Servo motoru kontrol için kütüphane ekliyoruz

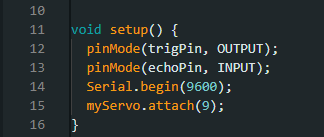
Trig yani ses göndermek için gerekli kanalı 4. pine

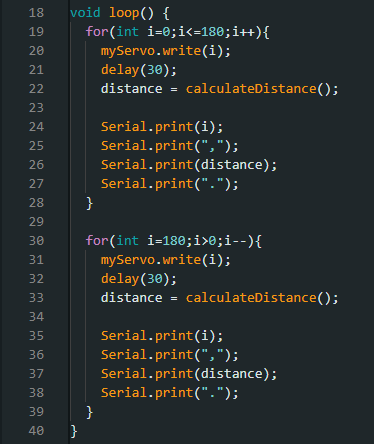
Echo yani sesi alacağımız kanalı 3. pine

Duration değişkeni bekleme süresini

Distance hedefin uzaklığını

myServo servo motorumuz temsil edecek şekilde atamalarını yaptık.

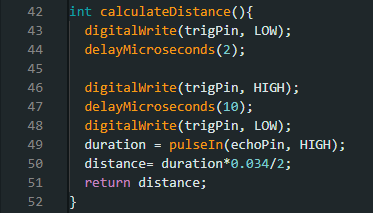
Setup metodu tanımlamalarımız yani kurulumumuzu tanımlıyor.

Trig çıkış, Echo giriş olarak atıyoruz

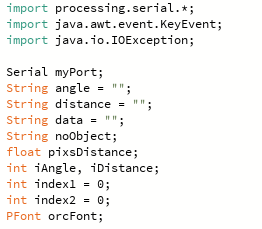
Seri haberleşmeyi başlatır

Servo motoru bağlar

Loop kısmı sürekli olarak çalışacak olan kod kısmıdır. 0 ile 180 dereceleri arasında dönmesini sağlar. İlk olarak 0’ dan 180’ e sonrada 180’ den 0’ a döndürme işlemini yapar bunu şu şekilde yapıyor; servo motoru i kaç ise o dereceye götürür 30 ms bekler ve servo motor o dereceye gelince mesafeyi hesaplamak için calculateDistance methodu çağrılır. Oradan gelen değerler PC’ye yazdırılır. Yazdırma işlemi şu şekildedir. Derece, Mesafe.



Bu metod tüm işlemler tamamlanıp hatasız sonuç almak için sensörü kapatır ve 2 ms bekler sonra sesörü açar ve 10 ms bekler o sırada sensör sinyal göndermiş olur. Sonra sensörü kapatır ve gelen değeri okur. Bu değer gidiş ve dönüş süresidir. Bu değer ile mesafe hesaplamak için değeri 2 böleriz ve liseden bildiğimiz x = v \* t formülü ile hesap yapılır x = gidiş süresi \* ms cinsinden ses hızının değeri çarpımı ile mesafe bulunup çağırıldığı yere gönderir.



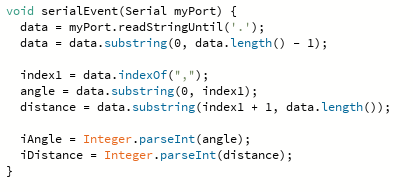
* **İmport java.awt.event.KeyEvent =** Veri almak için kullanılan bir kütüphane
* **Serial myPort** = Seri bağlantı nesnesini tanımlar.
* **String angle** **= ""** = Açıyı tutmak için bir string.
* **String distance** **= "" =** Mesafeyi tutmak için bir string.
* **String data = "" =** Seri porttan gelen veriyi tutan genel string.
* **String noObject** = Hedef olup olmadığını belirten bir string.
* **float pixsDistance** = Grafikte çizilecek hedefin uzaklığı.
* **int iAngle, iDistance** = Açı ve mesafe değerlerini saklamak için tamsayılar.
* **int index1 = 0** = Verideki ilk virgülün konumu.
* **int index2 = 0** = Verideki ikinci virgülün konumu.
* **PFont orcFont** = Kullanılacak yazı tipini tanımlar.



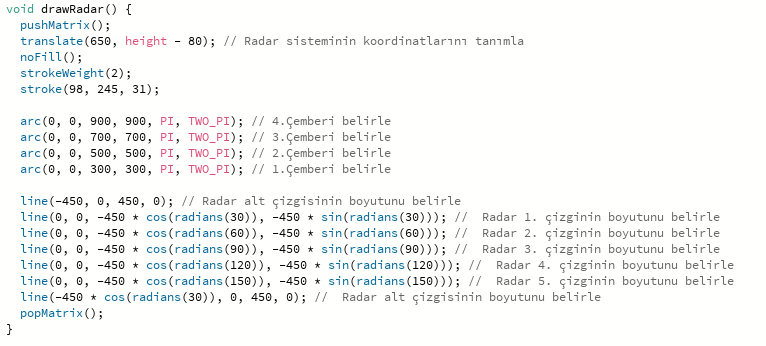
* Pencere boyutunu ve diğer başlangıç ayarlarını yapar.
* Smooth() daha yumuşak geçişler için kullanılır.
* Seri bağlantısını başlatır.
* MyPort.bufferUntil(‘.’) gelen veriyi ‘.’ Ya kadar almasını sağlar.
* Font dosyasını yükler.



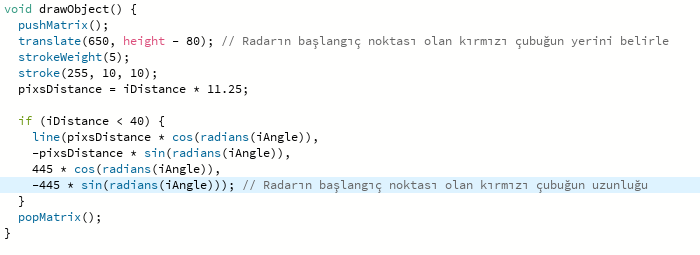
* **Fill =** Kullanılacak rengi belirler
* **TextFont =** Kullanılacak fontu belirler
* **NoStroke =** Çizilen şekillerin kenar çizgilerini kaldırır.
* **Rect =** Belirtilen koordinatlarda belirtilen en ve boyda bir bölge ayarlar



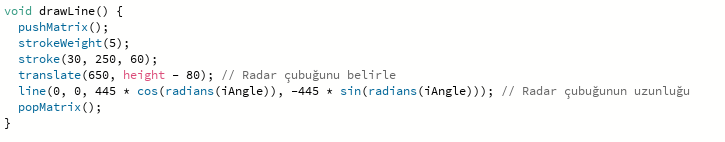
* Porttan herhangi bir giriş olduğu zaman çalışır ve mesafe ile açı hesaplamasını yapar.



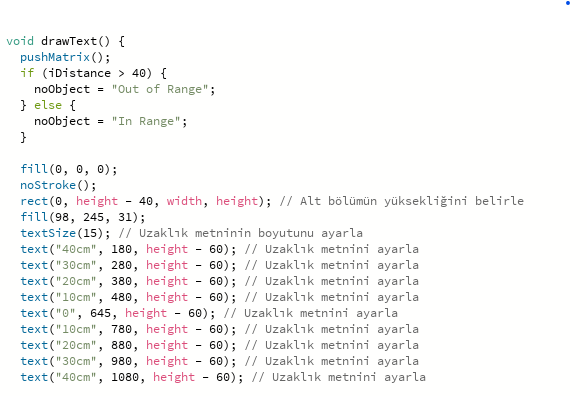
* PushMatrix ve popMatrix burada bir stack içine atılıp hareket etmesini sağlar. Belli bir süre hafızada tutup sonrada bırakma işlemi ile öteleme, döndüreme ve ölçekleme gibi işlemlerde kullanılır.
* Translate radar sistemin nereden başlayacağını belirler.
* NoFill() çizilen şekillin içinin boş olmasını sağlar.
* StrokeWeight(2) sınır çizgisinin kalınlığını belirler.
* Stroke kenar çizgisinin rengini belirler.
* Arc fonksiyonu ilk iki değeri koordinatlarında sonraki iki derece değerinde bir yarıçapa sahip yarım daire çizilmesine olanak tanır.
* Altındaki line fonksiyonu ise belirtilen uzunlukta ve açıda çizgiler çizilmesini sağlar.



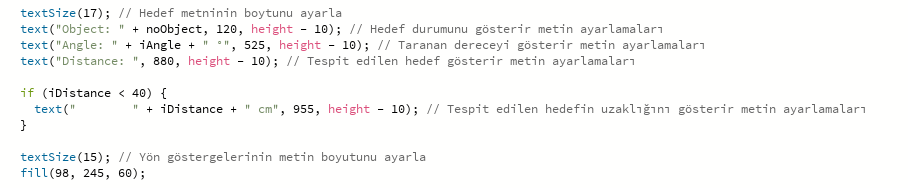
* Alınan uzaklık değeri ile hesaplama yaparak eğer 40 değerinden küçükse o değer kadar bir alanı kırmızıya boyamasına olanak tanır.



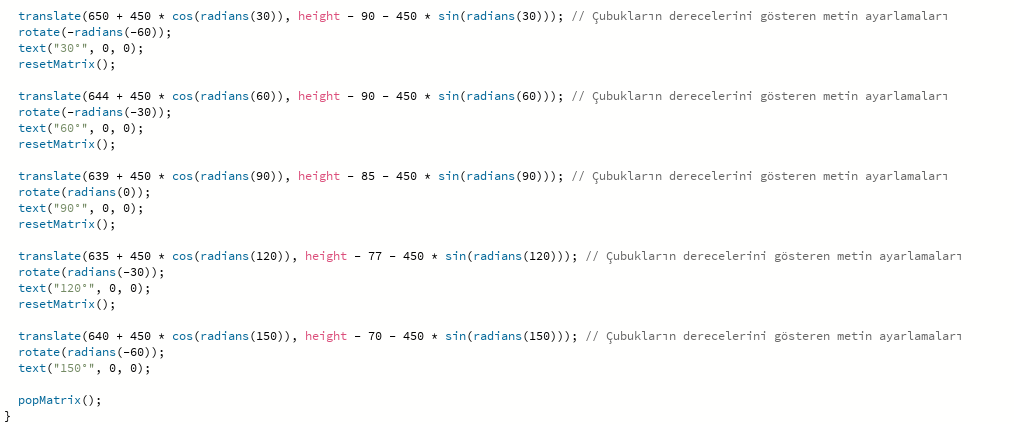
* Radar çubuğunun uzunluğunu ve koordinatını belirler



* Eğer hedef görünmüş ise In Range, hedef yok ise radarda o zaman Out of Range olarak tanımlama yapar.
* Koordinatlarını ve yazılacak metni girerek text metodu kullanılabilir. Burada alt bölmede hedefin uzaklığını belirlemeye yarayan kılavuz mesafeler yazıyor.



* En alt bölmedeki hedefin menzil içinde mi?, radar hangi açıda dönüyor ve hedef var ise uzaklığı ne kadar olduğu gibi bilgilerin tanımlamaları yazıyor. Metin boyutu, ve konumu belirleniyor.



* Son olarak bu kod parçası da derecelerin daha anlaşılır olması için kılavuz yazılar ekleyerek 30, 60, 90, 120 ve 150 gibi açıların radar sisteminin en dış katmanına yerleşmesini sağlıyor.

**Dinlediğiniz İçin Teşekkürler**