemlerinin İncelenmesi 21
2.3.1. Ackerman Sürüş yöntemi 21
2.3.2. Senkron sürüş yöntemi 23
2.3.3. Üç tekerlekli sürüş yöntemi 24
2.3.4. Diferansiyel sürüş yöntemi 24
2.4. Kontrol Elemanlarının Belirlenmesi 31
2.4.1. Robot merkez işlemcisinin belirlenmesi 34
2.4.2. Parça işleme istasyonlarının merkez işlemcisinin belirlenmesi 37
2.4.3. Motor sürücü ünitesi arduino motor sürücü devresi 39

3. SİSTEMİN PROTOTİP OLARAK ÜRETİMİ 40
3.1. Sistem Çalışma Prensibi 40
3.2. Sistemin Proje akış diyagramı 41
3.3. Kullanılan Malzemeler 42
3.4. Sistemin Mekanik Yapısı 43
3.5. Sistemin Elektronik Yapısı 48
3.6. Sistemin Programlanması 49
3.6.1. Çizgi izleme ünitesinin programlanması 49
3.6.2. Haberleşme ünitesinin programlanması 53
3.6.3. İstasyonların programlanması 55
3.6.4. Otomasyon işleminin programlanması 57
4. SONUÇLAR VE ÖNERİLER 58
KAYNAKLAR DİZİNİ 60
EKLER DİZİNİ 65
EK 1. ARDUINO MEGA 2560 PİN YERLEŞKESİ 66
EK 2. ARDUINO MEGA 2560 PİN YERLEŞKE TABLOSU 67
EK 3. ARDUINO MEGA 2560 DEVRE ŞEMASI 72
EK 4. ARDUINO UNO PİN YERLEŞKESİ 73
EK 5. ARDUINO UNO DEVRE ŞEMASI 74
EK 6. ARDUINO MOTOR KORUMA DEVRE ŞEMASI 75
EK 7. AĞIRLIK SENSÖRÜ BOYUTLAR VE GERİLİM BÖLÜCÜ DEVRESİ 76
EK 8. ÇİZGİ İZLEME ÜNİTESİNİN PROGRAMLANMASI 77
EK 9. HABERLEŞME ÜNİTESİNİN PROGRAMLANMASI 88
EK 10. İSTASYONLARIN PROGRAMLANMASI 93
EK 11. OTOMASYON İŞLEMİNİN PROGRAMLANMASI 105

ŞEKİLLER DİZİNİ

Şekil 2.1. Kameranın kontrast farkından yolu algılaması 6

Şekil 2.2. CCD Kameranın robot üzerine yerleşimi 7

Şekil 2.3. Açıya göre değişen manyetometre sensör çıktısı ve x-y konumları 9

Şekil 2.4. Manyetik alan ölçme modülü V2Xe 9

Şekil 2.5. Sonar Sensör 11

Şekil 2.6. LDR ve LED ekipmanlarının çalışma prensibi 12

Şekil 2.7. Optik sensör ile Çizgi takip eden robotta ana bölümler 13

Şekil 2.8. Optik algılama sensörleri 14

Şekil 2.9. Çizgi Takip Eden Robotların Çalışma Prensibi 14

Şekil 2.10. Ackerman sürüş yöntemi 22

Şekil 2.11. Senkron sürüş tekniği a. Alttan görünüş. b. Üstten görünüş 23

Şekil 2.12. Üç tekerlekli sürüş yöntemi 24

Şekil 2.13. Diferansiyel Sürüş yönteminde motorların hareket verme prensibi 25

Şekil 2.14. Diferansiyel Robot İçin anlık eğrilik merkezi 28

Şekil 2.15. Diferansiyel robotun kinetik modeli 29

Şekil 2.16. Robotun kinetik modeli 30

Şekil 2.17. Sistemde kullanılan kontrol elemanları 32

Şekil 2.18. Robotun merkezi işlemcisi olarak kullanılan arduino mega 2560 37

Şekil 2.19. Parça işleme istasyonlarında işlemci olarak kullanılan arduino uno ... 38

Şekil 2.20 Motor sürücü devresi arduino motor kartı 39

Şekil 3.1. Sistemin çalışma prensibi 41

Şekil 3.2. Robot şasesi ve motorların 3 boyutlu ortamda tasarımı..... 43

Şekil 3.3. Prototip robotun 3 boyutlu ortamda tasarımı..... 44

Şekil 3.4. Robotun ön-üst den görünümü 45

Şekil 3.5. Robotun arka-üst den görünümü 45

Şekil 3.6. Parça işleme istasyonlarının üstten görünümü 46

Şekil 3.7. Sistemin genel görünümü 47

Şekil 3.8. Ağırlık Sensörü Direnç – Kuvvet değişim diyagramı 47

Şekil 3.9. Sistemin elektronik sinyalizasyon şeması..... 48

Şekil 3.10. Çizgi izleme ünitesi programlama mantık şeması..... 51

Şekil 3.11. Çizgi izleme işleminin programlanması esnasında arduino yazılımının görüntüsü 52

Şekil 3.12. Haberleşme ünitesinin programlama mantık şeması 53

Şekil 3.13. Haberleşme işleminin programlanması esnasında arduino yazılımının görüntüsü 54

Şekil 3.14. İstasyonların programlanmasındaki mantık şeması.....55

Şekil 3.15. İstasyonların tanımlayıcı bilgilerini oluşturduğumuz programın bir görüntüsü……..……..56

Şekil 3.16. İstasyonların programlanmasında yazılımın ekran görüntüsü……....56

Şekil 3.17.Otomasyon işleminin programlanması esnasında arduino yazılımının görüntüsü…….57

ÇİZELGELER DİZİNİ

Tablo 2.1. Çizgi takip yöntemlerinin karşılaştırılması 16

Tablo 3.1. Robot hareket yöntemlerinin karşılaştırılması 19

Tablo 3.2. Robot sürüş yöntemlerinin karşılaştırılması 26

Tablo 3.4. Mikro İşlemcilerin karşılaştırılması 35

Tablo 4.1. Kullanılan malzemelerin maliyet tablosu 42

SİMGELER VE KISALTMALAR DİZİNİ

M Araç merkez noktası

v1 Araç sağ tekerlek hızı

v2 Araç sol tekerlek hızı

vm Araç merkez noktasının hızı

vx Araç merkez noktasının hızının yatay bileşeni

vy Araç merkez noktasının hızının dikey bileşeni

θ Aracın oryantasyon açısı

Mz Araç dönerken oluşan zahiri merkez

L Aracın tekerlekleri arasındaki mesafe

r Oluşan zahiri merkezin araç sol tekerleğe olan uzaklığı

x Araç merkezinin yatay konumu

y Araç merkezinin dikey konumu

δi iç teker sapma açısı

δo gerçek dış teker sapması

δoA ideal dış teker sapması (aracın iz genişliği ve aks aralığına bağlı Ackermann koşulu)

Δδ direksiyon hatası

Pwm darbe genişlik modülasyonu

CCD Şarj eşleştirmeli cihaz

LED Işık yayan diyot

LDR Işığa bağlı direnç

1. GİRİŞ

1.1. Problem Tanımı

Bu tezin amacı; endüstriyel lojistik faaliyetlerinde kullanılabilecek, önceden belirlenmiş üretim alanlarında dolum ve işleme istasyonlarına sahip, belirli bir rotası olan, üretim istasyonları ile kablosuz olarak haberleşebilen, otonom olarak üretim istasyonlarına hammadde ve malzeme besleyebilen, çalıştığı ortamdaki oparetörlerin işgüvenliğini riske atmayan küçük ölçekli bir endüstriyel çizgi takip eden robot prototipi geliştirilmesidir. Bu projede elde edilmek istenen sonuçlara ulaşmak için kullanılacak başlıca yöntemler,

- açık kaynak bir programlama dili kullanılarak robotta kullanılacak yazılımın anlaşılabilirliğini kolaylaştırmak,
- büyük uygulamalar için altyapı oluşturarak, istasyonlar ile robotun haberleşebilmesi için kablosuz modüllerin sisteme entegre edilmesini sağlamak,
- robota otonom çalışma yetisi kazandırmaktır.

Üretim sistemlerinde günümüz yöntemlerinin karşılaştığı en büyük problemlerden bir tanesi üretim hatlarına malzeme tedariğinin (Kumar et al., 2006) sağlanmasıdır. Bu tür lojistik problemlerinin giderilmesi için üretim lokasyonlarında malzeme hazırlama ekipleri oluşturulmakta ve bu ekipler forklift, sürücülü transpalet ve elektrikli çekici gibi kaldırma ve taşıma araçları ile üretim esnasında (Hirao et al., 2010) kullanılacak hammaddeleri bant kenarına taşımaktadır. Bu eylemin ortaya çıkardığı sonuçlar,

- ekstra insan gücü,
- kaldırma ve taşıma araçlarının harcadığı ekstra enerji,
- bu taşıma esnasında karşılaşılan işgüvenliği ve malzeme devrilme problemleridir.

Bu tezde cevaplanmaya çalışılacak olan sorular aşağıda maddeler halinde verilmiştir,

- Bu problemlerin aşılması için neler yapılabilir ?
- Nasıl bir mekanik tasarım yapılmalıdır ?
- Bu mekanizmada nasıl bir otomasyon sistemi bulunmalıdır ?

1.2. Motivasyon

Bu tez kapsamında ortaya çıkarılacak olan çıktıların bulunması için sanayi lojistik uygulamaları incelenmiştir. Bu incelemeler sonucunda ortaya çıkan sonuçlar aşağıda maddeler halinde verilmiştir.

Üretim hacminin yüksek olduğu fabrikalarda hatlara malzeme tedarik işlemini kaldırma ve taşıma araçları yaptığı için lojistik işlemlerine standart operasyon niteliği (Hase et al., 1998) verilememiştir, çünkü malzeme taşıma işlemi her seferinde kullanıcı operatörün araç sürüş şekline bağlı olmuştur.

Kaldırma ve taşıma araçları ile malzeme taşıma işleminde en az 3 hareket (M.E.B., 2011) gerektiğinden (kaldırma – taşıma – İndirme) bu eylemlerin herhangi birinde hammade ve malzeme devrilme, bozulma veya hasar alma riski ile karşı karşıya kalmıştır.

Üretim planlaması esnasında malzeme tedarik planlaması üretime başlamadan önce öngörü (Pechoucek et al., 2006) olarak yapılmaktadır. Bant kenarından anlık olarak kütle varlık kontrolu yapan herhangi bir sistem mevcut değildir.

Bu analizlerin ışığında oluşturulmak istenen otomasyon sistemine sahip ve lojistik işlemlerini otonom olarak sağlayabilecek "endüstriyel çizgi takip eden robot" projesinde robotun sahip olması gereken özellikler şu şekilde belirlenmiştir.

- İstasyon tanıma özelliğinin olması: Mevcut uygulamalar incelendiğinde robotlar belirli bir istasyon tanıma ve haberleşme yapısına sahip olmadığından (Sezen, 2003) taşıyacakları malzemeyi götüreceği mesafeler numerik veri olarak işlemcilerine girilmektedir. Bu durum robotun hareket

ederken ekstra manevralar yapması halinde malzemeyi istenilen pozisyona ulaşamaması durumunu (Schulze et al., 2009) doğurmaktadır. Bizim çalışmamızda bu durumu engelleyebilmek adına “Merkez İstasyon”, “Dolum İstasyonu” ve “Çalışma İstasyonu” gibi tanımlar yapılarak robotun takip ettiği rota boyunca belirli gezi modellemeleri gerçeklenmiştir.

- Malzeme kontrol ve haberleşme sisteminin olması: Prototip robotun otonom olarak çalışabilmesi için parça işleme istasyonlarındaki malzemenin bittiğini anlayabilmesi ve bu istayonlar ile haberleşebilmesi gerekliliği mevcuttur. Bu sayede robot sürekli olarak hareket halinde olmayacak, enerji tasarrufu ve batarya ömrü konularında daha verimli çalışacaktır.

- İş Sağlığı ve güvenliği kurallarına (6331 Sayılı Kanun Mevzuatı, 2012) uygun tasarım: Fabrika ortamında çalışacak olan prototip robotun güzergahında mutlak suretle insan ve taşıma aracı gibi “engel” olarak nitelendirebileceğimiz birçok varlık çıkacağından robot öncelikle bu nesneleri gördüğünde durmalı ve nesneler rotasından uzaklaşana kadar beklemelidir daha sonra aldığı komutları kaldığı yerden devam ederek aynı şekilde yerine getirmelidir.

1.3. Karşılaşılan Zorluklar

Bu projede karşılaşılan zorlukları şu şekilde sıralanabilir;

- Öncelikle robotun istasyon tanıma ve parça işleme istasyonlarından gelen verileri hafızasına alarak birden fazla malzemenin farklı istasyonlarda aynı anda bitmesi durumunda gerekli tepkileri verebilmesi ve tüm rotasyonu işlemcisine kaydedebilmesi gerekliliği,
- İstasyonlarda malzeme algılama, algılanan verinin robota gönderilerek kablosuz haberleşme işleminin sağlanması ve robotun üretim hattındaki işleyişi durdurmadan aynı anda birden fazla istasyondan gelen sinyalleri işleyebilmesi ve tepki verebilmesi durumu bizim bilgimiz dahilinde bu zamana kadar gerçeklenmemiş bir uygulama olduğundan çok sayıda deneme gerekliliği ve istasyonların birbirinden farklı niteliklere sahip

olduğunun tanımlanması ve programlanabilmesi.

- Robotun çizgi takip işlemini gerçekleştirirken hız ve açısal konumunu ayarlayabilmesi.
- Robot endüstriyel uygulamalarda kullanılacağıdan ekstra ücretler ödemeden açık kaynak bir programlama dili ile programlanabilir olması gerekliliği.

1.4. Değerlendirme

Bu tezin 2. Bölümünde endüstriyel çizgi izleyen robot prototipinin sistemsel olarak tasarlanması için gerekli aşamalar oluşturulmuştur. Öncelikle çizgi takip yöntemleri birbirleriyle karşılaştırılarak analiz edilmiş ve prototip robot için uygun yöntem belirlenmiştir. Sürüş yöntemleri karşılaştırılarak robota uygun sürüş yöntemi belirlenmiş ve bu yönteme uygun elektronik devre elemanları ve kontrol kartlarında bulunması gereken ihtiyaçlar analiz edilmiştir. Kontrol kartlarındaki ihtiyaçların belirlenmesi ile birlikte sisteme uygun elektronik devre elemanları karşılaştırılarak kullanılacak kontrol kartları belirlenmiştir. Kontrol kartlarının özellikleri ve kullanım durumları bu bölümde incelenmiştir.

Tezin 3. Bölümünde prototip robotun üretim aşaması anlatılmıştır. Robotun çalışma prensibi tasarlanmış ve oluşturulmuştur. Çalışma prensibi içerisinde robotun mekanik ve elektronik sistemlerinin birlikte çalışması nedeniyle mekanik yapı analiz edilerek 3 boyutlu ortamda tasarım şekli hazırlanmış ve üretim aşamalarından bahsedilmiştir. Mekanik kısım tamamalandıktan sonra robotun elektronik yapısı oluşturulmuştur. Robotun elektronik tasarım ve üretim aşamaları ile paralel gelişen sistem programlama işlemi prototip robotun üretimi başlığı altında analiz edilmiş ve tamamlanmıştır. Sistem programlanması başlığı altında çizgi takip işlemi, istasyonlar ve robotun haberleşme işlemi, otomasyon işlemi ve istasyonların programlanması işlemlerinden bahsedilmiştir.

2. SİSTEM TASARIMI

2.1. Yaygın Olarak Kullanılan Çizgi Takip Yöntemleri

Günümüz sanayiinde yaygın olarak kullanılan çizgi takip yöntemleri incelendiğinde karşımıza genellikle ihtiyaca yönelik veya robotun kullanılacağı çevre koşullarına uygun şekilde (Thomson, 2001) tasarımlar meydana getirilmiştir. Temel olarak incelendiğinde 4 takip yönteminin yaygın olarak (Borenstian, 1996) kullanıldığını görüyoruz. Bunlar,

- CCD Kamera Yöntemi ile Önceden Tanımlı Çizgi Takibi
- Manyetik Algılama Yöntemi ile Önceden Tanımlı Çizgi Takibi
- Ultrasonik Algılama Yöntemi ile Önceden Tanımlı Çizgi Takibi
- Optik Algılama Yöntemi İle Önceden Tanımlı Çizgi Takibi

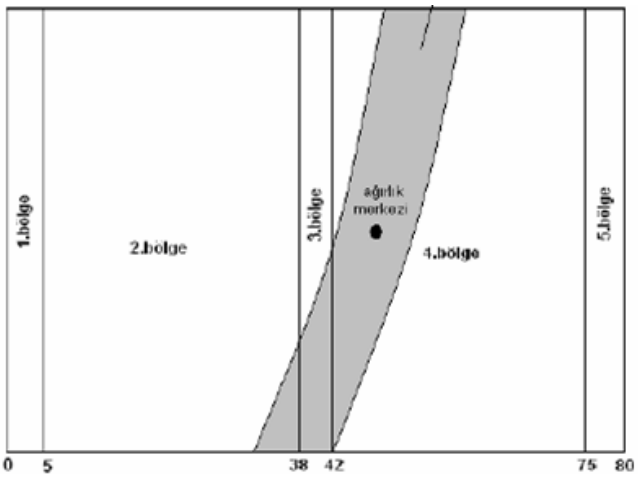
Bu yöntemlerin temelinde öncelikli olarak belirli seçim bilgileri mevcuttur ancak ana kriter oluşturulacak olan sensör yapılarının çalışma koşullarına uyum sağlayabilmesidir. Örnek olarak çalışma yüzeyi üzerinde manyetik alanların değişkenlik gösterdiği bir bölgede manyetik algılama yöntemi kullanılırsa alınan sonuçlarda hata paya yüksek olur. Aynı şekilde çalışma yüzeyinde yüksek miktarda ışık ve yansıma mevcutsa kontrast farkından yararlanan (Wolfbeis, 2004) optik algılama sensörü görme işlemini tam olarak yapamayacaktır. Belirli veriler dikkate alınarak çizgi takip işleminin yöntemi belirlenmektedir. Bunlar;

- Robotun kullanılacağı ortam koşulları,
- Robotun çalışma esnasındaki konumlandırma hassasiyetleri,
- Maliyet,
- Robotun istihap haddi,
- Bakım onarım aşamasında bakım sıklığı ve parça tedarik kolaylığı,
- Sağlamlık,
- Çevre koşullarından bağımsız çalışma,
- İş Güvenliği,

- Yazılım Geliştirme.

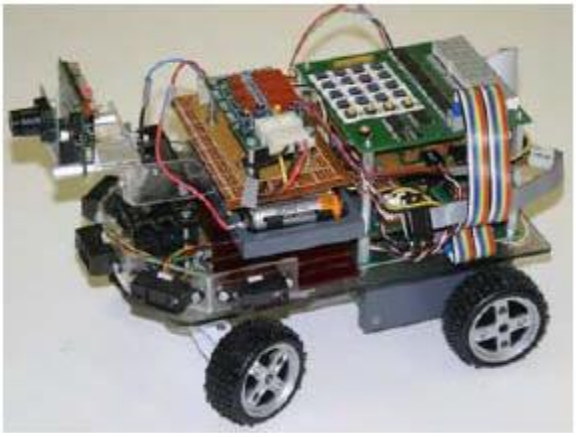
2.1.1. CCD Kamera Yöntemi ile Ön Tanımlı Çizgi Takibi

CCD kamera yöntemi ile çizgi takip işlemi temel olarak diğer tüm çizgi takip işlemlerinde olduğu gibi çizgi ile zemin arasındaki kontrast farkından yararlanılarak robotun kullanılacağı çalışma ortamına bağlı olarak (siyah zemin üzerinde beyaz çizgi veya açık renk zemin üzerinde koyu çizgi) gerçekleştirilmektedir. Bu yöntemde CCD kamera zemin üzerindeki çizgiyi algılayacak şekilde konuşlandırılmıştır. Bu durumda önemli olan nokta şekil 2.1.'de görüldüğü gibi CCD kameranın hassasiyetinden yararlanılarak zemin üzerindeki çizginin ortasına veya çizgiyi nesne olarak düşünürsek ağırlık merkezine kameranın odaklanmasını sağlamaktır (Dupuis et al., 2006). Odaklanma işleminin çizgi takip işlemindeki en önemli rolü ise hassasiyet ve toleranslardır. Bu odaklanma işlenmi esnasında çizgi takip işleminin gerektirdiği hassasiyet de oldukça önemlidir (http://www.seattlerobotics.org/encoder/200011/LineDetect2.htm). Hassasiyetin önemli olmasının nedeni çizgi takip işlemini sağlayacak olan CCD kamera sistemi odaklandığı çizginin ağırlık merkezinden uzaklaştıkça bağlı bulunduğu mikroişlemcinin algılayabileceği voltaj değişiklikleri (Özdemir, 1996) oluşturmasıdır. Bu sayede motorlar robotu merkezde tutabilmek için sağa ve sola robotun ön kısmında bulunan CCD kameranın hassasiyeti ve algılama yeteneği doğrultusunda manevralar yapmaktadır (Dupuis et al., 2006).



Şekil 2.1. Kameranın kontrast farkından yolu algılaması

Robotun hassas hareketler üzerine çalışması gereken durumlarda ise kameranın konuşlandırıldığı açı oldukça önemlidir (Özdemir, 1996). Kamera açısı Şekil 2.2.'de gösterildiği gibi robotun manevrayı hemen algılayarak döneceği şekilde ya da virajları önceden algılayarak yumuşak manevralarla döneceği şekilde oluşturulabilir, örneğin manevra hassasiyeti ve taşınan yükün önem derecesine göre robotun CCD kamerasını çizgiyi 50 mm önceden görecek şekilde belirli açı ile monte edilirse; robot dönüş manevrasını yapmadan 50mm önce çizginin kavislendiğini, farkedecek ve dönüş işlemi için gerekli yavaşlama işlemini voltaj değişikliği ile bu noktada başlatabilecektir. Eğer çok daha hassas ve önem derecesi yüksek malzemeler taşınıyorsa robotun üzerindeki CCD kamera yaklaşık 500 mm önceden çizgiyi algılayabilecek şekilde belirli açı ile konumlandırılmalıdır. Böylece robot bu konuma geldiğinde ön tanımlı rotası üzerindeki değişikliği farkedecek ve voltaj değişikliği ile hızını azaltarak dönüş noktasına geldiğinde istenilen manevrayı gerçekleştirebilecektir.



Şekil 2.2. CCD Kameranın robot üzerine yerleşimi

CCD kamera yöntemi ile ilgili olarak bir değerlendirme yaparsak kullanılan yöntem bakımından çizgi takip işleminin en hassas uygulamısıdır ancak günümüz teknolojisinin kaçınılmaz etkisi ile hassasiyet robot maliyetini oldukça yukarı çekmiştir. Kullanım bakımından CCD kamera kullanılan sistemlerde genel olarak

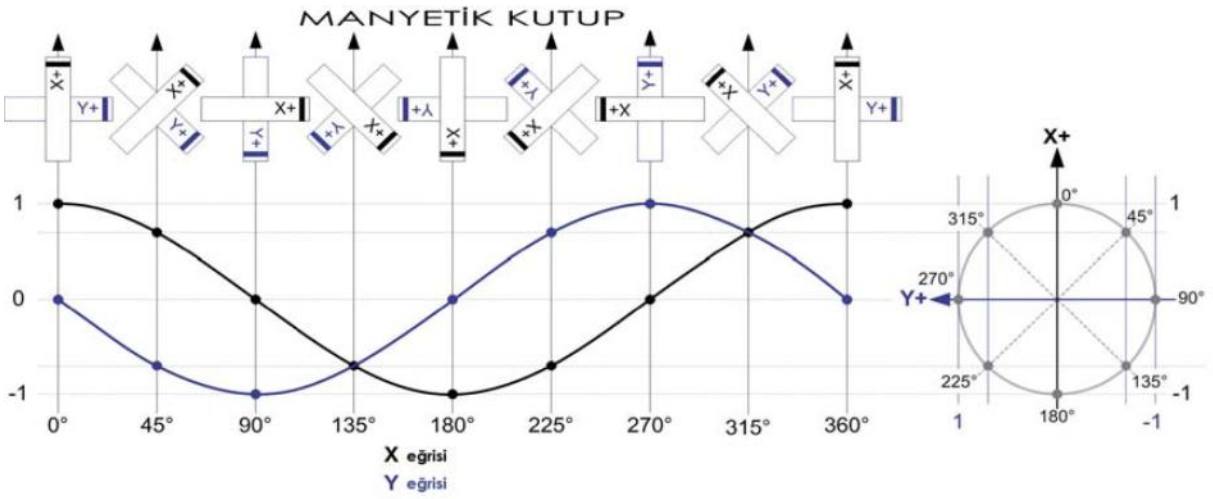
kamera ile takip edilecek çizginin ileri kısımları algılanmış ve güvenlik olarak çizgi takip işleminin yine optik algılama sensörleri ile yapılması sağlanmıştır. Bu noktada önemli olan konu ise sistemimizin hassasiyetini çok küçük toleranslarla belirlemek ve bu hassasiyete uygun olan çizgi takip yöntemini "kalite-maliyet-süre" üçgeninde gerçeklemek gerekir. CCD Kamera yönteminin diğer bir handikapı ise veri işleme yönteminini diğer yöntemlere göre daha fazla nitelik gerektiren işlemcilerle yapılmasıdır (http://electronics.howstuffworks.com/cameras-photography/digital/digital-camera2.htm). CCD kameraların görüntü işleme özelliği ile çalışıyor olması hem işlemciye ekstra yük hem de programlanabilme özelliğini daha zor hale getirmektedir. Görüntü işleme özelliği işlemciye sürekli olarak gelen verinin işlenmesi durumunu gerektirdiğinden ortaya hem müdahale edilmesi zor (Drimer, 2002) hem de yüksek maliyetli elektronik devreler ortaya çıkmaktadır.

2.1.2. Manyetik Algılama Yöntemi İle Ön Tanımlı Çizgi Takibi

Manyetik olarak çizgi takip etme işleminde 2 yöntem kullanılmaktadır. 1. Yöntem Elektronik pusula yöntemi (dünyanın manyetik alanından yararlanılarak yapılan çizgi takip işlemi) 2. Yöntem ise Manyetik algılama sensörleri (hall effect veya eddy current mantığı ile çalışan algılama sensörleri) kullanılarak manyetik çizgi veya şerit (http://literature.rockwellautomation.com/idc/groups/literature/documents/td/1442-td001_-en-p.pdf) izleme yöntemidir.

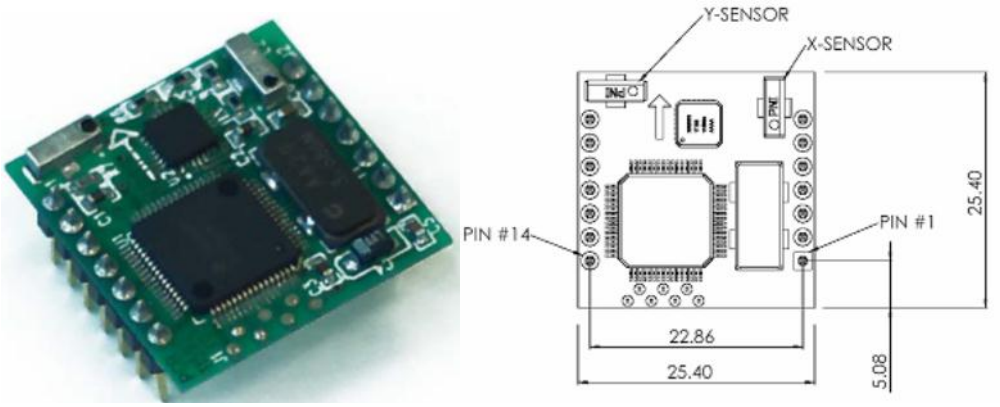
Elektronik Pusula yöntemini incelendiğinde; manyetik alan akı yoğunluğundan faydalanırlar. Manyetik alan akı yoğunluğunu ölçmek için kullanılan araçlara manyetometre (magnetometer) adı verilmektedir. Bu yöntemmanyetik akıyı; voltaja, akıma, freakansa veya bir başka elektronik olarak ölçülebilen bir forma çeviren sensörler kullanmaktadır (http://en.wikipedia.org/wiki/Magnetometer). Bunlar fluxgate manyetometresi, Hall efekti sensörü, manyeto-dirençsel sensör ve manyeto-indüktif sensör olarak en yaygın kullanım alanı olanlarıdır (Özsoy, 2010) (http://www.birsenyayinevi.com/urun/algilama-ve-olcme-esaslari--prof-dr-sedat-ozsoy.aspx). Birbirine dik olarak yerleştirilmiş iki adet manyetometre yön tayini için elektronik pusulalarda kullanılabilir. Sensör, ölçülecek manyetik alan ile paralel olduğunda maksimum çıktıyı verecektir. Dik olduğunda ise sensör çıktı

vermeyecektir. Manyetik alanlar, devredeki elektrik akımından (elektromanyetizma) veya benzer tertibatlarla üretilir. Robotun motorunda kalıcı mıknatıslar bulunur ve robotun konstrüksiyon malzemesinde manyetize olmuş metal bulunur. Aygıtlar, yer altı boruları, kanallar gibi robota göre dış ögeler, kendi alanlarına sahip olabilirler (Ripka, 2001) ve lokal olarak dünya'nın alanının doğrultusunu değiştirebilir dengeleyebilir veya değiştirebilirler.



Şekil 2.3. Açıya göre değişen manyetometre sensör çıktısı ve x-y konumları

Bu noktada manyetik alan etkileri göz önüne alınarak çizgi takip edilmesi istenen bölgede oluşturulacak manyetik alanın takibi oldukça kolaydır. Hall Effect sensöründe de aynı mantıkla oluşturulan manyetik şerit manyetik sensör tarafından algılanmakta ve optik uygulamalarda olduğu gibi birden fazla sensör dizilimi kullanılmaktadır (www.tri-m.com/products/.../**v2xe**_spec.pdf).



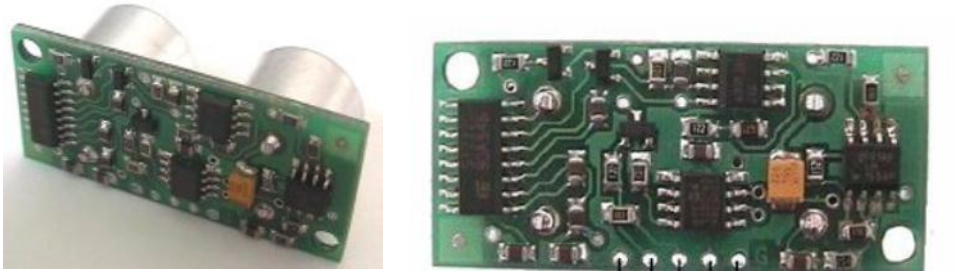
Şekil 2.4. Manyetik alan ölçme modülü V2Xe

Manyetik alan sensörleriyle çizgi takip yöntemlerini değerlendirildiğinde; robotun çalışma ortamları doğrudan sensörün çalışma mantığını etkilemektedir. Çalışma ortamında bulunan manyetik akıya etki edecek tüm araç gereçler, metal ve manyetizma özelliği olan malzemeler sensör ölçümlerinde sapmalara neden olacağından endüstriyel ortamda kullanılmaları uygun değildir (http://web2.deu.edu.tr/yo/diyyo/okm/MODULES/TERM%20I/T1Module3/DECK/SerimPaker_t1m3d/PUSULA.pdf). Manyetik sensörlerin kullanıldığı çizgi takip eden robotlar genellikle plastik malzemelerin lojistiğinin yapıldığı manyetizma özelliğinden izole edilmiş alanlarda kullanılmaktadır. Hassasiyet bakımından CCD kamera yöntemi ile karşılaştırıldığında daha az hassas olmakla birlikte maliyet bakımından daha az maliyetlidir (http://www.robitshop.com/Manyetik-Sensorler,LA_202-2.html). Manyetik algılama sensörlerinin genel fiyat aralığıda aynı CCD kameralar gibi sensör hassasiyetine bağlı olarak arttığından yaklaşık alım değerleri 20TL ile 300 TL arasında değişmektedir ancak CCD kameralar manyetik algılama sensörlerine göre daha karmaşık elektronik devreler içerdiğinden alım değerleri hassasiyete bağlı olarak 50 TL ile 10.000 TL arasında değişkenlik göstermektedir.

Tedarik kolaylığı bakımından CCD kamera ve manyetik algılama sensörü karşılaştırıldığında, endüstriyel uygulamalarda kullanılcak olan CCD kameraların tedarik edilebilirliği manyetik algılama sensörlerine göre daha zordur. Manyetik algılama sensörlerinin kullanım alanları CCD kameraya göre oldukça fazladır. Endüstriyel uygulamaların birçoğunda uyartım ve sinyalizasyon işlemlerini gerçekleştirmek için kullanılırlar. Bu nedenle manyetik algılama sensörlerinin tedarik edilebilmesi CCD kameralara göre daha kolaydır.

2.1.3. Ultrasonik Algılama Yöntemi İle Ön Tanımlı Çizgi Takibi

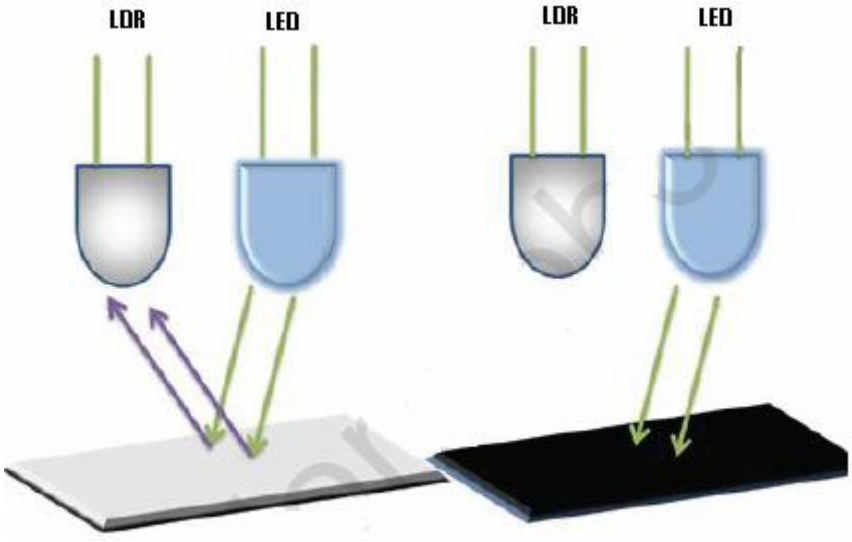
Ultrasonik Mesafe Algılama sensörleri ile çizgi takip yöntemleri incelendiğinde genel olarak çizgi takipten çok ortam koşullarını algılayarak (Wilson, 2005) algıladığı engellerden uzaklaşıp belirli bir rotada veya bulunduğu ortam koşullarına uygun hereket eden robot sistemlerini görmekteyiz. Bir başka deyişle; ultrasonik algılama yöntemi ile doğrudan çizgi izleme işlemi yapmak mümkün değildir. Ultrasonik algılama yöntemi ile çalışan robotların belirli bir rotada devam edebilmesi için genellikle ultrasonik sensörün algılayabilecegi yükseklikte bir rota veya parkur oluşturulur. Sensör bu rotayı algılayarak rotanın konumlandırıldığı alandan belirli mesafede ilerler (Bayram, 2006). Ultrasonik sensörlerin mesafe ölçme ve algılama özelliği kullanılarak gerçeklenen bu özellik sayesinde asıl kullanım amaçları çizgi takip işlemi olmasa bile ultrasonik sensörler çalışma ortamı koşullarının uygun hale getirilmesi ile bu işlemler için kullanılabilmektedir.



Şekil 2.5. Sonar Sensör

2.1.4. Optik Algılama Yöntemi İle Ön Tanımlı Çizgi Takibi

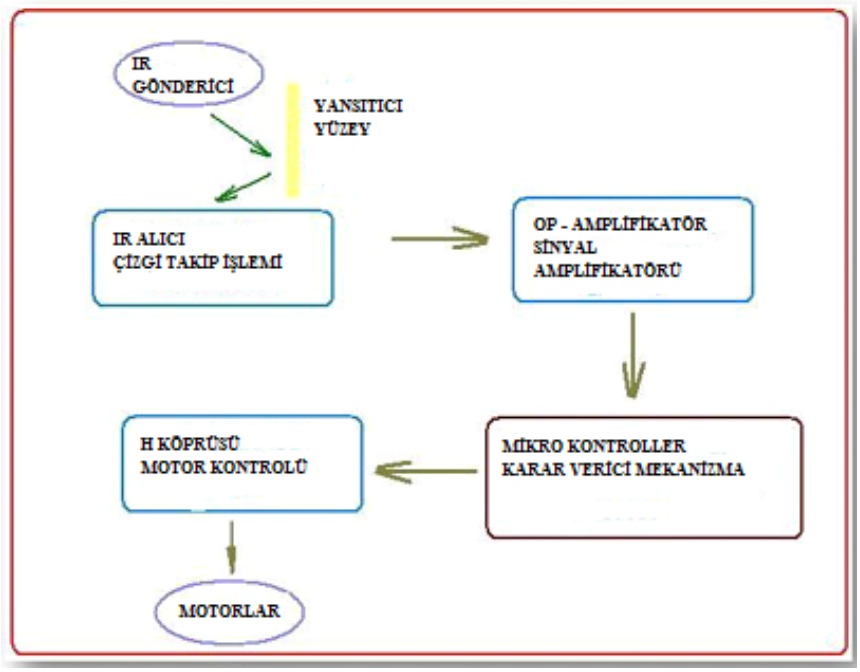
Optik algılama sensörleri ile çizgi takip yöntemlerinde kullanılan temel özellik kontrast farkından kaynaklanan ön tanımlı çizgiyi algılayabilen sensörlerin kullanımıdır (Bansal et al. 2010). Bu sensörler LDR ve LED özellikleri olan ve kontrast farkını bu iki ekipmanı birarada barındırması sayesinde anlayabilen (Wilson, 2005) sensörlerdir.



Şekil 2.6. LDR ve LED ekipmanlarının çalışma prensibi

LED den gönderilen optik sinyaller şekil 2.6.'da görüldüğü gibi siyah zemin üzerinde veya kontrast farkının olmadığı zeminlerde tamamen absorbe edilmektedir. Ancak kontrast farkının olduğu veya siyah zemin üzerinde beyaz çizgilerden oluşan bir parkurda LDR dan gönderilen sinyaller LED tarafından algılanmakta ve bu sayede çizgi takip işlemi gerçekleşmektedir (Wee, 2007).

Optik sensörlerle gerçeklenen çizgi takip etme yöntemlerinde diğer çizgi takip etme yöntemlerinde olduğu gibi alınan verilerin işlenmesi ve motorlara gönderilmesi aynı yöntemle olmaktadır. Şekil 2.7.'de gösterildiği gibi optik algılama sensörlerinden gelen sinyaller veri işleme biriminde yani robotun merkez işlemcisinde işlenmektedir. Sensörlerden gelen veriler robotun hareket yapısını oluşturacağından işlemciden motorlara hareket vermek üzere motor sürücü devresine merkez işlemci birimi tarafından gönderilmektedir.



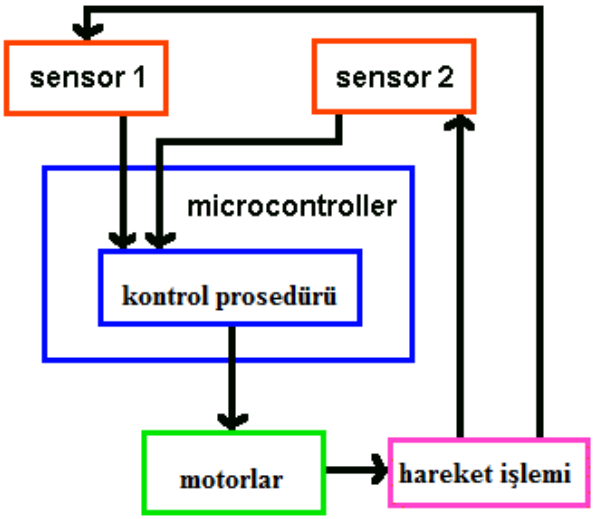
Şekil 2.7. Optik sensör ile Çizgi takip eden robotta ana bölümler

Optik sensörler ile çizgi takip işlemi incelendiğinde; malzeme bulunabilirliği ve tedariği bakımından oldukça uygun, düşük maliyetli, endüstriyel çalışma koşullarından etkilenmeyen ancak hassasiyet bakımından ince çizgi algılamada düşük bir performansa sahip bir sensör yapısı görülmektedir. QTR sensörlerinin (http://www.pololu.com/catalog/product/961) algılama mesafeleri led girişine bağlanan direnç değerleriyle değişkenlik göstermekte ve minimum 1mm maksimum 10 mm ile tanımlanmaktadır bu da çizgi algılama hassasiyetini belirli seviyede tutmaktadır. Endüstriyel ortamlarda çizgi takip etme işleminde çizgi kalınlıklarının istenildiği boyutlarda ayarlanabilir olması bu dezavantajla karşı karşıya kalma durumunu oldukça düşürmektedir.

Şekil 2.8. Optik algılama sensörleri

2.1.5. Genel Olarak Kullanılan Yöntemlerin Değerlendirmesi

Genel olarak çizgi takip yöntemlerini incelediğimizde her yöntemin kendi içerisinde gerçekleştiği çizgi takip işlemini belirli toleranslar, maliyetler ve teknik süreçler içerisinde yapabildiğini görmekteyiz. Bu noktada incediğimiz 4 yöntemin de aynı ana bölümler içerisinde bu işlemleri gerçekleştirdiğini görüyoruz.



Şekil 2.9. Çizgi Takip Eden Robotların Çalışma Prensibi

Bu prensipte çalışan çizgi takip robotlarının kullanıldığı alan ve işe göre belirli hassasiyetler ve donanımlar belirlenerek kullanılacak ortama uygun endüstriyel çizgi takip eden robot prortotipi geliştirmek mümkündür. Bu noktada üretimi doğrudan etkileyen ve duruşlara sebep olmayacak şekilde ince çizgi üzerinde

gidecek bir çizgi takip robot prototipi için CCD Kamera Yöntemi tercih edilebilir. CCD kamera yönteminde görüntü işleme özelliğinden yararlanılarak ve kamera pozisyonu ayarlanarak çizgi takip işlemi gerçekleştirilebilir. CCD kamera yönteminde karşılaşılabilecek zorluk ise görüntü işleme özelliğinden dolayı programlama zorluğudur. Metal ve manyetik alanların az olduğu endüstriyel alanlarda hızlı tepkime vererek anında dönüşler gerçekleştirmek için manyetik etkili sensörlerle çizgi takip işlemi gerçekleştirilebilir. Manyetik sensörler ile çizgi takip işleminin gerçekleştirildiği uygulamalarda programlama CCD kamera yöntemine göre daha kolay uygulanabilmektedir ancak manyetik alan oluşturabilecek ortamların etkileri uygulamayı zorlaştırmaktadır. Optik algılama yöntemi ile hazırlanan çizgi izleyen prototip robotlarda öne çıkan özellikler programlama ve sistem kurulum kolaylığıdır. Çizgi takip yöntemlerini karşılaştırdığımızda,

ÇİZGİ TAKİP YÖNTEMLERİNİN KARŞILAŞTIRILMASI												
KRİTER	SAĞLAMLIK	ÇEVRE KOŞULLARINDAN BAĞIMSIZ ÇALIŞMA	MALİYET	BAKIM	İŞ GÜVENLİĞİ	MÜDAHALE KOLAYLIĞI	DONANIM TEDARİK KOLAYLIĞI	YAZILIM GELİŞTİRME	TOLERANS	HASSASİYET	Σ+	Σ−	Σ0	Σ
KATSAYI	3	1	3	3	1	2	2	1	2	2				
CCD	0	-1	0	1	-1	-1	0	1	1	1	8	-4	3	4
ULTRASONİK	0	0	0	1	-1	-1	1	0	1	-1	7	-5	2	2
MANYETİK	-1	0	1	1	0	1	0	0	0	0	8	-3	6	5
OPTİK	1	1	1	1	0	1	1	0	-1	0	14	-2	4	12

Tablo 2.1. Çizgi takip yöntemlerinin karşılaştırılması

Tablo 2.1. de çizgi takip yöntemleri arasında yapılan karşılaştırma analiz çalışmasını incelediğimizde özellikle optik algılama yönteminin diğer yöntemlere göre avantajlı olduğu ortaya çıkmaktadır. Bu noktada tüm çizgi takip yöntemlerinin karşılaştırıldığı özellikler incelendiğinde, sağlamlık, maliyet, müdahale edilebilirlik ve donanım tedarik kolaylığı başlığı altında optik algılama yöntemi diğer yöntemlere göre avantajlıdır. Ancak tolerans ve hassasiyet konularında diğer yöntemlere göre dezavantajlıdır. Bu noktada bina içi robot uygulamalarında ve endüstriyel alanlarda kullanılacak olan robotlarda çizgi takip işlemini gerçekleştirecek seviyede hassasiyete ihtiyaç duyulmaktadır ancak büyük ölçekli alanlarda ve açık alan uygulamalarında otomatik yönlendirilmiş robotlarda GPS konumlama ve CCD kamera yöntemi uygulanabilmektedir(Khwang et al., 2000). Tolerans başlığı altında çizgi takip yöntemlerini karşılaştırınca optik algılama yönteminin kamera yöntemine göre tolerans aralığının daha geniş olduğu görülmektedir. Bu durum optik algılama yönteminde kamera yöntemine göre düşük hassasiyetlerle işlem yapılabildiğini göstermektedir.

Robotun çizgi takip işlemini gerçekleştirdiği esnada sürekli olarak veri işleme olayı gerçekleştirildiğinden kamera ile çizgi takip yönteminde kullanılacak olan merkezi işlem biriminin kapasitesi yüksek olmalıdır. Bu durumda merkezi işlem biriminin kapasitesinin artmasıyla birlikte maliyeti de artmaktadır. Ayrıca kamera ile çizgi takip yönteminde yazılım geliştirme durumu optik algılama yöntemine göre daha karmaşık bir yapı içermektedir.

Bu tezin amacı; endüstriyel ortamlarda kullanılabilecek çizgi izleyen bir robot prototipi gerçekleştirmek olduğundan mevcut şartlarda bu prototipi oluşturmak için kullanılacak yöntem “Optik Algılama” yöntemi olarak seçilmiştir.

2.2. Robot Sürüş Yönteminin Belirlenmesi

Robotun kartezyen kordinatlarda x ve y yönünde ilerlemesine imkan veren mekanik sistemine, robotun hareket sistemi adı verilir (Jones, 1998). Mobil robotlar bir başka deyiş ile hareketli robotlar, çeşitli alanlarda hareket etmek için tasarlanırlar. Hareket edecekleri ortamlar hava, deniz, kara ve uzay gibi birbirine göre çok farklılık gösterdiklerinden, hareketli robotlar da kendi içlerinde hareket sistemleri bakımından çok farklılıklar göstermektedir. Karada hareket eden robotlar, hareket sistemlerine göre;

- Tekerlekli,
- Paletli,
- Bacaklı,
- Eklemsi yapılı,
- Temel hareket sistemlerinin kombinasyonu

şeklinde sınıflandırılabilinir (Özen, 2000). Bu yöntemler arasında bir karşılaştırma yapılırsa; robot hareket yöntemlerinin bina içi uygulamalarında tekerlekli robotların diğer hareket yöntemlerini kullanan robotlara göre üstünlüğü göze çarpmaktadır. Bu yöntemlerin karşılaştırılması esnasında kullanılan karşılaştırma verilerinin "robotların bina içerisinde kullanılacağı" zemininine oturtularak hazırlanması gerekmektedir. Örneğinarazi şartlarında veya engebeli bir ortamda kullanılacak bir robot için paletli hareket yöntemi uygun olacaktır. Endüstriyel ortamlarda kullanılacak, malzeme taşıma kapasitesi yüksek olacak ve otonom çalışma yapısı nedeniyle hareket sistemi hızlı olacak bir robot prototip için uygun yöntem tekerlekli hareket yöntemi olabilir. Endüstriyel ortamlarda mevcut uygulamalar incelendiğinde bant tipi üretimlerde lojistik operasyonları üretimin gerçekleştiği "bant kenarında" gerçekleşmektedir(Pechoucek et al., 2006). Günümüz endüstrisi bant kenarlarına malzeme ve hammadde ikmali yapabilmek için yine bir tekerlekli sürüş yöntemine sahip ancak otonom çalışma niteliğine sahip olmayan forkliftleri kullanmaktadır. Bu durum kullanılacak olan robotun bant hızına uygun hızda parça besleyebilmesi ve yük kapasitesini kaldırabilmesi adına tekerlekli sürüş yönteminin uygun olduğunu göstermektedir.

ROBOT HAREKET YÖNTEMLERİNİN KARŞILAŞTIRILMASI											
ÖZELLİKLER	HAREKET KABİLİYETİ	MALZEME TAŞIMA KAPASİTESİ	STABİL VE DENGELİ ÇALIŞMA	SAĞLAMLIK	MALİYET	HASSASİYET	MÜDAHALE KOLAYLIĞI	BAKIM	ÇEVRE KOŞULLARINDAN BAĞIMSIZ ÇALIŞMA	YÜKSEKLİK (Z) EKSENİNDE ÇALIŞMA	Σ
KATSAYI	4	2	3	3	3	2	2	2	1	1	
PALETLİ	-1	1	1	1	0	0	0	-1	1	-1	2
TEKERLEKLİ	1	0	1	0	1	1	1	1	0	-1	15
BACAKLI	1	0	0	-1	-1	0	0	0	1	1	0
EKLEMSİ YAPILI	1	0	0	-1	-1	0	0	0	1	1	0

Tablo 2.2. Robot hareket yöntemlerinin karşılaştırılması

Tablo 2.2. de kapalı alanlarda sürüş yöntemleri incelendiğinde hareket kabiliyeti yüksek olan robot türü “Tekerlekli Robotlar” olarak (Türker, 2005) karşımıza çıkmaktadır. Fabrika içi lojistik uygulamalarının çizgi takip eden robotlarla geliştirildiği ve desteklendiği sanayi uygulamalarında kullanılan bu robotlarda tercih edilen hareket yöntemi “Tekerlekli Taşıma” (Elizandro et al., 2012) yöntemi olarak belirlenmiştir. Endsütriyel lojistik işlemlerinde kullanılan çizgi takip eden robotların yapısı incelendiğinde (http://www.agvindustrial.com), (http://www.mhi.org/agvs) karşımıza yine tekerlekli sürüş yöntemi çıkmaktadır. Kapalı alanlarda gerçekleştirilen endüstriyel uygulamalarda çizgi takip eden robotlara belirli bir rota planlaması gerekliliği; (Petrinec et al., 2003) çizgi takip eden robotlarda bu rotayı gerçekleştirme yetkinliğini; çalışma ve kullanım amacı haline getirmiştir. Çizgi takip eden robotlar rotasını takip etme işlemi esnasında malzeme taşıma işlemini de gerçekleştirmek için (Ulgen et al., 1990) tasarlanırlar. Endüstriyel uygulamalarda römork tipi veya vagon tipi çekme şeklinde gerçekleştirilen (Sezen, 2003) malzeme taşıma işlemine uygun çekme yöntemi tekerlekli sürüş (Dutt , 1991) yöntemidir. Robotların hareket yöntemlerinin karşılaştırma verileri incelendiğinde hareket kabiliyeti, dengeli çalışma, maliyet, hassasiyet, müdahale kolaylığı ve bakım konularında tekerlekli robotların diğer sürüş yöntemlerine göre prototip üretimine daha uygun olduğu ortaya çıkmaktadır.

Malzeme taşıma kapasitesi, sağlamlık ve çevre koşullarından bağımsız çalışma konularında ise tekerlekli sürüş yöntemi paletli sürüş yöntemine göre dezavantajlıdır. Tekerlekli sürüş yönteminin endüstriyel alanlarda bina içi kapalı uygulamalarda kullanılacak robot prototipi için uygulanacağı düşünüldüğünde çevre koşulları paletli sisteme göre uygundur anca malzeme taşıma kapasitesi düşüktür. Bu noktada paletli yöntemin dezavantajı malzeme taşıma kapasitesi yüksek olmasına rağmen manevra kabiliyeti ve mekanik tasarımının bina içi uygulamalarda kullanışlı olmamasıdır.

Tekerlekli sürüş yöntemi z ekseninde çalışma kategorisinde bacaklı ve eklemsi yapılı robotlara göre dezavantajlıdır. Ancak yine çalışma ortamı ve robottan beklenen temel özellik olan malzeme taşıma durumu incelendiğinde paletli ve eklemsi yapılı robotlar bu işlem için uygun değildirler.

Tekerlekli sürüş yöntemine sahip robotların düz yüzeylerde çekme kabiliyeti ve performansı yüksektir. Pürüzlü arazi şartlarında diğer paletli ve eklemsi yapılı hareket sistemleri öne çıksa da, tekerlekli sistemlerin basitliği, araştırılması ve geliştirilmesinde masraflarının az olması daha çok tercih edilmesine neden olmaktadır. Mobil robotlar hareket sistemlerine göre sınıflandırıldıkları gibi, tekerlekli robotlar da, kendi aralarında sürüş yöntemlerine göre ayrılırlar (Türker, 2005).

2.3. Tekerlekli Robotlar için Sürüş Yöntemlerinin İncelenmesi

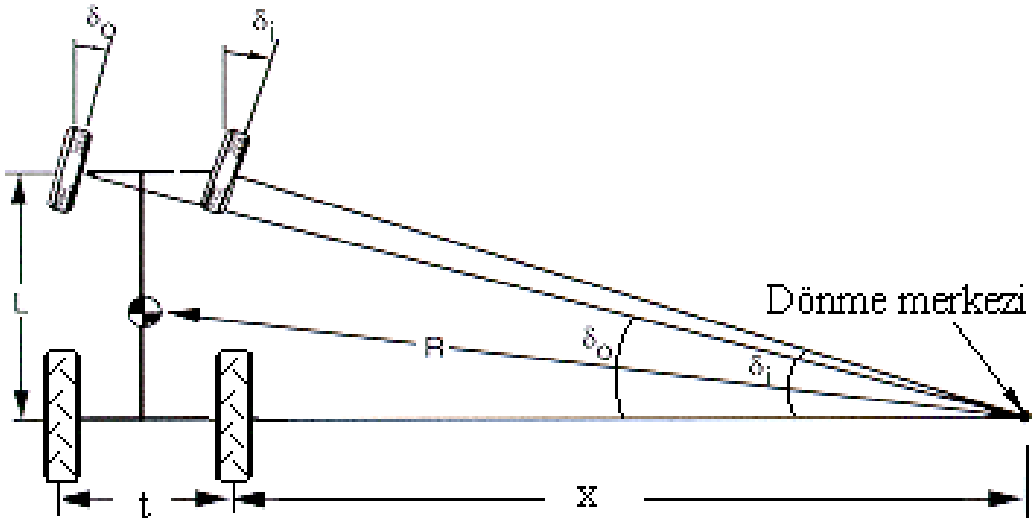
Tekerlekli robotların; hareket edecekleri ortam ve ortam şartları, sahip oldukları güç depolama birimleri, yapacakları iş ve kontrol mekanizmaları göz önüne alınarak uygulamaya en uygun hareket sistemleri (Jones, 1998) . belirlenmelidir. Görüldüğü üzere hareket sisteminin belirlenmesinde rol oynayan bir çok etken vardır. Hareket sistemi seçimi sırasında, sistemin genel yapısının yapılmak istenilen işi karşılayacak, pratik ve karmaşık olmaması beklenir.

Mevcut projemizde kullanılmak üzere değerlendirilen ve analiz edilen bazı sürüş teknikleri aşağıdaki verilmiştir.

2.3.1. Ackerman Sürüş yöntemi

Günlük hayatta en çok kullanılan sürüş tekniğidir (arabalarda). Ackerman sürüş tekniği; dönüş sırasında, iç kısımda kalan ön tekerlek, dışta kalan tekerleğe göre, biraz daha keskin bir açı ile döndürülerek, geometriksel olarak ortaya çıkan, lastiğin kayma problemini yok etmek için tasarlanmıştır. Ackerman sürüş yöntemi günümüz taşıtlarında tam anlamı ile hepsinde temel prensipte (Luca, 2009) kullanılmaktadır. Fabrika ve bina içi uygulamalarda ise az tercih (Belpaeme, 2004) edilmektedir. Bunun nedenlerini incelediğimizde Ackerman sürüş yönteminde öndeki motor direksiyon görevi görmekte ve yönlendirmeyi sağlamaktadır. Arkadaki motor ise hareketi (Sachdeva, 2011) sağlamaktadır. Direksiyon görevi gören motorun yönlendirme işleminde hassas olması isteniyor ise mevcut dişli sistemi diş aralıkları (Dağdeviren et al., 2001) değiştirilmeli ve servo motor kullanılmalıdır. Ackerman sürüş yönteminin farklı uygulamaları incelendiğinde biçerdöver sistemlerinde hareket motorları ve direksiyon mil sistemi ön tekerlekler

yerine arka tekerleklere dönel hareket vermektedir. Bunun nedeni biçerdöver sistemlerinde kesici kısım ön bölümde bulunduğu için dönel hareketin kısıtlanmasıdır. Bu durumun önüne geçmek ve hareket kabiliyetini arttırmak amacı ile arka tekerlekler ön tekerleklere göre daha küçük hale getirilmiştir ve direksiyon mil sistemi arka dişli sistemine bağlanmıştır. Bu uygulamalar da robotun maliyetini oldukça arttırmaktadır. Ayrıca robotun otonom hareket etmesi istendiğinde Ackerman sürüş yöntemi hem yazılımsal hem de donanımsal olarak programlanma ve tasarlanma zorluğu (Dusek et al., 2011) göstermektedir.



Şekil 2.10. Ackerman sürüş yöntemi

Şekil 2.10.'a göre ackerman sürüş yöntemi ile ilgili olarak dönüş işlemleri esnasında iç ve dış tekerleklerde meydana gelen açısal sapma şu şekilde tanımlanabilir.

$$\tan \delta i = \frac{L}{X} \tag{2.1}$$

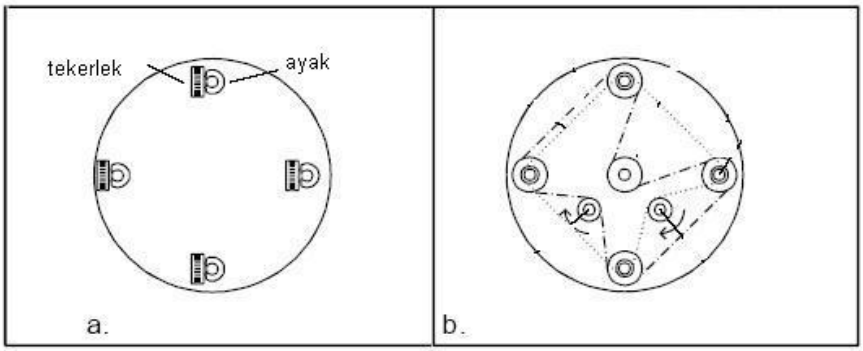
$$\tan \delta 0 = \frac{L}{x+t} \tag{2.2}$$

$$\frac{t}{L} = \frac{1/tan\delta 0}{1/tan\delta i} \quad (2.3)$$

$$\Delta\delta = \frac{\delta 0A}{\delta 0} \quad (2.4)$$

2.3.2. Senkron sürüş yöntemi

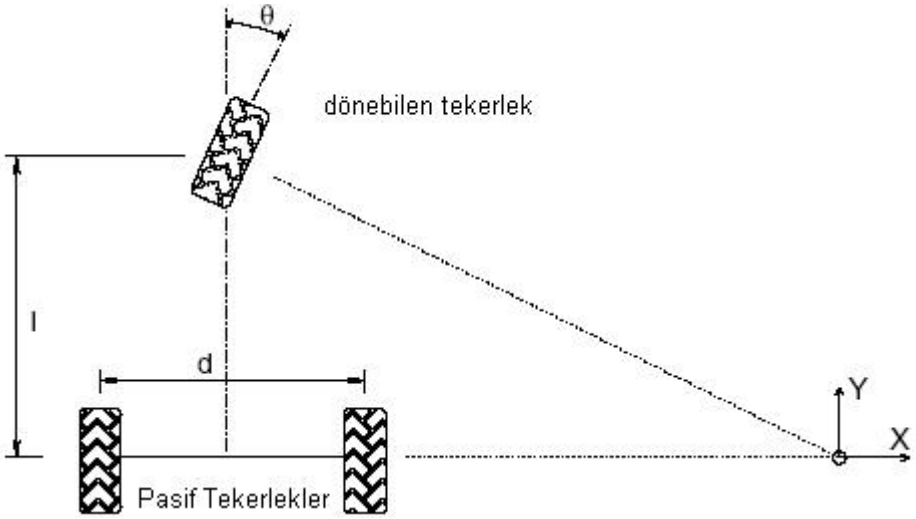
Senkron sürüşte, robotun 3 veya daha fazla tekerleğe sahip olması gerekir. Bu tekerlekler aynı yönde ve aynı hızda dönebilecek şekilde tasarlanmıştır. Tüm tekerlekler aynı hız ve yönde döndüğü için, odometri ve kayma hataları azalmış olur. Kapalı alanlarda bu yüzden tercih edilen bir sürüş tekniğidir.



Şekil 2.11. Senkron sürüş tekniği a. Alttan görünüş. b. Üstten görünüş

Senkron sürüş yöntemi incelendiğinde şekil 2.11.'de gösterildiği gibi hareket kontrol sistemi üstte konumlandırılmıştır tekerlek sayısı kadar hareket iletim makarası, merkezde 1 adet kayış gergi ve motorun buluduğu merkez makara mevcuttur. Bu makaralar 2 adet destek makarası ile desteklenebilir ya da bu destek makaralarına motorlar akuple edilebilir (Jones, 1998) . Senkron sürüş yöntemi kapalı alanlar için maksimum hareket kabiliyeti sağlamaktadır ancak mekanik olarak tasarımlarının zor oluşu, kayış mekanizmalarının ve tekerlek mekanizmalarının yük taşıma işlemleri için çok uygun olmaması ve otonom çalışma yöntemlerinde zor programlanabilirliği nedenleri ile projemizde tercih edilmemiştir.

2.3.3. Üç tekerlekli sürüş yöntemi



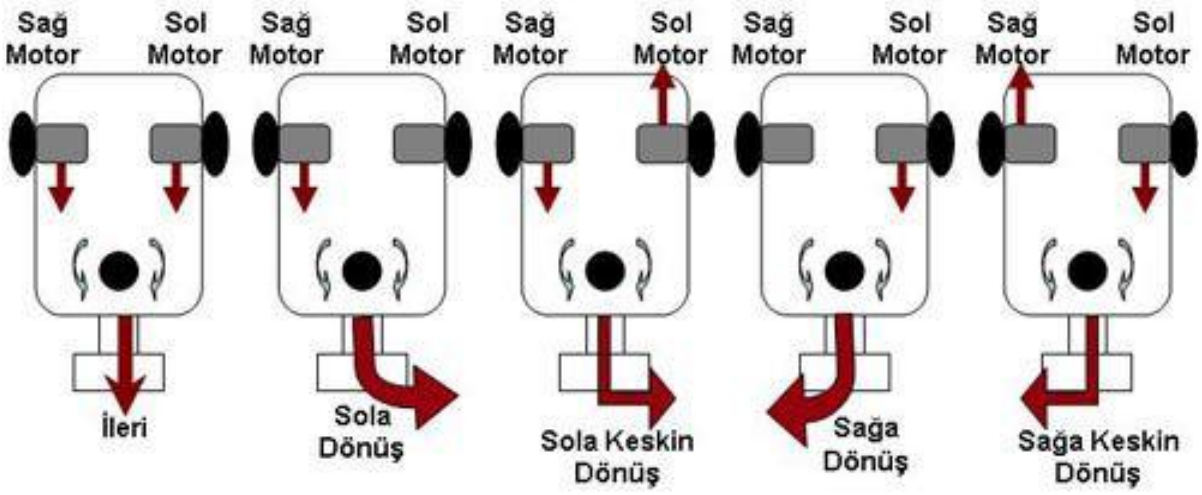
Şekil 2.12. Üç tekerlekli sürüş yöntemi

Üç tekerli sürüş tekniğinde, şekil 2.12.'de görüldüğü gibi ön teker sağa sola dönüşü sağlar arka tekerler ise dönüş işlemlerinde kullanılmaz. Basitliği nedeni ile otonom hareket eden araçlarda sıkça kullanılır. Bu sürüş yönteminde ön tekerlek sarhoş teker veya roll on teker yapısına sahip şekilde monte edilir (Jones, 1998) . ve arka 2 tekerleğe motor akuple edilerek hareket sağlanması halinde robotun hareket kabiliyeti oldukça artar ve bu yöntem diferansiyel sürüş yöntemine dönüşür.

2.3.4. Diferansiyel sürüş yöntemi

Diferansiyel sürüş tekniği, yabancı literatürde genellikle yanlış olarak "differential drive system" olarak kullanılmaktadır. "differantial drive system" otomotiv mühendisliğinde, robotlarda kullanılandan farklı olan bir çeşit sürüş sistemine karşılık gelmektedir. Robotik uzmanları bu sürüş tekniği için "differential steering system" Kullanmaya özen göstermektedirler (Lucas, 2001). Bina içi gezgin robot uygulamalarında en sık kullanılan sürüş tekniklerinden biridir. Diferansiyel sürüş, günlük hayatımızdan da alışık olduğumuz bir tekniktir. Tekerlikli sandalyelerde uygulanan mantık ile ayınıdır. İki birbirinden bağımsız teker aynı dingil üzerine oturtulmuştur. Tekerler birbirinden ayrı hızlarda dönebildikleri için; iki tekerlek aynı hızda döndüğü takdirde düz ilerleme hareketi gerçekleştirilir. Bu yöntem dar alanda manevra kabiliyetini arttırmaktadır. Diferansiyel sürüşte robotun iki tarafında bir

birinden bağımsız hareket ettirilen iki motor bulunur. Bu motorların bağlı olduğu tekerlekler dışında, dengeyi sağlamak amacı ile ön ve arka kısımlara kastor (castor) tekerlekler veya roll on tekerler (Jones, 1998) . konulabilir. Birbirinden bağımsız iki ayrı motora bağlı yan tekerlekler ile robotun yönü ve hızı tayin edilir. Tekerlekler aynı yönde ve aynı hızda hareket ettiği takdirde, robot düz bir şekilde ilerleyecektir. Ve bu hız aşağıda denklemlerde verileceği gibi sağ ve sol motor hızlarının toplamının yarısı kadar olacaktır. Sağ ve sol motor hızlarının farklı değerlerde olması durumunda ise, robot hızı robotun orta noktasına göre yine sağ ve sol motorların toplamının yarısına eşit olduğu halde, robot yönünde değişiklik olacaktır. Bu yön değişimi şekil 2.13.'de görüldüğü gibi daha az dönen motor yönündedir ve iki motor arasındaki hız farkı ile doğru orantılıdır.



Şekil 2.13. Diferansiyel Sürüş yönteminde motorların hareket verme prensibi

Yukarıda bahsetmiş olduğumuz sürüş yöntemlerini hareket edecekleri ortam ve ortam şartları, sahip oldukları güç depolama birimleri, yapacakları iş ve kontrol mekanizmaları göz önüne alınarak değerlendirip gerekli analizler yapıldığında ortaya çıkan karşılaştırma tablosu bize şu sonuçları vermektedir. Diferansiyel sürüş yöntemi bizim prototip robotu oluşturmak için ideal kapalı alanlarda kullanılmak üzere (Belpaeme, 2004) model olabilir. Karşılaştırma tablosunda yapmış olduğumuzu değerlendirme indisleri arasında en efektif sonuçlara ulaştığımız yöntem diferansiyel sürüş yöntemidir.

.

ÖZELLİKLER	KAT SAYI	ACKERMAN SÜRÜŞ YÖNTEMİ	SENKRON SÜRÜŞ YÖNTEMİ	3 TEKERLEKLİ SÜRÜŞ YÖNTEMİ	DİFERANSİYEL SÜRÜŞ YÖNTEMİ
HAREKET KABİLİYETİ	4	0	1	-1	1
MALZEME TAŞIMA KAPASİTESİ	2	1	1	0	0
STABİL VE DENGELİ ÇALIŞMA	2	1	1	0	0
SAĞLAMLIK	3	1	0	0	0
MALİYET	3	0	-1	1	1
KONTROL SİSTEMLERİNİN KARMAŞIKLIĞI	3	-1	-1	0	1
HAREKET SİSTEMLERİNİN KARMAŞIKLIĞI	3	-1	-1	1	0
BAKIM	2	0	-1	1	1
ÇEVRE KOŞULLARINDAN BAĞIMSIZ ÇALIŞMA	1	-1	1	0	1
$\sum$		0	-2	4	13

Tablo 2.3. Robot sürüş yöntemlerinin karşılaştırılması

Tablo 2.3. de incelediğimiz yöntemler arasında tez için prorotip robotu oluşturmakta kullanılacak uygun yöntem diferansiyel sürüş yöntemi olarak ortaya çıkmaktadır. Prototip robotun hareket veren motor bölümü haricindeki noktalarda bulunan destek tekelerinin yapısı ve pozisyonlarının sürüş yöntemine etkileri incelenmiştir. Diferansiyel sürüş yönteminde hareketi sağlayacak olan 2 adet tekerlek ve bu tekerleklere bağlı 2 adet motor bulunmaktadır. Bu motorlar farklı hızlarda ve farklı yönlerde hareket ederek robotun dönme kabiliyetini oluşturmaktadırlar. Bu iki tekere ek olarak kastor veya roll on teker denilen destek tekerleri ile sistem stabil hale getirilmektedir. Bu iki teker arasındaki farkları ve benzerlikleri incelediğimizde ortaya çıkan sonuçlar (Jones, 1998) ;

- Kastor tekerlekler ve roll on tekerler her yöne dönebilen mobilya tekerlekleri olduğu için, robotun hareket yönünde bir etkileri olmaz.
- Kastor tekerler yapısal olarak serbest dönebilen bir mile bağlı olduklarından dönme işlemini stabil şekilde gerçekleştiremezler ve fuleli dönüş meydana gelir. Roll on tekerler ise mekanik yapılarından dolayı herhangi bir mile bağlı olmadan her yöne istenilen dönüşü gerçekleştirebilirler.
- Kastor tekerlerin yük taşıma kapasitesi roll on tekerlere göre daha düşüktür.
- Kastor tekerler ve roll on tekerlerin mekanizmalarda bağlandığı yerler oldukça önemlidir. Diferansiyel sürüş yönteminde tekerleklere bağlı 2 motor hareketi sağladığı için motorların önde olması durumunda dönüşlerde aracın arka tarafında savurma meydana gelmektedir. Ayrıca yük taşıma işlemi robotun arka tarafında gerçekleşeceğinden motorların arka tarafta bulunması yükün taşıma işlemini stabil hale getirecektir.

Mevcut analiz ve karşılaştırmalarında ortaya çıkardığı gibi diferansiyel sürüş yöntemi tezimizde kullanacağımız yöntem olarak belirlenmiştir. Bu noktada diferansiyel sürüş yöntemi şu şekilde matematiksel olarak açıklanabilir;

Vr(t) = Sağ tekerleğin doğrusal hızı

Vl(t) = Sol tekerleğin doğrusal hızı

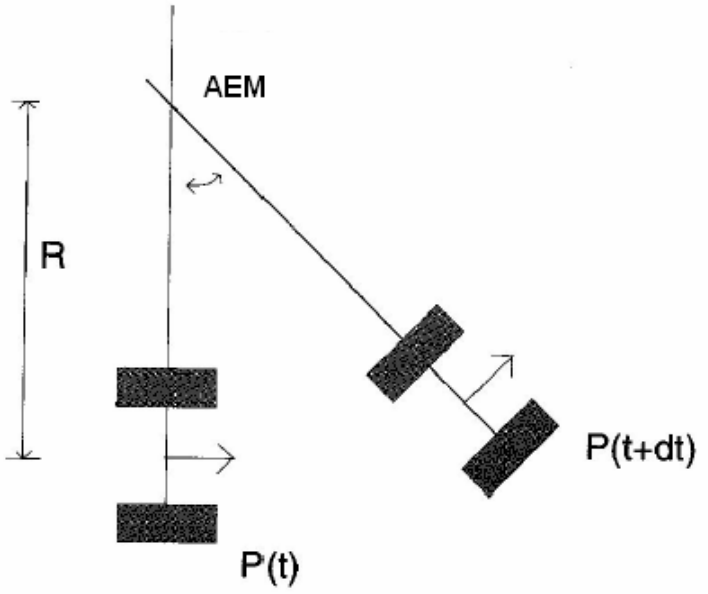
Wr(t) = Sağ tekerleğin açısal hızı

Wl(t) = Sol tekerleğin açısal hızı

L = iki tekerlek arasındaki uzaklık miktarı

R = Robot yörüngesinin anlık eğrilik derecesinin yarıçapı (tekerlerin bağlı bulunduğu şasenin orta noktasına göre)

AEM = Anlık eğrilik merkezi.

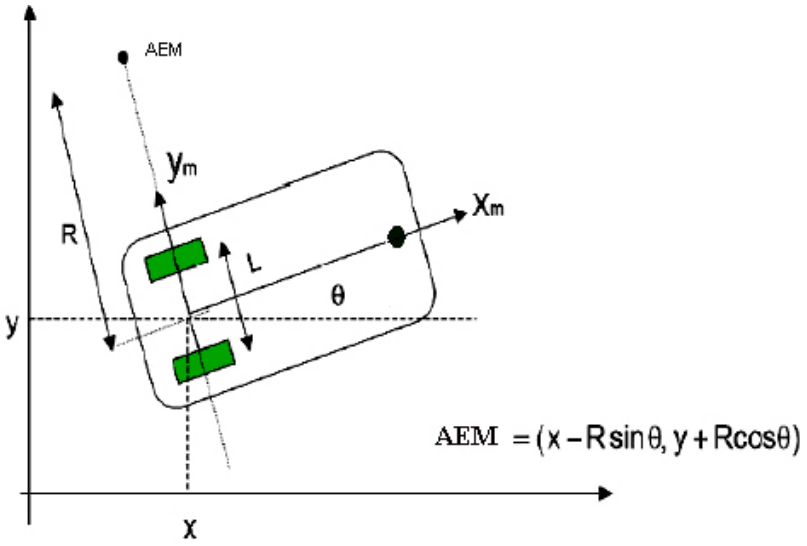


Şekil 2.14. Diferansiyel Robot İçin anlık eğrilik merkezi

Sağ ve sol tekerlek arasında her hız farkı olduğunda, bir yöne göre anlık eğrilik oluşacaktır. Bu eğrilik merkezi, hareketi sağlayan sağ ve sol tekerleklerin bulunduğu şasenin orta noktasına göre alınabileceği gibi, tekerleklere göre de alınabilir (Peri, 2000). Şekil 2.14'e göre anlık Eğrilik Merkezi sağ ve sol tekerleğe göre tanımlanırsa;

$$\text{Sol tekerleğe göre AEM} = \frac{2R-L}{2} \tag{2.5}$$

$$\text{Sağ tekerleğe göre AEM} = \frac{2R+L}{2} \tag{2.6}$$



Şekil 2.15. Diferansiyel robotun kinetik modeli

Şekil 2.15'e göre sistemin denklemleri aşağıda verilmiştir. AEM'e göre robotun açısal hızı şu şekilde verilir;

$$Wr(t) = \frac{Vr(t)}{R+(L/2)} \tag{2.7}$$

$$Wl(t) = \frac{Vr(t)}{R-(L/2)} \tag{2.8}$$

$$W(t) = \frac{Vr(t)-Vl(t)}{L} \tag{2.9}$$

Robotun anlık eğrilik yarıçapı, hareket sağlayan orta tekerleklerin bulunduğu şasenin orta noktasına göre şu şekilde bulunur;

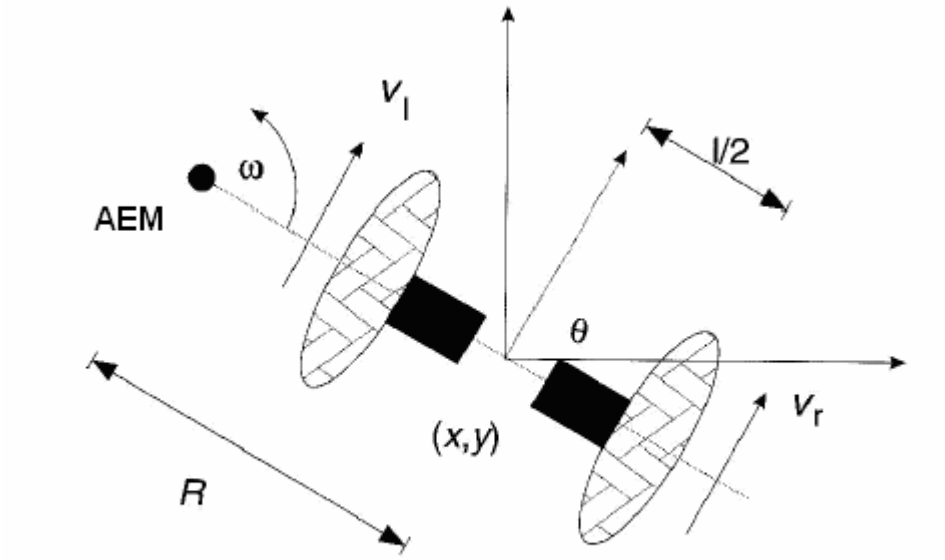
$$R = \frac{L(Vr(t)+Vl(t)}{2(Vl(t)-Vr(t)} \tag{2.10}$$

Robotun doğrusal hızı;

$$V(t) = W(t)R = \frac{Vr(t)+Vl(t)}{2} \tag{2.11}$$

Matris olarak denklemlerin gösterimi;

$$\begin{bmatrix} Vx(t) \\ Vy(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} \frac{(Vr+Vl)cos\theta}{2} \\ \frac{(Vr+Vl)sin\theta}{2} \\ \frac{Vr-Vl}{L} \end{bmatrix} = \begin{bmatrix} cos\theta/2 & cos\theta/2 \\ sin\theta/2 & sin\theta/2 \\ -1/L & 1/L \end{bmatrix} \begin{bmatrix} Vr \\ Vl \end{bmatrix} \tag{2.12}$$



Şekil 2.16. Robotun kinetik modeli

$$R = \frac{1}{2}\frac{Vl+Vr}{Vr-Vl} \quad (2.13)$$

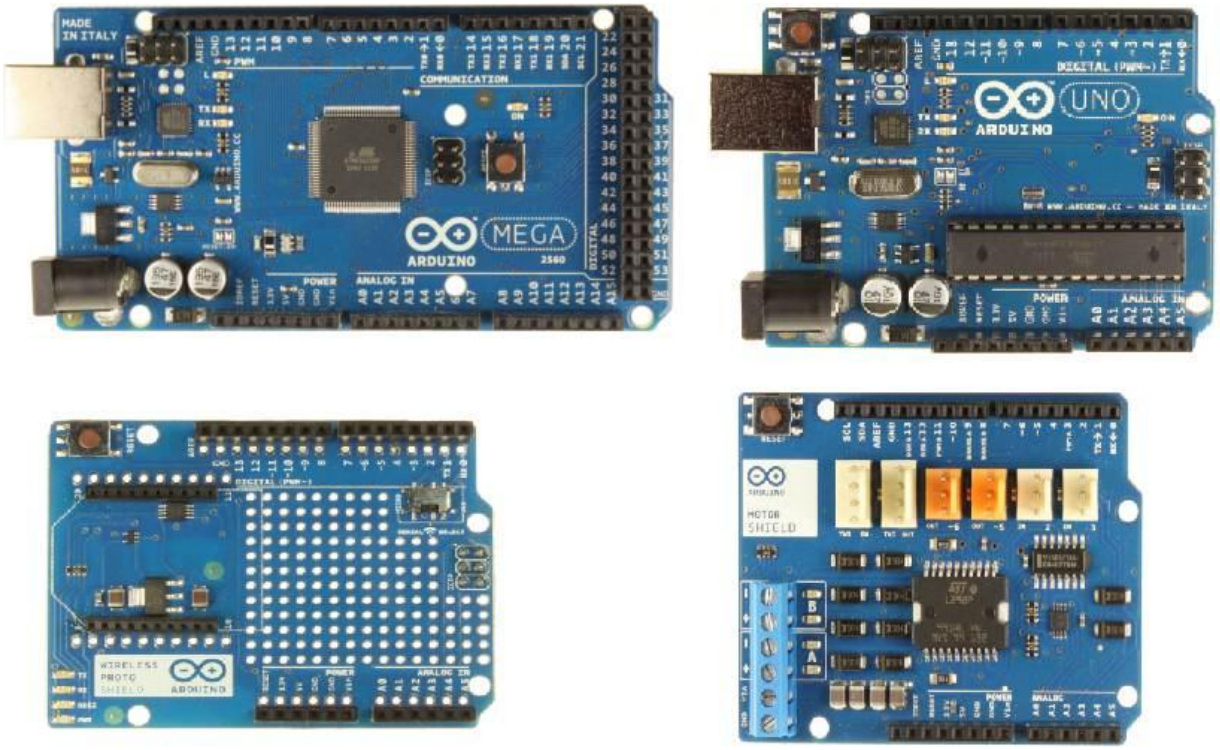
$$w = \frac{Vr-Vl}{l} \quad (2.14)$$

Şekil 2.16'ya göre bulduğumuz yarıçap ve açısal hız formüllerini bazı özel durumlar için incelendiğinde,

- Eğer sol motor hızı sağ motor hızına eşitse; Vl = Vr, R sonsuz olur, ve w açısal hız fiili bir şekilde oluşmaz.
- Eğer sol motor hızı ters yönde sağ motor hızına eşitse; Vl = - Vr, yarıçap R sıfır olacaktır. Robot hareket veren tekerlerin bulunduğu şasenin orta noktasına göre olduğu yerde dönme hareketi yapacaktır.
- Eğer sol motor dönüş hızı sıfıra eşitse; Vl = 0, R =1/2 olacaktır . Robot sol tekerleği merkez alarak dönüş yapacaktır. Aynı şekilde eğer sağ motor dönüş hızı sıfıra eşit ise; ; Vr = 0, R = -1/ 2 olacaktır.Robot Sağ tekerleği merkez alarak dönüş yapacaktır (Allen, 2005) .

2.4. Kontrol Elemanlarının Belirlenmesi

Bu bölümde proje kapsamında kullanılan kontrol elemanlarının analizine ve belirlenmesine değinilmiştir. Şekil 2.17. de projede kullanılan kontrol elemanları gösterilmiştir.



Şekil 2.17. Sistemde kullanılan kontrol elemanları

Sistemde kullanılacak olan kontrol elemanlarını belirlemek için ilk olarak sistem gereksinimleri ortaya çıkarılmalıdır. Sistem gereksinimlerini ortaya çıkarmak için robotun hareket kabiliyetini ve sistemin haberleşeme özellikleri incelenmelidir. Endüstriyel çizgi izleyen robot prototipi yapabilmek adına sistem gereksinimlerini incelediğimizde,

- Prototip robotun hareket sistemini oluşturacak 2 adet motor ve bu motorlara hareket verebilmek için motor sürücü devresi,
- Robotun engel algılayabilmesi için 1 adet analog giriş, ışıklı uyarı verebilmesi için 2 adet dijital çıkış ve sensör beslemelerinin yapılabilmesi için 5 adet dijital girişe sahip kontrol kartı,
- Prototip robotun ve parça işleme istasyonlarının birbirleri arasında kablosuz iletişim kurabilmeleri için kablosuz iletişim modülü ve bu modüllerin programlanabileceği kontrol kartları,
- Parça işleme istasyonlarında kablosuz iletişimi sağlayacak kablosuz iletişim modülü ve kontrol kartı,

- Parça işleme istasyonlarında malzeme ve hammadde bittiğini haber verecek ağırlık algılama sensörü ve her bir sensör için analog giriş,
- Parça işleme istasyonlarında ağırlık sensöründen aldığı veriyi işleyebilecek bir merkezi işlemciye sahip, ağırlık sensörü ve uyarı ledi için 1 adet analog giriş ve 1 adet dijital çıkışa sahip ve kablosuz iletişim modülü ile uyumlu merkezi işlemci birimi ve kontrol kartı,
- Prototip robot üzerinde parça işleme istasyonlarından ve ön tanımlı rotasından gelen verileri işleme yeteneğine sahip, 2 adet uyarı ledi (2 adet dijital çıkış) ve engel algılama sensörü için 1 adet analog girişi ve 5 adet çizgi takip sensörü için dijital giriş olan, programlama esnasında çizgi takip, haberleşme ve otomasyon programlamlarını bir arada barındıracağından yüksek program hafızasına sahip, prototip robot üzerindeki tüm diğer kontrol kartları ile giriş çıkış uyumu sağlayan ve kablosuz haberleşme işlemini destekleyen merkezi işlemci birimi ve kontrol kartı.

Sistem gereksinimlerinin yukarıdaki şekilde belirlenmesinden sonra kontrol kartlarınında bulunan giriş ve çıkışların belirlenmesi şu şekilde gerçeklenmiştir.

Sistem gereksinimi olan analog girişler aşağıda verilmiştir;

- Prototip robot engel algılama devresi için 1 adet engel algılama sensörü analog girişi,
- İstasyonlarda (3 adet) malzeme bittiğini algılamak için kullanılan ağırlık sensörü (3 analog giriş)

Sistem gereksinimi olan 5 adet dijital giriş aşağıda verilmiştir;

- Prototip robotta çizgi takip işlemini gerçekleştirecek olan kızılötesi sensörleri.

Sistem gereksinimi olan 10 adet dijital (pwm) çıkış aşağıda verilmiştir;

- Motor Pwm yön ve uyarı ledleri için toplamda 7 adet dijital (pwm) çıkış.
- Her istasyonda uyarı ledleri için 1er adet dijital (pwm) çıkış

2.4.1. Robot merkez işlemcisinin belirlenmesi

Sistemin tüm haberleşme, kontrol ve hareket birimlerini merkez işlemci yönetmektedir. Ayrıca kontrol işlemleri esnasında sürekli veri alışverişine imkan sağlayacak ve programlama esnasında hafıza durumu yeterli olacak bir merkez işlemciye ihtiyaç duyulmaktadır. Sistem gereksinimlerini incelediğimiz kablosuz haberleşme, motor kontrol, ve uyarı sistemininin ayrı şekilde programlanmasına imkan sağlayacak ve gerekli program hafıza yapısına sahip mikro işlemci ve bağlantı kartı olarak Arduino Mega 2560 kullanılmıştır. PLC ile karşılaştırıldığında prototip imalatı için düşük maliyetli ve modülerdir. Pik işlemciler ile karşılaştırıldığında kendi devre kartına sahip, uygulaması kolay ve program hafızası geniştir. Bu işlemcileri tablo 2.4. de karşılaştırırsak,

MİKRO İŞLEMCİLERİN KARŞILAŞTIRILMASI										
ÖZELLİKLER	PROGRAM HAFIZASI	MONTAJ DEMONTAJ KOLAYLIĞI	BAĞLANTI KART YAPISI	MALİYET	PROTOTİP ÜRETİMİNE UYGUNLUĞU	PROGRAMLAMA KOLAYLIĞI	TEDARİK EDİLEBİLİRLİK	MÜDAHALE EDİLEBİLİRLİK	ELEKTRONİK NİTELİKLERİ	Σ
KATSAYI	3	2	3	3	4	3	2	2	2	
PIC MİKRO İŞLEMCİ	-1	-1	-1	1	1	1	1	1	0	6
ARDUİNO	0	1	1	0	1	1	1	1	1	18
PLC	1	0	-1	-1	-1	0	0	0	1	-5

Tablo 2.4. Mikro İşlemcilerin karşılaştırılması

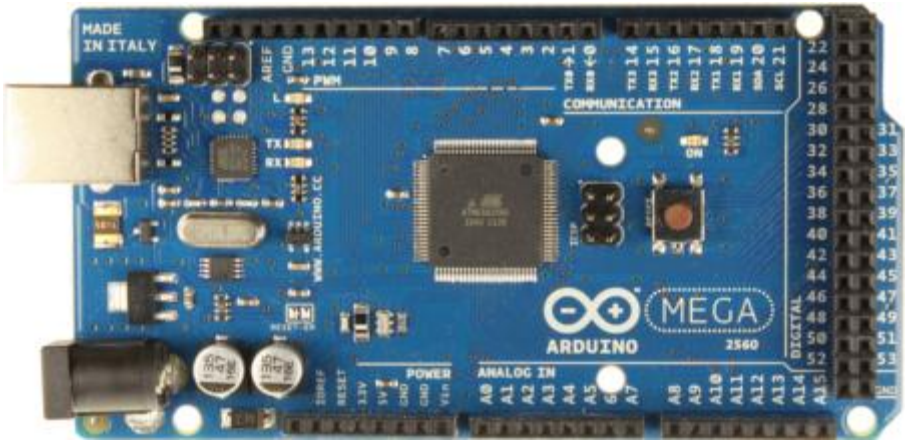
Arduino Mega 2560, bir I/O kartı ve İşleme/Kablolama dilini uygulayan geliştirme çevresi temelleri üzerine kurulu açık kaynaklı fiziksel hesaplama platformudur. Arduino mantıksal olarak bilgisayar üzerindeki yazılımlara da bağlanabilir (örn. Flaş, İşleme, MaxMPS). Açık kaynak IDE, ücretsiz olarak indirilebilir (mevcut olarak Mac OS X, Windows, ve Linux desteklenmektedir).

Arduino Mega Atmega2560 tabanlı bir mikroişlemci kartıdır. Üzerinde 54 dijital giriş / çıkış pini (bunlardan 14'ü PWM çıkışları olarak kullanılabilir), 16 analog girişten, 4 UART'tan (donanımsal seri portları), bir 16 MHz kristal osilatör, bir USB bağlantısı, bir güç girişi, bir ICSP bağlantısı ve bir reset butonundan oluşmaktadır. Bir mikroişlemciyi kontrol etmek için her şeye sahiptir; yalnızca bir USB kablosuyla beraber bilgisayara bağlanabilir. Çalıştırmak için ayrıca bir AC'den DC'ye çevirici adaptör ile ya da batarya ile kullanılabilir. Mega, Arduino için tasarlanmış pek çok shield ile uyumludur.

Projede arduino mega 2560 üzerine diğer işlemci katmanlarınında yerleştirilebilir olması nedeni ile işlemci bloğu yatay alanda fazla yüzey kaplamamıştır. Pinlerin modüler olarak bağlandığı noktalarda haberleşmesi ile birlikte motor kontrol ünitesi ve kablosuz iletişim modülü merkez işlemci sayesinde kontrol edilmiştir.

Özellikleri:

• ATmega2560 mikrodenetleyici
• Giriş voltajı -> 7-12V
• 54 Dijital I / O Pini (14 pini PWM çıkışı olarak kulanılabilir)
• 16 Analog Girişleri
• 256k Flaş Bellek
• 16Mhz Hız



Şekil 2.18. Robotun merkezi işlemcisi olarak kullanılan arduino mega 2560

2.4.2. Parça işleme istasyonlarının merkez işlemcisinin belirlenmesi

Parça işleme istasyonlarının merkez işlemcisinin belirlenmesinde robot merkez işlemci biriminin belirlenmesi ile aynı yol izlenmiştir. Parça işleme istasyonlarının nitelikleri incelendiğinde kablosuz iletişim modülü ve sinyal işleme biriminin olması gerekliliği prototip robota göre programlama kapasitesi küçük ve kablosuz iletişim modülü entegre edilebilecek bir mikro işlemcinin yeterli olduğunu göstermektedir. Bu noktada arduino Megaya göre düşük kapasiteli olan Arduino uno tercih edilmiştir. Arduino Uno bir I/O kartı ve İşleme/Kablolama dilini uygulayan geliştirme çevresi temelleri üzerine kurulu açık kaynaklı fiziksel hesaplama platformudur. Arduino mantıksal olarak bilgisayar üzerindeki yazılımlara da bağlanabilir (örn. Flaş, İşleme, MaxMPS). Açık kaynak IDE, ücretsiz olarak indirilebilir (mevcut olarak Mac OS X, Windows, ve Linux desteklenmektedir). ATmega328 işlemci kullanır. 14 dijital giriş/çıkış pini bulunur, bunlardan 6′sı PWM çıkışı olarak kullanılabilir. 6 analog giriş pinine sahiptir. 16 MHz kristal osilatörü, USb bağlantısı, 2.1mm güç girişi, ICSP başlığı ve reset butonu vardır. Mikroişlemciyi destekleyecek herşeye sahiptir. Çalıştırmak için DC 7~12V güç kaynağına bağlamak yeterlidir.

Özellikleri;

- Mikro denetleyici: ATmega328
- Çalışma gerilimi: 5V
- Tavsiye edilen besleme gerilimi: 7-12V
- Besleme gerilimi için alt ve üst sınırlar: 6-20V
- Dijital giriş/çıkış pinleri: 14 pin (6'sı PWM)
- Analog giriş pinleri: 6 pin
- Giriş/çıkış pini başına akım: 40mA
- 3.3V pini için akım: 50mA
- Flash: 32KB (0.5KB bootloader için kullanılır)
- SRAM: 2KB
- EEPROM: 1KB
- Saat frekansı: 16MHz

Şekil 2.19. Parça işleme istasyonlarında işlemci olarak kullanılan arduino uno

2.4.3. Motor sürücü ünitesi arduino motor sürücü devresi

Prototip robotta kullanılacak olan motor sürücü devresi merkez işlemci birimi olan arduino mega ile uyum sağlamalı ve montaj demontaj kolaylığı göstermelidir. Bu nedenle arduino motor sürücü kartı kullanılmıştır. Bu motor sürücü devresi, iki DC motor kontrol edecek olan Arduino için shield halinde hazırlanan bir motor sürücü modülüdür. L298 H-köprüsüne bağlı olarak, Ardumoto kanal başına 2A akıma kadar sürebilmektedir. Bu kart gücünü Arduino kartı ile aynı Vin hattından almaktadır, mavi ve yeşil LED'leri aktif yönü göstermeye yarar ve bütün sürücü hatları ters EMF'ten diyot korumalıdır.

OUT1/2 için bağlı olan motor kontrolü dijital hat 12'ye (yön A) ve dijital hat 10'a bağlanmaktadır (PWM A). OUT3/4 için bağlı olan motor kontrolü dijital hat 13'e (yön A) ve dijital hat 11'e bağlanmaktadır (PWM B).

Şekil 2.20 Motor sürücü devresi arduino motor kartı

3. SİSTEMİN PROTOTİP OLARAK ÜRETİMİ

Sistem; üretimde kullanılacak hammadde / malzemenin yüksek miktarda ve kısa zamanda tüketildiği bant usülü çalışılan üretim, lojistik ve montaj hatlarında operasyon işlemlerinin yapıldığı bölümlere otomatik olarak hammadde / malzeme yüklemeyi amaçlamaktadır.

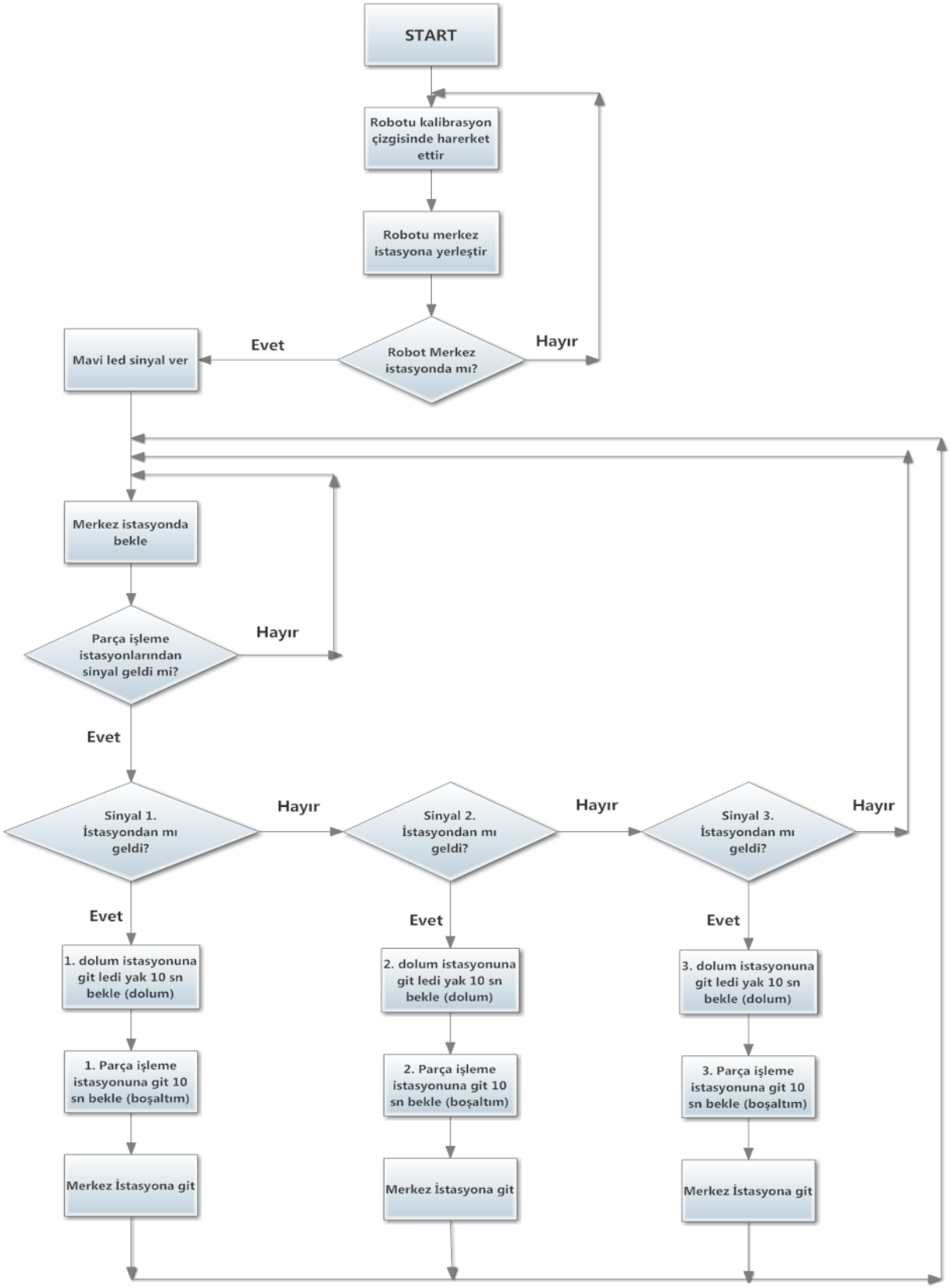
3.1. Sistem Çalışma Prensibi

Belirli bir yerleşkeye sahip olan üretim hatlarında öncelikli olarak Merkez istasyon belirlenir. Merkez istasyonun konumu belirlendikten sonra önemli olan üretim hatlarına parça besleyecek olan dolum istasyonlarının belirlenmesidir. Her istasyonun kendine ait bir istasyon numarası bulunmaktadır. Bu durum hem programlama hem de istasyon yerleşkesi hazırlama işlemini kolaylaştırmaktadır.

Dolum ve Parça işleme istasyonları numaralandırıldıktan sonra programlama işleminde bu bilgiler dikkate alınarak parça işleme istasyonlarına kablosuz iletişim modülü ve ağırlık sensöründen oluşan “istasyon izleme modülü” yerleştirilir. İstasyon izleme modülünün temel amacı; Parça işleme istasyonlarında hammadde/malzeme belirli bir oranın altına düştüğü anda bunun ağırlık sensörü yardımı ile algılanması ve kablosuz iletişim modülü ile Merkez istasyonda bekleyen malzeme taşıyıcı robota hareket sinyali göndermesidir. Bu sayede Malzeme taşıyıcı robot önce ilgili parça işleme istasyonunun dolum istasyonuna giderek gerekli hammadde/malzemeyi alacak ve ilgili parça işleme istasyonuna götürerek üretim işleminin devamlılığını sağlayacaktır.

Birden fazla parça işleme istasyonundan hammadde/malzeme sinyali geldiğinde malzeme taşıyıcı robot tüm sinyalleri işlemcisinde hafızasına alarak hammadde/malzeme biten tüm istasyonlara malzeme yükleme işlemini gerçekleştirecektir.

3.2. Sistemin Proje akış diyagramı



Şekil 3.1. Sistemin çalışma prensibi

3.3. Kullanılan Malzemeler

Aşağıda kullanılan malzemelerin referans numaralarını (http://www.robitshop.com/), kullanım adetlerini, birim ve toplam fiyatlarını gösteren tablo mevcuttur.

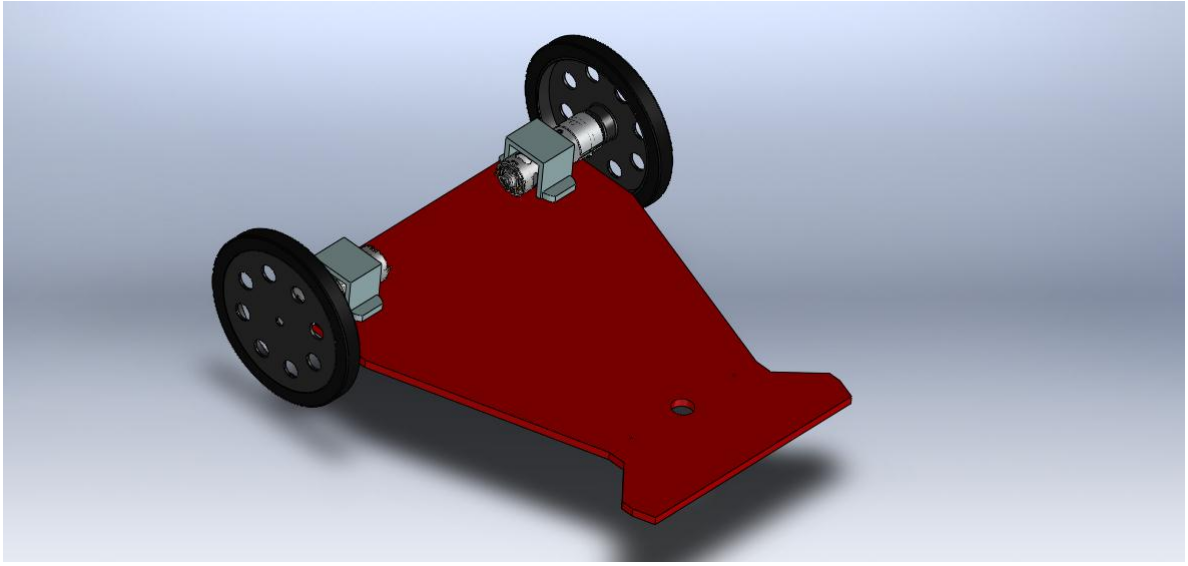
No	Malzeme	Kod	Adet	Birim Fiyat	Toplam Fiyat
1	Xbee explorer	WRL-08687	1	56,00 TL	56,00 TL
2	Motor Shield	RTL-09896	1	64,00 TL	64,00 TL
3	Arduino Mega	DEV -11061	1	120,00 TL	120,00 TL
4	Arduino Uno	DEV -11021	3	49,00 TL	147,00 TL
5	Ağırlık Sensörü (flexiforce sensör)	SEN-11207	3	18,00 TL	54,00 TL
6	Xbee modül	WRL-11217	4	58,00 TL	232,00 TL
7	298:1 Micro Metal Gearmotor	PL-1094	2	30,10 TL	60,20 TL
8	Arduino Connect Board (Konnektör Kartı)	RT-00001	1	20,00 TL	20,00 TL
9	LiPo Batarya 7.4V 800mAh	RBT-00747	1	13,56 TL	13,56 TL
10	Micro Metal Gearmotor Bracket Pair	PL-1086	1	10,00 TL	10,00 TL
11	QTR-8RC Reflectance Sensor Array	PL-961	1	28,20 TL	28,20 TL
12	Robot Şase-1	RT-05003	1	5,62 TL	5,62 TL
13	47x10 Teker (Hafif)	RT-18003	2	6,00 TL	12,00 TL
14	XBee Shield	WRL-09976	3	35,00 TL	105,00 TL
15	JST Soket Erkek-10cm Kablolu	PL-2181	1	3,00 TL	3,00 TL
16	Power soketi - Dişi Kablo Tipi	RT-01007	3	1,50 TL	4,50 TL
17	Ayarlı Adaptör-1000mA (Switch Mode)	RT001000	1	17,00 TL	17,00 TL
18	LED-3mm (4'lü Paket)	RBT00313	1	0,80 TL	0,80 TL
19	1K Direnç	RBT00332	1	0,29 TL	0,29 TL
20	Plastic Ball Caster	PL-174-1	1	1,50 TL	1,50 TL
21	Kablolu Krokodil	PRT-11037	2	0,75 TL	1,50 TL
				TOPLAM	956,17 TL

Tablo 3.1. Kullanılan malzemelerin maliyet tablosu

3.4. Sistemin Mekanik Yapısı

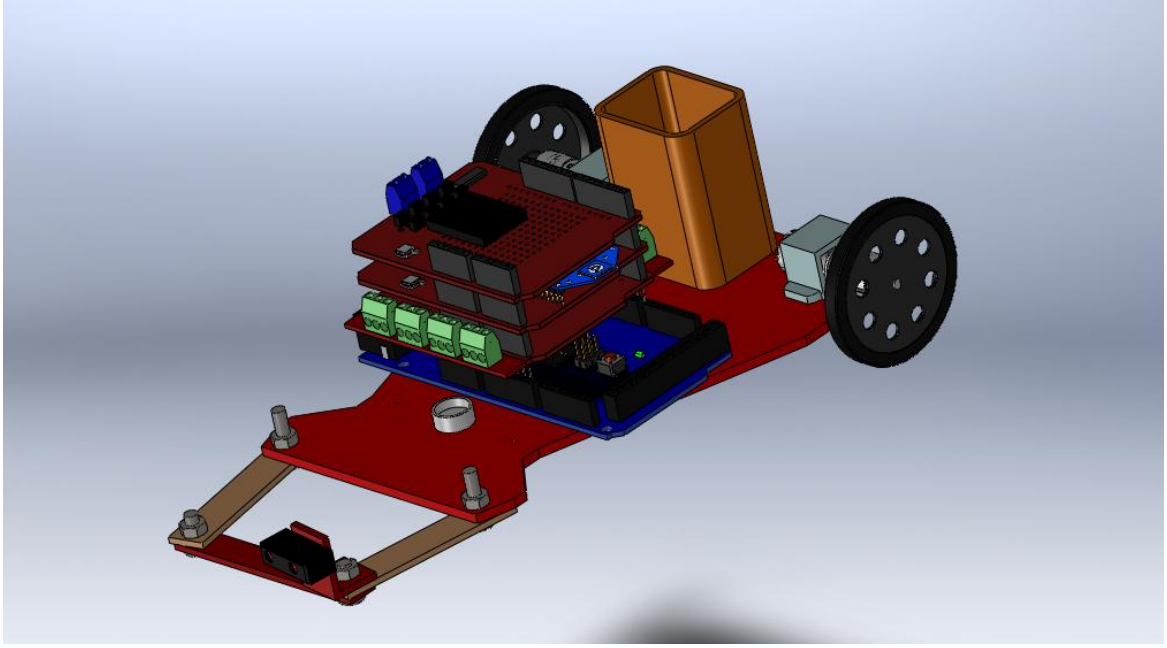
Sistemin mekanik yapısını incelediğimizde 3 bileşen ortaya çıkmaktadır. Bunlar;

- Robot,
- Parça işleme istasyonları,
- Dolum istayonlarıdır.



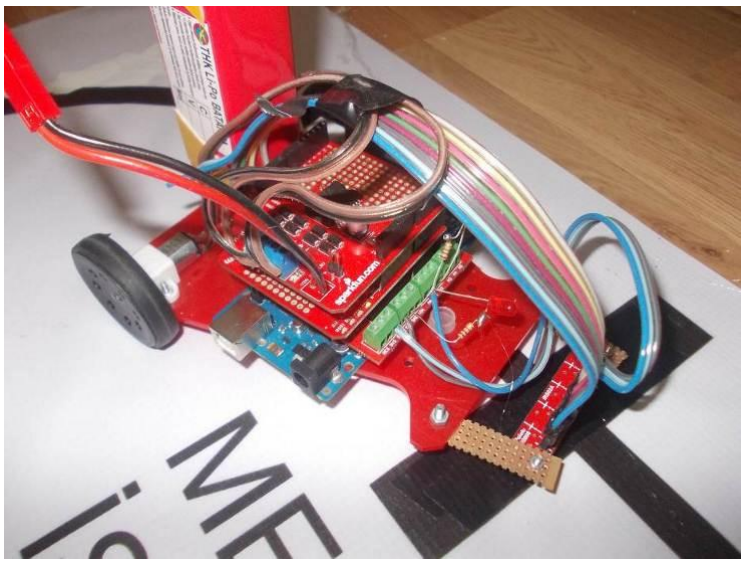
Şekil 3.2. Robot şasesi ve motorların 3 boyutlu ortamda tasarımı

Sistemin çalışma prensibine tekrar dönüş yaptığımızda her parça işleme istasyonunun kendine ait bir dolum istasyonu vardır. Parça işleme istasyonunda malzeme belirli oranın altına düştüğünde sinyal kablosuz iletişim sayesinde robota ulaşır. Robot bu sinyalin hangi istasyondan geldiğini anlar ve önce o parça işleme istasyonuna ait dolum istasyonuna gider ve dolum işlemi için önceden belirlenmiş süre kadar bekler. Daha sonra parça işleme istasyonuna gider ve boşaltım için önceden belirlenmiş süre kadar bekler. Endüstriyel uygulamalarda robotun malzemeyi getirdikten sonra tekrar hareket edebilmesi için operatör tarafından robot üzerindeki bir butona basılması veya robotun sensörlerle donatılmış platforma malzemeyi bıraktıktan sonra platformdaki sensörler tarafından uyarılarak malzeme boşaltım işlemininin bittiğini görmesi iş güvenliği kuralları gereğince uygulanmaktadır. Bu tezde prototip robot son olarak tüm işlemlerini bitirdiğinde merkez istasyona döner ve yeni bir sinyal için hazır durumda bekler.



Şekil 3.3. Prototip robotun 3 boyutlu ortamda tasarımı

Robot oluşturulan sistemin hem işlemcisine hem de hareket sistemine sahiptir. Robotun ön tarafında çizgi takip işlemi yapabilmesi için qtr rf (http://www.pololu.com/catalog/product/961) sensörler ve engel algılayabilmesi için gerekli uzaklık sensörü(http://robotus.net/wp-content/uploads/2012/07/SHARP-gp2y0d340k.pdf), dönüşlerde arka tarafın savrulmasını engellemek ve manevra kabiliyetini arttırabilmek adına ön tarafa konuşlandırılmış sarhoş teker, orta bölgede elektronik devrelerin bulunduğu merkez hareket işletim ve kontrol bölgesi bulunmaktadır.

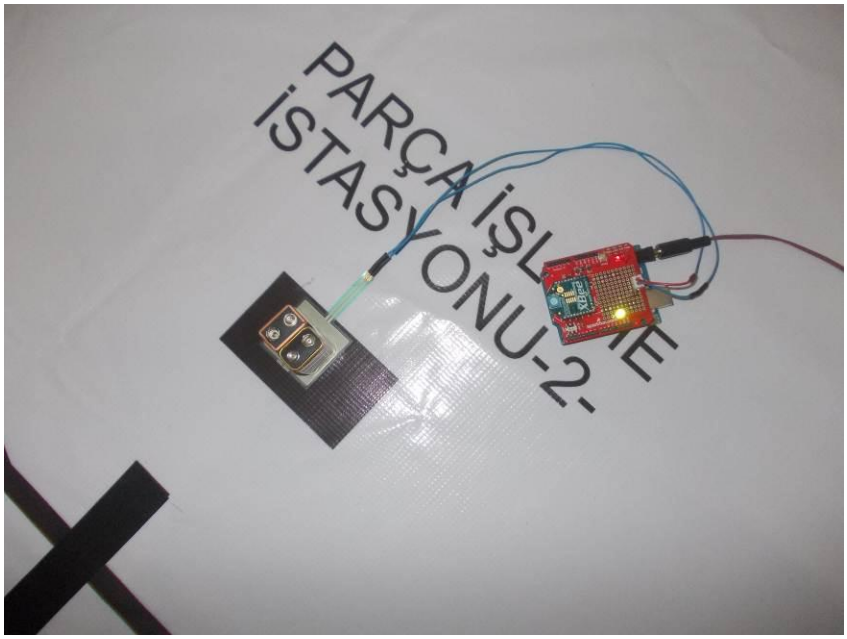


Şekil 3.4. Robotun ön-üst den görünümü

Arka tarafta her 2 yönde de aynı hızda gitmeyi sağlayacak sağ ve sol tekerleklere bağlı motorlar bulunmaktadır. Motorlar robot şasesinin üst tarafına yerleştirilerek hem montaj hem de bakım işlemlerinde süre tasarrufu sağlanmıştır. Arka tarafa ayrıca sistemin batarya kısmı da yerleştirilmiştir.

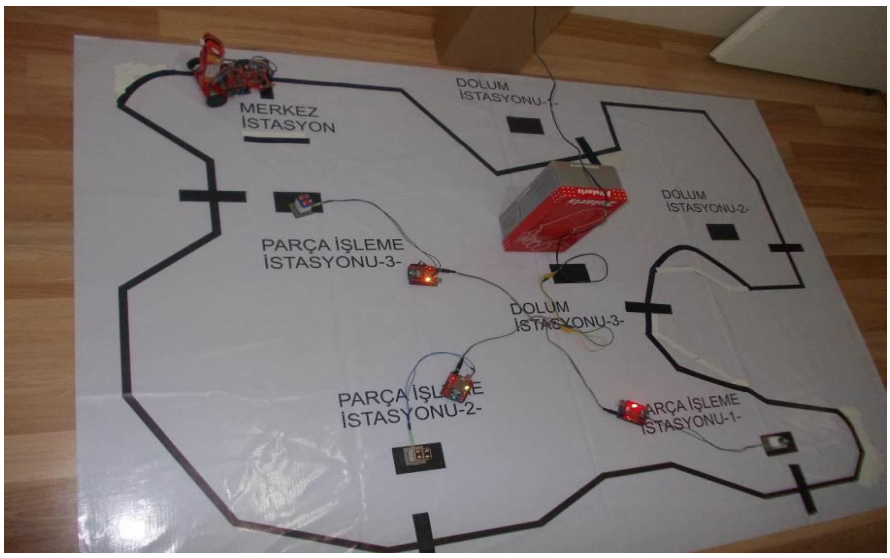
Şekil 3.5. Robotun arka-üst den görünümü

Oluşturduğumuz sistemde robota gönderilecek sinayallerin yönetiminden parça işleme istasyonları sorumludur. Parça işleme istasyonlarında mevcut sistem şu şekilde işlemektedir. İstasyona bağlı 1 adet “basınca duyarlı” ağırlık sensörü (http://dlnmh9ip6v2uc.cloudfront.net/verisheets/Sensors/Pressure/A401-force-sensor.pdf) vardır. Bu ağırlık sensörünün görevi; bağlı bulunduğu parça işleme istasyonunda hammadde/malzeme bitince veya ön tanımlı ağırlığın daha alt seviyelerine düşünce robota sinyal göndermektir. Robot sinyalin hangi istasyondan geldiğini mevcut programı sayesinde tanımlar ve o istasyona ait prosesin başlamasına izin verir.



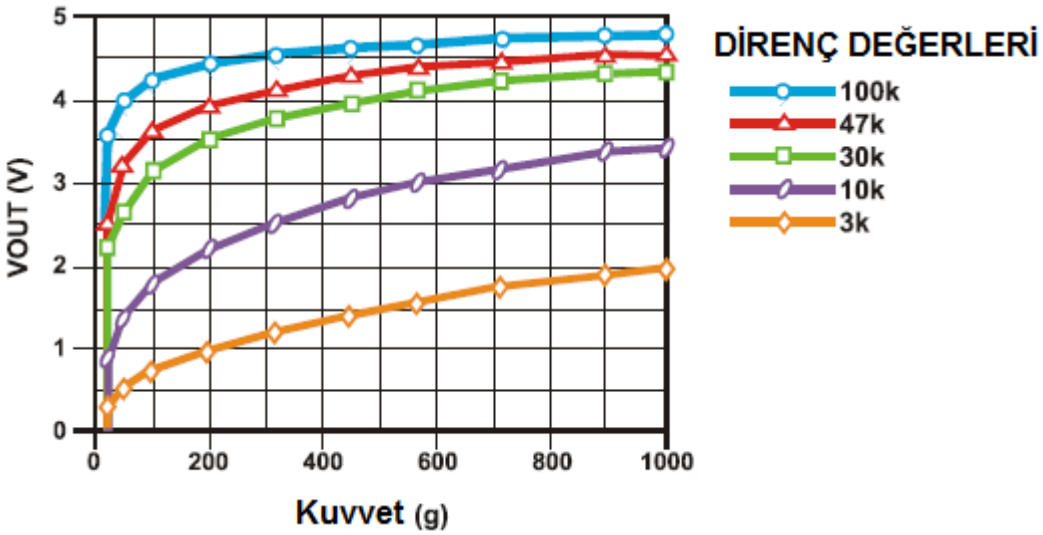
Şekil 3.6. Parça işleme istasyonlarının üstten görünümü

Parça işleme istasyonlarının fiziki yapısı; 1 adet ağırlık sensörü, kablosuz iletişim modülü ve bu işlemlerin gerçeklenmesini sağlayan işlemci biriminden meydana gelmektedir. Dolum istasyonları ise fiziki olarak parkura işaretlenmiş ve tüm çizgi takip sensörlerinin aynı anda aktif olabileceği bir çizgi şekilinde tanımlanmıştır.



Şekil 3.7. Sistemin genel görünümü

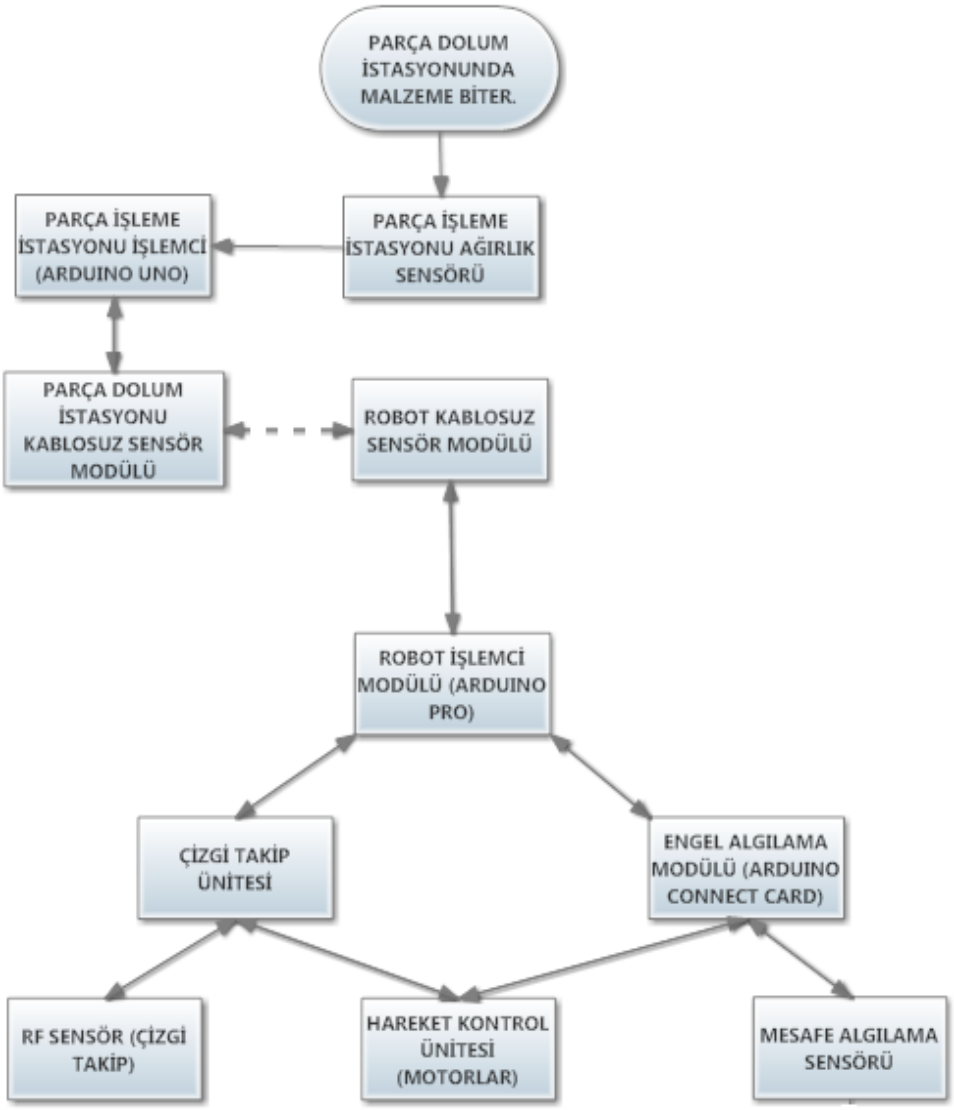
Parça işleme istasyonlarında bulunan yük algılama sensörleri (http://www.sparkfun.com/verisheets/Sensors/Pressure/fsrguide.pdf) aslında birer dirençtir. Parça işleme istasyonlarındaki ağırlık değişimlerini ölçebilmek için direnç değişimini ölçebilmek gerekir. Bu noktada ağırlık sensörüne ikinci bir direnç ile seri bağlayıp voltaj bölücü devresi kurularak iki direnç arasındaki voltaj değerleri sensörün üzerine baskı uygulayınca değişir. Sinyal kablosu ise iki direncin birleşim yerinden alınır ve arduino uno'ya beslenir. Ağırlık sensörüne bağlanılan direnç değerine göre ölçüm hassasiyeti ve ölçüm aralığı değişmektedir.



Şekil 3.8. Ağırlık Sensörü Direnç – Kuvvet değişim diyagramı

3.5. Sistemin Elektronik Yapısı

Sistemin elektronik yapısını incelediğimizde karşımıza çıkan haberleşme sistemi şu şekilde görünmektedir.



Şekil 3.9. Sistemin elektronik sinyalizasyon şeması

Robotun merkez kontrol ünitesi Arduino mega (http://www.arduino.cc/en/Main/arduinoBoardMega) ile hazırlanmıştır. Temel kontrol ünitesi üzerine 3 adet kart daha bağlanmıştır. Bu kartlar ile motor sürücü devresi Arduino Motor shield ile (http://arduino.cc/en/Main/ArduinoMotorShieldR3) ledler ve sensörlerin bağlantıları arduino connect card ile (http://www.robitshop.com/Arduino-Connect-Board-v2-Konnektor-Karti,PR-

2321.html) ve sistemde kablosuz iletişimi sağlayacak olan ekipman arduino kablosuz iletişim modülü(http://arduino.cc/en/Main/ArduinoWirelessShield) ile hazırlanmıştır.

Robottaki kablosuz iletişim modülü üzerinde sinyalizasyonu sağlayan eleman xbee (http://ftp1.digi.com/support/documentation/90000976_G.pdf) kablosuz iletişim modülüdür. Aynı zamanda parça işleme istasyonlarında bulunan kablosuz iletişim modülü robot üzerindeki ile aynı olup işlemci modülü ise arduino uno (http://www.arduino.cc/en/Main/arduinoBoardUno) olacak şekilde hazırlanmıştır.

3.6. Sistemin Programlanması

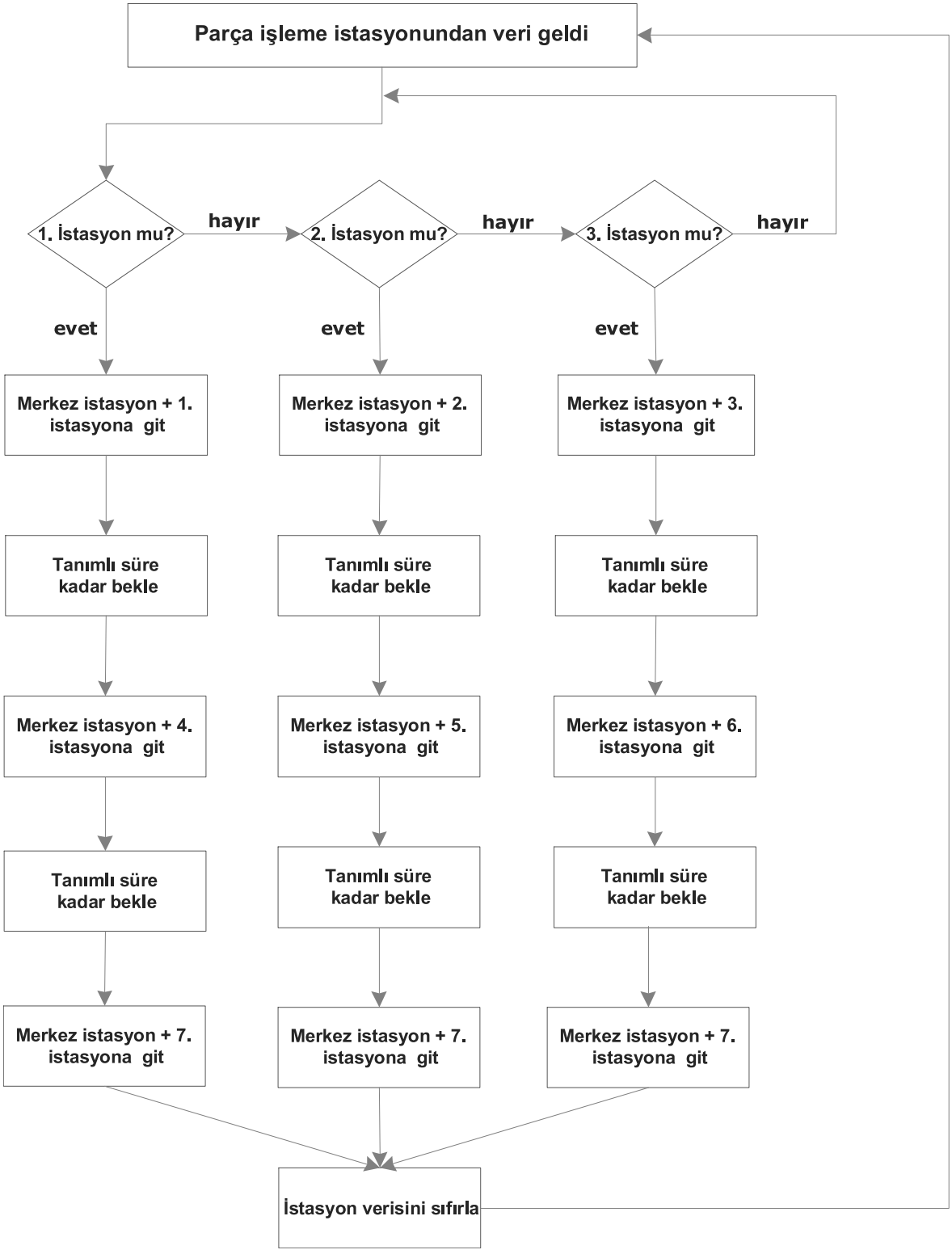
Sistemin programlanması esnasında kullanılan programlama dili arduino olarak belirlenmiştir. Arduino kodlarının kullanılmasındaki başlıca neden hem açık kaynak olarak (http://www.arduino.cc/en/Reference/HomePage) istenildiğinde ulaşılabilmesi hem de kullandığımız mevcut devrelerin arduino ile uyumlu olmasından kaynaklamaktadır. Robotu programlarken 4 ana programlama işlemi üzerinde çalışılmıştır. Bunlar ;

- Robotun ana fonksiyonu olan çizgi izleme ünitesinin programlanması,
- Haberleşme ünitesinin programlanması,
- Malzeme bitince sinyal gönderme görevini gerçekleştirecek olan istasyonların programlanması,
- Son olarak bütün bu verilerin bir arada haberleşmesini sağlayan otomasyon programının gerçekleştirilmesi.

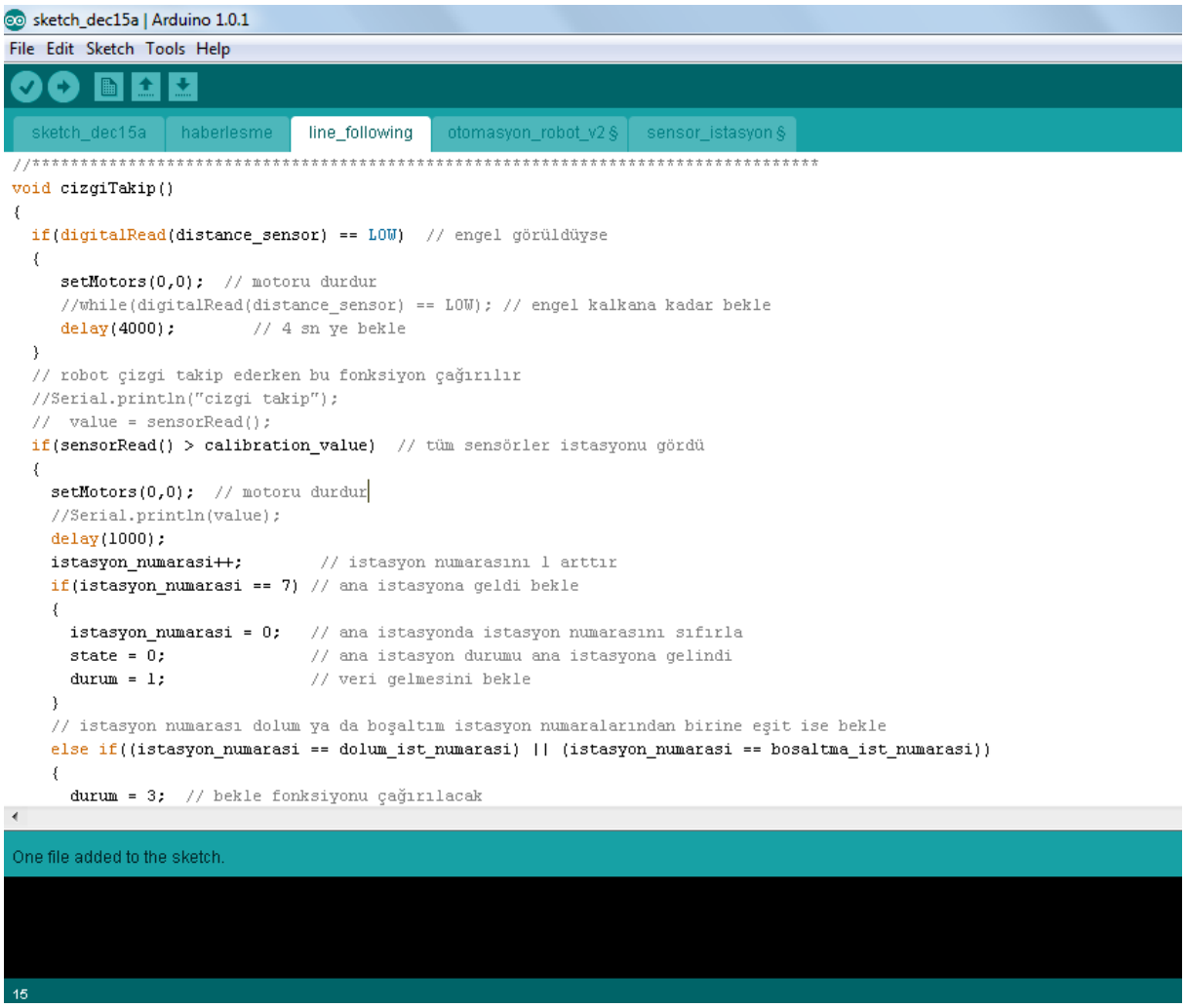
Çizgi izleme ünitesinin programlanması

Çizgi izleme ünitesinin programlanması esnasında yapılacak asıl işlem; robotun istasyonları tanıması, ve bu istasyonlara ulaşabilmesi için önceden belirlenmiş olan parkurda ilerlerken parkurun fiziki koşullarına uygun olarak hareket edebilmesidir. Bu noktada robotun ön kısımında bulunan sensörler parkuru takip etme işlemini gerçekleştirmektedir. Sensörlerden alınan değerler motorlara hareket bilgisi

oluşturarak yolun izlenmesini sağlamaktadır. Robotun endüstriyel ortamlarda çalışacağı göz önüne alınarak karşısına engel çıkması durumunda bu engeli algılayarak durması ve engel yolundan çekildikten sonra bir süre bekleyip tekrar yoluna devam edebilmesi tasarlanmıştır. Bu noktada programlama için mantık şeması şu şekilde geliştirilmiştir. Çizgi izleme işlemi parça işleme istasyonundan sinyal gelmesiyle birlikte başlamakta ve istasyon verisini sıfırlayarak bitirilmektedir. Çizgi izleme işleminin herhangi bir safhasında farklı bir istasyondan sinyal gelirse bu sinyal işlemci hafızasında tutulmakta ve aktif olan görevini yerine getirdikten sonra merkez işlemciye farklı bir istasyondan gönderilmiş olan sinyal ile tanımlanmış görevi gerçekleştirmektedir.



Şekil 3.10. Çizgi izleme ünitesi programlama mantık şeması

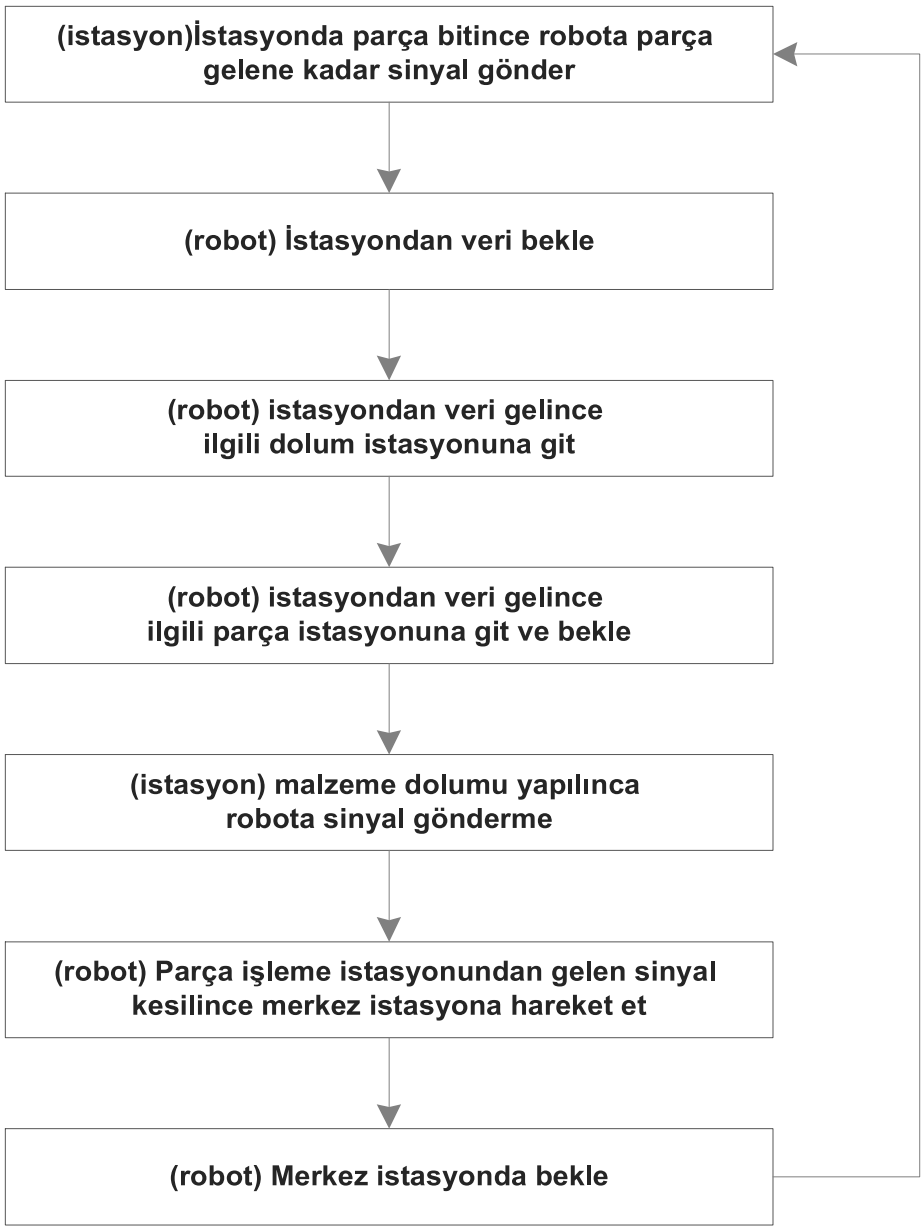


Şekil 3.11. Çizgi izleme işleminin programlanması esnasında arduino yazılımının görüntüsü

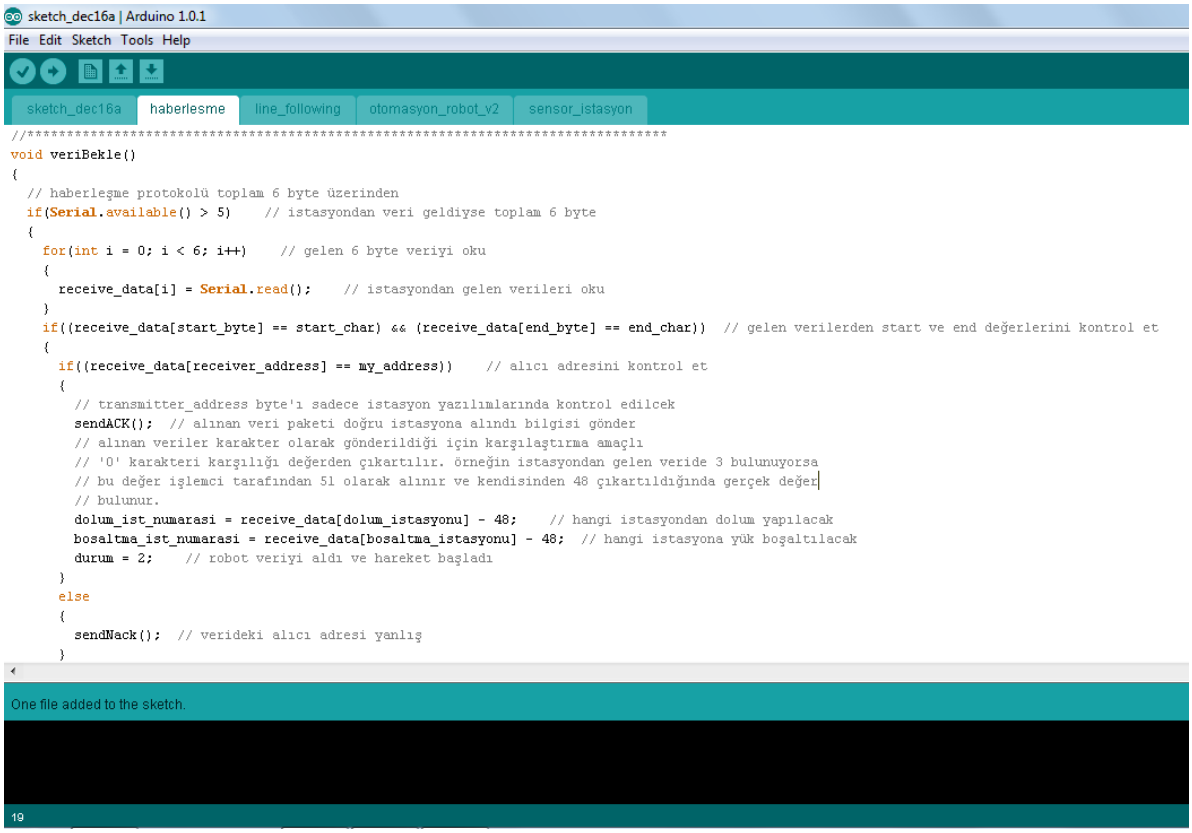
.

Haberleşme ünitesinin programlanması

Haberleşme ünitesinin programlanmasında aktif olarak 2 devre rol almaktadır. Birinci devre robot üzerindeki kablosuz iletişimi sağlayan modül, ikinci devre ise istasyonlardaki kablosuz iletişimi sağlayan modüldür. Haberleşme ünitesinde ağırlık sensöründen alınan veriler istasyonda malzemenin bittiğini göstermekte ve kablosuz haberleşme sayesinde istasyonlar ve robot karşılıklı olarak veri gönderebilmektedir. Bu noktada haberleşme ünitesinin programlanması için mantık şeması şu şekilde gerçekleştirilmiştir.



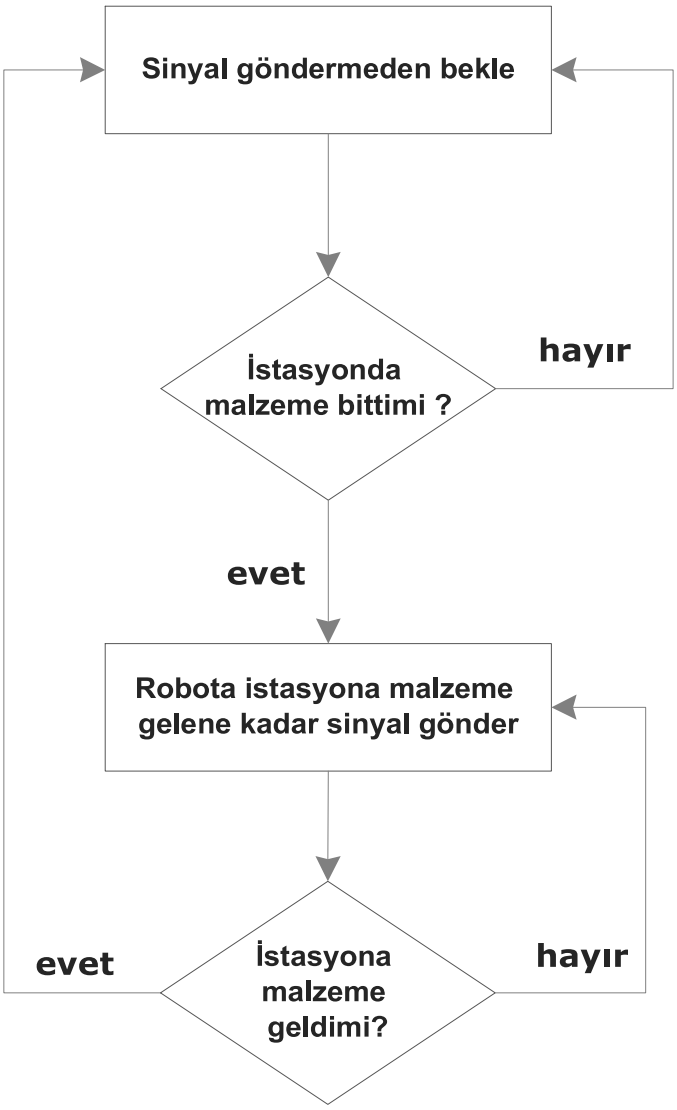
Şekil 3.12. Haberleşme ünitesinin programlama mantık şeması



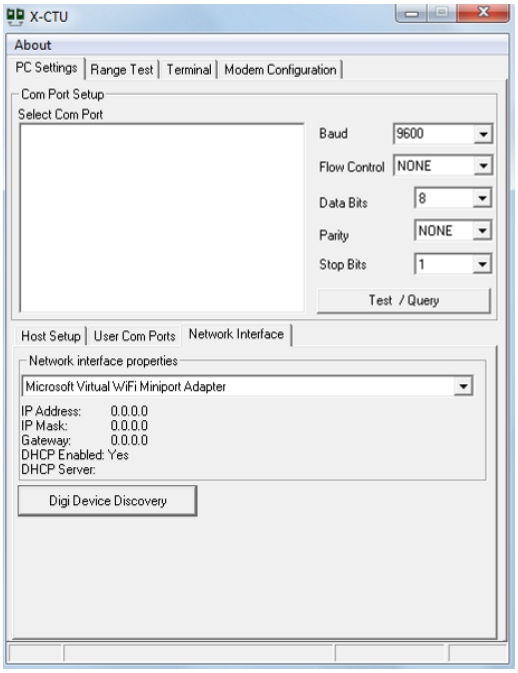
Şekil 3.13. Haberleşme işleminin programlanması esnasında arduino yazılımının görüntüsü

3.6.3. İstasyonların programlanması

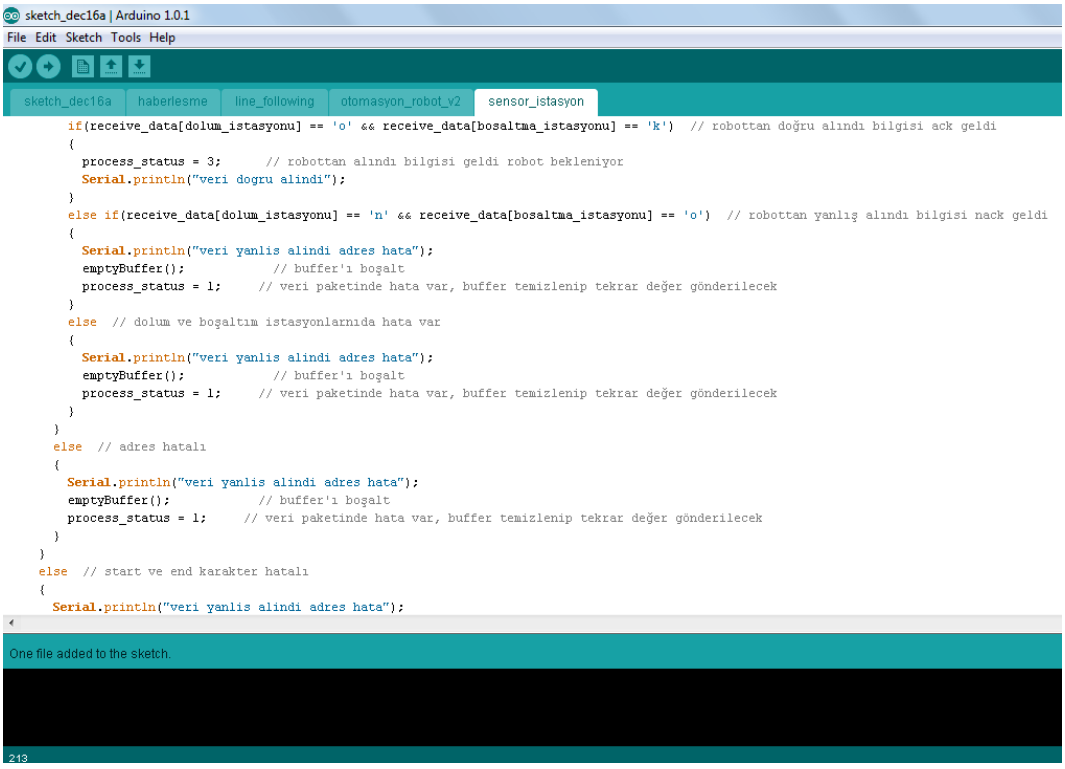
İstasyonların progranlanması esnasında en önemli durum; istasyonlardan gönderilecek verilerin robot tarafından algılanabilmesi ve bu verilerin kendi içerisinde ayırt edilebilir olması durumudur. Bu noktada kullandığımız program ile istasyonlara farklı kimlikler belirleyerek robotun istasyonlardan gelen verileri tanıyabilmesi ve bu verilere göre otonom hareket geliştirmesi sağlanmıştır.



Şekil 3.14. İstasyonların programlanmasındaki mantık şeması



Şekil 3.15. İstasyonların tanımlayıcı bilgilerini oluşturduğumuz programdan bir görüntü



Şekil 3.16. İstasyonların programlanmasında yazılımın ekran görüntüsü

3.6.4. Otomasyon işleminin programlanması

Robotun ve istasyonların ayrı ayrı programlanmasından sonra oluşan tüm bu verilere göre otonom çalışma ilişkilerini sağlaması amacı ile otomasyon bölümü programlanmıştır. Bu noktada otomasyon bölümünün amacı; diğer 3 programdan anlık durumlara göre prototip robotun pozisyon kontrolünü yaparak ve bağlantılar ile verileri alarak sistemin nasıl hareket edeceğine karar vermektir. Otomasyon döngüsünde robotun merkez istasyona yerleştirilmesi ve sistemin çalıştırılması ile başlayan döngü istasyonlardan malzeme bittiğinin sinyali gelmesi ile devam eder. Bu noktada robot hangi istasyondan malzeme sinyali geldiğini haberleşme programı ile analiz etmektedir. Daha sonra istasyonlara gitmesi için gerekli hareketleri ise çizgi izleme programını aktif ederek gerçekleştirmektedir.

Şekil 3.17. Otomasyonun programlanması esnasında arduino yazılımının görüntüsü

4. SONUÇLAR VE ÖNERİLER

Endüstriyel taşıyıcı robotların sanayi ortamındaki otomasyon kullanımındaki yeri kaçınılmazdır. Otomasyon sistemlerinin varlığı hem verimli çalışmayı hem de bakım ve onarım maliyetlerini doğrudan etkilemektedir. Bu tez kapsamında yapılan proje aynı mantık doğrultusunda daha büyük bir prototipinin yapılması ile seri imalatın yapıldığı, bant usulü veya otonom parça işleme yapısına sahip tüm fabrikalarda lojistik operasyonlarında kullanılabilir.

Mekanik ve elektronik çatısı tamamlanmış olan endüstriyel robotta çizgi takibi uygulamasında çizgi tespiti için manyetik algılama ve ultrasonik algılama yöntemlerinin kullanılmasının bazı ortamlarda verimsiz olduğu, çizgi takibinin stabil yapılabilmesi amacı ile manyetik nesnelerden ve sonar sensörün algılayabileceği yapılardan yeterli derecede korunması gerektiği görülmüştür. Bu faktörler göz önüne alınarak, çizgi takibi sensör modülünün tasarlanmasında, kızılötesi led ve kızıl ötesi alıcılar kullanılmıştır. Endüstriyel taşıyıcı robotun daha adaptif bir yapıya kavuşması için, yol izleme kısmı çizgi takibi yerine, görüntü işleme metodu ile de gerçekleştirilebilir ancak görüntü işleme teknolojisinin maliyeti ve bakım uygulamaları oldukça zordur.

Robotun otonom hareketinin sağlanması için çalışacağı ortamda ön tanımlı ve kablosuz haberleşme yetisine sahip parça işleme ve parça dolum istasyonları oluşturulmuştur. Uygulama nitelikleriyle bizim bilgimiz dahilinde ilk olma özelliğine sahiptir. Parça işleme istasyonlarındaki ağırlık sensörlerinin belirli bir değerin altına indiğinde robota sinyal göndermesi ve robotun önce dolum istasyonundan hammadde/malzeme alıp daha sonra ilgili parça işleme istasyonuna götürmesi durumu tam olarak otonom otomasyon sisteminin kurulması ile meydana getirilmiştir.

Robotun programlanmasında açık kaynak kodlara sahip arduino işletim sisteminin kullanılması robot maliyetlerini azaltmış ve programlama işlemlerinde kolaylık sağlamıştır. Ayrıca arduino işlemcilerin kullanılması hem parça tedarik kolaylığı hem de müdahale edilebilirlikte kolaylık sağlamıştır.

Robotun tasarımınıda sürüş yöntemi olarak diferansiyel sürüş yöntemi kullanılmıştır. Robot hareket yönteminin belirlenmesinde ise tekerlekli sürüş yöntemi kullanılmıştır. Diferansiyel sürüş ve tekerlekli sürüş yönteminin birarada kullanılması bina içi uygulamalarda kullanılmak için tasarlanmış olan prototip robota hareket kabiliyeti kazandırmıştır.

Prototip robotun merkez işlemcisi ve haberleşme üniteleri olarak tercih edilen arduino elektronik kartları tercih edilmiştir. Yazılım işlemlerinde açık kaynak kullanılması ve internet alışverişi sayesinde kolay tedarik edilebilen bu kartlar prototip üretimi için kullanışlıdır.

Robotun endüstriyel uygulamalarda kullanılabilecek bir modelinin yapılması esnasında elektronik kartları mekanik tasarımda değiştirilebilirlik adına robotun kolay ulaşılabilir bir bölgesinde olması ve tahrik sisteminin servo motorları yerine elektrik motoru veya adım motorları seçilerek yapılması bakım ve üretim maliyetlerini azaltmaktadır. Endüstriyel robot sabit bir enerji kaynağından beslenemeyeceğinden akülerin kolay sökülür takılır olması gerekmektedir. Robotun tahrik mekanizması arkadan olduğu için vagon sistemi veya robotun üst platformu düz yapılarak malzeme taşıma işlemi bu şekilde gerçekleştirilebilir.

Endüstriyel lojistik uygulamalarında çizgi izleyen robotların kullanımı artmaktadır. Yakın gelecekte üretim hatlarında operatörlerin kullandığı taşıma ve kaldırma araçlarının yerini çizgi izleyen robotlar alarak hatlarda otonom olarak besleme işlemlerini gerçekleştireceklerdir. Çizgi izleyen robotların kullanımı arttıkça fabrikalarda robotlar için otomatik batarya değişim istasyonları, kontrol merkezi gibi yeni ihtiyaçlar ortaya çıkacaktır. 2 veya 3 operasyonlu birleştirme işlemi gerçekleştikten sonra hammadde olarak kullanılan malzemeler içinse çizgi izleyen robotların üzerinde çeşitli birleştirme üniteleri oluşturulacak ve bu robotlar hatlara malzeme taşırken aynı zamanda yarı mamül üretebilecek kapasitelere uzak gelecekte de olsa ulaşabileceklerdir.

KAYNAKLAR DİZİNİ

Kumar, S., A., Suresh, N., 2006, Production and Operations Management, New Age International Publishers, New Delhi, 208 s.

Hirao, S., Tamaki, M., Ohno, K., 2010, Production Planning and Control, Taylor and Francais Group, Kobe, 746-753.

Hase, H., Okino, N., Tamura, H., Fuji, S., 1998, Advances in Production Management Systems Perspectives and Future Challanges, Chapman and Hall, London, 445-466.

Milli Eğitim Bakanlığı Ulaştırma Hizmetleri., 2011, Forkliftler Transpaletler ve Vinçler, 840uh0041, Ankara, 110 s.

İş Sağlığı ve Güvenliği Risk Değerlendirmesi Yönetmeliği, 2012, 6331 Sayılı İş Sağlığı ve Güvenliği Kanunu Mevzuatı, Resmi gazete, Madde 8-9-10.

Schulze, L., Behling, S., Buhrs, S., 2009, People Following Automated Guided Vehicles - Research and Application, Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol II, Hong Kong, 5 s.

Pechoucek, M., Rehak, M., Charvat, P., Vlcek, P., Kolar, M., 2006, Agent based approach to Mass-Oriented Production Planning: Case Study, MULTI-AGENT PLANNING IN MASS-ORIENTED PRODUCTION, Praha, 8 s.

Thomson, A., J., 2001, A Path Following System for Autonomus Robots with Minimal computing Power, Yüksek Lisans Tezi, University of Auckland, Auckland, 59 s.

Wolfbeis, O., S., Narayanaswamy, R., 2004, Optical Sensors Industrial, Environmental and Diagnostic Aplications, Springer Series, Heidelberg.

Ripka, P., 2001, Magnetic Sensors and Magnetometer, Artech House, London, 494 s.

Wilson, J., S., 2005, Sensor Technology Handbook, Newnes, Bulington, 690 s.

Jones, J., L., Flynn, A., M., Seiger, B., 1998, Mobile Robots Inspiration to implementation, AK Peters, Natick, 486 s.

Arkin, R. C., 1998, “Behavior Based Robotics”, The MIT Press, 491 s.

Bonasso, R. P., Kortenkamp, D., and Murphy, R., 1998, “Artificial Intelligence and Mobile Robots”, AAAI Press / The MIT Press, 400 s.

Murphy, R., 2000, “*Introduction to AI Robotics*”, The MIT Press, 400 s.

Arkin, R.C. and Balch, T., 1995, “Cooperative Multi-agent Robotic Systems”, 16 s.

Allen, P., 2005 , Computational Aspects of Robotics, Columbia University, 183 s.

Borenstian,J. Everett, H.R., Feng, L., 1996, Sensors and Methods for Mobile Robot Positioning, The university of Michigan, 282 s.

Özen, S., 2000, Bir Gezgin Robot için Elektronik Denetim Donanımının Tasarımı ve Uygulaması, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü

Türker, T., 2005, Tekerlekli Gezgin Robotlarda Sistematik Odometri Hatalarının Belirlenmesi ve Azaltılması, Yüksek Lisans Tezi, YTÜ Fen Bilimleri Enstitüsü

Lucas, G.W., 2001, A Tutorial and Elementary Trajectory Model for the Differential Steering System of Robot Wheel Actuators, http://rossum.sourceforge.net/papers/DiffSteer/.

Kola, K., 1998, Motion Desing in for Bearcat 1, MS, University of Cincinati, 9 s.

Oxer, J., Blemings, H., 2009, Practical Arduino Cool Projects for Open Source Hardware, Technology in action, 445 s.

Şahin, H.T., Zergero_lu, E., 2006, “Sınırlı Algılama Alanı Olan Tekerlekli Mobil Robotlar için Holonom Olmayan Bir Rota Planlama Yöntemi”, Türk Otomatik Kontrol Toplantısı, TOK, TOBB Üniv., Ankara.

Drimer, S., 2002, CCD Image Processing Prototyping Platform Senior Project Design Report, MDSP Group, 25 s.

Akbatı, O. ve Cansever, G., 2007, Depo Robot Sistemi, 4. Otomasyon Sempozyumu, 19 Mayıs Üniv., 2007, Samsun.

Fu, K. S., Gonzalez, R. C., Lee C. S. G., 1987, ROBOTICS : Control, Sensing, Vision, and Intelligence, McGraw-Hill Book Company, 580 s.

Altınbaşak O., 2000, Mikrodenetleyiciler ve Pic Programlama, Altaş Yayınları, İstanbul.

Bayram, Z., 2006, Ultrasonik Mesafe Algılayıcıları ile Mobil Robot Kontrolü, Yüksek Lisans Tezi, Osmangazi Üniv. Fen Bilimleri Enstitüsü, 64 s.

Wee, K. C., 2007, PID Control of Line Following Robot, Yüksek Lisans Tezi, Faculty of Electronics & Computer Engineering Universiti Teknikal Malaysia, 27 s.

Özdemir, K., 1996, Tek CCD Kamera Kullanılarak Bir Gezer Robotun Konumunun Belirlenmesi, Yüksek Lisans Tezi, O.D.T.U., Ankara, 107 s.

Roman, A., et al., 2006, Inteligent Line Follower Mini-Robot System, International Journal of Computers, Communications & Control, 73-83 s.

Aydoğmuş, Ö., 2006, Pic Mikrodenetleyici Yardımı ile Dc Motorun Hız Kontrolü, Yüksek Lisans Tezi, Fırat Üniversitesi, Elazığ, 104 s.

Özdemir, Y., 2006, Çizgi İzleyen Gezgin Bir Robotun İncelenmesi, Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi, Samsun, 91 s.

Akbatı, O., 2007, Çizgi Takip Robotunun Modellenmesi ve Denetleyici Tasarımı, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, İstanbul, 49 s
.

Özkan, S., B., 2006, Endüstriyel Tesisler İçin Çok Ajanlı Bir Otonom Robot Sistemi Tasarımı ve Simülasyonu, Yüksek Lisans Tezi, Galatasaray Üniversitesi , İstanbul, 144 s.

Yetişenler, Ç., 2005, Efficient Multiple Robot Path Planning Algorithm, Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi, İzmir, 102 s.

Banzi, M., 2008, Getting Started with Arduino, Make:Projects O'Reilly, 130 s.

Igoe, T., 2007, Making Things Talk, Make:Projects O'Reilly, 427 s.

Ünver, Ö., 2009, Design and Optimization of Miniature Climbing Robots using Flat Dry Elastomer Adhesives, Doktora Tezi, Pittsburgh, 164 s.

Mitchel, W., C., Stainfort, A., Scott, A., 2006, Analaysis of Ackeman Steering Geormetry, Michigan Technology University, Michigan, 11 s.

Dağdeviren, C., Kullukçu, A., 2001, Traktörlerde Hidrostatik Direksiyon Sistemi Tasarımı, 2. Uluslararası Hidrolik Pnömatik Kongresi Sergisi, 347- 356.

Petrineç, K., Kovaçiç, Z., Marozin, A., 2003, Simulator of Multi-AGV Robotic Industrial Environments, Vicenza, 5 s.

Sezen, B., 2003, Modeling Automated Guided Vehicle Systems in Material Handling, Doğuş Üniversitesi Dergisi, İstanbul, 207-216

Force Sensing Resistor Integration Guide and Evaluation Parts Catalog, 2002, Interlink Elektronics, State of the art solition for the OEM, 26 s.

Belpaeme, T., 2004, Autonomous Systems Manipulation and Locomotion, Autonomous Systems.

Cao, M., Liao, X., Hall, E., 1999, Motion Control Design of the Bearcat II Mobile Robot, Center for Robotics Research University of Cincinnati, Cincinnati, 9 s.

Eddy Current Probe Systems Specifications, 2011, Allen-Breadley, Rockwell Software and Automation, 17s.

Dusek, F., Honc, D., Rozsival, P., 2011, Mathematical model of differentially steered mobile robot, 18th International Conference on Process Control, Slovak University of Technology in Bratislava Institute of Information Engineering, Automation, and Mathematics, Pardubice, 220-230

Bansal, N., Arora, S., Shiva, S., 2010, Line Follower Robot, Department of Physics Panjab University, Chandigarh, 30 s.

Kwhang, T., et al., 2000, Development of a Test-Bed for Outdoor AGV Research, University of Arkansas, 3 s.

Chris and Dawn Schur's Robotics and Artificial Life Forms, http://www.schursastrophotography.com

Wallace, David N., Line Following Robot, http://www.lifekludger.net/category/weekly-links/page/2/

Denmarks Techniske Universitet, http://www.sweeper.org

Mike's Line Following Robot, Central Illinois Robotics Club, http://www.circ.mtco.com

Robotic Research works, http://www.leang.com

http://www.cs.umn.edu

http://www.ece.unm.edu

David Cook's Jet http://www.robotroom.com/jet.html

http://blog.makezine.com

The CBA Line Following Module, http://www.budgetbot.com

Line Detection Control on Led and Ldr Sensors , http://www.seattlerobotics.org/encoder/200011/LineDetect2.htm.

http://electronics.howstuffworks.com/cameras-photography/digital/digital-camera2.htm

http://literature.rockwellautomation.com/idc/groups/literature/documents/td/1442-td001_-en-p.pdf

http://en.wikipedia.org/wiki/Magnetometer

http://www.birsenyayinevi.com/urun/algilama-ve-olcme-esaslari--prof-dr-sedat-ozsoy.aspx

http://web2.deu.edu.tr/yo/diyyo/okm/MODULES/TERM%20I/T1Module3/DECK/SerimPaker_t1m3d/PUSULA.pdf

http://www.pololu.com/catalog/product/961

http://www.agvindustrial.com/

http://www.robitshop.com/

http://www.pololu.com/catalog/product/961

http://robotus.net/wp-content/uploads/2012/07/SHARP-gp2y0d340k.pdf

http://dlnmh9ip6v2uc.cloudfront.net/verisheets/Sensors/Pressure/A401-force-sensor.pdf

http://www.sparkfun.com/verisheets/Sensors/Pressure/fsrguide.pdf

http://www.arduino.cc

EKLER DİZİNİ

EK 1. ARDUINO MEGA 2560 PİN YERLEŞKESİ

EK 2. ARDUINO MEGA 2560 PİN YERLEŞKE TABLOSU

EK 3. ARDUINO MEGA 2560 DEVRE ŞEMASI

EK 4. ARDUINO UNO PİN YERLEŞKESİ

EK 5. ARDUINO UNO DEVRE ŞEMASI

EK 6. ARDUINO MOTOR KORUMA DEVRE ŞEMASI

EK 7. AĞIRLIK SENSÖRÜ BOYUTLAR VE GERİLİM BÖLÜCÜ DEVRESİ

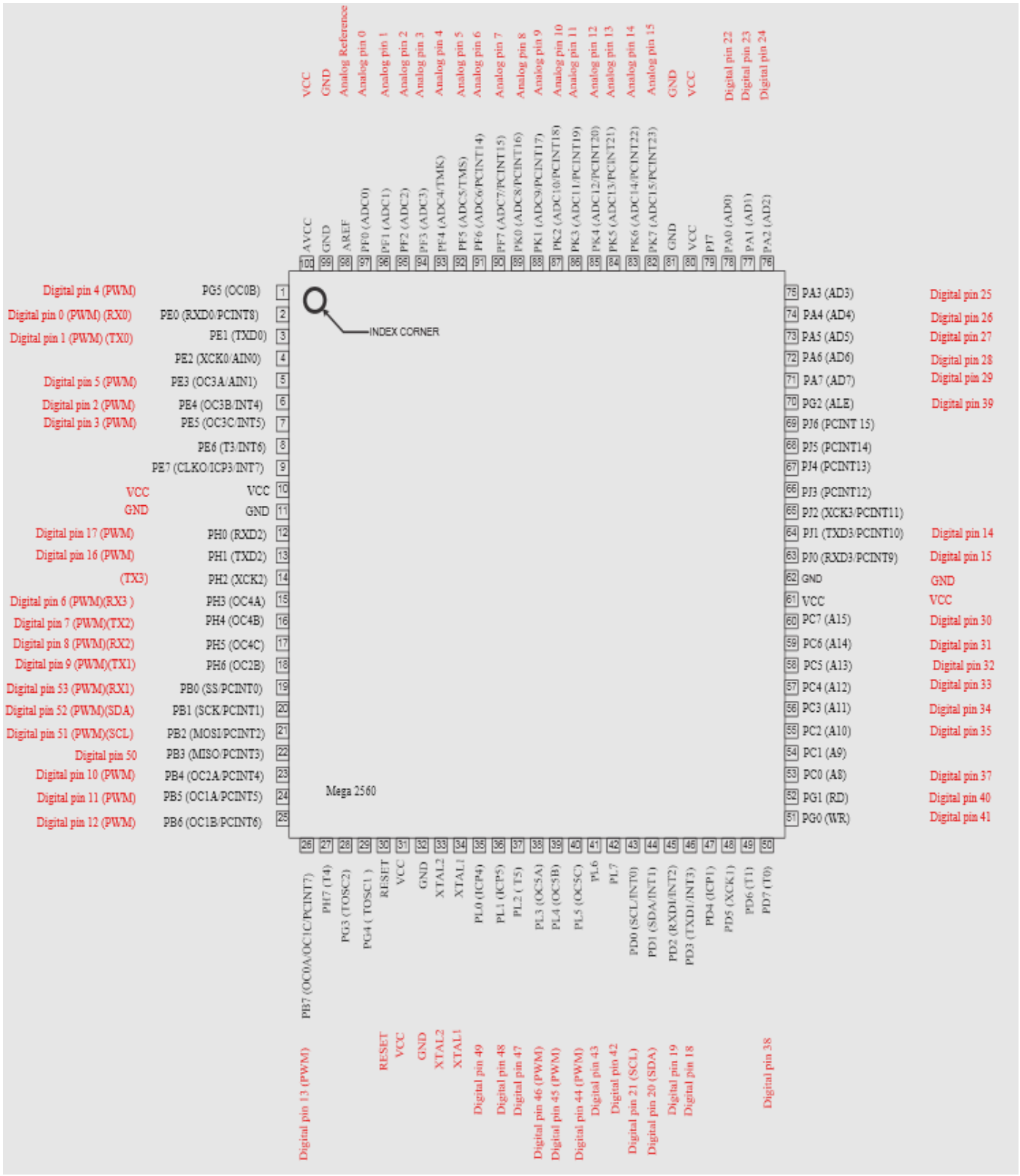
EK 8. ÇİZGİ İZLEME ÜNİTESİNİN PROGRAMLANMASI

EK 9. HABERLEŞME ÜNİTESİNİN PROGRAMLANMASI

EK 10. İSTASYONLARIN PROGRAMLANMASI

EK 11. OTOMASYON İŞLEMİNİN PROGRAMLANMASI

EK 1. ARDUINO MEGA 2560 PİN YERLEŞKESİ



EK 2. ARDUINO MEGA 2560 PİN YERLEŞKE TABLOSU

Pin No	Pin Adı	Pin Yerleşkesi
1	PG5 (OC0B)	Digital pin 4 (PWM)
2	PE0 (RXD0/PCINT8)	Digital pin 0 (PWM) (RX0)
3	PE1 (TXD0)	Digital pin 1 (PWM) (TX0)
4	PE2 (XCK0/AIN0)	
5	PE3 (OC3A/AIN1)	Digital pin 5 (PWM)
6	PE4 (OC3B/INT4)	Digital pin 2 (PWM)
7	PE5 (OC3C/INT5)	Digital pin 3 (PWM)
8	PE6 (T3/INT6)	
9	PE7 (CLKO/ICP3/INT7)	
10	VCC	VCC
11	GND	GND
12	PH0 (RXD2)	Digital pin 17 (PWM)
13	PH1 (TXD2)	Digital pin 16 (PWM)
14	PH2 (XCK2)	(TX3)
15	PH3 (OC4A)	Digital pin 6 (PWM)(RX3)
16	PH4 (OC4B)	Digital pin 7 (PWM)(TX2)
17	PH5 (OC4C)	Digital pin 8 (PWM)(RX2)
18	PH6 (OC2B)	Digital pin 9 (PWM)(TX1)
19	PB0 (SS/PCINT0)	Digital pin 53 (PWM)(RX1)
20	PB1 (SCK/PCINT1)	Digital pin 52 (PWM)(SDA)

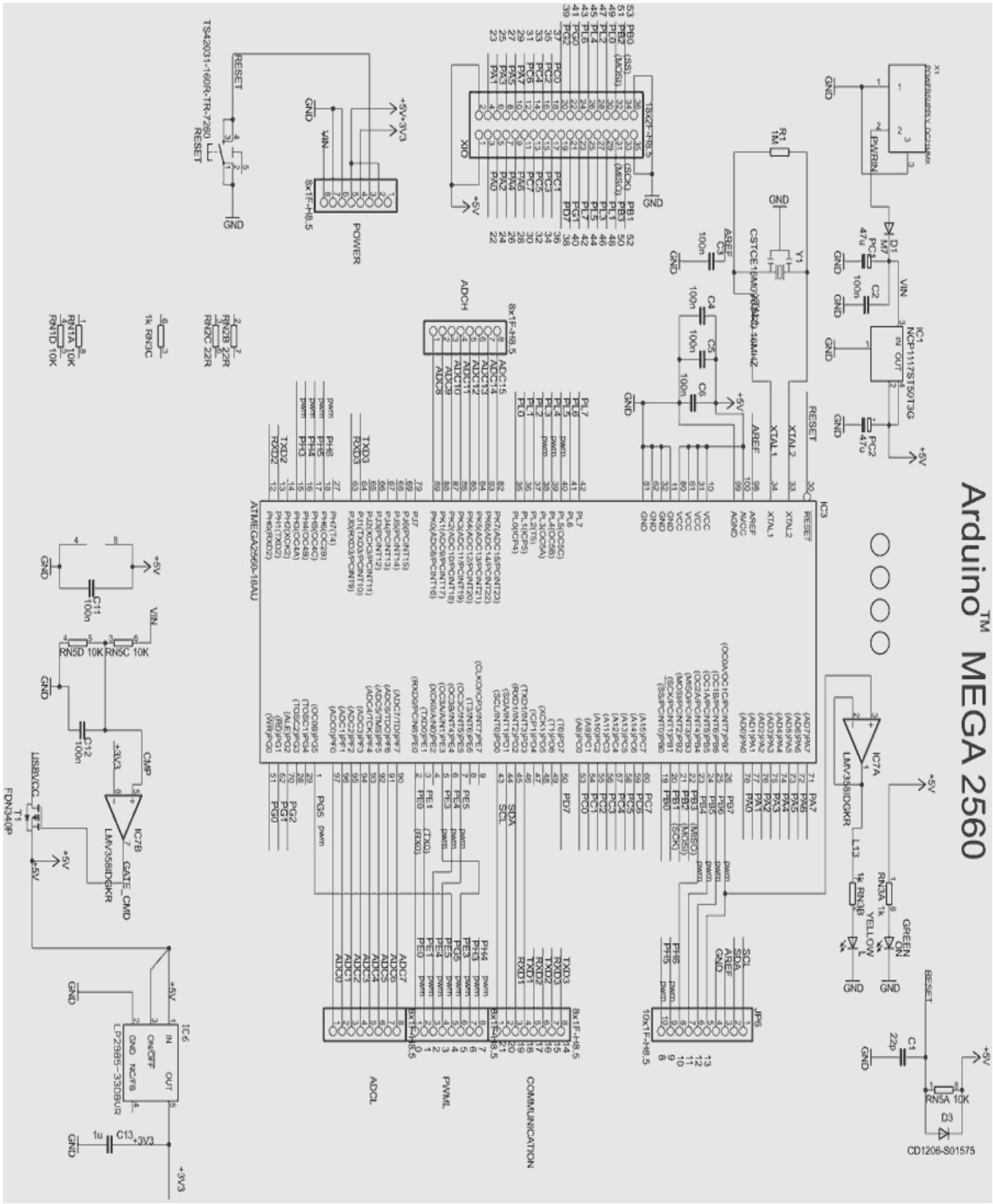
21	PB2 (MOSI/PCINT2)	Digital pin 51 (PWM)(SCL)
22	PB3 (MISO/PCINT3)	Digital pin 50
23	PB4 (OC2A/PCINT4)	Digital pin 10 (PWM)
24	PB5 (OC1A/PCINT5)	Digital pin 11 (PWM)
25	PB6 (OC1B/PCINT6)	Digital pin 12 (PWM)
26	PB7 (OC0A/OC1C/PCINT7)	Digital pin 13 (PWM)
27	PH7 (T4)	
28	PG3 (TOSC2)	
29	PG4 (TOSC1)	
30	RESET	RESET
31	VCC	VCC
32	GND	GND
33	XTAL2	XTAL2
34	XTAL1	XTAL1
35	PL0 (ICP4)	Digital pin 49
36	PL1 (ICP5)	Digital pin 48
37	PL2 (T5)	Digital pin 47
38	PL3 (OC5A)	Digital pin 46 (PWM)
39	PL4 (OC5B)	Digital pin 45 (PWM)
40	PL5 (OC5C)	Digital pin 44 (PWM)
41	PL6	Digital pin 43
42	PL7	Digital pin 42

43	PD0 (SCL/INT0)	Digital pin 21 (SCL)
44	PD1 (SDA/INT1)	Digital pin 20 (SDA)
45	PD2 (RXDI/INT2)	Digital pin 19
46	PD3 (TXD1/INT3)	Digital pin 18
47	PD4 (ICP1)	
48	PD5 (XCK1)	
49	PD6 (T1)	
50	PD7 (T0)	Digital pin 38
51	PG0 (WR)	Digital pin 41
52	PG1 (RD)	Digital pin 40
53	PC0 (A8)	Digital pin 37
54	PC1 (A9)	Digital pin 36
55	PC2 (A10)	Digital pin 35
56	PC3 (A11)	Digital pin 34
57	PC4 (A12)	Digital pin 33
58	PC5 (A13)	Digital pin 32
59	PC6 (A14)	Digital pin 31
60	PC7 (A15)	Digital pin 30
61	VCC	VCC
62	GND	GND
63	PJ0 (RXD3/PCINT9)	Digital pin 15
64	PJ1 (TXD3/PCINT10)	Digital pin 14

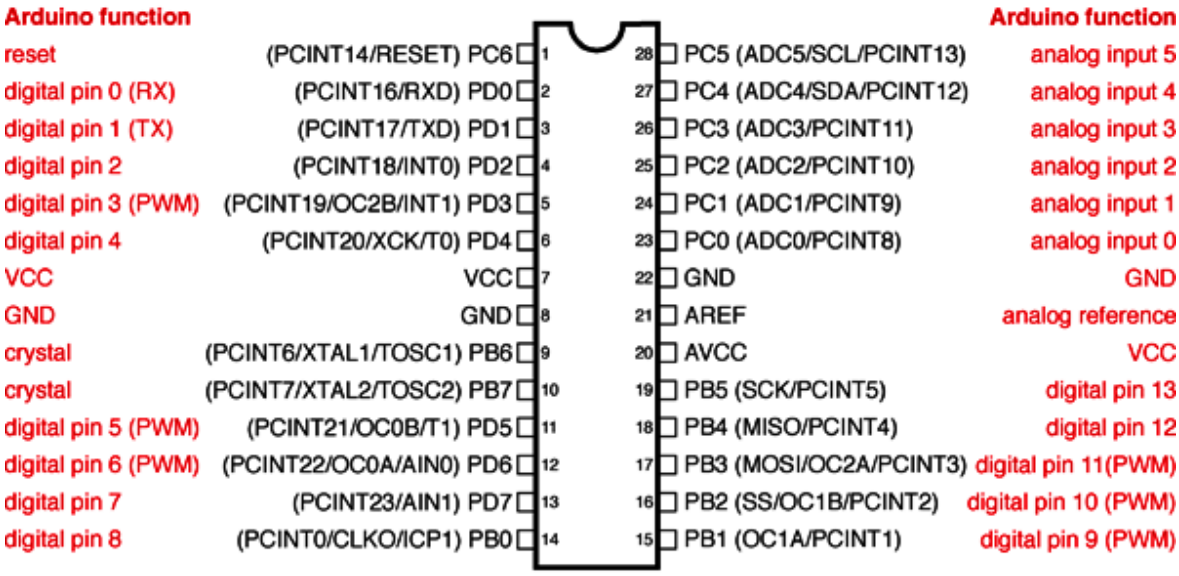
65	PJ2 (XCK3/PCINT11)	
66	PJ3 (PCINT12)	
67	PJ4 (PCINT13)	
68	PJ5 (PCINT14)	
69	PJ6 (PCINT 15)	
70	PG2 (ALE)	Digital pin 39
71	PA7 (AD7)	Digital pin 29
72	PA6 (AD6)	Digital pin 28
73	PA5 (AD5)	Digital pin 27
74	PA4 (AD4)	Digital pin 26
75	PA3 (AD3)	Digital pin 25
76	PA2 (AD2)	Digital pin 24
77	PA1 (AD1)	Digital pin 23
78	PA0 (AD0)	Digital pin 22
79	PJ7	
80	VCC	VCC
81	GND	GND
82	PK7 (ADC15/PCINT23)	Analog pin 15
83	PK6 (ADC14/PCINT22)	Analog pin 14
84	PK5 (ADC13/PCINT21)	Analog pin 13
85	PK4 (ADC12/PCINT20)	Analog pin 12
86	PK3 (ADC11/PCINT19)	Analog pin 11

87	PK2 (ADC10/PCINT18)	Analog pin 10
88	PK1 (ADC9/PCINT17)	Analog pin 9
89	PK0 (ADC8/PCINT16)	Analog pin 8
90	PF7 (ADC7/PCINT15)	Analog pin 7
91	PF6 (ADC6/PCINT14)	Analog pin 6
92	PF5 (ADC5/TMS)	Analog pin 5
93	PF4 (ADC4/TMK)	Analog pin 4
94	PF3 (ADC3)	Analog pin 3
95	PF2 (ADC2)	Analog pin 2
96	PF1 (ADC1)	Analog pin 1
97	PF0 (ADC0)	Analog pin 0
98	AREF	Analog Reference
99	GND	GND
100	AVCC	VCC

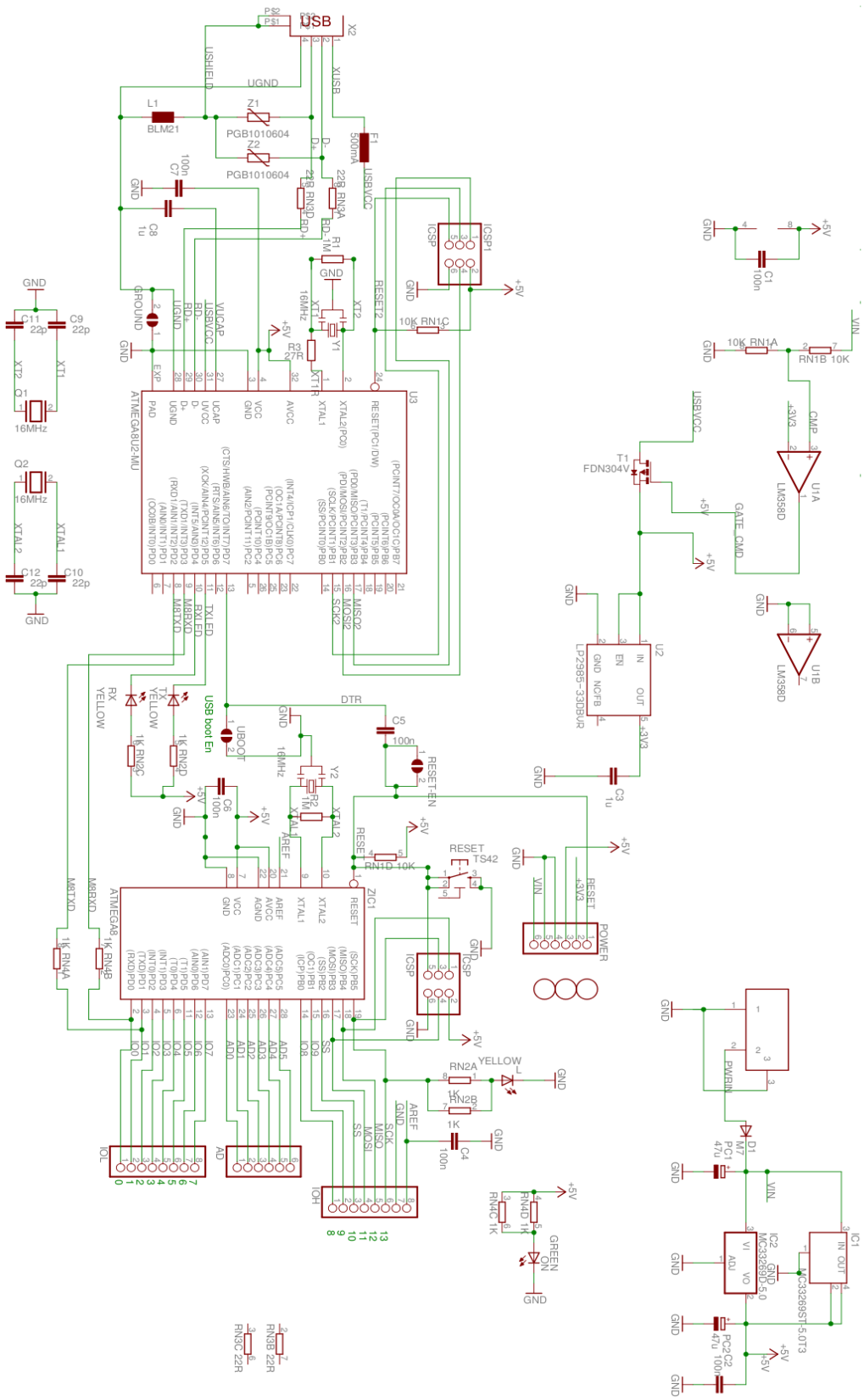
EK 3. ARDUINO MEGA 2560 DEVRE ŞEMASI



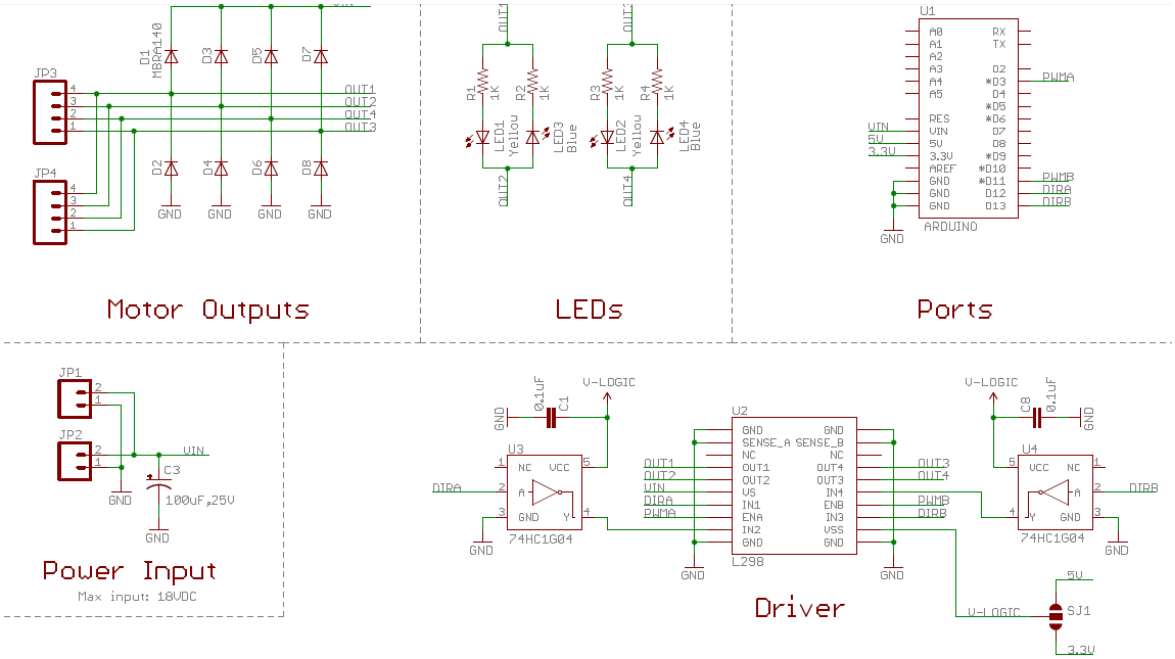
EK 4. ARDUINO UNO PİN YERLEŞKESİ



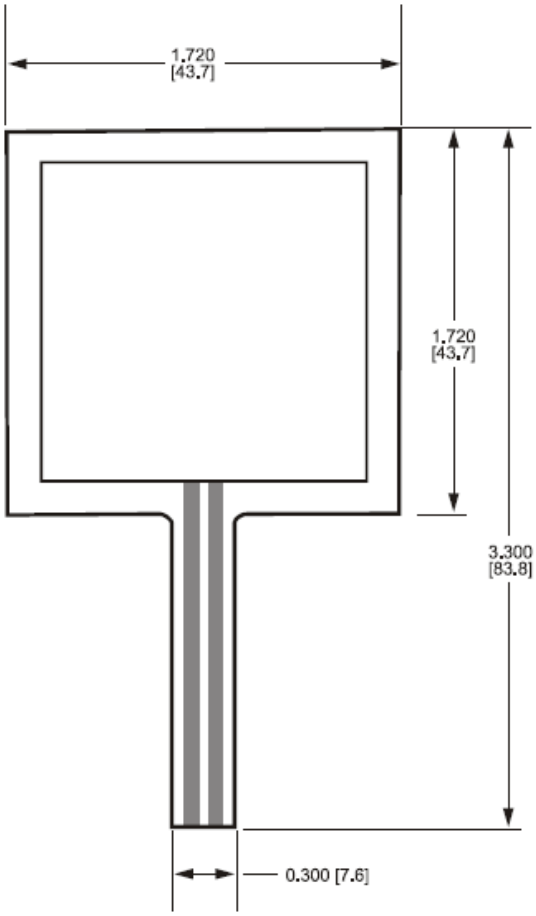
EK 5. ARDUINO UNO DEVRE ŞEMASI

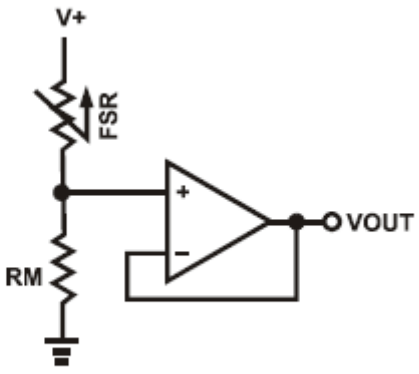


EK 6. ARDUINO MOTOR KORUMA DEVRE ŞEMASI



EK 7. AĞIRLIK SENSÖRÜ BOYUTLAR VE GERİLİM BÖLÜCÜ DEVRESİ





EK 8. ÇİZGİ İZLEME ÜNİTESİNİN PROGRAMLANMASI

```
//*********************************************************************************

void cizgiTakip()

{

  if(digitalRead(distance_sensor) == LOW)  // engel görüldüyse

  {

     setMotors(0,0);  // motoru durdur

     //while(digitalRead(distance_sensor) == LOW); // engel kalkana kadar bekle

     delay(4000);        // 4 sn ye bekle

  }

  // robot çizgi takip ederken bu fonksiyon çağırılır

  //Serial.println("cizgi takip");

  //  value = sensorRead();

  if(sensorRead() > calibration_value)  // tüm sensörler istasyonu gördü

  {

    setMotors(0,0);  // motoru durdur

    //Serial.println(value);

    delay(1000);

    istasyon_numarasi++;        // istasyon numarasını 1 arttır

    if(istasyon_numarasi == 7) // ana istasyona geldi bekle
```

```
  {
    istasyon_numarasi = 0;   // ana istasyonda istasyon numarasını sıfırla
    state = 0;               // ana istasyon durumu ana istasyona gelindi
    durum = 1;               // veri gelmesini bekle
  }
  // istasyon numarası dolum ya da boşaltım istasyon numaralarından birine eşit ise bekle
  else if((istasyon_numarasi == dolum_ist_numarasi) || (istasyon_numarasi == bosaltma_ist_numarasi))
  {
    durum = 3;  // bekle fonksiyonu çağırılacak
  }
  else
  {
    setMotors(M1_DEFAULT_SPEED,M2_DEFAULT_SPEED);      // beklemeden sonra ileri doğru hareket
    while(sensorRead() > calibration_value);        // robot istasyondan çıkana kadar bekle
    delay(500);
  }
 }
```

```
  else

  {

    checkError();                                  // hata kontrol ve robotu çizgi üzerinde
düzeltme işlemleri yapılacak

  }

}

//*****************************************************************************

void checkError()

{

  // Serial.println("check error");

  int error = position - 2000;      // toplam 5 adet sensör kullnıldığı için orta sensör ile
diğerleri arasındaki fark 2000'dir.

  // sensörlerden gelen veri 2000 olduğunda orta sensör çizgi üzerindedir ve hata
sıfırdır. gelen pozisyon orta sensör değerinden çıkartılarak hata bulunur ve bu
hataya göre robot düzeltilir.

  //  Serial.print(position);

  //  Serial.print("**");



  int motorSpeed = KP * error + KD * (error - last_error);     // pd kontrol işlemleri
hatayı işleme sok

  //  Serial.print(motorSpeed);

  //  Serial.print("**");
```

```
  last_error = error;                                  // son işlem yapılan hata değerni bir
önceki hata değeri olarak kaydet

  int leftMotorSpeed = M1_DEFAULT_SPEED + motorSpeed;        // motor hızlarını
hesapla

  //  Serial.println(leftMotorSpeed);

  //  Serial.print("**");

  int rightMotorSpeed = M2_DEFAULT_SPEED - motorSpeed;       // motor hızlarını
hesapla

  //  Serial.println(rightMotorSpeed);

  // set motor speeds using the two motor speed variables above

  setMotors(leftMotorSpeed, rightMotorSpeed);              // motor hızlarını değiştir

}

//*****************************************************************************

boolean istasyonKontrol()

{

  // merkez istasyonu kontrol et, robot merkez istasyonda ise uyarı ledini yak

  // değil ise söndür

  //Serial.println("istasyon kontrol");

  //Serial.println(value);

  if(sensorRead() > calibration_value)  // tüm sensörler istasyonu gördü

  {
```

```
    digitalWrite(led_uyari,HIGH);  // uyarı robot istasyonda
    return true;
  }
  else
  {
    digitalWrite(led_uyari,LOW);  // uyarı robot istasyonda değil
    return false;
  }
}
//*****************************************************************************
void bekle()
{
  // dolum ya da boşaltma istasyonuna gelindiğinde bekle, bekleme süresi 10 snye
  //Serial.println("bekle");
  digitalWrite(led_uyari,HIGH);  // uyarı robot istasyonda
  unsigned long time = millis();
  while(millis() - time < bekleme_suresi);    // doğru istasyonda 10 saniye bekle
  digitalWrite(led_uyari,LOW);  // uyarı robot istasyonda değil
  durum = 2; // bekleme yapıldıktan sonra eski duruma dönülerek robotun ilerlemesi sağlanacak
```

```
  setMotors(M1_DEFAULT_SPEED,M2_DEFAULT_SPEED);  // beklemeden sonra ileri doğru hareket

  delay(750);

}

//*****************************************************************************

unsigned int sensorRead()

{

  // QTR Sensörlerinden değerlerin okunması

  // ayrıntılı bilgi --> http://www.pololu.com/docs/0J13/1

  // kütüphane fonksiyonları --> (http://www.pololu.com/docs/pdf/0J19/QTR_arduino_library.pdf)

  // Serial.println("sensor read");

  int value = 0;

  unsigned int sensors[NUM_SENSORS]; // sensör değerlerinin saklanacağı array

  qtrrc.readCalibrated(sensors);      // kalibre edilmiş değerlere göre karşılaştırmalı olarak sensör değerlerini oku

  for (int i = 0; i < NUM_SENSORS; i++)     // sensör değerleri ile kalibre edilmiş değerleri karşılaştır

  {

    if(sensors[i] > max_val[i])          // karşılaştırma doğru ise value değerini 1 arttır

      value++;
```

```
  }
  position = qtrrc.readLine(sensors);      // pozisyon bilgisini oku
  return value;
}
//*****************************************************************************
void setMotors(int motor1speed, int motor2speed)
{
  // pd kontrol sonucu gelen hata değerine göre motor hızlarını kontrol et ve motorları sür
  //Serial.println("set motors");
  // motor hızı maksimum hızdan büyük olamaz
  if (motor1speed > M1_MAX_SPEED )
    motor1speed = M1_MAX_SPEED; // limit top speed
  if (motor2speed > M2_MAX_SPEED )
    motor2speed = M2_MAX_SPEED; // limit top speed
  //  motor hızı 0 dan düşük olamaz
  if (motor1speed < 0)
    motor1speed = 0; // keep motor above 0
  if (motor2speed < 0)
    motor2speed = 0; // keep motor speed above 0
```

```
  // motorları belirlenen hızlarda sür
  analogWrite(motor1,motor1speed);     // set motor speed
  analogWrite(motor2,motor2speed);     // set motor speed
}
//*****************************************************************************
/*
```

Sensörleri kalibre ederek ortadaki ışık yoğunluğuna göre daha sağlıklı ölçüm yapması sağlanabilir. Bu sebeple kırmızı led yandıktan sonra led sönene kadar robot çizgi üzerinde ileri geri sağa sola ve çizgiden dik geçecek şekilde hareket ettirlmelidir. Kırmızı led söndükten sonra kalibrasyon bitmiş demektir ve robot merkez istasyona yerleştirilmelidir.

```
 */
void calibrationQtr()
{
  // yansıma sensörlerinin kalibaryonu bu fonksiyonda yapılmaktadır.
  //Serial.println("calibration qtr");
  //Serial.println("Manuel kalibrasyona girdi");
  int i;
  for (i = 0; i < 250; i++)  // the calibration will take a few seconds
  {
    qtrrc.calibrate(QTR_EMITTERS_ON);          // IR ledler yanarken sensörlerden değer oku
```

```
  delay(20);
 }

 if (DEBUG) // if true, generate sensor dats via serial output
 {
  //Serial.println("min kablibrasyon degerleri");
  for (int i = 0; i < NUM_SENSORS; i++)
  {
   min_val[i] = qtrrc.calibratedMinimumOn[i];      // kalibrasyon sonucu minimum değerleri bul
  }
  //Serial.println();
  //Serial.println("max kablibrasyon degerleri");
  for (int i = 0; i < NUM_SENSORS; i++)
  {
   max_val[i] = qtrrc.calibratedMaximumOn[i];      // kalibrasyon sonucu maksimum değerleri bul
  }
  //Serial.println();
  //Serial.println();
  for (int i = 0; i < NUM_SENSORS; i++)
```

```
    {
      // maksimum ve minimum değerleri 0 ile 1000 arasına daralt
      // map fonksiyonu için --> http://arduino.cc/en/Reference/Map
      max_val[i] = map(max_val[i],0,2500,0,1000);
      min_val[i] = map(min_val[i],0,2500,0,1000);
      // aradki farkın 0.6 katı eşik değeri bu değer geçildiğinde sensörler
      // istasyonu görüyor demektir.
      max_val[i] = (max_val[i] - min_val[i]) * 0.6;
      //max_val[i] = max_val[i] * 0.65;
      //Serial.println(max_val[i]);
    }
  }
}
//*********************************************************************************
void merkezIstasyon()
{
  /* robot merkez istasyonda işken veri gelip robot istasyonlara doğru harekete başladığında  istasyon geçilene kadar robotun tekrar durmamaması bu fonksiyon ile sağlanmaktadır.
   */
  //Serial.println("merkez istasyon");
```

```
  setMotors(M1_DEFAULT_SPEED,M2_DEFAULT_SPEED);

  while(sensorRead() > calibration_value);

  state = 1;

  digitalWrite(led_uyari,LOW);  // uyarı robot istasyonda değil

  delay(500);

}

//*****************************************************************************
```

EK 9. HABERLEŞME ÜNİTESİNİN PROGRAMLANMASI

Haberleşme ünitesinin programlanmasında aktif olarak 2 devre rol almaktadır. Birinci devre robot üzerindeki kablosuz iletişimi sağlayan modül, ikinci devre ise istasyonlardaki kablosuz iletişimi sağlayan modüldür. Haberleşme ünitesinin programlanması aşağıdaki gibi gerçeklenmiştir.

```
//*****************************************************************************

void veriBekle()

{

  // haberleşme protokolü toplam 6 byte üzerinden

  if(Serial.available() > 5)    // istasyondan veri geldiyse toplam 6 byte

  {

    for(int i = 0; i < 6; i++)    // gelen 6 byte veriyi oku

    {

      receive_veri[i] = Serial.read();    // istasyondan gelen verileri oku

    }

    if((receive_veri[start_byte] == start_char) && (receive_veri[end_byte] == end_char))  // gelen verilerden start ve end değerlerini kontrol et

    {

      if((receive_veri[receiver_address] == my_address))    // alıcı adresini kontrol et

      {

        // transmitter_address byte'ı sadece istasyon yazılımlarında kontrol edilcek

        sendACK();  // alınan veri paketi doğru istasyona alındı bilgisi gönder
```

```
      // alınan veriler karakter olarak gönderildiği için karşılaştırma amaçlı

      // '0' karakteri karşılığı değerden çıkartılır. örneğin istasyondan gelen veride 3 bulunuyorsa

      // bu değer işlemci tarafından 51 olarak alınır ve kendisinden 48 çıkartıldığında gerçek değer

      // bulunur.

      dolum_ist_numarasi = receive_veri[dolum_istasyonu] - 48;          // hangi istasyondan dolum yapılacak

      bosaltma_ist_numarasi = receive_veri[bosaltma_istasyonu] - 48;    // hangi istasyona yük boşaltılacak

      durum = 2;    // robot veriyi aldı ve hareket başladı

    }

    else

    {

      sendNack();  // verideki alıcı adresi yanlış

    }

  }

  else  // start ve end karakterleri yanlış

  {

    sendNack();  // yanlış veri geldi istasyon tekrar veri göndersin

  }
```

```
  }
}
//*****************************************************************************
void sendACK()
{
  // veri doğru alındı bilgisi bu fonksiyonda gönderilir
  send_veri[start_byte] = '?';
  send_veri[receiver_address] = receive_veri[transmitter_address];  // son veri gönderen istasyona veri gönderilecek
  send_veri[transmitter_address] = my_address;        // veri gönderenin robot olduğunu belirlenecek
  send_veri[3] = 'o';      // istasyondan gönderilen veri doğru ise
  send_veri[4] = 'k';      // geri bildirim için "ok" gönderilir
  send_veri[end_byte] = '*';
  veriGonder();       // istasyona istek alındı bilgisi gönder
}
//*****************************************************************************
void sendNack()
{
  // veri yanlış alındı bilgisi bu fonksiyonda gönderilir
  send_veri[start_byte] = '?';
```

```
  send_veri[receiver_address] = receive_veri[transmitter_address];    // son veri gönderen istasyona veri gönderilecek

  send_veri[transmitter_address] = my_address;          // veri gönderenin robot olduğunu belirlenecek

  send_veri[3] = 'n';       // istasyondan gönderilen veri doğru ise

  send_veri[4] = 'o';       // geri bildirim için "ok" gönderilir

  send_veri[end_byte] = '*';

  veriGonder();

}

//*****************************************************************************

void veriGonder()

{

  // hazırlanan gönderilecek veri paketi bu fonksiyonda byte byte gönderilir

  for(int i = 0; i < 6; i++)

    Serial.print(send_veri[i]);

  emptyBuffer();  // seri haberleşme buffer'ı temizle

}

//*****************************************************************************

void emptyBuffer()

{

  // istenmeyen veriler bufferdan bu fonksiyon ile silinir.
```

```
  //Serial.println("garbage in");
  int garbage = 0;
  while( garbage >= 0)
    garbage = Serial.read();
  //Serial.println("garbage out");
}
//*************************************************************************
```

EK 10. İSTASYONLARIN PROGRAMLANMASI

İstasyonların progranlanması esnasında en önemli durum; istasyonlardan gönderilecek verilerin robot tarafından algılanabilmesi ve bu verilerin kendi içerisinde ayırt edilebilir olması durumuydu. Bu noktada kullandığımız program ile istasyonlara farklı kimlikler belirleyerek robotun istasyonlardan gelen verileri tanıyabilmesi ve bu verilere göre otonom hareket geliştirmesi sağlanmıştır.

```
/*

Xbee ile haberleşilirken 2 ve 3. dijital pinler RX,TX pinleri olarak kullanılacaktır. Bu sayede xbee ler çıkartılmadan Arduino'nun programlaması ve Xbee'ler aktif iken PC ile haberleşmesi sağlanmış olur. USB kablo takılı iken serial.print ya da benzeri komutlar ile işletilen komutlar serial monitörde görülmektedir. PC ile bağlantı yoksa bu komutlar işletilecek ama görülemeyecektir.

 */

#include <SoftwareSerial.h>     // 2 ve 3. pinlerin rx,tx olarak kullanılması için gerekli kütüphane

// http://arduino.cc/en/Reference/SoftwareSerial adresinden detaylı bilgi alınabilir

// sensör istasyonlarında ki xbee'ler zigbee router at olarak ayarlı, robot üzerindeki xbee'ler zigbee coordinator olarak ayarlı

// pand id 7788

//SoftwareSerial mySerial(2, 3); // RX, TX // 3 numaralı kart için

SoftwareSerial mySerial(3, 2);  // 1 ve 2 numaralı kart için

//*************************************************************************

// protokol byte'ları

//
```

||start_byte|receiver_address|transmitter_Address|dolum_istasyonu|bosaltma_istasyonu|end_byte||

```
char start_byte = 0;            // başlangıç byte'nın arrayda tutulduğu byte'ın indeksi

char receiver_address = 1;      // alıcı adresinin arrayda tutulduğu byte'ın indeksi

char transmitter_address = 2;   // verici adresinin arrayda tutulduğu byte'ın indeksi

char dolum_istasyonu = 3;       // dolum istasyonunun adresinin arrayda tutulduğu byte'ın indeksi

char bosaltma_istasyonu = 4;    // bosaltma istasyonu adresinin arrayda tutulduğu byte'ın indeksi

char end_byte = 5;              // bitiş byte'nın arrayda tutulduğu byte'ın indeksi

char receive_veri[6];           // alınan verilerin tutulduğu array

//char send_veri[] = {

//  '?','0','3','3','6','*'};              // gönderilecek verilerin tutulduğu array her istasyon için bu array değiştirilecek

char send_veri[] = {

  '?','0','1','1','4','*'};              // gönderilecek verilerin tutulduğu array her istasyon için bu array değiştirilecek

//  char send_veri[] = {

// '?','0','2','2','5','*'};              // gönderilecek verilerin tutulduğu array her istasyon için bu array değiştirilecek

char my_address = '1';                  // sensor istasyonu network adresi her istasyon için bu değer değişecek

char start_char = '?';
```

```
char end_char = '*';

//*****************************************************************************

int led_pin = 13;                       // sensör ölçüm uyarı ledi, led yandığında FSR sensör üzerinde yük var demektir

int analog_pin = 0;                 // FSR den ölçüm yapılan analog pin A0

int full_value = 5;                       // FSR den gelen değer 5 ten büyük ise FSR üzerinde yük var istasyon dolu demektir

int empty_value = 5;                       // FSR den gelen değer 5 ten küçük ise FSR üzerinde yük yok istasyon boş demektir

char coordinator_address = '0';        // robot adresi

char process_status = 0;              // sistem durum degeri

/*

status = 0 sensör degeri okunacak veri gönderilmedi

 status = 1 sensör degeri okundu, eşik değeri geçildi

 status = 2 robota veri gönderildi

 status = 3 robottan alındı bilgisi geldi, robot beklenecek

 */

unsigned long sensorValue = 0;       // analog pinden okunan sensör değeri bu değişkende tutulacak

//*****************************************************************

//*****************************************************************
```

```
//**********************MAIN FUNCTIONS**************************

//**************************************************************

//**************************************************************

void setup()

{

  Serial.begin(9600);        // PC ile seri haberleşme aktif baudrate 9600

  pinMode(led_pin,OUTPUT);   // uyarı ledi output

  delay(1000);

  mySerial.begin(9600);      // xbee ile seri haberleşme aktif baudrate 9600

}

//**************************************************************

//**************************************************************

void loop()

{

  switch(process_status)

  {

  case 0:                    // sistem başlatıldı sensörden ölçüm alınacak

    olcum();                 // ölçüm fonksiyonu

    if(mySerial.available()>0)    // robota veri gönderilmeden önce seri haberleşme
buffer'ını boşalt
```

```
    emptyBuffer();            // seri haberleşme buffer'ını temizle
   break;
  case 1:                     // istasyon boşaldı robota veri gönder
   sendMsg();
   break;
  case 2:                     // robota veri gönderildikten sonra robottan ok gelmesini bekle
/*
robota veri gönderildikten sonra robot gelen veriyi kontrol eder ve aldığı veri doğru ise istasyona veriyi doğru aldığını belirtmek için ack bilgisi gönderir. bu bilgi 6 byte olup ||start_byte|receiver_address|transmitter_Address|"o"|"k"|end_byte|| şeklinde gelir. Eğer istasyondan gelen veri hatalı ise robot yine 6 byte olacak şekilde ||start_byte|receiver_address|transmitter_Address|"n"|"o"|end_byte|| verisini gönderir ve istasyondan doğru veriyi göndermesini ister.
*/
   receiveMsg();
   if(process_status != 3)
     olcum();
   break;
  case 3:
   while(analogRead(analog_pin) < full_value)  // yükleme yapılana kadar bekle
      Serial.println(analogRead(analog_pin));
```

```
    process_status = 0;
    break;
  default:
    process_status = 0;
    break;
  }
}
//*****************************************************************
//*****************************************************************
//****************************FUNCTIONS****************************
//*****************************************************************
//*****************************************************************
/*
FSR sensörünün değeri bu fonksiyonda okunmaktadır. Gelen değer 5'ten büyük ise sensör üzerinde yük var istasyon dolu anlamındadır. Gelen değer 5'ten küçük olduğunda istasyon boş anlamındadır. Sensörden alınan değer 1000 kere okunarak ortalaması alınmıştır. Bu şekilde sensördeki gürültüler sonucu ortaya çıkacak yanlış ölçümler filtrelenmiş olmaktadır. Okunan ortalama değer 5'ten büyük ise uyarı ledi yakılmakta, aksi takdirde led söndürülmektedir.
*/
void olcum()
{
```

```
  sensorValue = 0;                          // ölçüm yapılmadan önce değişkeni sıfırla

  for(int i = 0; i < 1000; i++)             // analog pinden gelen sensör değerini 1000
kere oku ve

    sensorValue += analogRead(analog_pin);      // sensorValue değişkenine ekle

  sensorValue = sensorValue / 1000;           // 1000 değerin ortalamasını al

  Serial.println(sensorValue);                // değeri pc ye gönder ( USB kablo ile pc
ye takılı olduğunda serial monitörde değer gözükecek)

  if(sensorValue < empty_value)               // sensör değeri 5 ten küçük

  {

    process_status = 1;                       // status = 1 robota veri gönderilecek

    digitalWrite(led_pin,LOW);                // ledi söndür

  }

  else                                        // sensör değeri 5 ten büyük

  {

    process_status = 0;                       // ölçüm almaya devam et

    digitalWrite(led_pin,HIGH);               // ledi yak

  }

}

//*************************************************************

/*

Robota gönderilecek veri paketi sabittir. sizeof ile paket boyutu belirlenerek boyut
```

büyüklüğü kadar byte robota gönderilecektir.

```
  sizeof(send_veri) = 6

*/

void sendMsg()

{

  for(int i = 0; i < sizeof(send_veri); i++)     // gönderilecek veri 6 byte

    mySerial.write(send_veri[i]);               // her byte ı tek tek gönder

  process_status = 2;                           // veri gönderildi cevap bekleniyor

  Serial.println("veri gönderildi");            // veri gönderildi pc'ye bilgi gönder

}

//*****************************************************************

/*

 robota veri gönderildikten sonra robottan alındı bilgisinin gelmesi beklenir. Alındı bilgisi 6 byte'tır. alındı bilgisi gelemediği sürece ya da yanlış veri geldi bilgisi robot tarafından gönderildiği sürece 5 sn ye bir veri paketi tekrar gönderilir.

sizeof(send_veri) = 6

*/

void receiveMsg()

{

  int timeout = 5000;                 // 5 sn de bir veri gönderilecek
```

```
  unsigned long time = millis();        // şu anki süreyi time değişkenine at

  // millis() fonksiyonu için --> http://arduino.cc/en/Reference/Millis

  while((mySerial.available() < sizeof(receive_veri)) && (millis() - time < timeout));
// timeout süresi dolana kadar ya da veri gelene kadar bekle

  if(mySerial.available() > sizeof(receive_veri) - 1)  // robottan veri geldiyse

  {

    // robottan gelen verileri oku ve receive veri array'ine al

    for(int i = 0; i < sizeof(receive_veri); i++)

    {

      receive_veri[i] = mySerial.read();

      Serial.print(receive_veri[i]);

    }

    Serial.println();

    // veri kontrolü bu kısımda yapılıyor

    if(receive_veri[start_byte] == start_char && receive_veri[end_byte] == end_char)
// ilk ve son karakter doğru mu?

    {

      if(receive_veri[receiver_address]  ==  my_address  &&
receive_veri[transmitter_address] == coordinator_address)  // alıcı adresi doğru
mu? gönderen robot mu?

      {
```

```
if(receive_veri[dolum_istasyonu] == 'o' && receive_veri[bosaltma_istasyonu] == 'k')  // robottan doğru alındı bilgisi ack geldi

{

  process_status = 3;      // robottan alındı bilgisi geldi robot bekleniyor

  Serial.println("veri dogru alindi");

}

else if(receive_veri[dolum_istasyonu] == 'n' && receive_veri[bosaltma_istasyonu] == 'o')  // robottan yanlış alındı bilgisi nack geldi

{

  Serial.println("veri yanlis alindi adres hata");

  emptyBuffer();           // buffer'ı boşalt

  process_status = 1;      // veri paketinde hata var, buffer temizlenip tekrar değer gönderilecek

}

else  // dolum ve boşaltım istasyonlarnıda hata var

{

  Serial.println("veri yanlis alindi adres hata");

  emptyBuffer();           // buffer'ı boşalt

  process_status = 1;      // veri paketinde hata var, buffer temizlenip tekrar değer gönderilecek

}
```

```
      }
      else  // adres hatalı
      {
        Serial.println("veri yanlis alindi adres hata");
        emptyBuffer();            // buffer'ı boşalt
        process_status = 1;       // veri paketinde hata var, buffer temizlenip tekrar değer gönderilecek
      }
    }
    else  // start ve end karakter hatalı
    {
      Serial.println("veri yanlis alindi adres hata");
      emptyBuffer();
      process_status = 1; // veri paketinde hata var, buffer temizlenip tekrar değer gönderilecek
    }
  }
  else  // timeout
  {
    Serial.println("timeout");
    emptyBuffer();
```

```
    process_status = 1;  // timeout tekrar veri gönder
  }
}
/*
 robottan gelen veriler broadcast şeklinde her istasyona gönderildiği için adres değeri tutmayan diğer veriler bu fonksiyon ile buffer dan silinir.
 */
void emptyBuffer()
{
  Serial.println("garbage in");
  int garbage = 0;
  while( garbage >= 0)
    garbage = mySerial.read();
  Serial.println("garbage out");
}
```

EK 11. OTOMASYON İŞLEMİNİN PROGRAMLANMASI

Robotun ve istasyonların ayrı ayrı programlanmasından sonra oluşan tüm bu verilere göre otonom çalışma ilişkilerini sağlaması amacı ile otomasyon bölümü progranlanmıştır. Bu noktada otomasyon bölümünün amaca diğer 3 programdanda anlık durumlara göre kontroller ve bağlantılar ile verileri alarak sistemin nasıl hareket edeceğine karar vermektir.

```
#include <QTRSensors.h>          // qtr-8rc yansıma sensörleri için kullanılan kütüphane

//*****************************************************************************

// motor hız, pd kontrol katsayıları ve sensör değişkenlerinin tanımlanması

#define KP .33

#define KD 3.3

#define M1_DEFAULT_SPEED  84 // minimum motor hızı

#define M2_DEFAULT_SPEED  84 // minimum motor hızı

#define M1_MAX_SPEED  144    // maksimum motor hızı

#define M2_MAX_SPEED  144    // maksimum motor hızı

#define MIDDLE_SENSOR  3     // orta sensör

#define NUM_SENSORS  5       // kullanılan sensör sayısı

#define TIMEOUT  2500        // waits for 2500 us for sensor outputs to go low

#define EMITTER_PIN  14      // emitter is controlled by digital pin 2

#define DEBUG  1             // set to 1 if serial debug output needed

//*****************************************************************************
```

```
/* qtr kütüphanesi aktifleştirme ve istenilen değişken değerlerinin gönderilmesi
8'li diziden 5 adet sensör kullnılmıştır. bu sensörler sensör kartı üzerindeki
2,3,4,5,6 numaralı sensörlerdir. 4 numaralı sensör orta sensördür ve çizgi izlerken
düzeltmeler bu sensöre göre yapılmaktadır.
*/
QTRSensorsRC qtrrc((unsigned char[]) {
  4,5,6,7,8}
,NUM_SENSORS, TIMEOUT, EMITTER_PIN);
//*****************************************************************************
// Motor sürücü ve kullanılan diğer pinlerin tanımlamaları
int motor1 = 11;            // 1. motor pini
int motor2 = 3;             // 2. motor pini
int motor1_direction = 12;  // 1. motor yön pini
int motor2_direction = 13;  // 2. motor yön pini
int led_pin = A1;           // kalibrasyon uyarı ledi kırmızı led
int led_uyari = A2;         // istasyon uyari ledi mavi led
int distance_sensor = A3;   // uzaklık sensörü a3 pinine takılacak
//*****************************************************************************
// sensör pd kontrol ile kullanılan değişkenlerin tanımlanması
int last_error = 0;  // pd ile düzeltme kısmında bir önceki hata değeri bu değişkende
```

```
saklanacaktır

//int last_proportional = 0;

//int integral = 0;

int position;        // sensörden okunan konum bu değişkende saklanmaktadır.

//*****************************************************************************

// istasyon ID'leri

int ana_pozisyon = 0;

int dolum_1 = 1;

int dolum_2 = 2;

int dolum_3 = 3;

int istasyon_3 = 4;

int istasyon_2 = 5;

int istasyon_1 = 6;

//*****************************************************************************

// pozisyon için değişkenler

int istasyon_numarasi = 0;

int dolum_ist_numarasi = 0;

int bosaltma_ist_numarasi = 0;

//*****************************************************************************

// durum = 0 sistem çalışmaya başlamadı
```

```
// durum = 1 sistem çalışmaya başladı veri bekliyor

// durum = 2 veri alındı, hareket başladı

// durum = 3 istenilen istasyonlardan birisine gelindi

int durum = 0;  // process durumlarının tutulduğu değişken

int state = 0;

//*****************************************************************************

// protokol byte'ları

//
||start_byte|receiver_address|transmitter_Address|dolum_istasyonu|bosaltma_ista
syonu|end_byte||

char start_byte = 0;            // başlangıç byte'nın arrayda tutulduğu byte'ın indeksi

char receiver_address = 1;      // alıcı adresinin arrayda tutulduğu byte'ın indeksi

char transmitter_address = 2;   // verici adresinin arrayda tutulduğu byte'ın indeksi

char dolum_istasyonu = 3;        // dolum istasyonunun adresinin arrayda tutulduğu
byte'ın indeksi

char bosaltma_istasyonu = 4;     // bosaltma istasyonu adresinin arrayda tutulduğu
byte'ın indeksi

char end_byte = 5;              // bitiş byte'nın arrayda tutulduğu byte'ın indeksi

char receive_veri[6];           // alınan verilerin tutulduğu array

char send_veri[6];              // gönderilecek verilerin tutulduğu array

char my_address = '0';          // robot network adresi
```

```
char start_char = '?';

char end_char = '*';

//*****************************************************************************

int bekleme_suresi = 10000;     // doğru istasyona gelindiğinde bekleme süresi

int calibration_value = 4;          // istasyon tanıma için aynı anda çizgiyi görecek sensör adedi

unsigned int max_val[NUM_SENSORS];  // calibrasyon değerleri için kullanılan arrayler

unsigned int min_val[NUM_SENSORS];

//*****************************************************************************

//*****************************************************************************

void setup()

{

  Serial.begin(9600);     // xbee ile haberlşme aktif baudrate 9600

  // output pinleri

  // motor pwm, yön ve uyarı ledlerinin output olarak tanımlanması

  pinMode(motor1_direction,OUTPUT);

  pinMode(motor2_direction,OUTPUT);

  pinMode(motor1,OUTPUT);

  pinMode(motor2,OUTPUT);

  pinMode(led_pin,OUTPUT);
```

```
  pinMode(led_uyari,OUTPUT);
  // motorlar sadece ileri yönde hareket eder, geri dönüş yoktur
  digitalWrite(motor1_direction,HIGH);  // motor yönü ileri
  digitalWrite(motor2_direction,HIGH);  // motor yönü ileri
  delay(1000);
  digitalWrite(led_pin,HIGH);  // ledi yak, kalibrasyon başladı
  calibrationQtr();            // izleme sensörleri kalibrasyonu
  digitalWrite(led_pin,LOW);   // kalibrasyon başladı
  setMotors(0,0);              // motoru durdur
  delay(10000);                // kalibrasyondan sonra 10 snye robotun yerleştirilmesi için bekle
  // robotun ana pozisyonda olup olmadıgı kontrol ediliyor
  // robot ana istasyona gelene kadar program burada bekleyecektir.
  if(istasyonKontrol())  // motor ana istasyondan başlatılmalı
  {
    durum = 1;           // ana pozisyon doğrulandı sistem çalışmak için istasyonlardan veri bekliyor
  }
  else
  {
    //Serial.println("Robot baslangic konumunda degil");
```

```
    //Serial.println("Lutfen robotu baslangic konumuna getiriniz...");
    while(istasyonKontrol() == false);     // robot başlangıç konumuna getiriline kadar bekle
    durum = 1;
    //Serial.println("Robot baslangic konumunda getirildi");
    //Serial.println("Sistem baslatiliyor...");
  }
}
//*****************************************************************************
//*****************************************************************************
void loop()
{
  switch(durum)
  {
  case 1:
    veriBekle();     // bu fonksiyon içerisinde robot ana pozisyonda beklerken istasyonlardan gelecek veriyi dinler
    break;
  case 2:
    if(istasyon_numarasi == 0 && state == 0)
      merkezIstasyon();
```

```
    cizgiTakip();  // gelen veri doğrultusunda istenilen istasyonlara git

    if(Serial.available() > 0)  // merkez istasyonda bekleme durumu hariç herahngi bir veri alımı kabul edilemez.

    //gelen veriler bu fonksiyon ile silinir

      emptyBuffer();

    break;

  case 3:

    bekle();      // doğru istasyona gelindiğinde bekle

    break;

  default:

    break;

  }

}

//*************************************************************************************
```

ÖZGEÇMİŞ

Adı Soyadı : Kıvanç İREN

Doğum Yeri : Ankara

Doğum Yılı : 1984

Medeni Hali : Evli

E-Posta : kivanciren@gmail.com

Eğitim ve Akademik Durumu:

Lise : Mimar Sinan Anadolu Lisesi (1998-2002)

Lisans : Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Makine Mühendisliği Bölümü(2002-2006)

Yabancı Dil: İngilizce, Fransızca

İş Tecrübesi:

Yükseliş Asansör Ltd. Şti. Ankara, (2006-2008)

LNS SA. Ankara, İsviçre, (2008-2010)

Oyak Renault Otomobil Fabrikaları A.Ş., Bursa,(2010-2013)