T.C.

ERCİYES ÜNİVERSİTESİ

MÜHENDİSLİK FAKÜLTESİ

MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ

MEKATRONİK LABORATUVARI – II

MOBİL ROBOT YÖRÜNGE KONTROL DENEYİ

DENEY SORUMLUSU

Arş. Gör. Mehmet Safa BİNGÖL

ŞUBAT 2023

KAYSERİ

MOBİL ROBOT KONTROLÜ

1. GİRİŞ

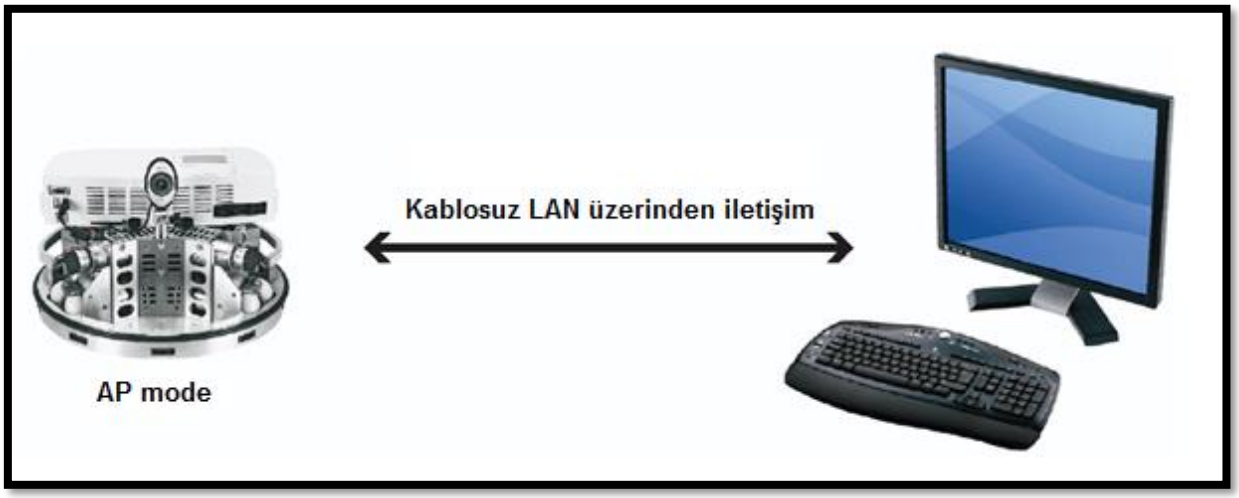
Mobil robotlar, sürekli olarak insan müdahalesi olmadan otomatik olarak, çalışma ortamlarına giderek istenilen görevleri gezgin olarak gerçekleştiren robotlardır. İsminden de anlaşılacağı gibi bu robotların en önemli avantajı, birçok endüstriyel robotun aksine, çalışma ortamlarında serbest olarak hareket edebilmeleri, hareket hacimlerinin büyük olması, hatta gerekirse statik ve dinamik çevre şartlarında engellerden kaçınarak istenilen yere gidebilmeleri ve beklenen görevleri yerine getirmeleridir. Mobil robotların bu gezginlik özellikleri uygulama alanında da geniş bir yelpaze sunmaktadır. Bu yüzden de uzun yıllar boyunca mobil robotlar araştırmacıların ilgi odağı olmuştur. 1960'ların sonlarında başlayan mobil robot çalışmaları, elektronik ve bilgisayar teknolojisindeki gelişmelerle de ivme kazanarak devam etmektedir.

Mobil robotlar üzerine yapılan araştırmalar daha çok harita çıkarma, yörünge takibi, hız ve tork kontrolü, engele çarpmadan hareketini gerçekleştirme, optimum yörünge ile hedefe ulaşma, hedef takibi gibi uygulamalar üzerinde yoğunlaşmaktadır. Görüldüğü üzere bir mobil robotun istenilen hedefe ulaşması ya da verilen bir yörünge boyunca hareket edebilmesi, bir hedefi takip edebilmesi, şayet gerekiyorsa bulunduğu ortamı algılayabilmesi, mobil robotlar için hedef araştırma geliştirme alanlarıdır.

2. ROBOTINO MOBİL ROBOTU

Bu deney çalışmasında Festo Didactic firmasının geliştirdiği bir mobil robot platformu olan Robotino mobil robotu kullanılacaktır. Bir mobil robot sisteminde olması öngörülen tüm sensörlerin, eyleyicilerin ve ara yüz yazılımlarının hazır olduğu kompakt bir yapıda tasarlanmıştır. Ayrıca harici modüller ile farklı görevler için özelleştirilebilmektedir. Robotino'nun en önemli diğer bir avantajı ise %100 açık-kaynak bir robot olması ve robot üzerinde yürütülen kaynak kodun ücretsiz olarak svn.openrobotino.org sitesinden temin edilebilmesidir. Robotino programlanmasında geniş bir dil yelpazesi sunmaktadır. C, C++, Java, .Net, Matlab, LabVIEW ve Microsoft Robotics Developer Studio ile yazılım geliştirilebilmektedir. Ayrıca "Robot Operating System (ROS)" için de desteği bulunmaktadır. Aşağıdaki Şekil 1'de Robotino'nun genel görünümü yer almaktadır. Ayrıca Şekil 2'de Robotino'nun üzerindeki access point sayesinde bilgisayar ile kablosuz haberleşebildiği gösterilmiştir. Tablo 1'de ise Robotino elektronik donanımının parametreleri ve bu parametrelerin değerleri verilmiştir.

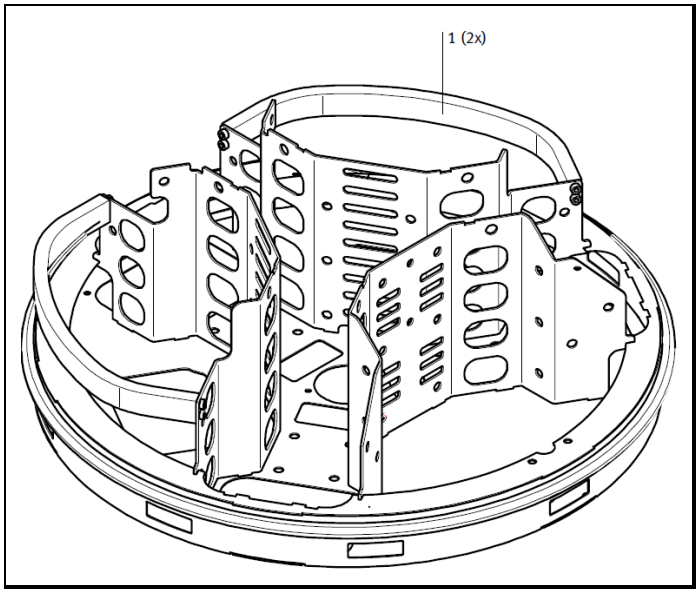
Şekil 1. Robotino Mobil Robot Sistemi.



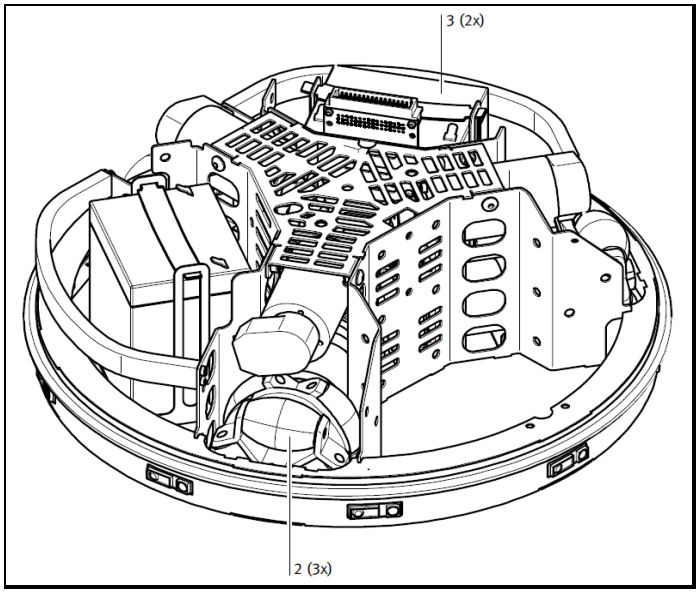
Şekil 2. Robotino-bilgisayar arasındaki kablosuz LAN üzerinden iletişim.

Tablo 1. Robotino parametreleri ve değerleri.

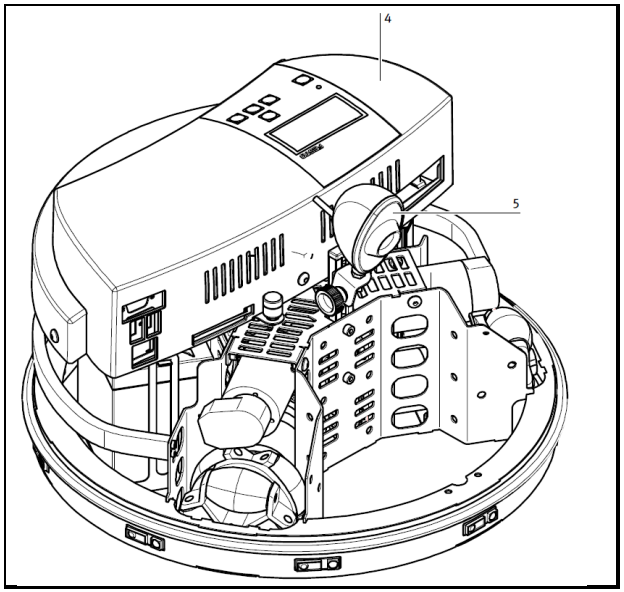
Parametre	Değer
Besleme Voltajı	24 V DC, 4.5 A
Dijital girişler	8
Dijital çıkışlar	8
Analog girişler	8 (0-10 V)
Röleli çıkışlar	2



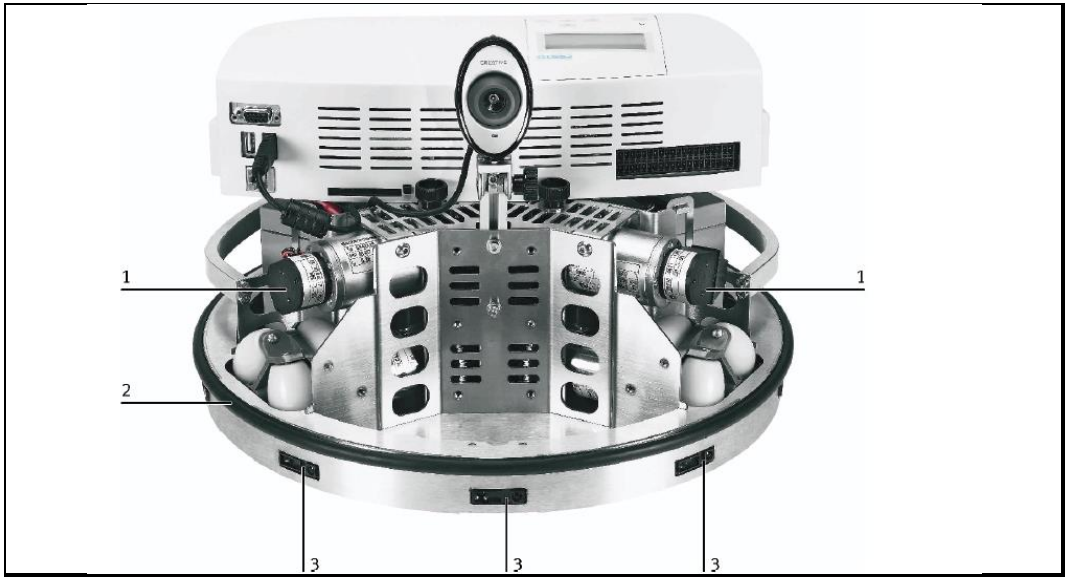
(1) El tutma yeri



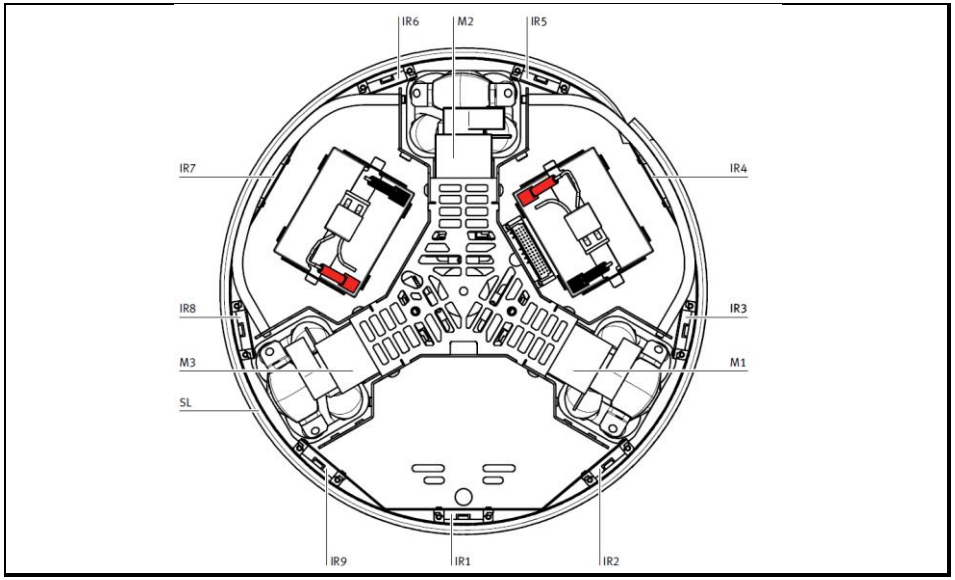
(2) Sürüş Ünitesi (3) Batarya



(4) Komut Ünitesi (5) Kamera



(1) Artımlı enkoder (2) Darbe bandı, çarpma önleyici sensör
(3) Mesafe ölçüm sensörleri

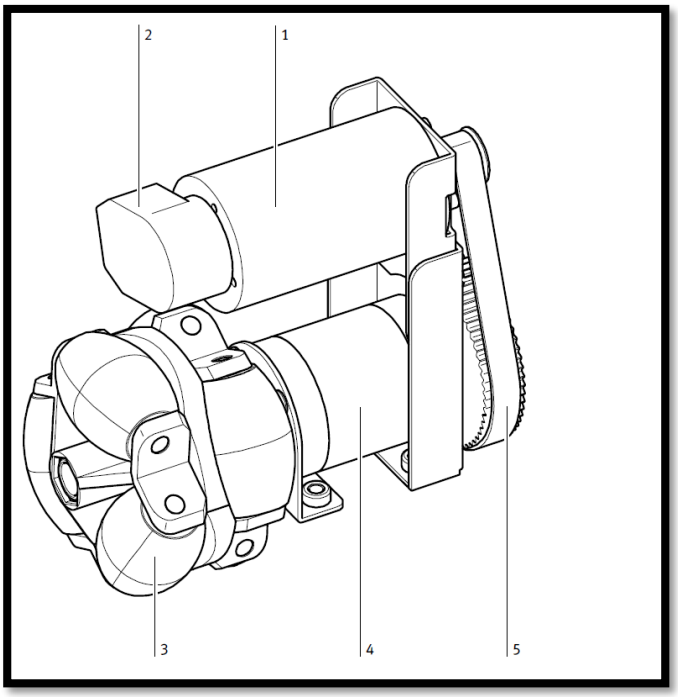


IR1 - IR9: mesafe ölçüm sensörleri (1-9)
M1 - M3: motorlar (1-3) SL = darbe bandı, çarpma önleyici sensör

Şekil 3. Robotino mekanik ve elektronik donanımı.

Robotino sürüş ünitesi Şekil 4'te görülmektedir. Mobil robot, 3 bağımsız çok-yönlü tekerlek ile sürülmektedir. Bu tekerlekler birbirleri ile 120º açı yapacak şekilde monte edilmiştir. 3 sürüş ünitesinin her biri şu komponentlerden oluşmaktadır:

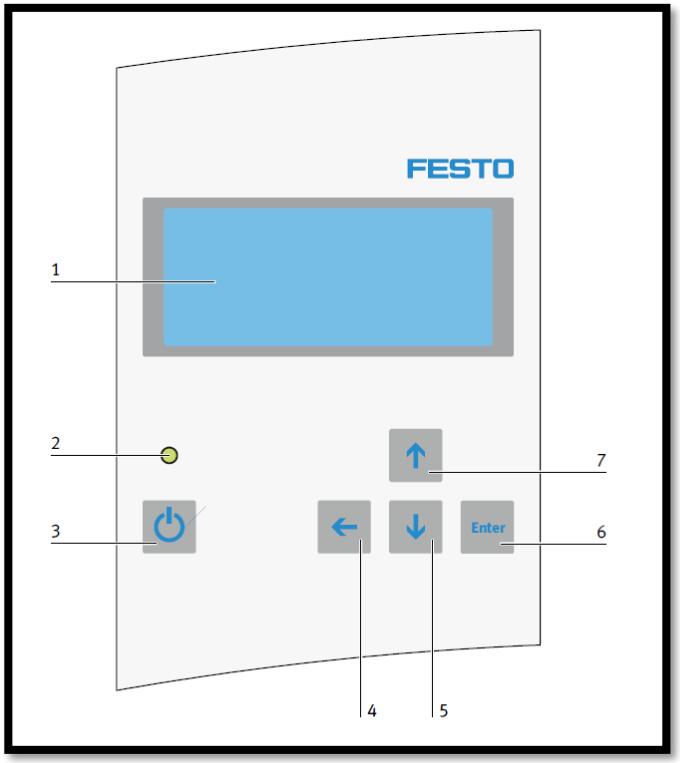
1. DC motor
2. Artımlı enkoder
3. Çok-yönlü tekerlek
4. 16:1 oranlı redüktör
5. Tırtıllı (dişli) kayış



Şekil 4. Robotino sürüş ünitesi.

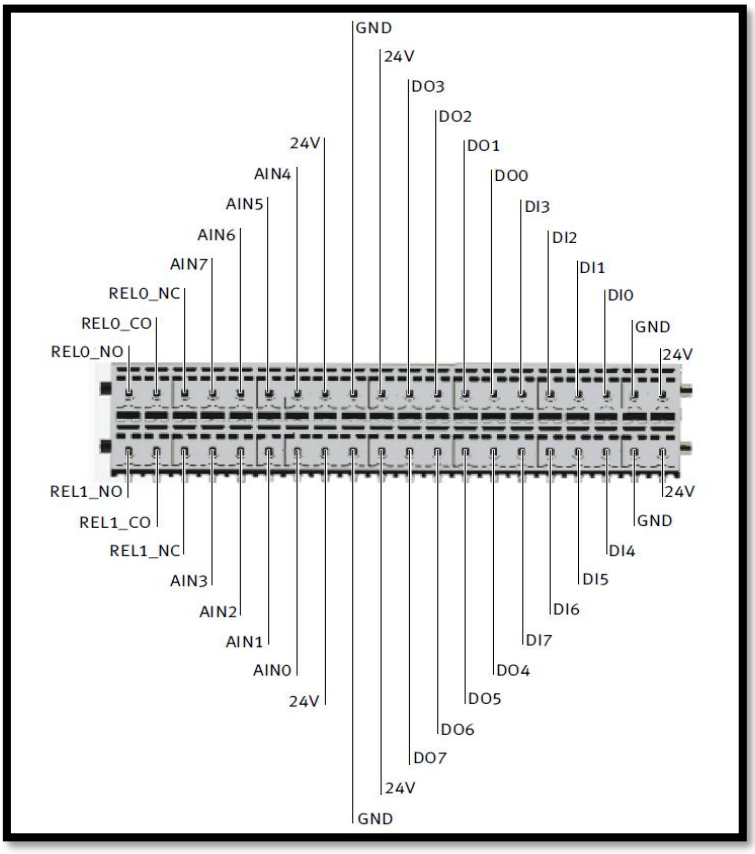
Robotino komut ünitesi kullanıcı ara yüzü ise Şekil 5'te gösterilmiştir. Bu ünite şu elemanlardan oluşmaktadır:

1. Display
2. LED
3. Aç/Kapa tuşu
4. Bir üst menüye çıkma tuşu
5. Menüde aşağı hareket tuşu
6. Seçimi onaylama tuşu
7. Menüde yukarı hareket tuşu



Şekil 5. Robotino komut ünitesi kullanıcı ara yüzü.

Şekil 6’da da dijital ve analog giriş çıkışlar görülmektedir.



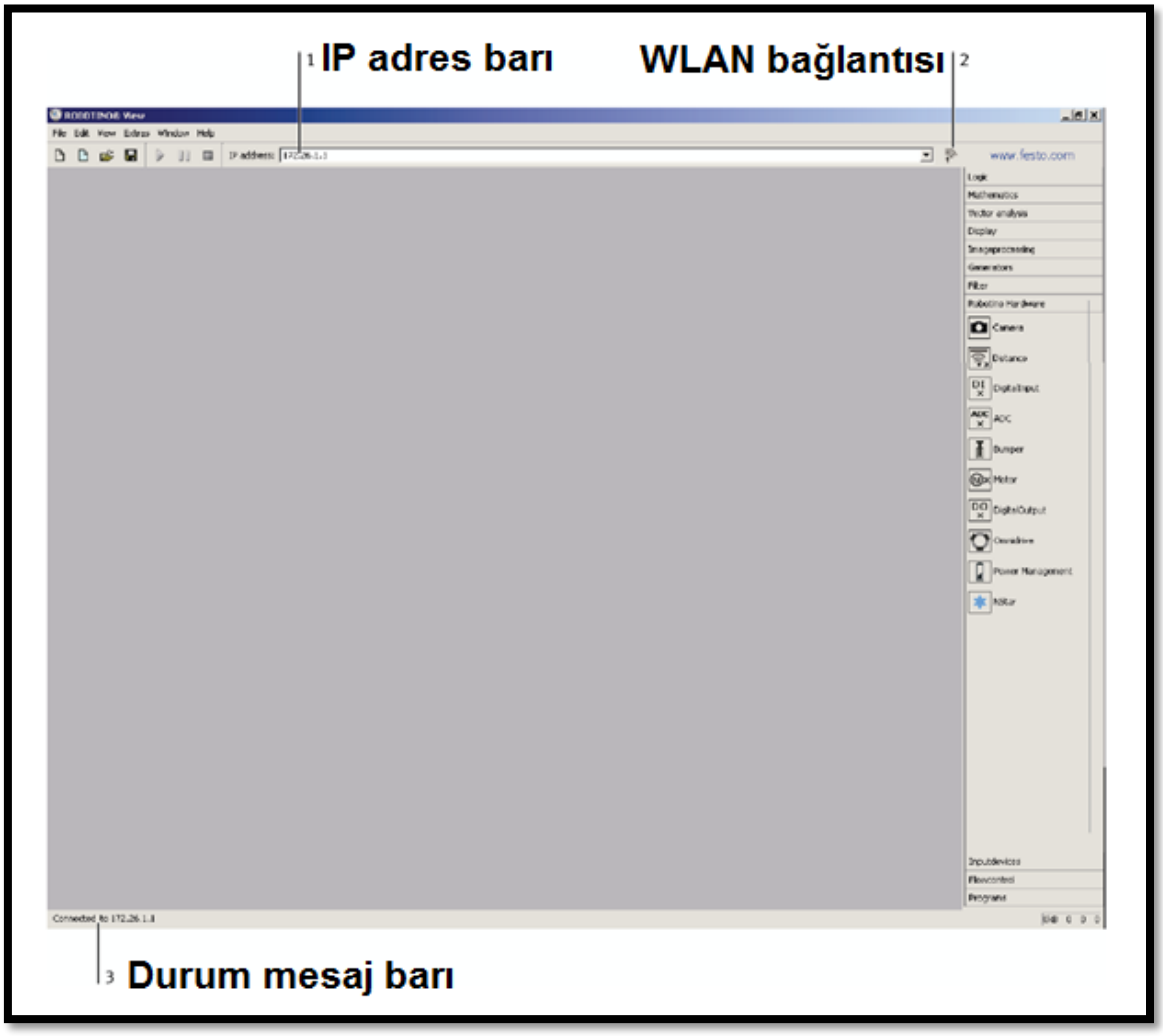
Şekil 6. Robotino dijital ve analog giriş çıkışlar.

3. ROBOTINO VIEW PROGRAMI

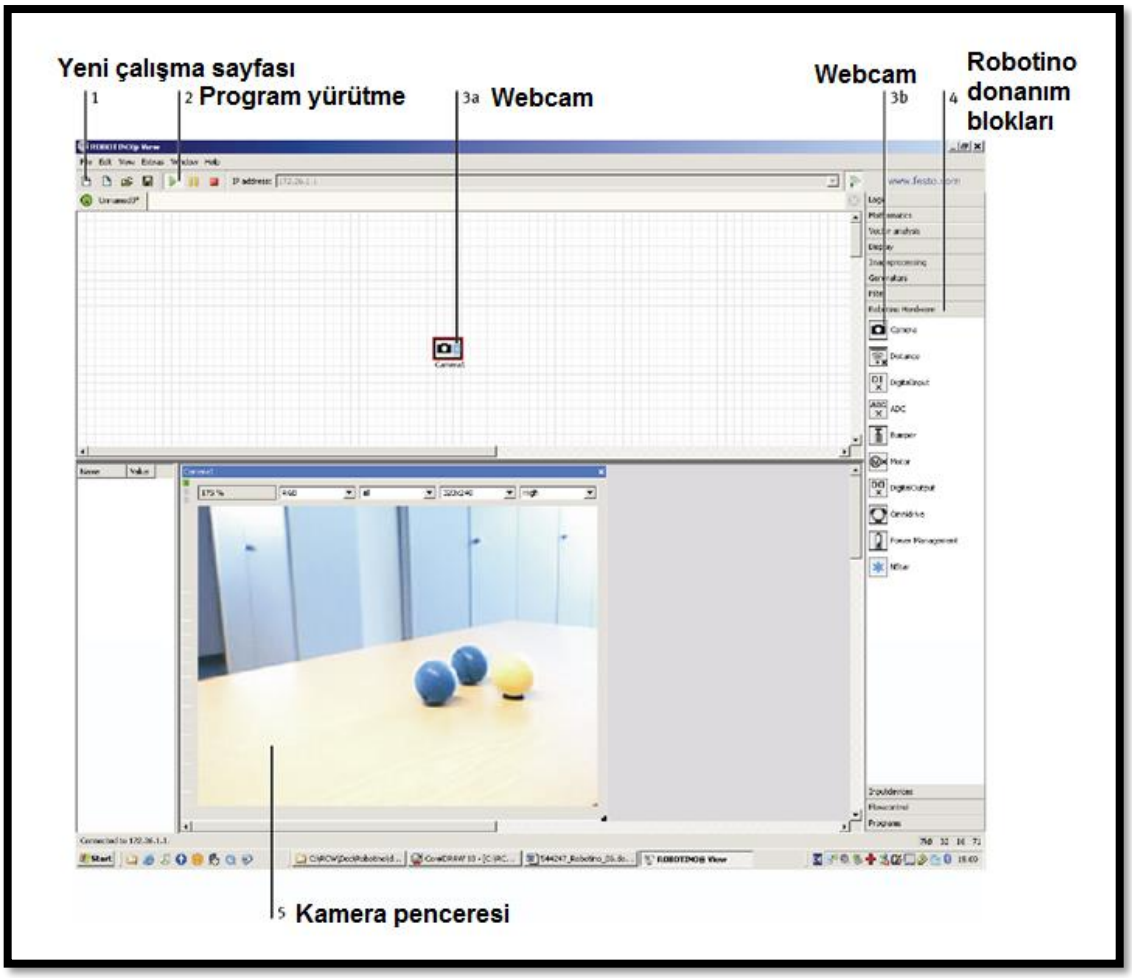
Robotino, C, C++, Java, .Net, Matlab, LabVIEW ve Microsoft Robotics Developer Studio gibi dil ve platformlarla programlanabildiği gibi Festo Didactic firmasının geliştirdiği Robotino View programı ile de programlanabilmektedir. Bu program kullanıcıya bloklar halinde fonksiyonlar sunarak, grafik programlama dili ile yazılım geliştirmeye ile olanak sağlamaktadır. Şekil 7 ve Şekil 8'de Robotino View programının genel görünümü yer almaktadır.

Şekil 9'da ise Robotino View programının arayüzünde hazırlanmış bir yazılımın blokları ve yazılım geliştirme penceresinin altında ise bu blokların görüntüleme veya ayar pencereleri gösterilmiştir.

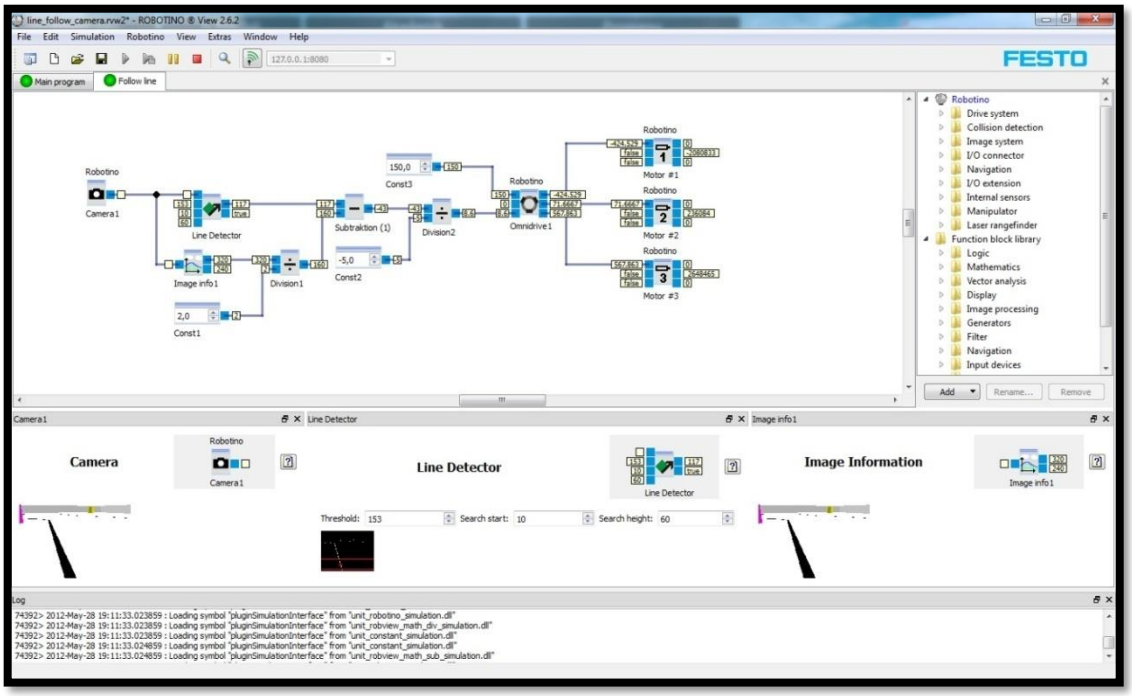
Ayrıca Robotino View programında hazırlanan yazılımlar, bir donanıma gereksinim duymaksızın yine Festo Didactic firması tarafından hazırlanan Robotino Sim programında görsel olarak simüle edilebilmektedir. Şekil 10'da da bu simülasyon programından bir görüntü yer almaktadır.



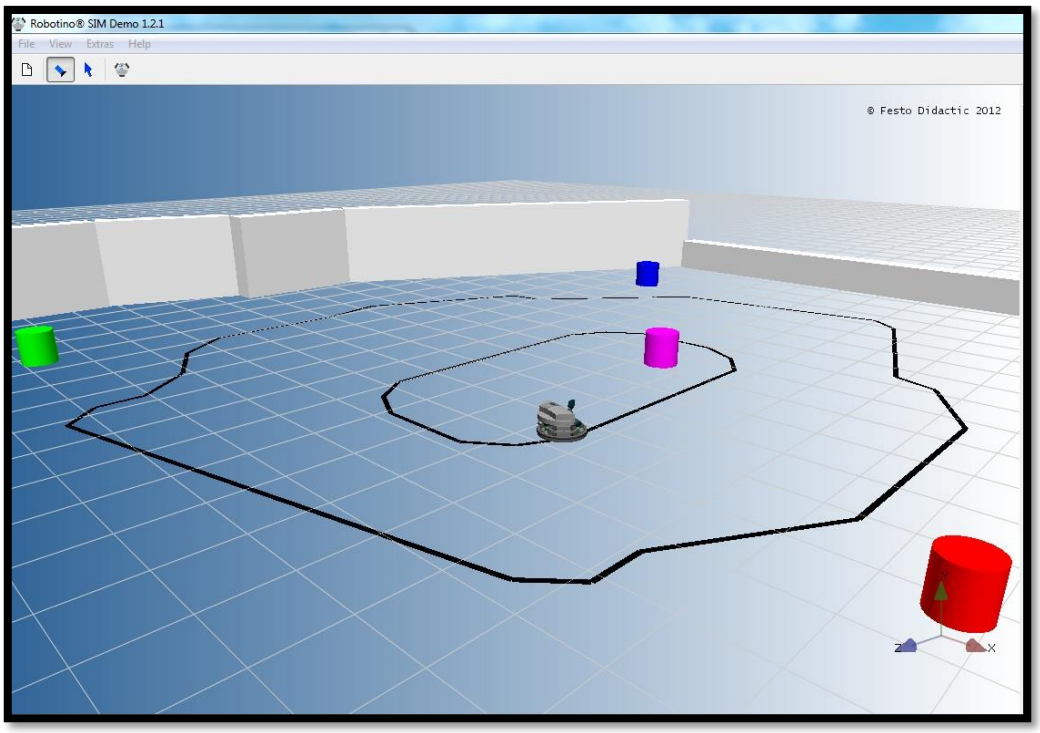
Şekil 7. Robotino View programı genel görünüm.



Şekil 8. Robotino View programı genel görünümü.



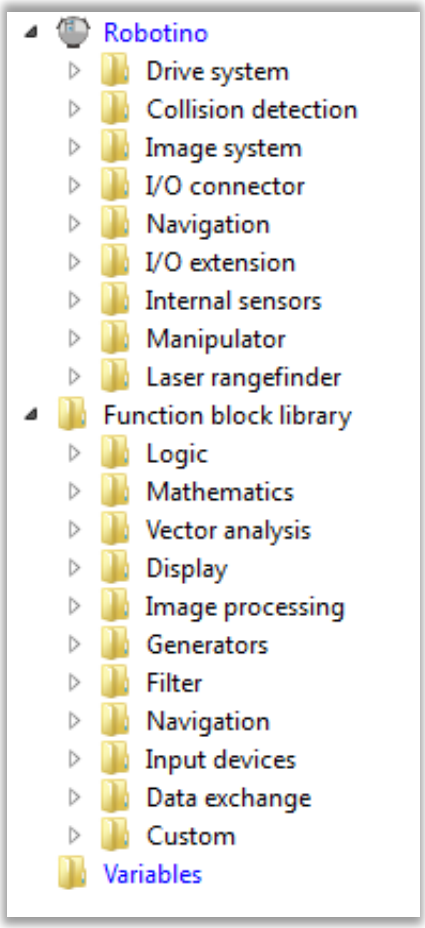
Şekil 9. Robotino View ile hazırlanmış örnek bir uygulama görüntüsü.



Şekil 10. Robotino Sim programının arayüzü.

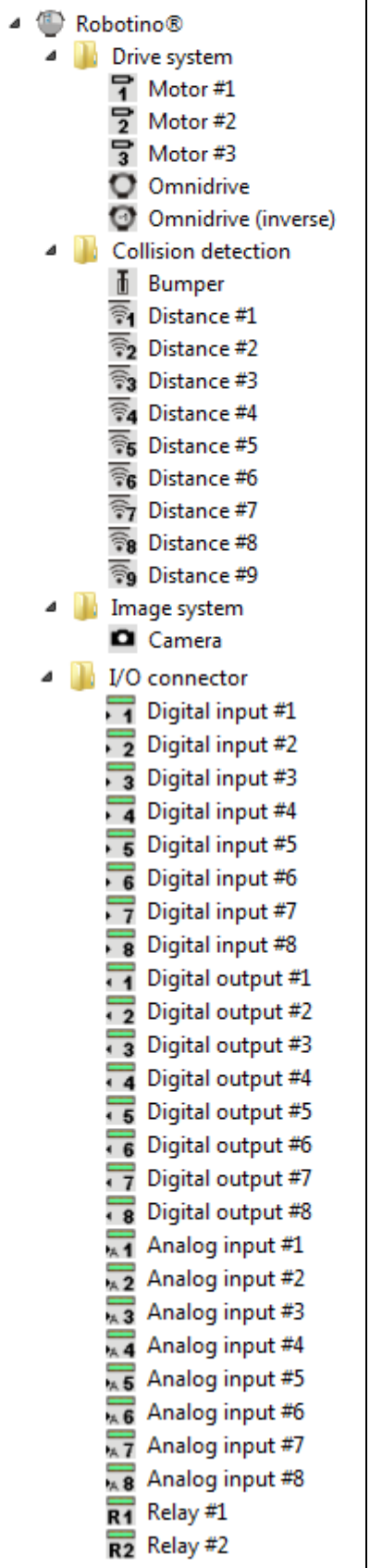
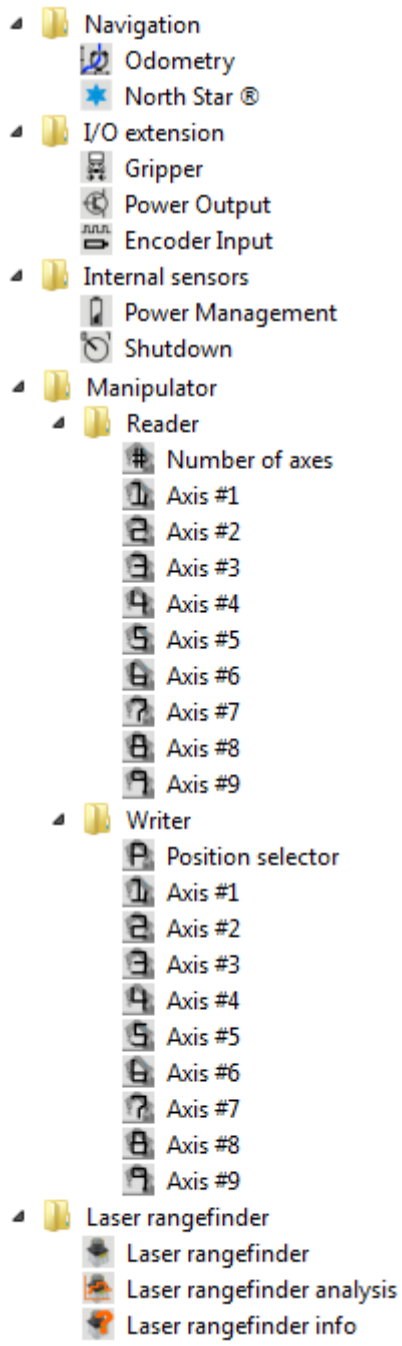
Robotino View programında sağ tarafta fonksiyon paleti yer almaktadır. Bu palet robotinonun donanımlarına erişebilmemizi, veri gönderip, veri okuyabilmemizi sağlayan "Robotino blok paleti" ve alışageldik cebirsel, lojik ve algoritmik fonksiyonların yer aldığı "Fonksiyon blok paleti" olmak üzere iki başlıktan oluşmatadır. Bu paletin genel görünümü ve alt menüleri Şekil 11'de gösterilmiştir.

Tablo 2 ve 3'te de sırası ile "Robotino blok paleti" ve "Fonksiyon blok paleti"nin alt menüleri ve blokları gösterilmiştir.

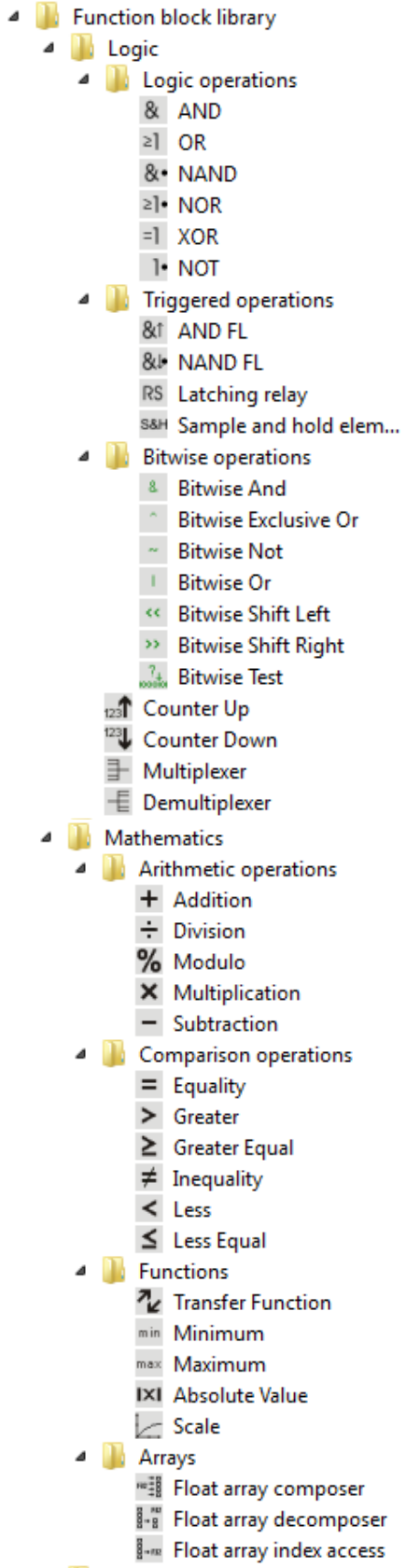


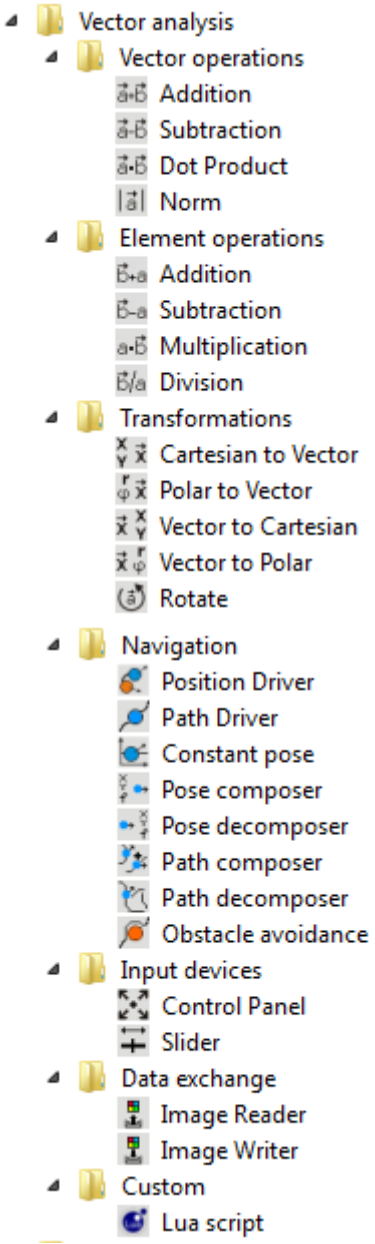
Şekil 11. Robotino View fonksiyon paleti.

Tablo 2. Robotino fonksiyon blok paleti.

ROBOTİNO FONKSİYON BLOK PALETİ			
	Bu palet içerisinde sol tarafta görüldüğü gibi motor, çok-yönlü sürüş blokları, mesafe sensörlerinden değer okumak için bloklar, kamera bloğu, dijital ve analog giriş çıkış blokları, yani robotino mobil robotunu kontrol etmek, sensörlerden veri okumak, motorları sürmek için gerekli olan temel donanım blokları yer almaktadır.		Ayrıca yine sol tarafta görüldüğü gibi, Robotino'ya eklenebilecek navigasyon, gripper, manipülatör ve lazer mesafe ölçüm sensörü gibi harici modüller (donanımlar) için fonksiyon bloklar yer almaktadır.

Tablo 3. Robotino fonksiyon blok paleti.

<table>
<tr><th colspan="4">FONKSİYON BLOKLARI KÜTÜPHANESİ</th></tr>
<tr>
<td>


</td>
<td>Fonksiyon blokları paletinde solda görüldüğü gibi lojik kapı ve işlem blokları, counter, mux, demux, blokları, cebirsel bloklar, karşılaştırma blokları ve dizi blokları yer almaktadır.</td>
<td>


</td>
<td>Ayrıca yine solda görüldüğü gibi vektör işlemleri için, transformasyonlar için bloklar, navigasyon, control paneli gibi bloklar yer almaktadır.</td>
</tr>
</table>

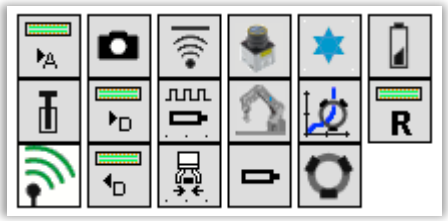
LabVIEW ile Yazılım Geliştirme

Robotino'nun LabVIEW için 'driver'ları da mevcut olup, bu 'driver'ların kurulumu ile robot LabVIEW üzerinden kontrol edilebilmekte ve yazılım geliştirilebilmektedir. İlk linkten ulaşılabilecek 'yardım manuel'inde, LabVIEW foksiyon paletindeki Robotino bloklarının (VI-Virtual Instrument, LabVIEW kullanıcıları bu ifadeye daha aşinadırlar) nasıl kullanıldığı ile ilgili detaylı bilgiye ve örnek uygulamalara, ikinci linkten Robotino'nun LabVIEW ile kullanımına dair genel bilgilere ulaşabilirsiniz. Bir sonraki linkten ise Robotino'nun LabVIEW 'driver'larına ulaşılabilir. Şekil 12' de Robotino VIs, Şekil 13' de LabVIEW üzerinde geliştirilmiş bir kontrol uygulaması (engelden sakınma) görülmektedir.

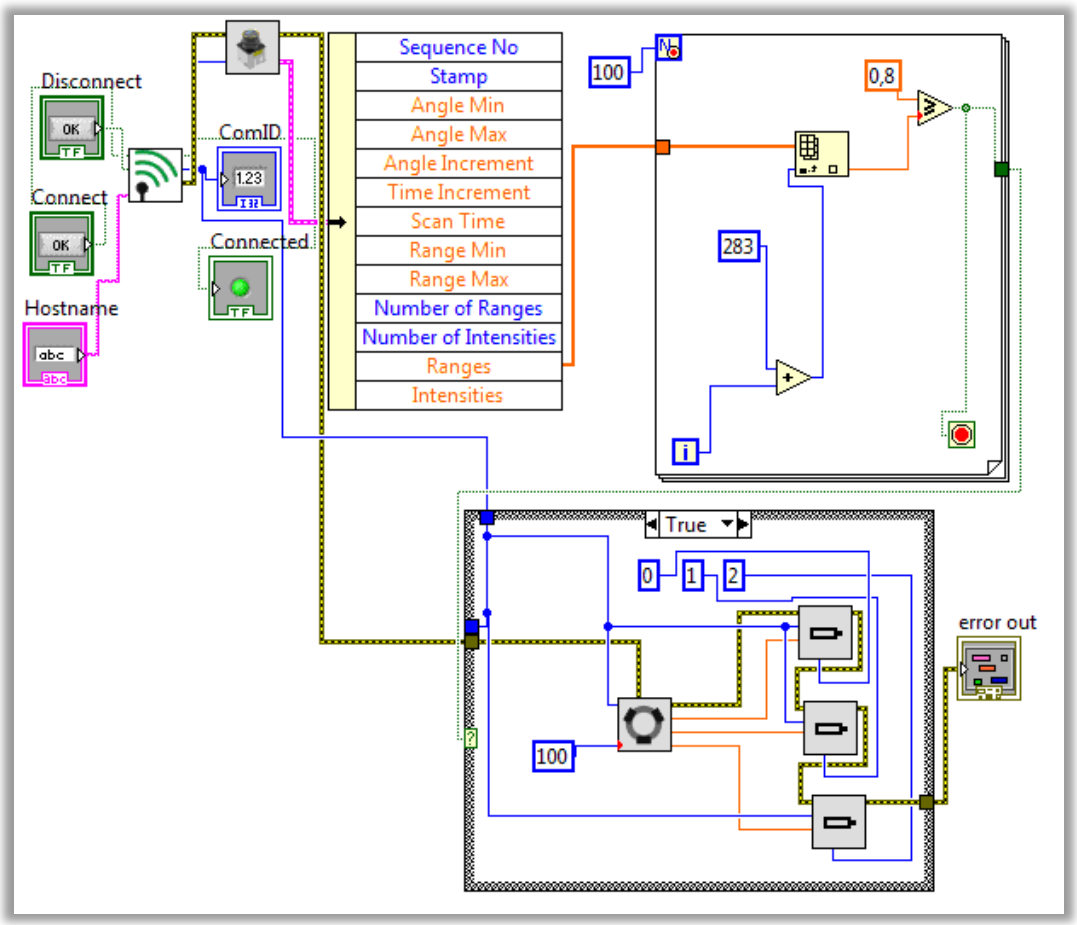
http://doc.openrobotino.org/documentation/LabView/

http://wiki.openrobotino.org/index.php?title=Labview

http://wiki.openrobotino.org/index.php?title=Downloads#LabVIEW



Şekil 12. Robotino VIs.



Şekil 13. Robotino için LabVIEW üzerinde geliştirilmiş bir engelden sakınma uygulaması.

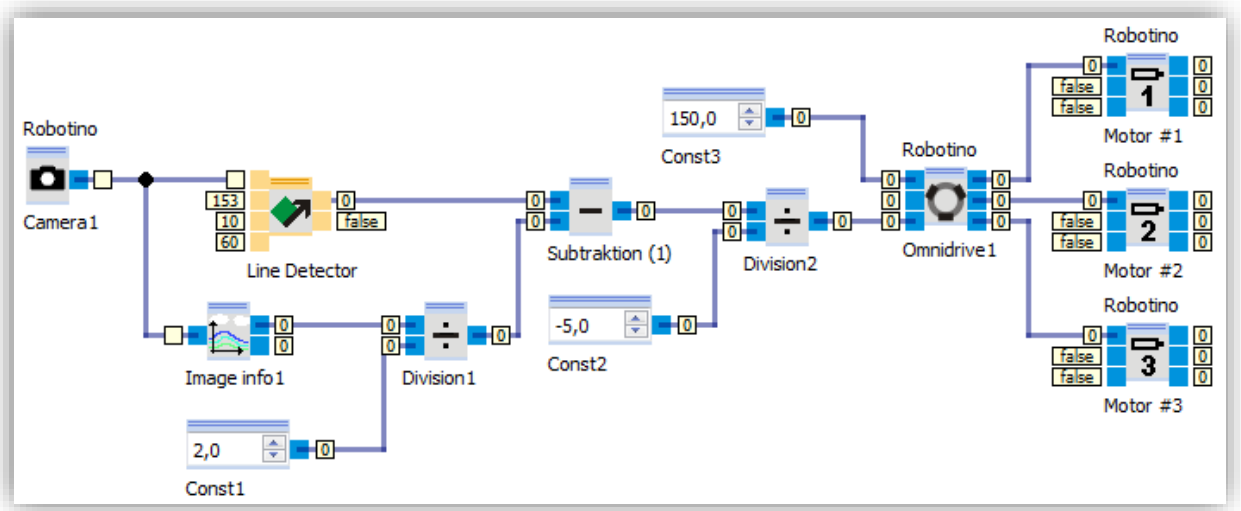
4. ROBOTINO VIEW UYGULAMA:
KAMERA İLE ÇİZGİ İZLEYEN UYGULAMASI

A) Uygulama Öncesi Notlar:

1. Bloklar paletten sürükle bırak ile çalışma alanına yerleştirilir.
2. Fare imleci blok üzerindeki iki çizgiden oluşan çubuk üzerine getirildiğinde, taşıma imleci belirir, bloklar buradan tutularak çalışma alanında herhangi bir noktaya taşınabilir.
3. Şayet var ise, bloğun özellik menüsüne, bloğa çift tıklayarak veya blok üzerinde fareye sağ tıklayarak açılan listeden "Properties" seçilerek ulaşılabilir.
4. Fare imleci blok üzerinde iken sağ tıklayarak açılan listeden "Help" seçeneği seçilerek, blok ile ilgili daha detaylı bilgiye ulaşılabilir.
5. Seçili olan blok aşağıda görüldüğü gibi sarı renkte vurgulanır.
6. Araç çubuğunda View sekmesinden, "Show Connector Values (Ctrl+D)" ve "Show Connector Descriptions (Ctrl+T)" seçeneklerinden blok giriş çıkışlarının aldığı değerler veya açıklamaları ile değer tipleri görünümleri açılıp kapatılabilir.

B) Uygulama Yönergesi:

1. Robotino'yu şarj ediniz. Robotino üzerinde (12 V, 5AH) 2 adet akü bulunmaktadır. Harici olarak veya robot üzerinden aküler şarj edilebilir. Şarj aleti üzerindeki led'in kırmızı renkte olması şarj olmadığını, sarı renkte olması şarj olduğunu, yeşil renkte olması ise şarjın tamamlandığını ifade etmektedir.
2. Robotino üzerindeki membran tuş takımı üzerindeki On/Off tuşuna basarak robotu çalıştırınız.
3. Robotino View programını çalıştırınız.
4. Kablosuz ağlardan Robotino 3.74 ağına bağlanınız.
5. Araç çubuğunda, Robotino'nun ip adresini ve portu giriniz: "172.26.1.1:8080"
6. Araç çubuğundaki  bağlantı simgesi ile Robotino'ya bağlanınız. Siyah renkli sinyal bağlantının gerçekleşmediğini, yeşil renkli sinyal gerçekleştiğini gösterir.
7. Kamera bloğunu çalışma ekranına sürükleyiniz ve araç çubuğundaki  "run" simgesine tıklayınız. Kamera bloğuna çift tıklayarak Robotino üzerindeki kameradan görüntü gelip gelmediğini doğrulayınız.  "stop" simgesine tıklayarak uygulamayı durdurunuz. Şayet görüntü gelmemiş ise, robota bağlantı işlemini tekrarlayınız (4., 5. ve 6. adımlar).
8. Aşağıdaki devreyi kurunuz.



Şekil 14. Çizgi izleyen uygulamasının devresi.

9. Run simgesine tıklayarak uygulamayı çalıştırınız. Hem, platform üzerinde çizgiye yerleştirdiğiniz robot üzerinde, hem de "camera" ve "line dedector" bloklarına çift tıklayarak uygulamanın çalışmasını gözlemleyiniz.

C) <u>Uygulamanın Algoritması:</u>

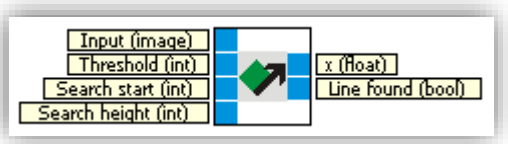
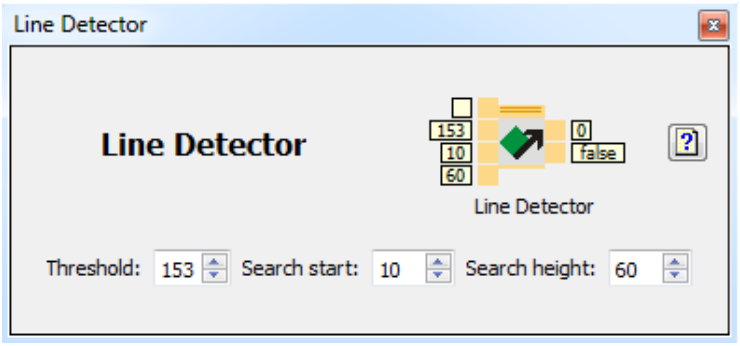
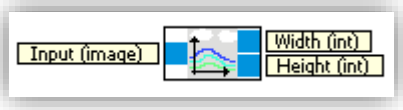
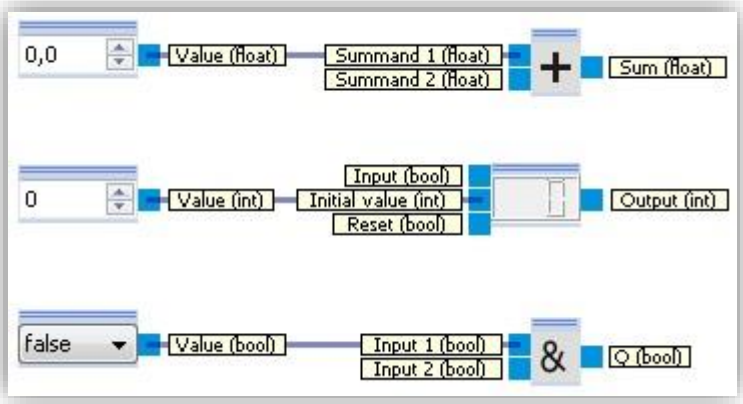
"Omnidrive" bloğuna baktığımız zaman V_x hız değerine sabit 150 değerinin uygulandığını görüyoruz. Aşağıdaki tabloda bu bloğun işlevi ile ilgili hücrede de tarif edildiği gibi, robot üzerindeki lokal koordinat sisteminde +x, robotun ileri doğru kat ettiği doğrultudur. y ise robotun yataydaki doğrultusudur. 150 sabit değeri V_x girişine uygulanarak, robotun belirli bir hızla ilerlemesi sağlanacaktır. Yataydaki yönlenmesi ise, çizgi takip algoritmasına göre belirlenecektir.

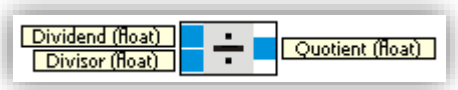
Kamera bloğu ile Robotino üzerindeki kameradan görüntü alınmakta ve alınan görüntü belirli aralıklarda elde edilmiş birer imge olarak uygulamada kullanılmaktadır. Bu imgeler "Line Detector" bloğuna gönderilir. Bu blok çizgiyi tespit etmek için özel olarak oluşturulmuş bir bloktur ve çıkışında çizginin en soldan ne kadar uzaklıkta olduğuna dair sayısal bir değer üretir.

"Image Info" bloğu ise imgenin genişliğini ve yüksekliğini elde eder ve çıkışına verir. Şekile baktığımız zaman bu uygulama için, blok çıkışından sadece genişlik bilgisini aldığımızı görmekteyiz. Alınan bu bilgi, ikiye bölünerek, tespit edilen çizgi mesafesinden çıkarılır. Amaç orta noktadan çizginin ne kadar saptığını ve elde edilen sayısal değerin işaretine göre sağda mı solda mı kaldığını tespit etmektir. Bu tespitten sonra değer bir sabite bölünür. Bu işlem ile robotun ne kadar dönmesi gerektiği fikrine ulaşılır. Sabitin eksi olmasının nedeni ise, robotun, çizgiyi ortalamak için sapma yönünün tersi yönünde dönmesi gerektiğindendir. Elde edilen bu değer, "omnidrive" bloğunda açısal hız girişine uygulanır. Girilen hız değerleri "omnidrive" bloğunda kinematik işlemlerden geçirildikten her bir motora uygulanması gereken hızlar elde edilir ve her bir hız değeri üç farklı motora gönderilir.

Uygulamada kullanılan blokların işlevleri ve giriş-çıkışları daha detaylı olarak bir sonraki bölümden incelenebilir.

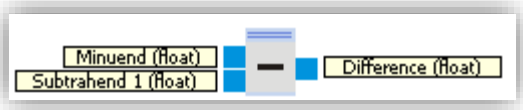
D) Fonksiyon Blokları ve İşlevleri

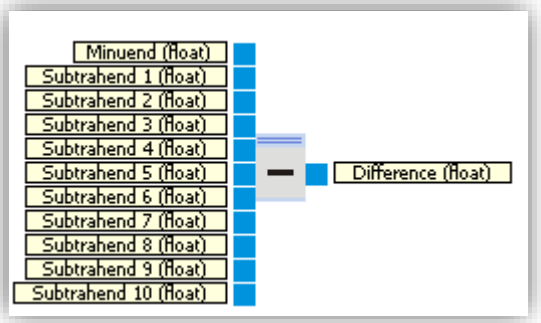
FONKSİYON BLOĞU	İŞLEVİ
	**Camera** Robotino'nun kamerasından elde edilen görüntüyü çalışmaya aktarır. Çözünürlük "device dialog" dan ayarlanabilir.
 	**Line Dedector** Input girişinden giren görüntüdeki çizgiyi algılar. Treshold girişi, çizgi tespitindeki hassasiyeti belirler. [0, 255] arasında bir değer verilebilir. Gürültü etkisini azaltmak için threshold değerini yüksek tutmakta fayda vardır.
	**Image Information** İmgenin genişlik ve yükseklik bilgisini verir.
	**Constant** Bir sabit üretmek için kullanılır. Sabitin tipi ve grafiksel görünümü bağlandığı girişin tipi ile değişmektedir.



Division

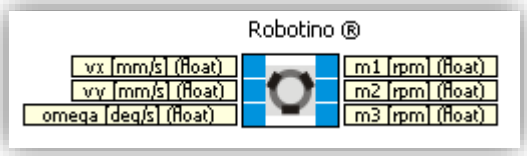
Bölme işlemi yapılır. Dividend 0 değil, Divisor 0 ise, simülasyon hata verir.





Subtraktion

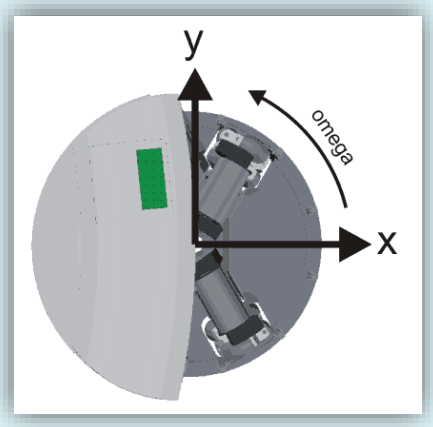
Minuend girişinden, 10'a kadar Subtrahend girişini çıkarmayı sağlar.



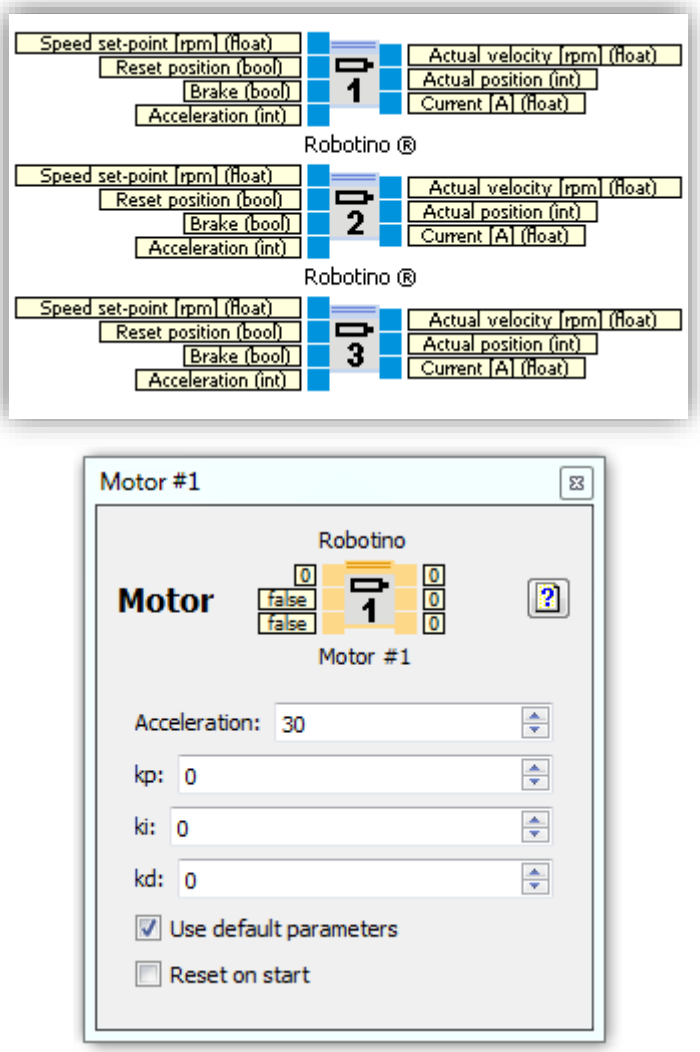
Omnidrive

vx, vy ve omega hız girişlerine göre, her bir motorun dönme hızları hesaplanır.

"Omnidrive (inverse)" fonksiyon bloğu ise motorun dönme hızlarından vx, vy ve omega hız değerlerini hesaplar.

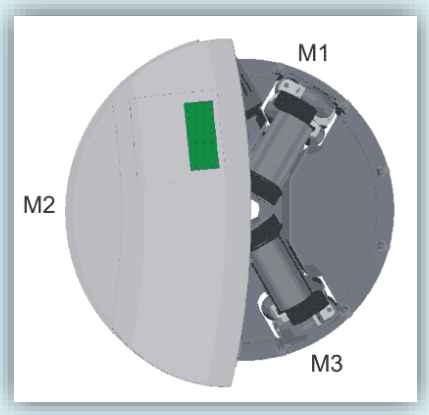


Yukarıdaki şekil Robotino'nun lokal koordinat sistemini göstermektedir. Yukarıdan bakıldığı zaman omega açısal hızının pozitif yönü saat yönünün tersidir.



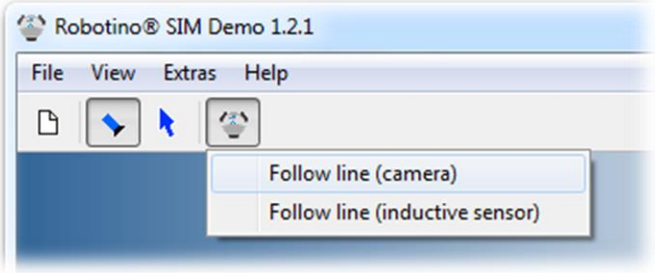
Motor #1 (, 2, 3)

Robotino motorlarına erişimi sağlar. Motor numarası blok üzerinde yer almaktadır.

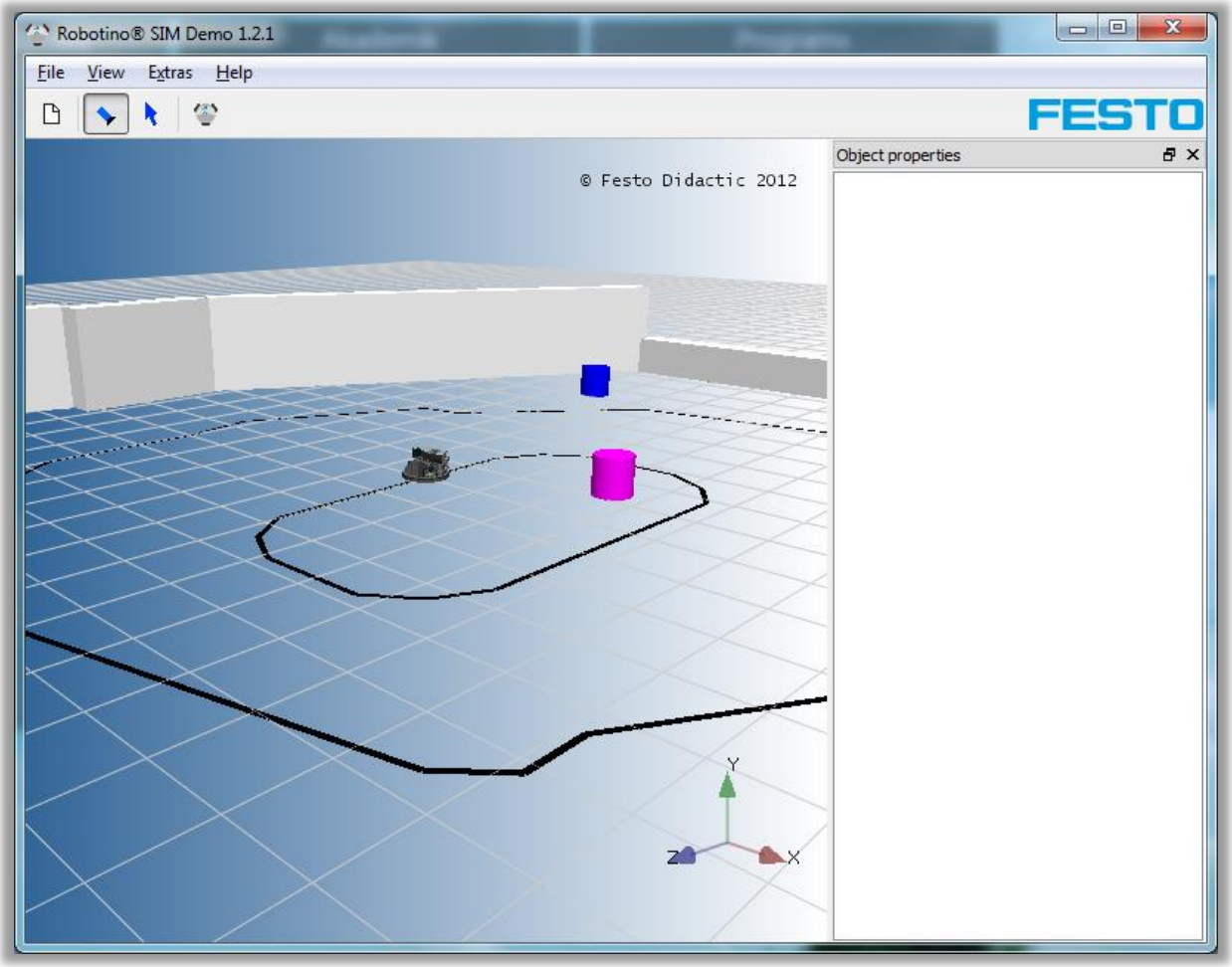


E) Robotino SIM Uygulaması

Aynı uygulama Robotino SIM programında da gerçekleştirilebilir. Araç çubuğundaki robotino simgesine tıklandığı zaman açılan listede "Follow line (camera)" seçilerek yukarıdaki uygulama Robotino SIM programında da çalıştırılabilir. Program ortamında hazırlanan çizgi izleme platformu üzerinde, geliştirilen program simüle edilebilmektedir. Şekil 15'de uygulama seçimi, Şekil 16'da program ara yüzündeki çalışma platformu görülmektedir.



Şekil 15. Robotino Sim uygulama seçimi.



Şekil 16. Robotino Sim programında çizgi izleyen uygulamasından bir görüntü.

5. DENEY RAPORUNDA İSTENENLER

1. Mobil robot bilgi veriniz.
2. Deneyde çalışılan çizgi izleyen uygulamasının devre elemanlarını tanıtınız.
3. Deneyde çalışılan çizgi izleyen uygulamasının prosedürünü açıklayınız.