

Proje 1 : Arduino Uno ile Matlab ortamının entegrasyonu

Bu projede Arduino ve Matlab arasındaki iletişimi sağlayıp, akıllı telefon üzerinden verilen belli frekanstaki sesin Arduinoya bağlı bir mikrofon tarafından okunup, Matlab'da verilen frekansın tespitini yapmaya çalıştım.

Proje Kısımları:

- 1- Arduino Uno ile Matlab arasındaki bağlantının kurulması
- 2- Mikrofonun Arduino ile okunup bilginin Matlaba aktarılması
- 3- Gelen bilginin Matlab üzerinde ölçeklendirilmesi ve frekansın bulunması için gerekli analizlerin yapılması

1-Arduino Uno ile Matlab arasındaki bağlantıyı kurmak:

Arduino ile Matlab arasında bağlantıyı kurmanın iki ayrı yolunu bulabildim. Birincisi Matlab'ın kendi hazırladığı bir eklenti ile Arduino üzerine tamamen kontrolü almak, ikincisi ise Arduinonun Serial Port'un dan gelen veriyi okumak.

Öncelikle ilk metodu anlatmak isterim. Öncelikle bu yöntemin kullanılması için Simulink paketinin bilgisayarda yüklü olması gerekiyor.

Direk olarak aşağıdaki Arduino bağlama kodunu girince Matlab uyarı verip yüklemeniz gereken eklentiyle ilgili linki gösteriyor.

```
a=arduino();
```

Bu şekilde açılmazsa aşağıdaki kodu girip eklenti yükleme kısmına girip Arduino için gereken kodu bulup yükleme yapabilirsiniz.

```
supportPackageInstaller
```

Eklentinin yüklenmesi için bir Matlab hesabına sahip olmak gerekiyor. Ücretsiz bir şekilde yeni hesap alınıp eklenti yüklenebilir.

Bu eklenti sayesinde Matlab üzerinden Arduino'nun direk olarak kontrolü mümkün, çalışma tarzı olarak da biraz değişiklik mevcut. Matlab'da komut satırına istediğimiz arduino kodunu girip kodun sadece o kısmında değişiklik yapmak mümkün. Normalde her bir değişiklik için tüm kodu açıp oradan değişiklik yapmak gerekir, ama Matlab eklentisiyle bu işe yeni bir metod getirmişler. Her ne kadar çalışma mantığı temelinde aynı olsa da, bu şekilde Matlab üzerinde çalışmaktan daha çok zevk aldığımı söylemek isterim.

Matlabın Arduino eklentisini kullanırken standart arduino komutlarını kullanamıyoruz.

Matlab bu eklenti için ayrı bir kütüphane oluşturmuş. Kütüphanedeki komutların açıklamalarına aşağıdaki linkten ulaşabilirsiniz.

<http://www.mathworks.com/help/supportpkg/arduinoio/>

Bu yöntemin bir eksiği ise, her yazdığınız komut için her seferinde Matlab kodun o kısmını Arduino'ya yüklüyor o kısmın çıktısını alıyor ve bir sonraki komuta geçiyor. Bu yüzden normal çalışma yöntemine göre biraz daha yavaş çalışıyor. Bu sebeple frekans analizi projesi için bu bağlantı yöntemi benim için yeterince hızlı gözükmedi. Sampling Rate'i arttırmak için yukarıda değindiğim ikinci yöntemi, direk olarak sadece Serial Port'u okuma yöntemini kullandım.

Matlab üzerinden Arduino Serial Portuna bağlanmak için herhangi bir eklenti yüklememize gerek yok. Ama bu metod biraz da manuel olduğu için biraz düzenleme gerektiriyor. İnternette yaptığım araştırma sonucunda ayarlamaların nasıl olması gerektiğini anlatan bir websitesine ulaşabildim.

<http://www.matlabarduino.org/serial-communication.html>

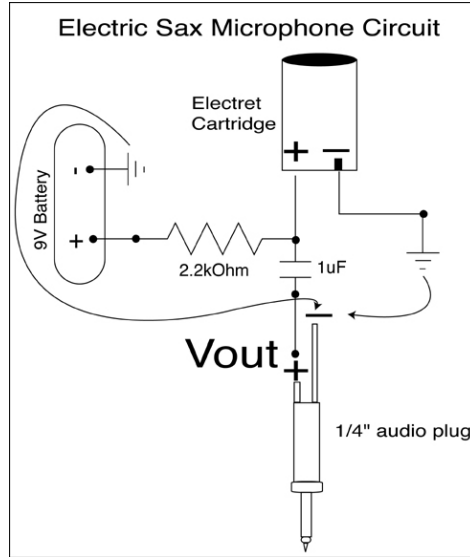
Bu yöntemi kullanırken Arduino'ya yüklenecek kodun setup kısmının ve Matlab kodunun düzenlenmesi gerekiyor. Yukarıdaki linkteki videodaki kodları biraz değişiklik yaparak kullandım. Arduino'da ve Matlab'da kullandığım koda (FrekansBulucu.ino ve ac.m) ekten ulaşabilirsiniz. BaudRate olarak 115200 yi tavsiye ediyorum. Bu yöntem ile 2360 Hz lik Sampling Rate'e ulaşabildim. Anladığım kadarıyla Sampling Rate ölçülebilecek en yüksek frekansı belirliyor. Bu yüzden değeri elimden geldikçe yükseltmeye çalıştım.

İkinci yöntemi kullanmak ekstra kodlama ve optimizasyon gerektiriyor. Kod üzerinde çok fazla değişiklik yapamıyorsunuz. Parametreleri kod çalışırken değiştiremiyorsunuz. Serial Porttan Matlaba aktarılan bilginin düzenlensi, ölçeklendirilmesi ve analize hazırlanması gerekiyor. Ama bunlardan ödün vererek daha yüksek bir hızda veri alabiliyoruz. Bu yüzden projede bu yöntemi kullandım.

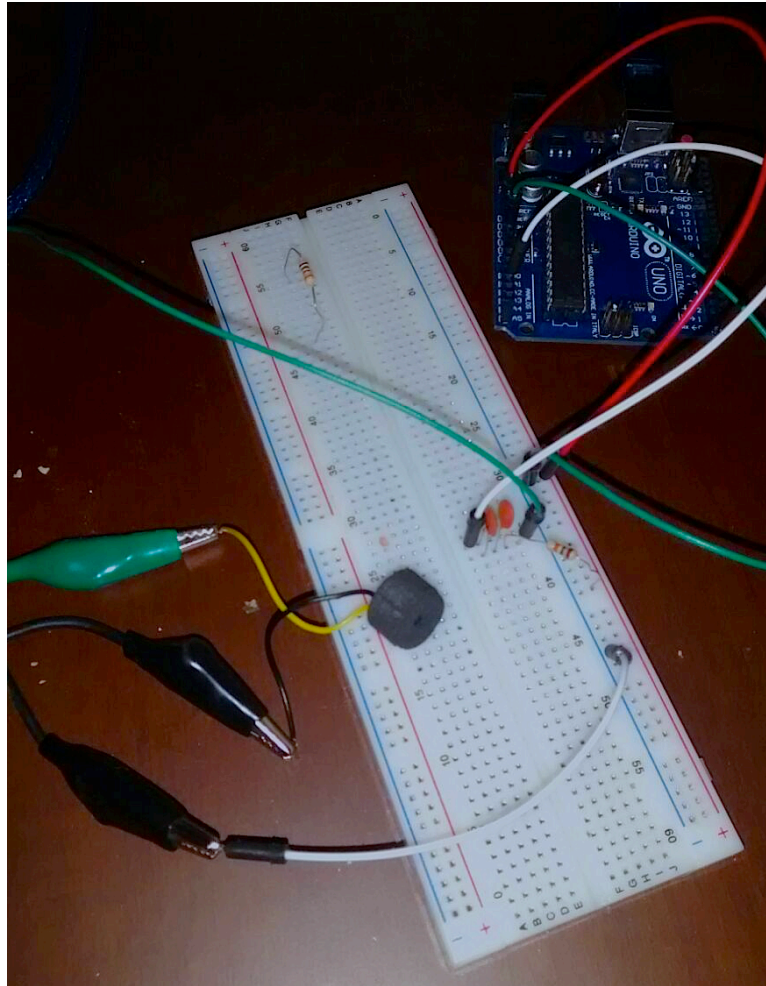
2- Mikrofonun Arduino ile okunup bilginin Matlaba aktarılması:

Öncelikle en çok bu aşamada sıkıntı yaşadığımı söylemek isterim. Mikrofonlar üzerinde daha önceden çalışmamıştım ve teorik olarak nasıl çalıştıklarını bilmediğim için Arduino'ya bağlarken zorluklar yaşadım.

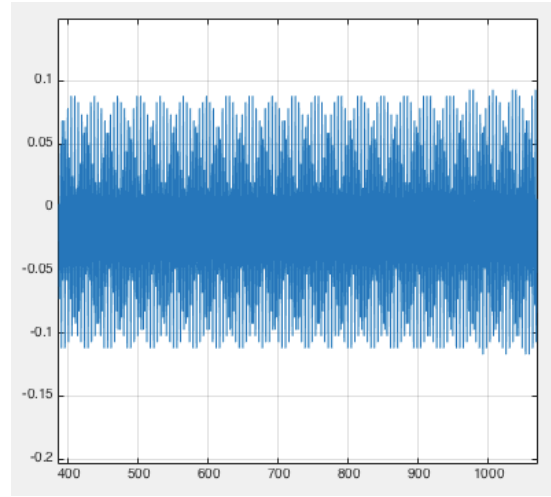
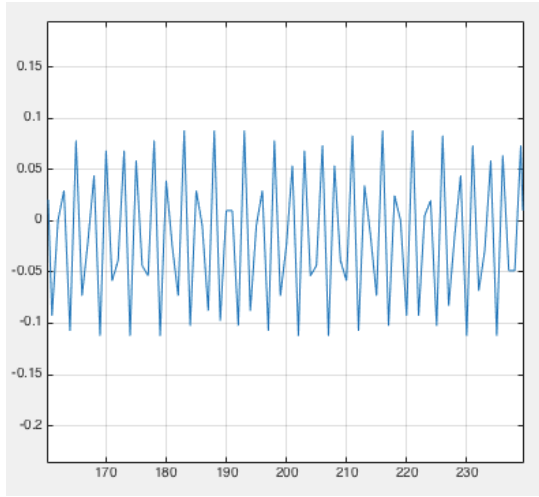
Projede eski bir telefonda çıkarttığım electret mikrofonu kullandım. Bu yüzden mikrofon için bir ödeme yapmamış oldum. Lakin bu yüzden mikrofon için kullanacağım devredeki kapasitör ve direnç değerini kendim deneyerek bulmam gerekti.



Yukarıda gösterilen devreyi Arduino'nun +5 Volt ve Ground çıkışlarını kullanarak kurdum. 1uF kapasitör yerine iki tane 0.1uF kapasitörü paralel bağladım (0.2uF) ve 2.2k direnç yerine de 10k direnç kullandım. Yukarıdaki şekilde Vout diye gösterdiği çıkışı Arduino'nun analog pininden (A3) okudum.



Mikrofon devresi ve Arduino bağlantısı



Analog Pin'den gelen datanın Matlab üzerinde çizilmiş grafiği

Bu değerlerdeki devre elemanlarını kullanarak analog çıkışından 3.25 V çevresinde ± 0.1 V luk bir signal alabildim.

Her seferinde 5000 kere voltaj değeri okuyacak şekilde kodları yazdım ve her seferinde çok az sapmalarla 2120 milisaniyede arduino bir turu tamamladı. Her turda tamamlanma süresini ve sampling rate i analizde kullanmak üzere hesaplayacak şekilde kodu yazdım. Bu koda ekteki test.m dosyasından ulaşabilirsiniz.

Gerçek zamanlı grafik çizdirdiğim veya değerleri okumaya çalıştığım zaman programın yavaşladığını ve Sample Rate'ın düştüğünü gözlemledim bu yüzden, devrenin son halini verdikten ve değerleri tam olarak okuduğumdan emin olduktan sonra gerçek zamanlı olarak değerleri ekrana bastırmadım.

Arduino'nun Serial Portu'ndan verileri okurken yaşadığım bir diğer sıkıntı ise programın 6 tekrardan sonra yanlış değerler vermesi oldu. Bu da analiz yaparken frekans ölçeğinin kaymasına sebep olduğu için çok yanlış değerler okumama sebep oldu. Bunun sebebi muhtemelen Arduino'nun belleğinin dolmasıyla alakalı olduğu düşünüyorum. Lakin bunu Arduino'ya yazdığım kod üzerinden düzeltemedim. Bu problemi serial portu kapatıp tekrar açarak düzeltebildim. Bunu da Matlab kodunun son halinde otomatikleştirdim ve 6 turdan sonra otomatik olarak serial portu kapatıp açmasını sağladım.

Belirli bir frekanstaki ses dalgası yollamak için akıllı telefonlarda çalışan bir uygulamayı telefonuma kurdum. Bu noktada dikkatli olunması gerekiyor, çünkü indirilen programın gösterdiği frekansı verip vermediğini test etmeniz önemli. İlk kullandığım program doğru değerleri vermediği için projede biraz aksaklığa sebep olmuştu. Size XYZ-App'in hazırladığı Signal Generator programını önerebilirim.

<https://play.google.com/store/apps/details?id=com.xyz.signal>

3- Gelen bilginin Matlab üzerinde ölçeklendirilmesi ve frekansın bulunması için gerekli analizlerin yapılması :

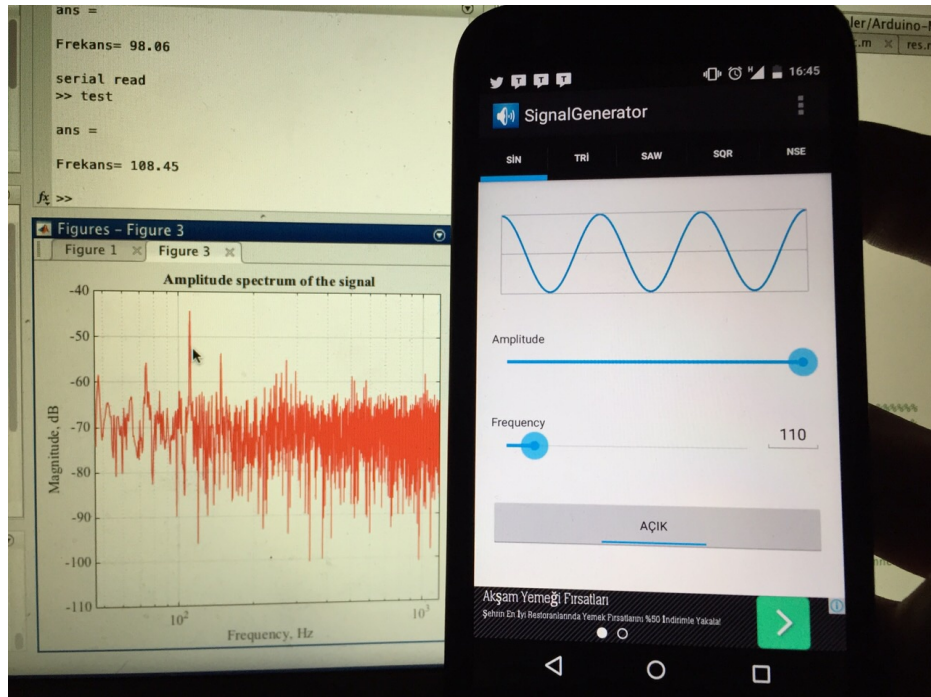
Projenin bu kısmına diğer kısımlardan daha önce başladım, çünkü daha önceden ses analizi ile hiç ilgilenmemiştim ve konuya tamamen yabancıydım. Sesin nasıl aktarıldığını, frekansın ne olduğunu ve Matlab üzerinden nasıl hesaplanacağı üzerine araştırma yaptım. Bilgisayarın mikrofonunu kullanarak Matlab üzerinden denemeler yaptım. İnternette de herkese açık olarak sunulmuş, Hristo Zhivomirov tarafından 2014 yılında hazırlanmış bir koddan faydalandım. Kodun gerekli kısımlarını kullanıp gerekli düzenlemeleri yaptıktan sonra bunu bilgisayarın mikrofonunu kullanarak çalıştırmayı başardım.

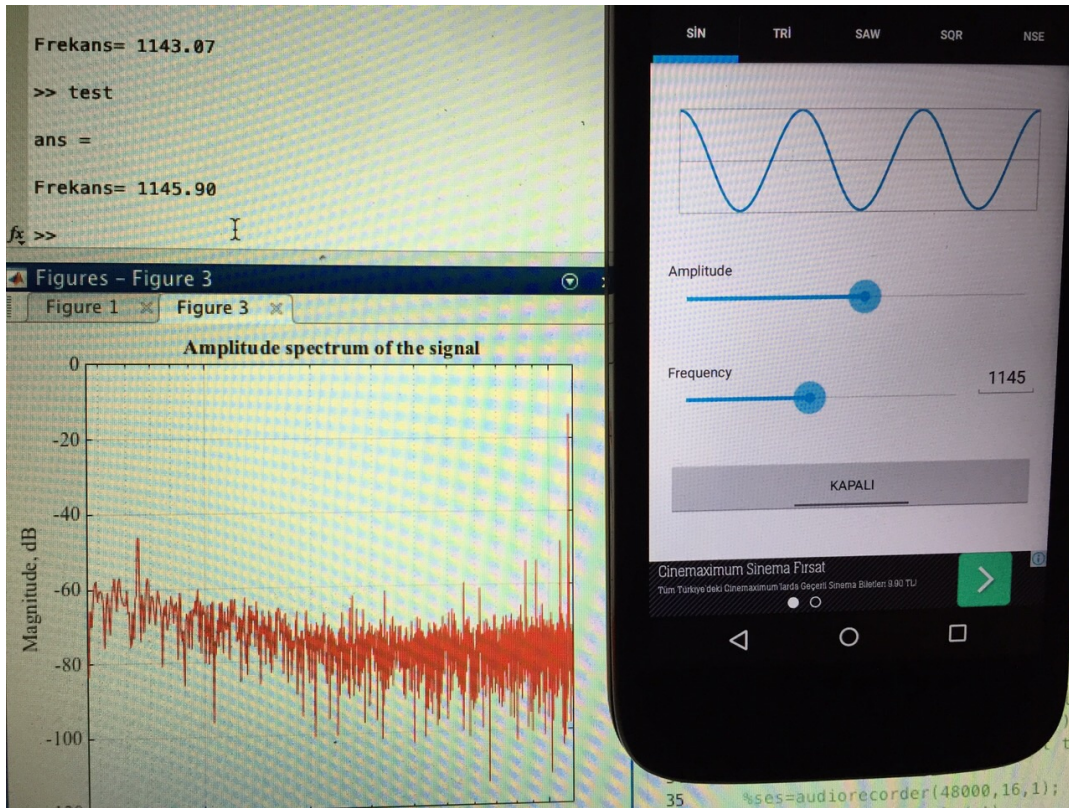
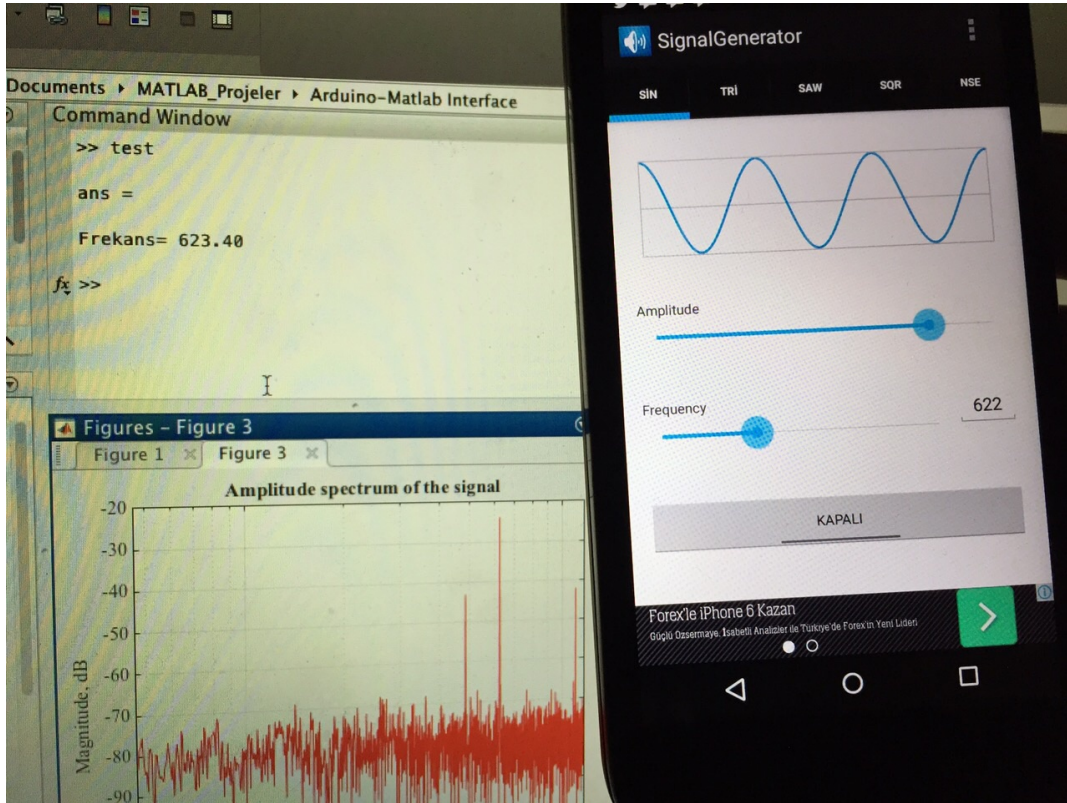
Daha sonrasında Arduino'dan okuduğum verileri bu kod için uygun hale getirip, verileri bu kod ile analiz ettirdim. Mikrofon devrem en iyi performansta çalışmadığı için çok fazla gürültü aldım ve özellikle düşük frekansların olması gerektiğinden daha yüksek olduğunu farkettim. Ben de kod üzerinde oynayarak, yaklaşık 100 Hz öncesini analiz edilecek frekans aralığından çıkarttım. Bu son değişikliği de yaparak daha net değerler elde ettim.

Matlab'da ses analiz kodunun çizdirdiği frekans grafiğinin maksimum noktasını hesaplayarak telefonda verdiğim frekansı ± 2 Hz hata payıyla Matlab'dan okuyabildim.

Yaptığım denemelere göre Sampling Rate 2360 Hz olmasına rağmen 1150 Hz den yukarısı için programın doğru değeri göstermediğini söyleyebilirim. 200 Hz den aşağısını bulurken de ses mikrofon tarafından yeterince yüksek miktarda algılanmadığı için hata oranı artabiliyor. Zaten 100 Hz den aşağısını kod üzerinden engellediğim için okumamız mümkün değil. Yani 200 Hz ile 1150 Hz arasında değerleri bu sistem ile görebiliyoruz. Bu aralığın yükseltilmesi için daha iyi bir mikrofon düzeneği kullanılabilir ve Arduino yerine daha gelişmiş bir kart üzerinde çalışılabilir.

Aşağıda bazı frekanslarda yaptığım denemeleri koyuyorum. Ekte daha fazla frekans mevcuttur.





Projede kullanılan malzemeler:

- 1 adet Arduino Uno
- 1 adet breadboard
- Matlab yüklü bir bilgisayar
- 1 adet electret mikrofon
- 1 adet direnç (1k-20k)
- 1 adet kapasitör (0,1uF – 1uF)

Arduino ve Matlab için yazılan kodların kullanılması:

- 1- FrekanBulucu.ino Arduino'ya yazılır.
- 2- Matlab açılır ve kodların bulunduğu klasör açılır.
- 3- 'ac' kodu ile Arduino serial port bağlantısı yapılır.
- 4- 'test' kodu ile yazılır, enter a basıldığı anda kayıt başlar. Frekansın daha önceden açılmış olması önerilir.
- 5- 'kapat' kodu ile Arduino serial portu kapatılır ve programdan çıkılır.