



**ULTRASONİK MESAFE ÖLÇÜMÜ VE DOĞRULANMASI**

**Muhammed ÇALIŞ**

**YÜKSEK LİSANS TEZİ**

**ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**HAZİRAN 2019**

Muhammed ÇALIŞ tarafından hazırlanan “ULTRASONİK MESAFE ÖLÇÜMÜ VE DOĞRULANMASI” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Elektrik Elektronik Mühendisliği Ana Bilim Dalında YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

**Danışman:** Doç. Dr. Ali SAYIN

Elektrik Elektronik Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ....

**Başkan:** Doç. Dr. İlhan KOŞALAY

Elektrik Elektronik Mühendisliği Ana Bilim Dalı, Ankara Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ....

**Üye:** Doç. Dr. Cemal YILMAZ

Elektrik Elektronik Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Yüksek Lisans Tezi olduğunu onaylıyorum. ....

Tez Savunma Tarihi: 21/06/2019

Jüri tarafından kabul edilen bu tezin Yüksek Lisans Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....  
Prof. Dr. Sena YAŞYERLİ  
Fen Bilimleri Enstitüsü Müdürü

## ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
  - Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
  - Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
  - Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
  - Bu tezde sunduğum çalışmanın özgün olduğunu,
- bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Muhammed ÇALIŞ

21/06/2019



ULTRASONİK MESAFE ÖLÇÜMÜ VE DOĞRULANMASI  
(Yüksek Lisans Tezi)

Muhammed ÇALIŞ

GAZİ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2019

ÖZET

Bu çalışmada ultrasonik sensörlerden alınan veriler doğrultusunda bir hedefe yönelmiş hızlı nesnelerin hızlarının, yörüngelerinin ve hedefe ne kadarlık mesafeden geçtiklerinin tespit edilmesi hedeflenmiştir. Ultrasonik sensörlerin çalışma mantığı, avantajları, dezavantajları ve kullanım sebepleri detaylandırılmıştır. Ultrasonik sensörlerden toplanan veriler, işlemciye iletdikten sonra filtrelenmiş, uygun matematiksel denklemler ile ihtiyaç duyulan veri formatına dönüştürülmüştür. İşlemcide analog veriler filtrelenip daha sonra dijital sinyallere dönüştürülmüştür. Bu dijital veriler kullanılarak nesnenin yörünge, mesafe ve hız bilgilerinin belirlenmesi sağlanmıştır. Ethernet protokolü üzerinden kontrol kartına erişim sağlanıp, veriler bilgisayara aktırılmıştır. Bilgisayara transfer edilen veriler matematiksel algoritmalar ile gerekli hesaplamalar yapılarak polar diyagram üzerinde gösterilebilecek formata dönüştürülmüştür. Bu işlemler sonucunda kullanıcı ara yüzünde yörünge tespiti ve hedefe olan uzaklık gösterimi yapılmıştır. Ayrıca yine aynı kullanıcı ara yüzü vasıtası ile hız bilgisi operatöre sunulduğu özel bir mimari kullanılmıştır. Daha önce radyo frekansında (RF) alıcı verici sistemler, optik sensörler gibi farklı sensör gruplarıyla benzer çalışmalar yapılmıştır fakat tespit edilecek cismin fiziksel özelliklerinden, dış ortam etkisinden etkilenmeyen, çalışma esnasında ortamda elektromanyetik gürültü oluşturmayan bir sistem geliştirmesi bu çalışmada sağlanmıştır. Yurt dışında yapılan çalışmalar ve üretilen cihazlar ise ihraç izinli olmaları sebebiyle ülkemizde temin edilememektedir. Hızlı nesnelerin yön ve hız testinde kritik kullanım alanı olan ultrasonik sensörlerin ülkemizdeki çalışmalarda kullanılabilmesinin imkanı sağlanmıştır.

Bilim Kodu : 90524

Anahtar Kelimeler : İşaret işleme, ultrasonik sensör, mesafe ölçümü, ses sinyalleri, yörünge tespiti, hız tespiti

Sayfa Adedi : 50

Danışman : Doç. Dr. Ali SAYGIN

# ULTRASONIC DISTANCE MEASUREMENT AND VERIFICATION

(M. Sc. Thesis)

Muhammed ÇALIŞ

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2019

## ABSTRACT

In this study, it is aimed to determine the velocities, trajectories and distance of high-speed objects oriented towards a target in accordance with the data obtained from ultrasonic sensors. The operating logic of ultrasonic sensors, advantages and disadvantages are detailed reasons for use in this type of work. After the data collected by the sensor is transmitted to the processor, it is filtered and converted into the required data format with appropriate mathematical equations. By using this data, the trajectory, distance and speed information of the object is determined. The Ethernet protocol provides access to the control card and the data is transmitted to the computer via the processor on the control board. The processor processes the analog data and then converts it into digital signals. The data transferred to the computer is converted into a format which can be displayed on the polar diagram by making necessary calculations with mathematical algorithms. As a result of these processes, trajectory detection and distance to the target are shown in the user interface. Also with the same user interface, the speed information is presented to the operator. Previous studies have been carried out with different sensor groups such as radio frequency (RF) transceiver systems and optical sensors. However, this study aims to develop a system with an improved user interface, which is not affected by the physical properties of the object to be detected, external environment effects, and no electromagnetic noise in the environment during operation. There is no academic or commercial study in our country that handles this subject and not possible to supply the products in other countries to our country.

Science Code : 90524

Key Words : Signal processing, ultrasonic sensor, distance measurement, voice signals, trajectory, trajectory detection, velocity calculation

Page Number : 50

Supervisor : Assist. Prof. Dr. Ali SAYGIN

## TEŞEKKÜR

Danışmalığımlı üstlenerek saat, gün, mekan ayrımı yapmaksızın bana yardımcı olan sayın Doç. Dr. Ali SAYGIN 'a teşekkürlerimi bildirmek isterim. Gerek kendisi, gerekse diğeri sayın öğretim üyelerimiz, çalışmama gereken her türlü desteğı sağlamışlardır. Eğitim öğretim süreci ve tez yazım aşamasında her türlü kolaylığı ve imkanı sağlayan çalıştığım kurum TÜBİTAK'ta mesai arkadaşlarım ve yöneticilerime de bu vesile ile teşekkürlerimi bildirmek isterim. Annem Emine ÇALIŞ ve Babam Harun Reşit ÇALIŞ ' a gösterdikleri sevgi, maddi manevi destek ve sabır dolayısıyla müteşekkirim. Son olarak özellikle proje çalışmalarında bana göstermiş olduğı sabır ve yardımları için sevgili eşim Çizgi ÇALIŞ 'a ayrıca teşekkür ediyorum.



## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET .....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER .....	vii
ÇİZELGELERİN LİSTESİ.....	viii
ŞEKİLLERİN LİSTESİ.....	ix
RESİMLERİN LİSTESİ.....	x
HARİTALARIN LİSTESİ.....	xi
SİMGELER VE KISALTMALAR.....	xii
1. GİRİŞ.....	1
2. ULTRASONİK SENSÖRLER .....	5
2.1. Ultrasonik Sensörlerin Çalışma Mantığı .....	5
2.2. LV-MaxSonar-EZ Sensörlerin Kullanımı .....	6
2.2.1. Genel kullanım şekli .....	7
2.2.2. Dizin şeklinde kullanımı .....	8
2.3. Ultrasonik Sensörlerin Veri Aktarımı ve Haberleşme Yöntemleri .....	11
2.3.1. Analog veri aktarımı .....	11
2.3.2. Seri haberleşme ile veri aktarımı.....	11
2.3.3. Darbe genişliği ile veri aktarımı.....	12
3. VERİLERİN TOPLANMASI VE İŞLENMESİ .....	13
3.1. Kontrol Kartı ile Verilerin Toplanması.....	14
3.2. İşlemci ile Verilerin İşlenmesi.....	14
3.2.1. ADC kullanımı .....	14

	<b>Sayfa</b>
3.2.2. Verilerin filtrelenmesi .....	15
3.2.3. Matematiksel algoritmalar ile anlamlandırılması.....	15
3.3. Verilerin Bilgisayar Ara Yüzüne Aktarılması .....	16
<b>4. KULLANICI ARAYÜZÜ .....</b>	<b>19</b>
4.1. Arayüzü Tasarımı .....	20
4.1.1. Windows Presentation Foundation (WPF) uygulaması .....	23
4.1.2. Windows From uygulaması .....	27
4.1.3. NI Measurement Studio kullanımı .....	27
4.2. Bilgisayara Aktarılan Verilerin VS ile Toplanıp İşlenmesi .....	28
4.2.1. Aç, mesafe, hız hesaplanması .....	28
<b>5. VERİ TABANI VE RAPOR OLUŞTURMA.....</b>	<b>37</b>
5.1. Toplanan Verilerin Veri Tabanında Saklanması .....	40
5.2. Otomatik Rapor Oluşturulması.....	40
<b>6. TARTIŞMA.....</b>	<b>43</b>
<b>7. SONUÇ VE ÖNERİLER.....</b>	<b>45</b>
<b>KAYNAKLAR .....</b>	<b>47</b>
<b>ÖZGEÇMİŞ .....</b>	<b>49</b>
<b>DİZİN .....</b>	<b>51</b>



## ÇİZELGELERİN LİSTESİ

<b>Çizelge</b>	<b>Sayfa</b>
Çizelge 3.1. Gömülü yazılım akış çizelgesi.....	21
Çizelge 4.1. Arayüz yazılımı kod akış çizelgesi.....	38





## ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil.1.1. Sistemin kurulum modeli.....	6
Şekil 2.1. Sensör pin yerleşimi .....	10
Şekil 2.2. Üç sensörün eş zamanlı çalıştırılma bağlantısı.....	14
Şekil 2.3. Sensör yerleşimi ve kapsama alanı .....	15
Şekil 2.4. Seri haberleşme veri sinyalleri.....	16
Şekil 2.5. Darbe genişliği yöntemi veri sinyalleri.....	16
Şekil 3.1. Analog veri ölçüm süresi .....	19
Şekil 4.1. Haberleşme akışı.....	32
Şekil 4.2. Tespit için gerekli olan trigonometrik denkleminin oluşturulmasında kullanılan üçgenlerin elde edilmesi. ....	33
Şekil 4.3. Gerekli üçgenin çıkarımı. ....	34
Şekil 4.4. Kullanılan trigonometrik yöntem .....	34
Şekil 4.5. Üçgen üzerinden gerekli olan açının elde edilmesi. ....	35
Şekil 4.6. İki ölçüm sonucunda oluşan üçgenler.....	36
Şekil 4.7. Alınan mesafenin hesaplanmasına imkan tanıyan üçgen .....	37

**RESİMLERİN LİSTESİ**

<b>Resim</b>	<b>Sayfa</b>
Resim 2.1. Sinyalin gönderilip yansıyanın geri alınması .....	9
Resim 2.2. Tekli kullanım sensör analog çıkışı .....	12
Resim 2.3. Üç sensörün aynı anda senkronizasyon sağlanmadan kullanımı.....	13
Resim 2.4. Üç sensörün aynı anda senkronizasyon sağlanarak kullanımı.....	13
Resim 2.5. Sensör kullanımı laboratuvar test ortamı.....	14
Resim 2.6. Analog ölçüm yöntemi.....	15
Resim 3.1. Kontrol kartı bağlantıları.....	18
Resim 4.1. Arayüz tasarımı görüntüsü.....	23
Resim 4.2. Araç kutusu ve arayüz tasarımı yapılan kısım.....	24
Resim 4.3. Arayüz yazılımı ve kodlama ekranı.....	25
Resim 4.4. Kullanılan araçların özelliklerinin ayarlanması.....	26
Resim 4.5. Hata ayıklama penceresi.....	27
Resim 4.6. Polar grafik.....	31
Resim 4.7. Kullanıcı arayüzü tasarımının görüntüsü ve ölçüm sonuçları.....	39
Resim 5.1. SQL server veri tabanı tablosu kayıt örnekleri.....	40

## SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

### Kısaltmalar

### Açıklamalar

<b>acos</b>	Kosinüs ifadesinin tersi
<b>ADC</b>	Analog dijital dönüştürücü
<b>cos</b>	Kosinüs
<b>UMÖD</b>	Ultrasonik mesafe ölçüm doğrulama
<b>Vcc</b>	Volt türünden besleme gerilimi
<b>WPF</b>	Windows presentation foundation

### Simgeler

### Açıklamalar

<b>T</b>	Açı ifadesi teta
<b>r</b>	Yarıçap
<b>cm</b>	Santimetre
<b>hz</b>	Hertz
<b>mS</b>	Milisaniye
<b>kHz</b>	Kilohertz
<b>s</b>	Saniye
<b>uS</b>	Mikrosaniye
<b>V</b>	Volt











## 1. GİRİŞ

Hareketli veya sabit durumdaki nesnelerin yön, hız, yörünge, şekil gibi özelliklerinin tespit edilmesi günümüzde birçok alanda ihtiyaç duyulan çok çeşitli yöntemler ile üzerine çalışmalar yapılmış, gelişime açık bir alandır. Savunma sanayisi, otonom araçlar, üretim bantları, güvenlik sistemleri ilk akla gelen başlıca alanlar olarak dikkat çekmektedir. Bu ihtiyaçları karşılamak için çeşitli sensör grupların yazılım ve donanım yöntemlerinden faydalanılmıştır. Ancak problem birçok alan için benzer görünse bile her alan için farklı yazılım ve donanım ihtiyaçları bulunmaktadır. Her sistemin artı ve eksi yönleri bulunmaktadır. Doğru sonuçları alabilmek için sistemin ihtiyacına göre kullanılacak donanım ve ona göre uygun kullanıcı arayüzü ve kontrol algoritmaları geliştirilmelidir.

Mesafe ölçümü gerektiren sistemlerde optik sensörler sıkça tercih edilmektedir. Ultrasonik sensörler ile kıyaslandığı zaman optik sensörler tespiti hedeflenen cismin üretildiği malzeme ve renginden yüksek oranda etkilenmektedir. Optik sensörlerin tespit edebildikleri maksimum uzaklık ultrasonik sensörlere göre ortalama 10 kat daha kısadır. Ortamın su altında olması ya da tozlu bir atmosferde çalışması durumunda da optik sensörler yüksek oranda olumsuz etkilenmektedir. Maliyet analizi kısmında yine optik sensörler yüksek maliyetleri ile tercih edilmesi şu aşamada zor görünmektedir [1].

Radar sistemleri problemin çözümü için başka bir seçenektir. Ancak bu sistemlerin donanım tasarımları karmaşık ve zordur. Kullanılması gereken sinyal çarpıcılar, filtreler, sinyal kuvvetlendiriciler gibi donanım bileşenleri gerektiği için yüksek maliyetleri bulunmaktadır. Tasarımları zordur. Çok yüksek frekanslarda çalışan bu sistemler çalışma ortamında yüksek oranda elektromanyetik gürültü oluşturmaktadırlar. Çok uzak mesafeler için kullanımı mümkündür [2]. Radyo frekansında çalışacak bir sistem ile yön ve cisim tespiti yapılacak ise yüksek frekansları sinyalleri yakalayabilmek ve gönderebilmek için ilk etapta bir anten ya da anten dizinine ihtiyaç duyulur. Daha sonra gerekli sinyallerin üretilmesi için ihtiyaca göre birden çok frekansta sinyal üretebilecek içerisinde anahtarlama devreleri, filtreler, yükseltici, alçaltıcı gibi elemanları barındıran bir sistem gereklidir. Sistemin kalibrasyonunu sağlamak için de bilinen gürültü seviyesi, sinyal seviyesi ve fazda test sinyalleri üreten bir donanıma ihtiyaç vardır. Sistemde kullanılan filtrelerin, kabloların ve alıcı bölümünün sisteme sinyal seviyesi, zaman ve gürültü seviyesi olarak ne gibi etkileri olduğunun bu

donanım ile tespiti gereklidir[3]. Sinyallerin işlenmesi ve tespit algoritmaları zor bir uzmanlık olup çok fazla test süreci ve geliştirme süresi gereklidir.

Basınç sensörleri bu tarz çalışmalarda kullanılan sensörlere bir diğer örnektir. Ses sinyalleri mekanik dalgalar olarak tanımlanır. Mekanik dalgalar yayılmak için hava, su gibi maddesel ortamlara ihtiyaç duyarlar. “Ses, nesnelere titreşiminden meydana gelen ve uygun bir ortam içerisinde (hava, su vb.) bir yerden başka bir yere, sıkışma ve genleşmeler şeklinde ilerleyen bir dalgadır. Dolayısıyla ses, bir basınç dalgasıdır”[4]. Bu ürünlerin çalışmamızla benzerlik gösteren genel özellikleri ses sinyalleri vasıtası ile tespit edilmesini, bir sensör dizininin faydalanmaları ve bilgisayar üzerinde bir arayüz vasıtası ile kullanıcıya polar grafik ile cisimlerin tahmini yerlerini göstermeleridir. Hava, kara, deniz platformlarında kullanım imkanı sağlayabilmektedirler. Bu sistemleri detaylı bir şekilde incelemek mümkün değildir çünkü çok kısıtlı bilgi ile firmalar ürünlerini tanıtmaktadır. Ürünler stratejik olarak önem arz ettiği için nasıl bir teknoloji kullanıldığının detaylı bir şekilde analiz edilmesi engellenmiştir. Ürünler ülkemizde ihtiyaç duyulan alanlarda kullanılmak istenmiş ancak satışı onaylanmamıştır. İnternet ortamında ürün bazında incelemek mümkündür ancak iç yapısı hakkında teknik bilgilere ulaşmak mümkün değildir.

Yurt dışında üretilen sistem incelendiğinde sistemde 4 adet basınç sensörü bulunmaktadır. Bu sistem de sadece alıcı konumunda bulunan sensörler ses ötesi hıza ulaşmış olan cisimlerin yaydığı ses sinyallerini oluşturduğu yüksek basıncın algılanması ile tespit yapmaktalar. Cismin tespitinin sağlanabilmesi için ultrasonik patlamanın gerçekleşmesi ve bunu sonucunda yüksek basınç oluşması şarttır. Aksi takdirde tespit yapılamamaktadır. Bu durum kullanım alanını oldukça sınırlandırmaktadır. Sistem tespit sonucunda hız ve yörünge bilgisini verememekte sadece bölge olarak tahmini alanlar ile cismin nereden geçtiğini operatöre göstermektedir. Sistem havada hareket halinde olan hedefe halat benzeri bir ip ile uzaktan bağlanmaktadır. Daha sonra aşağıdaki resimde görülen turuncu renkli perde açılmakta ve hedef alınarak vurulmaya çalışılmaktadır.

Tespit edilen cismin yönünü ve uzaklığını yakınlarda kurulmuş istasyondaki bir ekrana göndermekte ve atış sonucunda hedefi vuruş sonuçlarını anlık olarak operatörlere sunmaktadır. 0,10mS de bir ölçüm sonucu verebilen sistemin tespit edebileceği uzaklık 16 metreye kadar çıkabilmektedir[5].

Benzer çalışma yapısına sahip olan bir diğer sistem mikrofon ya da mikrofon dizini kullanılarak tasarlanmış ürünlerdir. Bu ürün grubu da benzer şekilde çalışmakta olup ortalama çapı 12cm den fazla cisimleri tespit edebilmektedir. Yine benzer şekilde pasif alıcı konumunda bulunan sensörlerden oluşmuşlardır. Ses ötesi hızlara çıkan cisimlerin yaydığı sinyalleri algılamakta ve kullanıcı arayüzüne ile tahmini konumlarını göstermektedir. Mermi benzeri mikrofon ile sinyalleri toplayıp adaptör benzeri ünite içerisinde işlemekte ve operatöre odasına kablosuz olarak gönderim sağlayarak operatöre sonuçları sunmaktadır [6].

Pasif sensör yapısıyla çalışan sistemlerin sonuç verebilmesi için ölçüm sahasında ses ötesi hıza ulaşması zorunluluğu bulunmaktadır. Bu hıza ulaşıldığı anda oluşan mikrofon ile algılanmakta ve tespit algoritmaları buradan çıkan sonuçlara göre çalıştırılmaktadır. Doğru sonuçların alınması ses seviyesinin oluşmasına bağlıdır. Bu durum sistemin tespit kabiliyetini ciddi anlamda sınırlandırmaktadır. Bu çalışmada da bir önceki ile benzer olarak bölgesel olarak cismin nereden geçtiği kullanıcıya verilmekte, hız ve yörünge bilgileri tespit edilememektedir.

Ultrasonik mesafe ölçümü ve doğrulanması (UMÖD) sisteminde alıcı verici birlikte bulunan bir sensör yapısı bulunduğu için böyle bir sınırlanma bulunmamaktadır. Kullanım alanı çok daha geniştir. Çok daha düşük hızda nesnelerin algılanması da hedeflenmektedir. Ancak sensör kabiliyetiyle artırılabilir olsa da ses hızından yüksek hızlarda hareket eden hedeflerin tespiti UMÖD için çok daha zordur. UMÖD sisteminde kullanıcıya mesafe ve yön bilgisinin yanı sıra hız bilgisine erişim imkanı da sunulması diğer ürünlerden ayıran önemli bir özellik olarak geliştirilecektir. Tespit edilebilecek mesafeler kıyaslandığı mevcut sistemlerle benzer sonuçlar alınacağı tespit edilmiştir. Mevcut sistemlerden gelişmiş olarak UMÖD sistemi kullanıcı arayüzü vasıtası ile tespit edilen nesnenin izlediği yörünge bilgisini de kullanıcıya sunulması amaçlanmaktadır.

Ultrasonik sensörler bu çalışma için uygun bir yapıdadır ancak dikkat edilmesi gereken bir takım unsurlar bulunmaktadır. Ölçüm sonuçlarının doğruluğu ölçümün yapılacağı hava ortamının sıcaklığı ve nem değerinden etkilenmektedir. Sonuçlar üzerinde çok büyük etkisi olmadığı için bu durum genellikle göz ardı edilmiştir [7]. Bu tez çalışmasında önerilen ultrasonik sensör, enerjisi kesilip tekrar verildikten sonra 250mS sürede ortam şartlarına göre kendisini kalibre edebilmektedir [8]. Dikkat edilmesi gereken bir diğer konu ise ses sinyalleriyle de bazı cisimleri tespit etmede zorlanmaktadırlar. Çok küçük cisimlerin geri

yasıma yapamaması ya da cismin şeklinden kaynaklanan yanlış geri yansımalar ile yansıyan sinyalin sensörün alıcı tarafında alınmaması gibi durumlar oluşabilmektedir. Ayrıca bazı cisimler (halı, elbise, vb.) ses dalgalarını yansıtmayıp sönümlemesine sebep olmaktadır. Bu gibi durumlarda geri dönüş sinyali alınamayacağı için tespit yapılması da mümkün değildir [7].

Bu çalışmada esas amaç özellikle savunma sanayimizde test sistemlerinde ihtiyaç duyulan hava, kara, deniz sistemlerinden ateşlenen vurulmak istenen hedefe gönderilen mühimmatların hız, yörünge, hedefe olan uzaklık gibi özelliklerinin belirlenip performans testlerini kolaylaştırmak, test süreçlerini kısaltıp sistemin gelişim sürecini hızlandırmak, sisteme doğru veriler sağlayarak hataların düzeltilip başarılı sonuçlar almasını sağlamaktır.. Kurulmak istenen sistemin ihtiyaçları göz önüne alındığı zaman ultrasonik sensörler ön plana çıkmıştır. Bu çalışmanın ihtiyaç olarak ortaya çıktığı alanda kullanılacak sistemin kesinlikle sahaya elektromanyetik bir kirlilik oluşturmaması gerekmektedir. Ayrıca hedef cismin rengi, üretildiği madde gibi özelliklerinden de etkilenmemesi önem arz etmektedir. Tabi ki en önemli beklentilerden biride düşük maliyetli bir çözüm olmasıdır.

Bu tez çalışmasında, sistemin tasarımına ilk etapta sensör seçimi yapılacak sonra veri toplama için gerekli olan altyapı oluşturulacaktır. Sensörlerden alınan veriler ARM STR 91 işlemcisinin üzerinde bulunan ADC tarafından dönüştürülebileceği ve sistemin ihtiyacı olan verilerin yeterli sürede elde edilebileceğine geçmişte yapılan çalışmalar ile kara verilirmiş ve kullanımı uygun olduğu düşünülmektedir. Sistem tasarımında sensörlerin bulunacağı ölçüm alanı ile operatör ortamının aynı yerde bulunmaması gerekmektedir. Bu nedenle uzaktan erişim için haberleşme altyapısı geliştirilecektir.

Benzer çalışmalar göz önüne alındığında operatör ortamının test sahasından çok daha uzakta olduğu dikkat çekmektedir. Bu amaçla seri haberleşme imkanı olan işlemci üzerinden operatör yazılımının bulunduğu bilgisayar ortamına veri akışını uzak mesafelerde sağlayabilmek için ethernet protokolüne dönüşümü hedeflenmiştir. Bu sayede 15 metre ile sınırlı olan seri haberleşme mesafesi 100 metre mesafeye kadar artırılabilme imkanı sağlamaktadır. Seri haberleşme ile 20 kbaut/s ile sınırlı olan iletim hızı ethernet haberleşmesi ile 10 gbps hızına kadar çıkabilmektedir [9].



Bilgisayar kısmında kullanıcı arayüzünün operatörler için günümüz yazılım teknolojisine uygun gösterimler ile sonuçları görüntülemelerini sağlamak amacı ile Visual Studio ortamında C# programlama dili ile algoritmalar yazılıp WPF platformunda görsel arayüz tasarımı gerçekleştirilecektir. Böylelikle sistemin ihtiyaç duyduğu bütünlük sağlanacaktır.

Donanım, donanım yazılımı, kullanıcı arayüzü yazılımı ve veri tabanı ile çalışmada sistem olarak bir eksiklik oluşması ihtimali ortadan kaldırılması çalışmanın temel amaçlarındandır.

Bu tez çalışmasından elde edilecek bilgi birikimi farklı sensörlerden yararlanılarak mesafe ölçümü ve hız bilgisi sonuçlarının iyileştirilebilmesine imkan sağlayacaktır. Tasarlanacak olan gömülü yazılım, arayüz yazılımı ve veri tabanı yazılımının farklı sensör gruplarının çalışmasına elverişli olması ve gerektiğinde gerek farklı türden sensörlerin birlikte kullanımı gerekirse tek tür sensör kullanımına imkan sağlayacaktır.

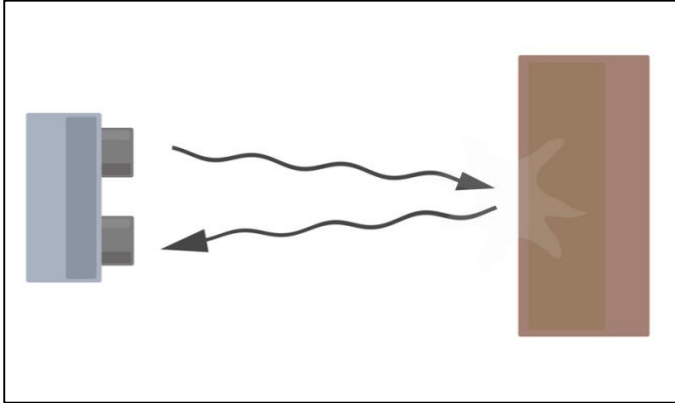


## 2. ULTRASONİK SENSÖRLER

Ultrasonik sensörler her hangi bir fiziksel temas olmaksızın havada iletilen ses sinyalleri ile nesnelerin tespitini mümkün kılmaktadır. Bu sensörler cisimlerin renklerinden, şekillerinden veya diğer fiziksel özelliklerinden etkilenmemektedir. Cisimlerin tespit edilmesi ve konumunun belirlenmesi için yüksek frekanslı ses sinyallerini kullanılmaktadır. Ultrasonik sensörler bir gönderici üzerinden ses sinyallerinin yollanıp en yakında bulunan nesne üzerinde geri yansması sonrası alıcı ile alınması arasında sürecinde geçen zamanı ölçme mantığı ile çalışırlar.

### 2.1. Ultrasonik Sensörlerin Çalışma Mantığı

Ultrasonik genellikle insan kulağının duyamayacağı yüksek frekanslı sesleri ifade etmek için kullanılır. İnsan kulağı için işitilebilecek aralık 20Hz ve 20kHz arasındadır. Yani ultrasonik dalgalar 20 kHz üstü frekansta bulunurlar. Bu analog sinyallerin bir hedefe yönelip geri yansması ile geçen zaman ve sinyalin hızı vasıtası ile nesnenin tespiti imkânını sağlamaktadırlar [1].



Resim 2.1. Sinyalin gönderilip yansıyanın geri alınması

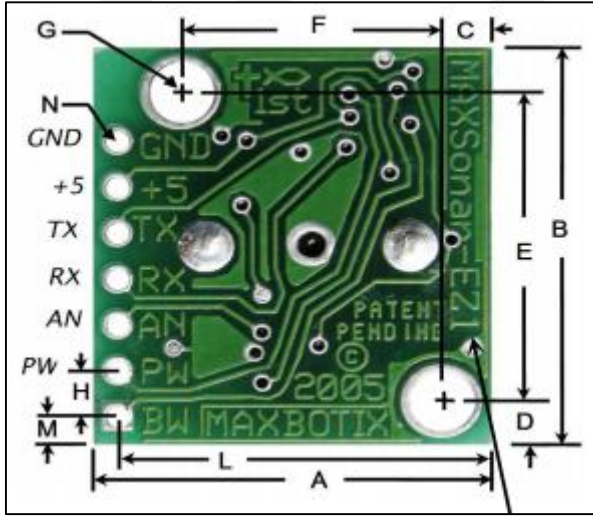
Bilindiği üzere ses sinyalleri hava ortamında ortalama 344 m/s hız ile harekete etmektedirler. Bir ses sinyalini gönderip geri aldığımız süreyi kaydedip bu süreyi 344 metre ile çarparsak bir tam turluk mesafesi hesaplamış oluruz [10].

$$Uzaklık = \frac{((Ses\ hızı) * (Geçen\ Süre))}{2} \quad (2.1)$$

Hesaplanan bu mesafe hedefe olan uzaklığı sinyalin iki kere kat etmesi sonucu yani gidip geri yansması sonucunda elde edildiği için bu değerin 2 ye bölünmesi ile sensör ile hedef arasındaki değer hesaplanmış olacaktır [10].

## 2.2. LV-MaxSonar-EZ Sensörlerin Kullanımı

2,5V - 5V arası gerilim ile çalıştırılabilen sensör yakın mesafelerden uzak mesafelere kadar tespit imkanı sağlamaktadır. Sensör 0m - 6,45m arasında nesnelere tespit edip 15,2cm - 6,45m arasında bulunan nesnelere uzaklık bilgilerini 2,54cm hassasiyet ile verebilmektedir. Sensör üzerinde 7 adet pin bulunmaktadır.



Şekil 2.1. Sensör pin yerleşimi

### Pin 1-BW

Normal kullanım için bu pin boşa bırakılmalıdır ya da 0V seviyesinde tutulmalıdır. 5V seviyesinde enerji verildiği zaman TX pini üzerinden seri haberleşme için gerekli olan veri akışı sağlanamaz bunun yerine birden çok sensörün birlikte kullanımı için TX pini darbe sinyalleri üretmeye başlar.

### Pin 2 –PW

Bu pin üzerinden darbe sinyalleri üretilmektedir. Darbe genişliğinden mesafe hesaplaması yapılabilir.

### Pin 3 –AN

Analog voltaj değeri okunarak mesafe tespiti yapılır.

### Pin 4 –RX

Sistem içerisinde başlangıç durumu olarak 3,3V seviyesinde tutulmaktadır. Her hangi bir bağlantı yapılmaz ya da 3,3V seviyesinde tutulur ise seri haberleşme için gerekli olan veri akışı sensörden elde edilebilir. 0V seviyesinde tutulur ise mesafe bilgisi alınmaz. Tekrar veri akışını sağlamak için en az 20uS ya da ne kadar süre mesafe bilgisi alınmak isteniyor ise 3,3V seviyesinde tutulmalıdır.

### Pin 5-TX

BW pini 0V seviyesinde tutulduğunda ya da herhangi bir bağlantı yapılmadığında asenkron seri veriler RS232 seri haberleşme formatında 0V- Vcc değerleri ile ifade edilerek bu pin üzerinde gönderilir.

### Pin 6-Vcc

Sistemin çalışması için gerekli olan enerji bu pin vasıtası ile verilir. 2,5V – 5,5V aralığında çalışmaktadır. Tavsiye edilen akım kapasitesi 5V için 3mA, 3V için 2mA seviyelerini karşılayabilen kaynaklar kullanılmalıdır.

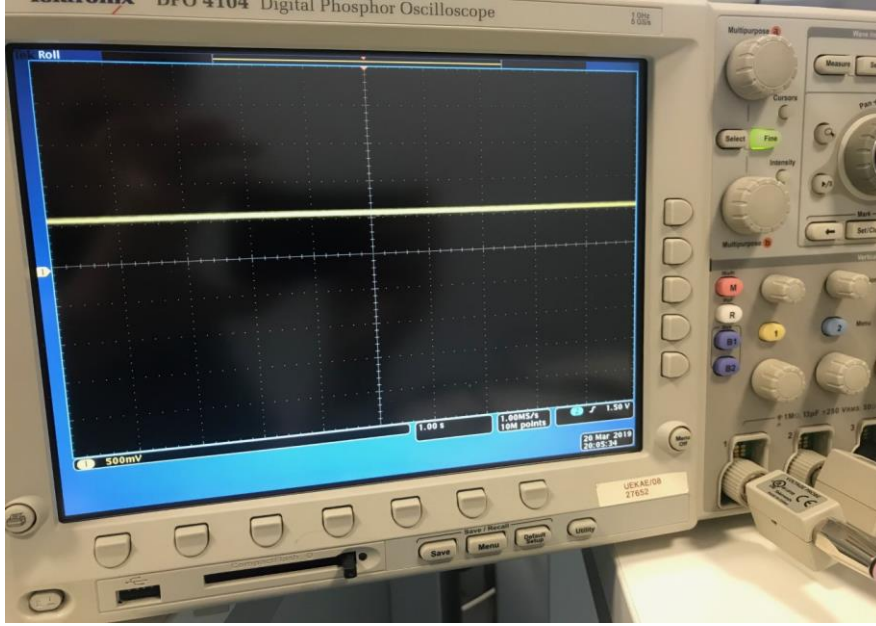
### Pin 7-GND

Sistem topraklamasının bağlanacağı kısımdır. Düzgün bir çalışma sağlanabilmesi için salınımın olmadığı temiz bir topraklama bağlantısı yapılması gerekmektedir.

#### **2.2.1. Genel kullanım şekli**

Sensöre enerji verildikten sonra ilk sinyalleri gönderip aldığı süreçte kendi kalibre etmektedir. Sensöre enerji verildiği zaman 250 ms sonra sensör veri alış verişi için hazır olmaktadır. Ardından ölçümleme işlemi 50 ms sürmektedir. Kalibrasyon sonrasında 50 ms

devirler ile sensörden mesafe bilgisi alınabilir. Kalibrasyon esnasında 35 cm den daha yakın bir mesafede hiçbir nesnenin olmaması sensör hassasiyeti ve doğru sonuçlar alabilmek için önemlidir. Çalışma esnasında gerçekleşen sıcaklık ve ya nem değişimleri için sensörün kendini otomatik kalibre etme özelliği yoktur. Sistem kapatılıp açılır ise yeni ortam şartlarına göre kendini tekrar kalibre etmektedir.



Resim 2.2. Tekli kullanım sensör analog çıkışı

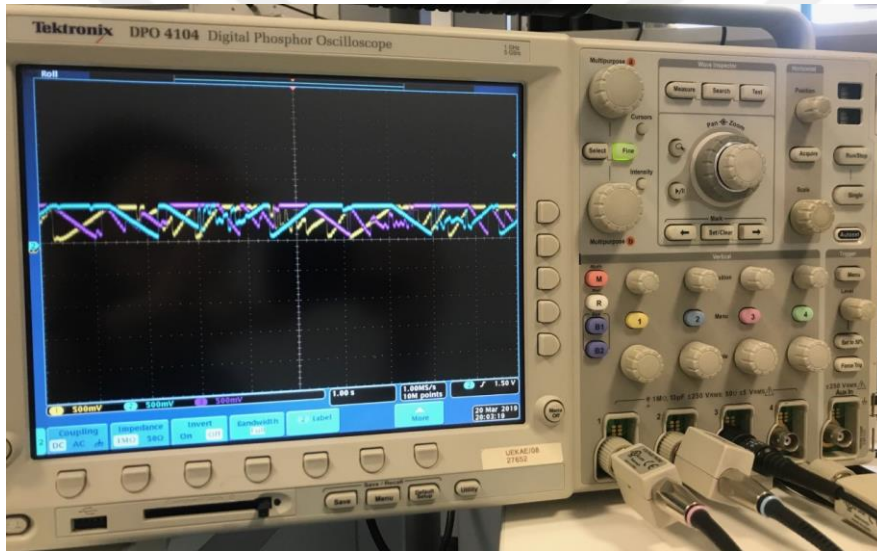
### 2.2.2. Dizin şeklinde kullanımı

Bir sistemde birden çok sensör aynı anda kullanılacağı zaman bu sensörlerin bir biri ile haberleşmesini sağlayacak bir protokol belirlenmiş olması gerekmektedir. Sürekli bir şekilde birden çok almanın aynı sistem üzerinde aynı anda başlatılması durumunda sensörler genellikle çalışmamaktadır.

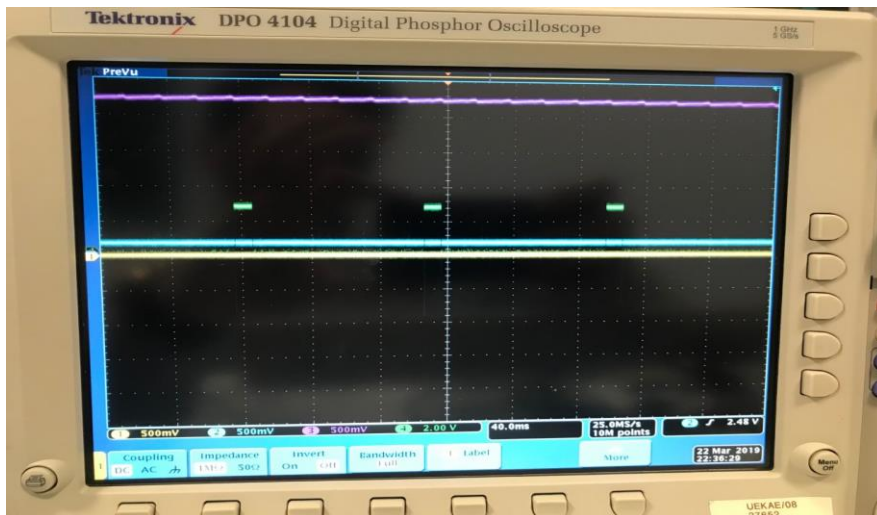
Sensörler üzerinde bulunan RX pini bağlantı yapılmaz ise, ilk etapta sensörler aynı anda ölçüm sonucunu verirler. Ancak sensörler senkronize olmadıkları için zaman geçtikçe ölçüm sonuçlarını verme süreleri arasında farklar oluşmaya başlar. Bu frekans kayması birçok uygulamada sensörler arasında girişim oluşmasına sebep olur. Eğer analog veri çıkışını takip ediyorsak voltaj seviyesini ölçerken düzenli olarak gürültü oluştuğunu görmeye başlarız. Ayrıca seri haberleşme kanalından da var olmayan ölçüm sonuçları gönderilmeye başlar.

Bu gürültü normal şartlarda ortamda bulunmamaktadır. Oluşma nedeni sensörlerin sinyallerinin birbirine girişim yapmasıdır.

Özellikle uzak mesafe ölçümlerinde bu durum çokça ortaya çıkmakta ve ölçüm sonuçları güvenilir olmamaktadır. Gürültü seviyesini olumsuz etkileyen bu durumun oluşmasına sensörlerin ölçüm sonuçlarını verme hızlarındaki farklılık neden olur. Bir sensör diğerinden çok az bir farkla da olsa hızlı ya da yavaş olabilir. Örneğin bir almacın çalışma süresi 49mS iken diğeri 49.2mS ise biri sinyal gönderim işlemine devam ederken diğeri alıcı durumuna geçmesi durumu oluşabilir. Bu durumda alıcı konumdaki sensör hedeften yansıyan kendi sinyali ile birlikte gönderici almacın sinyallerini de alır [11].



Resim 2.3. Üç sensörün aynı anda senkronizasyon sağlanmadan kullanımı

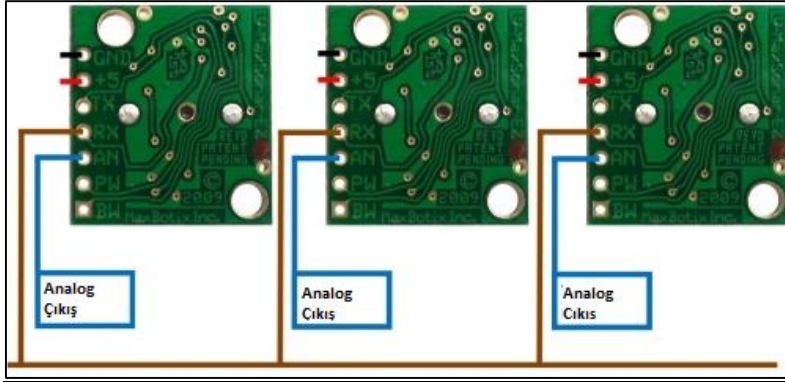


Resim 2.4. Üç sensörün aynı anda senkronizasyon sağlanarak kullanımı

Resim 2.4. de senkronizasyon yapılmış ölçüm sonuçları ve resim 2.3. de sensörlerin rastgele senkronizasyon olmaksızın birlikte çalışmaları durumunda oluşan ölçüm sonuçları verilmiştir. Görüldüğü üzere sensörler birlikte çalışmadıkları durumda eşleme sağlanmaz ise sonuçlar kararsız olarak gelmektedir. Bu durumda düzgün okuma yapmak çok zorlaşıp sisteme yanlış veri girişi sağlamakta bu da elde edilen değerlerin çoğunlukla yanlış olmasına yol açmaktadır. Bu durum da ortaya çıkan yanlış ölçümleri azaltmanın yolu sensörlerin eşleşmesinin sağlanmasıdır [11] .

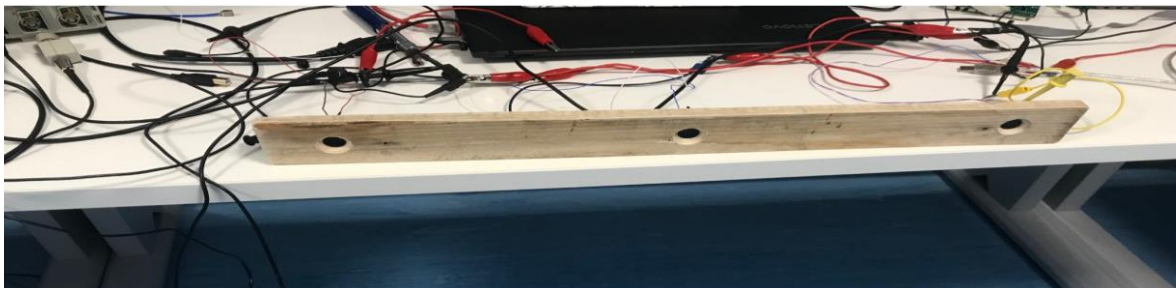
### Çoklu ultrasonik sensörlerin eş zamanlı olarak çalıştırılması

Birden çok almanın birlikte kullanımından önce eşzamanlı bir şekilde çalışmaları için yapılması gerekenler işlemler vardır. Kullanılacak olan sensörlerin RX pinlerini birbirine bağlanır. Daha sonra işlemci üzerinden çekilen bir pin bu ortak bağlantı noktasına bağlanır[11].



Şekil 2.2. Üç sensörün eş zamanlı çalıştırılma bağlantısı

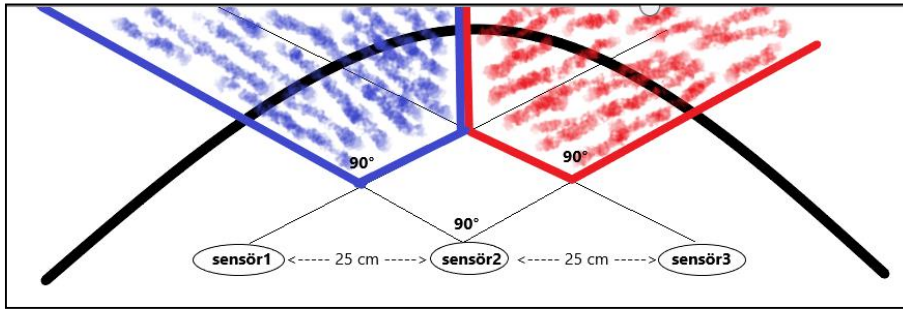
Tüm sensörlerin bağlandığı RX hattı üzerinde 0,20uS süre ile enerji verilir ise 50mS süre sonunda analog çıkışlar üzerinden veri okuması yapılabilir[12].



Resim 2.5. Sensör kullanımı laboratuvar test ortamı



Resim 2.5. sistemimiz üzerinde kullanılan üç adet almacın yerleşimini göstermektedir. Yaklaşık 0-15 cm mesafede sensörler cisimlerin var olduğunu tespit edebilmekte ancak doğru bir şekilde mesafe bilgisini verememektedir. Çalışmamızda bu limitler göz önüne alınarak yaklaşık 6,5 metre uzaklığa kadar tespit imkanı sağlayan sensörler 25 cm aralıklarla konumlandırılmış ve 90 derece açılık bir alanı taradıkları için aşağıdaki şekilde belirtilen aktif çalışma mesafeleri elde edilmiştir.



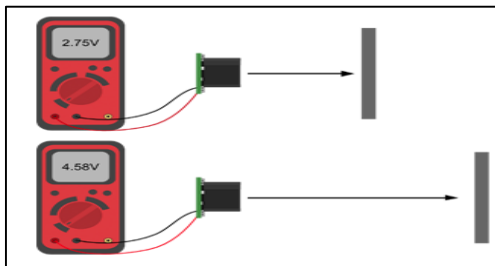
Şekil 2.3. Sensör yerleşimi ve kapsama alanı

### 2.3. Ultrasonik Sensörlerin Veri Aktarımı ve Haberleşme Yöntemleri

Sensör gerekli bağlantılar yapıp çalışması sağlandıktan sonra üç farklı yöntem ile veri alışverişi yapmak ve sensör üzerinden gelen verileri anlamlandırmak mümkündür.

#### 2.3.1. Analog veri aktarımı

Sensör gerekli işlemleri yaptıktan sonra ölçüm sonuçlarına göre mesafe bilgisini yaklaşık 50mS aralıklar ile voltaj türünden her bir 1 inç için  $V_{cc}/512$  oranında değere karşılık gelecek şekilde (Pin 3 –AN)\_üretmektedir [8]. Örneğin 5V ile beslenen bir sistemde 1V değeri okunuyor ise  $(5/512) \times 1 = 2,59m$  mesafede bir nesne olduğu anlamına gelmektedir.



Resim 2.6. Analog ölçüm yöntemi

### 2.3.2. Seri haberleşme ile veri aktarımı

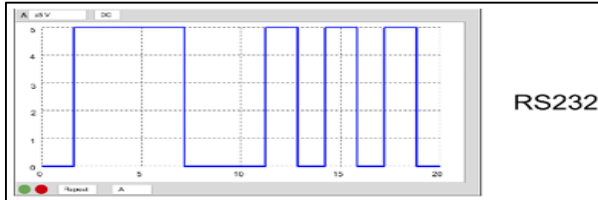
Sensör üzerinde bulunan seri haberleşme pinleri vasıtası ile mesafe bilgisini direk olarak RS232 haberleşmesi ile inç türünden verebilmektedir.

Sensör bütün karakterlerin ASCII karşılığını byte olarak göndermektedir. İlk olarak “R” harfinin ASCII karşılığı olan 82 sayısını gönderir. Bunu takip eden 3 karakter inç türünden mesafe bilgisini vermekte son karakter olarak ise “ENTER” anlamına gelen ASCII 13 sayısını göndermektedir. Örnek olarak 2,59 metre mesafedeki bir nesne tespit edildiği zaman R102<sup>↵</sup> serisi bilgi olarak okunacaktır [8]. 2,59 m ~ = 102 inç. R = 82 (ASCII), <sup>↵</sup> = 13(ASCII).

Her bir değer 8 bit ile ifade edilmektedir. Böylelikle 4 byte ile mesafe bilgisi ifade edilmiş olmaktadır.

$$(0101\ 0010) - (0000\ 0001) - (0000\ 0000) - (0000\ 0010) - (0000\ 1101) \quad (2.1)$$

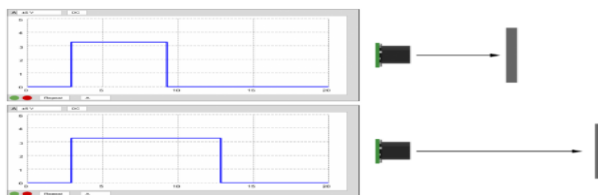
82   -   1   -   0   -   2   -   13



Şekil 2.4. Seri haberleşme veri sinyalleri

### 2.3.3. Darbe genişliği ile veri aktarımı

Her bir 1 inç 147uS’lik darbe genişliğine karşılık gelmektedir. 0,88mS ile 37,5mS’lik aralıkta üretilen darbe genişlikleri ile mesafe bilgisi elde edilebilir [8].



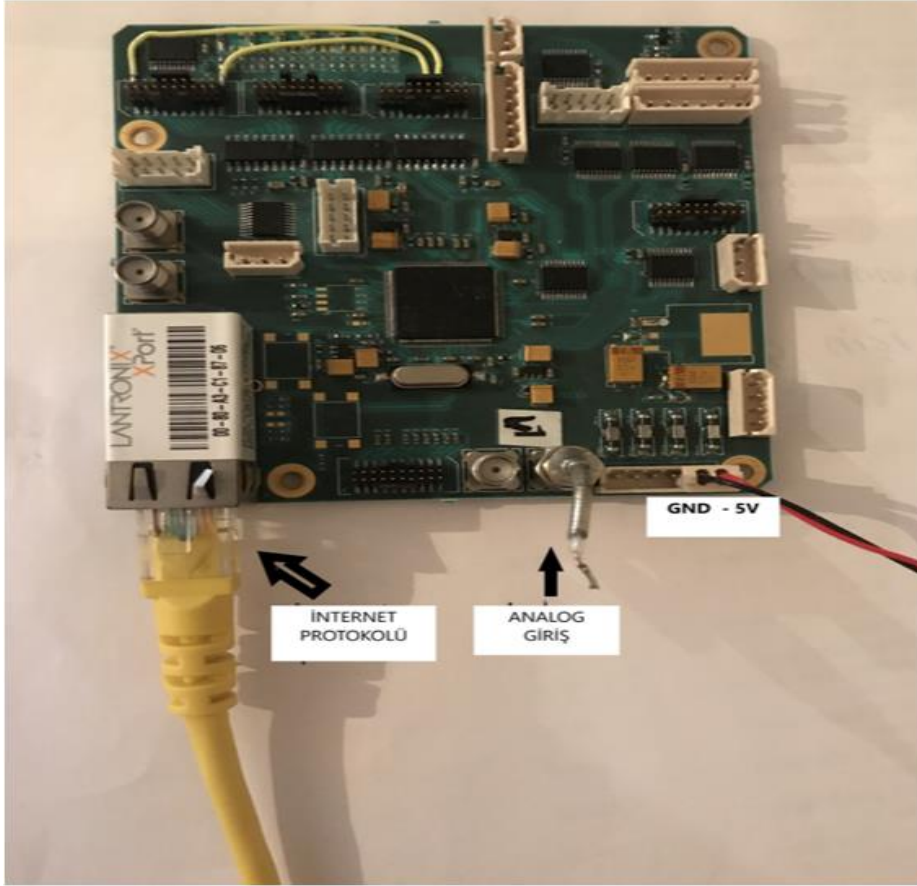
Şekil 2.5. Darbe genişliği yöntemi veri sinyalleri

### 3. VERİLERİN TOPLANMASI VE İŞLENMESİ

Bu çalışmada veri toplama yöntemi olarak sensör üzerinden analog sinyallerin alınması ve kontrol kartı üzerinde bulunan işlemci ile işlenmesi daha sonra bilgisayar ara yüzüne aktarılması şeklinde bir mimari tasarlanmıştır. Üç yöntemden analog yöntemin tercih edilmesi verilerin filtrelenmesine imkan tanıyacak, hassasiyeti arttıracak ve analog verilerin önce bir işlemci vasıtası ile alınıp dijital verilere dönüştürülmesi daha sonra internet protokolü ile bilgisayara aktarılması uzak mesafe haberleşmesine imkan sağlayacaktır.

Analog veri okuma işleminin üç sensör içinde aynı anda yapılması gerekmektedir. Aynı anda okuma yapılmaz ise yani sensörler senkron çalışır iken okuma işleminde gecikmeler ya da kaymalar yaşanır ise bu yanlış hesaplamalara neden olacaktır. Bunun için planlanan çözüm yöntemi tek bir işlemci ile analog veri okuması yapabilen üç ayrı kanal vasıtası ile verilerin işleme alınması gerekli hesaplamalar yapıldıktan sonra hangi verinin hangi sensörden geldiği bilgisinin de eklenmesi ile birlikte ölçüm sonuçlarının bilgisayar arayüzüne aktarılmasıdır. Mevcut sistemde kullanılan işlemci 8 ayrı kanaldan okuma yapabilme kabiliyetine sahiptir. Ancak daha önce tasarlanmış olan ve bu projede test amaçlı kullanılan kontrol kartı üzerinde 3 giriş için uygun bacaklar bırakılmadığı için üç kanaldan aynı anda okuma işlemi ancak simülasyon üzerinde yapılabilecektir.

Üç kanal vasıtası ile alınan veriler aynı dijital filtreden geçirilerek gürültü sinyallerinden arındırılmakta ve en doğru ölçüm sonuçlarının alınması için algoritmalarından geçirilmektedir. Aynı matematiksel hesaplamalar ile mikroişlemci içinde mesafe bilgisini elde edilmesinden sonra bu üç verinin hangisinin hangi sensörden geldiği bilgisi yörünge , yön ve hızın tespit edilebilmesi için doğru olarak tespit edilmelidir. Hangi kanala hangi sensörün bağlı olduğu bilindiği için mikroişlemci verilerin hangi bacadan geldiği bilgisine göre sensörleri birbirinden ayırt edebilir. Gerekli işlemler sonucunda mesafe bilgisi cm cinsinden elde edildikten sonra bu bilgiye bir ön kod eklenerek bilgisayar arayüzüne gönderimi sağlanacaktır. Arayüz yazılımı ön kod ile birlikte aldığı verilerin ön kodları kıyaslayarak hangi sensörden hangi verilerin geldiğini tespit edebilecek durumu gelmiş olacaktır. Bu tespit işlemi sonrasında cm cinsinden ön kodun devamında yer alan mesafe bilgisini alır ve yön, hız yörünge bilgilerini gösterebilir.



Resim 3.1. Kontrol kartı bağlantıları

### 3.1. Kontrol Kartı ile Verilerin Toplanması

Bu çalışmada kontrol kartı üzerinde bulunan işlemci vasıtası ile verileri toplayıp işleme ve bilgisayara iletme işlemlerinde kullanılmıştır. Kontrol kartı üzerinde resim 3.1. ile belirtilen giriş ile analog sinyaller toplanmakta ve işlemciye iletilmektedir. Daha sonra işlemci gerekli algoritmaları çalıştırıp veriyi bilgisayara aktarma görevini üstlenmiştir.

### 3.2. İşlemci ile Verilerin İşlenmesi

İşlemci olarak ARM STR91 kullanılmıştır. Keil programı ile C programlama dilinde kodlama yapılmıştır. İşlemci kontrol kartı ve sensörlere enerji verildiği andan itibaren veri toplamaya başlamaktadır.

### 3.2.1. ADC kullanımı

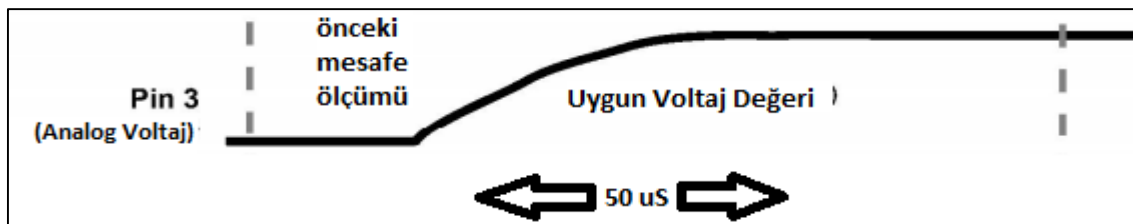
İşlemci üzerinde 8 adet 10 bit analog verileri dijital verilere dönüştürücü bulunmaktadır. 0V – 3,6V arasındaki analog değerleri 0,7uS sürede dijital veriye dönüştürmektedir [10]. ADC çözünürlüğü  $(3,6 / 1023) = 0,00352$  olarak hesaplanabilir. Sensör tarafından üretilen mesafe bilgisinin dijital dönüşümü işlemci üzerinde bulunan dönüştürücü ile yapılmaktadır. Örnek olarak 2,59m mesafede bir nesne tespit edilmiş ise sensör analog çıkışından 1V değerini üretecektir.

$$1 \times (1023 / 3,6) = 284,16 \quad (3.1)$$

Bu veri işlemci tarafında dijital 10 bit (00 0001 0001 1100) = 284 olarak ADC tarafından mikroişlemcinin kullanabilmesi için dönüştürülmekte ve işlemciye aktarılmaktadır.

### 3.2.2. Verilerin filtrelenmesi

Özellikle ani mesafe değişimlerinde ve ilk tespit anlarında okunan değerler çok salınım göstermektedir. Sinyal seviyesindeki bu ani değişimler işlemci tarafında yanlış okumalara sebep olabilmekte ve hatalı ölçüm sonuçları doğmasını sebep olabilmektedir. Bunun engellenmesi için işlemci içerisinde verileri filtrelemek gerekmektedir.



Şekil 3.1. Analog veri ölçüm süresi

ADC yaklaşık 0,7uS sürede bir işlemciye veri sağlamakta ancak sensörün doğru veriler üretmesi 50mS sürmektedir [8]. Öncelikli olarak bu süre zarfı beklenmeli öncesinde alınan bilgiler göz ardı edilmelidir. Sensör limitleri dışındaki değerler, çalışma aralığı dışındaki salınımlar bu kısımda filtrelenebilmektedir. Gelişmiş algoritmalar ile sistem verimliliği artırılabilir.

### 3.2.3. Matematiksel algoritmalar ile anlamlandırılması

Analog veri dijital veriye dönüştürülüp işlemciye gönderildikten sonra gerekli filtreleme işlemlerini yapmak, mesafe bilgisine dönüştürmek ve bilgisayar arayüzüne aktarmak için ölçülen voltaj değerinin karşılık geldiği analog değer işlemci içerisindeki algoritmalar ile yeniden hesaplanması gerekmektedir.  $(284 \times 3,6) / 1023 \sim = 1V$ .

Hesaplanan bu değer mesafe bilgisinin hesaplanması için bilgisayara aktarılmaktadır. Mesafe bilgisi işlemci tarafında hesaplanıp bilgisayara direk olarak da gönderilebilir.

### 3.3. Verilerin Bilgisayar Arayüzüne Aktarılması

Kontrol kartı üzerinde bulunan ethernet seri haberleşmeden ethernette ethernetten seri porta dönüş yapılmasını sağlayan ve bu sayede uzun mesafelerde iletim imkanı sağlan xport olarak isimlendirilen dönüştürücü eleman bulunmaktadır. Bu eleman mikroişlemci üzerinden seri port ile aldığı bilgileri dönüştürüp ethernet protokolü ile bilgisayara aktarmakta, aynı şekilde bilgisayar arayüzünden ethernet protokolü ile aldığı verileri seri haberleşme protokolü ile işlemciye gönderme görevini üstlenmektedir.

Veriler seri haberleşme ve ethernet için 64 byte olarak gönderilebilmektedir. Bu gönderim esnasında mesafe bilgisinin yanı sıra hangi ölçüç değerinin hangi sensörden geldiği bilgisinin de bilgisayar tarafına aktarılması gerekmektedir. Bu sebeple ölçüm değerini göndermeden önce ön kod olarak örneğin birinci sensörden gelen veri gönderiliyor ise ilk 7 byte'lık veriye SENSOR1 ifadesinin her bir harfinin gönderimi yapılacaktır. Takip eden 3 byte ile cm cinsinden ölçüm sonucu gönderilecektir.

S-E-N-S-O-R-1-6-2-5

(3.3)

Olarak 10 byte veri mikroişlemci tarafından bilgisayar arayüzüne gönderilir.

#### Mikroişlemci içerisinde yazılım akış şeması

Yukarıda belirtilmiş olan bütün bu işlemler C programlama dili ile programlanarak mikroişlemci içerisinde gömülmülü olacak şekilde planlanmıştır. Birbirini takip eden bu

işlemler için gerekli olan algoritmalar belirli bir sıra ile çağırılmalıdır. Bunu için gömülü yazılım kısmında yapılacak olan akış şeması ve algoritmaların yapacağı işlemler çizelge 3.1. ile açıklanmıştır.

Çizelge 3.1. Gömülü yazılım akış çizelgesi

Ana Fonksiyon

{

İşlemci için gerekli olan konfigürasyon ayarlarının yapılması.



Haberleşme ve veri gönderimi alımı için gerekli olan konfigürasyon ayarlarının yapılması.



While(true)

{

Sensörlerin senkronizasyonunu sağlanması için gerekli olan darbe sinyallerini üretilmesi.



Uç ayrı sensörden gelen verilerin alınması ve analog verilerin dijital verilere dönüştürülmesi .



Alınan verilerin voltaj seviyesine göre dijital filtreden geçirilmesi doğru mesafe bilgisinin elde edilmesi.

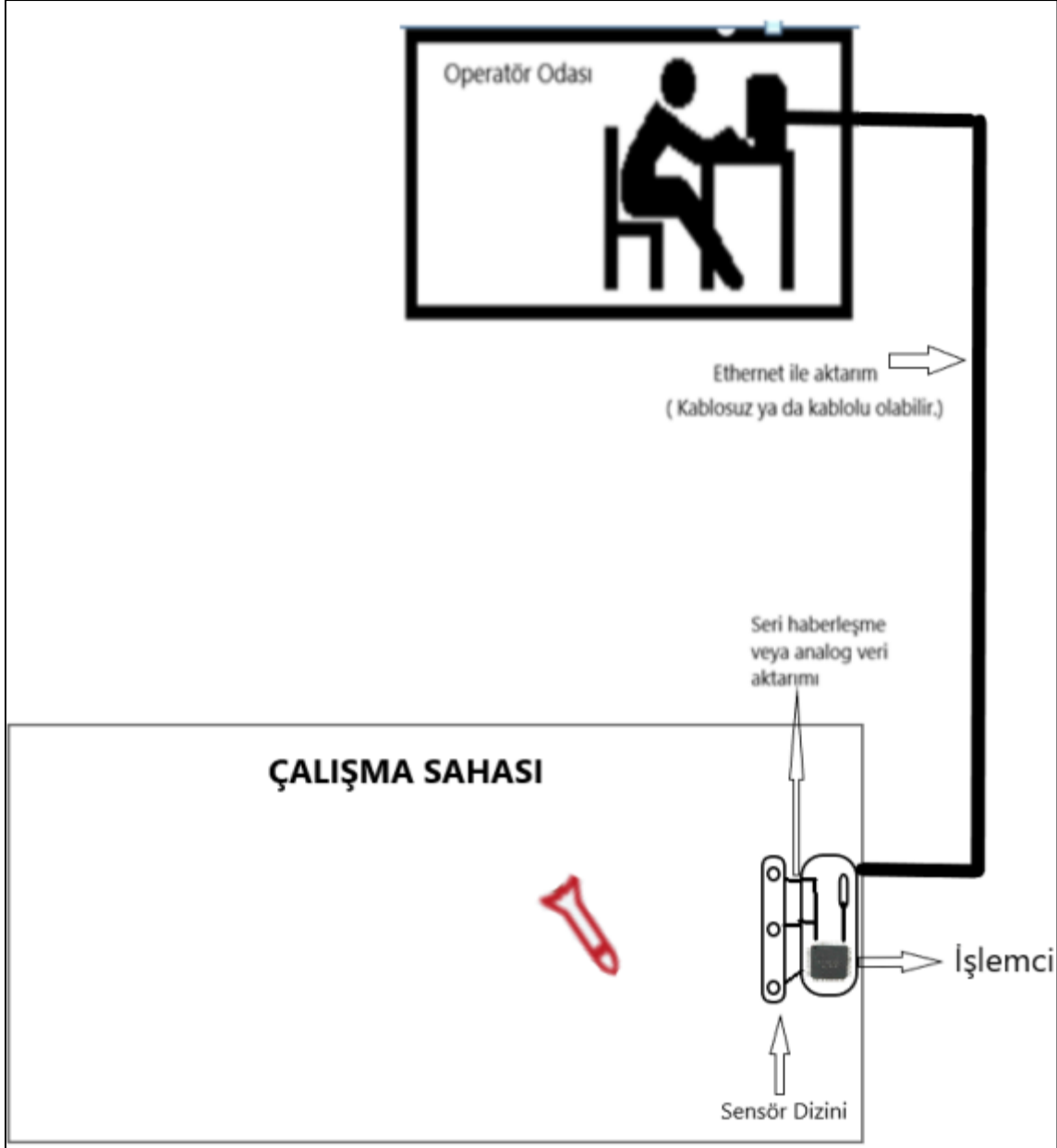


Elde edilen verilerin hangi sensörden geldiği bilgisi eklenerek bilgisayar arayüzüne aktarılması.

} }

### İşlenen verilerin aktarılması için sistem tasarımı

Sahadan elde edilen ve anlatılmış olan aşamalardan geçen veriler resim 3.2 de belirtildiği şekilde operatör odasına aktarılacak ve kullanıcı arayüzü ile operatöre gerekli olan bilgiler sağlanacaktır.

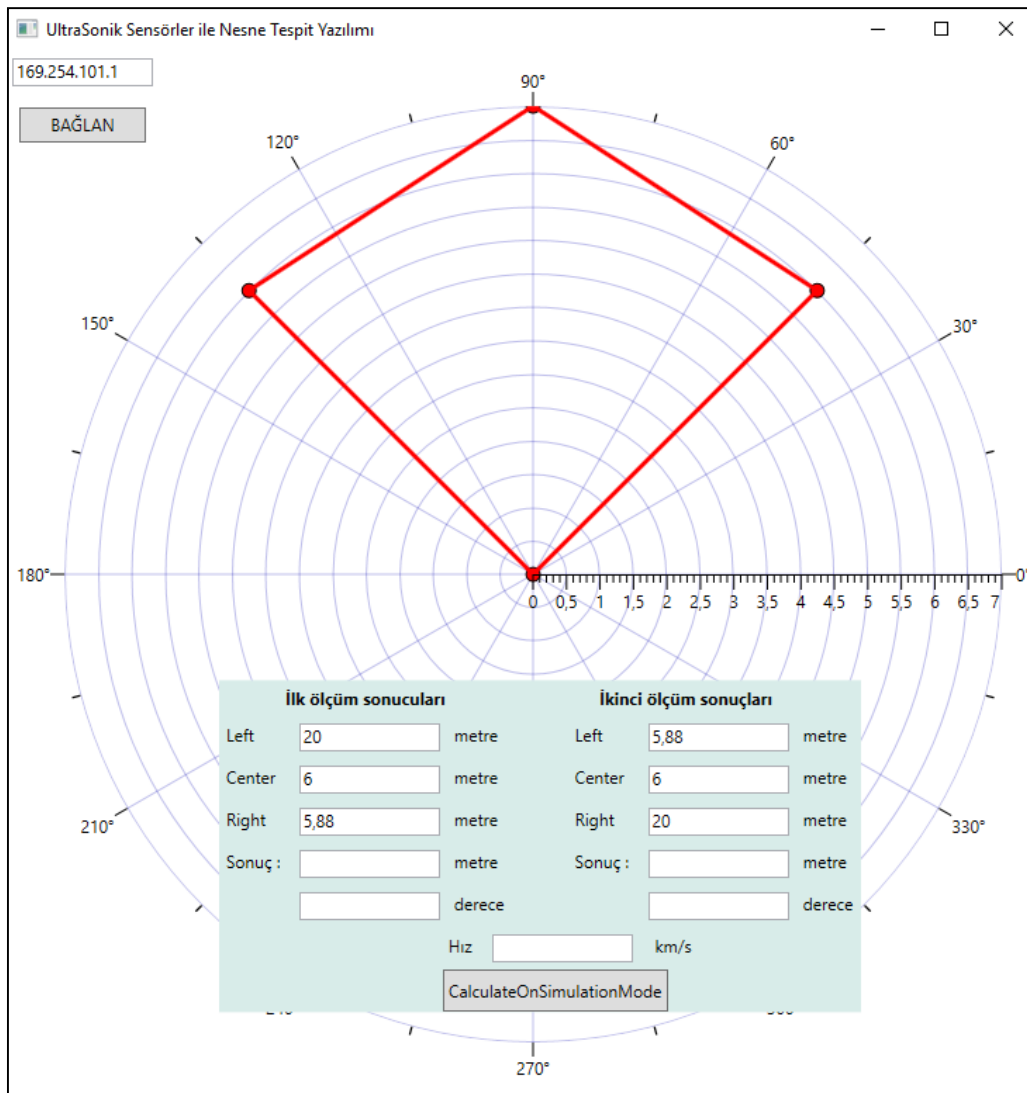


Resim 3.2. Çalışma prensibi ve veri aktarımı



## 4. KULLANICI ARAYÜZÜ

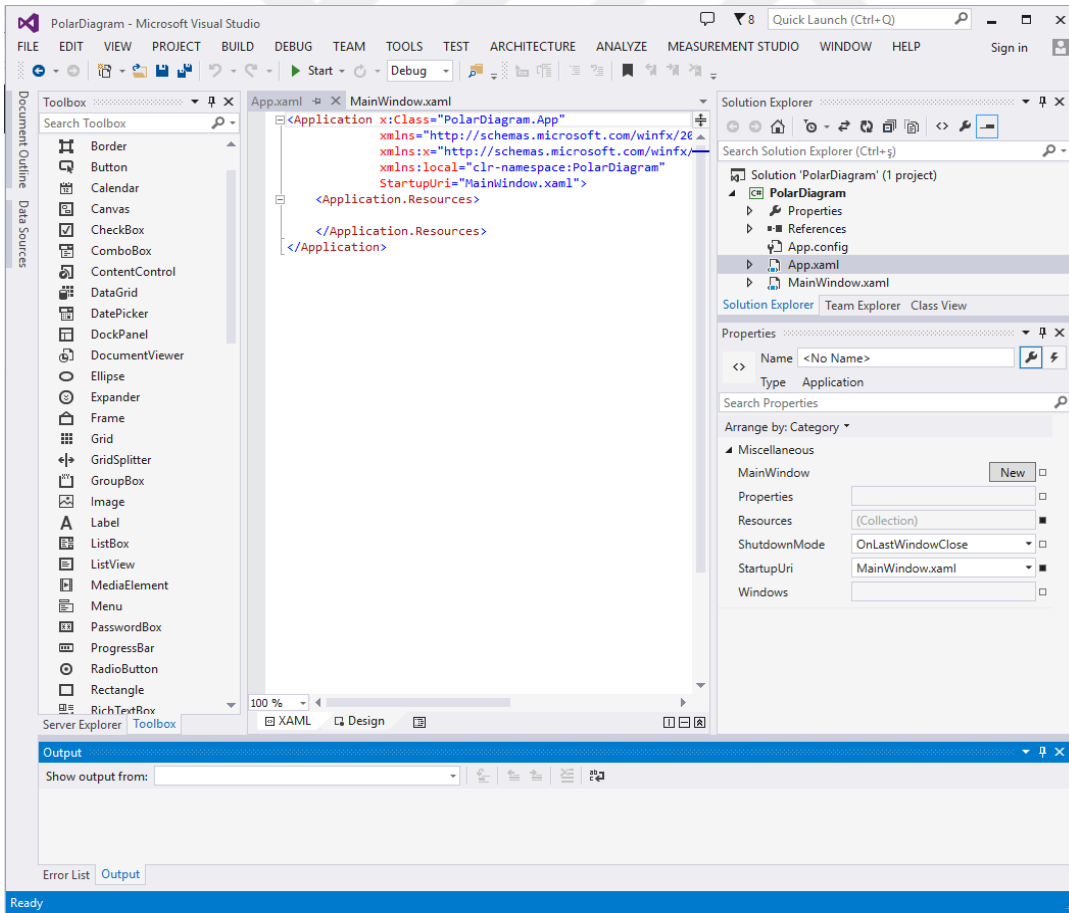
Özellikle uzaktan erişilen sistemlerde sistemin kontrolü, verilerin anlamlandırılması, görselleştirilmesi için kullanıcılara anlaşılır ve kullanımı kolay bir arayüz sunulması önem az ettiği için bu çalışmada gelişmiş arayüz teknolojileri kodlama metotları ve güncel görsel tasarım yöntemleri üzerine araştırma yapılarak en uygun olan seçimin yapılması için çalışılmıştır. Sade ve anlaşılır bir arayüz tasarımı ile kullanıcıya ihtiyacı olan tüm bilgiler sunulmuştur.



Resim 4.1. Arayüz tasarımı görüntüsü

#### 4.1. Arayüz tasarımı

Bu çalışmada kullanıcı arayüzü tasarımı için Microsoft Visual Studio isimli program tercih edilmiştir. Gelişmiş kodlama teknolojisine sahip olan bu program, C dilinin yanı sıra nesne tabanlı programlama dili olan C++ ve C# ile kodlanma yapılmasına imkan sağlamaktadır. Özellikle görsel bir kullanıcı arayüzünün bulunacağı projeler için C# programla dili ile geliştirme yapmak gelişmiş mimarisi ve sunduğu arayüz araçları ile sayesinde birçok ihtiyacı karşılayabilmektedir. Visual Studio tümleşik geliştirme ortamı bir düzenleme, hata ayıklama ve kod oluşturmak için kullanın ve ardından bir uygulama yayımlama imkanı tanıyan programdır. Bu tümleşik geliştirme ortamı (IDE) birçok yönüyle yazılım geliştirme için kullanılabilen zengin bir programdır. Standart düzenleyici ve hata ayıklayıcıların çoğu sağlamazken, Visual Studio yazılım geliştirme işlemini kolaylaştırmak için derleyiciler, kod tamamlama araçları, grafik tasarımcıları ve daha birçok özellik içerir. [12]



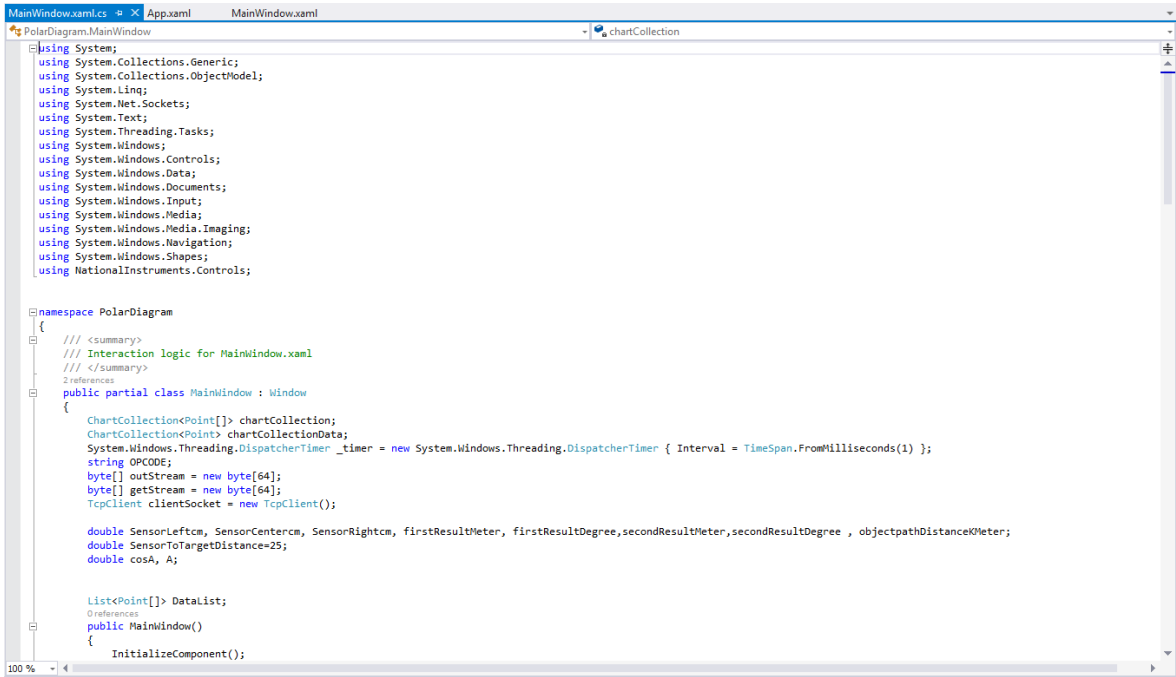
Resim 4.2. Araç kutusu ve arayüz tasarımı yapılan kısım

## Araç kutusu

Tasarım için kullanacağım ihtiyaç duyduğumuz araçları içeren penceredir. VS birçok gelişmiş araç ile tasarım yapma kabiliyetlerimizi artırmaktadır. Bu çalışmamızda TextBox, Label, Button, GroupBox, Grid, Stackpanel gibi araçların yanı sıra National Instruments firmasının Measurement Studio isimli kütüphanesinin iki boyutlu polar gösterim grafiği kullanılmıştır.

## Çözüm gezgini

Bu çalışma içerisinde bulunan tasarım projesi, arka plan kodlamaları, yapılandırma dosyaları, yazılım kütüphaneleri, referans olarak kullanılan diğer çalışmalar, projenin kullanılacağı ve kodlanacağı platformla alakalı birçok ayarın bulunduğu özelliklerin özellikler dosyası bu seçenek içerisinde bulunmaktadır. Bu çalışmada bulunan Mainwindow.xaml isimli dosya görsel tasarım arayüzünün proje dosyasıdır. Ayrıca yine bu dosyanın içerisinde yer alan Mainwindow.xaml.cs dosyası asıl algoritmaların çalıştırıldığı ve sistemin ihtiyaç duyduğu kodlamaların yapıldığı kısımdır.



```

using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using NationalInstruments.Controls;

namespace PolarDiagram
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    /// </summary>
    public partial class MainWindow : Window
    {
        ChartCollection<Point[]> chartCollection;
        ChartCollection<Point> chartCollectionData;
        System.Windows.Threading.DispatcherTimer timer = new System.Windows.Threading.DispatcherTimer { Interval = TimeSpan.FromMilliseconds(1) };
        string OPCODE;
        byte[] outputStream = new byte[64];
        byte[] getStream = new byte[64];
        TcpClient clientSocket = new TcpClient();

        double SensorLeftcm, SensorCentercm, SensorRightcm, firstResultMeter, firstResultDegree, secondResultMeter, secondResultDegree, objectpathDistanceKMeter;
        double SensorToTargetDistance=25;
        double cosA, A;

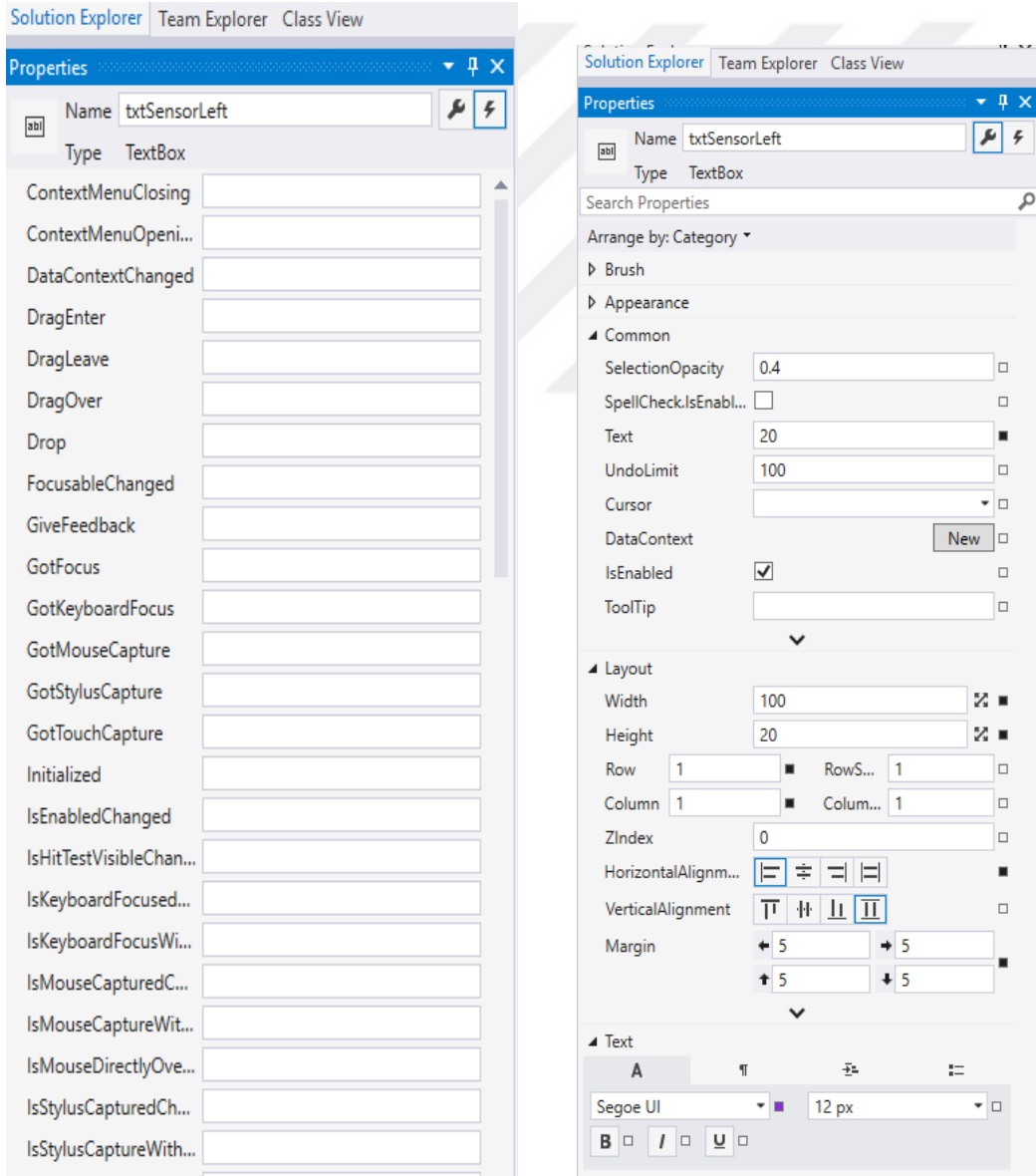
        List<Point[]> DataList;
        public MainWindow()
        {
            InitializeComponent();
        }
    }
}

```

Resim 4.3. Arayüz yazılımı ve kodlama ekranı

## Özellikler

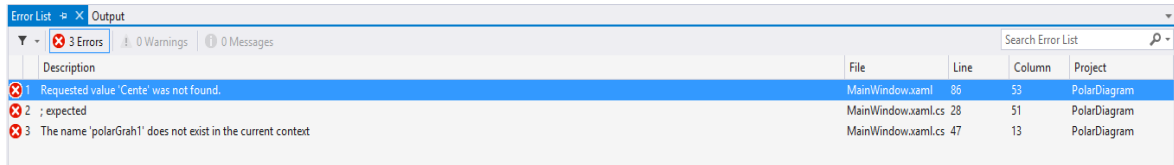
Araç kutusu içerisinde alınıp projeye eklenmiş olan her bir aracın bu pencere içerisinde yer alan birçok özelliği tasarım istekleri ve çalışmanın gereklilikleri çerçevesinde değiştirilebilir. Temel olarak renk, yazı boyutları, büyüklük gibi görsel özelliklerin yanı sıra kullanılan elemanın türüne göre değişmekle birlikte arka planda hangi özelliğiyle ilgili işlem yapılacak ona tepki vermesi için oluşturulmuş özelliklerinde biri seçilerek o olay gerçekleştiği zaman istenen işlemleri yapması, gerekli olan algoritmaların çalışması sağlanabilir.



Resim 4.4. Kullanılan araçların özelliklerinin ayarlanması

## Çıktılar-hata listesi

Görsel ekranların ve arka plandaki programlamaların yapılmasının ardından yazılım derlenir. Derleme sonucunda yazılım içerisinde düzeltilmesi önerilen uyarılar ve yapılmış olan hatalı kodlamalar bu pencere vasıtası ile yazılımcıya gösterilip gerekli düzeltmeleri yapması sağlanır. Bu pencere sayesinde dikkatimizden kaçan yanlışlıkları tespit etmemiz kolaylaşır ve projenin düzgün bir şekilde çalışması sağlanır.



Resim 4.5. Hata ayıklama penceresi

### **4.1.1. Windows Presentation Foundation (WPF) uygulaması**

WPF windows tabanlı istemci uygulamaları geliştirmemize yarayan bir kullanıcı arabirimidir. Bu geliştirme platformu uygulama modeli, kaynaklar, denetimler, grafik, düzeni, veri bağlama, belgeleri ve güvenlik dahil olmak üzere geniş bir uygulama geliştirme altyapısını desteklemektedir. WPF, Extensible Application Markup Language (XAML) olarak isimlendirilmiş olan programlama dili ile programlanır. XAML bildirim temelli bir biçimlendirme dilidir. Oluşturulan tasarımları biçimlendirmede ve ardında arka planda yapılacak kodlamalar için etkileşimleri oluşturmada kullanılmaktadır.

### Tasarım için gerekli olan kütüphanelerin eklenmesi

```
<Window xmlns:ni="http://schemas.ni.com/controls/2009/xaml/presentation"
x:Class="PolarDiagram.MainWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:local="clr-namespace:PolarDiagram"
  mc:Ignorable="d"
  Title="UltraSonic Sensörler ile Nesne Tespit Yazılımı" Height="790" Width="750" >
```

Pencere içerisinde yer alacak elemanların ihtiyaç duyduğu yazılım kütüphaneleri çalışmaya eklenmiş. Ana pencerenin boyutu belirlenmiş ve pencerenin isimlendirilmesi yapılmıştır.

## Görsel arayüzün tasarımı

```
<Grid>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="auto"></ColumnDefinition>
    <ColumnDefinition Width="auto"></ColumnDefinition>
    <ColumnDefinition Width="auto"></ColumnDefinition>
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="253*"></RowDefinition>
    <RowDefinition Height="189*"></RowDefinition>
    <RowDefinition Height="317*" ></RowDefinition>
  </Grid.RowDefinitions>
```

Bu kod bloğu ile tasarımın temelini oluşturan satır ve sütunlar oluşturulmuş ve araç kutusu penceresinden eklenecek olan elementler için düzenli bir yerleştirme sağlanmasına temel oluşturulmuştur. Uygun olan satır ve sütun numarası verilerek araç kutusundan eklenen elementler kolayca arayüz üzerinde konumlandırılabilir.

## Polar diyagram grafiğinin çalışmaya eklenmesi

```
<ni:PolarGraph Name="polarGraph1" VerticalAlignment="Top" Grid.Column="2" Grid.Row="0"
Margin="0,0,10,0" Height="744" Grid.RowSpan="3">
  <ni:PolarGraph.AngularAxis>
    <ni:PolarAngularAxis>
      <ni:PolarAngularAxis.MajorGridLines>
        <ni:GridLines Stroke="#400000B0"/>
      </ni:PolarAngularAxis.MajorGridLines>
    </ni:PolarAngularAxis>
  </ni:PolarGraph.AngularAxis>
  <ni:PolarGraph.RadiusAxis>
    <ni:PolarRadiusAxis Radius="10" Visibility="Visible" >
      <ni:PolarRadiusAxis.MajorGridLines>
        <ni:GridLines Stroke="#400000B0"/>
      </ni:PolarRadiusAxis.MajorGridLines>
    </ni:PolarRadiusAxis>
  </ni:PolarGraph.RadiusAxis>
  <ni:PolarGraph.Plots>
    <ni:PolarPlot Label="Plot 1">
      <ni:PolarPlot.Renderer>
        <ni:PlotRendererGroup>
          <ni:PointPlotRenderer Size="10,10" Shape="Ellipse" Stroke="Black" Fill="Red" StrokeThickness="1" />
          <ni:LinePlotRenderer Stroke="Red" StrokeThickness="3"/>
        </ni:PlotRendererGroup>
      </ni:PolarPlot.Renderer>
    </ni:PolarPlot>
    <ni:PolarPlot Label="Plot 2" >
      <ni:PolarPlot.Renderer>
        <ni:PlotRendererGroup>
          <ni:PointPlotRenderer Size="7,7" Shape="Ellipse" Stroke="Black" Fill="White" StrokeThickness="1" />
          <ni:LinePlotRenderer Stroke="Black" StrokeThickness="2"/>
        </ni:PlotRendererGroup>
      </ni:PolarPlot.Renderer>
    </ni:PolarPlot>
  </ni:PolarGraph.Plots>
</ni:PolarGraph>
```

Yapılan ölçümler sonucunda elde edilen verilerin iki boyutlu çiziminin yapılması için gerekli olan polar çizim grafiğinin projeye eklenmesi ve gerekli görülen görsel tasarımların yapılması sağlanmıştır.

### Bağlantı görselleri

```
<StackPanel Grid.Column="2" Grid.Row="0" Margin="0" >
    <TextBox x:Name="IpAddress" Margin="5" HorizontalAlignment="Left" Height="20" Width="100"
    Text="169.254.101.1" ></TextBox>
    <Button VerticalAlignment="Top" HorizontalAlignment="Left" Height="25" Width="90" Margin="10"
    Content="BAĞLAN" Click="Button_Click_2"></Button>
</StackPanel>
```

Kontrol kartı ile ethernet protokolü ile bağlantı kurmamızı sağlayan internet protokolü (İP) adresi ile kullanıcıya istediği cihazdan veri alması imkanı tanıyan kısmının tasarımı yapılmıştır.

### Modelleme için veri girişi sağlanması ve ölçüm değerlerini sayısal olarak ifade edilmesi

```
<Grid Grid.Column="2" Grid.Row="2" VerticalAlignment="Top" Margin="152,10,152,0"
HorizontalAlignment="Center" Height="236" Width="456" Background="#FFD8ECE9" >
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="auto" ></ColumnDefinition>
        <ColumnDefinition Width="auto" ></ColumnDefinition>
        <ColumnDefinition Width="auto" ></ColumnDefinition>
        <ColumnDefinition Width="auto" ></ColumnDefinition>
        <ColumnDefinition Width="auto" ></ColumnDefinition>
        <ColumnDefinition Width="auto" ></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
        <RowDefinition Height="auto" ></RowDefinition>
        <RowDefinition Height="auto" ></RowDefinition>
        <RowDefinition Height="auto" ></RowDefinition>
        <RowDefinition Height="auto" ></RowDefinition>
        <RowDefinition Height="auto" ></RowDefinition>
        <RowDefinition Height="auto" ></RowDefinition>
        <RowDefinition Height="auto" ></RowDefinition>
    </Grid.RowDefinitions>
    <Label Content="İlk ölçüm sonuçları" HorizontalAlignment="Center" Grid.ColumnSpan="3"
    FontWeight="Bold" ></Label>
    <Label Content="İkinci ölçüm sonuçları" HorizontalAlignment="Center" Grid.Column="3"
    Grid.ColumnSpan="6" FontWeight="Bold" ></Label>
    <Label Content="Left" Grid.Row="1" ></Label>
    <TextBox x:Name="txtSensorLeft" Grid.Row="1" Grid.Column="1" Margin="5" Height="20"
    Width="100" HorizontalAlignment="Left" Text="20" ></TextBox>
    <Label Grid.Row="1" Grid.Column="2" Content="metre" ></Label>
    <Label Grid.Row="2" Content="Center" ></Label>
    <TextBox x:Name="txtSensorCenter" Grid.Row="2" Grid.Column="1" Margin="5" Height="20"
    Width="100" HorizontalAlignment="Left" Text="6" ></TextBox>
```

```

<Label Grid.Row="2" Grid.Column="2" Content="metre"></Label>
<Label Grid.Row="3" Content="Right"></Label>
<TextBox x:Name="txtSensorRight" Grid.Row="3" Grid.Column="1" Margin="5" Height="20"
Width="100" HorizontalAlignment="Left" Text="5,88"></TextBox>
<Label Grid.Row="3" Grid.Column="2" Content="metre"></Label>
<Label Content="Left" Grid.Row="1" Grid.Column="3" Margin="40,0,0,0"></Label>
<TextBox x:Name="txtSensorLeft2" Grid.Row="1" Grid.Column="4" Margin="5" Height="20"
Width="100" HorizontalAlignment="Left" Text="5,88" ></TextBox>
<Label Grid.Row="1" Grid.Column="5" Content="metre"></Label>
<Label Grid.Row="2" Grid.Column="3" Content="Center" Margin="40,0,0,0"></Label>
<TextBox x:Name="txtSensorCenter2" Grid.Row="2" Grid.Column="4" Margin="5" Height="20"
Width="100" HorizontalAlignment="Left" Text="6"></TextBox>
<Label Grid.Row="2" Grid.Column="5" Content="metre"></Label>
<Label Grid.Row="3" Grid.Column="3" Content="Right" Margin="40,0,0,0"></Label>
<TextBox x:Name="txtSensorRight2" Grid.Row="3" Grid.Column="4" Margin="5" Height="20"
Width="100" HorizontalAlignment="Left" Text="20"></TextBox>
<Label Grid.Row="3" Grid.Column="5" Content="metre"></Label>
<Label Grid.Row="4" Content="Sonuç : "></Label>
<TextBox x:Name="txtDistance" Grid.Row="4" Grid.Column="1" Margin="5" Height="20"
Width="100" HorizontalAlignment="Left"></TextBox>
<Label Grid.Row="4" Grid.Column="2" Content="metre"></Label>
<TextBox x:Name="txtDegree" Grid.Row="5" Grid.Column="1" HorizontalAlignment="Center"
Height="20" Width="100" ></TextBox>
<Label Grid.Row="5" Grid.Column="2" HorizontalAlignment="Left" Content="derece"></Label>
<Label Grid.Row="4" Grid.Column="3" Content="Sonuç : " Margin="40,0,0,0"></Label>
<TextBox x:Name="txtDistance2" Grid.Row="4" Grid.Column="4" Margin="5" Height="20"
Width="100" HorizontalAlignment="Left"></TextBox>
<Label Grid.Row="4" Grid.Column="5" Content="metre"></Label>
<TextBox x:Name="txtDegree2" Grid.Row="5" Grid.Column="4" HorizontalAlignment="Center"
Height="20" Width="100" ></TextBox>
<Label Grid.Row="5" Grid.Column="5" HorizontalAlignment="Right" Content="derece"></Label>
<Button x:Name="btnCalculate25cm" Grid.Column="1" Grid.Row="5" Content="Calculate25cm"
Height="30" Visibility="Hidden" Width="100" Grid.ColumnSpan="2" HorizontalAlignment="Center"
Click="btnCalculate_Click" ></Button>
<Label Content="Hız" Grid.Row="6" Grid.Column="1" HorizontalAlignment="Right"
Grid.ColumnSpan="2" Margin="0,0,23,0" />
<TextBox x:Name="txtVelocity" Grid.Row="6" Grid.Column="2" Margin="32,5,0,5" Height="20"
Width="100" HorizontalAlignment="Left" Grid.ColumnSpan="2"></TextBox>
<Label Grid.Row="6" Grid.Column="4" Content="km/s" Grid.ColumnSpan="2"
Margin="5,0,83,0"></Label>
<Button x:Name="btnCalculateOnSimulationMode" Grid.Column="1" Grid.Row="7"
Content="CalculateOnSimulationMode" Height="30" Width="160" Grid.ColumnSpan="4"
HorizontalAlignment="Center" Click="btnCalculateOnSensor_Click" Margin="107,0,91,0" ></Button>
</Grid>
</Grid>
</Window>

```

Tasarım penceresi içerisinde temel ızgara yapısının içerisine 2. satır ve 2 sütuna denk gelecek şekilde 7 sütun 8 satırdan oluşan yeni bir ızgara yapısı oluşturulmuş ve içerisine gerekli olan elementler eklenerek çalışma için ihtiyaç duyulan tasarım ekranları oluşturulmuştur.

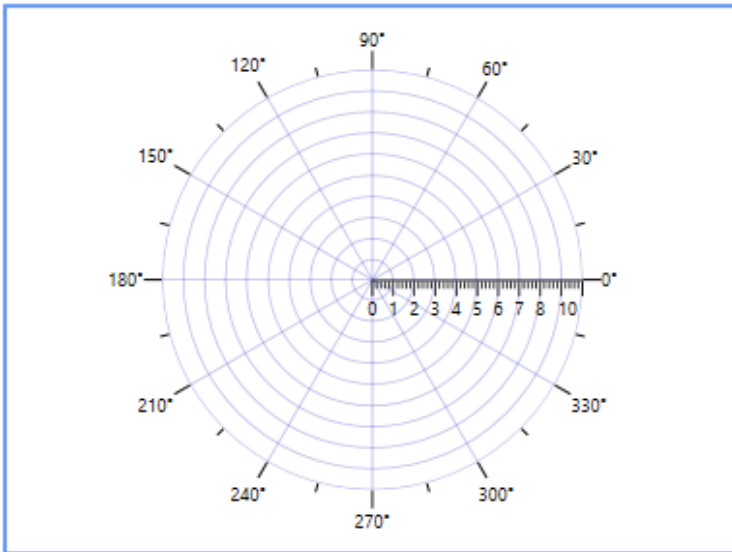


#### 4.1.2. Windows Forms uygulaması

Windows form arayüz tasarımı için WPF ile kıyaslanır ise tasarım yapmanın daha kolay olduğu bir platformdur. Tasarımda araç kutusu içerisinde yer alan elementler sürükleyip bırakılarak tasarımının yapılması mümkündür. Ancak WPF arayüz tasarımı için yeni ve daha gelişmiş bir programdır.[13] Bu çalışmada tasarım için ilk etapta windows form kullanılmıştır. Ancak çalışma esnasında yeni öğrenilen WPF uygulamasıyla yapılan tasarımların daha gelişmiş olması, özellikle grafik çizim işlemlerinde daha başarılı sonuçlar vermesi çalışmanın WPF uygulamasına taşınmasına ve geliştirmelerin WPF uygulaması üzerinde devam etmesine karar verilmesini sağlamıştır. Ayrıca çalışmaya daha sonra eklenen polar gösterim grafiğinin sadece WPF platformunda çalışıyor olması çalışmanın WPF taşınmasını gerekliliğini ortaya koymuştur.

#### 4.1.3. NI Measurement Studio kullanımı

WPF uygulamalarında araç kutusunda yer alan elementler dışınışında ihtiyaç duyduğumuz araç kutularını çalışmamıza referans olarak ekleyerek mevcut elementlerin dışında daha gelişmiş elementler bulunan araç kutuları üçüncü parti yazılım bileşenleri vasıtası ile kullanılabilir. Measurement Studio, National Instruments firmasının tasarlamış olduğu bir üçüncü parti yazılım bileşeni olup bu çalışmada faydalı olacağı tespit edilip çalışmaya eklenmiştir.



Resim 4.6. Polar grafik.

### Polar diyagram

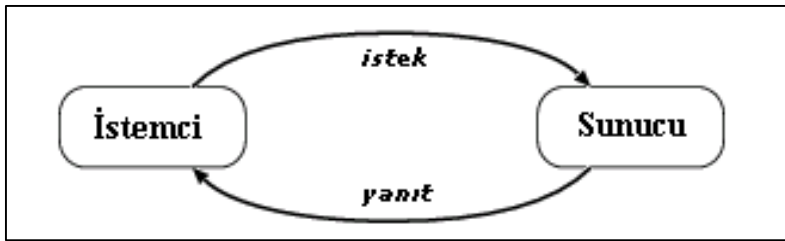
Kompleks ya da nokta verileri iki boyutlu olarak görüntülememizi sağlayan elemente verilen isimdir. Nokta verilerin grafik üzerinde çizimi için polar koordinatlar ( $(\theta, r)$ ) olarak ifade edilir ise  $\theta$  ölçülen açı değeri olarak,  $r$  ise sıfır noktasına olan uzaklık olarak ifade edilmektedir. Kompleks veriler için koordinatlar (faz, genlik) olarak ifade edilir. Faz açığı ifade ederken genlik ise sıfır noktasına olan mesafe ile ifade edilmektedir. Polar grafik aynı anda birden fazla nokta için çizim yapabilme imkanı sağlamaktadır. [14]

### **4.3. Bilgisayara Aktarılan Verilerin VS ile Toplanıp İşlenmesi**

İşlemci tarafından seri haberleşme protokolü ile XPORT isimli cihaza gönderilen verilen ethernet protokolüne dönüştürülüp bilgisayara gönderilmektedir. Bilgisayar VS yazılımı içerisinde tanımlanan transmisyon kontrol protokolü (TCP) ile verileri almaktadır.

### Transmisyon kontrol protokolü

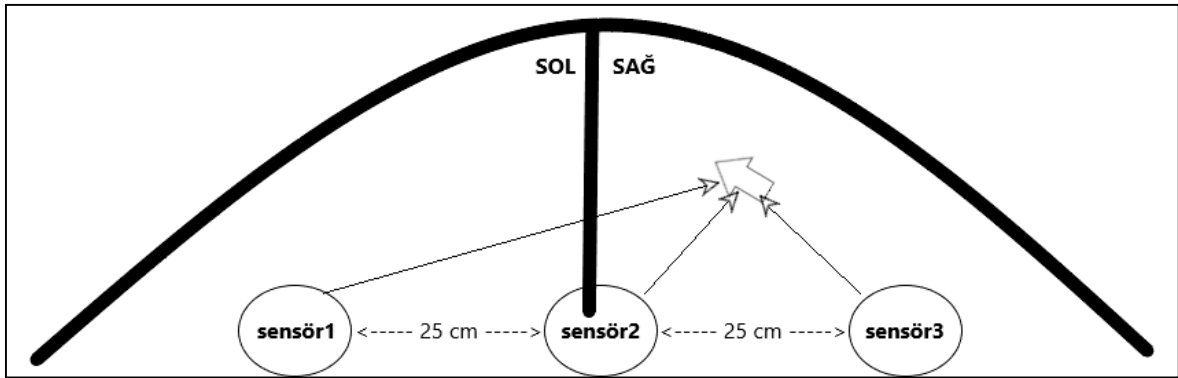
Çok yaygın kullanım ağına sahip olan bir haberleşme protokolüdür. Noktadan noktaya bağlantı sağlanan bu haberleşme protokolünde verinin nasıl paketleneyeceği, adresleneceği, gönderileceği, nereye yönlendirileceği ve alıcı kısımdan nasıl alınacağı sağlanmaktadır [15]. Bu haberleşme sistemi sunucu ve ona bağlı istemcileri olduğu bir mimari ile tasarlanmaktadır. Bu çalışmada sunucu tarafı kontrol kartı olup bilgisayar yazılımının olduğu kısım istemci konumunda yer almaktadır.



Şekil 4.1. Haberleşme akışı.

### 4.3.1. Açı, mesafe, hız hesaplanması

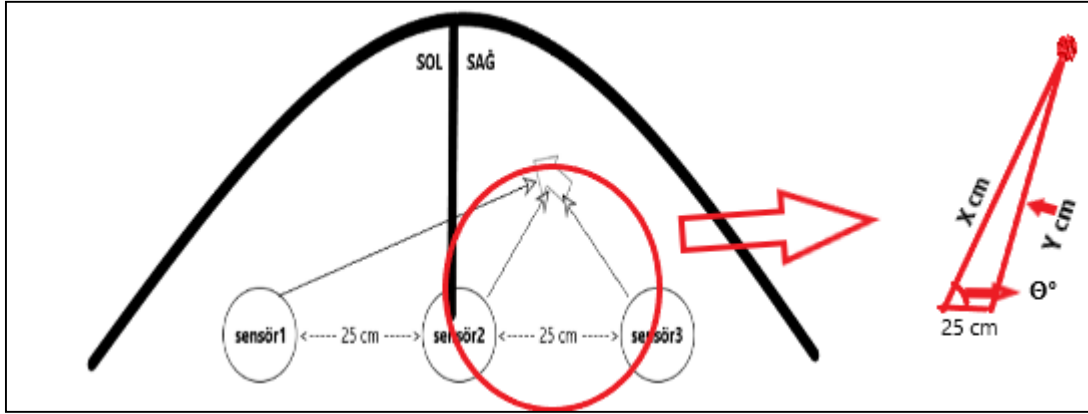
Bilgisayar tarafından toplanan üç ayrı sensörden alınan mesafe bilgisi içeren verilerin gerekli algoritmalarda geçirilerek polar diyagram grafiği üzerinde gösterilmesi ve hız hesaplamasının yapılması için matematiksel formüllerden ve kodlamalardan yararlanılmıştır. Bu çalışmada sıfır noktasında yer alan sensöre göre cisimlerin uzaklık tespiti yapılmaktadır. Üç sensörden gelen veriler ilk etapta kıyasalanarak cisimin merkezde bulunan sensöre göre hangi yöne denk geldiğinin tespit edilmesi işlemi yapılmaktadır.



Şekil 4.2. Tespit için gerekli olan trigonometrik denkleminin oluşturulmasında kullanılan üçgenlerin elde edilmesi.

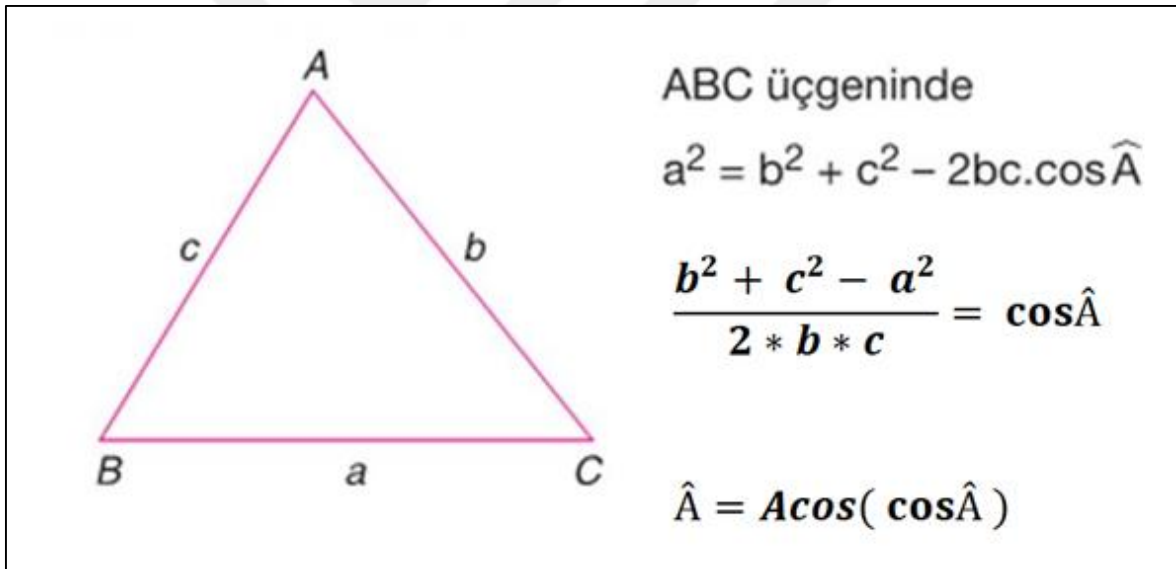
Algoritma ölçüm sonuçlarının gelmesiyle birlikte inç türünden gelen hesaplamaları cm çevirmektedir. Ardında 3 sensörden gelen veriler içerisinde sensör1 ve sensör3 den gelen ölçüm sonuçlarını kıyaslamakta ve tespit edilmeye çalışılan cismin sıfır noktasında bulunan sensöre göre hangi yönde olduğunu tespit etmektedir. Ölçüm sonuçlarına göre sensör1 ile gelen değerler eğer sensör3 den elde edilen değerlerden daha yakın bir mesafeyi ifade ediyor ise cisim sol tarafta daha uzak bir mesafeyi ifade ediyor ise bu cismin sol tarafta olduğunu göstermektedir.

Cismin hangi sensöre daha yakın olduğu tespit edildiğinde hangi yönden geldiğinde aynı anda tespit edilmiş olmuştur. Sonrasında iki sensör arasındaki mesafe zaten bilindiği için Sıfır noktasına olan uzaklık ve kenarlardan bulunan hangi sensöre daha yakın ise ona olan uzaklık bilgisi ile üç kenar uzunluğu bilinen bir üçgen elde edilmektedir.



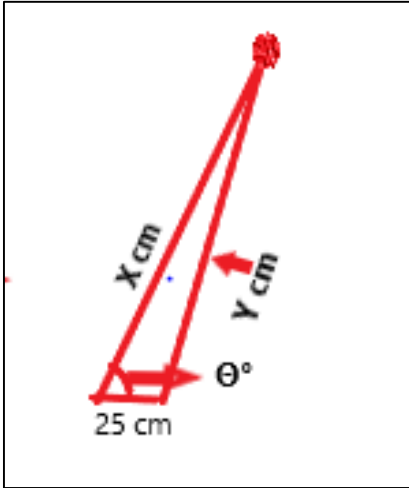
Şekil 4.3. Gerekli üçgenin çıkarımı

Yukarıdaki şekilde ifade edildiği gibi sensörlerden gelen bilgiler ile 3 kenar uzunluğu bilinen üçgenin  $\square$  açısı aşağıda belirtilen kosinüs teoremi kullanılarak hesaplanır ise,



Şekil 4.4. Kullanılan trigonometrik yöntem

$\square$  açısı kosinüs teoremindeki denklem vasıtası ile hesaplanabileceği yukarıdaki eşitlik vasıtası ile gösterilmiştir. Örnek ölçüm sonuçları ile bu hesaplama ifade edilecek olursa aşağıda belirtilen eşitlik ortaya çıkmaktadır. Acos ifadesi kosinüs fonksiyonunun tersini etmektedir.



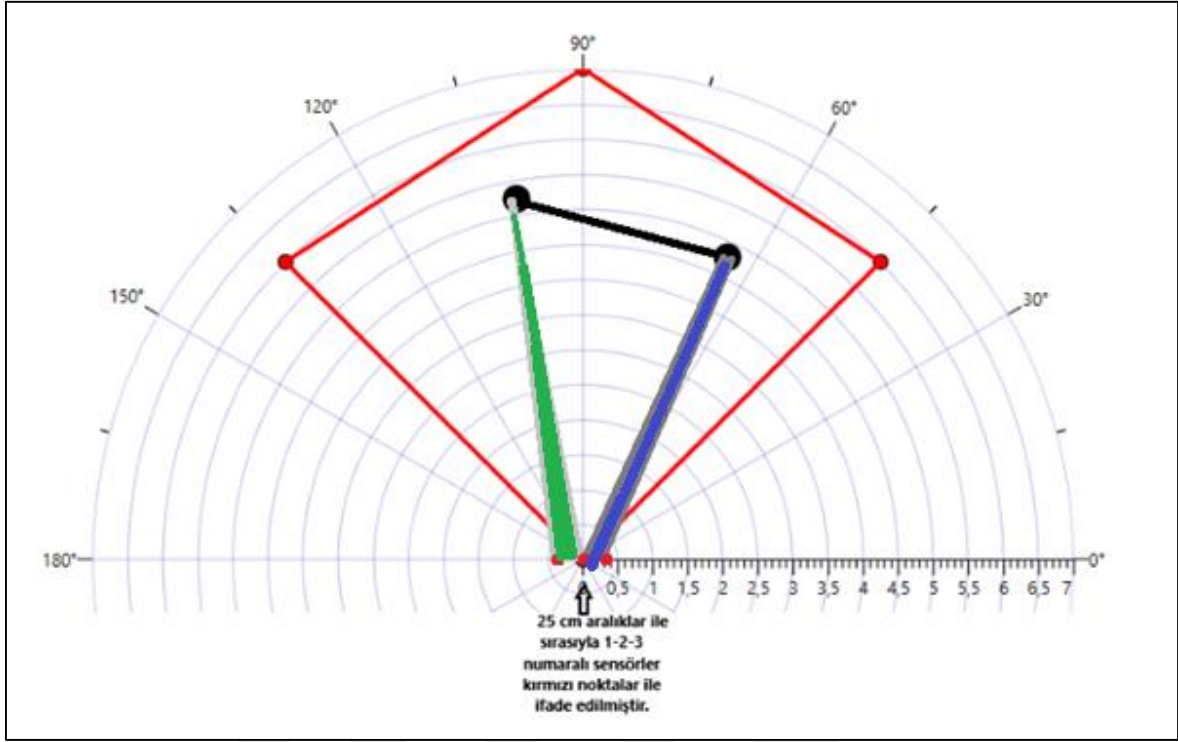
Şekil 4.5. Üçgen üzerinden gerekli olan açının elde edilmesi

$$T = \text{Acos} \left( \frac{X^2 + 25^2 - Y^2}{2 * X * 25} \right) \quad (4.1)$$

Eş. 4.1 ile  $\square$  açısı elde edilmiş olmaktadır. Bu hesaplanmalar sonucunda cismin sıfır noktasına olan uzaklık merkezde bulunan sensörden gelen veriler ile elde edilip açı ise yukarıdaki verilen eşitliklerin C# algoritmalarında hesaplanması ile elde edilmiş olmaktadır.

#### Açı değerinin hesaplanması

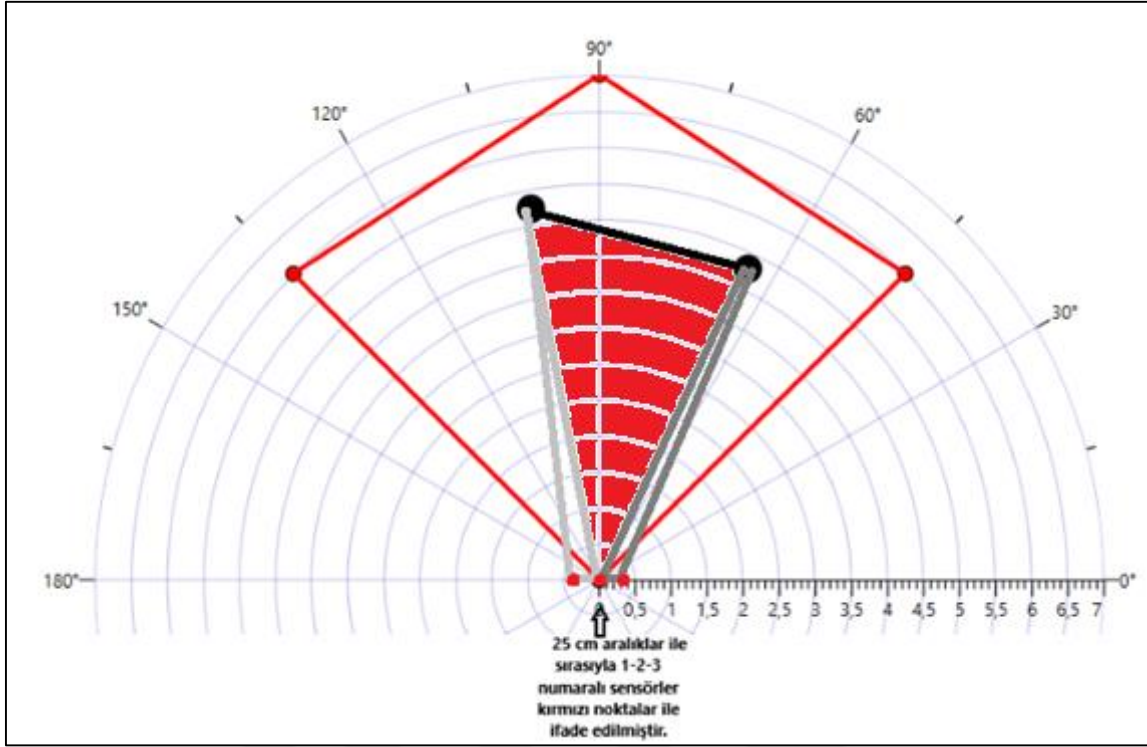
Bu çalışmada 3 sensörden alınan veriler ile bütün kenar uzunlukları bilinen aşağıdaki şekil 4.6. da belirtildiği gibi 2 adet üçgen elde edilmektedir. Polar diyagram üzerinde noktasal olarak ifade edilebilmesi için ihtiyaç duyulan açı değeri bu üçgenler üzerinde yukarıda belirtilen hesaplamaların yapılması ile mümkün olmaktadır. Sıfır noktasına olan uzaklık bilgisi merkezde bulunan sensörden zaten geldiği için 50 ms aralıklı iki ölçüm için uzaklık ve açı bilgileri elde edilmiş olduğu için grafik üzerinde nokta olarak tespit edilebilmesi mümkün olmaktadır.



Şekil 4.6. İki ölçüm sonucunda oluşan üçgenler

#### Mesafe bilgisinin hesaplanması

Ölçüm sonuçları sonunda elde edilen 2 üçgenin yanı sıra orta kısımdayukarıdaki şekilde de görülebileceği gibi bir adet üçgen daha bulunan sensörden gelen iki ölçüm değeri ile elde edilebilmektedir. Hesaplama için ihtiyaç duyulan açı değeri ise sağ ve sol tarafta bulunan üçgenler vasıtası ile hesaplanmıştır. Böylelikle orta bölgede oluşan bu üçgen vasıtası ile tespit edilen cismin 50mS içerisinde ne kadar yol aldığını hesaplamak mümkün olmaktadır.



Şekil 4.7. Alınan mesafenin hesaplanmasına imkan tanıyan üçgen

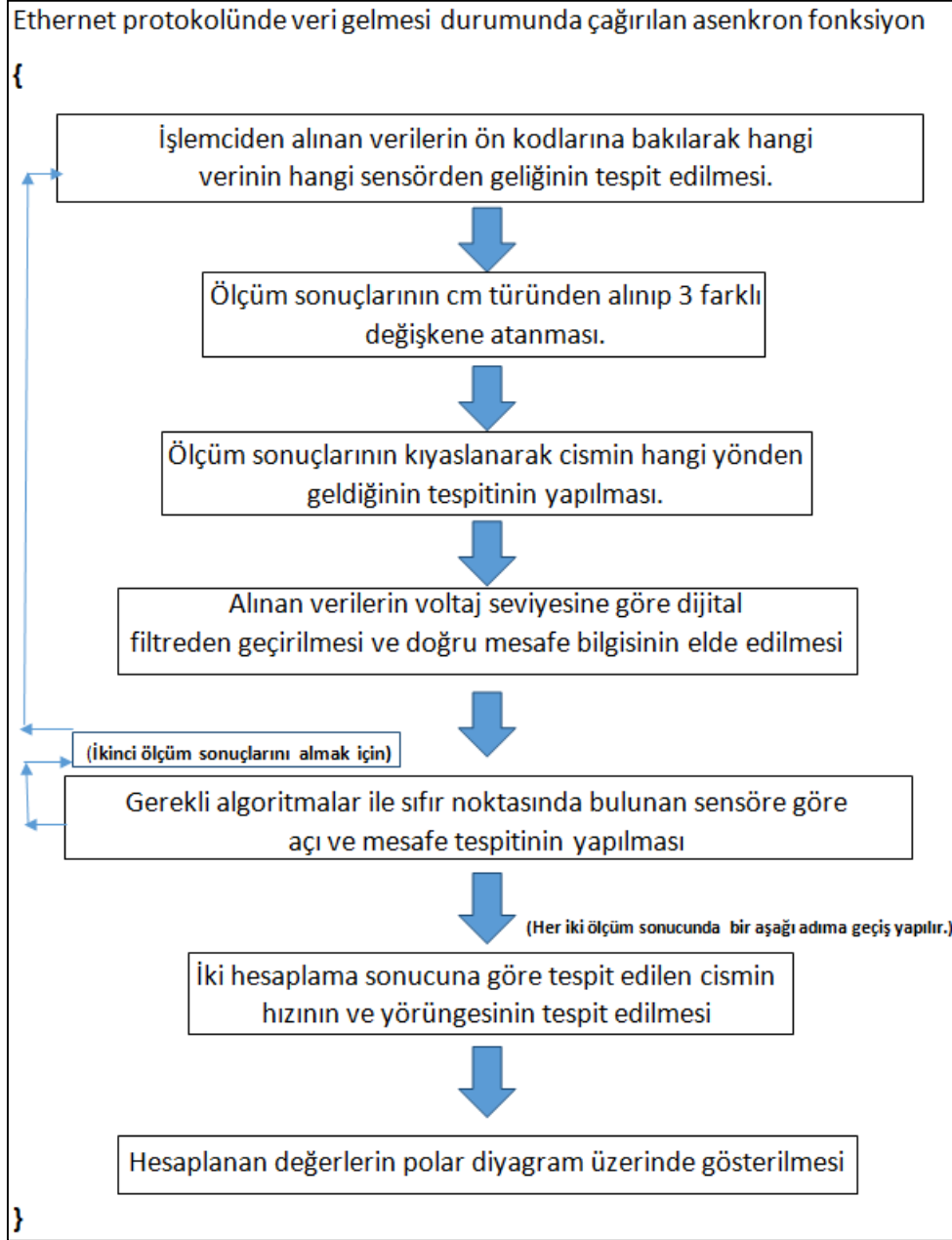
#### Hız değerinin hesaplanması

Mesafe bilgisinin elde edilmiş olması ile hız bilgisinin de hesaplanması mümkün olmuştur. Bu tespit edilmiş olan mesafeyi alma süresi iki ölçüm arasında geçen 50ms'lik süre olduğu için aşağıdaki eşitlik ile cismin hızı tespit edilmiş olur.

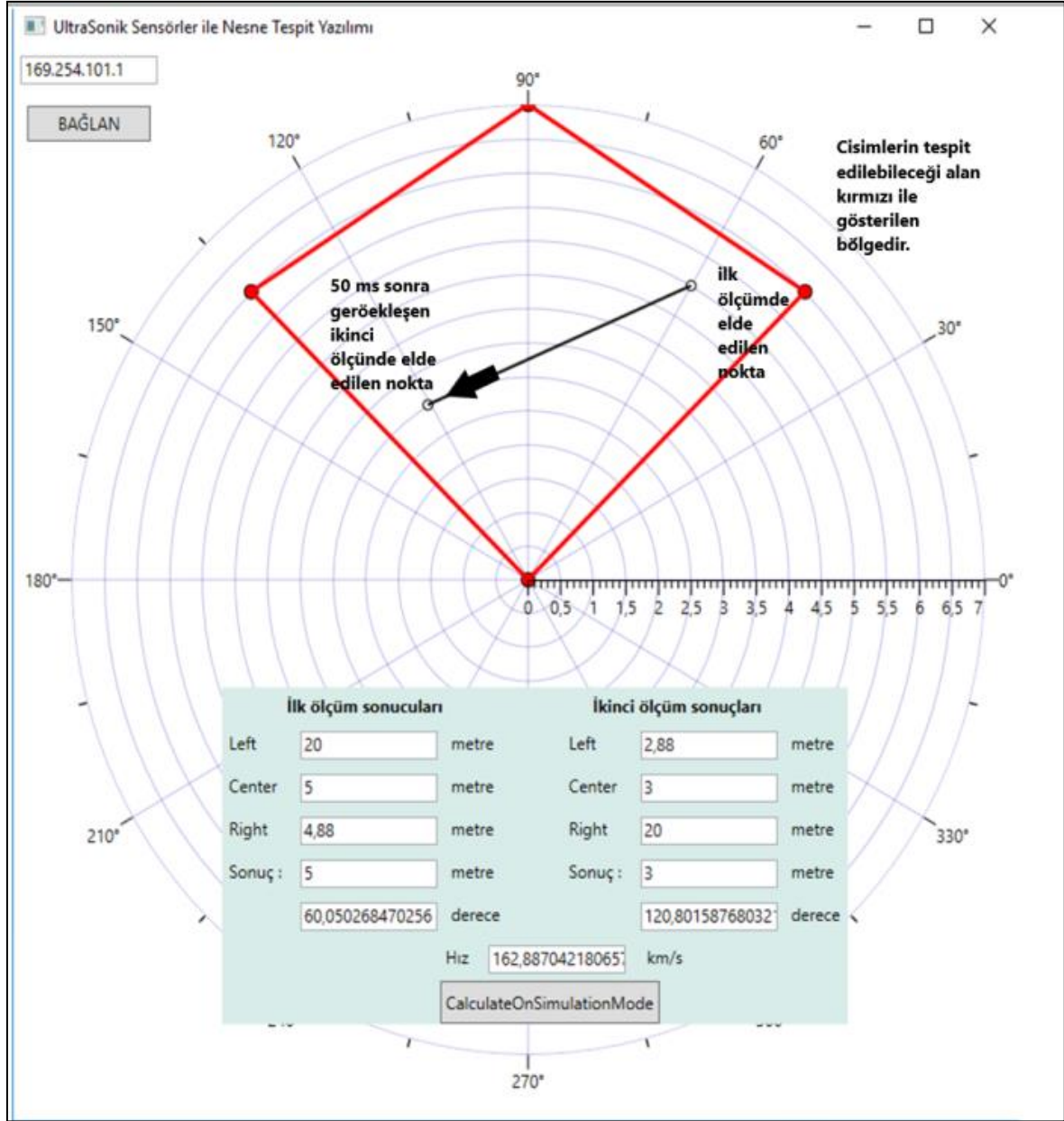
$$\text{Hız} = \text{Alınan Yol} \times \text{Geçen Süre}$$

(4.2)

Çizelge 4.1. Arayüz yazılımı kod akış çizelgesi







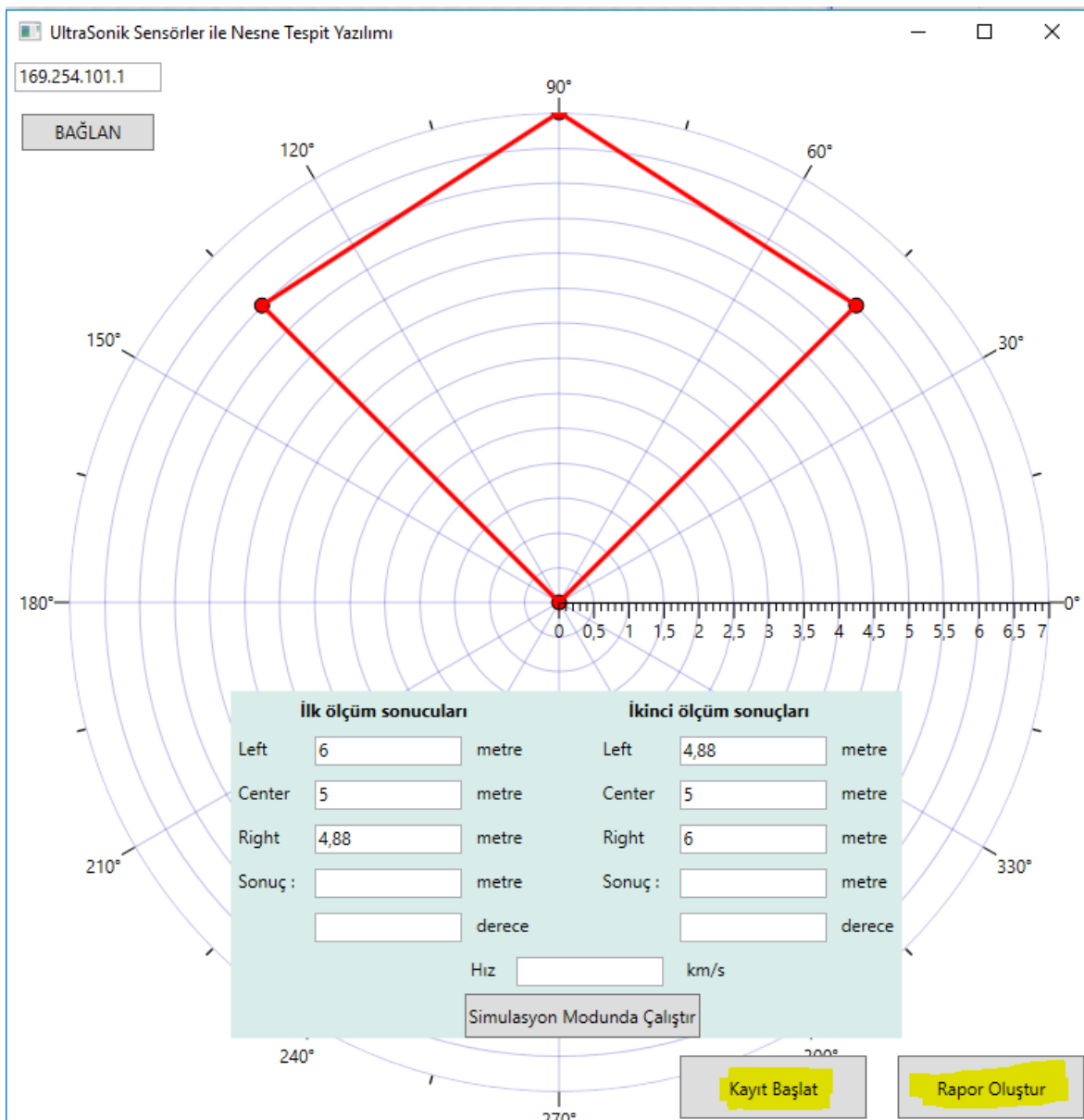
Resim 4.7. Kullanıcı arayüzü tasarımının görüntüsü ve ölçüm sonuçları



## 5. VERİ TABANI VE RAPOR OLUŞTURMA

Geliştirilen yazılım sistemin gereksinimlerini karşılamak, elde edilen verilerin kaybedilmesini engellemek ve ölçüm sonuçlarıyla yapılacak analizleri kolaylaştırmak için veri tabanına kayıt yapma imkanı sağlamaktadır.

Kayıt başlat ve rapor oluştur seçenekleri yazılımda sağ al köşeye yerleştirilmiştir. İhtiyaç duyulduğunda kullanıcı bu seçenekler seçilerek operatör tarafından kullanılabilir.



Resim 5.1. Arayüz üzerinde kullanım kayıt ve rapor oluşturma seçenekleri

### 5.1. Toplanan Verilerin Veri Tabanında Saklanması

Çalışmanın önemli ihtiyaçlarından biri olarak alınan ölçüm sonuçlarının istenilen yer ver zamanda kullanılmak üzere anlık olarak veri kaybı yaşamaksızın kaydedilmesi gerekliliği tespit edilmiştir.

Kullanım alanlarında saha çalışması esnasında ölçümlerin alındığı göz önünde bulundurulur ise yapılan testlerin tekrarının zor ve maliyetli olması kullanıcının verileri kaybetmemesi gerektiğini göstermektedir. Verilerin kaydedilmesi işleminin kullanıcılar tarafından bırakılması yanlışlık yapılması ve ya veri kaçırılması ihtimali artıracak ayrıca sürekli ölçüm alındığı varsayılır ise iş yükü ve zaman kaybı yaratacaktır.

Bu problem çerçevesinde SQL Server 2008 yazılımı üzerinde ihtiyaç duyulan verileri içeren tablo oluşturulmuştur. Bu tablo üzerinde kullanıcı arayüzünden gelen veriler kullanıcının ölçüm alanından o andan sonra oluşacak değerleri kaydetmek istemesi durumunda kaydı başlat seçeneğini seçmesi ile kayıt altına alınmaya başlayacaktır.

Ölçüm sonuçları hız bilgisi , merkez noktasına göre uzaklık bilgisi ve kayıt edildiği tarih bilgisiyle birlikte veri tabanına kaydedilmekte ve istenildiği zaman analiz imkanı tanınmış olmaktadır. Ayrıca bu bilgiler ihtiyaç duyulması durumunda yazdırılabilmekte ve analiz için görüntülenebilmektedir.

dbo.Ölçüm Sonuçları			
MUHAMMEDCALIS2...Ölçüm Sonuçları		MUHAMMEDCALIS2...Ölçüm Sonuçları	
	Uzaklık	Hız	Kayıt Tarihi
	3,5 m	120 m/s	2019-05-02 00:...
	6,1 m	165 m/s	2019-05-01 00:...
	4,9 m	220 m/s	2019-03-12 00:...
▶*	NULL	NULL	NULL

Resim 5.2. SQL server veri tabanı tablosu kayıt örnekleri

## 5.2. Otomatik Rapor Oluşturulması

Ölçüm işleminin sonuçlanmasının ardından tespit edilmiş olan cismin ölçüm sonuçları ile kullanıcı arayüzü vasıtası ile rapor oluşturulabilmesi için çalışma yapılmıştır. Bu çalışma sonucunda oluşturulacak rapora alınan ölçümün polar grafik üzerinde gösterimi ve diğer ihtiyaç duyulan özellikler otomatik olarak eklenmekte ve rapor yazdırmaya hazır bir şekilde kullanıcıya sunulmaktadır.



Resim 5.3. Oluşturulan rapor sayfasının içeriği

Hazırlanmış olan resim 5.3 ile gösterilen rapor eğer ihtiyaç duyulur ise ekstra bir işleme ihtiyaç duymadan raporun oluşturulduğu arayüz vasıtası ile yazdırılıp alınabilmektedir.



## 6. TARTIŞMA

Daha önce yapılan çalışmalar değerlendirildiğinde ultrasonik sensörler ile yapılmış bir çalışma bulunmamaktadır. Basınç sensörleri bu tür çalışmalar kullanılan başlıca sensör türü olarak dikkat çekmektedir. Basınç sensörleri ile alınan ölçüm sonuçları kullanıcıya sadece bölgesel olarak cismin nereden geçtiğini sunabilmektedir. Hız ve yörünge gibi bilgiler elde edilememektedir. Ayrıca ölçüm sonuçlarının alınabilmesi için tespit edilen cismin hareketi esnasında süpersonik bir patlama gerçekleşecek hıza ulaşması zorunludur. Aksi takdirde tespit yapılamamaktadır. Alternatif olarak kullanılan mikrofon dizini şeklindeki sistemlerde aynı sorun bulunmakta yani yüksek basınç ihtiyacından dolayı ses ötesi hıza ulaşmış cisimler için tespit yapılabilmektedir.

Yüksek maliyetli ve yüksek elektromanyetik gürültüye sahip RF sistemleri problemin çözümünde kullanımı mantıklı olmayan sistemler olarak analiz edilmiştir. Optik sensör sistemleri ise ortam şartlarından çok fazla etkilenmeleri , hedeflenen cismin fiziksel özelliklerinden yine aynı şekilde etkilenmelerinden dolayı verimli sonuçlar almaktan uzak sensörler olarak değerlendirilmektedir.

Ultrasonik sensörler basınç sensörlerinin ulaştığı tespit hızına ulaşmamaktadır. Ancak daha kesin ölçüm sonuçları ve daha fazla bilgi elde edilmesine imkan tanımaktadırlar. Ayrıca maliyet olarak da daha düşük maliyetlere sahip sistemlerdir. Ayrıca ses ötesi hıza ulaşmamış cisimlerin tespiti için kullanımları mümkündür. Bu açıdan mevcut sistemlere alternatif olarak dikkat çekmektedirler. Menzil olarak değerlendirildiklerinde ise benzer sonuçlar elde edilmektedir.





## 6. SONUÇ VE ÖNERİLER

Yapılan bu çalışmada donanım, donanım yazılımı, kullanıcı arayüzü yazılımı ve veri tabanı ile birlikte problemin çözümünde ihtiyaç olacak bütün konular üzerine çalışmalar yapılmış ve gerekli olan bağlantılar ile birlikte bütün alt yapı kurulmuştur. Yapılan bu çalışmaların sonucunda her biri farklı bir çalışma alanı olan bu konuların üzerine gidilir ve gerekirse konusunda uzman kişiler ile konu başlıklarına uygun iş bölümü yapılır ise burada kurulan alt yapı geliştirilerek bir çok alanda kullanılacak bir sistem elde edilmesi zor olmayacaktır.

Bu mevcut çalışmalar sonucunda elde edilen yapı modüler bir yapıda olup istenildiği durumda aynı yazılım farklı sensör çeşitleri ile ya da ihtiyaç duyulur ise farklı işlemciler ile kullanılabilir. Her bir sensör türünün daha iyi sonuçlar verdiği durumlar bulunmaktadır. Bu durum göz önüne alınarak sistemin farklı sensör tipleriyle birlikte çalışmasında bir problem olmadığı da değerlendirilerek birden çok sensörün birlikte kullanıldığı ya da yapılacak ölçümün ihtiyaçlarına göre sensör grubunun değiştirilebildiği şekilde çalışmanın geliştirilmesi ilerleyen aşamalarda değerlendirilecektir ve gelişimine katkı sağlayacak bir yöntem olarak değerlendirilmektedir.



## KAYNAKLAR

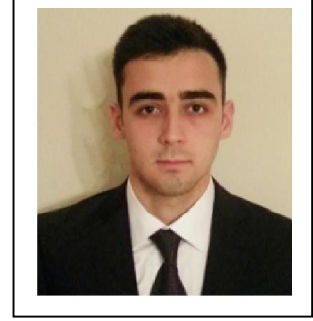
1. İnternet: Comparison between Ultrasonic Sensors and Optical Sensors, Web: <http://www.keyence.com/ss/products/sensor/sensorbasics/ultrasonic/comparison/>, adresinden 3 Mart 2019'da alınmıştır.
2. Skolnik, I.M. (1962). *Introduction to Radar Systems*, (Second edition). New York: Mc pGraw-Hill, 34
3. Robert, L. Kellogg Eldon and E. Mack Cathy, D. (2007), *Direction Finding Antennas and Systems*, (Fourth edition). New York: Mc Graw-Hill, 47 - 49
4. İnternet: Web: <http://koklea.tripod.com/ses.htm> adresinden 1 Nisan 2019'da alınmıştır.
5. İnternet: Web: [http://www.militarysystemstech.com/files/militarysystems/supplier\\_docs/MDSL%20ACOUSTIC%20MISS%20DISTANCE%20INDICATOR%20\(Scoring%20System\).pdf](http://www.militarysystemstech.com/files/militarysystems/supplier_docs/MDSL%20ACOUSTIC%20MISS%20DISTANCE%20INDICATOR%20(Scoring%20System).pdf) adresinden 1 Nisan 2019'da alınmıştır.
6. İnternet: Web: [https://www.airtarget.com/wpcontent/uploads/2018/11/MDI\\_133\\_techdata.pdf](https://www.airtarget.com/wpcontent/uploads/2018/11/MDI_133_techdata.pdf) adresinden 1 Nisan 2019'da alınmıştır.
7. İnternet: Web: [http://education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic\\_sensor/1.html](http://education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic_sensor/1.html) adresinden 1 Nisan 2019'da alınmıştır.
8. İnternet Web: [https://www.maxbotix.com/documents/LV-MaxSonar-EZ\\_Datasheet.pdf](https://www.maxbotix.com/documents/LV-MaxSonar-EZ_Datasheet.pdf) adresinden 11 Mart 2019'da alınmıştır.
9. Güven, B. ve Tuncay, U. (2007, Haziran) *Kontrol sistemlerinde kullanılan veri haberleşmesi teknolojileri*. 1. Haberleşme Teknolojileri Ve Uygulamaları Sempozyumu, İstanbul.
10. İnternet: Web: [http://education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic\\_sensor/1.html](http://education.rec.ri.cmu.edu/content/electronics/boe/ultrasonic_sensor/1.html) adresinden 15 Mayıs 2019'da alınmıştır.
11. İnternet: How to use multiple ultrasonic sensors , MaxBotix .Web: <https://www.maxbotix.com/tutorials1/031-using-multiple-ultrasonic-sensors.htm> adresinden 23 Nisan 2019'da alınmıştır.
12. İnternet: Web: <https://docs.microsoft.com/tr-tr/visualstudio/get-started/visual-studio-ide?view=vs-2017> adresinden 15 Nisan 2019'da alınmıştır.
13. İnternet: Web: [http://www.copybook.com/military/companies/airtargetswedenab/articles/a\\_irtargetsweden-acoustical-scoringequipment](http://www.copybook.com/military/companies/airtargetswedenab/articles/a_irtargetsweden-acoustical-scoringequipment) adresinden 1 Ocak 2019'da alınmıştır.
14. İnternet: Web: [zone.ni.com/reference/enXX/help/372636F01/mstudiowebhelp/html/wpfpolargraphplotting/](http://zone.ni.com/reference/enXX/help/372636F01/mstudiowebhelp/html/wpfpolargraphplotting/) adresinden 1 Nisan 2019'da alınmıştır.
15. Pranab, B. N. M. (2015). Mofiz Uddin TCP-IP Model in Data Communacation an Networking, *American Journal of Engineering Research*, 4 (10), 102-107



## ÖZGEÇMİŞ

### Kişisel Bilgiler

Soyadı, adı : ÇALIŞ, Muhammed  
Uyruğu : T.C.  
Doğum tarihi ve yeri : 28.02.1992, Zonguldak  
Medeni hali : Evli  
Telefon : 0 (538) 821 12 91  
e-mail : muhammed.calis@tubitak.gov.tr



### Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Yüksek lisans	Gazi Üniversitesi / Elektrik-Elektronik Müh.	Devam ediyor
Lisans	Gazi Üniversitesi / Elektrik-Elektronik Müh.	2014
Lise	Giresun Keşap Fen Lisesi	2009

### İş Deneyimi

Yıl	Yer	Görev
2016-Halen	TÜBİTAK-BİLGEM	Araştırmacı

### Yabancı Dil

İngilizce

### Yayımlar

-

### Hobiler

Yüzme, Basketbol, Futbol



## DİZİN

---

### **A**

Algoritma · 1, 2, 4, 5, 17, 18, 19,  
20, 25, 26, 32, 33, 35  
Analog · 9, 10, 12, 14, 15, 17, 18,  
19, 21  
Açı · 31, 34, 35, 36, 41  
Arayüz · 1, 7, 17, 19, 20, 21, 23,  
24, 25, 26, 28, 31, 39, 42

---

### **B**

Basınç · 2, 3, 41

---

### **C**

C# · 5, 24, 34  
C++ · 24

---

### **D**

Dalga · 2, 5, 9

---

### **E**

Eşitlik · 34, 35, 37

---

### **F**

Formül · 32

---

### **G**

Gömülü · 7, 20, 21

---

### **H**

Hız · 2, 3, 4, 5, 9, 12, 32, 37, 40,  
41  
Hata · 1, 24, 27

---

### **K**

Kaynaklar · 43, 44

---

### **M**

Mesafe · 1, 2, 4, 5, 9, 10, 11, 12,  
15, 16, 17, 19, 20, 32, 33, 35

---

### **Ö**

Özgeçmiş · 43

---

### **P**

Problem · 1, 41, 42

---

### **R**

Rapor · 40

---

### **S**

Sensör · 1, 2, 3, 4, 5, 6, 7, 9, 11,  
12, 13, 14, 15, 16, 17, 18, 19,  
20, 21, 33, 34, 35, 36, 41, 42

---

### **T**

Tespit · 1, 2, 3, 5, 9, 10, 15, 16,  
17, 19, 31, 33, 35, 36, 37, 41

---

### **U**

Ultrasonik · 1, 3, 5, 7, 9, 14, 15,  
27, 41

---

### **V**

Veri · 1, 4, 5, 6, 7, 10, 11, 12, 13,  
21, 38, 40

---

### **Y**

Yörünge · 1, 3, 4, 41



*GAZİ GELECEKTİR..*