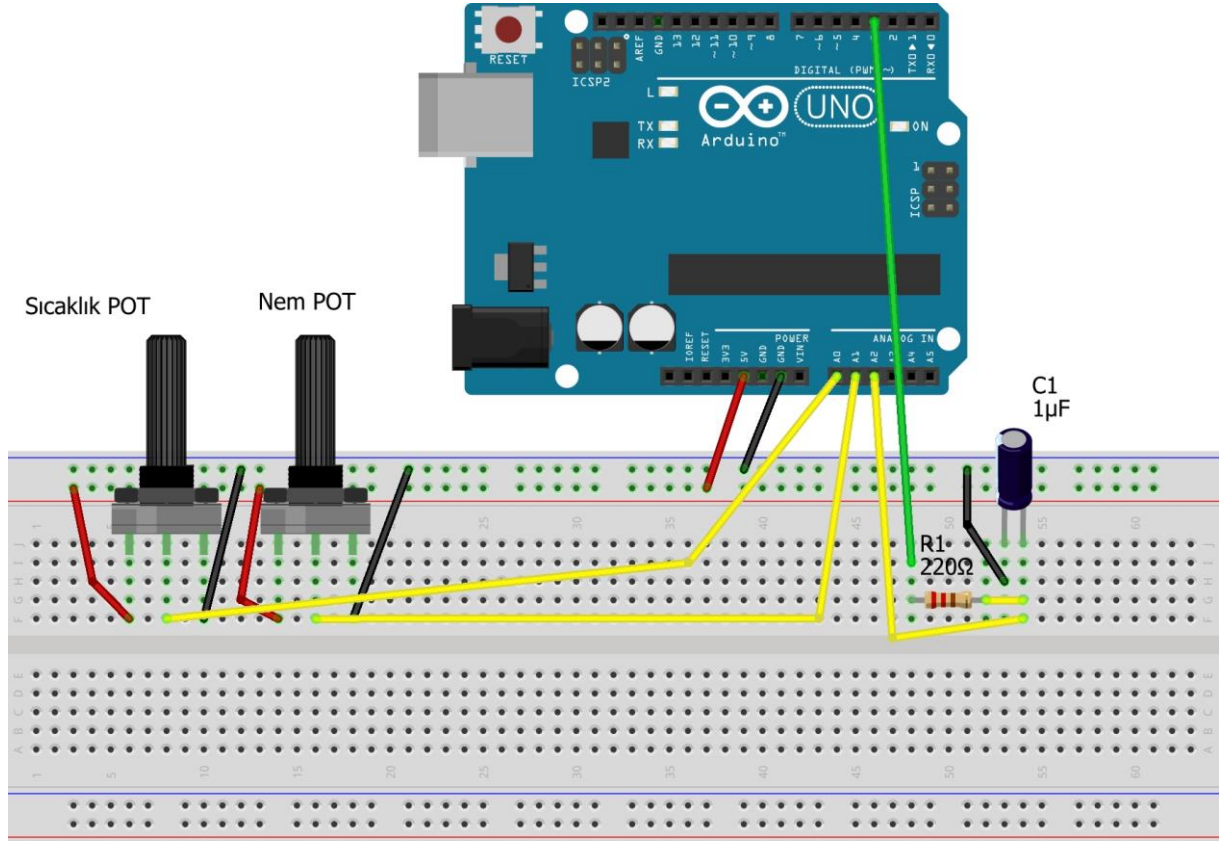


MOTOR DEVRİ ve MOTOR ÇALIŞMA SÜRESİ TAHMİNİ için BULANIK ÇIKARSAMA SİSTEMİ TASARIMI

Bu program girişleri sıcaklık ve nem, çıkışları ise motor devri ve çalışma süresi olan bir bulanık mantık kontrol sistemini çalıştırmak üzere yapılmıştır. Matlab ile Arduino haberleşmesi için destek paketi kullanılmıştır. Devre şeması:



fritzing

Bu şemada motor devrini gösteren PWM sinyali D3'ten alınıp bir kondansatör aracılığıyla analoğa çevrilmiş ve A2 girişinden Matlab ortamına aktarılmıştır. Böylece grafik üzerinden motor devri ve çalışma süresi izlenebilmiştir. Direnç ve kondansatör yerine D3 pwm çıkışına bir sürücü devre ile DC motor da bağlanabilir.

Matlab kodları:

```
clc;clear;close all;
delete(instrfindall);
a=arduino('COM4','Mega2560');
b = readfis('Fuzzy4'); % oluşturulan bulanık mantık kontrol
sistemi import edildi.
x=1;j=1;
while x
sckSensor=readVoltage(a,'A0'); %sıcaklık ve nem okunuyor.
nemSensor=readVoltage(a,'A1');
```



AKÜ TEKNOLOJİ FAKÜLTESİ MEKATRONİK MÜHENDİSLİĞİ

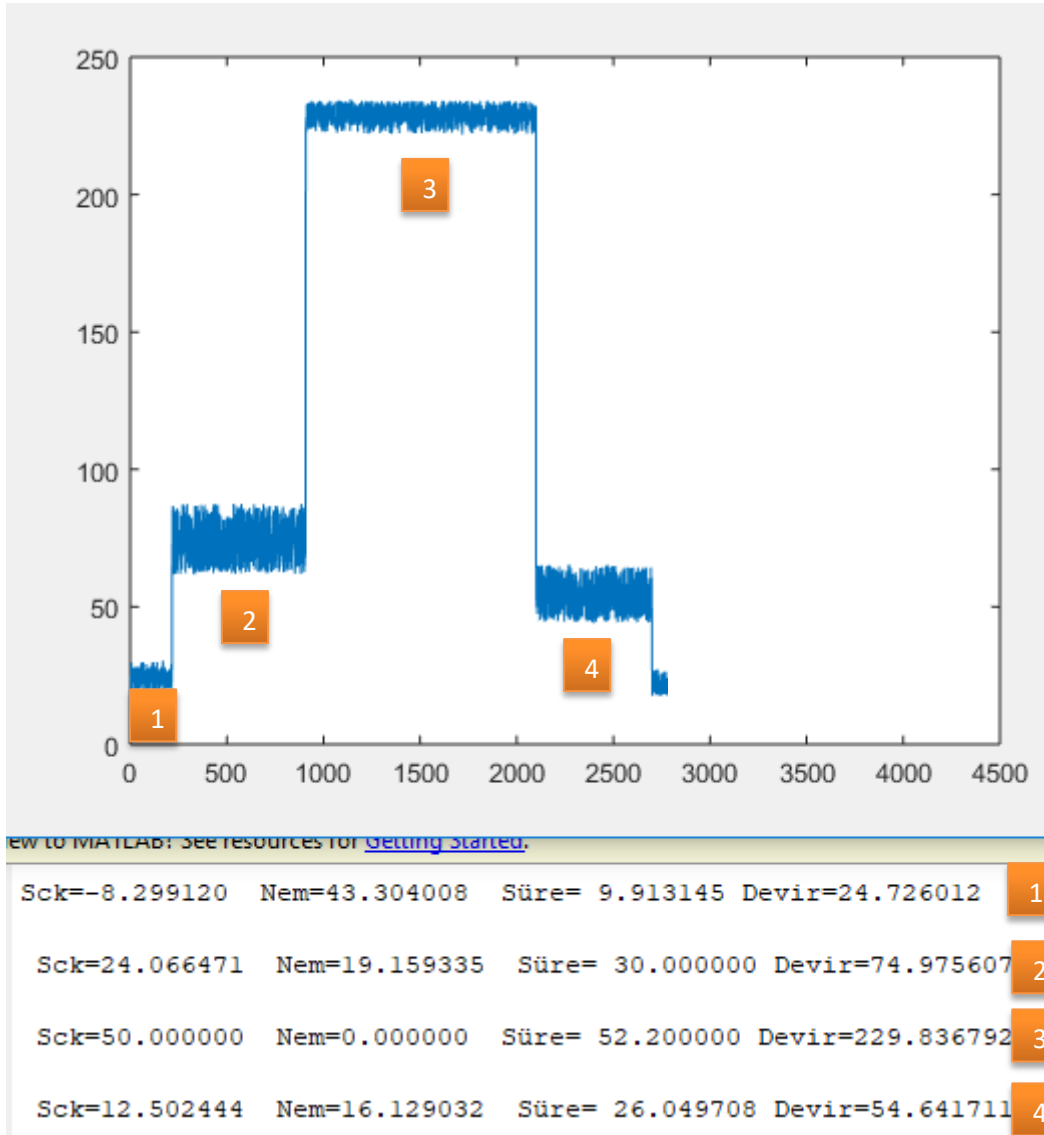


```
sicaklik(x)=14*sckSensor-20;%sensörden gelen değer (-  
20,50)santigrat arasına normalize edildi.  
nem(x)=20*nemSensor;%sensörden gelen değer (0,100) arasına  
normalize edildi.  
y=evalfis([sicaklik(x) nem(x)],b);%çıktılar fuzzy kontrol  
sistemşinden elde edildi.  
devir(x)=y(1)*(5/255); %devir değeri lede bağlanıp izlendiği  
için 0-5v arasına çekildi. Motor bağlansaydı bu işleme gerek  
olmayacaktı.  
sure(x)=y(2);  
fprintf('Sck=%f Nem=%f Süre= %f Devir=%f \r\n  
,sicaklik(x),nem(x),y(2), y(1));  
tstart=tic;  
% yyy=toc(tstart);  
while(toc(tstart)<sure(x))  
% ZamanliCalistir(a, devir(x),y(2));%belirli devirde ve sürede  
çalışmayı sağlayan fonksiyon.  
writePWmVoltage(a, 'D3',devir(x));  
PwmVoltage(j)=readVoltage(a, 'A2')*255/5;  
% fprintf('Sck=%f Nem=%f Süre= %f Devir=%f \r\n  
,sicaklik(x),nem(x),y(2), y(1));  
% text(10,10,sprintf('sck=%f nem=%f - devir=%f  
sure=%f\n',sicaklik(x),nem(x),devir(x),sure(x)));  
plot(PwmVoltage);  
drawnow;  
j=j+1;  
end  
% fprintf('Pot1=%f Pot2=%f Süre= %f Devir=%f \r\n  
,sicaklik(x),nem(x),y(2), y(1));  
x=x+1;  
end
```

Bu programın grafiksel çıktısı:



AKÜ TEKNOLOJİ FAKÜLTESİ MEKATRONİK MÜHENDİSLİĞİ



Motor devirleri ve motor çalışma süreleri numaralar ile yukarıdaki grafikte belirtilmiştir.

Yukarıdaki çalışmada Arduino destek paketi kullanılmıştı. Şimdi ise seri porttan veri göndererek aynı işlemi gerçekleştirelim. Seri port iletişimde Arduino tarafına da kod yazmamız gerekir. Kod aşağıda verilmiştir.

```
#define ADC0 A0 //ADC0 analog 0 pinini temsil ediyor.  
#define ADC1 A1 //ADC1 analog 1 pinini temsil ediyor.  
#define ADC2 A2 //ADC2 analog 2 pinini temsil ediyor.
```

```
int pwm;  
int ledPin=3;  
String a0;  
String a1;  
String a2;
```

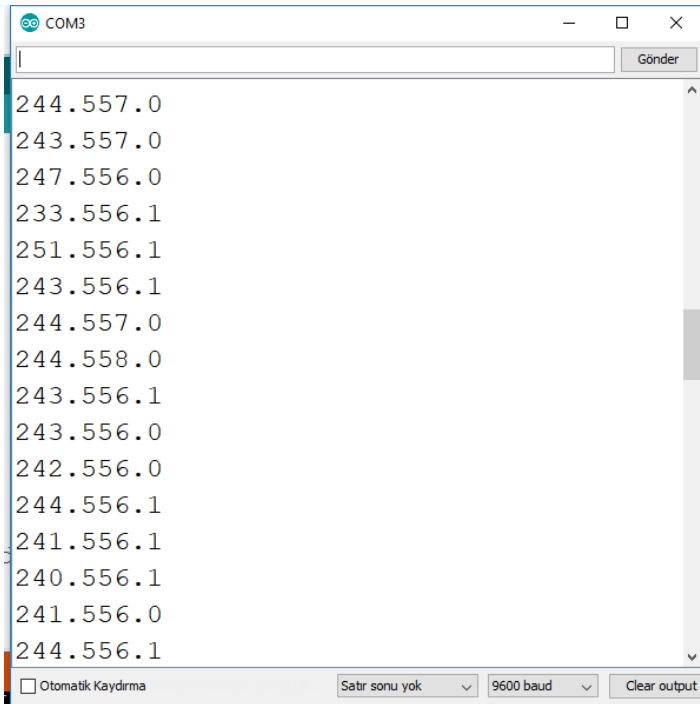


AKÜ TEKNOLOJİ FAKÜLTESİ MEKATRONİK MÜHENDİSLİĞİ



```
void setup() {  
  
  Serial.begin(9600);  
  pinMode(ledPin,OUTPUT);  
}  
void loop() {  
  
  if(Serial.available()>0) // Eğer okunacak seri data varsa  
  {  
    pwm=Serial.parseInt(); // veriyi oku  
  
    analogWrite(ledPin, pwm);  
  }  
  a0=analogRead(ADC0);  
  Serial.print(a0+"." ); // buradaki POT1 sıcaklık değeridir  
  delay(100);  
  a1=analogRead(ADC1);  
  Serial.print(a1+"." ); // buradaki POT2 nem değeridir  
  delay(100);  
  a2=analogRead(ADC2);  
  Serial.println(a2); // buradaki matlab dan gönderilen pwm'in oluşturduğu analog voltaj değeridir  
  delay(100);  
}
```

Bu kod ile Matlab tarafına gönderilen veri aşağıdaki gibidir. A0.A1.A2 şeklinde formatlanmıştır.





AKÜ TEKNOLOJİ FAKÜLTESİ MEKATRONİK MÜHENDİSLİĞİ



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% Bu program girişleri sıcaklık ve nem, çıkışları ise motor  
devri ve  
% çalışma süresi olan bir bulanık mantık kontrol sistemini  
çalıştırmak  
% üzere yapılmıştır. Matlab Arduino haberleşmesi Seri  
haberleşmedir.  
clc;  
clear;  
close all;  
delete(instrfindall);  
comport=serial('COM4', 'BaudRate',9600);  
fopen(comport);  
  
b = readfis('Fuzzy4'); % oluşturulan bulanık mantık kontrol  
sistemi import edildi.  
x=1;j=1;  
while x  
sckSensor=fscanf(comport,'%s');% seri porttan pot1 değeri  
okunuyor.  
%Gelen verinin SCK.NEM.PWM şeklindedir. Bu veriyi ayırarak  
yerli yerine  
%yazıyoruz.  
s=strsplit(sckSensor, '.');  
sckSensor=str2double(s(1)); %s(1) sıcaklık verisi  
sckSensor=sckSensor*5/1024; %gelen değer 0..1024 aralığından  
0..5 aralığına çekildi.  
nemSensor=str2double(s(2));%s(2) nem verisi  
nemSensor=nemSensor*5/1024;%gelen değer 0..1024 aralığından  
0..5 aralığına çekildi.  
sicaklik(x)=14*sckSensor-20;%sensörden gelen değer (-  
20,50)santigrat arasına normalize edildi.  
nem(x)=20*nemSensor;%sensörden gelen değer (0,100) arasına  
normalize edildi.  
y=evalfis([sicaklik(x) nem(x)],b);%çıkışlar fuzzy kontrol  
sisteminden elde edildi.  
devir(x)=y(1)*(5/255); %devir değeri PWM gönderileceği için 0-  
5v arasına çekildi. Motor bağlansaydı bu işleme gerek  
olmayacaktı.  
sure(x)=y(2); %y(2) motor çalışma süresidir. Sn. cinsinden  
alınmıştır.  
fprintf('Sck=%f Nem=%f Süre= %f  
Devir=%f\n',sicaklik(x),nem(x),y(2), y(1));  
tstart=tic;  
while(toc(tstart)<sure(x))  
fprintf(comport, '%d',round(y(1)));  
pause(0.5);
```

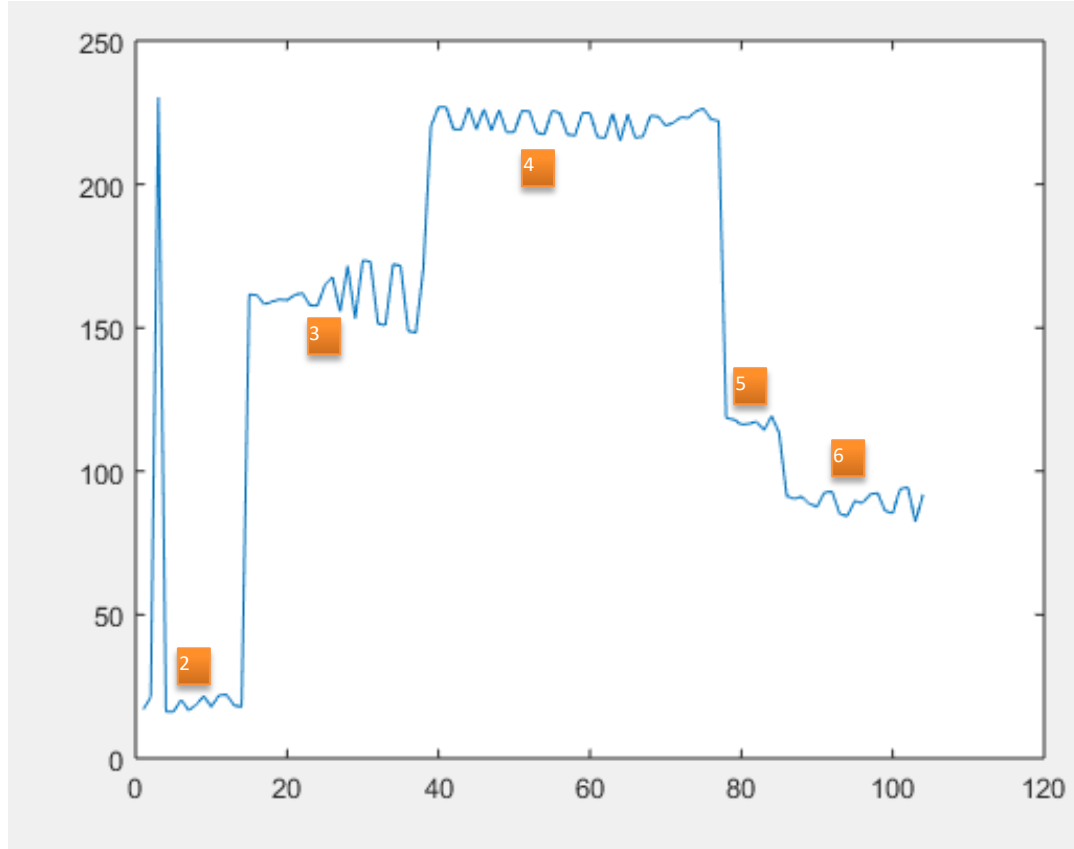


AKÜ TEKNOLOJİ FAKÜLTESİ MEKATRONİK MÜHENDİSLİĞİ



```
Pwm=fscanf(comport,'%s');  
s2=strsplit(Pwm,'.');  
Pwm=str2double(s2(3));  
Pwm=Pwm*5/1024;  
PwmVoltage(j)=Pwm*255/5;  
plot(PwmVoltage);  
drawnow;  
j=j+1;  
end  
x=x+1;  
end
```

Grafiksel çıktısı aşağıdaki gibi oluşmaktadır .



```
Sck=3.789063 Nem=5.957031 Süre= 7.893174 Devir=19.596270 1  
Sck=0.166016 Nem=0.000000 Süre= 9.028880 Devir=22.499824 2  
Sck=49.316406 Nem=56.640625 Süre= 30.000000 Devir=160.023819 3  
Sck=40.292969 Nem=0.683594 Süre= 50.119598 Devir=223.025957 4  
Sck=48.837891 Nem=76.562500 Süre= 9.735823 Devir=117.040721 5  
Sck=26.005859 Nem=45.703125 Süre= 10.618562 Devir=90.970037 6
```