

2209/A

ÜNİVERSİTE ÖĞRENCİLERİ ARAŞTIRMA PROJELERİ DESTEK  
PROGRAMI

SONUÇ RAPORU

PROJE BAŞLIĞI: **ARTTIRILABİLİR MENZİLLİ KONUMLANDIRMA  
SİSTEMİ**

PROJE YÜRÜTÜCÜSÜNÜN ADI: Ömer Faruk Çelik

DANIŞMANININ ADI: Doç. Dr. Ebru Karaköse

## GENEL BİLGİLER

<b>PROJENİN KONUSU</b>	Bu proje, sualtı araçlarının konumlarını tespit etmek için ses dalgalarını kullanan, GPS sinyallerinin sualtında etkisiz olduğu durumlarda güvenli, maliyeti az olan ve esnek olarak etki mesafesinin arttırılabileceği bir çözüm sunan bir konumlandırma sistemi geliştirmeyi amaçlamaktadır.
<b>PROJE YÜRÜTÜCÜSÜNÜN ADI</b>	Ömer Faruk Çelik
<b>DANIŞMANIN ADI</b>	Doç. Dr. Ebru Karaköse
<b>PROJE BAŞLANGIÇ VE BİTİŞ TARİHLERİ</b>	22/03/2024-22/06/2025

### 1. Giriş

Coğrafi keşifler, 15. yüzyılda başlamış ve 17. yüzyılın ortalarına kadar sürmüştür. Bu keşiflerin temel amacı, yeni ticaret yolları bulmaktır. Bu dönemde yalnızca yeni kara parçaları değil, aynı zamanda bilinmeyen okyanuslar da keşfedilmiştir [1]. İnsanlık, tarih boyunca bilinmeyene duyduğu bitmek bilmeyen bir merakla keşifler yapmıştır[1,2]. 15. yüzyılda başlayan keşif hareketleri sonucunda, yeryüzünün yaklaşık %29'unu oluşturan kara parçalarının büyük bir bölümü keşfedilmişken, gezegenimizin %71'ini kaplayan okyanusların yalnızca %5 ila %24'lük bir bölümü araştırılabilmektedir [1, 3]. Karaların büyük ölçüde keşfedilmesini sağlayan en önemli etkenlerden biri, modern teknolojiyle donatılmış yörünge uyduları ve Küresel Konumlandırma Sistemi'dir (GPS). GPS, kara yüzeyinin ayrıntılı bir şekilde haritalandırılmasını mümkün kılarken, okyanusların keşfi konusunda aynı başarı sağlanamamıştır. Bunun nedeni, GPS sinyallerinin yüzeyde etkili olmasına karşın, elektromanyetik dalgaların fiziksel özellikleri gereği su altında yayılmamasıdır [3, 4].

Elektromanyetik dalgaların alternatifi olarak mekanik bir dalga biçimi olan ve yayılmak için fiziksel bir ortama ihtiyaç duyan ses dalgalarının su altı keşiflerinde kullanımı 20. Yüzyılın başlarına dayanmaktadır [1, 5]. 1912 yılında Kanadalı bir mühendis olan Reginald Fessenden tarafından Boston limanında yaklaşık 3 kilometre mesafedeki buzdağını tespit etmek için ilk kez deneysel bir sistemde ses dalgaları kullanılmıştır [1, 5]. Su altı akustik sistemlerinin gelişimi, deniz suyunun iletkenlik özelliklerinin anlaşılmasıyla hız kazanmıştır [5]. Deniz suyunun derinliğe bağlı olarak değişen iletkenlik özellikleri, akustik sistemlerin tasarımında önemli bir rol oynamaktadır [6, 7].

1950'li yıllarında denizaltıların ve su altı keşiflerin artmasıyla tarihteki ilk uzun menzilli bir akustik sistem olan LBL (Long Baseline) sistemi geliştirildi [1, 3, 4, 8]. Bu sistem üçgenleme yöntemi kullanarak, üç ayrı konumu bilinen istasyondan ses dalgalarıyla yayın yapar [4]. LBL sisteminde deniz altında 3 ya da daha fazla transponder (Verici-Yanıtlayıcı) bulunur [1]. Su altı aracı (AUV), bu Verici-Yanıtlayıcıdan gelen sinyallerin yayılma süresini ölçerek kendi konumunu belirler [3]. LBL sistemi, geniş alanlarda yüksek hassasiyetle konumlandırma yapabilmesine rağmen, kurulumu zahmetlidir ve deniz tabanına sabit transponderler yerleştirilmesini gerektirir [1, 3, 4]. Bu sistemlerin

performansı, sığ sularda ses dalgalarının yayılım özelliklerinden önemli ölçüde etkilenir [9].

1970'li yıllarda, su altı araçları ve dalgıçlar için daha kısa menzilli ancak daha hızlı konumlandırma sağlayan SBL (Short Baseline) sistemi geliştirildi [1, 3]. Bu sistemlerde veri iletişimi için I2C ve SPI protokolleri kullanılmaya başlanmıştır [10]. SBL sisteminde, transponderler bir yüzey gemisine veya platforma monte edilir ve araçtan yayılan sinyallerin farklı transponderlere ulaşma süresi ölçülerek üçgenleme yapılır. LBL sistemine göre daha esnek ve hızlı olmasına rağmen, menzil açısından daha sınırlıdır ve sadece gemiye bağlı bir bölgede etkin şekilde çalışabilir [1, 3].

1980'lerde, ultra kısa menzilli ve dinamik konumlandırma sağlayan USBL (Ultra-Short Baseline) sistemi geliştirildi [1, 3]. USBL sisteminde, tek bir anten dizisi içinde bulunan transponderler kullanılarak su altı aracının yönü ve mesafesi belirlenir [11]. USBL, gemi tabanlı olup taşınabilirliği yüksek bir sistemdir ve dinamik deniz operasyonlarında büyük kolaylık sağlar [1]. Ancak, USBL'nin doğruluğu genellikle LBL kadar yüksek değildir ve ses dalgalarının su içindeki yayılma özelliklerine bağlı olarak hassasiyeti değişebilir [1, 3]. Birden fazla AUV'nin birlikte çalışması gereken durumlarda konumlandırma daha da zorlaşmakta ve bu amaçla ters USBL (iUSBL) gibi özel sinyal tasarımları önerilmektedir [12].

Günümüzde, su altı konumlandırma teknolojileri yalnızca akustik sistemlerle sınırlı değildir [13]. Modern sistemlerde OFDM (Orthogonal Frequency Division Multiplexing) tekniği kullanılarak veri iletim hızı ve güvenilirliği artırılmaktadır [14]. Ataletsel Navigasyon Sistemleri (INS), aracın kendi ivmeölçer ve jiroskop verilerini kullanarak konumunu hesaplar, ancak zamanla hata birikimi yaşar [3, 4]. Bu sistemlerin doğruluğunu artırmak için Benford yasası gibi istatistiksel yöntemler kullanılmaktadır [15, 16]. Ayrıca, 2024 yılında yayınlanan "An Underwater Passive Electric Field Positioning Method Based on Scalar Potential" adlı makalesinde önerilen yöntem gibi, su altı araçlarının doğal olarak oluşturduğu pasif elektrik alanlarını kullanan yenilikçi yöntemler de geliştirilmektedir. Bu yöntemde, sualtı araçlarının konumları, araçların korozyon ve anti-korozyon akımları tarafından oluşturulan elektrik alanların analizi altında tahmin edilir. Akustik yaklaşım ile karşılaştırıldığında, böyle bir yöntem akustik sinyal gecikmesi olmadan çalışabilir, başka bir alete ihtiyaç duymaz, pasif problem ve gizlilik sağlar, ancak araçtan araca doğrudan iletişim mümkün değildir [6].

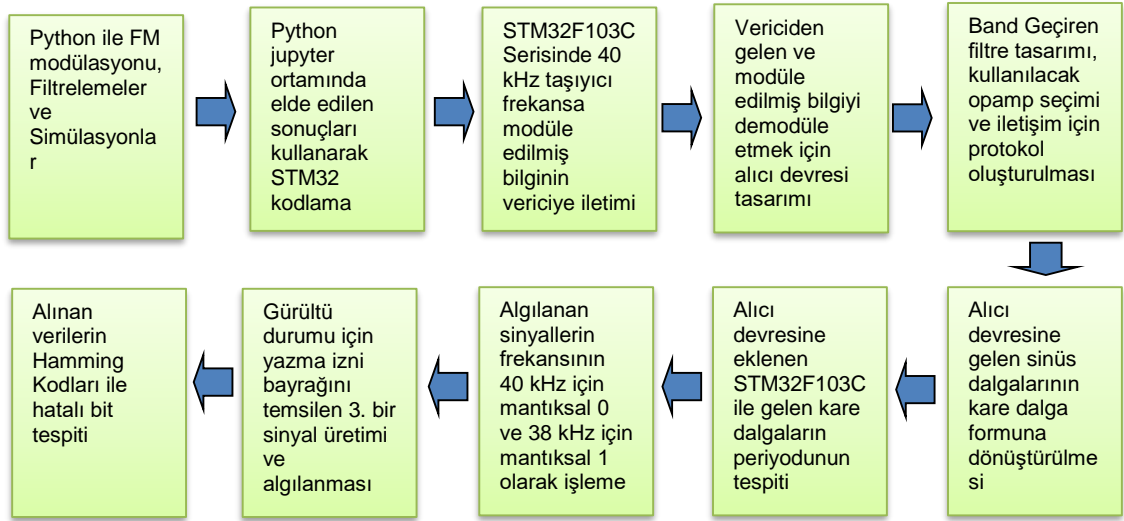
Teknoloji	Değişkenler	Yenileme Hızı	Mesafe	Çalışma Frekansı	Yöntem	Veri İletim Türü	Ölçüm Şekli
LBL (LongBaseline)	XYZ -Konum	0.1 - 1 Hz	1 – 10 Km	10 - 30 kHz	Ses Dalgaları	Sinyal Gecikme zamanı	2 Yollu Ölçüm
SBL(Sort-Baseline)	XYZ - Konum	0.5 - 5 Hz	1 - 5 Km	20 – 50 kHz	Ses Dalgaları	Sinyal Gecikme zamanı	2 Yollu Ölçüm
USBL (Ultra Sort-Baseline)	XYZ - Konum	1 – 10 Hz	1 – 5 Km	20 – 100 kHz	Ses Dalgaları	Sinyal Gecikme zamanı	2 Yollu Ölçüm

<b>Passive Electric Field Positioning Method</b>	XYZ - Konum	Pasif Olarak Ölçüm Yapar	1 Km	Veri Alma Yöntemine Bağlı Esnektir	Statik Elektrik	Su altında iletilen elektriği kullanır	Tek Yollu Ölçüm
<b>AMKS (Arttırılabilir Menzilli Konumlandırma Sistemi)</b>	XYZ – Konum XYZ – Hız XYZ - İvme	Esnektir	1 – 5 Km Menzil ve sınırsız olarak arttırılabilir	20 - 100 kHz	Ses Dalgaları Frekans Modülasyonu	Binary olarak her türlü veri iletimi sağlar	Tek Yollu Ölçüm

**Tablo 1.** Literatürde Bulunan Farklı Konumlandırma Sistemlerinin Kıyaslanması [3-8].

**Tablo-1’de** çeşitli su altı konumlandırma teknolojileri ile Arttırılabilir Menzilli Konumlandırma Sistemi karşılaştırılmıştır. Su altı konumlandırma sistemlerinde genellikle ses dalgaları kullanılır. Ses dalgasının frekansı arttıkça su altında daha hızlı sönmeye başlaması, önerilen Arttırılabilir Menzilli Konumlandırma Sistemi (AMKS) de dahil olmak üzere, kullanılan teknolojiler için en büyük dezavantajlardan biridir [8, 12]. Mevcut teknolojilerin (LBL, SBL, USBL) bir diğer eksikliği mesafelerinin sınırlı olmasıdır [1, 12]. Ayrıca LBL, SBL, USBL ve iUSBL gibi sistemler genellikle iki yönlü (gidiş-geliş) sinyal süresine dayalı ölçüm yaptıkları için, bu teknolojilerle araçlar arasında doğrudan veri iletimi genellikle mümkün değildir [1, 13]. Pasif Elektrik Alan Konumlandırma yöntemi ise tek yönlü ölçüm yapar ancak yaklaşık 1 Km gibi bir menzil sınırlamasına sahiptir [6]. AMKS, Frekans Modülasyonu (FM) tekniğini kullanarak [4] hem konumlandırma hem de tek yönlü veri iletimini mümkün kılmayı hedefler. Belli noktalara yerleştirilebilen sinyal tekrarlayıcılar (repeater) ile sinyalin etki mesafesinin arttırılabilmesi amaçlanmaktadır. AMKS, mevcut literatür teknolojilerinin ortaya çıkardığı mesafe ve veri aktarımı sorunlarıyla ilgilenmeyi amaçlamaktadır. Geçerli çalışmada, FM (Frekans Modülasyonu) işlemi için bir taşıyıcı sinyal olarak 40 kHz frekansında bir ses dalgası alındı. Verici modülünde, 20-40 kHz arasında çalışan 3 ultrasonik yayıcı uygulandı ve alıcı tarafında da aynı tip ultrasonik alıcılar tercih edildi [21]. Sistem aslında deneysel olarak test edilmeden önce, farklı sistem senaryolarını keşfetmek amacıyla Python programlama dili kullanılarak bir simülasyon gerçekleştirildi. Bu çalışmada, [17-19] numaralı Python ile birlikte C programlama dili de uygulandı.

Uzatılabilir Menzil Konumlandırma Sistemi için alıcı ve vericinin gerçek tasarımından önce, sistemin stabilitesi ve doğruluğunu değerlendirmek için 20 kHz’de ön veritabanlı (ses) testler yaptık. Bu simülasyonda modüle edilmemiş 20 kHz taşıyıcı frekansı seçildi; 20 kHz mantıksal 0, 15 kHz mantıksal 1 olarak alındı. Gürültü etkilerini minimuma indirmek amacıyla, yazma izni veren 18 kHz’lik bir dalga daha sisteme eklenmiştir. Çevre faktörlerine bağlı olarak gürültü seviyesi ses dalgaları ile veri iletimini zorlaştırmaktadır. Bu sebeple 3. olarak eklenen 18 kHz’lik dalga sistemin algılama başarısını %70’lere çıkarırken aynı oranda sistemin hızını düşürdüğü gözlemlenmiştir. Aynı şekilde Hamming Kodları gibi hata düzeltme kodlarının kullanımı da incelenmiştir. Sistem tasarımının temel işlem adımları **Şekil 1’de** verilmiştir.



**Şekil 1.** Sistem tasarım aşamaları

### 1.1. Python İle Frekans Modülasyonu ve Filtre Seçimi

Arttırılabilir Menzilli Konumlandırma Sisteminde başlangıç olarak yollanacak veri formatı, 11 bit uzunluğunda ikili sayı sisteminde test edilmiştir. Sistemde 3 tane verici kullanılmıştır.

#### Verici Kimliği

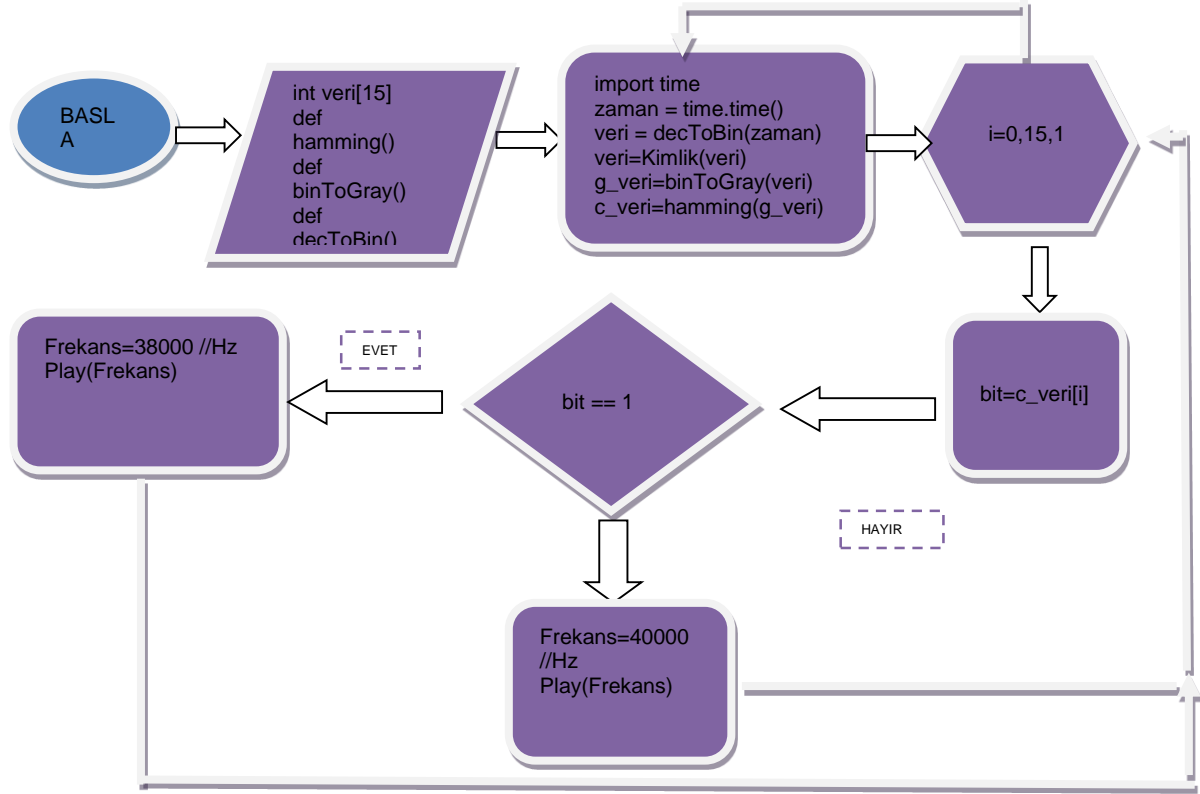
Sistemde kullanılan 3 verici (A, B ve C) arasında ayırım yapmak için 11 bitlik veri formatının 2 biti verici kimliği olarak ayarlanmıştır. A vericisi 00, B vericisi 01 ve C vericisi 10 ikili sayılarla temsil edilmiştir. Bu sayede, alıcı gelen sinyalin hangi vericiden geldiğini belirleyebilir.

#### Kodlama Aşamaları

1. Zamanlayıcı Başlatma ve Veri Dönüşümü: İlk aşamada, verici modül üzerindeki STM32F103C modeli [10] ile 1.95 ms hassasiyetinde bir zamanlayıcı başlatılmıştır. Bu zamanlayıcıdaki değerler saat olarak alınıp onluk tabandan ikili tabana 9 elemanlı bir dizi olarak dönüştürülmüştür.
2. Verici Kimliği Eklenmesi: STM32 serisi [11], sinyali yollayacağı vericiyi seçtikten sonra sinyalin hangi vericiye ait olduğunu göstermek için iki bitlik kimlik bitlerini dizinin sonuna ekleyerek 11 bitlik bir veriye dönüştürür.
3. Gray Koduna Dönüşüm: Binary kodlama sistemlerinde en anlamlı basamağın değişmesi hatayı olabildiğince arttırdığından, ikili sayı sistemi her bir bit değişimine aynı hatayı verecek olan Gray koduna dönüştürülmüştür.
4. Hamming Kodu ile Hata Düzeltme: Gray koduna dönüşen verimizin, vericiden yollanırken çevresel faktörlerden en az hatadan etkilenmesi için hata bulma konusunda oldukça yetenekli olan Hamming kodu seçilmiştir. Donanım sınırlamaları nedeniyle bu sistem için (15,11) Hamming kodları seçilmiştir. (15,11) Hamming kodları 11 bit veri ve 4 bit hata düzeltme içermektedir ve tek bitlik hataları düzeltebilir, iki bitlik hataları ise tespit edebilir.
5. Modülasyon ve Gönderim: Şekil 2'de, 11 bitlik verinin modülasyonunu yapan ve yapılan modülasyonu vericiye gönderen algoritmanın akış diyagramı verilmiştir.

## 1.2. Binary Kodu ve Gray Kodu Arasındaki Farklar ve Avantajlar

1. **Hata Azaltma:** Gray kodunda, iki ardışık sayının yalnızca bir biti değiştiğinden, sayısal sistemlerde bit değişiminden kaynaklanan hatalar minimize edilir. Binary kodda, birden fazla bit değişebilir ve bu da hata olasılığını artırır[16].
2. **Donanım Basitliği:** Gray kodu, dijital sistemlerde daha basit donanım ile kullanılabilir, çünkü bit değişimlerinden kaynaklanan hatalar daha azdır[16].
3. **Hassasiyet:** Sensör ve analog-dijital dönüştürücülerde Gray kodu kullanımı, hassas ölçümlerde daha güvenilir sonuçlar sağlar[16].



Şekil 2. Verici Kodu Akış Diyagramı

## 1.3. Jupyter Notebook Ortamında Yapılan Simülasyonlar ve STM32 Kodlama

Şekil 2'de görüldüğü üzere, vericiye milisaniye hassasiyetinde alınan saat bilgisinin 40 kHz'lik taşıyıcı sinyale modüle edilmesi sonucu; 1 için 38 kHz, 0 için ise 40 kHz frekanslı sinyaller yollanmıştır. Frekans modülasyonu (FM), iletilmek istenilen bilginin genliğiyle doğru orantılı olarak taşıyıcı dalganın ilgili zaman dilimindeki frekansının değişmesine denir [4]. FM'de,  $C(t)$  taşıyıcı dalgayı temsil ederken,  $F_c$  taşıyıcı dalganın frekansını,  $Q$  radyan cinsinden faz açısını ve  $2\pi k_F \int x(t)dt$  denklemi taşıyıcı frekansın ne kadar değişeceğini temsil etmektedir [4].

$$C(t) = A_c \cdot \cos(\omega_c \cdot t + Q) \quad (\text{Denklem 1})$$

$$\omega_c = 2 \cdot \pi \cdot F_c \quad (\text{Denklem 2})$$

$$Y(t) = A_c \cdot \cos\left(A_c \cdot \pi \cdot F_c \cdot t + 2 \cdot \pi \cdot k_F \cdot \int x(t) \cdot dt + 2 \cdot \pi \cdot k_F \cdot \int x(t) \cdot dt\right) \quad (\text{Denklem 3})$$

```

# Sabitler
carrier_freq = 15000 # Taşıyıcı frekansı 15 kHz
sampling_rate = 100000 # Örnekleme frekansı 100 kHz
duration = 0.001 # Sinyal süresi 10 ms
mod_index = 20.0 # Modülasyon indeksi

# Zaman dizisi oluşturma
t = np.arange(0, duration, 1/sampling_rate)

# Taşıyıcı sinyal
carrier = np.sin(2 * np.pi * carrier_freq * t)

# 16 bitlik dijital sinyal (örnek olarak rastgele bir sinyal)
digital_signal = np.random.randint(0, 2, len(t)) * 2 - 1 # -1 ve 1 arasında rastgele değerler

# Frekans modülasyonu
modulated_signal = np.sin(2 * np.pi * carrier_freq * t + mod_index * np.cumsum(digital_signal) / sampling_rate)

# Sinyalleri görselleştirme
plt.figure(figsize=(10, 8))

plt.subplot(3, 1, 1)
plt.plot(t, carrier)
plt.title("Taşıyıcı Sinyal (15 kHz)")
plt.xlabel("Zaman (s)")
plt.ylabel("Genlik")

plt.subplot(3, 1, 2)
plt.plot(t, digital_signal)
plt.title("Dijital Sinyal (16 Bit)")
plt.xlabel("Zaman (s)")
plt.ylabel("Genlik")

plt.subplot(3, 1, 3)
plt.plot(t, modulated_signal)
plt.title("Frekans Modüleli Sinyal (FM)")
plt.xlabel("Zaman (s)")
plt.ylabel("Genlik")

plt.tight_layout()
plt.show()

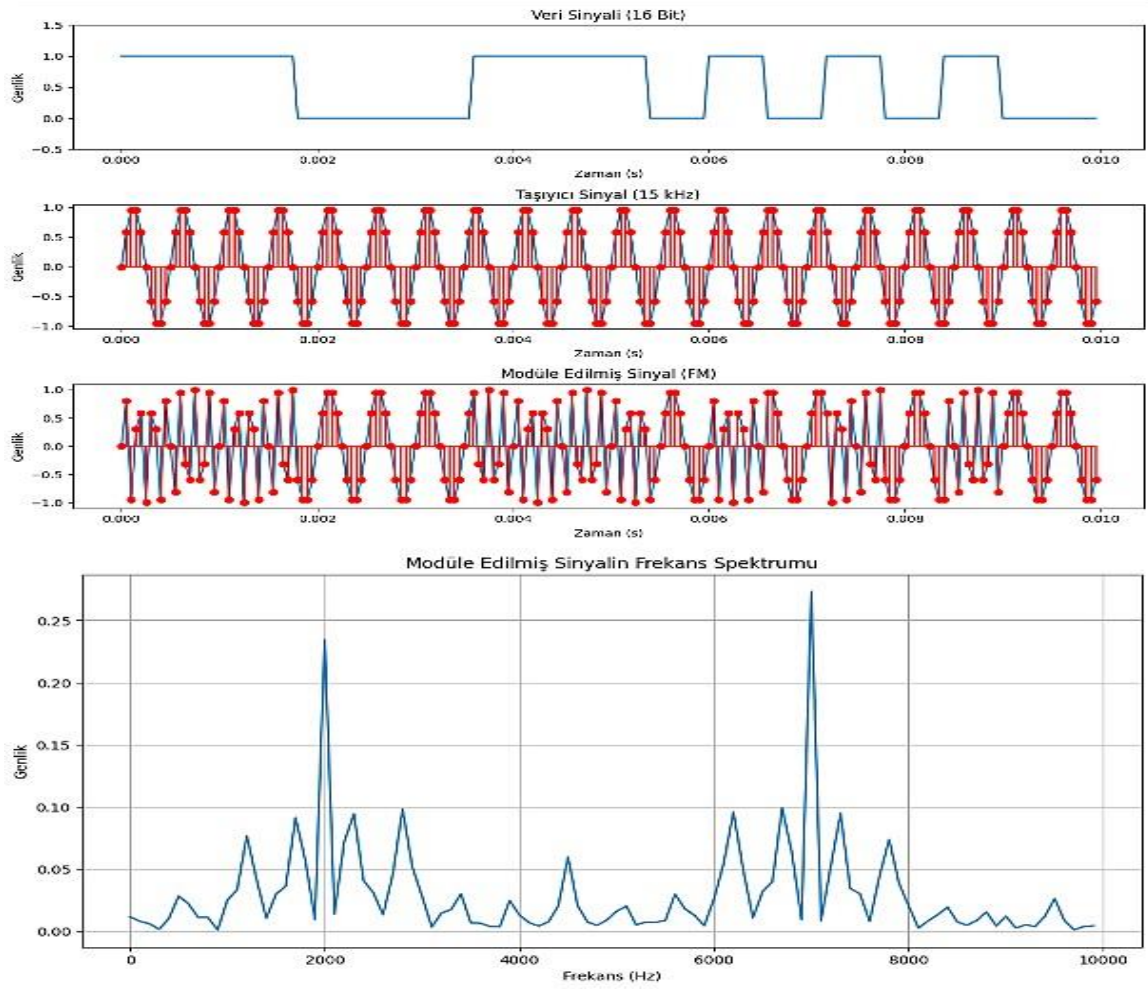
```

**Şekil 3.** Jupyter Notebook Ortamında Taşıyıcı Dalga Modüle etme

**Şekil 3'te**, Python Jupyter Notebook ortamında **Denklem 3** kullanılarak bilginin taşıyıcı dalgaya modülasyonu analiz edilmiştir. Bu analizde, taşıyıcı frekans ( $F_c$ ) 2 – 15 kHz aralığında seçilirken örneklem değeri olan chunk değeri 20 – 44,1 kHz aralığında seçilip test edilmiştir. Ayrıca,  $F_c$ , veri biti 1 olduğu durumlarda en fazla sapmaya uğrayacağı frekans (mod\_index) 1 – 5 kHz aralığında ayarlanmıştır. Modüle edilmiş bir dalganın frekansını bulmak için ters Fourier dönüşümünden yararlanılmıştır. Ters Fourier dönüşümü ile  $F_c$ 'si sapmaya uğrayan bölgeler tespit edilerek mantıksal 1 olduğu bulunmuştur. Bu testte sinyal süresi (duration) 0.1 – 0.01 ms aralıklarında test edilerek örneklem sayısı analiz edilmiş ve ters Fourier dönüşümü ile sapma frekansının tespiti gözlemlenmiştir.

0.01 ms süre (duration) seçilirken 2 kHz taşıyıcı frekansta her bir bit başına düşen örneklem sayısı hesaplanmıştır. Bu durumda, 0.01 ms sürede 2 kHz taşıyıcı frekansta her bir bit başına düşen örneklem sayısı 20 olarak bulunur ( $2 \text{ kHz} * 0.01 \text{ ms} = 20$  örneklem). Taşıyıcı frekans 2 kHz'den 7 kHz'e çıkarken ( $2 \text{ kHz} + 5 \text{ kHz mod\_index}$ ) her bir bit başına düşen örneklem sayısı ise 70 olarak hesaplanır ( $7 \text{ kHz} * 0.01 \text{ ms} = 70$  örneklem). Bir dalgayı temsil etmek için minimum örneklem sayısı, Nyquist kriterine göre belirlenir ve bu kriter, taşıyıcı frekansın en az iki katı kadar örneklem alınmasını gerektirir. Bu durumda, 2 kHz taşıyıcı frekans için minimum örneklem sayısı 4 olur ( $2 * 2 \text{ kHz} = 4$  örneklem) [20].

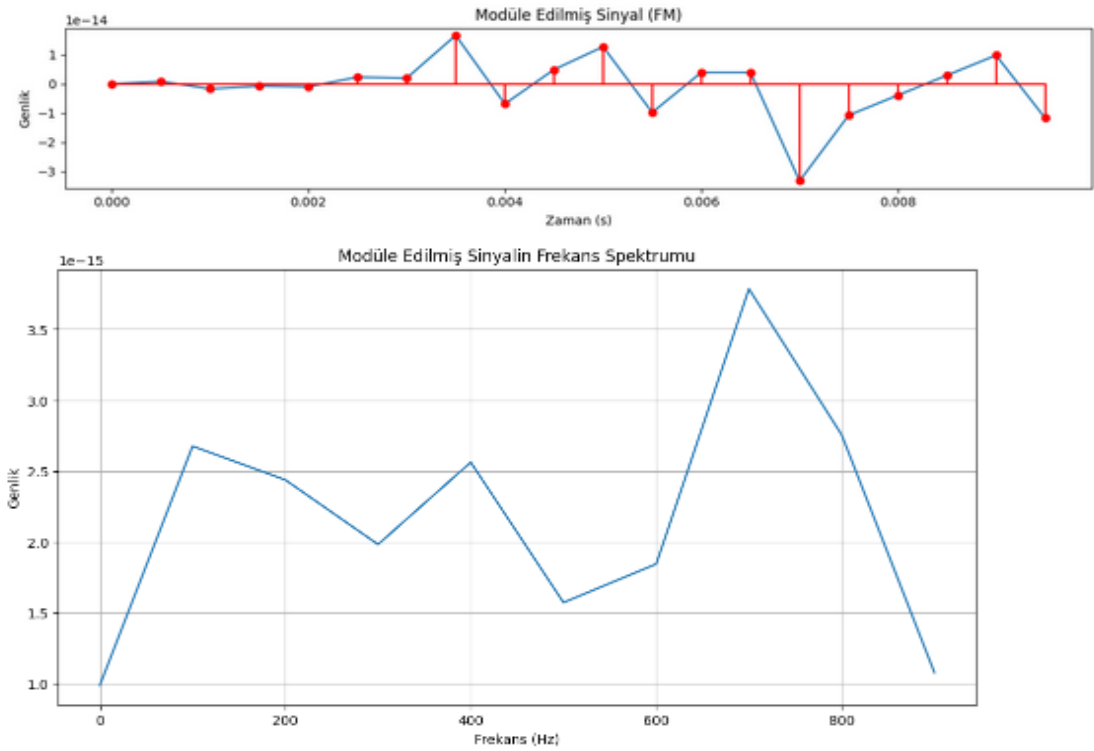




**Şekil 4.** 20 kHz Örnekleme Yapısı İle Belirgin Dalga Yapısı

**Şekil 4'te** module edilen sinyalin bilgi sinyaline göre modülasyonu gösterilmiştir. Bit başına düşen örnekleme sayısı 20 olduğundan dalga fiziksel bir bozulmaya uğramadan 2 kHz'lık ve 7 kHz'lık frekansların ters Fourier dönüşümü ile kolayca tespit edilebildiği gözlemlenmiştir. **Şekil 5'te** örnekleme sayısı yarıya indirilerek dalganın yapısı tekrar incelenmiştir.





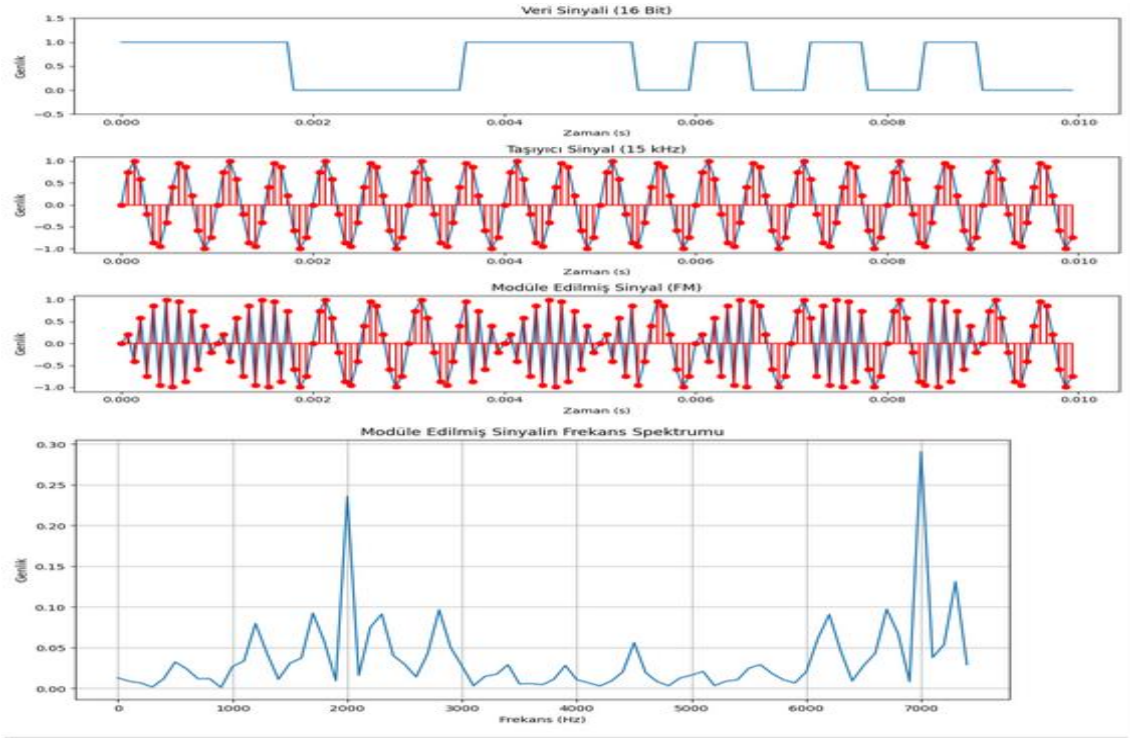
**Şekil 5.** 2 kHz Örneklem Sayısında Modüle Edilen Bozuk Dalga Yapısı

**Şekil 5'te** de görüldüğü üzere örneklem sayısı 20 kHz'den 2kHz'e düşürüldüğünde dalganın yapısını bozulduğu gözlemlenmiştir. Aynı şekilde frekans spektrumun da yeterli veri olmadığından baskın frekansı bulamadığı gözlemlenmiştir. Nyquist Teoremi bize saf bir sinüs dalgasının frekansından en az 2 katı örneklem olması gerektiğini söyler[20]. Bu teoreme göre taşıyıcı frekans 2 kHz ve sapma frekansını belirleyen mod\_index değeri 5 kHz seviyesindeyken örneklem sayısı **Denklem 4'e** göre hesaplanmıştır.

$$\text{OrneklemSayisi} = (\text{Fc} + \text{mod\_index}) * 2$$

**(Denklem 4)**

**Denklem 4'e** göre hesaplanan örneklem sayısı 14 kHz bulunmuştur. 14 kHz sınır olarak kabul edilerek **Şekil 6'da** 15 kHz olarak ayarlanan örneklem sayısının dalga yapısının değişimi görülmektedir.



**Şekil 6.** 15 kHz Örneklem İle Modüle Edilen Dalga Yapısı

Python ortamında elde edilen veriler ile STM32F103C serisinin verici için kodlama adımları **Şekil 2’deki** algoritma akış şeması **Şekil 7’de** gerçekleştirilmiştir. STM32 kodlamasından sonra verici ile ilgili bütün aşamalar bu noktada tamamlanmıştır.

```

1 #include <string.h> //memset fonksiyonu için
2 // Ondalık sayıyı ikili sayı sistemine çeviren fonksiyon
3 void dectoBin(int num, int binArray[], int size) {
4     memset(binArray, 0, size * sizeof(int)); // Diziyi sıfırlıyoruz
5     for (int i = size - 1; i >= 0 && num > 0; i--) {
6         binArray[i] = num % 2;
7         num /= 2;
8     }
9 }
10 // Kinlik bitlerini ekleyen fonksiyon
11 void kinlik(int binArray[], int size, char verici) {
12     if (verici == 'A') {
13         binArray[size-2] = 0;
14         binArray[size-1] = 0;
15     } else if (verici == 'B') {
16         binArray[size-2] = 0;
17         binArray[size-1] = 1;
18     } else if (verici == 'C') {
19         binArray[size-2] = 1;
20         binArray[size-1] = 0;
21     }
22 }
23 // Binary dizisini Gray koduna dönüştüren fonksiyon
24 void binToGray(int binArray[], int grayArray[], int size) {
25     grayArray[0] = binArray[0]; // İlk bit aynı kalır
26     for (int i = 1; i < size; i++) {
27         grayArray[i] = binArray[i] ^ binArray[i-1];
28     }
29 }
30 // Hamming (15,11) kodlama fonksiyonu
31 void hamming_encode(int binArray[], int hammingArray[], int size) {
32     int i, j = 0, k = 0;
33     int parity_count = 4; // (15,11) için 4 parite bitt
34
35     // Hamming kodlaması için 15 elemanlı dizi
36     for (i = 0; i < 15; i++) {
37         // Parite bitlerinin yerlerini belirle
38         if (i == (1 << k) - 1) {
39             hammingArray[i] = 0;
40             k++;
41         } else {
42             hammingArray[i] = binArray[j];
43             j++;
44         }
45     }
46
47     // Parite bitlerini hesapla
48     for (i = 0; i < parity_count; i++) {
49         int pos = (1 << i);
50         int parity = 0;
51         for (j = pos; j < 15; j += (pos << 1)) {
52             for (k = j; k < j + pos && k < 15; k++) {
53                 parity ^= hammingArray[k];
54             }
55             hammingArray[pos - 1] = parity;
56             pos = pos << 1;
57         }
58     }
59 }
60 // SysTick Timer ayarları ve başlangıç
61 void SysTick_Init(void) {
62     // SysTick frekansını ayarla
63     SysTick_Config(SystemCoreClock / 1000); // 1 ms aralıklarla tetiklenecek
64 }
65 // SysTick ISR fonksiyonu
66 volatile uint32_t mSticks = 0;
67 void SysTick_Handler(void) {
68     mSticks++; // Her SysTick tetiklenişinde 1 artır
69 }
70 // Zamanı milisaniye cinsinden döndüren fonksiyon
71 uint32_t zaman(void) {
72     return mSticks;
73 }
74
75 #include "stm32f10x.h"
76
77 void TIM1_Init(void) {
78     // Timer 1 ayarları
79     RCC_APB2PeriphClockCmd(RCC_APB2Periph_TIM1, ENABLE);
80
81     TIM_TimeBaseInitTypeDef timerInitStructure;
82     timerInitStructure.TIM_Prescaler = 0; // Prescaler'ı sıfırla (direct clock)
83     timerInitStructure.TIM_CounterMode = TIM_CounterMode_Up;
84     timerInitStructure.TIM_Period = 1894; // 38 kHz için
85     timerInitStructure.TIM_ClockDivision = TIM_CKD_DIV1;
86     timerInitStructure.TIM_RepetitionCounter = 0;
87     TIM_TimeBaseInit(TIM1, &timerInitStructure);
88
89     TIM_OCInitTypeDef outputChannelInit;
90     TIM_OCStructInit(&outputChannelInit);
91     outputChannelInit.TIM_OCMode = TIM_OCMode_PWM1;
92     outputChannelInit.TIM_OutputState = TIM_OutputState_Enable;
93     outputChannelInit.TIM_Pulse = 947; // Duty Cycle %50
94     outputChannelInit.TIM_OCPolarity = TIM_OCPolarity_High;
95
96     // TIM1 Channel 4
97     TIM_OC4Init(TIM1, &outputChannelInit);
98     TIM_OC4PreloadConfig(TIM1, TIM_OCPreload_Enable);
99
100    TIM_Cmd(TIM1, ENABLE);
101    TIM_CtrlPWMOutputs(TIM1, ENABLE);
102 }
103
104 void setFrequency(uint32_t frequency) {
105     if (frequency == 38000) {
106         TIM1->ARR = 1894; // 38 kHz için Auto-reload değeri
107         TIM1->CCR4 = 947; // 38 kHz için %50 Duty Cycle
108     } else if (frequency == 40000) {
109         TIM1->ARR = 1799; // 40 kHz için Auto-reload değeri
110         TIM1->CCR4 = 899; // 40 kHz için %50 Duty Cycle
111     }
112 }
113
114 void processHammingArray(int hammingArray[], int size) {
115     for (int i = 0; i < size; i++) {
116         if (hammingArray[i] == 1) {
117             setFrequency(38000); // 38 kHz
118         } else {
119             setFrequency(40000); // 40 kHz
120         }
121     }
122
123     // Bir süre bekle (örneğin 1 ms)
124     for (volatile int j = 0; j < 1000; j++);
125 }

```

Şekil 7. STM32F103C ile Verici Kodlama

**1.4. Alıcı Devre Tasarımı ve Demodülasyon İşlemi** 40 kHz alıcıların (receiver) ve kullanılan alıcıların her biri belirli bir frekansı algılamak üzere bu alıcılardan en az iki tane olacak şekilde sistem tasarlanmıştır. İki alıcılı sistemde paralel olarak iki ayrı dalgayı alınması sağlanır. İki alıcı arasındaki frekansın yeterince uzak olması, girişim ve gürültünün olabildiğince azaltılması anlamına gelir [13]. Bu adımda, tasarlamak istediğimiz frekans aralığını belirliyoruz. 40 kHz  $\pm 1$  kHz ile 37 kHz  $\pm 1$  kHz hatalarla çalışan iki alıcı için iki tane bant geçiren filtre tasarımı yapılacaktır. Burada 40 kHz frekansı '0'ları temsil ederken, 37 kHz frekansı '1'leri temsil edecektir. Bant geçiren filtre için olası en iyi üç opamp seçeneğinden TL084CN [13] ve LM741CN [4] opamplarını karşılaştıracacağız. TL084CN, düşük gürültü ve yüksek hız özellikleri ile bilinirken, LM741CN daha eski bir tasarıma sahip olup genellikle düşük maliyetli projelerde tercih edilmektedir [5, 13].

TL084CN Özellikleri [13]:

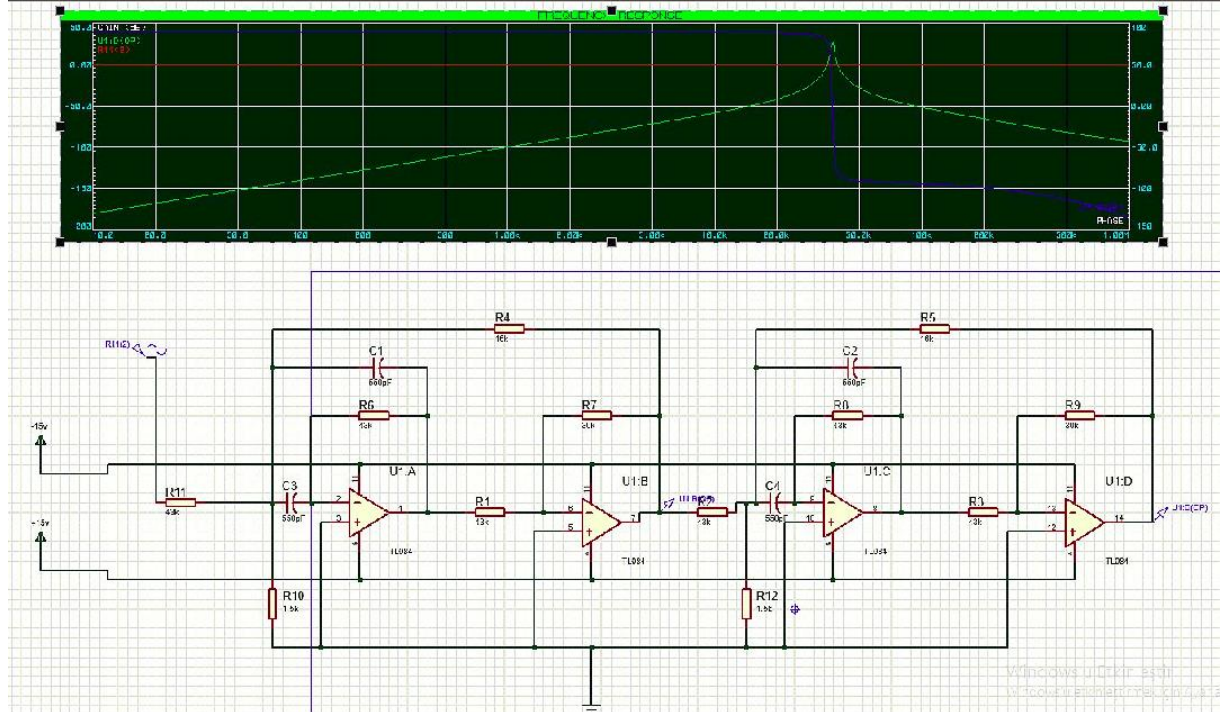
- Düşük gürültü

- Yüksek hız
- Düşük giriş empedansı
- Sinyal tepki süresi: 3V/ $\mu$ s

LM741CN Özellikleri [5]:

- Yüksek gürültü
- Düşük hız
- Yüksek giriş empedansı
- Sinyal tepki süresi: 0.5V/ $\mu$ s

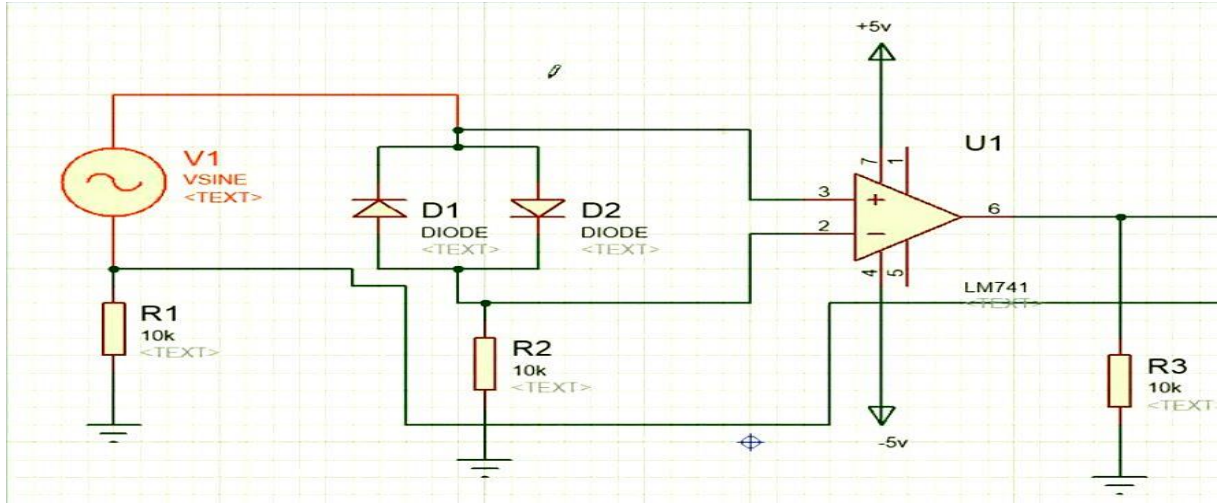
TL084CN, düşük gürültü ve yüksek hız özellikleri ile LM741CN'ye kıyasla daha üstün performans sağlar [5, 13]. Ayrıca, TL084CN'nin sinyal tepki süresi daha hızlıdır ve giriş empedansı daha düşüktür. Bu nedenle, sinyal yükseltme ve bant geçiren filtre tasarımı TL084CN tercih edilmiştir. Bant geçiren filtre devre tasarımı ve buna ilişkin görseller **Şekil 8'de** sunulmuştur.



**Şekil 8.** TL084CN ile Bant Geçiren Filtre Devre Tasarımı [5,13]

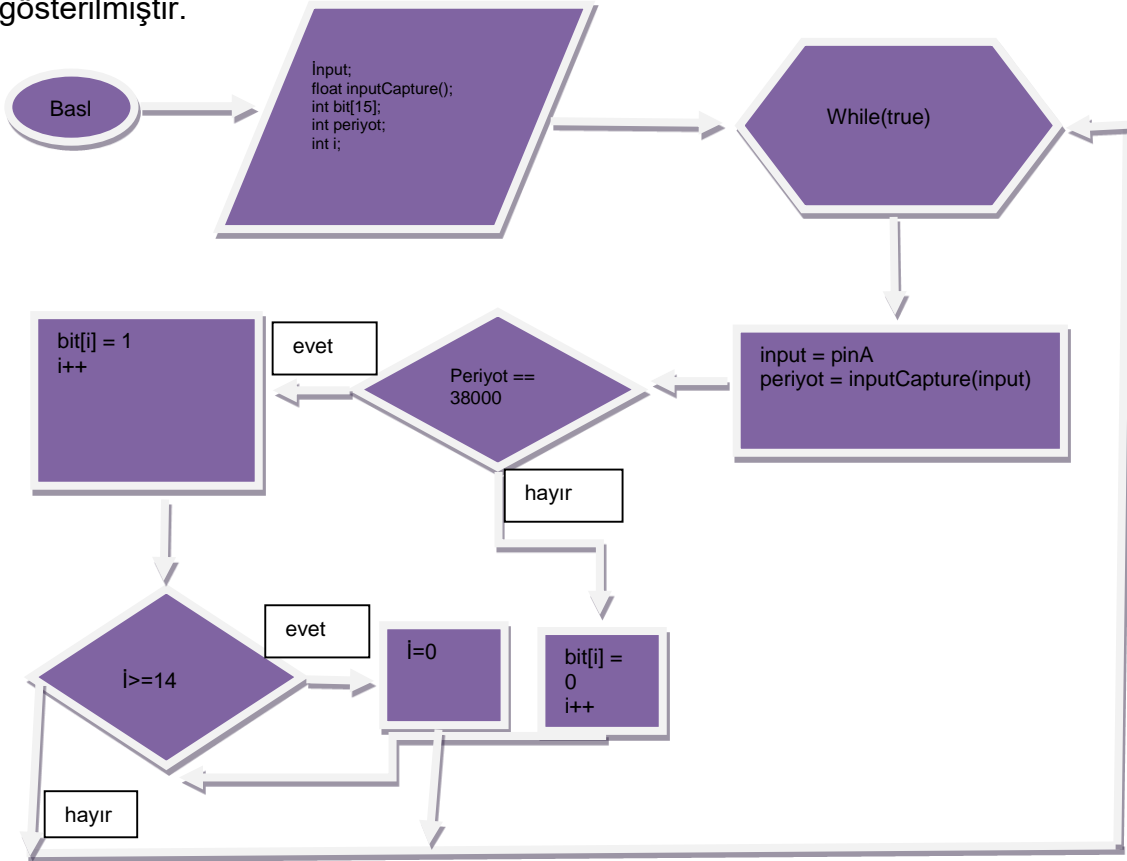
**Şekil 8'de** Proteus 8 ortamında tasarlanan filtre kullanılarak gelen sinyalleri yükseltmek için kullanılan devre şeması ve sinyali arttırmak için yapılan işlemler ayrıntılarıyla sunulmuştur. Manual ayarlama ile en iyi sonucu bulmak için dört tane pot kullanılmıştır. İlk iki pot, 1. ve 2. opamlara gelen alıcıdan sinyalin genliğini azaltıp arttırmaya yararken, 3. pot 3. opampa gelen 2. opamptaki sinyali daha da artırır. Sinyalin taşmasını önlemek için bir tane 3.7V zener diyot kullanılmıştır. Böylece bu

voltajdan yüksek olan sinyaller kısa devre olarak toprağa gider. Yeterince yükseltilmiş sinyali kare dalgaya çevirmek için bir tane LM741 [5] ile kare dalga çevirici yapılmıştır.



**Şekil 9.** LM741 Kare Dalga Çevirici [4]

**Şekil 9'da** görüleceği üzere, kare dalga çevirici devresi, sinyali kare dalgaya çevirir ve bu kare dalga STM32F103C üzerinde input capture pini kullanılarak alınır. Bu pin ile dalganın periyodu tespit edilir ve buna ilişkin kodlar ile akış diyagramı **Şekil 10'da** gösterilmiştir.



**Şekil 10.** STM32F103C İle Alıcı Devre Kodu Akış Diyagramı

## 2. Simülasyonlar ve Deneysel Sonuçlar

Bu çalışmada, çeşitli donanım ve yazılım bileşenleri kullanılarak benzetim ve deneysel sonuçlar elde edilmiştir.

### Donanım:

- Bilgisayar özellikleri:
  - İşlemci: Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz
  - Çekirdek Sayısı: 4
  - Bellek: 8 GB RAM
  - Mimari: x86\_64
  - Maksimum İşlemci Hızı: 3.4 GHz

### Yazılım:

- İşletim Sistemi: Ubuntu 22.04
- Python Sürümü: Python 3.11
- Kullanılan Kütüphaneler: Matplotlib, Numpy, Ursina, Sound
- IDE: Jupyter Notebook

Bu çalışmada, dalga ve su altı simülasyonları için Ursina kütüphanesi, veri görselleştirme için Matplotlib ve Numpy kütüphaneleri kullanılmıştır. Yazılım testlerinde 8-16 bitlik iletimler gerçekleştirilmiştir. Bu çalışmada STM32F103C mikrokontrolcüsü uygulanmıştır. STM32F103C'nin özellikleri: [11]

- ARM Cortex-M3 çekirdeği
- 72 MHz saat frekansı
- 64 KB Flash bellek
- 20 KB SRAM
- Farklı iletişim arayüzleri (SPI, I2C, USART)

STM32F103C'de Giriş Yakalama özelliğinde yüksek frekans tespit edilmiştir. PWM pini kullanılarak 30-40 kHz frekansında dalgalar üretilmiş ve bu dalgalar çıkışta 40 kHz ultrasonik vericilere iletilmiştir.

### 2.1. Kullanılan Devre Elemanları ve Özellikleri

- Mercekli ve kutuplu kondansatörler (100 uF ila 1 nF) bant geçiren filtre sentezlemek için kullanılmıştır. Bu kondansatörler, bir sinyalin belirli bir frekans bandında kabul edilebilir bir zayıflama ile geçmesine izin verirken diğer tüm frekansları zayıflatır. [5]
- Deneyleri gerçekleştirmek için BC237 transistörü kullanıldı. TL084CN ve LM741 op-ampları modellendi.
- TL084CN dörtlü op-amp IC'dir. Kare dalgalar üretmek için LM741 OPAMP kullanılmıştır.
- Alıcı devresinde, sinyal 400 kat büyütülerek op-ampları ile bant geçiren filtre kullanılarak Şekil 10 gösterildiği gibi sinyalin demodülasyonu gerçekleştirilmiştir.



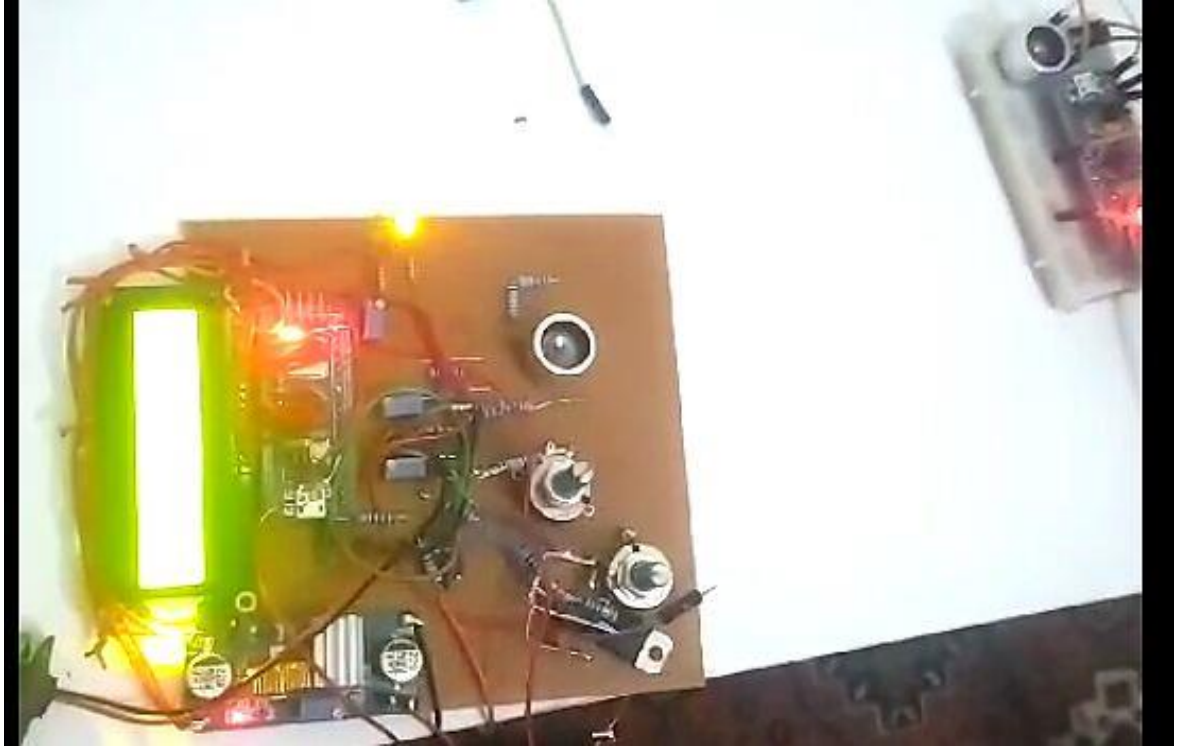
Bu sinyal, Şekil 9'daki dönüştürücü aracılığıyla kare dalgaya dönüştürülmüş ve STM32F103C'ye iletilmiştir.

## 2.2. Ultrasonik Dönüştürücüler

Farklı bir STM32F103C modeli verici olarak kullanılarak, pin'e uygulanan PWM ile çıkışta 40 kHz ultrasonik dönüştürücülere bağlı olan 30-40 kHz frekansında dalgalar üretilmiştir. 40 kHz ultrasonik dönüştürücü alıcı olarak da kullanılmıştır. Donanım kısıtlamaları nedeniyle, alıcı genellikle iki tonu eşzamanlı alabilmek için iki ultrasonik dönüştürücü ile donatılmalıdır, ancak deneyde yalnızca bir tane kullanılmıştır. Bu deneyde, 37 kHz frekansı 0'ı temsil ederken, 38 kHz frekansı 1'i temsil etmektedir.

## 2.3 Simülasyon Akış Şeması ve Veri Görselleştirme

Bu bölüm, karşılaştırılan yöntemin (Geliştirilmiş AMKS) temel parçalarını ve iletişim protokollerini doğrulamak için kullanılan deneysel sonucu açıklar. Prototip donanım test edildi ve temel algoritmalar duyulabilir frekansta doğrulandı. Ayrıca AMKS için, ultrasonik alıcı-verici prototipi işlevsellik ve veri iletimi için tasarlanmış ve test edilmiştir.



**Şekil 11.** Geliştirilen Ultrasonik Alıcı-Verici Prototip Donanımının Genel Görünümü

Şekil 11'in açıklamasında gösterildiği gibi, bu projede mikrokontrolcü STM32F103C ile alıcı ultrasonik devre prototipi geliştirilmiştir. Devre, sinyal işleme için bir dönüştürücü, alınan sinyalin frekansını gösteren bir 16x2 LCD ekran, sinyal şartlandırma için bir op-amp ve kontrol potansiyometreleri ve sinyal şartlandırma potansiyometresi de eklenmiştir. Ayrıca, verici modülünün, ultrasonik alıcısı içeren bir kısmı

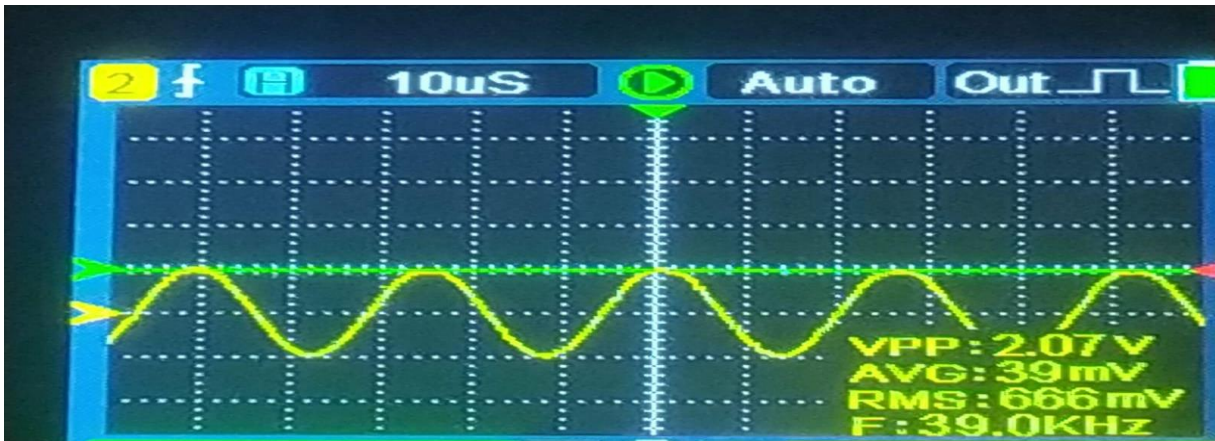


gösterilmektedir.



**Şekil 12.** Alıcı LCD Ekranında Ultrasonik Sinyal İle Tespit Edilen Frekans Ve Çözömlenen Bit Dizisi Örneđi.

Açıklama: Şekil 12, alıcı LCD'sinde, dönüştürücünün aldığı ultrasonik sinyalden anlık olarak tespit edilen ve işlenen "Frekans: 38461 Hz" değeri ile ilgili yorumlanan bit dizisini "F38K F37K: 1 00" göstermektedir. Çıktıdan, hedeflenen ~38 kHz frekansının '1' biti olarak, takip eden ~37 kHz gözlemlenebilir. Gözlemlenen nominal frekans değerlerinden sapmalar, sistemin osilatörünün tolerans ofsetleri ile birlikte sistemin değışken çevresel koşullarının sonucu olabilir.



**Şekil 13.** Alıcı Devresinde Filtrelenmiş ve Yükseltilmiş Ultrasonik Sinyalin Osiloskop Görüntüsü (yaklaşık 39.0 kHz).

Şekil 13, alıcı ultrasonik transdüserden alınan, bant geçiren filtreleme ve operasyonel amplifikatör katlarında yükseltme işlemlerinden sonra elde edilen analog sinyalin osiloskop ekranındaki dalga formunu göstermektedir. Osiloskop ölçümleri, sinyalin frekansının yaklaşık 39.0 kHz (F: 39.0KHz), tepe-tepe geriliminin (VPP) 2.07V, ortalama geriliminin (AVG) 39mV ve RMS geriliminin 666mV olduğunu göstermiştir. Bu dalga formu, sinyalin dijital işleme veya kare dalga çevirici katına girmeden önceki analog durumunu temsil etmekte ve kullanılan filtreleme devresinin hedef frekans bandındaki sinyali etkin bir şekilde geçirdiğini teyit etmektedir.

## 2.4. Malzeme Ortamlarındaki Farklı Ses Yayılımları

Dalga yayılımının hızı, dalgaları destekleyen ortamın temel fiziksel özelliklerine, yani ortamın sertliğine (ya da gazlar söz konusu olduğunda sıkıştırılabilirliğine) ve kütle yoğunluğuna büyük ölçüde bağlıdır.

Genel bir kural olarak, bir ortam ne kadar sertse (sıkıştırmaya karşı ne kadar hoşgörüsüzse), ses dalgaları o kadar hızlı yol alır.

Öte yandan, ortamın yoğunluğu ne kadar yüksekse, ses o kadar yavaş hareket eder. Gazlar genellikle en yavaş ses iletim ortamlarıdır çünkü yüksek sıkıştırılabilirlikleri vardır; (örneğin, ses 0°C'de havada yaklaşık 331 m/s hızla hareket eder), oysa sıvılar ve katılar gazlara göre çok daha az sıkıştırılabilir ve daha sert yapıya sahiptir.

Bu nedenle, bu ortamlarda ses çok daha hızlı yol alır; 20°C'de durgun su içindeki ses hızı yaklaşık 1480 m/s iken çelik gibi katı bir malzeme içindeki hız 5960 m/s'dir. Sesin hızı ( $v_w$ ), frekansı ( $f$ ) ve dalga boyu ( $\lambda$ ) arasındaki evrensel ilişki Denklem 5 ile tanımlanır:

$$v_w = f \lambda \quad (\text{Denklem 5})$$

Bu denklemde  $v_w$ , sesin yayılma hızını,  $f$ , ses dalgasının frekansını (birim zamanda bir noktadan geçen dalga sayısı) ve  $\lambda$ , dalga boyunu (dalganın ardışık özdeş noktaları arasındaki mesafe) ifade eder. Ses hızının genel formülü, ortamın elastikiyetine ve yoğunluğuna bağlı olarak Denklem 6 ile verilir:

$$v = \sqrt{\frac{B}{\rho}} \quad (\text{Denklem 6})$$

Burada  $B$ , bulk modulus (elastikiyet modülü),  $\rho$ , yoğunluktur.

**Tablo 2**, çeşitli maddesel ortamlarda ses hızı değerlerini özetlemektedir:

**Tablo 2** Farklı Maddesel Ortamdaki Sesin Hızı [22-23]

Ortam	Sıcaklık (°C)	Ses Hızı ( $v_w$ ) (m/s)
	<b>Gazlar</b>	

Hava	0	331
Karbondiyoksit	0	259
Oksijen	0	316
Helium	0	965
Hidrojen	0	1290
<b>Sıvılar</b>		
Etanol	20	1160
Mercur	20	1450
Su, taze	20	1480
Deniz suyu	20	1540
İnsan dokusu	20	1540
<b>Katılar (Uzunlamasına veya Toplu)</b>		
Vulkanize kauçuk	-	54
Polietilen	-	920
Mermer	-	3810
Cam, Pyrex	-	5640
Kurşun	-	1960
Alüminyum	-	5120
Çelik	-	5960

## 2.5. Sesin Sıcaklıkla İlişkisi

Sesin yayılma hızı, ortamın sıcaklığından da etkilenir ve bu etki özellikle gazlarda çok daha belirgindir. Bir gazın sıcaklığı ne kadar yüksek olursa, onu oluşturan moleküllerin hareket enerjisi o kadar fazla olur ve dolayısıyla hareket hızı artar. Bu da, ortamda ses dalgalarının hızlı yayılmasına neden olur.

Havada, sesin hızı 0 °C'de 331 m/s (1,088 ft/s) iken, 20 °C'de 343 m/s (1,125 ft/s), yani %3,8'lik bir artış gösterir.

Sıvı ve katılarda, moleküler bağlar daha güçlü olduğu ve bozunmaların yayılabileceği yapılar daha düzenli olduğu için, sıcaklık değişimlerinin bu tür ortamlarda ses hızında yarattığı etki genellikle gazlardakinden daha azdır ve farklı mekanizmalarla gerçekleşir.

### Formüller:

Deniz seviyesindeki hava için ses hızının ( $v_w$ ) sıcaklığa ( $T$ , Kelvin cinsinden) olan bağımlılığı Denklem 7 ile ifade edilir:

$$v_w = 331.3 \sqrt{\frac{T}{273}} \quad (\text{Denklem 7})$$

Alternatif olarak, Celsius cinsinden sıcaklık ( $T$ ) için şu formül kullanılabilir:

$$v_w = 331.3 + 0.606 \times T$$

(Denklem 8)

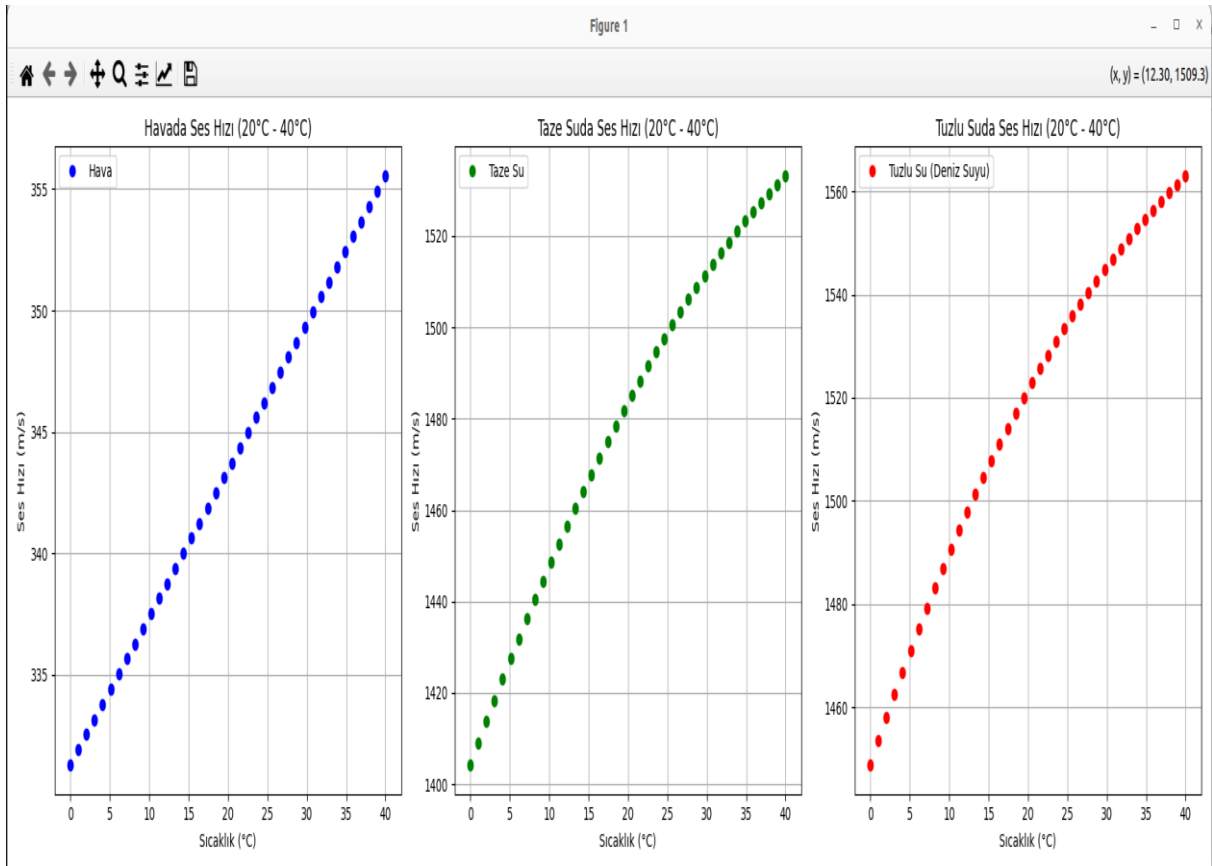
#### Veri Tablosu:

**Tablo 3**, farklı sıcaklıklarda havadaki ses hızını özetlemektedir:

**Tablo 3** Farklı Sıcaklıklarda Havadaki Sesin Hızı [22-23]

Sıcaklık (°C)	Sıcaklık (K)	Ses Hızı (m/s)
0	273	331
10	283	338
20	293	343
30	303	349
40	313	355

Sıcaklığın ses hızına etkisini görselleştirmek için bir çizgi grafik kullanılmıştır. Şekil 14, sıcaklık (x-ekseni) ve ses hızı (y-ekseni) arasındaki doğrusal ilişkiyi gösterir. Grafik, ses hızının sıcaklıkla doğru orantılı olduğunu gösterir.



**Şekil 14.** Tablo 2'e Göre 3 Farklı Ortamdaki Sıcaklığa Bağlı Ses Hızı Grafiği

## 2.6. İletişim Algoritmalarının Sesli Frekans Aralığında Doğrulanması ve Ön Performans Testleri

Bu alt bölümde, Manchester kodlama, CRC hata denetimi ve Start/End bit protokolü gibi temel iletişim algoritmalarının işlevselliği, ultrasonik sistem entegrasyonu öncesinde sesli frekans aralığında (örneğin 8-16 kHz) doğrulanmıştır. Bu testler, PC tabanlı veya Raspberry Pi gibi platformlar kullanılarak gerçekleştirilmiştir.

---

**Farksal Manchester Kodlaması ile Frekans Gönderici**

Bit Dizisi:

"1" Frekansı (Hz):

"0" Frekansı (Hz):

Süre (saniye):

Durum: Bekliyor...

**Şekil 15.** Raspberry Pi Tabanlı Manchester Kodlamalı Sesli Frekans Vericisi için Geliştirilen Web Tabanlı Kullanıcı Arayüzü.

Şekil 15, iletişim algoritmalarının ön testleri için geliştirilen ve Raspberry Pi üzerinde çalışan bir web arayüzünü göstermektedir. Bu arayüz, kullanıcının Manchester kodlaması için bit akışı, '0' bitleri için yüksek ve düşük frekans, '1' bitleri için kullanılan özel bir frekans ve başlangıç/bit bitleri için yarım bit zamanı gibi parametreleri belirtmesini sağlar. "Frekansları Gönder" komutu aracılığıyla, bu parametrelere göre frekans kodlu bir ses sinyali üretilir ve Raspberry Pi'nin ses çıkışına gönderilir. Bu platform, öncelikle Manchester kodlamasının, veri paketlemesinin ve CRC algoritmalarının ilk fonksiyonel doğrulama aşamaları için kullanılmıştır.



```
import numpy as np
import sounddevice as sd
import time

# Frekanslar
FREQ1 = 5500
FREQ0 = 5000
FREQ_START = 4500
FREQ_STOP = 4000
THRESHOLD = 2.8
HALF_BIT_DURATION = 0.1 # <-- Varsayılan süre güncellendi
SAMPLING_RATE = 100000
WINDOW_SIZE = int(HALF_BIT_DURATION * SAMPLING_RATE)
DEBOUNCE_TIME = HALF_BIT_DURATION * 1.1

# CRC-3 (x^3 + x + 1) hesaplama
def crc3(bits):
    reg = 0
    for bit in bits:
        b = int(bit)
        reg = ((reg << 1) | b) & 0xF # 4 bit
        if reg & 0x8:
            reg ^= 0xB # 0b1011
    return str((reg >> 2) & 1), str((reg >> 1) & 1), str(reg & 1)]

def detect_strongest_freq(data, sampling_rate):
    fft = np.fft.rfft(data)
    freqs = np.fft.rfftfreq(len(data), 1/sampling_rate)
    amplitudes = np.abs(fft)

    idx1 = np.argmin(np.abs(freqs - FREQ1))
    idx0 = np.argmin(np.abs(freqs - FREQ0))
    idxstart = np.argmin(np.abs(freqs - FREQ_START))
    idxstop = np.argmin(np.abs(freqs - FREQ_STOP))
    amp1, amp0, ampstart, ampstop = amplitudes[idx1], amplitudes[idx0], amplitudes[idxstart], amplitudes[idxstop]

    amps = [amp1, amp0, ampstart, ampstop]
    freqs_list = [FREQ1, FREQ0, FREQ_START, FREQ_STOP]
    if max(amps) < THRESHOLD:
        return None
    return freqs_list[np.argmax(amps)]

def manchester_decode(freqs):
    # Her iki frekans bir bit demek: [1,0]=1, [0,1]=0
    bits = []
    if len(freqs) % 2 != 0:
        return None # Hatalı uzunluk
    for i in range(0, len(freqs), 2):
        first, second = freqs[i], freqs[i+1]
        if first == FREQ1 and second == FREQ0:
            bits.append('1')
        elif first == FREQ0 and second == FREQ1:
            bits.append('0')
        else:
            return None # Hatalı Manchester kodlama/uzunluk!

    return bits

# Terminal Çıktısı
>> START biti alındı, veri bekleniyor...
>> HATALI Manchester kodlama/uzunluk!
>> START biti alındı, veri bekleniyor...
>> HATALI Manchester kodlama/uzunluk!
>> START biti alındı, veri bekleniyor...
>> HATALI Manchester kodlama/uzunluk!
>> START biti alındı, veri bekleniyor...
>> HATALI Manchester kodlama/uzunluk!
>> START biti alındı, veri bekleniyor...
>> HATALI Manchester kodlama/uzunluk!
>> Çözülen bitler: 11001100 (data: 11001, crc: 100)
>> Hesaplanan CRC: 100
>> CRC DOĞRU, veri geçerli!

>> START biti alındı, veri bekleniyor...
>> Çözülen bitler: 11001100 (data: 11001, crc: 100)
>> Hesaplanan CRC: 100
>> CRC DOĞRU, veri geçerli!

>> START biti alındı, veri bekleniyor...
>> Çözülen bitler: 11001100 (data: 11001, crc: 100)
>> Hesaplanan CRC: 100
>> CRC DOĞRU, veri geçerli!

>> START biti alındı, veri bekleniyor...
>> HATALI Manchester kodlama/uzunluk!
>> START biti alındı, veri bekleniyor...
>> HATALI Manchester kodlama/uzunluk!
```

**Şekil 16.** Python ile Geliştirilmiş Sesli Frekans Alıcısında Manchester Kodlu Veri Paketinin Başarılı Bir Şekilde Çözümlemesine İlişkin Terminal Çıktısı.

Şekil 16, ses frekansları aracılığıyla iletilen (örneğin, Şekil 16'daki arayüzden gönderilen) Manchester kodlu bir veri paketinin, Python tabanlı bir alıcı programı (receiver2.2.py) tarafından işlenmesi sonucu elde edilen terminal çıktısını göstermektedir. Terminalde, algılanan Start/End bitleri, çözümlenen ham bit dizisi, bu diziden ayrıştırılan veri bitleri ve CRC bitleri ("Çözülen bitler: 11001100 (data: 11001, crc: 100)") ile CRC kontrolünün sonucu ("CRC DOĞRU, veri geçerli!") net bir şekilde görülmektedir. Bu sonuçlar, geliştirilen kodlama/çözme algoritmalarının ve CRC hata denetim mekanizmasının sesli bir iletişim ortamında beklenen performansı uygun koşullar sağlandığında gösterdiğini doğrulamaktadır.

## 2.7. Sesli Frekans Aralığında Sinyal Yayılım Karakteristiklerinin İncelenmesi

Bu alt bölümde, frekans.csv dosyasında toplanan deneysel veriler kullanılarak, farklı sesli frekansların (1-20 kHz) çeşitli mesafelerdeki sinyal gücü ve algılanabilirliği incelenmiştir. Bu analizler, ses dalgalarının yayılımı ve zayıflaması hakkında ön bilgiler sunmaktadır.

```
In [13]: import pandas as pd
import matplotlib.pyplot as plt

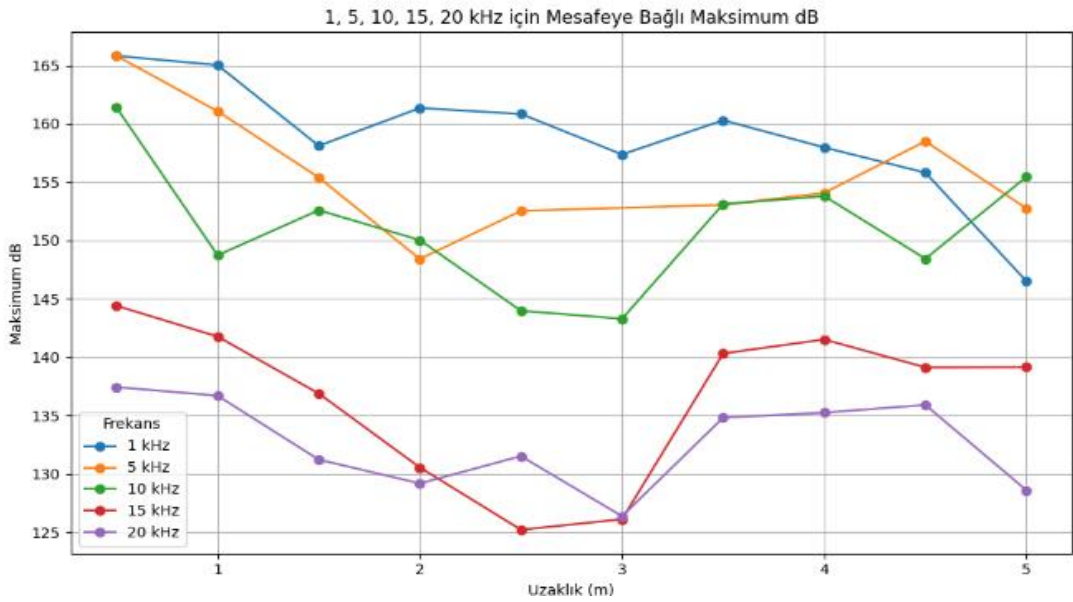
# CSV'yi oku
df = pd.read_csv("frekans.csv")

# Sayısal tip dönüşümü ve temizlik
df["frekans"] = pd.to_numeric(df["frekans"], errors='coerce')
df["uzaklik"] = pd.to_numeric(df["uzaklik"], errors='coerce')
df["maks_db"] = pd.to_numeric(df["maks_db"], errors='coerce')
df = df.dropna(subset=["frekans", "uzaklik", "maks_db"])

# İstediğin frekanslar (Hz)
freqs = [1000, 5000, 10000, 15000, 20000]

plt.figure(figsize=(10,6))
for freq in freqs:
    df_f = df[df["frekans"] == freq].sort_values("uzaklik")
    if not df_f.empty:
        plt.plot(df_f["uzaklik"], df_f["maks_db"], marker='o', linestyle='-', label=f"{freq//1000} kHz")

plt.xlabel("Uzaklık (m)")
plt.ylabel("Maksimum dB")
plt.title("1, 5, 10, 15, 20 kHz için Mesafeye Bağlı Maksimum dB")
plt.legend(title="Frekans")
plt.grid(True)
plt.tight_layout()
plt.show()
```



**Şekil 17.** Farklı Sesli Frekanslarda (1-20 kHz) Mesafeye Bağlı Olarak Alınan Maksimum Sinyal Gücündeki (dB) Değişim.

Şekil 17, 1 kHz ile 20 kHz arasında değişen çeşitli sesli frekanslar için, sinyal kaynağına olan uzaklığın (1 metreden 5 metreye kadar) bir fonksiyonu olarak ölçülen maksimum sinyal gücündeki (dB cinsinden) değişimi göstermektedir. 1-20 kHz arası frekanslarla yapılan deneyde elde edilen veriler, genel olarak tüm frekanslarda mesafe arttıkça sinyal gücünde bir düşüş olduğunu göstermektedir. Ancak, zayıflama oranı frekansa bağlı olarak farklılık göstermektedir; düşük frekansların (örn: 1 kHz, 5 kHz) belirli mesafelerde daha yüksek sinyal gücünü koruma eğiliminde olduğu, yüksek frekansların (örn: 15 kHz, 20 kHz) ise daha hızlı sönümlendiği veya belirli mesafelerden sonra sinyal gücünde daha keskin düşüşler sergilediği gözlemlenmiştir.



```

in [11]: import pandas as pd
import matplotlib.pyplot as plt

# CSV'yi oku
df = pd.read_csv("frekans.csv")

# Sayısal tip dönüşümü ve temizlik
df["frekans"] = pd.to_numeric(df["frekans"], errors='coerce')
df["algilama_sayisi"] = pd.to_numeric(df["algilama_sayisi"], errors='coerce')
df["maks_db"] = pd.to_numeric(df["maks_db"], errors='coerce')
df = df.dropna(subset=["frekans", "algilama_sayisi", "maks_db"])

# Her frekans için ortalama algılama sayısı ve ortalama maks dB
mean_stats = df.groupby("frekans").agg({
    "algilama_sayisi": "mean",
    "maks_db": "mean"
}).reset_index()

fig, ax1 = plt.subplots(figsize=(9,5))

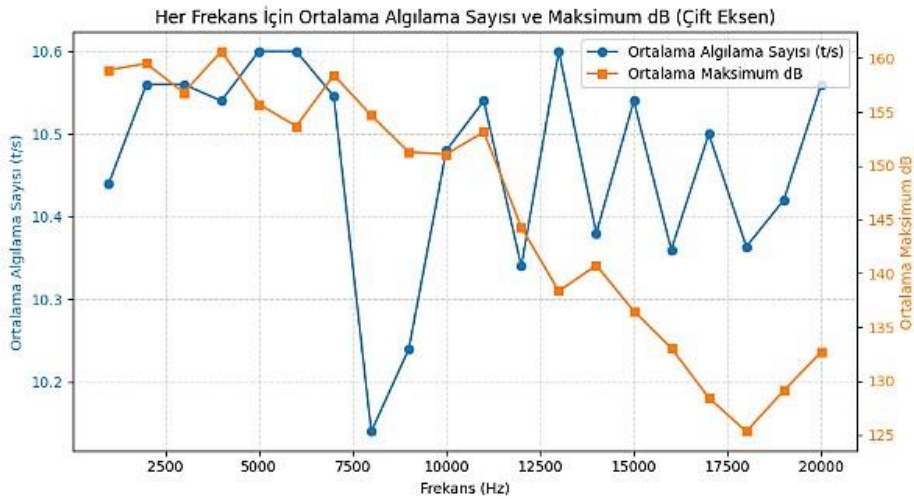
color1 = 'tab:blue'
ax1.set_xlabel("Frekans (Hz)")
ax1.set_ylabel("Ortalama Algılama Sayısı (t/s)", color=color1)
ln1 = ax1.plot(mean_stats["frekans"], mean_stats["algilama_sayisi"], marker='o', linestyle='--', color=color1, label="Ortalama Algılama Sayısı (t/s)")
ax1.tick_params(axis='y', labelcolor=color1)
ax1.grid(True, which='both', axis='both', linestyle='--', alpha=0.5)

ax2 = ax1.twinx()
color2 = 'tab:orange'
ax2.set_ylabel("Ortalama Maksimum dB", color=color2)
ln2 = ax2.plot(mean_stats["frekans"], mean_stats["maks_db"], marker='s', linestyle='--', color=color2, label="Ortalama Maksimum dB")
ax2.tick_params(axis='y', labelcolor=color2)

# Legend'ları birleştir
lns = ln1 + ln2
labs = [l.get_label() for l in lns]
ax1.legend(lns, labs, loc="upper right")

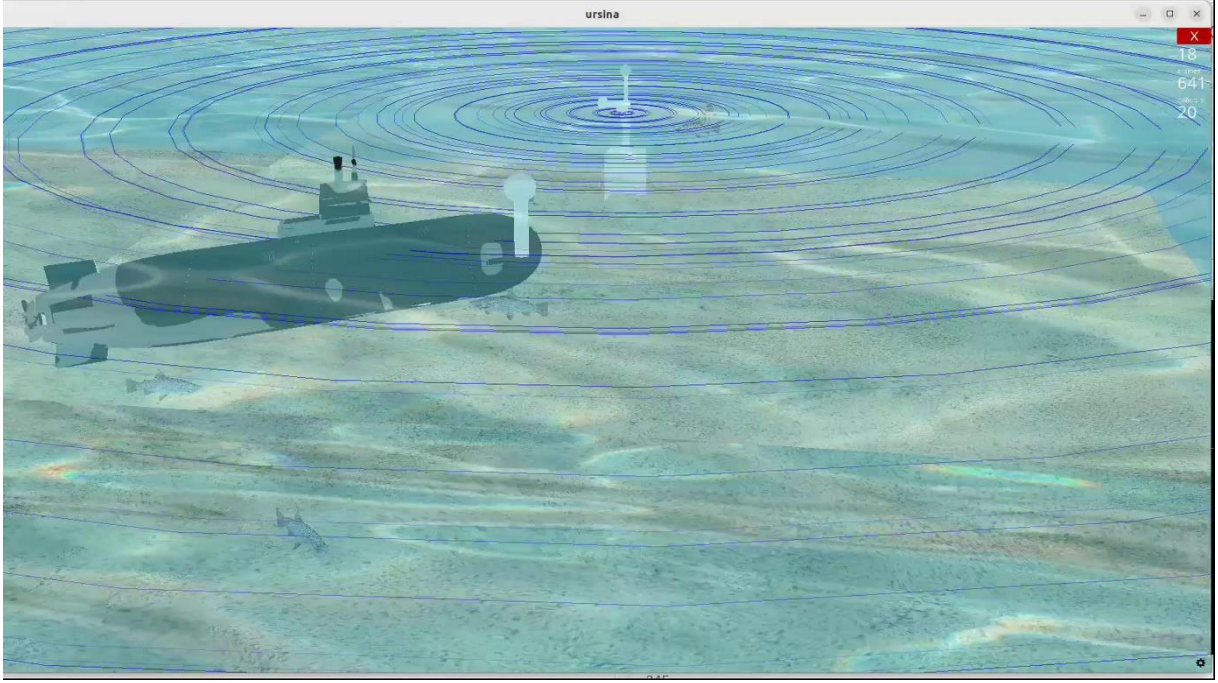
plt.title("Her Frekans İçin Ortalama Algılama Sayısı ve Maksimum dB (Çift Eksen)")
plt.tight_layout()
plt.show()

```



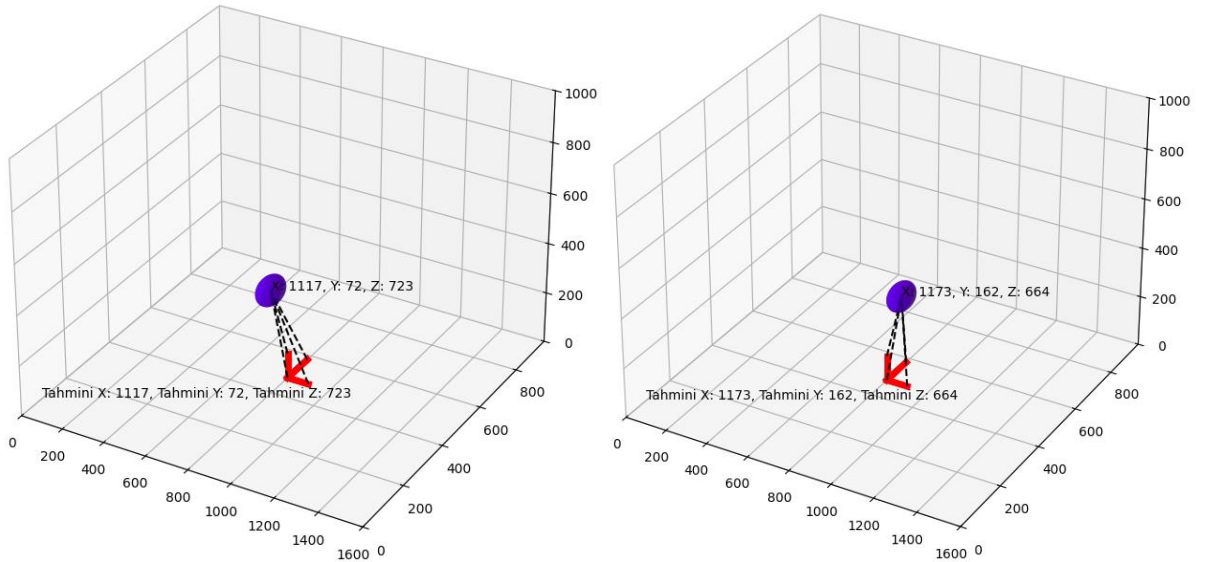
**Şekil 18.** Farklı Sesli Frekanslarda Ortalama Algılama Sayısı (tespit/saniye) ve Ortalama Maksimum Sinyal Gücünün (dB) Karşılaştırmalı Analizi (Çift Eksenli Grafik).

Şekil 18, 1 kHz ile 20 kHz arasındaki sesli frekanslar için frekans.csv verilerinden hesaplanan ortalama algılama sayısını (sol y-ekseni, tespit/saniye, mavi eğri) ve bu frekanslarda ölçülen ortalama maksimum sinyal gücünü (sağ y-ekseni, dB, turuncu eğri) çift eksenli bir grafik üzerinde sunmaktadır. Analizimizden, DSIA sayısının belirli frekans pencerelerinde (örneğin, 10-12,5 kHz bandında) oldukça yüksek olduğu ve sinyal gücünün de yüksek olduğu açıktır. Ancak, bazı daha yüksek frekanslarda (örneğin, 17,5 kHz ve üzeri) her iki çürümeye eğiliminde azalma görülebilir. Bu sonuçlar, farklı frekanslarda sistemin algılama verimliliğini ve alınan sinyalin kalitesini nicelleştirmemize olanak tanır.



**Şekil 19.** Python Ursina İle AMKS Benzetim Modelleme

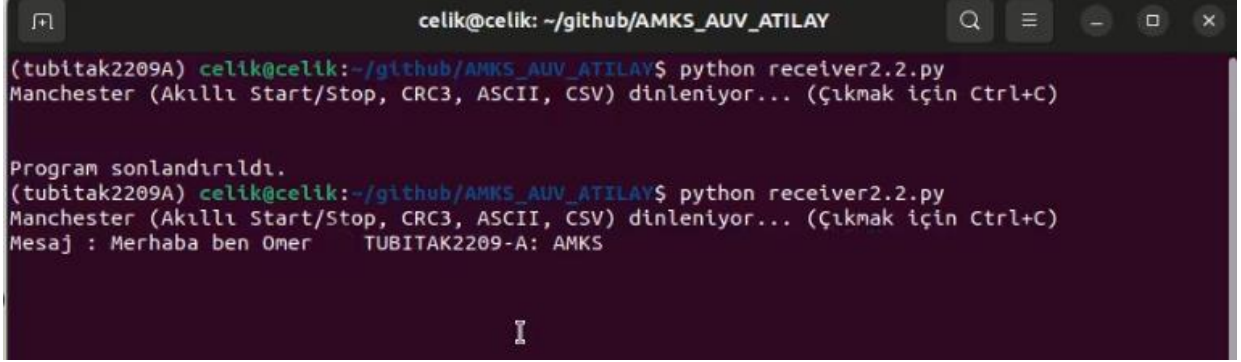
Şekil 19'da Python programlama dili kullanarak Ursina Kütüphanesi ile AMKS benzetiminin bir modeli tasarlanıp birbirine dik olacak şekilde I uzunluğunda özdeş çubuklara bağlı vericilerle saat bilgisi denizaltının alıcı sistemine ulaştıktan sonra Şekil 11'de verilen demodülasyonu işlemini yapan cihaz benzetimi ile çözülüp mesafe bilgisine dönüştürülmüştür.



**Şekil 20.**T Saniye Sonra Alınan Sinyalden Konum Hesaplama

Şekil 19'da verici sistemden denizaltına gelen sinyalin demodülasyonu gerçekleştikten sonra Şekil 20'de hedeften uzaktaki denizaltını temsil eden mavi noktanın birbirine dik

I uzunluğundaki 3 vericinin uzaklığı bilgisinden Tahmini X ve Tahmini Y noktaları hesaplanmıştır.



```
celik@celik: ~/.github/AMKS_AUV_ATILAY
(tubitak2209A) celik@celik:~/.github/AMKS_AUV_ATILAY$ python receiver2.2.py
Manchester (Akıllı Start/Stop, CRC3, ASCII, CSV) dinleniyor... (Çıkmak için Ctrl+C)

Program sonlandırıldı.
(tubitak2209A) celik@celik:~/.github/AMKS_AUV_ATILAY$ python receiver2.2.py
Manchester (Akıllı Start/Stop, CRC3, ASCII, CSV) dinleniyor... (Çıkmak için Ctrl+C)
Mesaj : Merhaba ben Omer TUBITAK2209-A: AMKS
```

**Şekil 21.** Hava Ortamında 5 kHz Frekanslı Ses Dalgası İle Gerçek Zamanlı Mesaj İletimi

Şekil 21’de konum hesaplama için kullanılan AMKS prototipi kullanılarak hava ortamında ayrıca mesaj iletimi de gerçekleştirilmiştir. İletim sadece yazılımsal olarak 3 metreden Manchester kodlama kullanılarak yine hava ortamında Şekil 17 ve Şekil18’deki grafiklerde de görüleceği üzere en yüksek verimi verilen 1 - 5 kHz aralığındaki ses dalgaları kullanılarak gerçekleştirilmiştir. Bu sistem sayesinde mesaj iletimi tıpkı karada tekrarlayıcı (repeater) kullanılarak menzilin artırılmasıyla uzak mesafelerde bile gerçekleştirilebilir. Bu sistemin sağlayacağı en büyük esneklik birbirine dik en az 3 tane vericiden X ve Y koordinatlarının hesaplanabileceği gibi mesaj iletiminin de haliyle saat bilgisini iletmek için tasarlanan sistemle çok kolay bir şekilde iletilebiliyor olmasıdır.

### 3. Rapor Dönemlerinde Yapılan Çalışmalar:

1. **Kapsamlı Literatür Taraması:** Sualtı konumlandırma ve iletişim teknolojileri üzerine derinlemesine bir literatür araştırması yapıldı.
2. **Ses Fiziği ve Yayılımı İncelemesi:** AMKS'nin temel aldığı ses dalgalarının sualtındaki fiziksel davranışları, yayılım karakteristikleri ve sönümlenme faktörleri incelendi.
3. **Mevcut Akustik Sistemlerin Analizi:** Literatürde yer alan LBL, SBL, USBL gibi mevcut sualtı akustik konumlandırma sistemlerinin çalışma prensipleri, avantajları, dezavantajları, kurulum ve maliyet durumları detaylıca incelendi.
4. **Yeni Nesil Sualtı Teknolojilerinin Araştırılması:** TÜBİTAK destekli SAM (Su Altı Akustik Modem) projesi gibi yerli ve milli teknoloji geliştirme çabaları takip edildi.
5. **Alternatif İletişim Teknolojilerinin Değerlendirilmesi:** ASELSAN'ın LiFi modem teknolojisi incelenerek, görünür ışıkla haberleşmenin sualtı AMKS sistemine potansiyel entegrasyonu ve avantajları araştırıldı.
6. **Pasif Elektrik Alanı Yönteminin İncelenmesi:** 2024 tarihli "An Underwater Passive Electric Field Positioning Method Based on Scalar Potential" makalesi analiz edilerek, bu yöntemin AMKS'ye entegrasyonu ve potansiyel olarak ışık hızında veri iletimi imkanları değerlendirildi.

7. **Sualtı Simülasyon Altyapısının Kurulması:** Derin sualtı araştırmalarından elde edilen veriler ve fiziksel modeller kullanılarak Python programlama dili ile gerçekçi sualtı ortamı ve akustik yayılım simülasyonları geliştirildi.
8. **Veri Kodlama ve Modülasyon Stratejilerinin Belirlenmesi:** Veri iletimi için 11 bitlik veri formatı, verici kimliği tanımlaması, Gray kodu ve (15,11) Hamming kodu gibi hata düzeltme teknikleri seçildi ve Python'da simüle edildi.
9. **Frekans Modülasyonu (FM) Simülasyonları:** Python (Jupyter Notebook) ortamında, 40 kHz taşıyıcı frekans ve farklı sapma frekansları (örn: 38 kHz için '1', 40 kHz için '0') kullanılarak FM sinyallerinin üretimi ve analizi yapıldı. Taşıyıcı frekans, modülasyon indeksi ve örneklem sayısının (Nyquist kriteri) etkileri incelendi.
10. **Hava Ortamında Sesli Frekans Testleri (Ön Doğrulama):**
  - İletişim algoritmalarının (Manchester kodlama, CRC) doğrulanması için Raspberry Pi tabanlı bir web arayüzü ile 8-16 kHz aralığında sesli frekans testleri gerçekleştirildi.
  - Python tabanlı bir alıcı ile gönderilen verilerin başarılı bir şekilde çözümlendiği teyit edildi.
  - Farklı sesli frekansların (1-20 kHz) çeşitli mesafelerdeki sinyal gücü ve algılanabilirliği analiz edildi.
11. **STM32F103C Mikrodenetleyici ile Verici Kodlaması:** Python simülasyonlarından elde edilen verilere dayanarak, seçilen veri kodlama ve FM modülasyon algoritmaları STM32F103C mikrodenetleyicisi üzerinde C dili ile implemente edildi.
12. **Alıcı Devre Tasarımı ve Komponent Seçimi:**
  - 40 kHz ve 37 kHz frekanslarını algılamak üzere iki ayrı kanal için TL084CN op-amp tabanlı bant geçiren filtre devreleri tasarlandı ve Proteus'ta simüle edildi.
  - Sinyal yükseltme ve kare dalgaya çevirme (LM741 ile) devreleri tasarlandı.
13. **STM32F103C Mikrodenetleyici ile Alıcı Devre Kodu ve Demodülasyon:** Gelen kare dalga'nın periyodunu ölçerek frekansı tespit eden ve orijinal veriyi çözümleyen algoritma STM32F103C üzerinde geliştirildi ve akış diyagramı oluşturuldu.
14. **Ultrasonik Alıcı-Verici Prototip Donanımının Geliştirilmesi ve Testi:** STM32F103C tabanlı, ultrasonik transdüser, LCD ekran ve sinyal şartlandırma devrelerini içeren bir prototip donanım geliştirildi. Bu prototip ile 30-40 kHz aralığında ultrasonik darbeler iletilip alınarak frekans tespiti ve bit dizisi çözümlemesi test edildi.
15. **Python Ursina ile AMKS Benzetim Modellemesi:** Üç vericiden gelen saat bilgisinin denizaltı tarafından alınıp işlenerek konumun (X, Y) hesaplandığı bir 3D benzetim modeli Ursina kütüphanesi kullanılarak geliştirildi.

#### 4. Sonuç:

Bu çalışmada, hedef su altı sisteminin konumlandırılmasını esnek ve güvenilir hale getiren bir prototip tasarlandı. Yeteneği kısmen deneylerle test edilmiştir. Prototip



donanımının yeni işlevsellikleri uyguladığı, STM32 tabanlı olduğu, ultrasonik darbeleri ilettiği frekanslarda ölçüm yapabildiği, frekansları ayırt edebildiği ve yüksek frekanslı olanları ayıkladığı küçük paket dizelerini kod çözebildiği yüksek seviye testlerle doğrulanmıştır.

İlk testler, filtre ve amplifikatör devresinin gerekli sinyal dalgasını sağladığını göstermektedir (ses frekansı aralığında; 1-20 kHz olan Manchester algoritmasıyla kodlanmış sinyalleri alıcı kısmında çözebilir). Duyulabilir frekans aralığındaki uygulamalara yayılma özellikleri çalışmalarının tahsisleri, farklı frekans bantlarının mesafe algılanabilirliği, etkisi ve performansında farklılıklar olduğunu göstermiştir.

Tüm gerçek zamanlı çalışmalar ve testler, mevcut imkânlar çerçevesinde hava ortamında gerçekleştirilmiş olsa da, su altı verileri için daha önce uygulanmış yöntemler detaylı bir şekilde incelenmiştir. Bu yöntemlerden elde edilen veri setleri esas alınarak, Python kullanılarak gerçeğe yakın benzetimler yapılmış ve bu doğrultuda çalışmalar yürütülmüştür. Günümüzde su altı akustik modemlerin geliştirilmesine yönelik çabalar artarken, bu projenin gerçekleştirilen testler ve benzetimler ışığında yüksek uygulanabilirlik potansiyeline sahip olduğu ve daha da geliştirilebileceği ortaya konmuştur.

Mevcut çalışmalar ve LBL (Long Baseline), SBL (Short Baseline) ve USBL (Ultra-Short Baseline) gibi sistemler, ses dalgalarının uygun koşullarda su altında yaklaşık 1500 m/s hızla, onlarca kilometre mesafeye yayılabildiğini göstermektedir. Bu durum, geliştirilen sistemin su altında uygulanabilirliği için güçlü bir temel sunmaktadır. Ayrıca, 2024 yılında yayımlanan “An Underwater Passive Electric Field Positioning Method Based on Scalar Potential” adlı makalede önerilen yöntemin uyarlanmasıyla, su altında, karadaki gibi ışık hızında ve uzun mesafelere veri iletimi potansiyelinin mümkün olabileceği değerlendirilmektedir. Bu, sistemin yenilikçi bir yaklaşımla su altı iletişimde yüksek potansiyele sahip olduğunu ortaya koymaktadır.

## 5. Çıktılar (Yayınlar, sunumlar vb.)

Projenin modellenmesi ve ilk örnek testlerinden elde edilen verilerin de eklenmesiyle makalenin son halinin yazılması aşaması danışman hocam Doç. Dr. Ebru Karaköse gözetimi ve akademik desteğiyle devam etmektedir. Bu makalenin temel amacı, sualtı sistemleri veya sinyal işleme alanında hakemli bir mühendislik dergisinde yayımlanmasıdır.

Şu an elimizde kullanılan STM32 tabanlı bir alıcı-verici devresi, ultrasonik ses verileri gönderip alabilmektedir ve hem donanım hem de yazılım açısından test edilmiştir. Bu proje kapsamında geliştirilen Python simülasyonlarının ve STM32 gömülü sistem kodlarının kaynak dosyaları, sırasıyla, [https://github.com/OmerFaruk-Celik/AMKS\\_AUV\\_ATILAY](https://github.com/OmerFaruk-Celik/AMKS_AUV_ATILAY) adresinde paylaşılmıştır..

## 6. Proje ile ilgili harcama kalemleri hakkında ayrıntılı bilgi

Projenin araştırma, geliştirme ve prototipleme aşamalarında ihtiyaç duyulan donanım, elektronik bileşenler ve sarf malzemeler aşağıda fatura ve sipariş bilgileriyle birlikte detaylı olarak verilmiştir. Listede faturalandırılmış ana kalemlerin yanı sıra, prototipleme sürecinde esneklik sağlamak ve sonradan ortaya çıkan ihtiyaçları gidermek amacıyla temin edilen ek bileşenler ve sarf malzemelerin fiyatları da yazılmıştır

### 1. Mikrodenetleyici ve Geliştirme Kartları:

- **STM32F103C8T6 Geliştirme Kartı Modülü (2 Adet):** Prototipin alıcı ve verici devrelerinin beyin takımını oluşturan ana işlemci birimi.
  - *Tutar: 316,70 TL (Sipariş No: 402-6384954-3225918)*
- **DollaTek NRF52832 Bluetooth Modülü (1 Adet):** İleriki aşamalarda kablosuz veri aktarımı ve kontrol için planlanan modül.
  - *Tutar: 549,28 TL (Sipariş No: 402-5141069-1021946)*

### 2. Test ve Ölçüm Ekipmanları:

- **Dijital Osiloskop Kiti (1 Adet):** Sinyal analizi, dalga formu görüntüleme ve devre hata ayıklama işlemleri için temel ölçüm cihazı.
  - *Tutar: 5.388,09 TL (Sipariş No: 402-2488929-2396334)*
- **USB Logic Analyzer 24M 8CH (1 Adet):** Dijital sinyallerin ve haberleşme protokollerinin (SPI, I2C vb.) analizi için kullanılmıştır.
  - *Tutar: 459,64 TL (Fatura No: 20250109672)*

### 3. Aktif Devre Elemanları (Op-amplar, IC'ler):

- **74LS158 DIP-16 Multiplexer Entegrasi (4 Adet):** Sinyal yönlendirme ve seçim devrelerinde kullanılmıştır.
  - *Tutar: 351,88 TL (Sipariş No: 402-0425003-1623518)*
- **LM741CN DIP-8 Operasyonel Amplifikatör (10 Adet):** Özellikle kare dalga çevirici ve temel sinyal yükseltme devrelerinde kullanılmıştır.
  - *Tutar: 144,61 TL (Sipariş No: 330324199311020215)*
- **TL084CN DIP14 Operasyonel Amplifikatör (10 Adet):** Düşük gürültü özelliği sayesinde bant geçiren filtre ve sinyal yükseltme katmanlarında tercih edilmiştir.
  - *Tutar: 148,87 TL (Sipariş No: 330324199311020215)*

### 4. Pasif Devre Elemanları:

- **BOJACK Direnç Çeşitleri Kiti (1000 Adet):** Filtre ve sinyal şartlandırma devrelerinde kullanılan çeşitli değerlerde dirençler.
  - *Tutar: 293,66 TL (Sipariş No: 402-5141069-1021946)*
- **Walfront Elektronik Bileşen Seti (1390 Parça):** Projede ihtiyaç duyulan çeşitli direnç, kondansatör, diyot ve transistörleri içeren kapsamlı set.
  - *Tutar: 734,43 TL (Sipariş No: 402-7262695-0602723)*

### 5. Akustik ve Elektromekanik Bileşenler:

- **Yuvarlak Rezonans Hoparlör (Ultrasonik Transdüser) (1 Paket):** Ses dalgalarını üretmek ve algılamak için kullanılan ana bileşen.

- Tutar: 604,89 TL (Sipariş No: 402-9064251-3897135)

#### 6. Prototipleme, Sarf Malzemeleri ve Ek Bileşenler (Faturasız Olarak Giden Harcamalar):

- **Isı Ayarlı Havya İstasyonu (1 Adet):** Devre elemanlarının PCB ve breadboard üzerine montajı için temel lehimleme ekipmanı.
  - Tutar: 750,00 TL
- **Lehim Teli ve Lehim Pastası:** Lehimleme işlemleri için gerekli sarf malzemeleri.
  - Tutar: 170,00 TL
- **9V Alkalın PİL (4'lü Paket):** Prototipin taşınabilir testleri ve op-amp devrelerinin beslemesi için güç kaynağı.
  - Tutar: 250,00 TL
- **Voltaj Regülatörleri (7805, LM317):** 9V pilden elde edilen gerilimi, devrelerin ihtiyaç duyduğu kararlı 5V seviyesine düşürmek ve ayarlanabilir gerilim sağlamak için.
  - Tutar: 220,00 TL
- **830 Delikli Breadboard (2 Adet):** Devrelerin lehimsiz olarak kurulması ve test edilmesi için kullanılan platform.
  - Tutar: 180,00 TL
- **Jumper Kablo Seti (E/E, E/D, D/D):** Breadboard üzerinde devre elemanları arası bağlantıyı sağlamak için.
  - Tutar: 220,00 TL
- **Seramik "Mercimek" Kondansatör Seti:** Filtre devrelerinde ve osilatörlerde yaygın olarak kullanılan (100nF, 22pF, 10nF vb.) çeşitli değerlerde kondansatörler.
  - Tutar: 250,00 TL
- **PCB Prototipleme ve Baskı Devre Hizmeti:** Tasarlanan son alıcı ve verici devrelerinin daha stabil çalışması için PCB üzerine basılması.
  - Tutar: 500,00 TL

**PROJE TOPLAM MALİYETİ: 11.452,05 TL**

PROJE YÜRÜTÜCÜSÜNÜN ADI – SOYADI - İMZA	DANIŞMANIN ADI – SOYADI - İMZA
Ömer Faruk Çelik	Ebru Karaköse
	



Tarih :20/06/2025

**Ek1:**  
**Kaynakça**

- [1] Sakınç, M. (2024). Okyanusların keşfi ve HMS Challenger. *Deniz Bilimleri ve Teknolojileri Dergisi*, 15(2), 32-43.
- [2] Bayatlı, A. (2022). Osmanlı ve Kutup Stratejisinin Haritacılık Tarihi ve Coğrafi Keşifler Perspektifinden İncelenmesi. *Trakya Üniversitesi Kalite ve Strateji Yönetimi Dergisi*, 2(1), 39-91.
- [3] Stutters, L., Liu, H., Tiltman, C., & Brown, D. J. (2008). Navigation technologies for autonomous underwater vehicles. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, 38(4), 581-589.
- [4] Wu, P., Nie, W., Liu, Y., & Xu, T. (2024). Improving the underwater navigation performance of an IMU with acoustic long baseline calibration. *Satellite Navigation*, 5(7), 1-15.
- [5] Academy of Positioning Marine and Bathymetry. (2024). Underwater acoustic positioning system (Scientific Manifest 2). APOMAB Technical Reports.
- [6] Zhang, Y., Chen, C., Sun, J., Qiu, M., & Wu, X. (2024). An underwater passive electric field positioning method based on scalar potential. *Mathematics*, 12(12), 1832.
- [7] Ru, X., Gu, N., Shang, H., & Zhang, H. (2022). MEMS inertial sensor calibration technology: Current status and future trends. *Micromachines*, 13(6), 879.
- [8] Yang, Y., & Huang, G. (2017). Acoustic-inertial underwater navigation. *IEEE International Conference on Robotics and Automation (ICRA)*, 4927-4933.
- [9] Mercer, J. A., & Schulkin, M. (1985). Colossus revisited: A review and extension of the Marsh-Schulkin shallow water transmission loss model. *Journal of the Acoustical Society of America*, 77(2), 456-464.
- [10] Leens, F. (2009). An introduction to I2C and SPI protocols. *IEEE Instrumentation & Measurement Magazine*, 12(1), 8-13.
- [11] STMicroelectronics. (2015). STM32F103x8, STM32F103xB Medium-density performance line ARM-based 32-bit MCU technical reference manual.
- [12] Sun, D., Gu, J., Han, Y., & Zhang, J. (2019). Inverted ultra-short baseline signal design for multi-AUV navigation. *Ocean Engineering*, 185, 133-143.
- [13] Texas Instruments. (2024). TL084, TL084A, TL084B Quad operational amplifiers [Data sheet].

- [14] Arslan, E., Doğukan, A. T., & Başar, E. (2020). Orthogonal frequency division multiplexing with codebook index modulation. IEEE Transactions on Signal Processing, 68, 1-14.
- [15] Karagün, V., & Taşdemir, E. (2019). Benford yasasının denetimde kullanımı ve bir uygulama. Muhasebe ve Finansman Dergisi, 82, 247-262.
- [16] Türkyener, C. M. (2007). Benford yasası ve mali denetimde kullanımı. Sayıştay Dergisi, 64, 111-122.
- [17] Hitit Üniversitesi. (2024). Deney-11: Frekans demodülasyonu. Elektrik-Elektronik Mühendisliği Laboratuvar Föyü.
- [18] Zheng, Z., Fu, Y., Liu, K., Xiao, R., Wang, X., & Shi, H. (2018). Three-stage vertical distribution of seawater conductivity. Ocean Science, 14(3), 521-530.
- [19] Hitit Üniversitesi. (2024). OP 2300 - OP AMP deney seti föyü. Elektrik-Elektronik Mühendisliği Laboratuvar Dokümanları.
- [20] Yanık, R., & Samancı, T. H. (2013). Benford kanunu ve muhasebe verilerinde uygulanmasına ait kamu sektöründe bir uygulama. Mali Çözüm Dergisi, 115, 17-33.
- [21] Korkmaz, S. (t.y.). SES FİZİĞİ. Atatürk Üniversitesi. Erişim adresi: <https://avesis.atauni.edu.tr/korkmaz.selcuk/indir>
- [22] Farklı Sıcaklıklarda Havadaki Ses Hızı Kaynaklar: OpenStax College. (2012). College Physics. [http://cnx.org/contents/031da8d3-b525-429c-80cf-6c8ed997733a/College\\_Physics](http://cnx.org/contents/031da8d3-b525-429c-80cf-6c8ed997733a/College_Physics);
- [23] Lumen Learning. (n.d.). 17.2 Sesin Hızı, Frekans ve Dalga Boyu. SUNY Physics. Erişim tarihi: 17 Mayıs 2025, <https://courses.lumenlearning.com/suny-physics/chapter/17-2-speed-of-sound-frequency-and-wavelength/> ve yazarın hesaplamaları