



SERİ PORTLAR ÜZERİNDEN “MATLAB – ARDUINO” HABERLEŞMESİ

Matlab Arduino haberleşmesi için 2. yöntem seri haberleşmedir. İlk yöntemde Arduino üzerinden kod yazmaya gerek yoktu ancak bu yöntemde Arduino IDE üzerinden kod yazmamız gerekir.

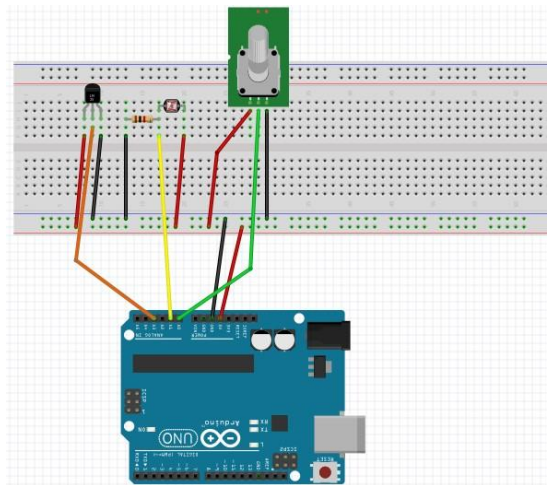
Bu kodlar ile toplanan veriler seri port üzerinden Matlab yazılımının kurulu olduğu bilgisayara gönderilir.



Arduino IDE açık kaynak kodlu bir yazılımdır. Yazılım dili oldukça basit ve anlaşılır bir dildir. İnternet üzerinde bolca kaynak bulunmaktadır. <http://arduino.cc> adresi üzerinden indirilebilir.

Seri port üzerinden Matlab Arduino haberleşmesi için Arduino tarafında verileri seri port üzerinden gönderen, Matlab tarafında verileri seri port üzerinden alan birer program yazılması gerekir.

Kurulması gereken devre aşağıdaki gibidir.



Arduino tarafında seri porttan veri göndermek için yazılması gereken program:

```
#define ADC1 A0 //ADC1 analog 0 pinini temsil ediyor. POT bağlı
#define ADC2 A1 //ADC2 analog 1 pinini temsil ediyor. LDR bağlı
void setup() {
  // Başlangıçta seri iletişim hızını 115200 bps hızına set ediyoruz. Hız 9600 vb. de olabilir
  Serial.begin(115200);
}
```

```
void loop() {
  // Sürekli run edilecek kodlarımız burada kullanılıyor.
  //Pot'tan okunan değer seri porttan dışarı gönderiliyor.
  Serial.println(analogRead(ADC1));
  delay(100); // analog dijital çeviriciye süre tanımak için 100 ms. gecikme ver
  //Ldr değeri seri porttan dışarı gönderiliyor.
  Serial.println(analogRead(ADC2));
  delay(100);
}
```

Görüldüğü üzere Arduino'dan dışarıya veri gönderebilmek için

Serial.println(Veri) fonksiyonunu kullanıyoruz.

Matlab tarafında yazılması gereken program.

```
delete(instrfindall);
clc;
clear;
%Seri port bağlantısı tanımlandı.
%Burada seri port arduino'nun bağlı olduğu seri port olmalı.
comport = serial('COM7','BaudRate',115200);
fopen(comport); %port açılıyor.

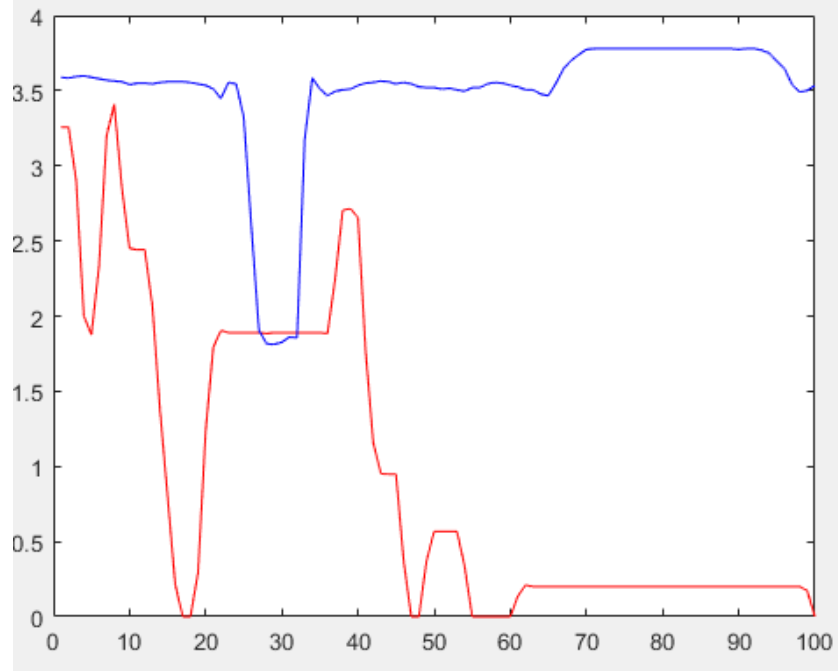
%arduino üzerindeki sensörlerden aldığımız her değeri biz dizide (array)
% tutmamız gerekir. Üç sensörümüz var.
%1.) Potansiyometre 2.) Işık sensörü LDR
%Dizide indis değişkenini "i" olarak seçtik.
%Potansiyometre verilerini "pot" dizisinde
%LDR verilerini "ldr" dizisinde
%100 adımlık bir for döngüsü içinde tüm değerleri okuyoruz.

for i=1:100
% seri porttan pot değeri okunuyor ve pot isimli diziye yazılıyor.
pot(i)=fscanf(comport,'%d');
% pause(0.1); %gerektiğinde iki okuma arasında biraz bekleme yapılabilir.
% seri porttan ldr değeri okunuyor ve ldr isimli diziye yazılıyor.
ldr(i)=fscanf(comport,'%d');
%grafik çiziliyor.
plot(pot,'r'); kırmızı renkte çiz
hold on
plot(ldr,'b')
drawnow;
end
fclose(comport);%port kapatılıyor ve siliniyor.
delete(comport);
```

Programın txt çıktıları:

```
Potansiyometre=1.89 LDR=3.33
Potansiyometre=1.89 LDR=2.60
Potansiyometre=1.89 LDR=1.90
Potansiyometre=1.88 LDR=1.82
Potansiyometre=1.89 LDR=1.81
Potansiyometre=1.89 LDR=1.83
Potansiyometre=1.89 LDR=1.86
Potansiyometre=1.89 LDR=1.86
Potansiyometre=1.89 LDR=3.17
Potansiyometre=1.89 LDR=3.58
Potansiyometre=1.89 LDR=3.51
Potansiyometre=1.88 LDR=3.47
Potansiyometre=2.25 LDR=3.50
Potansiyometre=2.71 LDR=3.51
Potansiyometre=2.71 LDR=3.51
Potansiyometre=2.66 LDR=3.54
Potansiyometre=1.77 LDR=3.55
```

Programın grafik çıktıları:



Yukarıdaki Matlab programında dikkat edilmesi gereken husus; pot ve ldr sensör verilerinin sırayla okunmasıdır. Eğer sıralamadaki senkronizasyon bir şekilde kaybedilirse sensör verileri birbirine karışabilir.

Bu durumu çözebilmek için Arduino'dan ön etiketli veri göndermek ve Matlab üzerinden etikete göre veri almak gerekir.

Ayrıca bir diğer önemli konu ise seri haberleşme ile Matlab'dan Arduino'ya veri gönderebilmektir. Şu ana kadar sadece Arduino'dan Matlab'a veri gönderdik. Aşağıdaki kodlar ile

Arduino'dan Matlab'a ön etiketli veriyi seri port üzerinden göndereceğiz.

Matlab'dan da Arduino'ya seri port üzerinden veri gönderip Arduino üzerinde bir led yakacağız.

Arduino'daki program:

```
#define ADC0 A0 //ADC0 analog 0 pinini temsil ediyor.  
#define ADC1 A1 //ADC1 analog 1 pinini temsil ediyor.  
  
String pot;  
String ldr;  
int veri;  
int ledPin=46;  
void setup() {  
  
  Serial.begin(115200);  
  pinMode(ledPin,OUTPUT);  
}  
  
void loop() {  
  
  if(Serial.available()>0) // Eğer okunacak seri data varsa bu if ifadesi çalışır.  
  {  
    veri=Serial.parseInt(); // veriyi oku ve integer'a çevir.  
  
    if(veri==1)  
      digitalWrite(ledPin,HIGH); // ledi yak  
    else  
      digitalWrite(ledPin,LOW); // ledi söndür  
  }  
  
  pot=analogRead(ADC0);  
  Serial.println("POT_"+ pot); // buradaki POT_ ön etikettir.  
  delay(100);  
  ldr=analogRead(ADC1);  
  Serial.println("LDR_"+ldr); // buradaki LDR_ ön etikettir.  
  delay(100);  
}
```

Matlab tarafındaki program biraz daha karışık. Kodu anlayabilmek için lütfen aşağıdaki açıklamalara bakınız.

```
delete(instrfindall);
clc;
clear;close all;
comport = serial('COM7','BaudRate',115200);
fopen(comport);
sinir=21;
pot=[];
ldr=[];
i=1;j=1;k=1;

while (i<sinir || j<sinir )
a=fscanf(comport,'%s'); %string olarak okunuyor.
%Gelen veriler alt çizgi '_' karakterine göre bölünüyor.
[d1 d2]=strsplit(a,'_');
    if (strcmp(d1(1),'POT'))%etiket "POT" ise pot dizisine yazdırılıyor.
        pot(i)=str2double(d1(2));
        if (pot(i)>900) %pot değeri 900'den büyükse arduino'ya "1" gönder.
            fprintf(comport,'%d',1);
        else
            fprintf(comport,'%d',0);% 900'den küçük ise arduino'ya "0" gönder.
        end
        i=i+1;
    elseif (strcmp(d1(1),'LDR')) )%etiket "LDR" ise ldr dizisine yazdırılıyor.
        ldr(j)=str2double(d1(2));
        j=j+1;
    end
plot(pot, 'b');
hold on;
plot(ldr, 'r');
drawnow;
end
fclose(comport);
delete(comport);
```

Kod açıklaması:

Görüldüğü üzere COM (seri) portlardan veri okumak için *fscanf()*, seri porttan veri göndermek için ise *fprintf()* fonksiyonları kullanılıyor.

Öncelikle seri portta bekleyen değeri okuyalım ve “a” değişkenine atayalım.

```
>> a=fscanf(comport,'%s');
```

Örnek olarak seri porttan okunan değer “a=POT_2.42” olsun.

```
>>[d1 d2]=strsplit(a,'_');
```

Burada *strsplit()* fonksiyonu “_” karakterine göre a değerini böler.

d1 değişkenine [“POT”, “2.42”] değerlerini, *d2* değişkenine “_” değerini yazar. Bu durumda;

d1(1)=”POT” ve *d1(2)*=”2.42” olur. Ancak dikkat edelim her ikisi de string’dir. Yani 2.42’nin sayısal bir değeri şimdilik yoktur. 2.42’ye sayısal bir değer kazandırmak için *str2double()* fonksiyonu kullanılır.

```
>> pot(i)=str2double(d1(2))
```

fonksiyonu ile 2.42 değeri artık sayısal hale gelmiş olur ve Matlab tarafında *pot* isimli dizinin *i*. elemanı olarak yazılır.

```
>> if (strcmp(d1(1),'POT'))
```

 demek aslında if *d1(1)*== ’POT’ demektir. Bu şekilde *d1(1)*’deki string değer “POT” ise parantez içi TRUE olur ve program buna göre dallanır.

Matlab üzerinde aşağıdaki gibi bir menü yapalım:

Led değerini giriniz 1 veya 2 (1=ON, 2=OFF, 0=çıkış):

Menüden 1'e basıldığında Arduino üzerinde led yanacak, 2'ye basıldığında led sönecek. O'a basıldığında programdan çıkacak. Haberleşme seri port üzerinden olacak.

Yanıt:

Matlab tarafı:

```
clear;
clc
yanit=1; %kullanıcının cevabı
arduino=serial('COM7','BaudRate',9600); %seri iletişim nesnesi
oluşturuldu
fopen(arduino); % seri iletişim başlatıldı
while yanit
    fprintf(arduino,'%d',yanit); % cevap değişkenini arduinoya gönder
    yanit=input('led değerini giriniz 1 veya 2 (1=ON, 2=OFF, 0=çıkış):
'); % cevap değişkeni için kullanıcıya sor
end
fclose(arduino); % arduino ile iletişimi kes
```

Arduino tarafı:

```
int ledPin=2;
int veri;

void setup()
{
    pinMode(ledPin,OUTPUT);
    Serial.begin(9600);
}

void loop()
{

    if(Serial.available()>0)
    {
        veri=Serial.parseInt();
        Serial.println(veri);
        if(veri==1)
            digitalWrite(ledPin,HIGH);
        else
            digitalWrite(ledPin,LOW);
    }
}
```