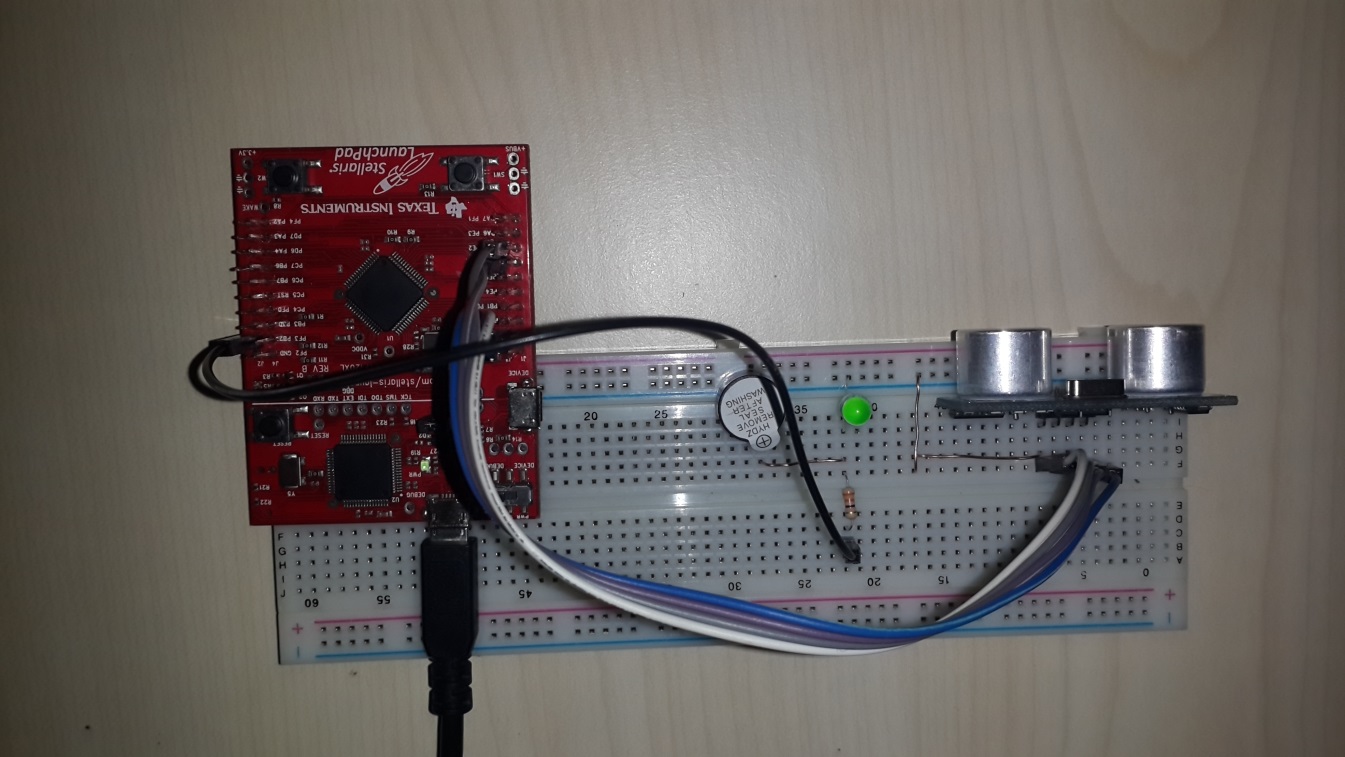
**Proje Konusu :**



**Grup Üyeleri Ad Soyad Numara**

**CİHAN SAVAS - 130202098**

**CİHAT DUYKU - 150202106**

**MEHMET SAMET YILDIZ - 150202104  
ADEM DERVİŞ YILMAZ - 150202121**

**Sunum Tarihi :**

**26/05/2016**

## 1 GİRİŞ

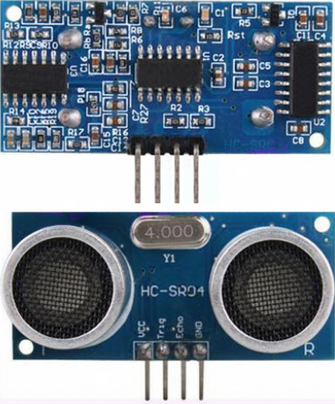
***HC-SR04***

*Hc-sr04 ultrasonik sensör sonar(Sound Navigation and Ranging ) kullanarak karşısındaki nesneye olan mesafesini hesaplayan bir input kaynağıdır.Sonar dediğimiz sistem ses dalgalarını kullanarak cismin uzaklığını boyutunu elde etmemizi sağlar. Bu tür sensörlerin esin kaynağı yunuslar ve yarasalardır. Onlarda sonar ile iletişim kurar ve hareket eder.*

*2cm ile 400cm  arası mesafe en sağlıklı okuma yaptığı aralıktır.Üzerinde bir alıcı ve bir verici modül bulunur.Çalışma mantığını Hc-rs04 Ultrasonik sensör nasıl çalışır kısmında ayrıntılı anlatacağım.*

***Özellikler***

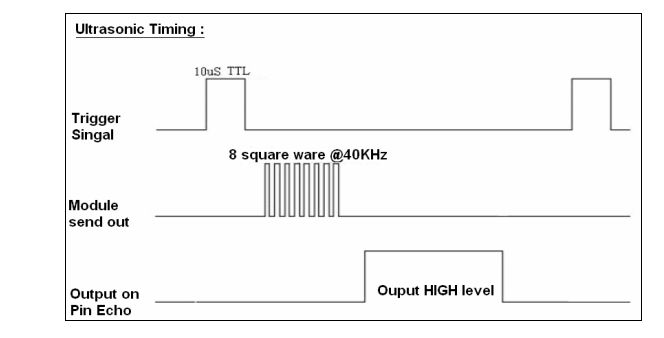
* *Güç Kaynağı \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_: +5V DC*
* *Çalışma frekansı \_\_\_\_\_\_\_\_\_\_\_\_: 40 kHZ*
* *Efektif Açı   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_: <15 derece*
* *Mesafe ölçüm arası  \_\_\_\_\_\_\_\_\_\_: 2cm – 400cm*
* *Tetikleme girişi darbe genişliği \_\_\_: 10uS*

*[](http://sinancanbayrak.com/wp-content/uploads/2015/08/Hc-sr04-ultrasonik-sensor-layout.png) *

***Buzzer’ın Layoutu***

***Hc-sr04 ultrasonik sensor layout***

* *VCC = +5V  DC*
* *Trig = Sensörün tetikleme kısmı*
* *Echo = Sensörün alıcı kısmı*
* *GND = Topraklama kısmı*

*[](http://sinancanbayrak.com/wp-content/uploads/2015/08/Ultrasonik-zamanlama.png)*

* ***Trigger signal\_\_\_\_\_\_:*** *Trig pini aktif olduğunda gönderilen 1 adet sinyaldir.*
* ***Module send out\_\_\_\_:*** *Trig pini aktif olduğunda tek seferde 40kHz de  8 adet kare dalga sinyal gönderir.*
* ***Output on pin Echo\_\_:*** *Yansıyan dalgaların dönüşte Echo pininde oluşturduğu sinyaldir.*

*Ölçümün başlaması için Trig pini en az 10 uS de +5V almalıdır. Böylelikle sensörden  40 kHZ’de 8 devir ses dalgası dışarıya iletilir ve yansıması beklenir. Hc-sr04 alıcıdan ses dalgasını aldığında Echo pini 0V dan 5V a geçer ve mesafeyle orantılı bir süre bekler. Biz burda Echo pinindeki genişliği ölçerek , aradaki mesafeyi elde edebiliriz.*

*Zaman = Echo pin genişliği , uS (mikro saniye)*

* *Cm cinsinden mesafe = Zaman / 58*
* *İnç cinsinden mesafe = Zaman / 148*

## 2 Problem Çözümü ve Uygulama

### 2.1 Probleme Çözüm Yaklaşımı

*Projeye 1 adet daha timer eklenmiştir bu timer buzzer’ın tam döngüsü için ne kadar sürmesi gerektiğini ölçmektedir. Projede ilk olarak timersiz meşgul bekleme yaparak yapılmaya çalışmıştır fakat buzzer için süre hesaplanırken aynı anda 2 işlem yapılmadığı için hc sr 04 askıya alınarak çalışmaz hale gelmekteydi. Bu sorunu üstesinden böyle gelinmiştir.*

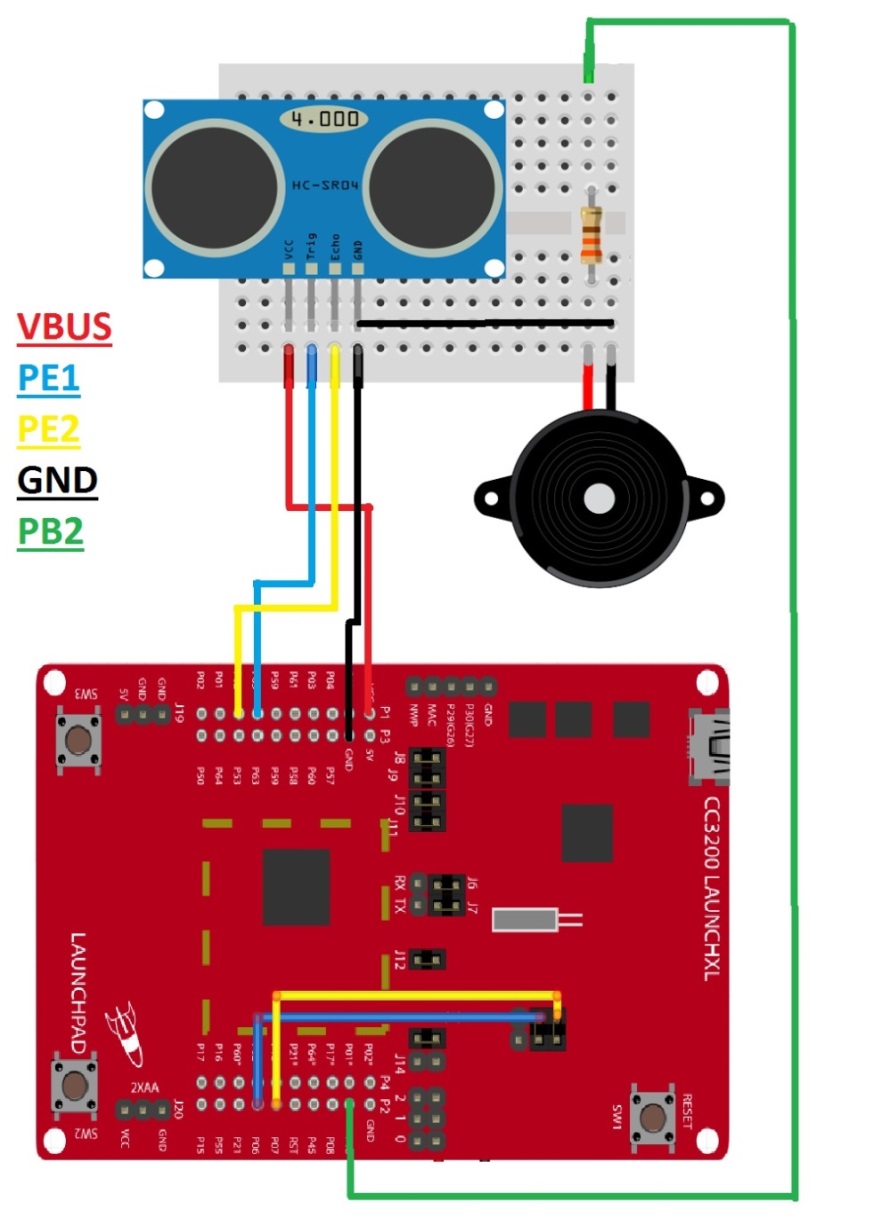
*Bir diğer sorun ise her sensör çalıştığında ki bu saniyede 6 olarak ayarlanmıştı buzzer için timer saniyede 6 kere kurulunca sadece kurulduğu ile kalıyordu ve hiç ötmüyordu. Veya hep ötülü şekilde kalıyordu. Bizde cm değerinin bi önceki değerini say değişkeninde tuttuk ve 1 fazla veya 1 az durumunda timer kuruluyor ve buzzer kesmeleri oluşuyordu. Yani özetlemek gerekirse her cm değeri ölçüldüğünde timer kurmak gerekmiyor sadece uzaklıkta değişiklik yaşandığı zaman timer kuruluyor.*

### 2.2 Donanım Mimarisi

**Projede Kullanılan Malzemeler:**

* HC-SR04
* Stellaris Launchpad
* 330 ohm direnç
* Buzzer
* LED

**Hc-sr04 Ultrasonik Mesafe Sensörü Stellaris Devre Şeması**



### 

### 2.3 Yazılım Mimarisi

Get\_mm()fonksiyonundan gelen mm değeri cm ‘e çevriliyor. Sonra bu değer seri portta yazdırılıyor.

Eger cm değeri 99 dan büyükse timer durdurulup buzzer hep kapalı konumuna getiriliyor. Burda timerın kapatılma sebebi buzzer yanıp sönmesini engellemek. Kırmızı led açık konumuna getirilyor. Hata göstergesi olarak kullanılmıştır.

Eger cm değeri 100 ile 5 arasında ölçülürse kırmızı led kapatılıyor. Çünkü istenilen aralık budur. Burada kilit bir nokta var eğer bir önceki değerden fark olduğu anlaşılırsa timer ayarlanıyor. Buzzerın uzaklığa göre yanıp sönmesi sağlanıyor.

Eger cm değeri 5 veya 5 den küçükse timer kapatılıp hata ledi etkinleştiriliyor aynı zamanda buzzer açık duruma getiriliyor.

SysCtlDelay fonksiyonu ile saniyede 6 ölçüm yapacak şekilde ayarlanmıştır.

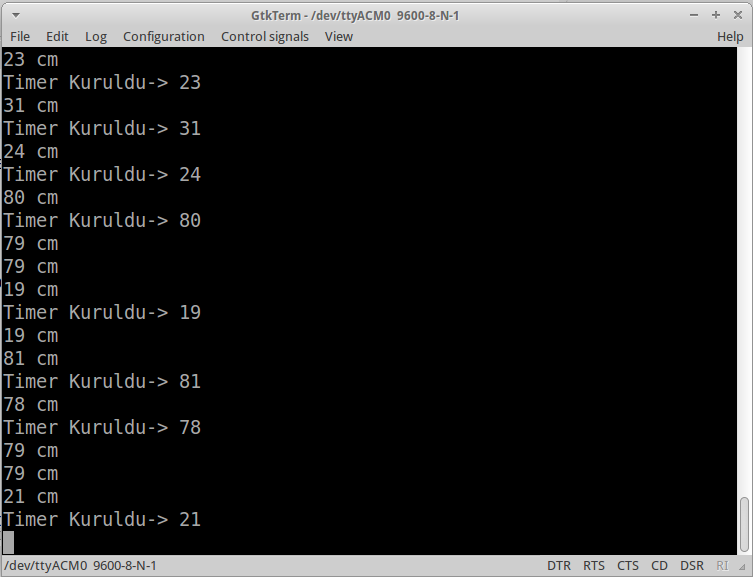
Timer ayarlandıktan sonra kesmeler yapılmaya başlanıyor. Her kesme sırasında buzzerın durumu kontrol ediliyor ve ters konuma çevriliyor. Bu kesmeler void Timer3IntHandler(void) fonksiyonunda yapılmaktadır.

### 2.4 Sistemin Çalıştırılması

Sistemde herhangi bir buton kullanımı yoktur. Usb bir kaynağa bağlandığında çalışmaktadır kullanıcı dostu bir proje olmasına özen gösterilmiştir ledler ile uyarı ve bildirim sağlanmıştır.

* Eğer RGB ledlerinden kırmızı led yanıyorsa 1 metre üstünde ve 5 cm nin altında bir uzaklık ölçülmüştür.
* Eger yeşil led yanıp söndü ise timer ayarlandığı anlamına gelmektedir bip sesindeki frekans tekrar kurulmuştur. Veya uzaklıkta bir değişim olduğu anlamı da çıkarılabilir.
* Buzzer eger hiç tepki vermiyorsa istenilen uzaklık içinde değildir.

## Sonuçlar



* Geliştirilen mimaride 0-100 cm aralığında ölçümler alınmakta ve 100 cm’den 5 cm’e yaklaştıkça uyarı sıklıkları artmakta 5 cm ile 0 cm aralığında ise kesintisiz uyarı vermektedir.
* Resimde de görülceği üzere cm de bir değişiklik olduğu sürece timer kurulmaktadır.
* Mesela resmin orta satırına baktığımızda 79 cm iken birden 19cm ye düşüyor bu da program tarafından algılanarak ilgili satıra yönlendiriyor.

## 4 Kaynaklar

* <https://engineering.purdue.edu/ece477/Archive/2014/Spring/S14-Grp1/docs/software/LM4F-LaunchPad-04%20-%20Interrupts%20&%20Timers.pdf>
* <http://www.engr.usask.ca/classes/EE/392/Lab­Guides/11­ARM/lm4f120h5qr.pdf>
* <http://users.ece.utexas.edu/~valvano/arm/>