Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi
Afyon Kocatepe University – Journal of Science and Engineering
https://dergipark.org.tr/tr/pub/akufemubid

Araştırma Makalesi / Research Article
e-ISSN: 2149-3367
DOI: https://doi.org/10.35414/akufemubid.1645113
AKÜ FEMÜBİD **25** (2025) 065202 (1372-1385)
AKU J. Sci. Eng. **25** (2025) 065202 (1372-1385)

Aviyonik Video Aktarımında Python ile ARINC 818 Simülasyon ve Analizi

Makale Bilgisi / Article Info
Alındı/Received: 23.02.2025
Kabul/Accepted: 01.06.2025
Yayımlandı/Published: 03.12.2025

Simulation and Analysis of ARINC 818 in Avionic Video Transmission Using Python

Muhammed Ali BELGRAT*, Ömer KARAL

Ankara Yıldırım Beyazıt Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği Bölümü, Ankara, Türkiye

© 2025 The Authors | Creative Commons Attribution-Noncommercial 4.0 (CC BY-NC) International License

Öz

Makale, aviyonik sistemlerde yüksek bant genişliği ve düşük gecikme süresi gibi avantajlar sunan ARINC 818 (Aviyonik Dijital Video Veri Yolu - ADVB) standardının Python tabanlı bir simülasyon ortamı ile modellenmesini sunmaktadır. Hali hazırda bu protokolün mevcut veya uygulanmış simülasyon modelleri bulunmamaktadır. Geliştirilen araç, kullanıcıdan alınan video parametrelerini ARINC 818 standardına uygun biçimde işleyerek model tabanlı bir simülasyon yaklaşımıyla standardın işleyişini göstermektedir. Bu simülasyon gerçek donanım üzerinde koşmak yerine, standardın paket yapısı ve veri akış mekanizmalarının yazılımsal olarak modellemesini sağlar, böylece standardın zamanlama mekanizmalarının ve veri yapılarının ayrıntılı biçimde analiz edilmesine olanak tanır. Etkileşimli ve görsel yapısı sayesinde mühendisler, araştırmacılar ve öğrenciler için hem öğretici hem de tasarım öncesi değerlendirme süreçlerinde kullanılabilecek esnek bir platform sunmaktadır. Bu yönüyle çalışma, ARINC 818'in anlaşılmasını kolaylaştıran ve mühendislik uygulamalarına katkı sağlayan öğretici bir kaynak niteliğindedir.

Anahtar Kelimeler: *ARINC 818, Aviyonik Dijital Video Veri Yolu, Python, Simülasyon ve analiz*

Abstract

This paper presents the modeling of the ARINC 818 standard (Avionics Digital Video Bus - ADVB), which offers advantages such as high bandwidth and low latency in avionics systems, through a Python-based simulation environment. Currently, there are no existing or implemented simulation models for this standard. The developed tool processes video parameters provided by the user in accordance with the ARINC 818 standard and demonstrates the standard's operation through a model-based simulation approach. Rather than running on actual hardware, the simulation provides a software-based model of the standard's packet structure and data flow mechanisms, enabling a detailed analysis of its timing characteristics and data structures. With its interactive and visual structure, the tool provides a flexible platform suitable for engineers, researchers, and students, serving both educational purposes and pre-design evaluation processes. In this regard, the study serves as an educational resource that facilitates understanding of the ARINC 818 standard and supports its application in engineering practices.

Keywords: *ARINC 818, Avionics Digital Video Bus, Python, Simulation and analysis*

1. Giriş

ARINC 818, ARINC ve havacılık topluluğu tarafından 2007 yılında ticari havacılığın daha yüksek performanslı dijital videoya yönelik acil ihtiyaçlarını karşılamak için geliştirilen bir standarttır. Bu düşük gecikmeli standart, fiber kanal arayüzüne dayalı yüksek bant genişliği ve sıkıştırılmamış video görüntüsü iletimine olanak verir. İlk olarak havacılık elektroniğinde, helikopterlerde ve kara araçlarında askeri görüntüleme sistemlerinde kullanım için incelenmiştir (Alexander ve Keller 2007). Havacılık elektroniğinde tercih edilmesinin başlıca nedenleri, uygulama kolaylığı, hızı ve ölçeklenebilirliği ve fiziksel protokolün uygulama protokolünden ayrılmasıdır. Standart ilk olarak Boeing tarafından 787 ticari uçak programında ve Airbus tarafından A400M programında benimsenmiştir (Bisson 2007). Yayımlandığı tarihten bu yana Airbus A350, C-130 AMP, C-17, F15, F18 yükseltme programları ve diğer birçok ticari ve askeri uçaktaki kokpit ekranları için video taşıma standardı olarak kullanılmıştır (İnt. Kyn. 1). Bu süreçte kullanım ihtiyaçları ve öneriler doğrultusunda anahtarlama, kanal bağlama, çift yönlü iletişim, daha yüksek hızlarda veri aktarımı ve şifreleme ve veri sıkıştırmaya karşı alınabilecek önlemler konularında revizyonlar yapılmış ve ARINC 818 Supplement-2 yayımlanmıştır (Grunwald 2014). Bu gelişmelerle birlikte başlangıçta kokpit ekranları için tasarlanan bu standart, düşük gecikme ve yüksek hız gibi gereksinimlerin gerektiği kızılötesi, optik sensör ve kameralarda da kullanılmaya başlanmıştır (Keller ve Grunwald 2014, Zimmerman 2017). Teknolojinin ilerlemesiyle birlikte, aviyonik

*Sorumlu Yazar/Corresponding Author: Muhammed Ali BELGRAT e-posta/e-mail: belgrat65@gmail.com

sistemlerde yaygın olarak kullanılan ARINC 818 standardı, güncel kalabilmek için önerilen yeniliklerin değerlendirilmesi sonrasında ARINC komitesi tarafından 2018 yılında güncellenmiştir. ARINC 818 standardına yönelik bu son güncelleme (Supplement-3), test için görüntü emülasyon modu, gecikme hesaplamaya yönelik kılavuz bilgileri ve en önemlisi de 10 Gbit'in üzerindeki hızlara yönelik 64/66 Bit kodlamanın kullanılmasına dair yönergeler içerir (Alexander ve Grunwald 2019). ARINC 818, bu yenilikler sayesinde yalnızca bir görüntü arayüzü değil, aynı zamanda kızılötesi ve optik sensörlerden işlemcilere ve ekranlara kadar tüm görev sistemleri arasında bir sistem seviyesi video veri yolu olarak da öne çıkmaktadır (Keller ve Finnegan 2024).

ARINC 818 standardı, modern aviyonik sistemlerdeki yüksek bant genişliği ve düşük gecikme gereksinimlerini başarıyla karşılasa da uygulama ve test süreçlerinde çeşitli zorluklarla karşılaşmaktadır. Örneğin, Krotov (2024) çalışmasında, elektromanyetik parazit (HIRF) gibi zorlu ortam koşullarında protokolün güvenilirliğini artırmak amacıyla ileri düzey hata düzeltme yöntemlerinin (LDPC-IRA) kullanımını önermiştir. Bu tür çevresel etkenlerden kaynaklanan veri güvenliği problemlerinin yanı sıra, sistemin tasarımı sırasında ortaya çıkabilecek zamanlama ve senkronizasyon hataları da kritik öneme sahiptir. Özellikle, ARINC 818'in esnek yapısı ve senkronizasyon gereksinimleri, tasarım aşamasında dikkat edilmesi gereken karmaşık zamanlama mekanizmalarını beraberinde getirir. Bu bağlamda yapılan tasarım hataları, proje maliyeti ve takviminde aksamalara yol açabileceği gibi, sistem güvenliği ve hata raporlama süreçlerinde de ek zorluklar ortaya çıkarabilir. Zhang vd. (2025), ARINC 818 üzerinden alınan video sinyalini HDMI formatına dönüştürerek görüntü tanıma temelli bir arayüz test ortamı oluşturmuştur. Bu gibi uygulamalarda görüntülerin doğru zamanda yakalanması ve işlenmesi, senkronizasyonun sağlanmasına doğrudan bağlıdır. Bu nedenle, ARINC 818 tabanlı sistemlerin tasarımı, zamanlama kısıtlamalarını ve senkronizasyon gereksinimlerini karşılayacak şekilde titizlikle planlanmalıdır. Bu noktada, fiziksel ekipmanlara geçmeden önce yapılan simülasyonlar, tasarım sürecinde olası hataları tespit etme ve çözme imkânı sunarak bu problemlerin etkisini en aza indirmek için etkili bir araç haline gelir (Gadde 2024).

Yukarıda değinilen zamanlama kısıtlamaları ve senkronizasyon gereksinimleri kapsamında küresel ölçekte ARINC 818 protokolünün test ve simülasyonuna yönelik çeşitli ticari çözümler geliştirilmiştir. Örneğin, Great River Technology, TECHWAY, iWave Systems, Logic Fruit Technologies ve New Wave Design gibi firmalar; ARINC 818 tabanlı sistemlerin geliştirilmesi, doğrulanması ve test edilmesine yönelik PCIe/XMC kartları, protokol analizörleri, video dönüştürücüler, anahtarlar ve yazılım geliştirme kitleri gibi çok çeşitli donanım ve yazılım araçları sunmaktadır (İnt. Kyn. 2, İnt. Kyn. 3, İnt. Kyn. 4, İnt. Kyn. 5, İnt. Kyn. 6).

Bu çözümler çoğunlukla donanım temellidir ve sistem seviyesinde test imkânları sağlar. Ancak bu tür ticari çözümler oldukça maliyetlidir ve özel donanım gerektirirler. Diğer yandan sistemin donanımsal olarak test edilmesinden önce yazılımsal olarak analiz edilmesi ihtiyaç fazlası donanımların alınmasının önüne geçerek zaman ve maliyet açısından kazanç sağlayacaktır. Bundan dolayı özellikle gereksinime yönelik donanımı belirleyebilecek yazılım tabanlı simülasyon araçlarına duyulan ihtiyaç hâlâ devam etmektedir. Ayrıca yazılım tabanlı simülasyon araçları, özellikle düşük bütçeli araştırma projeleri ve gerekli donanım sisteminin ön tasarım aşamalarında donanım tabanlı ticari ürünlerin aksine farklı senaryoların detaylı analizine olanak tanıyabilir. Bu doğrultuda, çalışmamızda ARINC 818 protokolü temelinde farklı senaryolar üzerinde analiz yapılabilmesini sağlayacak yazılım tabanlı bir simülasyon ortamı geliştirilmiştir.

Aviyonik sistemlerdeki bu tür karmaşık tasarım ve test gereksinimleri, literatürde farklı protokollere yönelik çeşitli simülasyon temelli yaklaşımların geliştirilmesine neden olmuştur. Örneğin, Bilge (2014) tarafından İstanbul Teknik Üniversitesi'nde gerçekleştirilen "Entegre Modüler Aviyonikler için OMNET++ Simülasyon Modeli" başlıklı yüksek lisans tezinde, aviyonik gerçek zamanlı işletim sistemi (RTOS) standardı olan ARINC 653 ve AFDX gibi kritik yazılım uygulamaları için zaman ve bellek bölümlendirmesi sağlayan protokollere dayalı entegre modüler aviyonik (IMA) sistemleri OMNET++ ortamında modellenmiş ve sistem performansı fiziksel donanım gerekmeden analiz edilmiştir. Bu çalışmada ise video veri iletim protokolü olan ARINC 818 protokolüne özgü zamanlama ve veri yapısı detaylı olarak analiz edilmektedir.

Python programlama dili son yıllarda oldukça geniş kullanım alanı bulmuş ve farklı disiplinlerde yaygın olarak uygulanmıştır. Örneğin, Değirmenci ve Karal (2022) aykırı değer tespitinde, Kılıç vd. (2022) biyomedikal görüntü analizinde, Yalman vd. (2022) elektrik şebekesinde voltaj gecikmesinin tahmininde, Aslan vd. (2024) insansı robotların nesne manipülasyonunu öğrenmesinde, Aslan vd. (2024) ayak bileği stratejisi kullanarak Robotis-OP2 için itme kurtarma kontrol yöntemlerinin karşılaştırılmasında Python programlama dilini kullanmıştır.

Bu çalışma, ARINC 818 standardı için Python tabanlı bir simülasyon ve analiz aracı sunarak bu zorluklara çözümler sağlamayı amaçlamaktadır. Geliştirilen simülasyon ve analiz ortamı, kullanıcıların ARINC 818 standardının farklı senaryolardaki davranışını gözlemlemelerine ve standardın zamanlama mekanizmalarını ve veri yapısını ayrıntılı olarak analiz etmelerine olanak tanır, böylece hem öğrenme hem de tasarım doğrulama süreçlerini destekler. Bu şekilde, mühendisler ve araştırmacılar sistem tasarımında daha bilinçli kararlar alabilir ve fiziksel testlere geçmeden önce daha sağlam bir temel oluşturabilirler. Çalışma, ARINC 818 standardının eğitim, analiz ve tasarım süreçlerine önemli bir katkı sağlamayı amaçlamaktadır.

2. ARINC 818 Standardının Anlaşılması

ARINC 818, modern aviyonik sistemlerde kritik görev uygulamalarda veri iletimi için geliştirilen, fiber kanal tabanlı bir dijital video standardıdır. Ticari ve askeri havacılık uygulamalarında baş aşağı göstergeler (Head-Down Display, HDD), baş üstü göstergeler (Head-Up Display, HUD), radarlar, video anahtarlama sistemleri ve kızılötesi sensörler gibi yüksek çözünürlük ve düşük gecikme gerektiren sistemlerde yaygın olarak kullanılır (Wood ve Howells 2015). Standart, yüksek bant genişliğine sahip sıkıştırılmamış dijital videonun taşınmasını sağlarken, senkronizasyonu ve veri bütünlüğünü korumada kritik bir rol oynar. Ayrıca, esnek yapısı sayesinde farklı çözünürlükler, kare hızları ve video formatlarıyla çalışabilir ve bu özelliğiyle hem günümüz ihtiyaçlarına hem de geleceğin teknolojilerine uygun bir altyapı sunar. ARINC 818'in temel mimarisi, veri aktarımı sırasında düşük gecikme sağlamak üzere optimize edilmiştir ve gerçek zamanlı video uygulamaları için özel olarak tasarlanmıştır. ARINC 818'in bu geniş uygulama alanları ve teknik özellikleri, standart zaman içinde sürekli geliştirilmesini ve güncellenmesini gerektirmiştir.

2.1 ARINC 818'in Gelişimi ve Tarihçesi

İlk olarak 2007'de yayınlanan standart, öncelikle sıkıştırılmamış video veri iletimi için optimize edilmiştir. Yayınlanmasından sonra, kokpit ekranlarında yüksek kaliteli video iletimi sağlamak için kullanılmıştır. İlk uygulamalarda, standardın esnekliği farklı çözünürlüklerde ve kare hızlarında video iletimi sağlayan bir yapı sağlamıştır. Bu özellik, Boeing 787 ve Airbus A350 gibi ticari projelerde standart haline gelmesine katkıda bulunmuştur.

ARINC 818, sektörün değişen ihtiyaçlarına uyum sağlamak amacıyla zaman içinde çeşitli güncellemelerle geliştirilmiştir. 2013 yılında yayımlanan Supplement 2, standardın esnekliğini ve kapasitesini artıran önemli yenilikler sunmuştur. Bu güncelleme; kanal birleştirme, çift yönlü iletişim, video sıkıştırma desteği ve daha yüksek veri aktarım hızları gibi özellikleri içermektedir. Özellikle kanal birleştirme yeteneği, birden fazla veri akışının tek bir bağlantı üzerinden taşınmasını sağlayarak daha karmaşık sistemlerin entegrasyonunu kolaylaştırmıştır. Ayrıca, sıkıştırma desteği veri yoğun uygulamalarda bant genişliği kullanımını optimize etmiştir. Bu iyileştirmeler, ARINC 818'in yalnızca kokpit ekranları için değil, radar ve hareketli harita sistemleri gibi veri yoğun uygulamalarda da kullanılmasına olanak tanımıştır.

2018'de yayımlanan Supplement 3 ile gelen yenilikler, modern aviyonik sistemlerde yalnızca daha hızlı veri iletimini sağlamakla kalmamış, aynı zamanda sistem entegrasyonu ve doğrulama süreçlerini daha esnek ve güvenilir hâle getirmiştir. Örneğin, 64/66 bit kodlama, yüksek hızlarda veri iletim doğruluğunu artırarak hata tespitinde etkinlik sağlamış; test amaçlı görüntü emülasyonu ise tasarım sürecinde zamanlama ve senkronizasyon hatalarını önleyerek sistem doğrulamasını kolaylaştırmıştır. Ayrıca, gecikme hesaplamalarına yönelik sağlanan kılavuz bilgiler, kritik sistem tasarımında daha hassas zamanlama ve senkronizasyon gereksinimlerini karşılamaya yardımcı olmuştur.

Bu yenilikler, ARINC 818'i yalnızca bir veri aktarım standardı olmaktan çıkarıp, modern aviyonik sistemlerin güvenilirliği ve verimliliği açısından bir standart hâline getirmiştir. Güncellemelerle gelen gelişmeler, standardın modern aviyonik sistemlerdeki önemini artırmış ve kullanım alanlarını genişletmiştir. Düşük gecikme süresi, yüksek bant genişliği, esnek veri yapısı ve güçlü hata kontrol mekanizmaları gibi özellikleri, ARINC 818'i ticari ve askeri uygulamalarda kritik bir standart hâline getirmiştir.

Aşağıdaki bölümde, ARINC 818'i diğer standartlardan ayıran bu teknik özellikler detaylı şekilde ele alınmaktadır.

2.2 ARINC 818 Standardının Temel Teknik Özellikleri

ARINC 818, aviyonik sistemlerde karmaşık veri aktarım ihtiyaçlarını karşılamak amacıyla geliştirilmiş, esnek ve ölçeklenebilir bir standarttır. Bu bölümde, standardın yapısı ve teknik detayları incelenerek çeşitli bileşenleri ve özellikleri ele alınacaktır.

Gecikme: Pilotların gerçek zamanlı kararlar almasını gerektiren aviyonik sistemlerde, görüntülerin hızlı ve doğru işlenmesi hayati önem taşır. HDD, HUD ve başa takılan ekranlar (HMD) gibi sistemlerde gecikme süresi, pilotun dış dünyayı video görüntüsüyle eş zamanlı algılayabilmesi için kritik bir parametredir. Yüksek gecikme süresi, görüntüler ile gerçek dünya arasında

zaman farkı oluşturarak pilotun doğru ve hızlı karar almasını zorlaştırabilir. Örneğin, iniş sırasında HUD ekranındaki bir gecikme, pistin gerçek konumu ile görüntülenen konumu arasında tutarsızlık yaratabilir ve güvenliği riske atabilir. Benzer şekilde, hedef izleme ve kızılötesi görüntüleme gibi kritik görevlerde yüksek gecikme, tehditlerin veya hedeflerin zamanında tespit edilmesini engelleyebilir. ARINC 818 standardı, bu gereksinimlere uygun olarak tasarlanmış olup düşük gecikme süreleriyle sistem performansını optimize eder.

Fiber Kanal Mimarisi ile Yüksek Ban Genişliği Özellikleri: ARINC 818, yüksek çözünürlüklü sıkıştırılmamış dijital video ve sensör verilerinin güvenilir iletimi için yüksek bant genişliği sunar. 4K ve 8K çözünürlükteki video akışlarını destekleyerek modern aviyonik sistemlerin artan veri ihtiyaçlarını karşılar. Özellikle, gerçek zamanlı görüntü işleme ve çoklu sensör verilerinin eş zamanlı aktarımı gibi veri yoğun uygulamalar için optimize edilmiştir. Fiber Channel tabanlı yapısı, yüksek veri aktarım hızlarını mümkün kılarken düşük gecikme süreleri ve güvenilir bağlantılar sunar. ARINC 818'in farklı hızlardaki veri aktarım kapasitesine ilişkin detaylar, Çizelge 1'de gösterilmektedir(Grunwald 2013). Ayrıca, çizelgede belirtilen 8B/10B ve 64B/66B kodlama teknikleri sayesinde hata tespiti ve veri bütünlüğü artırılarak sistemin genel güvenilirliği güçlendirilmiştir.

Çizelge 1. Desteklenen hızlar.

Bit Hızı (Gbps)	Gerekli Fiber Kanal Hızı
1.0625	FC 1x Hız (8b/10b Şifreleme)
1.5	
1.62	
2.125	FC 2x Hız (8b/10b Şifreleme)
2.5	
3.1875	FC 3x Hız (8b/10b Şifreleme)
4.25	FC 4x Hız (8b/10b Şifreleme)
5.0	
6.375	FC 6x Hız (8b/10b Şifreleme)
8.5	FC 8x Hız (8b/10b Şifreleme)
12.75	FC 12x Hız (8b/10b Şifreleme)
14.025	FC 16x Hız (64b/66b Şifreleme)
21.0375	FC 24x Hız (64b/66b Şifreleme)
28.05	FC 32x Hız (64b/66b Şifreleme)

Hata Tespiti ve Güvenlik: ARINC 818, veri iletiminin doğru ve güvenilir şekilde gerçekleşmesini sağlamak için hata tespiti ve güvenlik mekanizmalarına sahiptir. Döngüsel artıklık denetimi (Cyclic Redundancy Check, CRC), aktarım sırasında oluşabilecek hataları tespit ederek veri bütünlüğünü korurken, şifreleme verilerin yalnızca yetkili alıcılar tarafından okunmasını ve kullanılmasını güvence altına alır.

CRC Algoritması: ARINC 818'de hata tespiti için CRC algoritması kullanılmaktadır. Bu matematiksel algoritma, veri iletimi sırasında oluşabilecek hataları tespit etmeye yönelik çalışır. ADVB protokolünde hem veri çerçevelerinin hem de görüntünün bütünlüğünü ve doğruluğunu sağlamak için farklı seviyelerde CRC algoritması uygulanmaktadır.

- **Çerçeve Bazlı CRC (Frame CRC):** Her çerçeve, başlık ve yük veri alanları üzerinden hesaplanan bir CRC kodu içerir. Bu kod, her ADVB çerçevesinin bütünlüğünü doğrular ve iletim sırasında oluşabilecek hataları tespit eder. Hatalı çerçeveler, sistemde bozuk verilerin yayılmasını önlemek amacıyla genellikle varsayılan bir değerle (örneğin, siyah piksel) değiştirilir.
- **Görüntü Bazlı CRC (Image CRC):** Mevcut görüntünün tamamının veya belirli bölümlerinin bütünlüğünü doğrulamak için kullanılır. Bu CRC hem ekrandaki hem de ekran dışı piksel verilerini kapsar.
- **Önceki Görüntü CRC (Prior Image CRC):** Daha önce iletilmiş bir görüntünün veya parçalı bir görüntünün yeniden oluşturulmuş hâlinin bütünlüğünü doğrulamak için kullanılır.

Bu üç CRC türü, ADVB protokolü kapsamında birbirini tamamlayan işlevler sunar. Çerçeve CRC, veri iletiminin temel birimi olan çerçevelerin bütünlüğünü sağlarken, Önceki Görüntü CRC geçmiş verilerin doğruluğunu kontrol eder. Görüntü CRC ise mevcut görüntünün tam veya kısmi bütünlüğünü doğrular. Bu mekanizmalar, ADVB'nin yüksek güvenilirlik ve sağlamlık standartlarına ulaşmasını sağlar.

Veri Güvenliği ve Şifreleme: ARINC 818, hassas verilerin iletimi sırasında güvenliği artırmak için şifreleme yöntemlerini destekler. Bu mekanizmalar, verilerin yalnızca yetkili sistemler tarafından erişilmesini ve kullanılmasını sağlayarak, özellikle uçuş kritik uygulamalarda önemli bir güvenlik katmanı oluşturur. Şifreleme, standardın diğer özellikleriyle entegre şekilde çalışarak iletilen verinin hem gizliliğini hem de bütünlüğünü korur.

Video ve ses gibi hassas veriler, yetkisiz erişimi ve müdahaleyi önlemek amacıyla şifrelenerek iletilir. Kullanılan şifreleme algoritmaları, belirli uygulama gereksinimlerine göre Arayüz Kontrol Dokümanı (Interface Control Document, ICD) aracılığıyla özelleştirilebilir. Bu esneklik, sistem tasarımcılarının operasyonel ortamlarına en uygun şifreleme yöntemlerini seçmelerine olanak tanır. Örneğin, kokpit ekranlarında gösterilen kritik uçuş bilgileri, verilerin kesintiye uğramasını veya manipüle edilmesini önlemek için şifrelenebilir. Benzer şekilde, askeri uçaklardaki sesli

iletişimler, yetkisiz dinlemelere karşı şifrelenmiş iletim yöntemleri kullanılarak korunabilir.

ARINC 818, sıkıştırma ve şifreleme yöntemlerinin bir arada kullanılmasını da destekler. Sıkıştırma, veri boyutunu küçülterek yüksek çözünürlüklü görüntülerin veya büyük veri setlerinin daha verimli iletilmesini sağlarken, şifreleme, bu sıkıştırılmış verilerin güvenliğini garanti altına alır. Özellikle uçuş kritik uygulamalarda, bu iki yöntemin birlikte kullanılması hem bant genişliği tasarrufu sağlar hem de hassas verilerin gizliliğini ve bütünlüğünü korur.

Örneğin, HUD gibi yüksek çözünürlüklü video akışlarının hem güvenlikten ödün vermeden hem de verimli bir şekilde iletilmesi, bu kombinasyon sayesinde daha etkili hâle getirilebilir. Ayrıca, sıkıştırılmış verilerin şifrelenmesi, sınırlı bant genişliğine sahip ortamlarda dahi güvenli ve optimize edilmiş veri aktarımına olanak tanır.

Zamanlama Hassasiyeti ve Modüler Ölçeklendirilebilirlik: ARINC 818, modern aviyonik sistemlerin gereksinimlerini karşılamak üzere hassas zamanlama, ölçeklenebilirlik ve esneklik sunan bir standart olarak öne çıkar. Hassas senkronizasyon yetenekleri, özellikle uçuş kontrol ekranları, sensör veri akışları ve diğer zaman duyarlılığı yüksek uygulamalarda kritik bir rol oynar. Ayrıca, standardın modüler ve ölçeklenebilir yapısı, farklı sistem ihtiyaçlarına uyum sağlayarak gelecekteki teknolojik yeniliklerin entegrasyonunu kolaylaştırır.

Kanal Yönetimi ve Anahtarlama: ARINC 818, veri aktarımı sırasında esneklik ve verimlilik sağlamak için kanal yönetimi ve anahtarlama mekanizmalarını etkin bir şekilde kullanır. Kanal birleştirme (channel bonding) yöntemi, birden fazla bağlantının aynı anda kullanılmasını sağlayarak daha yüksek bant genişliği gereksinimlerini karşılar. Bu yöntem, büyük veri setlerinin paralel bağlantılar üzerinden bölünerek iletilmesine olanak tanır ve böylece yüksek çözünürlüklü video akışları veya yoğun veri trafiğinin güvenilir bir şekilde aktarılmasını mümkün kılar. Kanal birleştirme, özellikle 4K ve 8K video gibi yüksek bant genişliği talep eden uygulamalarda kritik bir rol oynar.

ARINC 818'in sunduğu bir diğer önemli özellik ise anahtarlama (switching) yeteneğidir. Standart, çoklu kanal sistemlerinde veri akışlarının senkronize edilmesini ve yönlendirilmesini kolaylaştırır. Anahtarlama işlemleri, özellikle dikey boşluk (vertical blanking) periyodunda gerçekleştirilerek veri kaybı ve gecikme minimuma indirilir. Bu özellik, birden fazla veri kaynağının aynı hedefe yönlendirilmesi veya bir kaynağın birden fazla hedefe dağıtılması gibi durumlarda sistemin esnekliğini artırır.

Esnek Görüntüleme ve Veri İletim Modları: ARINC 818, modern aviyonik sistemlerin karmaşık veri ve görüntüleme ihtiyaçlarına yanıt vermek için esnek görüntüleme ve veri iletimi modlarını destekler. Standart, stereo görüntüler, bölgesel görüntüleme (region-of-interest) ve görüntü dilimleme (tiling) gibi çeşitli tekniklerle yüksek çözünürlüklü ekranlarda hedef odaklı görüntülemeyi mümkün kılar. Örneğin, yatay ve dikey dilimlerin birleştirilmesiyle belirli bir ilgi bölgesinin görüntülenmesi sağlanabilir. Bu yaklaşım, yalnızca gereken alanların görüntülenmesine olanak tanıyarak veri işleme yükünü azaltır ve sistem performansını optimize eder.

Bunun yanı sıra, veri odaklı bağlantılar da ARINC 818'in sağladığı önemli bir avantajdır. Dokunmatik ekran koordinatları, durum bilgisi ve komut kontrolü gibi spesifik veri aktarımı ihtiyaçları için optimize edilmiş bağlantılar, standardın esnekliğini daha da artırır. Sadece veri aktarımı için uyarlanmış bu tür bağlantılar, farklı hızlarda çalışabilir ve operasyonel gereksinimlere göre özelleştirilebilir. Böylece hem görsel hem de veri tabanlı işlemlerde sistem tasarımcılarına daha geniş bir kontrol ve uyum imkânı sunar.

2.3 ARINC 818 YAPISI

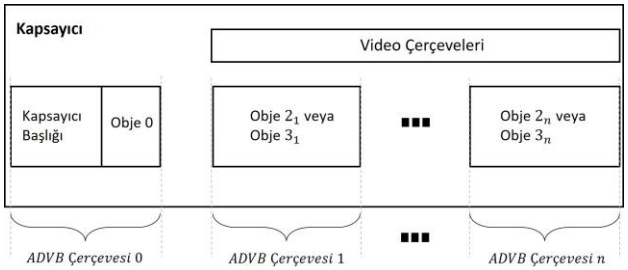
ARINC 818, modern aviyonik sistemlerin yüksek bant genişliği ve düşük gecikme gereksinimlerini karşılamak üzere geliştirilmiş, modüler ve esnek bir protokol standardıdır. Standart, veri iletimi sırasında verileri ADVB protokolü aracılığıyla organize eder ve taşır. ADVB, verileri kapsayıcılar (containers) içinde düzenler. Her kapsayıcı, video karelerini taşımak için gerekli olan video, ses ve yardımcı (ancillary) verileri içerir. Bu veriler, kapsayıcılar içerisinde bir dizi ADVB çerçevesinin diziliminden oluşur.

ADVB protokolü, fiber kanal tabanlı bir protokol olduğu için, tüm veriler 32-bit kelimeler (words) halinde taşınır. ARINC 818, bu 32-bit kelimeleri temel veri birimi olarak kullanarak veri bütünlüğünü ve senkronizasyonu korur.

Aşağıdaki başlıklarda, ARINC 818 standardında tanımlanan kapsayıcı ve ADVB çerçeve yapıları detaylı bir şekilde ele alınmaktadır.

ADVB Kapsayıcısı: ARINC 818, veri organizasyonu ve taşınmasında ADVB Kapsayıcısı olarak adlandırılan kapsayıcı yapılardan oluşur. Her bir kapsayıcı, kapsayıcı başlığı ve sıralı ADVB çerçevelerinden oluşur. Bu sayede, her bir kapsayıcı bir video çerçevesine karşılık gelir. Bu nedenle, kapsayıcılar ekran görüntüsünün sürekli olarak

güncellenmesini sağlamak için ardışık olarak iletilir. ADVB kapsayıcısının genel yapısı, Şekil 1'de verilmiştir.



Şekil 1. ADVB Kapsayıcısının genel yapısı

Her bir kapsayıcı video, ses ve yardımcı verilerini içerir ve bu verilerin ADVB protokolünde tanımlanması Çizelge 2'de gösterilmektedir (Alexander ve Keller 2011). ADVB kapsayıcısı; kapsayıcı başlığı, yardımcı veri, ses verisi ve video verisi bölümlerinden oluşur.

Çizelge 2. Obje türleri.

Obje Numarası	Obje Türü
Obje 0	Yardımcı Veri
Obje 1	Ses Verisi
Obje 2	Video Verisi (Progressive Scan)
Obje 3	Video Verisi (Interlaced Scan)

Kapsayıcı Başlığı: Her ADVB kapsayıcısı, bir adet ADVB kapsayıcı başlığı içerir. Bu başlık, Şekil 1'de görüleceği üzere, kapsayıcının iletilen ilk ADVB çerçevesinde Obje 0'dan önce gelir. 22 kelimeden oluşan bu başlık, kapsayıcının türünü ve içeriğini belirten meta verileri içerir. Ayrıca, veri akışının doğru bir şekilde işlenebilmesi için zamanlama ve senkronizasyon bilgilerini de içerir.

Yardımcı Veri (Obje 0): Obje 0, yardımcı verileri taşıyan bölüm olarak işlev görür. Bu bölüm, video çerçevesinin satır ve sütun sayısı, video piksellerinin format bilgisi, piksel en-boy oranı gibi bilgilere sahiptir.

Ses Verisi (Obje 1): ADVB protokolünde, Ses Verisi (Obje 1), sistemlerin ihtiyaçlarına göre sıkıştırılmış, sıkıştırılmamış veya şifrelenmiş formatlarda ses bilgisini içerir. Ses verileri, ADVB kapsayıcısı içinde belirli bir ofsetten başlayarak yer alır ve kullanılan sistemin gereksinimlerine uygun şekilde yapılandırılır. Bu veri türü, ARINC 818'in standartlarına uygun paketleme ve formatlama kurallarına göre sistemlere entegre edilir.

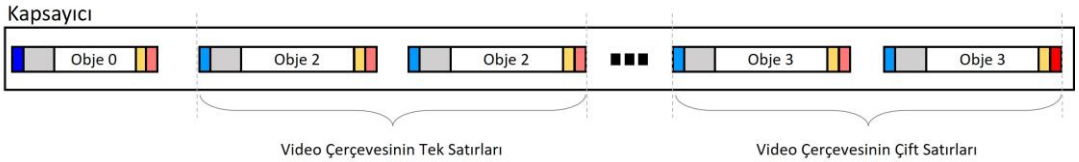
Geçmeli ve Tek Geçişli Tarama Verileri (Obje 2 ve Obje 3): Obje 2 ve Obje 3, video verisinin taşınması için kullanılır. Taşınacak video çerçevesinin çözünürlüğüne bağlı olarak, Obje 2 ve Obje 3'ün içerdiği kelime sayısı değişir ve maksimum 528 kelime (2112 byte) taşıyabilir. Eğer video çerçevesinin bir satır uzunluğu 512 kelimeyi geçiyorsa, bu video satırı peş peşe gelen ADVB çerçeveleri ile taşınır.

Tek geçişli tarama (progressive scan) formatında, video çerçevesinin tüm satırları yalnızca Obje 2 ile taşınırken; geçmeli tarama (interlaced scan) formatında, video verilerinin tek numaralı satırları Obje 2 ile, çift numaralı satırları ise Obje 3 ile taşınır.

Tek geçişli video iletiminin Obje 2'ler ile nasıl gerçekleştirildiği, Şekil 2'de gösterilmektedir.



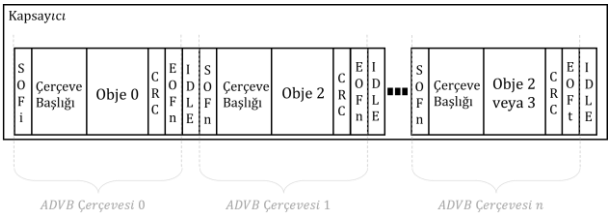
Şekil 2. ADVB Kapsayıcısı – Tek Geçişli Video



Şekil 3. ADVB Kapsayıcısı - Geçmeli Video

Geçmeli video iletiminde, Obje 2 ile Obje 3'lerin nasıl kullanıldığı Şekil 3'de gösterilmektedir. ADVB kapsayıcısı içinde, Obje 0'dan sonra ilk olarak iletilecek video çerçevesinin tek satırları Obje 2 formatında iletilir. Ardından, video çerçevesinin çift satırları Obje 3 formatında iletilir.

ADVB Çerçeve Formatı: ADVB çerçeveleri, ARINC 818 standardının temel yapı taşıdır ve veri iletimi sırasında kullanılan en küçük birimdir. Her ADVB çerçevesi, Şekil 4'te belirtildiği gibi bir çerçeve başlangıç bilgisini (SOF), çerçeve başlığını, veri yükünü (Obje 0, Obje 2 veya Obje 3), hata kontrol bilgisini (CRC) ve çerçeve bitiş bilgisini (EOF) içerir.



Şekil 4. ADVB Çerçeve Yapısı

Bekleme Kelimeleri (IDLE): Çerçeveler arasında veri akışını düzenlemek ve senkronizasyonu sağlamak için "idle" kelimesi kullanılır. Bu özel kelime, çerçeveler arasındaki boşlukları doldurarak sistemin kararlılığını artırır.

Çerçeve Başlangıcı (SOF): Veri akışının düzenli bir şekilde başlamasını sağlayan SOF kelimesi, her bir ADVB çerçevesinin başında bir kelimelik alanı temsil eder. SOF_i (Start of Frame Initiate) ve SOF_n (Start of Frame Normal) olmak üzere iki farklı SOF kelime dizilimi vardır. SOF_i, her bir kapsayıcı başlangıcında ilk ADVB çerçevesinin geldiğini belirtmek için ilk ADVB çerçevesinin başında kullanılır. SOF_n ise geri kalan ADVB çerçevelerinin başında kullanılır.

Çerçeve Başlığı: Çerçeve başlığı, her bir ADVB paketinde SOF kelimesinden sonra gelen 6 kelimelik bir alan için kullanılır. Bu alanda, anahtarlama kontrolleri için kaynak ve hedef bilgileri, veri türü, ADVB kapsayıcı numarası ve ADVB çerçeve numarası gibi meta veriler yer almaktadır.

Veri Alanı: Veri alanı, ADVB çerçevesinin ana veri yükünün taşındığı alanı temsil eder. Yardımcı veri (Obje 0), ses verisi (Obje 1) veya video verisi (Obje 2-3) bu alanda bulunur. Veri yükü boyutu, maksimum 2112 bayt ile sınırlıdır.

Döngüsel Artıklık Kontrolü (CRC): Her ADVB çerçevesi, Çerçeve Başlığı ve Veri Alanı alanlarının hata durumu analizi için kullanılan 1 kelimelik bir CRC değeri içerir. Dolayısıyla, her ADVB çerçevesinin kendine özgü bir CRC değeri vardır. Bu CRC değeri, aşağıdaki polinom ile hesaplanır:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1 \quad (1)$$

Çerçeve Sonu (EOF): Veri akışının düzenli bir şekilde sonlandırılmasını sağlayan EOF kelimesi, her bir ADVB çerçevesinin sonunda bir kelimelik alanı temsil eder. EOF_t (end of frame terminate) ve EOF_n (end of frame normal) olmak üzere iki farklı EOF kelime dizilimi vardır. EOF_t, her bir kapsayıcının son ADVB çerçevesinin geldiğini belirtmek için son ADVB çerçevesinin sonunda kullanılır. EOF_n ise geri kalan ADVB çerçevelerinin sonunda kullanılır ve ADVB paketinin bitip IDLE kelimesine geçileceğini belirtir.

Arayüz Kontrol Dokümanı (ICD): ARINC 818 standardının esnek ve özelleştirilebilir yapısının en önemli unsurlarından biri, ICD olarak adlandırılan dokümandır. ICD, bir sistemde ARINC 818 standardının nasıl uygulanacağını ve özelleştirileceğini belirleyen teknik bir kılavuz niteliğindedir. Bu belge, sistem bileşenlerinin birbiriyle uyumlu çalışmasını sağlamak için kritik öneme sahiptir.

ICD, ARINC 818'in uygulama gereksinimlerine göre nasıl yapılandırılacağını tanımlar (Keller ve Alexander 2009). Özellikle video formatları, çözünürlük, renk alanı, zamanlama parametreleri ve veri türleri gibi birçok detayı içerir. Bu doküman hem veriyi gönderen hem de alan cihazların standardı aynı şekilde anlamasını ve işlemesini garanti eder.

3. Materyal ve Metot

ARINC 818 tabanlı sistemlerin tasarımı ve analizi, karmaşık zamanlama gereksinimleri, yüksek bant genişliği ihtiyaçları ve veri organizasyonu gibi birçok teknik zorluğu beraberinde getirir. Bu bağlamda, fiziksel sistemlerin geliştirilmesine geçmeden önce simülasyon araçları, tasarım doğrulama ve hata analizi süreçlerinde kritik bir rol oynar. Bu çalışma kapsamında geliştirilen Python tabanlı ARINC 818 simülasyon aracı, hem standardın teknik detaylarının analizini desteklemek hem de farklı senaryoları test ederek tasarım sürecine katkıda bulunmak için tasarlanmıştır. Geliştirilen simülasyon ortamı, ARINC 818 standardının çeşitli bileşenlerini (örneğin, kapsayıcı ve çerçeve yapıları, zamanlama mekanizmaları ve hata tespiti gibi) simüle ederek, kullanıcıların standardın davranışını detaylı bir şekilde gözlemlemesine olanak tanır. Bu simülasyon ortamı, mühendisler ve araştırmacılar için hem eğitim hem de tasarım doğrulama süreçlerinde etkili bir araç olarak kullanılabilir. Bu bölümde, Python tabanlı simülasyonun geliştirilme süreci, yapısı ve fonksiyonları detaylı bir şekilde ele alınacaktır.

3.1 Simülasyon Ortamına Genel Bakış

Python tabanlı ARINC 818 simülasyon sistemi, modüler bir mimari üzerine inşa edilmiştir ve farklı kullanım senaryolarına uyarlanabilir ve ölçeklenebilir bir yapı sunar. Bu ortam, dinamik giriş yapılandırması ve etkileşimli analizler sağlayan bir grafiksel kullanıcı arayüzü sunacak şekilde PyQt5 kullanılarak geliştirilmiştir. Temel amaç, ARINC 818 kapsayıcı ve çerçevelerini doğru bir şekilde modellemek, veri iletimini simüle etmek ve zamanlama diyagramlarını görselleştirmektir. Bu simülasyon ortamının temel mimarisi, aşağıdaki ana bileşenlerden oluşur:

Kullanıcı Girdisi ve Parametre İşleme: Kullanıcıdan alınan çözünürlük, kare hızı, renk formatı gibi girişlerin sistem için uygun veri yapılarına dönüştürülmesi.

Zamanlama ve Çerçeve Hesaplamaları: ARINC 818 çerçevelerinin zamanlama gereksinimlerine uygun şekilde modellenmesi.

ADVB Çerçevesi ve ADVB Konteyner Üretimi: ARINC 818'in standartlarına uygun veri yapılarını oluşturma.

Çıktı Analizi ve Görselleştirme: Zamanlama diyagramlarının ve veri akışının grafiksel olarak kullanıcıya sunulması.

Çizelge 3. Kullanıcıdan alınan parametreler.

Parametre	Alabileceği Değer
Sıralı Küme Sınıfı	Sınıf 1, Sınıf 3
VESA Görüntü Formatı	VGA, SVGA, XGA, HD 720, WXGA, SXGA, SXGA+, 1440x900, 1600x1024, UXGA, WSXGA+, HD 1080, WUXGA, QXGA, WQHD, Özel
Satır Başına Aktif Piksel Sayısı (VESA formatı "Özel" ise)	Tam sayı
Kare Başına Aktif Satır Sayısı (VESA formatı "Özel" ise)	Tam sayı
Hedef Kare Hızı (Hz)	15, 20, 24, 24*1000/1001, 24 (parçalı çerçeveler), 24*1000/1001 (parçalı çerçeveler), 25 (PAL), 30, 30*1000/1001 (29.97 NTSC), 50, 60, 60*1000/1001 (59.94 NTSC), 50 (VESA DMT), 60 (VESA DMT), 75 (VESA DMT), 85 (VESA DMT), 50 (VESA CVT), 60 (VESA CVT), 75 (VESA CVT), 85 (VESA CVT)
Piksel Formatı	Tek renkli 8-bit, YCrCb 24-bit (4:2:2), YIQ 24-bit (8:8:8), RGB 24-bit (8:8:8), aRGB 32-bit (8:8:8:8), RGGB 32-bit (8:8:8:8)
Bağlantı Hızı (Gb/s)	1.0625, 1.5, 2.125, 3.1875, 4.25, 5, 6.375, 8.5, 10, 12.75, 14.025, 21.0375
Satır Başına Kare Sayısı Manuel Olarak Belirlensin mi?	Evet, Hayır
Manuel Kare Sayısı Değeri (Eğer manuel belirleneceği belirtilmişse)	Tam sayı
Obje 0'a Ait Yardımcı Veri Boyutu	Tam sayı
Obje 0 Öncesindeki Pasif Satır Sayısı	Tam sayı
Obje 0 Sonrasındaki Pasif Satır Sayısı	Tam sayı

3.2 Simülasyon Akışı

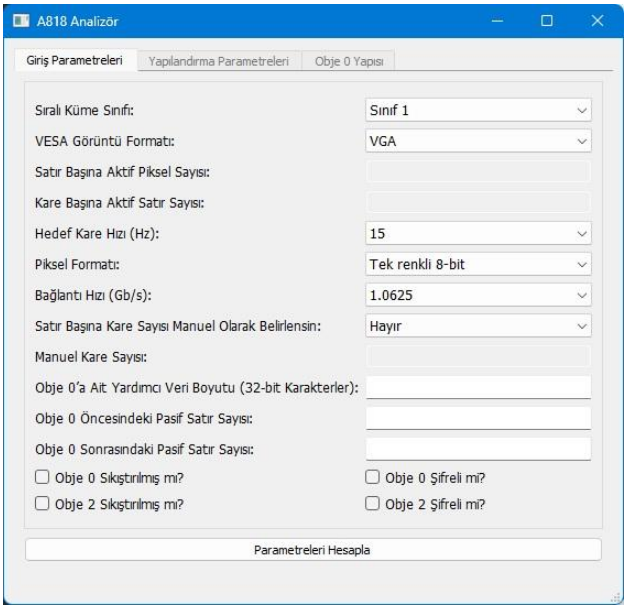
Simülasyon ortamı, standardın temel bileşenlerini simüle etmek ve analiz etmek için bir dizi mantıksal adımı takip eder. Simülasyon akışı, kullanıcıdan alınan girdilerden başlayarak, zamanlama hesaplamaları, veri işleme, çerçeve ve kapsayıcı oluşturma, simülasyonun yürütülmesi ve sonuçların doğrulanmasına kadar ilerleyen bir süreçtir. Kullanıcı dostu bir grafiksel arayüzden oluşan simülasyon, kullanıcıların standardın yapısını anlamalarını kolaylaştırırken, aynı zamanda parametreleri yapılandırmalarına ve simülasyon sonuçlarını analiz etmelerine olanak tanır. Aşağıda, arayüzün detaylı bir şekilde açıklaması yapılacaktır.

Giriş Parametreleri Sayfası: Kullanıcılar, simüle etmek istedikleri ARINC 818 standardına uygun parametreleri, Çizelge 3'de verilen listeye göre, Şekil 5'deki sayfada tanımlar.

Yapılandırma Parametreleri Sayfası: Bu sayfa, giriş parametreleri sayfasında kullanıcıdan alınan girdiler doğrultusunda oluşturulacak ARINC 818 paketlerinin teknik özelliklerinin ve zamanlama detaylarının gösterildiği bölümdür. Kullanıcı, bu sayfa üzerinden oluşturacağı ARINC 818 paketlerinin özelliklerini gözlemleyebilir ve gerekirse doğruluklarını kontrol edebilir. Şekil 6'da da gösterildiği gibi, sayfa, 'Parametreler' ve 'Zamanlama Değerleri' olmak üzere iki ana bölümden oluşmaktadır.

Parametreler bölümünde, oluşturulacak ADVB paketlerine ilişkin temel yapısal özellikler sunulmaktadır. Bu bölümde:

- 1 kelimenin iletim süresi,
- Bir ADVB kapsayıcısının kaç ADVB çerçevesi içereceği,
- Bir video satırında kaç ADVB çerçevesinin yer alacağı,
- Her bir ADVB çerçevesinin veri alanının kaç byte'tan oluşacağı gibi teknik detaylar bulunmaktadır.

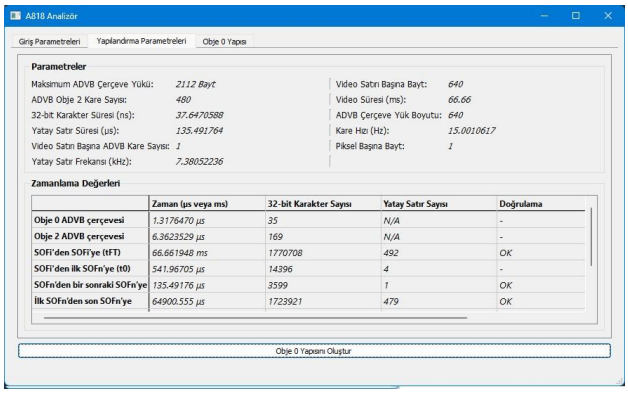


Şekil 5. Giriş Parametreleri Sayfası

Zamanlama Değerleri bölümü ise, oluşturulacak ADVB paketlerine ait iletişim süresi ve kelime sayısı gibi zamanlama analizlerini içermektedir. Bu bölümde:

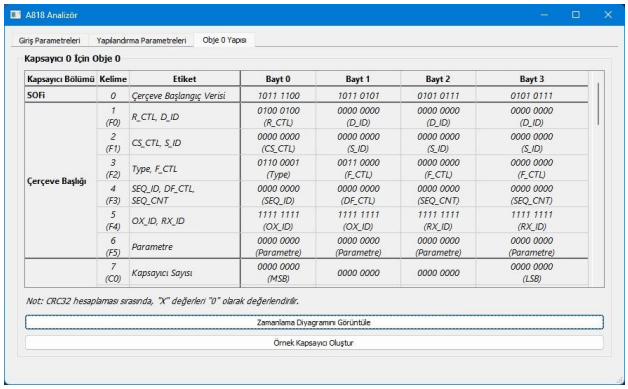
- İlk ADVB çerçevesinin kelime sayısı ve iletim süresi,
- Video verilerini içeren ADVB çerçevelerinin kelime sayısı ve iletim süresi,

- Bir kapsayıcının toplam kelime sayısı ve iletim süresi gibi bilgiler yer almaktadır.



Şekil 6. Yapılandırma Parametreleri Sayfası

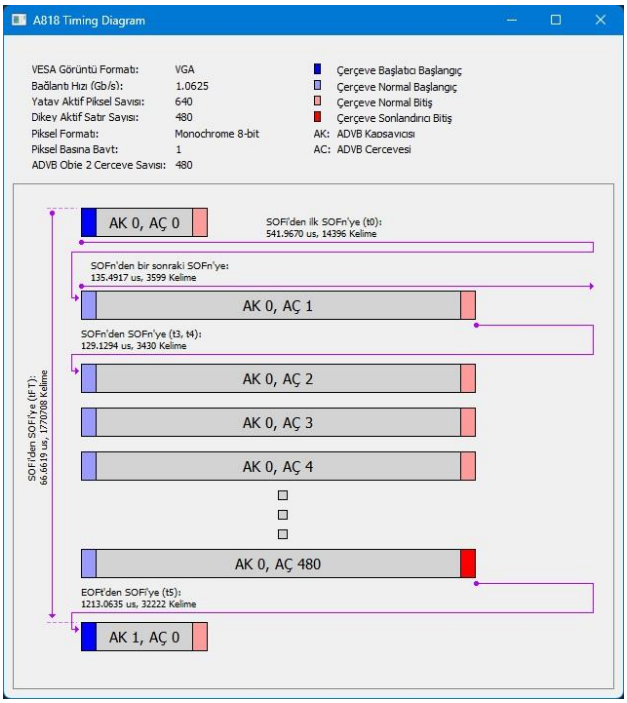
Obje 0 Yapısı Sayfası: Bu sayfa, kullanıcının giriş parametreleri sayfasında tanımladığı girdiler doğrultusunda oluşturulan bir ADVB kapsayıcısının Obje 0 yapısını, Şekil 7'de görüleceği üzere detaylı bir şekilde sunar. Sayfa, SOF, Çerçeve Başlığı, Kapsayıcı Başlığı, CRC ve EOF bölümlerini, ilgili tanımlayıcılar ve bayt düzeyindeki içerikleriyle birlikte kapsamlı olarak göstermektedir. Ayrıca, kullanıcı bu sayfadan, kapsayıcının zamanlama diyagramını inceleyebileceği 'Zamanlama Diyagramı' sayfasına veya örnek bir kapsayıcı oluşturabileceği 'Örnek Kapsayıcı Oluşturma' sayfasına kolayca geçiş yapabilir.



Şekil 7. Obje 0 Yapısı Sayfası

Zamanlama Diyagramı Sayfası: Zamanlama diyagramı sayfası, ARINC 818 standardına ait ADVB kapsayıcısı ile ADVB çerçevelerinin yapılandırma parametreleri sayfasının 'Zamanlama Değerleri' bölümünde belirtilen zamanlama bilgilerini görselleştiren bir diyagram sunar. Şekil 8'de verilen örnek diyagramda, her bir ADVB çerçevesinin (AK #, AÇ #) başlangıç ve bitiş zamanları ile çerçeveler arasındaki zamanlama detayları görselleştirilmiştir. Örneğin, SOF_i karakterinden ilk SOF_n karakterine kadarki olan süre ile kelime sayısı açıkça gösterilir. Kullanıcı, bu diyagramı inceleyerek simülasyonun zamanlama gereksinimlerini değerlendirebilir ve sistemin senkronizasyon özelliklerini analiz edebilir.

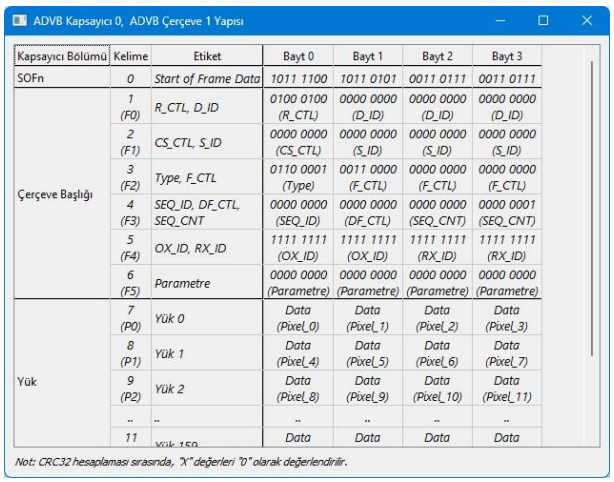
Diyagramın etkileşimli yapısı, kullanıcıların her bir çerçeveyi detaylı olarak incelemesine olanak tanır. Kullanıcı, bir çerçeveye tıkladığında, o çerçeveye özgü bilgilerin yer aldığı yeni bir sayfa açılır. Bu sayfa, seçilen çerçevenin başlangıç ve bitiş zamanları, kelime sayısı, CRC değeri, başlık bilgileri gibi detayları içerir. Böylece kullanıcı, her bir çerçevenin yapısını ve zamanlamasını daha ayrıntılı bir şekilde inceleyebilir, simülasyon sürecindeki olası sorunları daha kolay tespit edebilir.



Şekil 8. Zamanlama Diyagramı Sayfası

Çerçeve Detayları Sayfası: Zamanlama diyagramında görüntülenen her bir ADVB çerçevesi, kullanıcı tarafından tıklanarak daha ayrıntılı bir şekilde incelenebilir. Örneğin, AK 0, AÇ 1 (ADVB Kapsayıcı 0, ADVB Çerçeve 1) çerçevesine tıklanıldığında, çerçevenin yapısını detaylandıran yeni bir sayfa açılır. Bu sayfa, seçilen ADVB çerçevesinin çeşitli bileşenlerini detaylı bir şekilde sunar. Şekil 9'da görülen sayfada, çerçevenin SOF, Çerçeve Başlığı, Veri Yükü, CRC ve EOF bölümleri yer alır. Bu detaylı inceleme, kullanıcının standardın her bir çerçevesinin içeriğini ve zamanlama özelliklerini anlamasına yardımcı olur. Ayrıca, simülasyonun doğruluğunu kontrol etmek ve sistemin performansını değerlendirmek için önemli bilgiler sağlar.

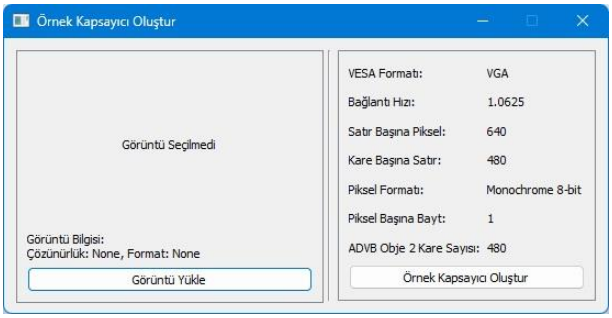
Örnek Kapsayıcı Oluşturma Sayfası: Örnek kapsayıcı oluşturma sayfası, Şekil 10'da görüleceği üzere kullanıcıların görsel dosyalarını ARINC 818 formatına dönüştürmesini sağlayan etkileşimli bir arayüzdür.



Şekil 9. ADVB Çerçevesi Bilgileri Sayfası

Bu sayfa, kullanıcıların görsel dosyalarını yüklemelerini ve ardından bu görselleri uygun ARINC 818 veri yapısına dönüştürmelerine olanak tanır. Sayfanın işleyişi şu adımlardan oluşmaktadır:

Öncelikle, kullanıcı bilgisayarından dönüştürmek istediği görsel dosyasını seçerek yükleme işlemini gerçekleştirir. Yüklenen dosya, ARINC 818 standardına uygun şekilde işlenmek üzere hazırlanır. Ardından, kullanıcı "Örneği Üret" butonuna tıklayarak görselin ARINC 818 formatına dönüştürülme sürecini başlatır. Bu işlem sırasında, görsel ARINC 818 standardının gereksinimlerine uygun şekilde işlenir, çözünürlüğü, renk formatı ve diğer teknik özellikleri dikkate alınarak ADVB kapsayıcısı ve çerçeveleri oluşturulur. Dönüştürme işlemi tamamlandıktan sonra, oluşturulan ARINC 818 veri yapısı görselleştirilmiş şekilde kullanıcıya sunulur. Bu sayede kullanıcı, görselin ARINC 818 formatındaki yapısını ve içerdiği verileri detaylı olarak inceleyebilir.



Şekil 10. Örnek Kapsayıcı Oluşturma Sayfası

Özetle, bu araç, ARINC 818 standardı ile uyumlu bir görsel veri akışının nasıl oluşturulacağına dair önemli bir örnek sağlar ve kullanıcıların standart hakkında daha iyi bir anlayış kazanmasına yardımcı olur.

4. Bulgular

Bu bölümde, Python tabanlı ARINC 818 simülasyon ortamının performans analizi sonuçları sunulmaktadır. Simülasyon ortamı, standardın farklı yapılarını modelleme ve analiz etme sürecinde yüksek esneklik sağlayarak, veri yapıları, zamanlama gereksinimleri ve hata tespiti mekanizmalarının etkin bir şekilde test edilmesine olanak tanımaktadır. Farklı çözünürlükler ve giriş parametreleri kullanılarak yapılan testlerle, simülasyon süresi ve performans kriterleri değerlendirilmiş ve bu kapsamda parametrelerin etkileri detaylı bir şekilde analiz edilmiştir.

4.1 Test Ortamı ve Donanım Konfigürasyonu

Performans testleri, Python tabanlı simülasyon ortamının aynı donanım üzerinde çalıştırılmasıyla gerçekleştirilmiştir. Bu yaklaşım, simülasyon sürelerini etkileyebilecek harici faktörleri en aza indirerek, daha tutarlı ve güvenilir performans sonuçları elde edilmesini sağlamıştır.

Simülasyonların koştuğu bilgisayar özellikleri;

- İşletim Sistemi: Windows 11 Pro
- İşlemci: Intel Core i7-10750H (12 çekirdek)
- Bellek (RAM): 8 GB RAM

Python tabanlı simülasyon ortamı, ARINC 818 standardını modelleyen bir yazılım olarak çalıştırılabilir bir dosya formatında çalıştırılmıştır. Kullanıcı girişlerine dayalı olarak ADVB kapsayıcıları ve zamanlama diyagramları üretilmiştir.

4.2 Test Sonuçları

Python tabanlı simülasyon ortamının performansı, farklı çözünürlükler ve çeşitli giriş parametreleri kullanılarak analiz edilmiştir. Bu kapsamda, çözünürlük (VGA, XGA, UXGA, WXGA ve WUXGA), hedef kare hızı (15 Hz, 30 Hz ve 60 Hz), piksel formatı (tek renkli 8-bit, RGB 24-bit ve aRGB 32-bit) ve bağlantı hızı (1.0625 Gb/s, 2.125 Gb/s, 3.1875 Gb/s, 5 Gb/s ve 6.375 Gb/s) olmak üzere dört temel parametrenin farklı kombinasyonları incelenmiştir. Diğer tüm parametreler sabit tutulmuş ve Çizelge 4'te belirtilen değerler kullanılmıştır. Her bir çözünürlük için elde edilen simülasyon süreleri sırasıyla Şekil 11 (VGA), Şekil 12 (XGA), Şekil 13 (UXGA), Şekil 14 (WXGA) ve Şekil 15 (WUXGA) üzerinde görselleştirilmiştir.

Görseller incelendiğinde çözünürlük değişiminin simülasyon süresi üzerinde belirgin bir etkisi olduğu anlaşılmaktadır. Çözünürlük arttıkça simülasyon süresinde belirgin bir artış meydana gelmiştir. VGA çözünürlüğünde elde edilen süreler en düşük değerlerde kalırken, WUXGA çözünürlüğünde en uzun simülasyon süreleri kaydedilmiştir. Bu artış, çözünürlüğün artmasıyla birlikte işlenen toplam veri miktarının ve dolayısıyla aktarılması gereken ADVB çerçevesi sayısının artmasına bağlanmaktadır.

Hedef kare hızı değişiminin de simülasyon süresi üzerinde etkili olduğu görülmüştür. 15 Hz hedef kare hızında

gerçekleştirilen simülasyonlar, 60 Hz hedef kare hızına kıyasla daha uzun sürelerde tamamlanmıştır. Bu durum, düşük kare hızlarında çerçeve aktarım süresinin uzaması ve zamanlama hesaplamalarının daha geniş zaman pencerelerine yayılmasıyla açıklanabilir. Buna karşılık, yüksek kare hızlarında, birim zamanda işlenen veri miktarı artsa da her çerçeve için ayrılan zaman penceresinin daralması nedeniyle simülasyonlar daha hızlı tamamlanmıştır.

Piksel formatındaki değişimin de simülasyon performansı üzerinde önemli bir etkisi olduğu gözlemlenmiştir. Tek renkli 8-bit piksel formatı en düşük veri yoğunluğuna sahip olduğundan tüm test senaryolarında en kısa simülasyon sürelerini sağlamıştır. RGB 24-bit formatı ise, veri yoğunluğundaki artış nedeniyle simülasyon süresini önemli ölçüde artırmıştır. 32-bit aRGB formatı, 24-bit formata kıyasla daha fazla veri içerdiği için, en uzun simülasyon sürelerine neden olmuştur. Bu sonuçlar, piksel başına aktarılan veri miktarındaki artışın, her bir video karesi için işlenen toplam veri hacmini artırarak doğrudan simülasyon süresine yansıdığını ortaya koymaktadır.

Simülasyon süreleri üzerinde etkili olan bir diğer kritik değişkenin de bağlantı hızı olduğu gözlemlenmiştir. Yapılan analizler, bağlantı hızındaki değişimin iki farklı mekanizma üzerinden simülasyon süresine etki ettiğini ortaya koymuştur. İlk mekanizmada, çözünürlük, kare hızı ve piksel formatı sabit tutulduğunda (örneğin VGA çözünürlüğü, 30 Hz kare hızı ve 24-bit renk derinliği gibi), yalnızca bağlantı hızının artırılması taşınan gerçek video verisi miktarını değiştirmemektedir. Bu tür senaryolarda, protokolün zamanlama gereksinimlerini sağlamak amacıyla daha fazla IDLE karakteri eklenmekte ve toplamda işlenen veri hacmi artmaktadır. Dolayısıyla yalnızca bağlantı hızının artırılması durumunda, IDLE karakterlerindeki artış nedeniyle işlenen veri miktarı artmakta ve buna bağlı olarak simülasyon süresi uzamaktadır. İkinci mekanizmada ise, çözünürlük veya piksel formatı gibi parametrelerin artırılmasıyla taşınan gerçek video verisi miktarı doğrudan artmakta; bu da hem fiziksel bant genişliği ihtiyacını yükseltmekte hem de simülasyon süresinin veri yükündeki artışa bağlı olarak uzamasına yol açmaktadır. Bu analizler, ARINC 818 protokolünde bağlantı hızının yalnızca taşıma kapasitesine değil, aynı zamanda zamanlama yapısına ve IDLE karakter yerleşimine de doğrudan etki eden kritik bir parametre olduğunu göstermektedir.

Simülasyon sonuçları, ARINC 818 standardının zamanlama performansının çözünürlük, kare hızı, piksel formatı ve bağlantı hızı gibi temel parametrelerden doğrudan etkilendiğini ortaya koymuştur. Özellikle yüksek çözünürlük, geniş piksel derinliği ve yüksek bağlantı hızı gibi koşullar altında, taşınan veri hacminin artması nedeniyle simülasyon süresinin belirgin şekilde uzadığı gözlemlenmiştir. Elde edilen bulgular, ARINC 818 standardında yüksek çözünürlüklü video aktarımlarında bir video karesinin birden fazla ADVB çerçevesine bölünmesinin gerekebileceğini göstermektedir. Bu durumda, her ek çerçeve için ilave başlık ve senkronizasyon bilgisi taşınmakta, bu da toplam aktarım süresini ve veri yükünü artırmaktadır. Sonuçlar doğrultusunda, geliştirilen simülasyon ortamı ile ARINC 818 standardının farklı çözünürlük, kare hızı, piksel formatı ve bağlantı hızı kombinasyonlarındaki zamanlama davranışları detaylı şekilde incelenmiş; bu parametrelerin standardın veri yapısı ve zamanlama gereksinimleri üzerindeki etkileri kapsamlı biçimde ortaya konmuştur. Geliştirilen ortam, ARINC 818 tabanlı video sistemlerinin erken aşama tasarımında performans öngörüsü sağlamak ve sistem gereksinimlerinin uygunluğunu değerlendirmek için etkili bir analiz platformu olarak öne çıkmaktadır.

Çizelge 4. Simülasyonda kullanılan sabit girdiler.

Parametre	Alabileceği Değer
Sıralı Küme Sınıfı	Sınıf 1
Satır Başına Kare Sayısı Manuel Olarak Belirlensin mi?	Hayır
Obje 0'a Ait Yardımcı Veri Boyutu	4
Obje 0 Öncesindeki Pasif Satır Sayısı	8
Obje 0 Sonrasındaki Pasif Satır Sayısı	4

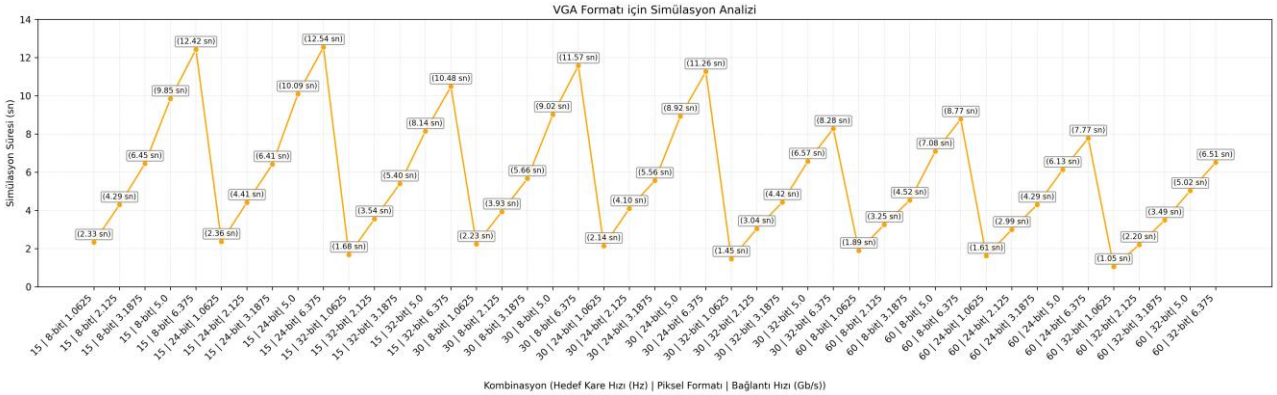
5. Sonuç

Bu çalışma, ARINC 818 standardının zamanlama davranışlarını yazılım ortamında modellemek ve analiz etmek amacıyla geliştirilmiş Python tabanlı bir simülasyon aracını sunmaktadır. ARINC 818, yüksek bant genişliği ve düşük gecikme gereksinimleri nedeniyle özellikle aviyonik sistemlerde tercih edilen bir standarttır. Literatürde doğrudan ARINC 818'in zamanlama analizine yönelik bir simülasyon yaklaşımı bulunmadığı için, bu çalışma önemli bir boşluğu doldurmaktadır.

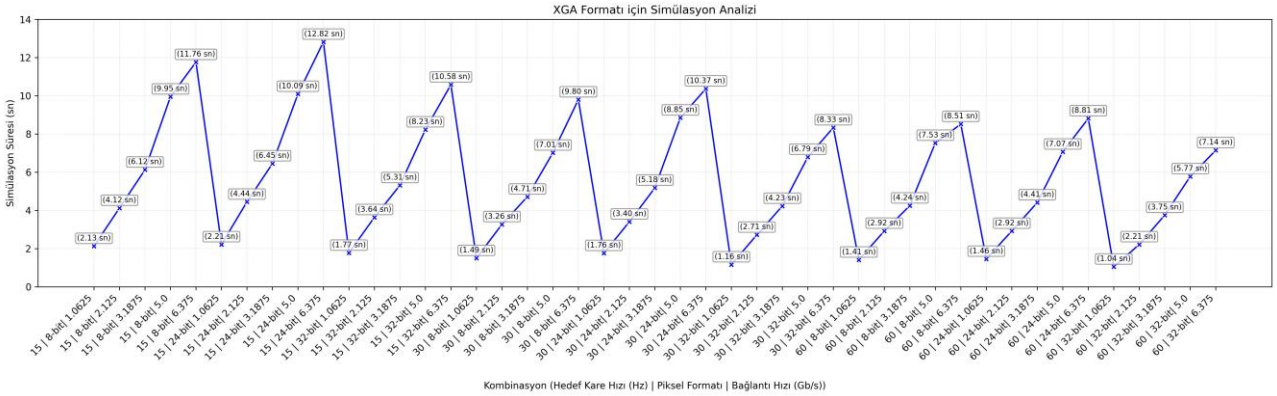
Sunulan simülasyon ortamı, çözünürlük, kare hızı, piksel formatı ve bağlantı hızı gibi temel parametrelerin protokol üzerindeki etkilerini kapsamlı bir şekilde analiz etme imkânı sağlamıştır. Gerçekleştirilen testler sonucunda, çözünürlüğün ve veri yoğunluğunun artmasının simülasyon süresinde belirgin artışlara yol açtığı; düşük kare hızlarında zamanlama pencerelerinin genişleyerek simülasyon süresini uzattığı; piksel formatındaki veri yoğunluğunun simülasyon yükünü doğrudan etkilediği ve bağlantı hızının hem IDLE karakter yerleşimi hem de toplam veri hacmi üzerinde önemli değişimlere neden olduğu ortaya konmuştur. Bu bulgular, ARINC 818

protokolü ile geliştirilecek sistemlerin erken tasarım aşamalarında performans tahmini ve gereksinim analizi yapılmasına imkân sağlamaktadır.

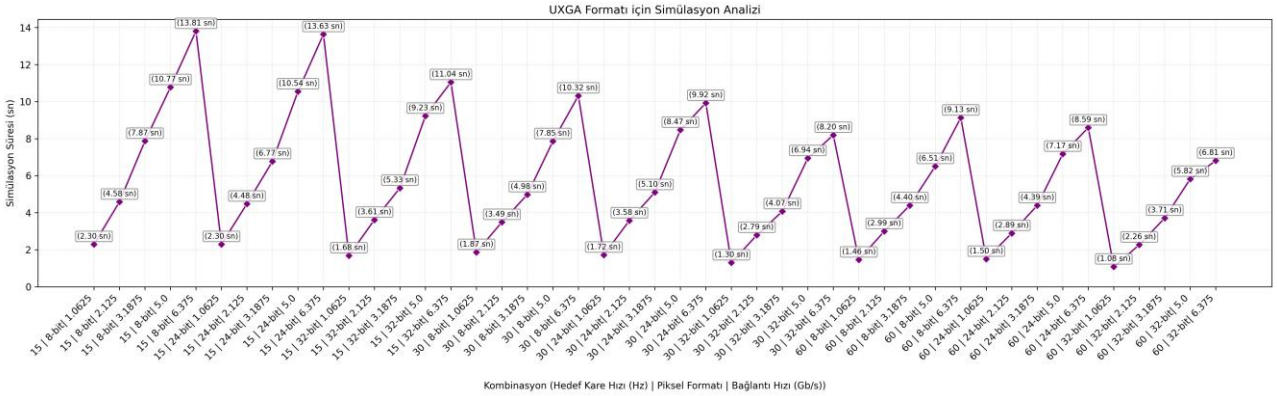
Bununla birlikte, geliştirilen simülasyon aracı, yalnızca zamanlama analizleri sunmakla kalmayıp, kullanıcı tarafından belirlenen parametreler doğrultusunda örnek bir ARINC 818 ADVB kapsayıcısı da oluşturarak veri yapısının somut biçimde gözlemlenmesine olanak tanımaktadır. Bu özellik sayesinde kullanıcılar, ARINC 818 veri çerçevesinin protokol gereksinimlerine uygun şekilde nasıl yapılandığını doğrudan inceleyebilmekte ve tasarımlarını bu referans yapıya göre şekillendirebilmektedir. Geliştirilen yazılım tabanlı simülasyon ortamı, donanım tabanlı test çözümlerine göre düşük maliyetli, erişilebilir ve esnek bir analiz imkânı sunmaktadır. Bu ortam, fiziksel donanım testlerine geçmeden önce gerçekleştirilebilecek analizlerle geliştirme süresini kısaltmakta ve maliyetleri düşürmektedir. Bunun yanı sıra, eğitim ve öğretim faaliyetlerinde kullanılarak ARINC 818 standardının daha geniş bir kullanıcı kitlesi tarafından anlaşılmasına katkı sağlayabileceği değerlendirilmektedir.



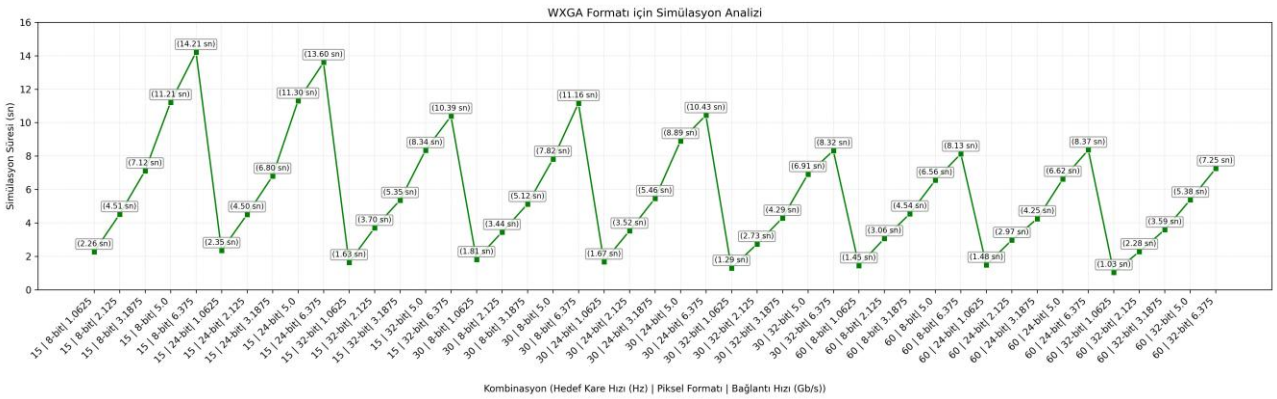
Şekil 11. VGA Standardı için Simülasyon Analizi



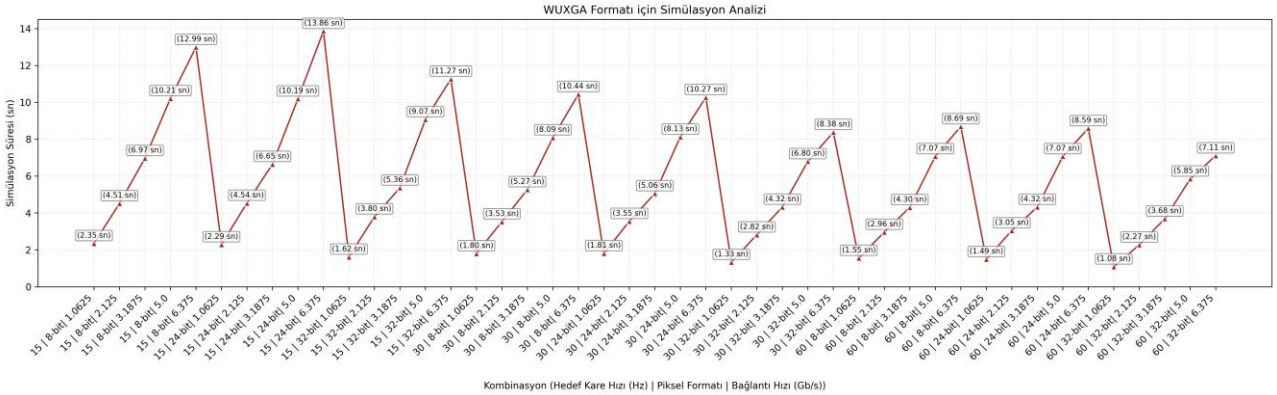
Şekil 12. XGA Standardı için Simülasyon Analizi



Şekil 13. UXGA Standardı için Simülasyon Analizi



Şekil 14. WXGA Standardı için Simülasyon Analizi



Şekil 15. WUXGA Standardı için Simülasyon Analizi

Sonuç olarak, bu çalışma, ARINC 818 standardının yazılım tabanlı simülasyon ile etkin bir şekilde analiz edilebileceğini ortaya koymakta ve bu alanda bir referans noktası oluşturma potansiyeli taşımaktadır. Çalışmanın sunduğu yaklaşım, aviyonik video sistemlerinin tasarım ve geliştirme süreçlerinde performans öngörüsü sağlama, veri yapısı oluşturma ve sistem gereksinimlerinin ön değerlendirilmesi açısından önemli katkılar sunmaktadır. Elde edilen sonuçlar, düşük maliyetli ve erişilebilir bir simülasyon yaklaşımının, aviyonik video veri yolu uygulamalarında tasarım sürecini kolaylaştırabileceğini göstermektedir. Geliştirilen simülasyon ortamının gelecekte benzer protokollerin analizinde de temel bir referans noktası oluşturabileceği değerlendirilmektedir.

Etik Standartlar Bildirgesi

Bu çalışmanın hazırlanma sürecinde bilimsel ve etik ilkelere uyulduğu ve yararlanılan tüm çalışmaların kaynakçada belirtildiği beyan olunur.

Yazarlık Katkı Beyanı

Yazar 1: Kaynaklar, Araştırma, Deney, Yazma – orijinal taslak Görselleştirme, Yazma – orijinal taslak

Yazar 2: Kaynaklar, Araştırma, Deney, Yazma – orijinal taslak Görselleştirme, Yazma – orijinal taslak

Çıkar Çatışması Beyanı

Yazarların bu makalenin içeriğiyle ilgili olarak beyan edecekleri hiçbir çıkar çatışması yoktur.

Verilerin Kullanılabilirliği

Yazarlar, bu çalışmanın bulgularını destekleyen ana verilerin makale içerisinde mevcut olduğunu beyan ederler.

6. Kaynaklar

Aslan, E., Arserim, M.A. and Uçar, A., 2024. Ayak bileği stratejisi kullanarak Robotis-OP2 için itme kurtarma kontrol yöntemlerinin karşılaştırılması. Journal of the Faculty of Engineering & Architecture of Gazi University, 39(4), 2551. https://doi.org/10.17341/gazimmfd.1359434

Alexander, J. and Grunwald, P., 2019. ARINC 818 Revision 3. 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC), San Diego, ABD. https://doi.org/10.1109/DASC43569.2019.9081760

Alexander, J. and Keller, T., 2007. Using ARINC 818 Avionics Digital Video Bus (ADVB) for Military Displays. Proceedings of SPIE - The International Society for Optical Engineering, 6558. https://doi.org/10.1117/12.720041

Alexander, J. and Keller, T., 2011. ARINC 818 for Video and Display Control. Proceedings of SPIE - Display Technologies and Applications for Defense, Security, and Avionics V; and Enhanced and Synthetic Vision, 8042, 80420L. https://doi.org/10.1117/12.884261

Aslan, S.N., Özalp, R., and Uçar, A., 2022. New CNN and hybrid CNN-LSTM models for learning object manipulation of humanoid robots from demonstration. Cluster Computing, 25, 1575–1590. https://doi.org/10.1007/s10586-021-03348-7

Bisson, K., 2007. Arinc-818 Testing for Avionics Applications. 2007 IEEE Autotestcon, 2007. https://doi.org/10.1109/AUTEST.2007.4374236

Değirmenci, A. and Karal, Ö., 2022. iMCOD: Incremental multi-class outlier detection model in data streams. Knowledge-Based Systems, 258, 109950. https://doi.org/10.1016/j.knosys.2022.109950

Gadde, M., 2024. Testing, Simulating and Validating ARINC 818. Aerospace Testing International, 2024, 100–101. https://doi.org/10.12968/S1478-2774(25)50026-1

Gayretli, M. G., 2023. OMNET++ Simulation Model for Integrated Modular Avionics, Master's Thesis, Istanbul Technical University, Graduate School of Defence Technologies, Istanbul, 84 pages.

Grunwald, P., 2014. ARINC 818 Specification Revisions Enable New Avionics Architectures. Proceedings of SPIE - Display Technologies and Applications for Defense, Security, and Avionics VIII, 9086, 908608. https://doi.org/10.1117/12.2050965

Grunwald, P., 2013. What's New in ARINC 818 Supplement 2. 2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC), 2013. https://doi.org/10.1109/DASC.2013.6712534

Keller, T. and Alexander, J., 2009. Applications of ARINC 818 in Avionics Video Systems. SAE International Journal of Aerospace, 2(1), 95–102. https://doi.org/10.4271/2009-01-3141

Keller, T. and Finnegan, D., (2024). Standardizing Video-Based Avionics and Mission Systems around the ARINC 818-3 Video Bus. 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC). https://doi.org/10.1109/DASC62030.2024.10748847

Keller, T. and Grunwald, P., 2014. ARINC 818 Adds Capabilities for High-Speed Sensors and Systems. Proceedings of SPIE - Infrared Technology and Applications XL, 9070, 90703F. https://doi.org/10.1117/12.2050972

Krotov, A., 2024. Development of Optical Communication Systems with High-Reliability Optical Sensors, Doctoral Thesis, Riga Technical University, Riga, Latvia, 119 pages.

Kılıç, C., Değirmenci, A. and Karal, Ö., 2024. Segmentation of the Area Between Anterior and Posterior Vertebral Elements in Axial MR Images Using U-Net. 2024 Innovations in Intelligent Systems and Applications Conference (ASYU), Ankara, Türkiye. https://doi.org/10.1109/ASYU62119.2024.10756992

Yalman, Y., Uyanık, T., Atlı, İ., Tan, A., Bayındır, K.Ç., Karal, Ö., Golestan, S., and Guerrero, J.M., 2022. Prediction of Voltage Sag Relative Location with Data-Driven Algorithms in Distribution Grid. Energies, 15(18), 6641. https://doi.org/10.3390/en15186641

Zhang, X., Liu, H., & Zhan, R., 2025. Avionics System Interface and Image Automation Testing Platform. Proceedings of SPIE – The International Conference Optoelectronic Information and Optical Engineering (OIOE2024), 13513, 135133E. https://doi.org/10.1117/12.3049162

Zimmerman, M., 2017. High Bandwidth, Real-Time Video Transport with ARINC 818. Proceedings of SPIE - Real-Time Image and Video Processing 2017, 10223, 102230B. https://doi.org/10.1117/12.2262794

Wood, R. B., & Howells, P. J., 2015. Head-Up Display. Digital Avionics Handbook, 3rd ed., edited by C. R. Spitzer, U. Ferrell, and T. Ferrell, CRC Press. https://doi.org/10.1201/b19009

İnternet kaynakları

1- ARINC 818 Ecosystem, https://www.arinc818.com/applications/, (23.12.2024)
2- ARINC 818 PRODUCT SUITE, https://www.greatrivertech.com/arinc-818-products, (10.04.2025)
3- ARINC 818 Solutions, https://www.iwavesystems.com/product-category/product-solutions/avionics/arinc818-solutions/, (10.04.2025)
4- ARINC 818 Solutions, https://www.techway.com/c/arinc-818-solutions/, (10.04.2025)
5- ARINC 818 Streaming IP Core, https://newwavedesign.com/products/ip-cores/arinc-818-streaming-ip-core/, (10.04.2025)
6- Defence and Aerospace, https://www.logic-fruit.com/solutions/defence-and-aerospace/, (10.04.2025)