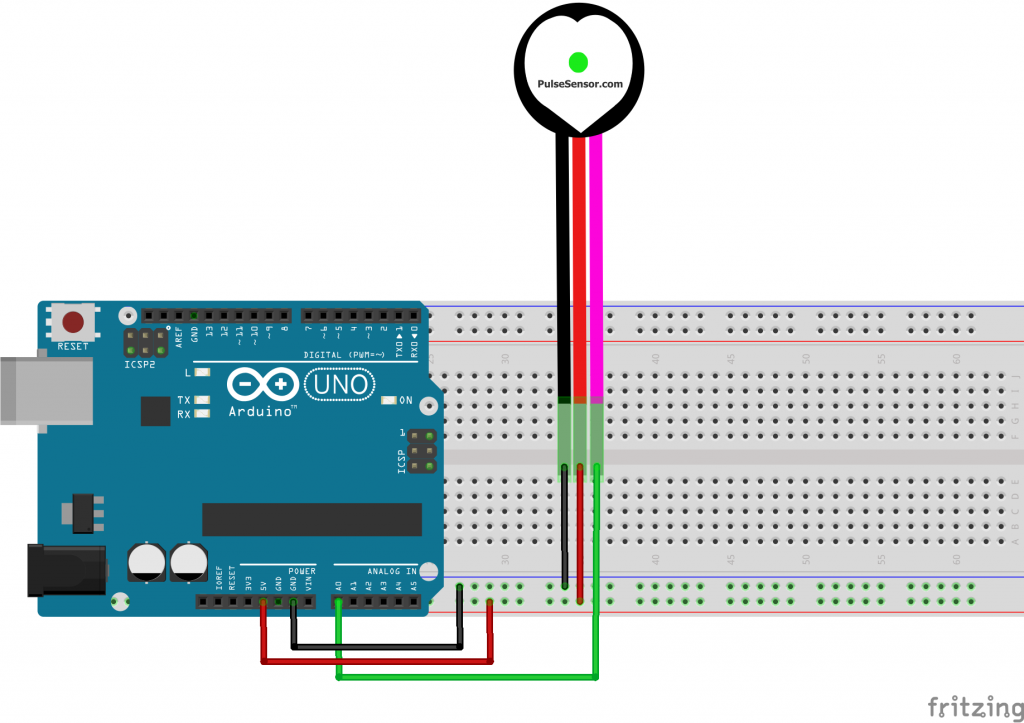
**Pulse Nabız Ölçer Nedir ve Nasıl Çalışır?**

Öncelikle, projemizin en önemli malzemelerinden biri olan Pulse Nabız Ölçer’i tanıyarak başlayalım. Pulse Nabız Ölçer, parmak ucunuza veya kulağınıza sabitleyerek kolayca nabzınızı ölçmeye yarayan bir sensördür. Bu sensörü 3V veya 5V ile beslemek mümkündür. Üzerinde bulunan gürültü engelleme özellikli devre sayesinde temiz ve stabil bir ölçüm alabilirsiniz.

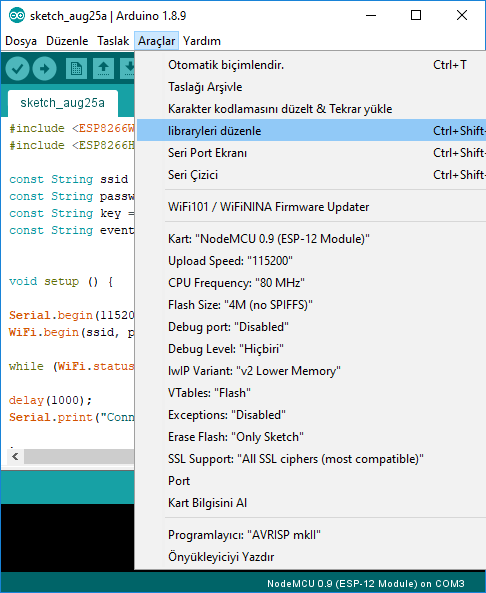
Pulse Nabız Ölçer’in çalışma prensibi şu şekildedir: Sensör parmak ucunuza veya kulağınıza gönderdiği ışığın ne kadarının yansıdığı bilgisini ölçerek sinyal pini üzerinden 0 ile 1023 arasında analog bir değer verir. Bu değer nabız atışı sırasında yükselir ve daha sonra tekrar düşer. Arduino’ya yazacağımız kod bu değişimleri kullanarak dakikadaki nabız sayımızı ölçer.

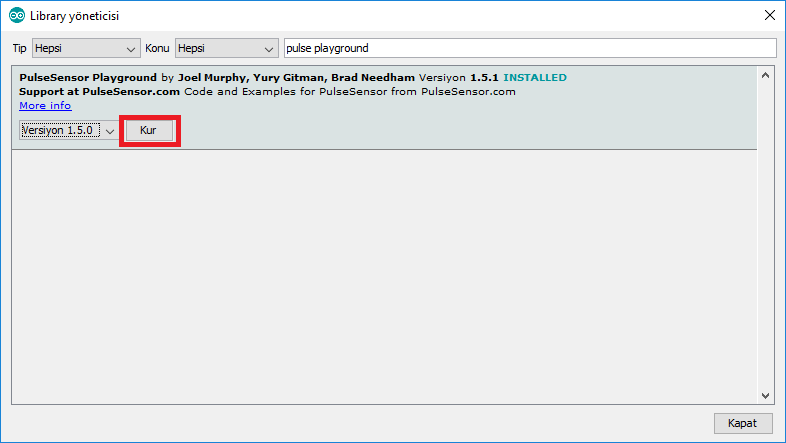
**Nabız Ölçer Projesi**

Projemiz iki kısımdan oluşacak. İlk kısımda, Pulse sensörü ile Arduino arasındaki bağlantıyı gerçekleştireceğiz.



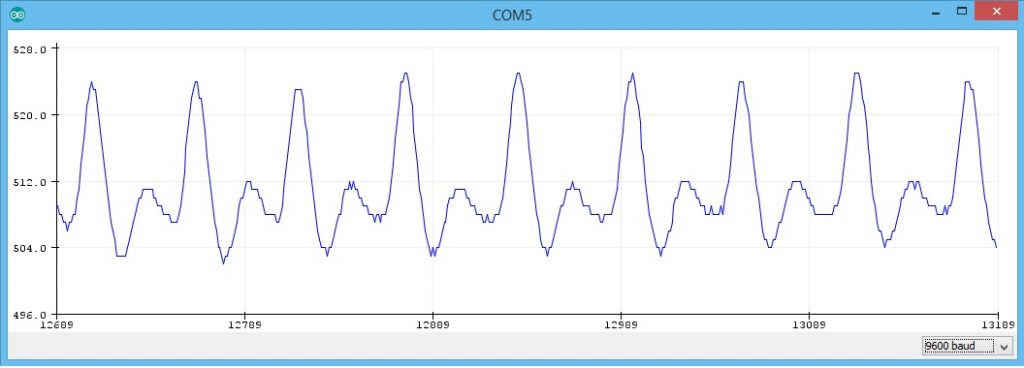
Sıra Arduino için olan kodu yazmaya geldi. Arduino IDE’yi açtıktan sonra, sensörümüzü kullanabilmek için yapmamız gereken ilk şey gerekli kütüphaneyi bilgisayarımıza kurmak. Bunun için, yukarıdaki seçeneklerden “Araçlar” seçeneğine tıklayıp oradan “libraryleri düzenle” seçeneğine tıklamalıyız. Bu işlemin ardından “Library Yönetici” olarak adlandırılan sekme açılacaktır. Buradan gerekli kütüphaneyi kurmak için yapmamız gereken arama kısmına “pulse sensor playground” yazmak, “versiyon seç” kısmından en güncel versiyonu seçmek ve “Kur” butonuna basmak.





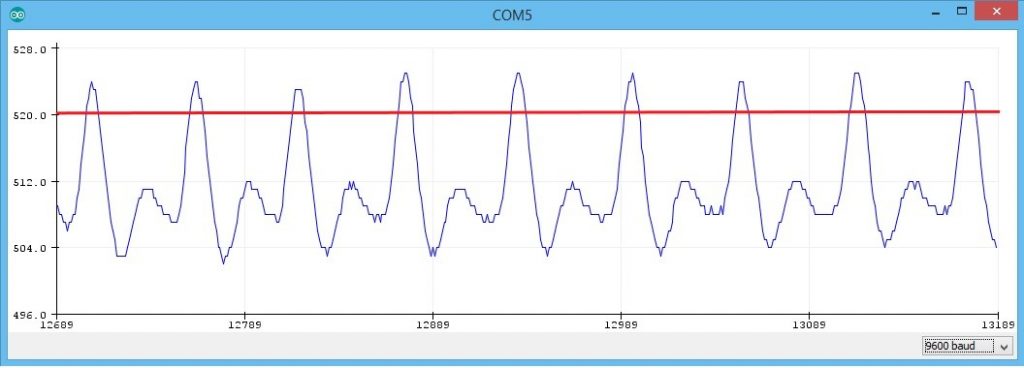
#### **Pulse Sensöründen Gelen Veriyi Okuma**

Gerekli kütüphaneyi başarıyla kurduğumuza göre sıradaki adıma geçebiliriz. Yukarıdaki kısımda da anlattığım üzere, Arduino nabız atışının gerçekleşip gerçekleşmediğini sensörden gelen analog değerin büyüklüğüne göre karar veriyor. Bu nedenle bizim önceden bir threshold, yani eşik değeri, belirlememiz gerekiyor. Bu sayede Arduino belirlediğimiz eşik değerinin üzerinde bir sinyal geldiğinde nabız atışının gerçekleştiğini anlayabilecek. Eşik değerini belirleyebilmek için sensörden gelen değerleri görmemiz gerekiyor. Bunu yapmanın en kolay yolu indirmiş olduğumuz kütüphanede yer alan örnek kodlardan birini kullanmak. Gerekli örnek kod dosyasını açabilmek için sırasıyla Dosya>Örnekler>Pulse Sensor Playground>GettingStartedProject seçeneklerine tıklıyoruz. Daha sonra, ekranda açılmış olan kodu Arduino’muza yüklüyoruz. Şimdi gelen sinyalleri bilgisayarımızda görüntüleyebilmek için Araçlar>Seri Çizici seçeneklerine tıklıyoruz. Ekranda, sensörden gelen veriler ışığında ilerleyen bir çizgi oluştuğunu göreceksiniz. Şimdi yapmanız gereken sensöre parmak ucunuzu yerleştirmek. Şimdi ise ekranda aşağıdaki görseldeki gibi dalgalar oluştuğunu görmeniz gerekiyor. Eğer durum buysa herşey yolunda gidiyor demektir ve yolumuza devam edebiliriz. Ancak aşağıdakine benzer bir sinyal akışı elde edemediyseniz bu yazının en aşağısında bulunan “Sorun Giderme” kısmını inceleyebilirsiniz.

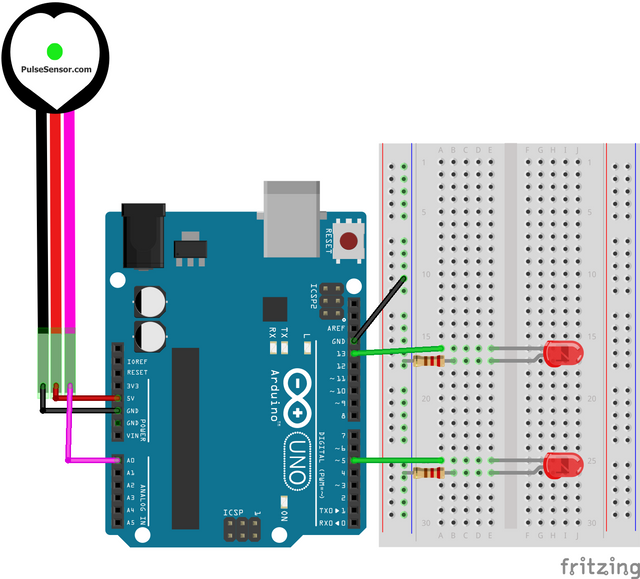


#### **Eşik Değerini Belirleme**

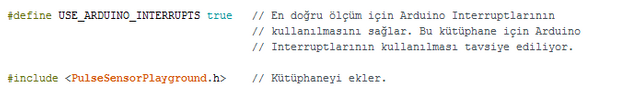
Sıradaki adım, ekranınızdaki grafiğe göre eşik değerini belirlemek. Bunun için öyle bir değer belirlemeliyiz ki grafikteki yükselişler bu değerin üzerine çıkmalı ama yükselişler dışındaki kısımlar da bu değerin altında kalmalı. Örneğin aşağıdaki görselde 520 uygun bir eşik değeri. Ama siz kendi değerinizi ekranınızdaki grafiğe göre belirlemelisiniz çünkü farklılık gösterebilir.



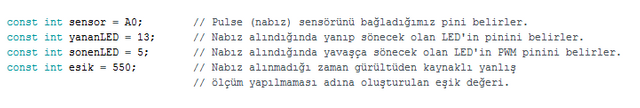
#### **Kod ve Son Devre Şeması**



Kütüphanemizi sketch’imize eklemeden önce, Arduino Interruptlarının kullanılmasını sağlayan komutu ekleyin. Bu komutu eklememizin nedeni, bu kütüphaneyi kullanırken en doğru nabız ölçümünü almak için Arduino interruptlarının kullanılması gerektiğindendir. Ardından kütüphaneyi eklemek için [#include](https://steemit.com/trending/include) <PulseSensorPlayground.h> yazın.



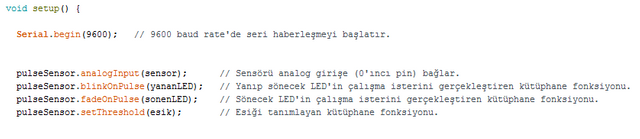
**3.** Kütüphaneyi ekledikten sonra Arduino’da kullanacağınız pinleri belirleyen kodu yazın. İki LED ve bir sensör kullanacağımızdan, bunlar için 3 değişken tanımlayın. Sensörü analog herhangi bir pine bağlayın, yanıp sönecek olan LED’i PWM çıkışı olmayan herhangi bir pine bağlayın, nabızla sönecek olan LED’i ise PWM çıkışı olan bir pine bağlayın. Ardından nabız alınmadığı zaman gürültüden kaynaklı yanlış ölçüm yapılmaması adına bir eşik değeri oluşturun.



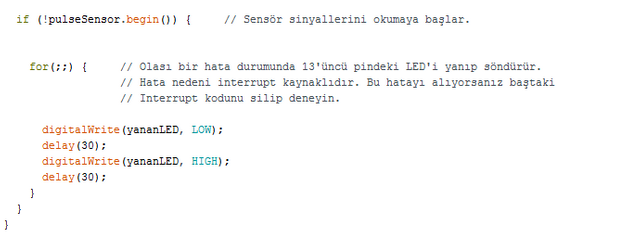
**4.** Kullanacağımız kütüphanenin fonksiyonlarını tanımlayan kodu yazın.



**5.** void setup() fonksiyonunu içine, 9600 baud rate hızında seri haberleşmeyi başlatacak komutu ekleyin. Sonrasında sırayla, nabız sensörünü Arduino’nun analog girişine atayacak olan fonksiyonu, yanıp sönecek LED'in çalışma isterini gerçekleştiren kütüphane fonksiyonunu, nabız alındığında sönecek olan LED'in çalışma isterini gerçekleştiren kütüphane fonksiyonunu ve eşik değerini tanımlayan kütüphane fonksiyonunu ekleyin.



**6.** Bu fonksiyonların devamına, sensör sinyalleri okunduğu zaman çalışacak bir if koşulu ekleyin. If koşulunun içine olası bir sensör okuma hatası durumunda (13.pindeki) yanıp sönen LED’i yakıp söndürecek komutu ekleyin. Eğer çalıştırma sırasında bu hatayı alıyorsanız, Arduino’nuzun interruptlarında sorun var demektir. En baştaki interrupt kodunu silip tekrar deneyin. Alacağınız ölçüm daha az güvenilir olacaktır.



**7.** void loop() fonksiyonunun içine bir integer değişkeni tanımlayın. Bu değişkeni sensörden okunan nabız verisini integer değerine çevirecek kütüphane fonksiyonuna eşitleyin. Ardından sensörden gelen nabız verisi varsa çalışacak bir if koşulu oluşturun. If koşulunun içine ölçülen nabzı ekrana yansıtacak kodu ekleyin. Koşuldan sonra okuma kolaylığı açısından bir delay komutu ekleyin.

