

# Askeri Kara Araçlarında Test Amaçlı Hızlı Kullanıcı Arayüzü Yazılımı Geliştirme

Cankat AYKURT <sup>1,2</sup>

<sup>1</sup> FNSS Savunma Sistemleri A.Ş., Ankara, Türkiye

<sup>2</sup> Bilgisayar Mühendisliği Bölümü, Orta Doğu Teknik Üniversitesi, Ankara, Türkiye

cankat.aykurt@fnss.com.tr

cankat.aykurt@metu.edu.tr

**Özet.** Askeri kara araçları üzerindeki elektronik sistemler ve bu sistemlerin entegrasyonu yoğun test gerektirmektedir. Araçların tasarım ve test aşamalarında gerek bu sistemlerin fiziksel olarak erişilebilir olmaması, gerekse güvenlik sebeplerinden, testlerin gerçekleştirilmesi çoğu zaman mümkün olmamaktadır. Hem sistem maliyetleri hem de mühendislik maliyetleri göz önüne alındığında; bu testlerde, tek bir sistemin veya sistemlerin benzetimini yapabilen yazılımlara ihtiyaç duyulmaktadır. Farklı projelerdeki ve farklı varyantlardaki araçlarda tekrarlanan bu mühendislik eforunu azaltmak üzere gerçekleştirilen bu çalışmada kullanılan haberleşme protokolleri üzerinde karmaşık veri tipleri için otomatik kullanıcı arayüzleri oluşturan bir yazılım çerçevesi tasarlanmıştır. Bu makalede, belirtilen amaç doğrultusunda geliştirilen yazılım çerçevesinin teknik detayları ve proje çıktısının kullanımı ile elde edilen kazanımlar sunulmaktadır.

**Anahtar Kelimeler:** test uygulamaları, sistem test, haberleşme protokolleri, simülör, test yazılımı, yazılım çerçevesi

## Development of Rapid Software User Interface for Testing on Military Land Vehicles

Cankat AYKURT <sup>1,2</sup>

<sup>1</sup> FNSS Savunma Sistemleri A.Ş., Ankara, Türkiye

<sup>2</sup> Bilgisayar Mühendisliği Bölümü, Orta Doğu Teknik Üniversitesi, Ankara, Türkiye

cankat.aykurt@fnss.com.tr

cankat.aykurt@metu.edu.tr

**Abstract.** Electronics systems and their integration on the military land vehicles requires intensive testing. Most of the times required test are hardly conducted because of both the unavailability of the target systems and the importance of

the safety factors. While cost of these electronics systems and engineering efforts are taken into account; the required tests need test software which can simulate a system or most of the unavailable systems. In an attempt to decrease the repetitive engineering effort spent on different vehicle projects and different vehicle variants, a software framework is designed to create automated graphical user interfaces for complex data types on frequently used data buses in the vehicles. This article presents the purpose, motivation and technical development details of the software framework and the gained achievements of the project realization.

**Keywords:** test applications, system test, communication protocols, simulators, test software, software framework

## 1 Giriş

### 1.1 Problem Tanımı

Endüstride; sistem tasarımı, yazılım geliştirme ve test aşamalarında, alt sistemlerin haberleştiği gerçek sistemlerin benzetimini yapabilen yazılımlara ihtiyaç duyulmaktadır. Askeri kara araçları üzerindeki veri yollarına bilgisayar üzerinden bağlanarak alt sistemlerin anlık durumlarını görüntüleyip komuta etme ihtiyacı, tasarım ve test aşamalarında sürekli olarak gerekmektedir. Elektronik ve bağlı mekanik birimlerin testlerinde, diğer sistemlerden izole ve güvenli bir test ortamı sağlamak için çevre birimlerin davranışlarını taklit eden haberleşme uygulamalarının geliştirilmesi gerekmektedir. Grafiksel kullanıcı arayüzüne (GUI<sup>1</sup>) sahip bu yazılımların geliştirilme sürecindeki mühendislik eforunu azaltmak için askeri kara araçlarında sıklıkla kullanılan haberleşme protokolleri (CAN<sup>2</sup> Bus, seri haberleşme) öncelikli olarak belirlenmiş ve çalışma bu doğrultuda gerçekleştirilmiştir.

### 1.2 Çalışmanın Hedefi ve Yöntem

Askeri kara araçları operasyonel görevleri (komuta, haberleşme, istihkam, v.b.) ve fiziksel özellikleri (paletli, tekerlekli, amfibi v.b.) doğrultusunda birbirlerinden farklı donanımlarda tasarlanmaktadır. Bu çeşitlilik, tasarım ve test aşamasında yazılım gereksinimlerinin de çeşitlenmesini beraberinde getirmektedir.

Bu çalışmada, her araç projesi ve her araç varyantı için ayrı ayrı test arayüz yazılımları geliştirmek yerine hızla haberleşme ve grafiksel kullanıcı arayüzlerinin oluşturulmasına olanak sağlayacak bir yazılım çerçevesi geliştirmek hedeflenmiştir. Araç üzerinde haberleşme protokollerine bağlı sistemlere ait mesajların basit bir kullanıcı arayüzü üzerinden tanımlanmasıyla, durum görüntüleme ve komuta kontrol amaçlı grafiksel kullanıcı arayüzlerinin otomatik oluşturularak haberleşmenin sağlanması amaçlanmıştır. Bu sayede sistem tasarım, geliştirme ve test aşamalarında

---

<sup>1</sup> Graphical User Interface

<sup>2</sup> Controller Area Network

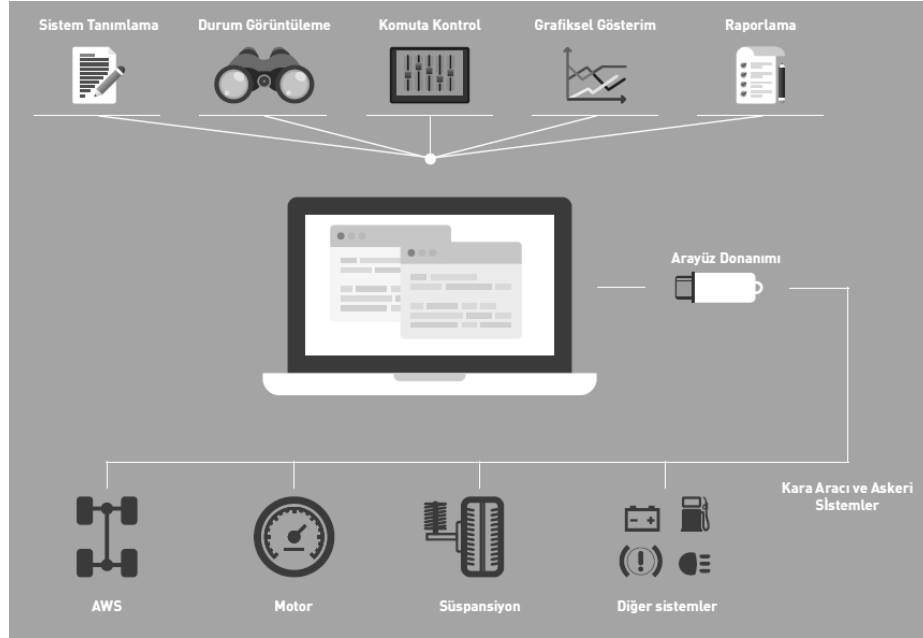
hızlı bir test ve doğrulama çözümü elde edilmiştir. Protokol üzerindeki verilerin anlık tanınması, sistemlerin durumlarının gösterimi ve kontrolü, grafik gösterimi ve raporlanması yazılım üzerinden hazırlanmıştır. Aynı zamanda, bu çalışma ile araçlara özel mesajların da tasarlanması ve arayüz kontrol dokümanlarının yazılım üzerinden otomatik oluşturulması sağlanmıştır.

Çalışma sonucunda ortaya çıkan yazılımın, ihtiyaç duyulduğunda yeni veri tipleri, yeni kullanıcı arayüz komponentleri ve yeni haberleşme protokolleri tanımlanmasına imkan verecek esneklikte modüller olarak tasarlanması sağlanmıştır.

## 2 Uygulama

Bu çalışmada yazılımın geliştirilmesi için Microsoft Visual C# dili [6] ve LINQ metodolojisi [7], sistem mesaj dosyalarının hazırlanması için XML dili [8] ve alternatif olarak JSON formatı [9] kullanılmış, raporlar HTML5 [10] ile dışa aktarılmış, dokümanların ve yazılımın konfigürasyon kontrolü SVN sürüm takip sisteminde [11] sağlanmıştır.

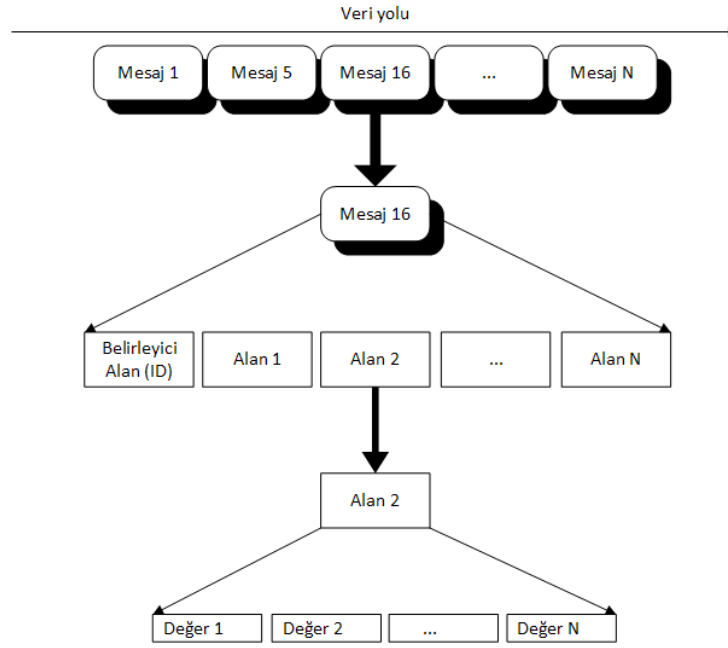
Yazılım çerçevesi, geliştirilebilir yapıda olup bu çalışma kapsamında yerine getirmeyi amaçladığı beş temel fonksiyona ait modüllerden ve haberleşme arayüzlerinden oluşmaktadır. Haberleşme arayüzleri için ilk etapta askeri kara araçlarında sıklıkla kullanılan CAN [12]-[15] protokolü ve RS-232 [16] seri haberleşme protokolü arayüzleri tasarlanmıştır. Amaçlanan beş temel fonksiyon; sistem tanımlama, durum görüntüleme, komuta kontrol, grafiksel gösterim ve raporlamadır.



Şekil 1. Geliştirilen yazılım çerçevesine ait fonksiyonel modüller

## 2.1 Sistem Tanımlama

Sistem Tanımlama modülünde, araç üzerindeki haberleşme protokollerine bağlı sistemlere ait mesajların yazılım çerçevesi üzerinde kolayca tanımlanması amaçlanmıştır. Bağlı sistemlere ait arayüz mesajları için genelleyici bir model oluşturulmuştur. Bu model vasıtasıyla sistem mesajlarına ait özelliklerin uygulama üzerinden basit bir şekilde tanımlanarak XML<sup>3</sup> ve/veya JSON<sup>4</sup> dosyalarında saklanması hedeflenmiştir. Veri yolu üzerindeki akış aşağıdaki şekilde modellenmiştir:



Şekil 2. Veri yolu üzerindeki mesajların en üst seviyeden modeli

Veri yolu üzerindeki mesajlar, belirleyici alanları ile tanımlanmış ve bunlara birer isim ataması yapılmıştır. Mesajların belirleyici alanları dışında en az bir adet anlamlı bölüm içereceği kabullenmesiyle yola çıkılarak mesajlar, alanlara bölünmüştür. Mesajın belirleyici alanı ve ismine ek olarak mesaja ait diğer alanlar bir liste olarak alt düğümlerde kaydedilmiştir.

Mesaj alanlarına ait kaydedilecek bilgiler; alan isimlerini, başlangıç ve bitiş konumlarını (bit sıraları), varsayılan değerlerini, veri tiplerini ve (varsa) çarpan katsayısı ile offset miktarını içermektedir. Alan isimleri, mesajları anlaşılır parçalara bölmek için kullanıcının tanımlaması veya arayüz kontrol dokümanında belirlenen ismin kullanılması için planlanmıştır. Bu alanın mesaj içindeki konumu başlangıç ve bitiş bit numaraları ile tanımlanmıştır. Varsayılan değerler, sistemin başlangıç durumunda

<sup>3</sup> eXtensible Markup Language

<sup>4</sup> JavaScript Object Notation

bu veri alanında görülmesi beklenen değeri içermektedir. Belirtilen başlangıç ve bitiş konumları arasındaki bit diziliminin nasıl ayrıştırılacağı da veri tipi bilgisi olarak saklanmaktadır.

Alanların alabileceği potansiyel değerler listesi aşağıdaki veri tiplerinden oluşmaktadır:

- Enum<sup>5</sup>: Kullanıcının tanımladığı veya arayüz kontrol dokümanında belirlenmiş tanımlı değerleri içeren veri,
- Byte: 8 bit uzunluğunda tamsayı içeren veri,
- UInt16: 16 bit uzunluğunda işaretli tamsayı içeren veri,
- UInt32: 32 bit uzunluğunda işaretli tamsayı içeren veri,
- UInt64: 64 bit uzunluğunda işaretli tamsayı içeren veri,
- Int16: 16 bit uzunluğunda işaretli tamsayı içeren veri,
- Int32: 32 bit uzunluğunda işaretli tamsayı içeren veri,
- Int64: 64 bit uzunluğunda işaretli tamsayı içeren veri,
- Float: 32 bit uzunluğunda işaretli reel sayı içeren veri,
- Double: 64 bit uzunluğunda işaretli reel sayı içeren veri,
- None: Henüz anlamlandırılmamış veri; mesaj içerisinde rezerve olarak bırakılmış boş bit dizilimleri için kullanılabilir.

Şekil 3, üç pencereyi göstermektedir: Mesaj Bilgileri, Alan Bilgileri ve Değer Bilgileri. Mesaj Bilgileri penceresinde Mesaj İsmi: Motor ECU Sıcaklık Mesajı ve Betirleyici Alan (ID): 0x0CA1201 yazmaktadır. Alan Bilgileri penceresinde Alan İsmi: Fan Durumu, Başlangıç: 0, Bitiş: 3, Varsayılan Değer: 0010 ve Veri Tipi: Enum yazmaktadır. Değer Bilgileri penceresinde Bit Dizilimi: 0010 ve Anlamı: Fan Çalışıyor yazmaktadır. Her pencere Tamam butonuna sahiptir.

Şekil 3. Mesaj, Alan ve Değer Tanımlamaları

Mesaj alanlarının alabileceği değerler, yukarıda verilen veri tiplerine göre belirlenmiştir. Değerler, bit dizilimleri ve değer isimlerini içermektedir. Veri tipi enum olarak tanımlanan alanların olası değerleri, sisteme bit dizilimleri ile ilişkilendirilerek isimlendirilir. Bu pratik üst seviye ayrıştırma metoduyla veri yolu üzerindeki test edilecek sistem veya sistemler, yazılım çerçevesi üzerinde tanımlanır ve kaydedilir. Böylece sistemlerin veri arayüzlerinin kayıt altında tutulması ve ilerleyen zamanlarda arayüz kontrolleri üzerinde yapılacak olası değişikliklerin kolayca güncellenmesi de sağlanmaktadır.

<sup>5</sup> Enumerated Type

## 2.2 Durum Görüntüleme

Durum Görüntüleme modülünde, araçta bağlanılan veri yolu üzerindeki sistemlere ait mesajların grafiksel kullanıcı arayüzlerinin otomatik olarak hazırlanması amaçlanmıştır. Sistem Tanımlama modülü ile tanımlanan kayıt dosyaları bu modül ile açıldığında yazılım çerçevesi mesajlar için sekme sayfaları hazırlamakta ve bu sayfaları mesajlara bölerek, önceden tanımlanmış alternatifler arasından veri tiplerine uygun grafiksel kullanıcı arayüz araçlarını (metin kutusu, çoktan seçmeli kutu, doğrusal gösterge, dairesel gösterge, vb.) yerleştirmektedir.

Mesaj 1	Mesaj 2	Mesaj 3	Mesaj 4	Mesaj 5	Mesaj 6
Alan 1	Alan 2	Alan 3	Alan 4	Alan 5	
Numerik Değer	Numerik Değer	Enum	Enum	Numerik Değer	
Alan 6	Alan 7	Alan 8	Alan 9	Alan 10	
Numerik Değer	Enum	Numerik Değer	Numerik Değer	Enum	
...	Alan N				
Enum	Numerik Değer				

Şekil 4. Durum Görüntüleme ekranları düşük duyarlıklılı modeli

Bağlanılan haberleşme protokolü üzerindeki veriler anlık minimum gecikmeyle ayrıştırılarak alanlara ait değerler otomatik olarak oluşturulan grafiksel kullanıcı arayüzleri üzerinde görüntülenmekte ve güncellenmektedir.

## 2.3 Komuta Kontrol

Komuta Kontrol modülünde, araçta bağlanılan veri yolu üzerindeki sistemlere ait mesajların diğer sistemlere ilgili mesajları gönderebilmesi için gerekli grafiksel kullanıcı arayüzlerinin otomatik olarak hazırlanması amaçlanmıştır. Sistem Tanımlama modülü ile tanımlanan kayıt dosyaları bu modül ile açıldığında yazılım çerçevesi gönderilecek olası mesajlara, bu mesajların alabileceği olası değerlere göre sekme sayfaları ve önceden tanımlanmış alternatifler arasından veri tiplerine uygun grafiksel kullanıcı arayüz araçlarını (metin kutusu, çoktan seçmeli kutu, doğrusal gösterge, dairesel gösterge vb.) yerleştirmektedir.

Bağlanılan haberleşme protokolü üzerinde ilgili mesajlar, oluşturulan grafiksel kullanıcı arayüzü üzerinden seçimler yapılarak kullanıcı isteğine göre veya arayüz

kontrol dokümanında tanımlandığı şekilde tek seferlik veya belirlenen periyotlarda gönderilebilmektedir.

## 2.4 Grafikselleştirme

Grafikselleştirme modülünde, araçta bağlanılan veri yolu üzerindeki sistemlere ait mesajların kullanıcı tarafından seçilen alanlarının zamana bağlı grafiklerinin çizdirilmesi amaçlanmıştır. Farklı mesajların bir arada gösterilmesiyle sistem içi ve/veya sistemler arası ilişkilerin test edilmesi, gecikmelerin, durum ve değer değişikliklerinin görüntülenmesi sağlanarak kullanıcıya görsel bir araç daha sağlanmıştır.

Bu modül üzerinde, grafiklerin serileştirilerek saklanması ve daha sonra yeniden çizdirilerek test durumlarının gerektiğinde tekrar görselleştirilmesi amaçlanmıştır.

## 2.5 Raporlama

Raporlama modülünde, araçta bağlanılan veri yolu üzerindeki sistemlere ait mesajların kullanıcı tarafından seçilen alanlarının istenilen zaman aralıkları içinde doküman olarak gösterilmesi ve raporlama amacıyla kaydedilmesi amaçlanmıştır. Bu modül sayesinde sistem içi ve/veya sistemler arası ilişkilerin test edilmesi, durum ve değer değişikliklerinin zamanlarının kayıt altına alınması ve geriye dönük incelenmesi sağlanmıştır.

Bu modül üzerinde hazırlanan raporlar Microsoft Word, PDF ve HTML5 olarak dışa aktarılabilir.

## 3 Kazanımlar

Proje sonunda ortaya çıkan yazılımın sağladığı kazanımın ölçülebilmesi için kurumsal ERP<sup>6</sup> yazılımındaki verilerden yardım alınmıştır. Buna göre, hedef test platformu olarak seçilen M113 aracı için önceden hazırlanan durum görüntüleme ve komuta kontrol amaçlı simulator yazılımının tasarımı için sisteme giren adam saat bilgisi çekilmiştir. Yazılımı hazırlayan aynı personelden proje sonucunda ortaya çıkan yazılımı kullanarak M113 ICD<sup>7</sup>'sine göre tekrar tasarım yapılması istenmiştir. Bu bilgiler ışığında proje ile elde edilen kazanımları sayısal olarak aşağıdaki şekilde ifade edebiliriz:

- M113 aracı için daha önce 80 adam saatlik (2 haftalık) bir çalışma sonucunda ortaya çıkan grafikselleştirme kullanıcı arayüzü yazılımı, bu çalışma sonucu elde edilen yazılım çerçevesi ile 2 adam saatte hazırlandı.
- M113 aracı için 4445 satır kod ile hazırlanan grafikselleştirme kullanıcı arayüzü yazılımı, bu çalışma sonucu elde edilen yazılım çerçevesi ile kod yazılmadan sadece veri yolundaki mesajlara ait parametrelerin girilmesiyle otomatik olarak elde edildi.

---

<sup>6</sup> Enterprise Resource Planning

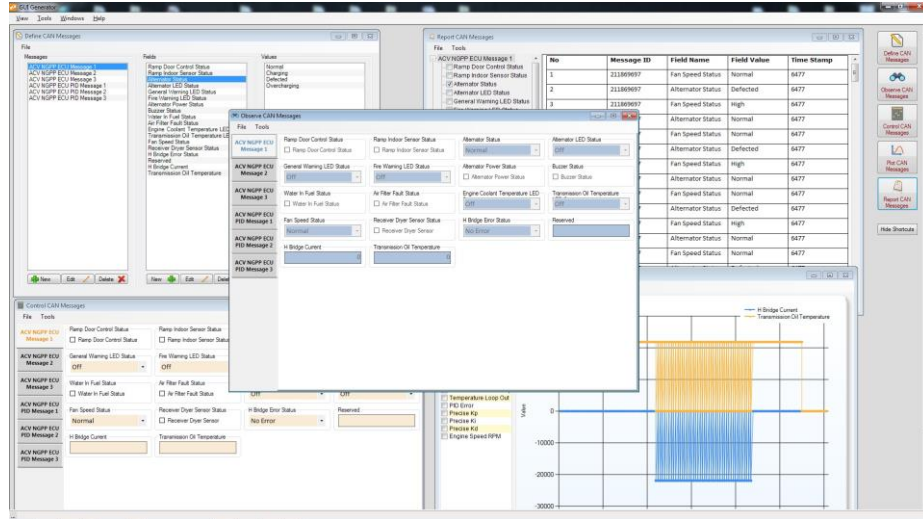
<sup>7</sup> Interface Control Document

Bu çalışma ile her araç ve her varyant için tekrarlanan grafiksel kullanıcı arayüzü yazılımı geliştirme aşamalarını hızlandıracak ve harcanan mühendislik eforunu ciddi ölçekte azaltacak bir yazılım ve çerçeve oluşturuldu.

#### 4 Sonuçlar ve Gelecek Çalışmalar

Sistem tasarımı, yazılım geliştirme ve test aşamalarında, alt sistemlerin haberleştiği gerçek sistemlerin benzetimini yapabilen yazılımlara duyulan ihtiyacı karşılamak ve harcanan mühendislik eforunu azaltmak amacıyla geliştirilen bu yazılım çerçevesi sayesinde operasyonel görevleri (komuta, haberleşme, istihkam, v.b.) ve fiziksel özellikleri doğrultusunda (paletli, tekerlekli, amfibi v.b.) birbirlerinden farklı donanımlarda tasarlanan askeri araçlarda grafiksel kullanıcı arayüzüne sahip test ve benzetim yazılımları ihtiyacının hızlı bir şekilde karşılanması sağlanmıştır.

Araçlar üzerinde haberleşme protokollerine bağlı sistemlere ait mesajların basit bir kullanıcı arayüzü üzerinden tanımlanmasıyla, durum görüntüleme ve komuta kontrol amaçlı grafiksel kullanıcı arayüzlerinin otomatik oluşturularak sistemlerle haberleşmenin sağlanması; bu sayede sistem tasarım, geliştirme ve test aşamalarında hızlı bir test ve doğrulama çözümünün elde edilmesi sağlanmıştır. Protokol üzerindeki verilerin anlık tanımlanması, sistemlerin durumlarının gösterimi ve kontrolü, grafik gösterimi ve raporlanması ile farklı amaçlarda tasarım ve test mühendislerinin harcayacağı efor azaltılmış, insan bilgisayar etkileşimi yüksek [17], [18] bir yazılım çözümü sunulmuştur.



Şekil 5. Geliştirilen yazılım çerçevesinden testler esnasında alınan bir ekran görüntüsü

Çalışma sonucunda ortaya çıkan yazılımın, ihtiyaç duyulduğunda yeni veri tipleri, yeni kullanıcı arayüz araçları ve yeni haberleşme protokolleri tanımlanmasına imkan



verecek esneklikte modüler olarak tasarlanması sağlanmıştır. Doğacak gereksinimler doğrultusunda yeni yetenekler kazandırılacaktır. Öngörülen yetenekler arasında; halihazırda eklenen haberleşme arayüzlerine ek olarak TCP/IP ve UDP ulaşım katmanlarının ilave edilmesi ve testler esnasında araçlara kablosuz erişim bulunmaktadır.

## 5 Teşekkür

Proje süresince desteğini esirgemeyen ve emeği geçen yöneticilerime, test süreçlerinde katkı sağlayan başta Burak Arısoy olmak üzere tüm çalışma arkadaşlarıma teşekkür ederim.

## Kaynakça

1. IEEE Std 1058-1998, IEEE Standard for Software Project Management Plans, The Institute of Electrical and Electronics Engineers, Inc., December 1998
2. IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, The Institute of Electrical and Electronics Engineers, Inc., June 1998
3. Wiegers, K., Beatty, J. 2013. Software Requirements. 3<sup>rd</sup> ed. Microsoft Publishing.
4. IEEE Std 1016-2009, IEEE Standard for Information Technology—Systems Design—Software Design Descriptions, The Institute of Electrical and Electronics Engineers, Inc., July 2009
5. IEEE Std 829-2008, IEEE Standard for Software and System Test Documentation, The Institute of Electrical and Electronics Engineers, Inc., July 2008
6. C# Guide  
<https://docs.microsoft.com/en-us/dotnet/csharp/csharp>
7. Language Integrated Query  
<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/>
8. Extensible Markup Language (XML)  
<https://www.w3.org/XML/>
9. Introducing JSON  
<http://www.json.org/>
10. HTML5  
<https://www.w3.org/TR/html5/>
11. Apache Subversion  
<https://subversion.apache.org/>
12. BOSCH CAN Specification, Version 2.0, Robert Bosch GmbH, 1998
13. Di Natale, M., Zeng, H., Giusto, P., Ghosal, A. Understanding and Using the Controller Area Network Communication Protocol: Theory and Practice. 1<sup>st</sup> ed., 2012
14. SAE Vehicle Application Layer, J1939-71, Society of Automotive Engineers, 2003
15. PCAN-USB User Manual, CAN Interface for USB, Peak-System Technik GmbH, 2016
16. EIA standard RS-232-C: Interface between Data Terminal Equipment and Data Communication Equipment Employing Serial Binary Data Interchange, Electronic Industries Association. Engineering Dept., 1969
17. Mayers, B. A., Rosson M. B., Survey on User Interface Programming, 1992
18. Gajos, K. Z., Czerwinski, M., Tan, D. S., Weld, D. S., Exploring the Design Space for Adaptive Graphical User Interfaces, 2006