

**T.C.**

**MİLLİ EĞİTİM BAKANLIĞI**

**SİVAS MERKEZ**

**Sivas Fen Lisesi**

**Bilgisyar Bilimi Dersi**

**Rapor**

|  |  |
| --- | --- |
| **Rapor No** | Proje-2 |
| **Rapor Tarih** | 29.05.2018 |
| **Proje Adı** | Eldivenle kontrol edilebilen robot |

Bilgisayar Bilimi Öğretmeni

Ersin TÜTÜNCÜ

2017-2018



**T.C.**

**MİLLİ EĞİTİM BAKANLIĞI**

**SİVAS MERKEZ**

**Sivas Fen Lisesi**

**Proje Grup**

|  |  |  |
| --- | --- | --- |
| **Proje Görev** | **Numara** | **Ad-Soyad** |
| Proje Yönetimi | 251 | Melih Altun |
| Doküman Yönetimi | 115 | Büşra Çakır |
| Lojistik Yönetim | 174 | Gökçenur Günaltay |
| Yazılım Geliştirme |  |  |
| Web ve GitHub Yönetimi | 184 | Mert Kılıç |
| Sunum Yönetimi | 413 | Semanur Yağmur |

**İÇİNDEKİLER**

[ÖZET 3](#_30j0zll)

[Anahtar Kelimeler 3](#_1fob9te)

[ABSTRACT 3](#_3znysh7)

[Key Words 3](#_2et92p0)

[Proje Görev Dağılımı Listesi ve Görev Dağılımı Açıklaması 4](#_tyjcwt)

[∙](#_3dy6vkm) Görev Dağılımı ve Sorumlusu 4

[∙](#_1t3h5sf) Görev süresince sürdürülen eylemler 5

[∙ Görevlerin iş yükü şeması](about:blank) 6

[∙](#_2s8eyo1) Yoklama Çizelgeleri 7

[∙](#_17dp8vu) Haftalık İş Katkı Cetvelleri 8

[GİRİŞ 8](#_3rdcrjn)

[1.Projenin Açıklaması 8](#_26in1rg)

[2.UML Diyagramlar 8](#_lnxbz9)

[3.Donanım Yapısı: 9](#_35nkun2)

[a.Gömülü Sistemler Mimarisi ve Devre Tasarımı 9](#_1ksv4uv)

[b.Mekanik Sistem Mimarisinin Tanıtılması: 12](#_44sinio)

[PID 17](#_2jxsxqh)

[4.Yazılım Yapısı 18](#_z337ya)

[SONUÇ 19](#_3j2qqm3)

[1.Bilgi Düzeyine Katkıları: 19](#_1y810tw)

[2.Teknolojik Katkıları: 19](#_4i7ojhp)

[3.Ekip Çalışması Katkıları 19](#_2xcytpi)

[4.Aksayan Yönler: 20](#_1ci93xb)

[5.Görüş ve Öneriler: 20](#_3whwml4)

**ÖZET**

Birinci proje kapsamında kullanılan elemanlar ile çizgi takip eden robot yapılacaktır. Bu projeden belirlenmiş olan çizgiyi sorunsuz ve istenen bir şekilde izleyerek pisti tamamlaması beklenmektedir. Bu amaçla kullanılacak olan QTR8A çizgi sensörünün siyah ve beyaz gibi renk ayrımı yapabilmesi sayesinde gerekli kodlamalar ile robot istenilen siyah çizgi üzerinde pisti tamamlayacaktır. Proje gerçekleştirilirken devre tasarımı, devre elemenlarının lehimlenmesi,test ve deneme aşamaları,kodlama,sunum gibi aşamalar gerçekleştirilecektir.

**Anahtar Kelimeler**

Arduino,Çizgi İzleyen Robot,QTR8A,Çizgi Sensörü,

**ABSTRACT**

The elements used in the first project and the robot following the line will be made. It is expected to complete the runway by following the line that is determined from this project without any problems. The QTR8A line sensor to be used for this purpose can distinguish between black and white color, so that the robot will complete the run on the desired black line with the necessary coding. During the project, stages such as circuit design, soldering of circuit elements, testing and testing stages, coding, presentation will be realized.

**Key Words**

Arduino, Line Follower Robot, QTR8A,Line Follower Sensor

**Proje Görev Dağılımı Listesi ve Görev Dağılımı Açıklaması**

* **Görev Dağılımı ve Sorumlusu**

|  |  |  |
| --- | --- | --- |
| Proje Yönetimi | Görev dağılımı ve takibinden sorumlu kişi, aynı zamanda proje grubunun çalışma takvimini ve düzenini ayarlamaktadır.Grupta bulunan kişilerle iletişim halinde olup projenin yönetimini sağlar. | Ad soyad  Melih Altun |
| Döküman Yönetimi | Projenin tüm tasarım ve çizimlerinden,proje raporunun sunulmasından,dökümanların uygun forma getirilmesinden kodlamaya ait diagram ve modellerin hazırlanması ve web sitesi tasarımından sorumlu olan kişidir. | Büşra Çakır |
| Lojistik Yönetimi | Projede kullanılacak tüm elemanların, malzemelerin belirlenmesi ve temin edilmesi,en uygun tasarımın yapılması için geliştirmelerin yapılmasıyla ve projenin donanımsal kısmının tanıtılması ile ilgilenen kişidir. | Gökçenur Günaltay |
| Yazılım Geliştirme Yönetimi | Yazılım için araştırmaların yapılması, yazılım aşamalarının proje grubuna dağıtılması,Yazılım ile ilgili raporların hazırlanarak ilgili bölüme(döküman yönetimine) aktarılması yazılım ve süreç testlerinin gerçekleştirilmesi ile ilgilenen kişidir. |  |
| WEB ve GitHub Yönetimi | Proje tanıtımı için WEB sayfasının hazırlanması, projenin GitHub yönetiminin yapılması,döküman yöneticisinden almış olduğu raporlar ile WEB sitesine ve GitHub'a işlemekle sorumlu olan kişidir. | Mert Kılıç |
| Sunum Yönetimi | Proje teslim zamanında sunumun, yapılan tüm işlemlerin uygun bir biçimde anlatılmasından,rapor ve evrakların eksiksiz bir şekilde sunulmasından ve önerilere,  sorulara uygun çözümler üretmekten sorumludur. | Semanur Yağmur |

* **Görev süresince sürdürülen eylemler**

|  |  |
| --- | --- |
| 1.Hafta | Proje Hakkında Bilgi Edinme |
| 2.Hafta | Malzeme Seçimi |
| 3.Hafta | Mekanik ve Elektronik Tasarım |
| 4.Hafta | Yazılım |
| 5.Hafta | Grup elemanlarına ait iş yükünün tamamlanması (rapor,web,github) |

SUNUM

## Görevlerin iş yükü şeması

* **Yoklama Çizelgeleri**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tarihler**  **Grup** | **28.11.2017** | **05.12.2017** | **12.12.2017** | **19.12.2017** | **26.12.2017** |
| **Melih Altun** | **+** | **+** | **+** | **+** | **+** |
| **Büşra Çakır** | **+** | **+** | **+** | **+** | **+** |
|  | **+** | **+** | **+** | **+** | **+** |
| **Gôkçenur Günaltay** | **+** | **+** | **+** | **+** | **+** |
| **Mert Kılıç** | **+** | **+** | **+** | **+** | **+** |

* **Haftalık İş Katkı Cetvelleri**

|  |  |
| --- | --- |
|  | **Yapılan iş** |
| **24.04.2018** | Proje araştırması |
| **01.05.2018** | Malzeme Teminatı |
| **08.05.2018** | Elektronik ve Mekanik Tasarım |
| **15.05.2018** | Yazılım |
| **22.05.2018** | Deneme ve Test Aşamaları |

**GİRİŞ**

**1.Projenin Açıklaması**

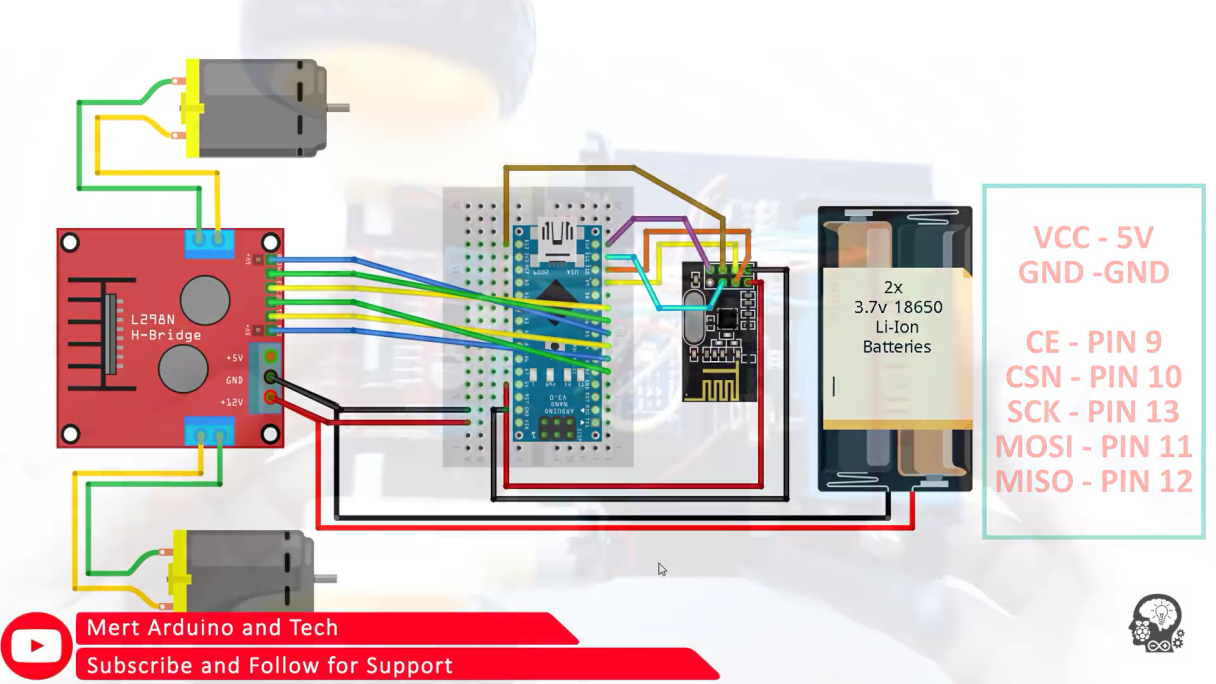
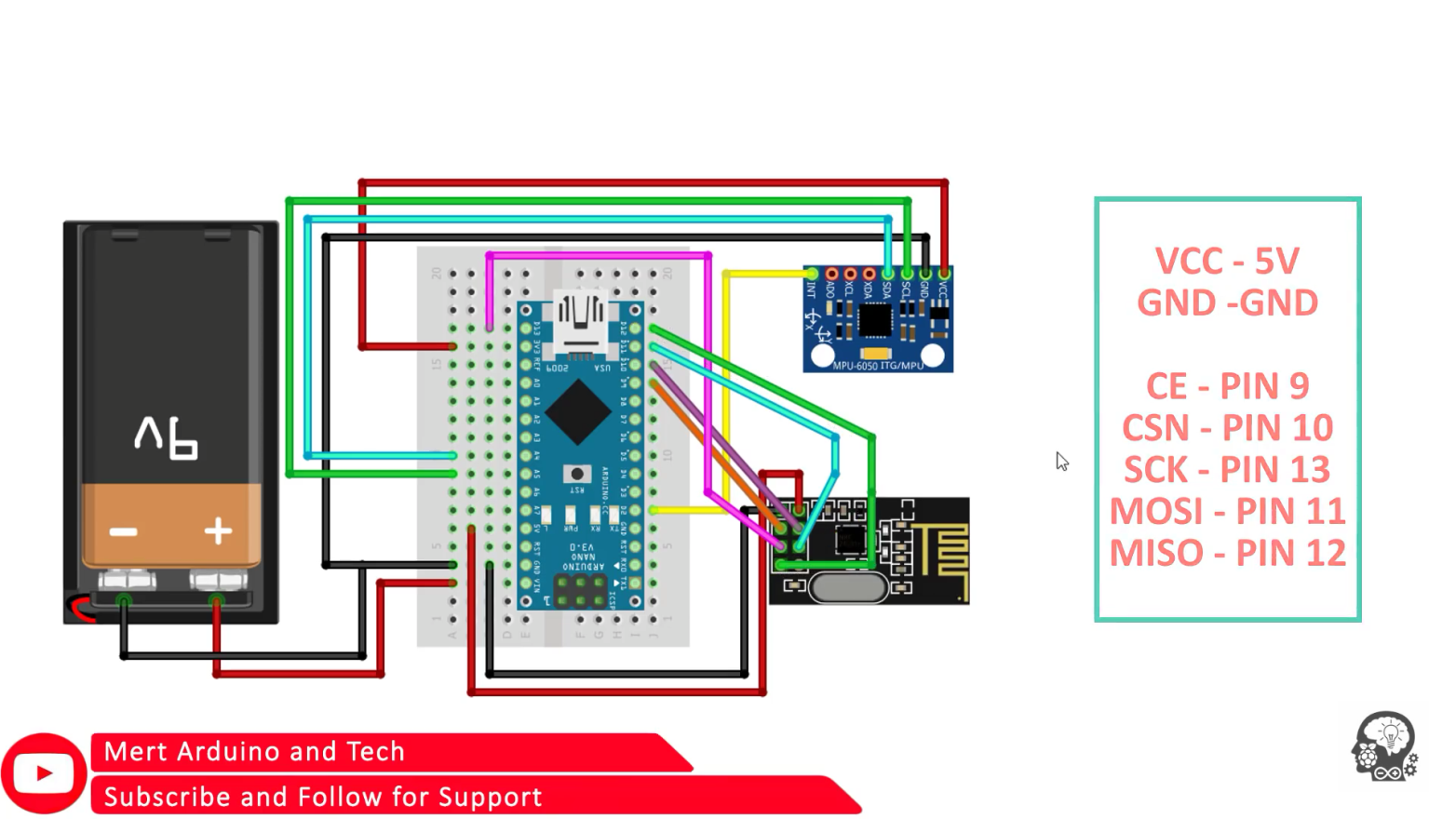
Yapılan projede amaç; hazırlanan pistte, çizgiyi takip edebilecek bir çizgi izleyen robot tasarımı gerçekleştirebilmektir. Robot yolu öz denetimli olarak takip etmelidir.Robotun çalışma prensibi;QTR-8A çizgi sensörü ile döndürülen değerler şart yapıları ile hesaplamaları sayesinde robotun konumunu güncelleyerek pisti tamamlamaktır.Robot ilerlerken hedeflenen çizgi robotun sağ tarafında kalırsa motorlar gerekli komutlarla sağ tarafa,sol tarafında kalırsa sol tarafa yönecelecektir.Dönme işlemi de hedeflenen çizgiye gelene kadar devam edecektir.Çizgi üzerindeyken robot düz devam edecektir.

**3.Donanım Yapısı:**

**a.Gömülü Sistemler Mimarisi ve Devre Tasarımı**

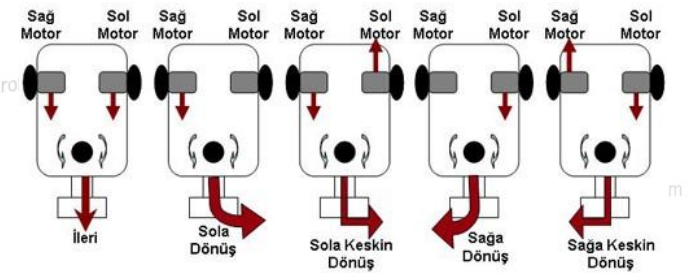
|  |  |  |
| --- | --- | --- |
| **C:\Users\Çakır\Desktop\indir.jpg** | | **Arduino Nano**  Arduino Nano; Atmega328 temelli bir mikrodenetleyici kartıdır. Üzerinde 14 adet dijital giriş/çıkış pini (6 tanesi PWM çıkışı olarak kullanılabilir), 8 analog giriş, 16Mhz kristal, usb soketi, ICSP konektörü ve reset tuşu bulundurmaktadır. Kart üzerinde mikrodenetleyicinin çalışması için gerekli olan her şey bulunmaktadır. Kolayca usb kablosu üzerinden bilgisayara bağlanabilir, adaptör veya pil ile çalıştırılabilir.. |
|  | | **L298N Voltaj Regulatörlü Çift Motor Sürücü Kartı**  24V’a kadar olan motorları sürmek için hazırlanmış olan bu motor sürücü kartı iki kanallı olup, kanal başına 2A akım vermektedir.Kart üzerinde L298N motor sürücü entegresi kullanılmıştır.DC motorlardan ayrı olarak step motor kontrolüne de imkan sağlamaktadır. |
| **C:\Users\Çakır\Desktop\mpu6050-6-eksen-ivme-ve-gyro-sensoru-6-dof-3-axis-accelerometer-and-gyros-4175-17-O.jpg** | | **MPU6050 6 Eksen İvme ve Gyro Sensörü - GY-521**  MPU-6050 çeşitli hobi, multicopter ve robotik projelerinde sıklıklı kullanılan üzerinde 3 eksenli bir gyro ve 3 eksenli bir açısal ivme ölçer bulunduran 6 eksenli bir IMU sensör kartıdır.   Kart üzerinde voltaj regulatörü bulunduğundan 3 ile 5 V arası bir besleme voltajı ile çalıştırılabilir. İvme ölçer ve gyro çıkışlarının her ikisi de ayrı kanallardan I²C çıkışı vermektedir. Her eksende 16 bitlik bir çözünürlükle çıkış verebilmektedir.   Pinler arası boşluk standart olarak ayarlandığı için breadboard veya farklı devre kartlarında rahatlıkla kullanılabilir. |
|  | | **Pertinaks**  Prototip devre elemanlarını pertinaks üzerine lehimleyip farklı uygulamalar için kullanabiliriz. |
|  | | **Jumper Kablo (E-E/D-E)**  Devre elemanlarının bağlantılarını gerçekleşirmek için kullanabiliriz. |
| **C:\Users\Çakır\Desktop\2526d9c2-87e1-4267-896d-c839104b45ac.jpg** | | **65mm Plastik Tekerlek**  Teker Çapı: 65mm   * Teker Kalınlığı: 30mm * Teker Ağırlığı: 38gr |
| **C:\Users\Çakır\Desktop\2526d9c2-87e1-4267-896d-c839104b45ac.jpg** | | **Mini DC Motor(B2D8X)**   * Çalışma Voltajı: 3-12V * Redüksiyon Oranı: 1:48 * Hız: 250 Rpm-6V * Çalışma Akımı: 95mA-160mA * Net Ağırlık: 29gr |
| **C:\Users\Çakır\Desktop\delikli-metal-sarhos-teker-4594-19-O.jpg** | | **Pololu Sarhoş Teker 21mm**  M4 vida delikleri sayesinde robot gövdelerinize ve farklı ortamlara kolayca sabitlenebilir.  Bilye kısmının çapı 14mm, yüksekliği 21mm'dir. Çizgi izleyen robot başta olmak üzere bir çok robot projesinde kullanılabilir. Bilyesi metaldir. |
| **C:\Users\Çakır\Desktop\indir (1).jpg** | Soket Adaptör Modülü Kurulu 8 Pin NRF24L01+ Kablosuz Alıcı-Verici Gerilim: 3.3V 8 Pin NRF24L01+ Modülü için NRF24L01 kablosuz modül için basit bir soket kartı Bütünleşik AMS1117-3.3 yongası NRF24L01 kablosuz modül ile kullanılabilir Boyut: 26 x 19 x 12mm | |
| **C:\Users\Çakır\Desktop\wireless-nrf24l01-24ghz-transceiver-modul-24ghz-alici-verici-modul-1-2154-84-O.jpg** | **Wireless NRF24L01 + 2.4 GHz Transceiver Modül - 2.4 GHz Alıcı Verici Modül**  Nordic firmasınca geliştirilen NRF24L01 kablosuz modül, 2.4GHz frekansında kablosuz haberleşme yapmanıza imkan sağlayan düşük güç tüketimine sahip modüldür.  Çeşitli hobi, robotik ve endüstriyel projelerde sıklıkla kullanılabilecek 2MBps haberleşme hızına sahip olup, SPI arabirimini destekler. | |

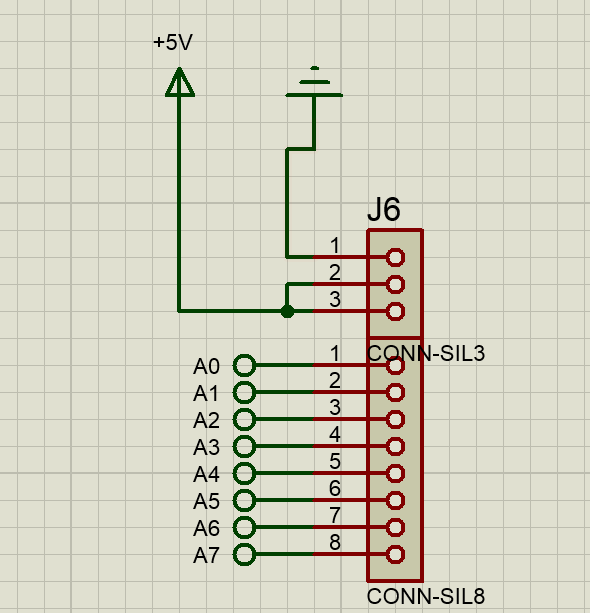
**Devre Tasarımı:**

****

**b.Mekanik Sistem Mimarisinin Tanıtılması:**

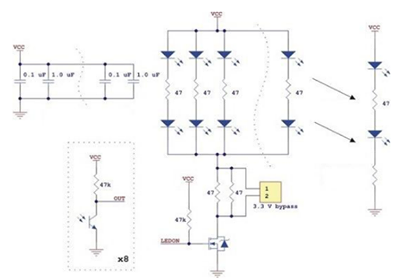
Robotun dönüşü birbirinden bağımsız çalışan iki motorun arasındaki hız farkıyla sağlanır.Aşağıdaki şemada bu sürüş sisteminin çalışılması gösterilmiştir.



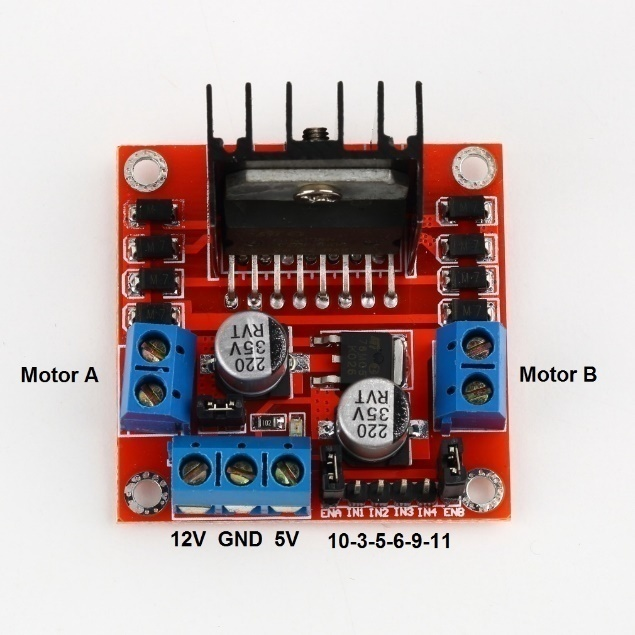
**1.QTR-8A Çizgi Sensörü**

Burada bulunan 8 tane çizgi sensörü pin’i arduınonun analog bacaklarına bağlıyoruz.Vcc ve led on pin’ini +5V’a Gnd pin’inide  toprağa bağlıyoruz.QTR-8A Sensör kartı bir kaç farklı şekilde çalıştırılabilir. Bir mikrodenetleyicinin analog giriş pinlerine bağlanarak ADC(Analog Dijital Çevirici) işlemine tabi tutulabilir. Eşik değeri ayarlanabilir bir karşılaştırıcı kullanılarak gelen analog voltajı dijitale çevirerek işlemler yaptırılabilir. Her çıkışı mikrodenetleyicinin I/O pinlerine bağlanarak mikrodenetleyici içindeki karşılaştırıcı kullanılarak okuma yapılabilir. (Bu yöntemle yüksek yansıma olan ortamlarda daha iyi sonuç alınır.).QTR-8A sensörü 8 sensörün tamamını kullanmak istemeyen kullanıcılara da imkan sağlamaktadır.Ayrıca sensör yakınlık ve cisim algılama sensörü olarak da kullanılabilmektedir.Çalışma voltajı 3.3 volt ile 5 votl arasındadır.Çektiği maksimum akım 100mAh tir.İdeal algılama mesafesi 3mm dir.Bu uzaklıktan fazla olduğunda kararlı çalışmaz.Sorunlar çıkartabilir.

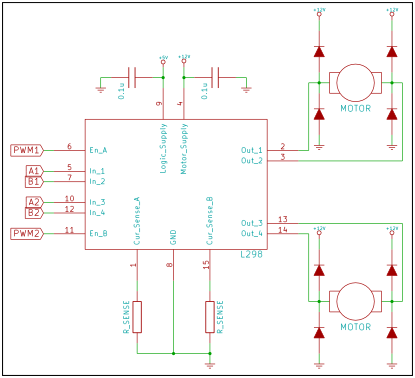
**QTR sensör iç yapısı:**

****

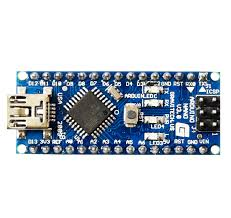
**2.L298N Motor Sürücü Kartı:**

****

2 adet H köprüsü bulunur. H köprüsü DC motorun iki yönde de hareket etmesini sağlayan faydalı bir yöntemdir.Bu yöntem için 4 adet transistör kullanılır.Bu entegre içerisinde toplam 15 adet bacak bulunmaktadır.Bunlardan IN1,IN2,OUT1,OUT2,ENA,SENSA A köprüsü için IN3,IN4,OUT3,OUT4,ENB,SENSB B köprüsü içindir.IN1 ve IN2 girişleri 5V’a duyarlı girişlerdir.OUT1 ve OUT2 isminden de anlaşılacağı gibi çıkış işlemleri içindir.

****

**3.Arduino Nano**

****

**Özellikler**

|  |  |
| --- | --- |
| Mikrodenetleyici | ATMega 328 |
| Çalışma Gerilimi | +5 V DC |
| Giriş Gerilimi(önerilen) | 7-12V DC |
| Dijital I/O Pinleri | * 14 tane (6 tanesi PWM çıkışını destekler |
| Analog Giriş Pinleri | 8 |
| Her I/O için Akım | 40mA |
| * 3.3V Çıkış için Akım | * 50 mA |
| EEPROM | * 1KB |
| SRAM | * 2KB |
| Flash Hafıza | * 32 KB (0.5 KB bootloader için kullanılır) |
| Uzunluk | * 45mm |
| Genişlik | 18mm |
| Ağırlık | * 5gr |

**Güç:**

Arduino Nano gücünü usb üzerinden veya harici güç kaynağından alabilir. Harici güç kaynağı AC-DC adaptör olabileceği gibi bataryada olabilir. Adaptör ve batarya kart üzerindeki GND ve Vin pinleri üzerinden bağlanabilir.

Kartın çalışması için sürekli olarak usb'nin bağlı olması şart değildir. Kart sadece adaptör veya batarya ile çalıştırılabilir. Bu sayede kart bilgisayardan bağımsız olarak çalıştırılabilir.

Harici güç kaynağı olarak 6-20V arası kullanılabilir. Ancak bu değerler limit değerleridir. Kart için önerilen harici besleme 7-12V arasıdır. Çünkü kart üzerinde bulunan regülatör 7V altındaki değerlerde stabil çalışmayabilir. 12V üstündeki değerlerde de aşırı ısınabilir.

Nano kartının üzerindeki mikrodenetleyicinin çalışma gerilimi 5V'dur. Vin pini veya güç soketi üzerinden verilen 7-12V arası gerilim kart üzerinde bulunan voltaj regülatörü ile 5V'a düşürülerek karta dağılır.

Güç pinleri aşağıdaki gibidir:

* **VIN:**Harici güç kaynağı kullanılırken 7-12V arası gerilim giriş pini.
* **5V:** Bu pin regülatörden çıkan 5V çıkışı verir. Eğer kart sadece usb (5V) üzerinden çalışıyor ise usb üzerinden gelen 5V doğrudan bu pin üzerinden çıkış olarak verilir. Aynı zamanda bu pin üzerinden 5V girişi yapılabilir.  Eğer karta güç Vin (7-12V) üzerinden veriliyorsa regülatörden çıkan 5V doğrudan bu pin üzerinden çıkış olarak verilir.
* **3V3:**Kart üzerinde bulunan 3.3V regülatörü çıkış pinidir. Maks. 50mA çıkış verebilir.
* **GND:** Toprak pinleridir.

**Hafıza:**

Atmega328 32 KB'lık flash belleğe sahiptir (2 KB kadarı bootloader tarafından kullanılmaktadır).  2 KB SRAM ve 1 KB EEPROM'u bulunmaktadır.

**Giriş ve Çıkış:**

Nano üzerindeki 14 adet dijital pinin hepsi giriş veya çıkış olarak kullanılabilir. 8 tane analog giriş pinide bulunmaktadır. Bu analog giriş pinleride aynı şekilde dijital giriş ve çıkış olarak kullanılabilir. Yani kart üzerinde toplam 20 tane dijital giriş çıkış pini vardır. Bu pinlerin tamamının lojik seviyesi 5V'dur. Her pin maks. 40mA giriş ve çıkış akımı ile çalışır. Ek olarak, bazı pinlerin farklı özellikleri bulunmaktadır. Özel pinler aşağıda belirtildiği gibidir.

* **Seri Haberleşme, 0 (RX) ve 1 (TX):**TTLSeri veri alıp (RX), vermek (TX) için kullanılır. Bu pinler doğrudan kart üzerinde bulunan FT232 usb-seri dönüştürücüsüne bağlıdır. Yani bilgisayardan karta kod yüklerken veya bilgisayar-nano arasında karşılıklı haberleşme yapılırkende bu pinler kullanılır. O yüzden karta kod yüklerken veya haberleşme yapılırken hata olmaması için mecbur kalınmadıkça bu pinlerin kullanılmamasında fayda vardır.
* **Harici Kesme, 2 (interrupt 0) ve 3 (interrupt 1):**Bu pinler yükselen kenar, düşen kenar veya değişiklik kesmesi pinleri olarak kullanılabilir. Ayrıntılı bilgi için [attachInterrupt()](http://www.arduino.cc/en/Reference/AttachInterrupt) fonksiyon sayfasını inceleyebilirsiniz.
* **PWM, 3,5,6,9,10 ve 11:**8-bit çözünürülükte PWM çıkış pinleri olarak kullanılabilir.
* **SPI, 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK):**Bu pinler SPI haberleşmesi için kullanılır.
* **LED, 13:**Nano üzerinden 13. pine bağlı olan dahili bir led bulunmaktadır. Pin HIGH yapıldığında led yanacak, LOW yapıldığında led sönecektir.
* **Analog, A0-A7:**Nano 8 tane 10-bit çözünürlüğünde analog giriş pinine sahiptir. Bu pinler dijital giriş ve çıkış içinde kullanılabilir. Pinlerin ölçüm aralığı 0-5V'dur. AREF pini ve analogReference() foksiyonu kullanılarak alt limit yükseltilip, üst limit düşürülebilir.
* **I2C, A4 veya SDA pini ve A5 veya SCL pini:**Bu pinler I2C haberleşmesi için kullanılır.
* **AREF:**Analog giriş için referans pini.
* **Reset:**Mikrodenetleyici resetlenmek istendiğinde bu pin LOW yapılır. Reset işlemi kart üzerinde bulunan Reset Butonu ile de yapılabilir.

Arduino nano ile Atmega328 arasındaki [pin haritalaması](http://www.arduino.cc/en/Hacking/PinMapping168) sayfayını inceleyebilirsiniz. 

**Haberleşme:**

Arduino Nano'nun bilgisayarla, başka bir arduino veya mikrodenetleyici ile haberleşmesi için bir kaç farklı seçenek vardır. Atmega328, 0 (RX) ve 1 (TX) pinleri üzerinden UART TTL (5V) seri haberleşme imkanı sunar. Kart üzerinde bulunan FT232 usb-seri dönüştürücüde bilgisayarda sanal bir com port açarak Atmega328 ile bilgisayar arasında bir köprü kurar. Arduino bilgisayar programı içerisinde barındırdığı seri monitör ile arduino ile bilgisayar arasında text temelli bilgilerin gönderilip alınmasını sağlar. Usb-seri dönüştürücü ile bilgisayar arasında usb üzerinden haberleşme olduğu zaman kart üzerinde bulunan RX ve TX ledleri yanacaktır.

Nano üzerinde donanımsal olarak bir adet seri port bulunmaktadır. Ancak [SoftwareSerial kütüphanesi](http://www.arduino.cc/en/Reference/SoftwareSerial) ile bu sayı yazılımsal olarak arttırılabilir.

Atmega328 aynı şekilde I2C ve SPI portlarıda sağlamaktadır. Arduino bilgisayar programı ile gelen [Wire kütüphanesi](http://www.arduino.cc/en/Reference/Wire) I2C kullanımını, [SPI kütüphanesi](http://www.arduino.cc/en/Reference/SPI) de SPI haberleşmesini sağlamak için kullanılır.

**4. 65mm Plastik Tekerlek + Mini DC Motor(B2D8X)**

****

**Şekil 8.DC motor ve tekerlek**

**Özellikler:**

**Tekerlek Özellikleri:**

* Teker Çapı: 65mm
* Teker Kalınlığı: 30mm
* Teker Ağırlığı: 38gr

**Motor Özellikleri:**

Çalışma Voltajı: 3-12V

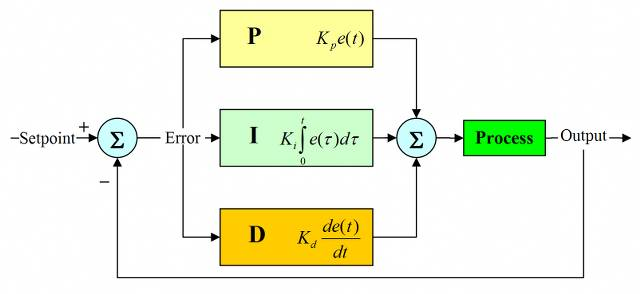
Redüksiyon Oranı: 1:48

Hız: 250 Rpm-6V

Çalışma Akımı: 95mA-160mA

Net Ağırlık: 29gr

**PID**

****

**Şekil 9: PID kontrol Diyagramı**

PID,Oransal İntegral Türev için kullanılan bir kısaltmadır.En genel tanımıyla bir kontrol geri bildirim mekanizmasıdır.PID yönteminin en temel amacı hatayı minimize etmek,en aza indirmektir.PID kontrolünü uygulamak ve kavramak oldukça zordur.PID kontrolünde öncelikle hata tanımlaması yapılmalıdır.Hata ise referans değere olan uzaklık olarak tanımlanabilir.Açıklamak gerekirse;

Referans=İstenilen değer

Gelen=Şuan ki Konum

HATA=Referans-Gelen

**Oransal Terim(P):**

Oransal terim, sistemden gelen hatayı bir katsayı ile çarparak hatayı küçültmeyi hedefler.Bozucu etkileri de mevcuttur.Projemizin yazılım kısmında oldukça ağırlık verdiğimiz PID kontrolünde deneme ve test aşamasında bu katsayıya büyük bir değer vermemiz gerektiğini anladık.

P=Kp\*HATA

**Integral Terimi(I):**

Integral hatanın alanını bulmak anlamına gelir. Integralin çok yükselmesini önlemek için sınırlandırmak gereklidir. Sürekli toplandığı için integral çok artarsa tekrar azalmasını beklemek zaman alır. Bu yüzden integrali sınırlamak sistemin çabuk toparlamasını sağlayacaktır.

I = I + (Ki \* HATA \* dt)

dt:PID fonksiyonuna her girdiğinde geçen zaman.

**Türev Terimi(D):**

Türev sistemdeki iki örnek arasındaki zamanı hesaplar.Eğer hatada bir değişim olmadıysa türev sıfır olur.

EHata=Bir önceki hatanın değeri

HD=HATA-EHata

D=(Kd\*HD)/dt

**PID Algoritması**

Kp , Ki ve Kd katsayılarından oluşur.Bu katsayılar deneme yanılma yöntemiyle bulunur.Yapacağınız sistemde optimum katsayıları bulmak için değerde değişiliklik yapıp sistemi gözlemlemeniz gerekmektedir.

HATA = Referans - Gelen

HD = HATA - EHata

P = Kp \* Hata

I = I + (Ki \* HATA\* dt)

D = (Kd \* HD)/dt

PID = P + I + D

EHata = Hata

**4.Yazılım Yapısı**

**a.Algoritmik olarak:**

A0.Başla

A1.Eğer ileri komutu veriyorsa ileri git

A2.Eger geri komutu veriyorsa geri git

A3.Eger sağa komutu veriyorsa sağa git

A4.Eger sola komutu veriyorsa sola git

A5.Bitir

**Yazılımı:**

Source Code (Transmitter - Glove)  
//Mert Arduino and Tech YouTube Channel -- https://goo.gl/ivcZhW  
  
//Add the necessary libraries  
//You can find all the necessary library links in the video description.  
#include <SPI.h> //SPI library for communicate with the nRF24L01+  
#include "RF24.h" //The main library of the nRF24L01+  
#include "Wire.h" //For communicate  
#include "I2Cdev.h" //For communicate with MPU6050  
#include "MPU6050.h" //The main library of the MPU6050  
  
  
//Define the object to access and cotrol the Gyro and Accelerometer (We don't use the Gyro data)  
MPU6050 mpu;  
int16\_t ax, ay, az;  
int16\_t gx, gy, gz;  
  
//Define packet for the direction (X axis and Y axis)  
int data[2];  
  
//Define object from RF24 library - 9 and 10 are a digital pin numbers to which signals CE and CSN are connected.  
RF24 radio(9,10);  
  
//Create a pipe addresses for the communicate   
const uint64\_t pipe = 0xE8E8F0F0E1LL;  
  
void setup(void){  
 Serial.begin(9600);  
 Wire.begin();  
 mpu.initialize(); //Initialize the MPU object  
 radio.begin(); //Start the nRF24 communicate   
 radio.openWritingPipe(pipe); //Sets the address of the receiver to which the program will send data.  
}  
  
void loop(void){  
   
 //With this function, the acceleration and gyro values of the axes are taken.   
 //If you want to control the car axis differently, you can change the axis name in the map command.  
 mpu.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);  
  
 //In two-way control, the X axis (data [0]) of the MPU6050 allows the robot to move forward and backward.   
 //Y axis (data [0]) allows the robot to right and left turn.  
 data[0] = map(ax, -17000, 17000, 300, 400 ); //Send X axis data  
 data[1] = map(ay, -17000, 17000, 100, 200); //Send Y axis data  
 radio.write(data, sizeof(data));  
}

Source Code (Receiver - Robot Car)  
//Mert Arduino and Tech YouTube Channel -- https://goo.gl/ivcZhW  
  
//Add the necessary libraries  
//You can find all the necessary library links in the video description  
#include <SPI.h> //SPI library for communicate with the nRF24L01+  
#include "RF24.h" //The main library of the nRF24L01+  
  
//Define enable pins of the Motors  
const int enbA = 3;  
const int enbB = 5;  
  
//Define control pins of the Motors  
//If the motors rotate in the opposite direction, you can change the positions of the following pin numbers  
const int IN1 = 2; //Right Motor (-)  
const int IN2 = 4; //Right Motor (+)  
const int IN3 = 7; //Left Motor (+)  
const int IN4 = 6; //Right Motor (-)  
  
//Define variable for the motors speeds  
//I have defined a variable for each of the two motors   
//This way you can synchronize the rotation speed difference between the two motors  
int RightSpd = 130;  
int LeftSpd = 150;  
  
//Define packet for the direction (X axis and Y axis)  
int data[2];  
  
//Define object from RF24 library - 9 and 10 are a digital pin numbers to which signals CE and CSN are connected  
RF24 radio(9,10);  
  
//Create a pipe addresses for the communicate  
const uint64\_t pipe = 0xE8E8F0F0E1LL;  
  
void setup(){  
 //Define the motor pins as OUTPUT  
 pinMode(enbA, OUTPUT);  
 pinMode(enbB, OUTPUT);  
 pinMode(IN1, OUTPUT);  
 pinMode(IN2, OUTPUT);  
 pinMode(IN3, OUTPUT);  
 pinMode(IN4, OUTPUT);  
   
 Serial.begin(9600);  
 radio.begin(); //Start the nRF24 communicate   
 radio.openReadingPipe(1, pipe); //Sets the address of the transmitter to which the program will receive data.  
 radio.startListening();   
 }  
  
void loop(){  
 if (radio.available()){  
 radio.read(data, sizeof(data));  
  
 if(data[0] > 380){  
 //forward   
 analogWrite(enbA, RightSpd);  
 analogWrite(enbB, LeftSpd);  
 digitalWrite(IN1, HIGH);  
 digitalWrite(IN2, LOW);  
 digitalWrite(IN3, HIGH);  
 digitalWrite(IN4, LOW);  
 }  
   
 if(data[0] < 310){  
 //backward   
 analogWrite(enbA, RightSpd);  
 analogWrite(enbB, LeftSpd);  
 digitalWrite(IN1, LOW);  
 digitalWrite(IN2, HIGH);  
 digitalWrite(IN3, LOW);  
 digitalWrite(IN4, HIGH);  
 }  
   
 if(data[1] > 180){  
 //left  
 analogWrite(enbA, RightSpd);  
 analogWrite(enbB, LeftSpd);  
 digitalWrite(IN1, HIGH);  
 digitalWrite(IN2, LOW);  
 digitalWrite(IN3, LOW);  
 digitalWrite(IN4, HIGH);  
 }  
  
 if(data[1] < 110){  
 //right  
 analogWrite(enbA, RightSpd);  
 analogWrite(enbB, LeftSpd);  
 digitalWrite(IN1, LOW);  
 digitalWrite(IN2, HIGH);  
 digitalWrite(IN3, HIGH);  
 digitalWrite(IN4, LOW);  
 }  
  
 if(data[0] > 330 && data[0] < 360 && data[1] > 130 && data[1] < 160){  
 //stop car  
 analogWrite(enbA, 0);  
 analogWrite(enbB, 0);  
 }  
 }

**SONUÇ**

**1.Bilgi Düzeyine Katkıları:**

Herşeyden önce proje yönetiminin nasıl olması gerektiğini ve projenin sunumunun nasıl yapılması gerektiğini öğrendik. 4 Haftalık süreçte birçok devre elemanının kullanımıyla ilgili tecrübeler edindik.Devrenin kurulumu,gerekli kodlamanın yapılması,devrelerin şematik olarak gösterilmesi,fritzing,proteus kullanımı ve daha birçok konuda bilgi edindik.Bir robotun elektonik ve mekanik tasarımının nasıl olması gerektiğini öğrendik ve yaptığımız yanlışlar ile tasarım aşamasında yapılmaması gerekenleri farkettik.

PID kontrolü konusuna yoğunlaştık.Hata oranını minimize ederek, robotun sorunsuz bir şekilde çizgiyi tam anlamıyla takip etmesi için araştırmalar yapıp Kp,Kd ve Ki değerlerine deneme ve yanılma yöntemiyle uygun katsayıları bulmaya çalıştık.Bizi en zorlayan kısım PID kontrolüve robotun çizgi takip işlemini gerçekleştirirken hız ve açısal konumunun ayarlanması oldu. Test aşamasında gördüğümüz hatalarımızı düzelttik. 1.Proje sayesinde devre tasarımında devre elemanlarının zarar görmemesi için yapılması gerekenleri devre tasarımının nasıl olması gerektiğini ve bağlantıları öğrendik.

**2.Teknolojik Katkıları:**

Çizgi izleyen robotlar günümüzde gelişen teknolojiyle birlikte endüstriyel alanda bir süredir kullanılmaktadır.Genellikle lojistik ve otomasyon bölümleri içerisinde oldukça gereksinim duyulan niteliksiz insan gücü ile yapılan taşıma işlemlerini bir süredir çizgi izleyen robotlar yapmaya başlamışlardır.Günümüzde bu kadar yaygın olarak kullanılan çizgi takip eden robot tasarımını gerçekleştirmiş olmamız bize iş hayatımızda kazanım olarak döneceğini ve bizim için önemli bir tecrübe,deneyim olduğunu düşünüyoruz.

**3.Ekip Çalışması Katkıları:**

Ekip çalışmasının bize kattıkları:

* Her grup elemanının teknik bilgisini geliştirdi. konuya bütünlük açısından bakmamıza yardımcı oldu.
* Proje çalışanların sorun çözme alışkanlığını geliştirdi.
* Çalışanların birbiriyle iletişim alışkanlıklarını geliştirdi.
* Ekip, bir kişinin tek başına üretebileceğinden daha fazla fikir üretebilir ve çözüm önerisi geliştirebilir.Yani; bir kişinin çözüm üretmesinin zor olduğu durumlarda ekip olarak çalışmanın faydalı olduğunu gördük.
* Yanlış karar verme ve yanlış uygulama riski ekip çalışması ile en aza indirgenmiş oldu.

**4.Aksayan Yönler:**

Projeyi gerçekleştirirken karşılaştığımız sorunlar:

* PID Kontrolü:Kp,Kd,Ki katsayılarına uygun değerleri bulmakta zorluk çektik.Verilen yanlış değerlerden dolayı robotun dönme açısının doğru olmadığını farkettik.Bu konuyla ilgili birçok araştırma yaparak sorunu çözmeye çalıştık.
* Devre Tasarımının Çizimi(Fritzing,Proteus,Circuits ):Devre tasarımının çiziminde yaşanılan aksaklık; kullandığımız devre elemanlarının fritzing ve proteus gibi devre tasarımı programlarında bulunmaması.Çözüm olarak paint ile devre tasarımını gerçekleştirdik.

**5.Görüş ve Öneriler:**

Bu proje elektronik ve mekanik iki kısım bulunmaktadır ve aynı zamanda bu iki kısmın birbirleriyle uyumu açısından irdelendiğinde hem teoride hem uygulamada proje grubumuza birçok bilgi birikimi sağlamıştır. Edindiğimiz bu bilgiler sayesinde yapacağımız diğer projelerde daha başarılı olacağımıza inanıyoruz.Çizgi takip eden robotumuzu daha fazla geliştirebiliriz. (ultrasonik mesafe sensörü ile mesafe algılamayı sağlamak,bluetooth ile kablosuz haberleşmeyi sağlamak ve daha birçok devre elemanı ekleyerek)