**HAZIRLAYAN :ÖMER SAMİ YAĞMUR**

**NO:110101007017**

**KONU:SANTRİFÜJ MAKİNESİ TASARIMI**

**İÇİNDEKİLER**

**2. SANTRİFÜJ NEDİR?SANTRİFÜJ MAKİNASI NEDİR?**

**3.. SANTRİFÜJ ÇEŞİTLERİ NELERDİR?**

**4. SANTRİFÜJ PARÇALARI NELERDİR?**

**5. SANTRİFÜJ ÇALIŞMA PRENSİBİ?**

**6. PROJENİN AMACI?**

**7,8,9,10 PROJEDE KULLANILAN MALZEMELER?**

**11. PROJENİN ÇALIŞMA PRENSİBİ?**

**12,13,14,15 SİSTEMİN KODLAMASI**

**16. KAYNAKÇA**

**PROJEDE KULLANILAN MALZEMELER**

**1.Button**

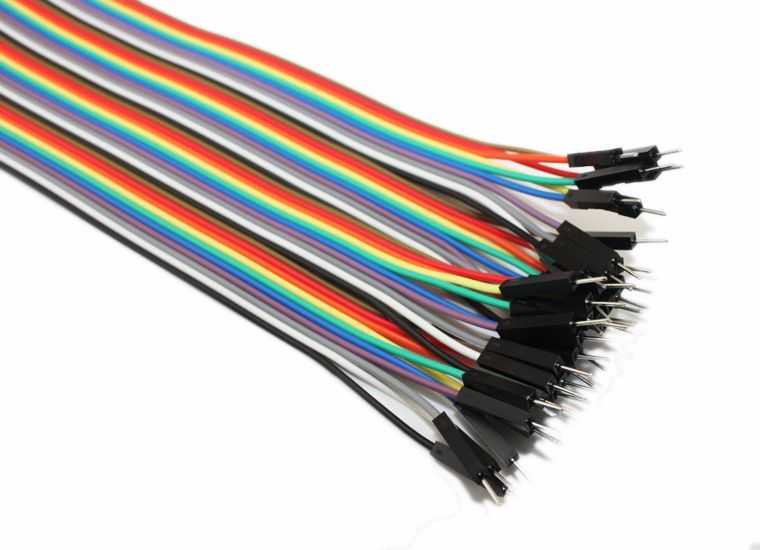
**** Projemizde 4 adet buton kullanılmıştır. Bu butonların görevi rotor hız kademesini ayarlamak, çalıştırma görevi ve kapatma görevidir. İlk buton başlama görevi görüp motorun dönmeye başlamasını sağlamaktadır.2. ve 3. Butonlar ise motor kademesini Rpm cinsinden ayarlama görevini üstlenmektedir. Butonlarda istediğim hız kademesine getirip başlangıç yapabiliririz . Örneğin 5000 rpm dönmesini istiyorsak kademeyi 5000 e ayarlayıp başlattığımızda 5000 rpmde dönecektir. Makine çalışır durumdayken tekrar butonlar ile kademesini değiştirebileceğiz. Örneğin 5000 rpmde çalışan makinaya 4000 rpm kademe ayarlaması yaptığımızda ve başlat butonuna bastığımızda 4000 rpm de dönecektir.

*Şekil 4 (4 bacaklı buton örneği*)[16]

**2.Jumper**

Projede bağlantıları yaptığımız kablolara denir.3 ayrı çeşittir vardır. Bunlar;

* Erkek-erkek
* Dişi-dişi
* Dişi-erkektir

Bizim projemizde en çok kullandığımız erkek-erkek tipidir. Erkek-erkek kablo 2 tarafında da kablonun uç verdiği için bu isim konulmuştur.

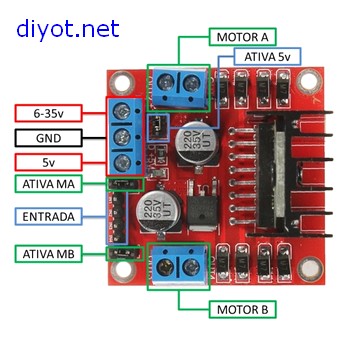
*Şekil 5 (Erkek uçlu jumper)* [17]

**3.Dc Motor**

Santrifüj için motor en önemli parçadır. Motorun çalışmaması durumu santrifüjün çalışmaması durumudur. Motoru seçerken dikkatli davrandım.

Motorumuzun üzerine binen yük sadece rotor dönerken oluşturduğu rüzgarın sürtünmesi olacaktır. Çünkü motor rotorun tam ortasına ağılık merkezine montajı yapılacaktır. Bu yüzden rotorun ağırlığı mile binecektir. Motorun ağırlığından dolayı motora bir yüklenme olmayacaktır. Buna bağlı olarak çekeceği akım miktarı çok az değişecektir. Çekeceği akım miktarı boştaki çektiği akım miktarına yakın olacaktır. Buda motorun uzun süre kullanılabileceği anlamına gelir. *Şekil 6 (Mabushi rs545 dc motor)[18]*

Kullandığımız dc motor mabushinin rs545 modeli olup teknik özellikleri şunlardır;

* Çalışma gerilim aralığı: 6-32 VDC
* Nominal Gerilimi: 24 VDC
* Hız: 9950 Rpm
* Boşta Çektiği Akım: 250mA
* Zorlanma Akımı(24V): 5A
* Mil: 3.17mm
* Uçtan Uca Uzunluk: 67mm
* Motor Çapı: 35.6mm
* Ağırlık: 158 gr [19-robotistan]

**4.L298 (H bridge):**Santrifüj motorunun sürmeye yarayan devre kartlarına denir. Hız kontrolü, iki yönlü sürme ayrı 5v çıkış özellikleri mevcuttur. Bu kart ile 2 motoru aynı anda sürebiliriz. Şekil 7 (L298 motor sürücü kartı)[20]

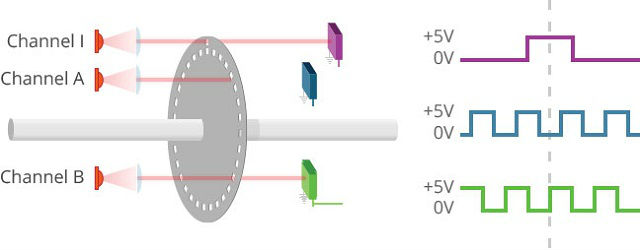
Sürücüde 7 adet çıkış pini 5 adet ise giriş pinleri mevcuttur.Çıkış pinlerinden 4 ü motor giriş-çıkışı,1 tanesi Gnd,1 tanesi Vcc,1 tanesinde 5v çıkış olacak şekildedir. Giriş pinleri ise 2 tanesi motor yönünü belirlemek için,2 tanesi de motor hızını belirlemek içindir.

Devre kartın teknik özelliklerine bakacak olursak;

* Boyutları: 57mm x 43mm
* Ağırlık: 29gr
* Pin Bağlantıları:
* ENA: Sol motor kanalını aktif etme pini
* ENB: Sağ motor kanalını aktif etme pini
* IN1: Sol motor 1. girişi
* IN2: Sol motor 2. girişi
* IN3: Sağ motor 1. girişi
* IN4: Sağ motor 2. girişi
* OUT1: Sol motor 1. çıkışı
* OUT2: Sol motor 2. çıkışı
* OUT3: Sağ motor 1. çıkışı
* OUT4: Sağ motor 2. çıkışı
* GND: Toprak bağlantısı
* VS: Besleme voltaj girişi(4.8V-46V)
* VSS: Kart besleme girişi (6V-12V)[21]

**5.Enkoder diski**

Motorun miline takılan üzerinde eş merkezli eş delikler olan motorun hızını ölçmeye yarayan malzemelere encoder diski diyoruz. Genelde delik sayısı 4 yada 20’dir. Delik sayısına göre programlama işlemi değişmektedir. Bu konuda daha sonra bahsedeceğiz.Lm393 kartı ile birlikte çalışır. Birlikte hız ölçme işlemini gerçekleştirirler.



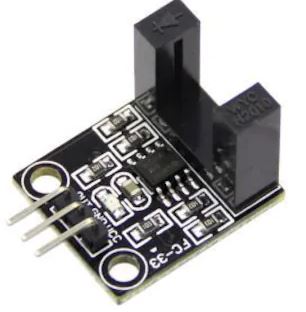
*Şekil 8 (encoder diski ile lm393 çalışma prensibi)[21]*

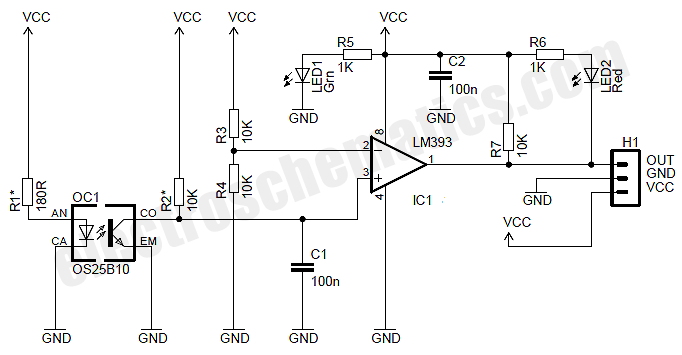
Şekil 8 e baktığımızda Işın her Channel B geçtiğinde yükselen kenar yani 5v,Chanel A ya geçtiğinde düşen kenar yani 0 v olarak lm393 kartından gerilim verilecektir.

**6.Lm393**

Bir kızılötesi devre kart modülüdür. Bir çok şekilde kullanımı mevcuttur. Mesafe ölçümü, uzaktan kumanda, engel görme, Hız ölçümü gibi uygulamalarda kullanılır. Bizim projemizde Bu sensörü rpm kontrolü için kullanacağız.

Kartın üzerinde bir adet lm393 entegrei,7 adet direnç,2 kapasitör, bir adet led,1 adet alıcı,1 adet verici bulunmakta. Vcc, Gnd, Digitpin olmak üzere 3 adet çıkış ucu vardır. Üzerindeki led engelin olup olmadığını anlamak için kolaylık sağlanmıştır.

Çalışması foto diyot prensibine dayanıyor. Her saniye vericideki foto diyota düsen fotonlar foto diyot üzerinden akımın geçmesine neden olur. Böylece arduino akımın geçtiğini görür ve ona göre programladığımız işlem ne ise onu uygular.

*Şekil 8 (Lm 393 entegre kartı) [23] şekil 7 (Lm 393 entegre şeması) [22]*

**7.Lcd ekran:**

Projede kullandığımız LCD ekran 16x2 ‘dir.16,yataydaki yazılabilecek karakter sayısı.2 ise dikeyde yazılabilecek karakter sayısıdır. LCD’nin görevi ise, yazdıracağımız rpm bilgisinin ve rpm ayarlanmasının göstergesi olacaktır. Hemen yan tarafında karanlıkta da ekranı görmemizi sağlayacak bir led bulunmaktadır.

*Şekil 9 (16x2 lcd ekran)* [24]

**8.I2C lcd pin düşürme modülü**

Arduino 13 adet digital pin bulunmaktadır. Bu pinleri tasarruflu kullanmamız gerekir. LCD ekran 6 adet pin kullanır.6 adet pin kullanım için çok fazladır. Diğer digital modüller için kullanılacak pin kalmamaktadır.I2C modülü 6 adet digital pini 2 adet pine analog pine dönüştürmektedir. Buna bağlı olarak düğer modülleri için kullanılacak alan açmaktadır.



I2C modülünün 16 adet girişi 4 adet çıkışı vardır. Girişler sırası işe Vcc,Gnd,A5,A6 olarak görev yapmaktadır. Giriş pini ise LCD’nin 16 adet çıkışına lehimleyerek bağlantısı yapılmaktadır. *Şekil 10 (I2C LCD ekran modülü)*[25]

I2C modülün bunlardan hariç avantajı ise üzerinde LCD ekran için kontrast ayarı yapılabilen potansiyometre bulundurmasıdır. Potansiyometre ile rahatça LCD kontrast ayarı yapılabilir.

**8.Arduino uno**

Arduino bir derleyici ile programlama yaptığımız, sayısız elektronik projesi gerçekleştirebileceğimiz ,mikrodenetleyicilerle ile benzer mantıkta çalışan hazır devre kartılarına denir. Üzerinde AT mega işlemci bulunmaktadır. Çalışması için bilgisayardan önce yazdığımız kodu içine derlememiz gerekir. Arduino bir çok çeşidi ve her çeşidin pinlerinin sayısı ve boyutu farklıdır bu çeşitlere bakacak olursak; Şekil 10 (*Ardunio uno)* [26]

* Arduino uno
* Arduino mega
* Arduino lilypad
* Arduino ethernet
* Arduino bluetooth
* Arduino mini
* Arduino nano
* Arduino leonardo
* Arduino esplora[27]

Biz projemizde arduino uno’yu kullandık. Arduino uno modelinde 13 adet digital giriş-çıkış pini bulunmakta. Bu pinleri derleyici yardımı ile giriş yada çıkış olarak ayarlayabiliyoruz.

Diğer kısımlarına bakacak olursak 6 adet analog pini,3 adet gnd,1 adet 5 v giriş,1 adet 5v çıkış,1 adet 3.3 v çıkış olmak üzere en bilindik pinleri bunlardır. Biz arduino beslememizi 5 v giriş pininden yapacağız.

**9.Direnç(10k ohm):**Dirençler devre içinde potansiyel farkı düşüren,devreden bir akımın akmasını sağlayabilen belirli büyüklükleri olan yapılara denir.Dirençleri düz bir yol üzerinde koşarken önünüze çıkan kayalıklı yürümesi zor bir alan olarak görebilirsiniz.Burda normal düz yolda koşarken harcadığının güçten daha fazlasını harcarsınız. *Şekil 11 (10k ohm direnç)[28]*

Biz projede dirençi kullanmamızda ki amaç buttonlar üzerinde **pull-down** etkisi içindir.Direnç buttonun çıkışına paralel olacak şekilde toprağa bağlanır.Böylece arduino butona basılamadığı sürece sürekli butona basılıyomuş gibi görmesini engeller.

**ÇALIŞMA PRENSİBİ VE BAĞLANTI ŞEMASI**

**1.Arduino bağlantıları**

**Buton bağlantıları:**

Arduino 3,4,5,6 pinleri 4 adet puton pini ile bağlantılı şekilde ayarlanmıştır.3 pin baştatma,4 pin hız artırım,5pin hız azaltım,6 pin ise kapatma butonu olarak ayarlanmıştır. Butondan gelen basılma bilgisine göre arduino işlem yapmaktadır.

**Motor bağlantırları:**

Arduinonun 7,8,9 pinleri motor bağlantılarını için atanmıştır.7 ve 8 pinler motorun hangi yöne döneceğini tayin eder.9 pin (enab pini) ise motorun hız bilgisi için tayin edilmiştir.

7 pin çıkış low yani 0,8pin çıkışı high yani 1 olarak ayarlanmıştır.Burada 7 pine (-),8 pin ise(+) olarak çalışmaktadır.8 pinden çıkan 5 volt 7 pine 0 volt olarak dönmektedir.

9 pin ise 3 ve 4 butonlardan ayarlanılan voltage değeri kadar motor hızını tayin etmektedir. Butonlardan ayarlanılan voltage değeri 0 ile 5 v arasında değişmektedir. Buna bağlı olarak 9 pinden motora 0 ile 5 volt arası gerilim verilmektedir. Böylece motorun hız değeri 3 ve 4 putonlara bağlı olarak değişmektedir.

**Encoder bağlantısı:**

Şekil 8 deki encoder (Lm393) entegresi diskteki yükselen değerlerde 5v,düşen değerlerde 0 v olarak 2 pine bilgi vermektedir. Yükselen değerler encoder diskindeki boşluklara denk gelmekle beraber düşen değerler encoder diskinde boşluğun olmaması durumuna denk gelmektedir.

2ci pin kodlamada input(giriş ) olarak atanmıştır.2 ci pine verilen bilgi her 5 v olduğunda programda atanan sayaç 1 artırılmaktadır. Artırılan sayaç değeri okunarak Rpm hesabı yapılmakta buna bağlı olarak arp değişkenine kaydedilmektedir. Arp değişkeni Lcd ekrana yazdırılmaktadır.

**Lcd bağlantıları:**

Lcd ile arduino arasındaki bağlantıları I2C modülü ile sağlanmaktadır. LCD den çıkan 16 adet pin I2C modülüne gelmektedir.I2C2nin çıkışında bulunan sda çıkışı Analog 5 pinine,scl çıkışı ise Analog 6 pinine bağlantısı yapılmaktadır. Diğer 2 çıkış ise arduino üzerindeki 5v ve gnd pinlerine bağlantısı yapılmaktadır. Butonlardan ve encoderden okunan bilgiler bu sayede LCD üzerine yazılmaktadır.

**Sistem çalışması:**

* Başlangıçta sistemi açtığımızda lcd ekran (rpm:0) olarak görüntülemektedir.4 ve 5 pinlere bağlı olan butonlar sayesinde RPM artırıp, azalma talimatı verilmektedir.
* Hız değerini ayarladıktan sonra 3 pine bağlı olan butona bastığımızda motor çalışmaya başlayacaktır.
* Hız değerinin yeterli olmadığını yada fazla olduğunu düşündüğümüzde tekrardan 4 ve 5 pin ile hız kademesini ayarlayıp başlat butonuna tekrardan bastığımızda ayarladığımız hız değerinde çalışmaya başlayacaktır.
* İşlemi bitirdiğimizi düşündüğümüz zaman motoru kapatmak için 6 pine bağlı olan buton ile motorun durdurma işlemi gerçekleşecektir.
* Yapılan bu aşmalarda Rpm değeri sürekli olarka lcd ekrana yazdırılacaktır.

**SİSTEMİN KODLANMASI**

**1.Pinlerin atamaları ve gişiş-çıkış atamaları**

**Buton pinlerinin değişkenlerin atanması ve giriş-çıkış atamaları:**

const int basla=3**; //Başlat butonunun 3. Pine atanması**

const int artir=4; **//Artır butonunun 4. Pine atanması**

const int azalt=5; **//Azalt butonunun 5. Pine atanması**

const int bitir=6; **//Bitir butonunun 6. Pine atanması**

int baslabasili; **// Başlat butonun basılı olduğu durumun bir değişkene atanması**

int artirbasili; **// Artır butonun basılı olduğu durumun bir değişkene atanması**

int azaltbasili; **// Azalt butonun basılı olduğu durumun bir değişkene atanması**

int bitirbasili; **// Bitir butonun basılı olduğu durumun bir değişkene atanması**

pinMode(basla,INPUT); //**Başlat butonunu Arduino ya bilgi girişi (5V) olarak atanması**

pinMode(artir,INPUT); // **Artır butonunu Arduino ya bilgi girişi (5V) olarak atanması**

pinMode(azalt,INPUT); // **Azalt butonunu Arduino ya bilgi girişi (5V) olarak atanması**

pinMode(bitir,INPUT); // **Bitir butonunu Arduino ya bilgi girişi (5V) olarak atanması**

**Motor pinlerinin değişkenlerin atanması ve giriş-çıkış atamaları:**

const int enba=9;//**Hız ayarı için enba değişkenin 9. pine atanması**

const int n1=8;//**Yön taiyini için n1 değişkenin 8. Pine atanması**

const int n2=7;// **Yön taiyini için n2 değişkenin 7. Pine atanması**

pinMode(n1,OUTPUT);//**n1 değişkenin olduğu pin çıkış olarak atanması**

pinMode(n2,OUTPUT);// **n2 değişkenin olduğu pin çıkış olarak atanması**

pinMode(enba,OUTPUT);// **enba değişkenin olduğu pin çıkış olarak atanması**

digitalWrite (n1,HIGH);//**n1 bağlı olduğu pinden 5 v gerilim verilmesi**

digitalWrite (n2,LOW);//**n2 bağlı olduğu pinden 0 v gerilim verilmesi**

**Encoder pinlerinin değişkenlerin atanması ve giriş-çıkış atamaları:**

const int encoder =2;// **encoder değişkeni 2. pine atanması**

pinMode(encoder,INPUT);//**Encoder bağlı olduğu pini giriş pini yani bilgi alma olarak ataması.**

**Lcd pinlerinin değişkenlerin atanması ve giriş-çıkış atamaları:**

#include<Wire.h> //**Lcd için gerekli olan kütüphaneler**

#include<LiquidCrystal\_I2C.h> //**Lcd için gerekli olan kütüphaneler**

LiquidCrystal\_I2C lcd(0X27,16,2); //**Lcd için gerekli olan kütüphaneler**

lcd.clear(); // **Lcd** **her başlangıçta bir önceki kalan çalışmadan kalanları temizler.**

lcd.setCursor(0,0); //**Başalangıç metninin nereye yazılacağını belirler.1.satır 1.stun**

lcd.print("lutfen baslama"); //**Başlarken yazılacak mesaj**

lcd.setCursor(0,1);//

lcd.print("tusuna basiniz"); //**Başlarken yazılacak mesaj**

lcd.clear(); //**Bir önceki metni temizler.**

delay(1000); //**1 sn gecikme verir.**

**2.Azaltım işlemi:**

while (rpm>=0) //**Rpm değerleri 0 değerleri üzerinde çalıştığında Döngüye girip alttaki işlemeleri yapcak.**

{azaltbasili=digitalRead(azalt); //**Azalt butonunun basılı olduğu durumun okunması**

if (azaltbasili == HIGH) // **Eğer azalt butonu basılı ise alttaki işlemmleri yapcak**

{rpm=rpm-500 ; //**Buton her basıldığında rpm değerini 500 azaltacak**

if (rpm<=0) //**Eğer rpm değeri 0 dan küçük olduğunda alttaki işlemi uygulayacak**

{rpm=0;} //**Rpm değerini 0 a eşitlicek. Rpm 0 değerlerinin altına düşmemesi durumunda yapılan işlemdir.Eğer bunu yapmaz ise buton azaltım sırasında 0 değerinin altına düşecek ve motor kendini 9000 rpm değerlerinde çalışmaya götürecektir.**

lcd.setCursor(1,0); //**Her rpm okundupunda LCD ekran üst satırdan bir kayarak yazcak.**

lcd.print("rpm:"); // **LCD’de üst satırdan bir kayarak sabit sabit olarak rpm yazacak.**

lcd.print(rpm);} //**Butonlarla girilen rpm değerini yazacak.**

if (azaltbasili == HIGH) // **Alttaki 5 kod satırı bir kod yapısını oluşturmaktadır.Bu kod yapısına göre azalt butonuna her basıldığında 1 kez azaltım içindir.**

{

while (1)

{

azaltbasili = digitalRead(azalt);

if (azaltbasili == LOW)

break;

}

}

break;

}

**3.Artırım işlemi:**

while (rpm>=9000) //**Rpm değerleri 9000 değerleri altında çalıştığında Döngüye girip alttaki işlemeleri yapcak.**

{ artirbasili =digitalRead(artir); //**Artir butonunun basılı olduğu durumun okunması**

if (artirbasili == HIGH) // **Eğer artır butonu basılı ise alttaki işlemleri yapacak**

{rpm=rpm+500 ; //**Buton her basıldığında rpm değerini 500 artıracak**

if (rpm<=9000) //**Eğer rpm değeri 9000 dan küçük olduğunda alttaki işlemi uygulayacak**

{rpm=9000;} //**Rpm değerini 9000 a eşitlicek. Rpm 9000 değerlerinin üstüne çıkması durumunda yapılan işlemdir.Eğer bunu yapmaz ise buton artırım sırasında 9000 değerinin üstüne çıkacak ve motor kendini 0 rpm değerlerinde çalışmaya götürecektir.**

lcd.setCursor(1,0); //**Her rpm okundupunda LCD ekran üst satırdan bir kayarak yazcak.**

lcd.print("rpm:"); // **LCD’de üst satırdan bir kayarak sabit sabit olarak rpm yazacak.**

lcd.print(rpm);} //**Butonlarla girilen rpm değerini yazacak.**

if (azaltbasili == HIGH) // **Alttaki 5 kod satırı bir kod yapısını oluşturmaktadır.Bu kod yapısına göre azalt butonuna her basıldığında 1 kez azaltım içindir.**

{ while (1)

{ azaltbasili = digitalRead(azalt);

if (azaltbasili == LOW)

break; } }

break;}

**4.Map işlemi:**

kademe=map(rpm,0,9000,0,255); //**Arduino işlemleri 8 bitte yapar.Yani 0 ile 255 arasında Yukardaki artırım azaltım işlemi sırasında 0’ın altına düştüğü durumda motorun hızlanması 9000 üzerinde çıktığı zaman motorun durması bu yüzdendir.**

**Map komutu 8 bitlik girilen bir veriyi istediğimiz değerler arasına orantılayarak çıkış vermesini sağlar.Girdiğimiz rpm değeri aslında 0 ile 255 arasındadır.Biz bunu bu komut sayesinde 0 ile 9000 arasında görürüz.**

**0 ile 255 aralığı ise 0 ile 5 v arası voltage değerleri arasında oranlanmıştır.Bu sayede hız kademesini okuruz.**

**5.Başlatma işlemi:**

baslabasili=digitalRead(basla); // **Başlat butonun basılı olduğu durumu okuyarak bir değişkene atar.**

if(baslabasili==HIGH){ // **Eğer başlat butonu basılı ise alttaki işlemleri yapar**

analogWrite (enba,kademe);} // **Başlat butonu basılı olduğu durumda Enba pinine (9 pin) Butonların belirlediği voltage değerine göre motoru başlatır.**

**6.Bitirme işlemi:**

bitirbasili=digitalRead(bitir); // **Bitir butonun basılı olduğu durumu okuyarak bir değişkene atar.**

if(bitirbasili==HIGH) // **Eğer bitir butonu basılı ise alttaki işlemleri yapar**

{analogWrite(enba,0);} // **Bitir butonu basılı olduğu durumda Enba pinine (9 pin) 0 V verir.Buna bağlı olarak motor durur.**

**7.Hız ölçüm işlemi:**

int rpms=getRPM(); //**Rpms ölçüldüğü fonksiyon rpms değilkenine atanmıştır.**

displayRPM(rpms); //**Rpms lcd ekrana yazılmıştır.**

}

int getRPM() //**Rpm fonksiyonunun ölçüldüğü fonksiyon**

{

int sayıcı=0; //**sayıcı değişkeni başlangıçta 0 a eşitlenmiştir.**

boolean kbayrak=LOW;// **Encoder diskinin boşluk olmadığı durumu kbayrağa ataması**

unsigned long simdikizaman=0; //**Simdikizaman değişkenini 0 a eşitle**

unsigned long baslazaman=millis();//**baslazaman değişkeni milis fonksiyonunun durumuna eşitle.Milis fonksiyonu milisaniye cinsinden sayma işlemi yapar.50 gün boyunca sayma işlemini gerçekleştirebilir.Makina 50 gün boyunca çalışabilir.**

while (simdikizaman <=basitzaman)// **Eğer şimdikizaman basit zamana eşitse döngüye gir**

{ if (digitalRead(encoder)==HIGH)//**Eğer encoderun bağlı olduu pine 5v gelirse alttaki işlemleri yap.Yani Encoder diskinin döndürülme sırasında delik oldğu kısımda bu komut uygulanacaktır.**

{kbayrak=HIGH; // **Üstteki durum için kbayrağı 5v yap**

}

if (digitalRead(encoder)==LOW && kbayrak==HIGH)//**Eğer encoderdan 0 v gerilim geliyosa ve kbayrak yükselen kenarda ise alttaki işlemi yap.Yani burda şunu demek istiyoruz.Daha önceden de bahsettiğimiz gibi yükselen kenar boşluğu düşen kenar boşluğun olmamasını teşkil ediyodur.Her kbayrak durumu boşluk durumuna atanmış ise ama encoder boşluk olmama durumunda ise işlemi yap. Şekil 8 de kare dalga halinde olan yükselen ve düşen kenarlar mevcuttur.Burda yükselen ve düşen kenarın toplamı 1 pulse olmuş olacak.Buna bağlı olarak Alttaki sayma işlemi yapılacaktır.**

{sayici++; //**Yukarıdaki şartlarda sayıcıyı 1 artır.**

kbayrak=LOW; // **Kbayrağı 0 yap.Çünkü 0 yapılmassa eğer encoder takılacak istenilen işlemi gerçekleştiremeyecektir.**}

simdikizaman=millis()-baslazaman;//**milis fonksiyonundan başlama zamanını çıkar.Şimdiki zaman değişlenine eşitle**}

int sayici2rpm = int(60000./float(basitzaman))\*sayici;//**Burada sayıcıya bağlı olarak rpm hesabı yapılmaktadır.1dk 60000 milisaniye eder.Bunu zaman değişkenine böldüğümüzde ve sayıcı nın değeri ile çarptığımızda rpm değerini buluruz.**

return sayici2rpm;// Fonksiyodan sayici2rpm değişkenine çek

}

void displayRPM(int rpms) //**rpm yazılması fonksiyonu**

{ lcd.clear();//**Her döngü döndüğünde yeni değerler için lcd temizler.**

lcd.setCursor(7, 1); //**rpm bilgisini lcd’nin 2. satır 7. Sütununa yazılacağının ayarlanması**

lcd.print(rpms,DEC);//**Rpm bilginin yazılması**

lcd.setCursor(0,1);//**bir sonraki yazılacak için 2.satır ve ilk stunun ayarlanamsı**

lcd.print("RPMS:");// **sabit RPMS nin yazısının yazılması.Böylece yazının karşısına RPM yazılmış olcak**

}

**KAYNAKÇA**

[1]….[15]------------------------- <https://www.makaleler.com/santrifuj-nedir-ozellikleri-cesitleri>

[16],[17],[18],[20],[21],[22],[23],[24],[25],[26],[28]------------------------------Google görseller

[19]------------------------------------------------------ <https://www.robotistan.com/rs545-dc-motor>

[21]---------------------------------------<https://www.robotistan.com/l298n-voltaj-regulatorlu-cift-motor-surucu-arti?language=tr&h=333c963a&gclid=EAIaIQobChMI9fKBuoir5gIVC8jeCh3-4ALTEAQYAyABEgJUAvD_BwE>

[27]--------------------- <https://www.projehocam.com/arduino-cesitleri-ve-programlama/>