

**T.C.  
SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**ELEKTROMOBİL ARAÇ KONTROL SİSTEMİ  
TASARIMI VE ENTEGRASYONU**

**YÜKSEK LİSANS TEZİ  
Yunus Emre DURGUN**

**Enstitü Anabilim Dalı : MEKATRONİK MÜHENDİSLİĞİ**  
**Tez Danışmanı : Dr. Öğretim Üyesi Barış BORU**

**Mayıs 2019**

T.C.  
SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**ELEKTROMOBİL ARAÇ KONTROL SİSTEMİ  
TASARIMI VE ENTEGRASYONU**


**YÜKSEK LİSANS TEZİ**


**Yunus Emre DURGUN**

**Enstitü Anabilim Dalı : MEKATRONİK MÜHENDİSLİĞİ**

**Bu tez 28/05/2019 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.**

  
**Doç. Dr.**  
**Cüneyt BAYILMIŞ**  
**Jüri Başkanı**

  
**Dr. Öğr. Üyesi**  
**Barış BORU**  
**Üye**

  
**Dr. Öğr. Üyesi**  
**Ahmet KARACA**  
**Üye**

## **BEYAN**

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Yunus Emre DURGUN  
28/05/2019

## TEŞEKKÜR

Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Dr. Barış BORU'ya teşekkürlerimi sunarım.

Bütün hayatım boyunca benden desteklerini esirgemeyen annem Ayşe DURGUN ve babam Mahmut Recai DURGUN'a teşekkür ederim.

Tez sürecimde teknik anlamda yardımlarını esirgemeyen sevgili Bedreddin KARAKUŞ'a, Ertan ALDOĞAN'a, Erdoğan GÖK'e ve Oktay KARAKAYA'ya teşekkür ederim.

# İÇİNDEKİLER

TEŞEKKÜR.....	i
İÇİNDEKİLER .....	ii
SİMGELER VE KISALTMALAR LİSTESİ.....	iv
ŞEKİLLER LİSTESİ .....	v
TABLolar LİSTESİ .....	vii
ÖZET.....	viii
SUMMARY .....	ix
BÖLÜM 1.	
GİRİŞ .....	1
1.1. Literatür Taraması .....	1
1.2. Elektrikli Araçlar .....	6
1.3. Çalışmanın Özgünlüğü.....	7
BÖLÜM 2.	
GENEL BİLGİLER.....	9
2.1. Araç Kontrol Sistemi.....	9
2.2. ARM İşlemci Mimarisi .....	10
2.3. Haberleşme Protokolleri.....	12
2.3.1. CANBUS haberleşme.....	12
2.3.1.1. Standart ve genişletilmiş mesaj çerçevesi.....	13
2.3.1.2. CANBUS mesaj çerçeve yapısı.....	15
2.3.1.3. CANBUS hattı .....	15
2.3.1.4. CANBUS hata önleme.....	17
2.3.2. UART haberleşmesi.....	18
BÖLÜM 3.	
MATERYAL VE YÖNTEM.....	21
3.1. Materyal .....	21
3.1.1. STM32F407 Discovery geliştirme kartı .....	21

3.1.2. Geliştirme ortamı.....	24
3.2. Yöntem ve Algoritmalar.....	27
3.2.1. Gömülü yazılım.....	27
3.2.1.1. ADC okuma.....	32
3.2.1.2. Frekans okuma.....	35
3.2.1.3. PWM çıkış.....	39
3.2.1.4. Kalibrasyon yapısı.....	41
3.2.1.5. CANBUS haberleşme.....	42
3.2.1.6. Seri haberleşme.....	43
3.2.2. Arayüz yazılımı.....	52
3.2.2.1. Seri porttan veri ama.....	53
3.2.2.2. Verilerin gösterilmesi.....	54
3.2.2.3. Kalibrasyon sekmesi.....	55
3.2.2.4. Giriş testi.....	56
3.2.2.5. Araç Bilgileri.....	57
<b>BÖLÜM 4.</b>	
<b>ARAŞTIRMA BULGULARI VE TARTIŞMA.....</b>	<b>58</b>
4.1. Test Aracı ve Sistem Bilgileri.....	58
4.2. Araç Teknik Özellikleri.....	59
4.3. Araç Kontrol Sistemi Bağlantı Devresi.....	59
4.4. Test Düzeneği.....	61
4.5. Araç Kontrol Sistemi Kayıt Edilen Veriler.....	62
4.6. AKS ile GÇÜ Arası CANBUS Haberleşme Testi.....	64
<b>BÖLÜM 5.</b>	
<b>ÖNERİLER VE SONUÇ.....</b>	<b>66</b>
5.1. Sonuç.....	66
5.2. Öneriler.....	67
<b>KAYNAKÇA.....</b>	<b>68</b>
<b>ÖZGEÇMİŞ.....</b>	<b>71</b>

## SİMGELER VE KISALTMALAR LİSTESİ

ABS	: Anti-Lock Braking System (Kilitlenme Karşıtı Fren Sistemi)
ADC	: Analog-Digital Converter (Analog-Dijital Çevirici)
AKS	: Araç Kontrol Sistemi
ARM	: Acorn Risc Machine
ARR	: Auto Reload Register – Otomatik Yükleme Kaydedicisi
BYS	: Batarya Yönetim Sistemi
CAN	: Control Area Network (Kontrol Alan Ağı)
GKS	: Gövde Kontrol Sistemi
GM	: Gösterge Modülü
GÇB	: Giriş Çıkış Birimi
HS	: Havalandırma Sistemi
IC	: Input Capture – Giriş Yakalama
IOT	: Internet of Things (Nesnelerin İnterneti)
ISO	: International Standardization Organization (Uluslararası Standardizasyon Örgütü)
MKS	: Motor Kontrol Sistemi
PWM	: Pulse Width Modulation (Darbe Genişlik Modülasyonu)
USB	: Universal Serial Bus (Evrensel Seri Yol)

## ŞEKİLLER LİSTESİ

Şekil 1.1. Elektrikli araçlar için tasarlanmış örnek topoloji.....	3
Şekil 2.1. Elektrikli araç topolojisi.....	10
Şekil 2.2. ARM Cortex-M işlemci mimarileri. ....	11
Şekil 2.3. ISO standart mimarisi. ....	13
Şekil 2.4. Standart mesaj çerçevesi. ....	14
Şekil 2.5. Genişletilmiş mesaj çerçevesi. ....	14
Şekil 2.6. CAN hattı gösterimi. ....	16
Şekil 2.7. Hata sayacı çalışma şeması.....	18
Şekil 2.8. Seri haberleşme paket yapısı.....	19
Şekil 3.1. STM32F4-Discovery kitinin bölümleri. ....	22
Şekil 3.2. STM32CubeMX işlemci giriş-çıkış seçimi. ....	25
Şekil 3.3. STM32F4 clock ayarları. ....	25
Şekil 3.4. Yazılım mimarisi. ....	28
Şekil 3.5. 1ms timer kesme algoritması. ....	29
Şekil 3.6. Gömülü yazılım akış diyagramı.....	31
Şekil 3.7. ADC giriş hesaplama algoritması. ....	33
Şekil 3.8. ADC parametre ayarları.....	34
Şekil 3.9. ADC DMA ayarları. ....	35
Şekil 3.10. Giriş yakalama donanımı blok diyagramı.....	36
Şekil 3.11. Frekans giriş hesaplama algoritması.....	37
Şekil 3.12. Timer2 frekans giriş parametreleri.....	38
Şekil 3.13. Timer2 frekans giriş DMA ayarları. ....	39
Şekil 3.14. Farklı Duty Cycle değerlerinde PWM sinyalleri. ....	40
Şekil 3.15. PWM çıkış parametreleri. ....	40
Şekil 3.16. Lineer interpolasyon. ....	41
Şekil 3.17. Haberleşme paket yapısı. ....	44
Şekil 3.18. Arayüz bağlantı konfigürasyon seçenekleri.....	53



Şekil 3.19. Bilgisayar arayüzü verilerin gösterilmesi. ....	55
Şekil 3.20. Kalibrasyon sekmesi GÇÜ.....	55
Şekil 3.21. Kalibrasyon sekmesi AKS. ....	56
Şekil 3.22. Bilgisayar arayüzünde giriş test sekmesi. ....	57
Şekil 3.23. Bilgisayar arayüzünde aracın teknik bilgileri. ....	57
Şekil 4.1. Test aracı genel görünüm.....	58
Şekil 4.2. Sistemin blok diyagramı. ....	59
Şekil 4.3. Bağlantı kartı baskı devre çizimi. ....	60
Şekil 4.4. Bağlantı kartı 3 boyutlu görünüm. ....	60
Şekil 4.5. Test aracı ve test kartı resimleri. ....	61
Şekil 4.6. Seri port okuma programı ile paketlerin ayrılmış gösterimi.....	62
Şekil 4.7. Kayıtlı veriler. ....	62
Şekil 4.8. Gelen paketlerin kayıt şekli. ....	63
Şekil 4.9. AKS ve Giriş-Çıkış ünitesi arasındaki CAN mesajları.....	65

## TABLolar LİSTESİ

Tablo 2.1. Hat uzunluđuna bađlı maksimum veri iletim hızı.....	16
Tablo 3.1. STM32CubeMx Giriş-Çıkış Ünitesi pin tanımlamaları.....	26
Tablo 3.2. STM32CubeMx Araç Kontrol Sistemi pin tanımlamaları.....	27
Tablo 3.3. CAN hattına gönderilen bilgiler. ....	42
Tablo 3.4. Grup tanımlamaları. ....	43
Tablo 3.5. Mesaj ID tanımlamaları. ....	43
Tablo 3.6. Komut açıklamaları.....	45
Tablo 3.7. Sistem durumu görüntüleme komut paketi.....	46
Tablo 3.8. Arayüz tarafından AKS ye gönderilen kalibrasyon bilgileri istek paketi. ....	46
Tablo 3.9. AKS tarafından gönderilen gaz pedalı kalibrasyon paketi. ....	46
Tablo 3.10. AKS tarafından gönderilen direksiyon açısı kalibrasyon paketi. ....	47
Tablo 3.11. AKS tarafından gönderilen sol tekerlek açısı kalibrasyon paketi.....	47
Tablo 3.12. AKS tarafından gönderilen sağ tekerlek açısı kalibrasyon paketi. ....	48
Tablo 3.13. AKS tarafından gönderilen PWM kalibrasyon paketi. ....	48
Tablo 3.14. AKS tarafından gönderilen batarya gerilim kalibrasyon paketi. ....	49
Tablo 3.15. AKS tarafından gönderilen batarya sıcaklık kalibrasyon paketi. ....	49
Tablo 3.16. Arayüz tarafından gönderilen gaz pedalı kalibrasyon paketi.....	49
Tablo 3.17. Arayüz tarafından gönderilen direksiyon açısı kalibrasyon paketi.....	50
Tablo 3.18. Arayüz tarafından gönderilen sol tekerlek açısı kalibrasyon paketi. ....	50
Tablo 3.19. Arayüz tarafından gönderilen sağ tekerlek açısı kalibrasyon paketi. ....	51
Tablo 3.20. Arayüz tarafından gönderilen PWM kalibrasyon paketi. ....	51
Tablo 3.21. Arayüz tarafından gönderilen batarya gerilim kalibrasyon paketi.....	51
Tablo 3.22. Arayüz tarafından gönderilen batarya sıcaklık kalibrasyon paketi.....	52
Tablo 4.1. Araçta kullanılan üniteler listesi. ....	59
Tablo 4.2. Seri porttan alınan işlenmiş veri tablosu.....	64

# **ELEKTROMOBİL ARAÇ KONTROL SİSTEMİ TASARIMI VE ENTEGRASYONU**

## **ÖZET**

Günümüzde çevreye yüksek oranda zarar veren geleneksel fosil yakıt sistemlerinden, hibrit(benzin-elektrik) ve elektrikli araç sistemlerine olan yönelim artmaktadır. Elektrikli araçlarda geleneksel yakıtlı araçlara göre kontrol edilmesi gereken elektrik motoru ve batarya ünitesi gibi alt komponentlerin kontrolü Araç Kontrol Sistemi(AKS) tarafından yapılmaktadır.

Bu çalışmada AKS sisteminin tasarımı, yazılımının yapılması ve araç içindeki diğer aygıtlardan bilgi alınması ve verilerin değerlendirilmesi işlemi yapılacaktır.

Gömülü yazılım ve bilgisayar arayüz programı bu sistemin iki temel yazılım bileşenidir. Arayüz yazılımında araç kontrol sisteminin giriş çıkış bilgilerinin görüntülenmesi, kalibrasyon değerlerinin görüntülenmesi ve değiştirilmesi, test verilerinin alınarak kaydedilmesi sağlanmıştır. Mikrodenetleyici olarak STM firmasının STM32F407 işlemcisi C dili kullanılarak ve sistemin kontrolü için gerekli arayüz programı C# programlama dili kullanılarak geliştirilmiştir.

Araç içi donanımların birbirleri ile haberleşmesi için CANBUS haberleşme yöntemi seçilmiştir. CANBUS haberleşme uzun yıllardır araçlarda kullanılan yüksek doğruluk oranı ve hızlı haberleşmesi ile güvenilir ve araç içi sistemimizde ihtiyaçları karşılayan bir haberleşme protokolüdür.

Anahtar kelimeler: CANBUS Araç Kontrol Sistemi, Kontrol Alan Ağı, Elektrikli Araç Kontrol, Araç Kontrol Ünitesi

# **DESIGN AND INTEGRATION OF ELECTROMOBILE VEHICLE CONTROL SYSTEM**

## **SUMMARY**

Today, hybrid and electric vehicle systems are increasing trend according to fossil fuel systems which have high impact on the environment. According to fossil fuel systems electric vehicles have some components like electrical motor and battery unit, Vehicle Control Unit controls these components.

In this study, the design of Vehicle Control Unit, software and communication of other devices in the vehicle will be provided by using CAN network.

Embedded software and user interface are most important parts of this project. In user interface, inputs and outputs, calibration parameters, test values are shown. STM32F407 discovery board is used to develop embedded software. User interface is written with C# language.

**Keywords:** CANBUS Vehicle Control System, Control Area Network, Electromobile Vehicle Control, Vehicle Control Unit

## **BÖLÜM 1. GİRİŞ**

Günümüzde otomotiv endüstrisi hızla büyümektedir. Çevreye verdikleri zararlar ve dünyada fosil yakıt rezervlerinin azalması sebebiyle bu tür araçlara alternatif olan elektrikli ve hibrit araçlara önem artmıştır. Elektrikli araçlarda ‘Batarya Yönetim Ünitesi’, ‘Araç Kontrol Sistemi’, ‘Gösterge Modülü’ vb. donanımlar mevcuttur. Aracın düzgün çalışabilmesi için bu donanımların birbirleri ile doğru ve hızlı bir şekilde haberleşmesi gerekmektedir. CAN haberleşme protokolü araç içi haberleşme için uygun ve etkili bir çözüm sunar. Elektrikli araçta bulunan ünitelerin haberleşmesi

Projede ST firmasının geliştirme kartı olan STM32F4-Discovery kartı kullanılmıştır. Discovery kartının üst kısmına araç içi bağlantıların yapılabilmesi için bir tasarım yapılarak baskı devresi çizilmiş üretilmiştir.

Tezin ikinci bölümünde araç kontrol sistemi, ARM mimarisi ve kullanılan haberleşme protokolleri hakkında bilgi anlatılmaktadır. Üçüncü bölümde projede kullanılan donanım ve yazılım araçları ile kullanım şekilleri açıklanmıştır.

### **1.1. Literatür Taraması**

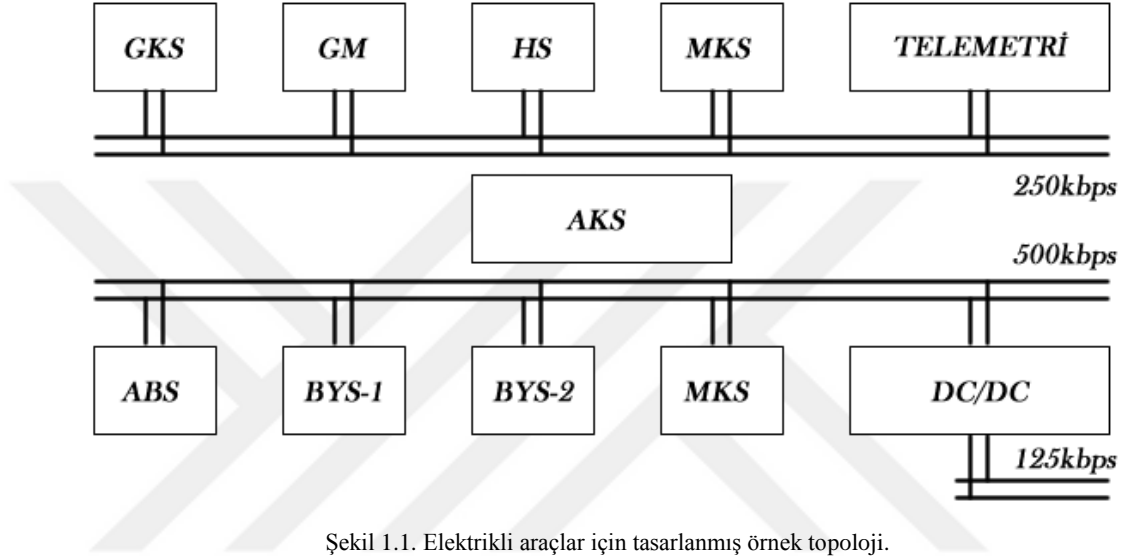
Elektrikli araçlar bilinenden daha eski bir geçmişe sahiptir. Elektrikli araçlar ilk olarak Joseph Henry tarafından DC güç beslemeli motorlar tanıtıldıktan sonra görülmüştür. 1834’te Hollandalı Professor Stratingh küçük model bir elektrikli araç üretmiştir. Gerçek boyutlarda olan elektrikli araç ise 1834 yılında Thomas Davenport tarafından Amerika’da üretilmiştir. Daha sonra 1847’de Moses Darmer tarafından ilk yolcu taşıma amaçlı elektrikli araçlar üretimi yine Amerika’da gerçekleştirilmiştir. Yeniden şarj edilebilir hücreler bu tarihlere henüz bulunmamaktaydı. İlk kez elektrikli araçların kullanılabilir olması durumu, 1865’te Frenchman Gaston Plante’nin depolanabilir bataryaları

keşfetmesi ve 1881’de Camille Faure’nin bunları geliştirmesi ile mümkün olmuştur (Didik, 2007). Elektrikli araçlar 1900’lü yıllarda çeşitli uygulamalar altında gelişimine devam etmiştir. Tarihsel olarak 19. yüzyıl ortalarındaki ilk araçların bataryalı elektrikli araçlar olmasının nedeni içten yanmalı motorlarla karşılaştırıldığında elektrikli motorların daha basit bir yapıya sahip olmasıydı. Ancak içten yanmalı motora sahip araçlar, bataryalı elektrikli araçların daha düşük menzile sahip olması ve o dönemdeki bataryaların fosil yakıtlarla karşılaştırıldığında daha düşük enerji beslemesi sağlamasından dolayı daha yaygın olarak tercih edilmiştir. Bataryalı elektrikli araçların yaygınlaşmasında batarya teknolojisindeki gelişmeler önemli bir role sahiptir. Ayrıca büyük üretici firmaların aerodinamik dizayn ile sistem entegrasyonu konusundaki geliştirmeleri sayesinde elektrikli araçların caddelere çıkmasında etkisi büyük olmuştur (Chen , Round, & Duke, 2002). Yeni nesil bataryalar daha düşük hacimde daha yüksek enerji depolama kabiliyetine sahiptir. Elektrikli araçlarda kullanılan bataryalar sürüş menzilinini elde edebilecek uygun boyutta olmak zorundadır. Bu nedenle batarya paketleri oluşturulur. Batarya paketleri, birçok hücrenin seri ve paralel bağlanması ile elde edilmektedir. Böylece elektrikli motorların sürülmesinde gereken uygun gerilim miktarını ve uzun süre sağlanması mümkün olabilir. Çeşitli durumlarda batarya paketlerinin dizaynı, uzun ömürlü kullanım için şarj yöntemleri konuları üzerinde durulması gerekebilmektedir (Chakraborty & Simões, 2013)

Elektrikli araçlarda sürüş esnasında her türlü sensör ve aracın diğer bilgileri araç kontrol ünitesi tarafından paylaşılmalıdır. Araçlarda haberleşme ağı kullanımı yeni bir konsept değildir. 1990’larda tasarlanan CANBUS haberleşme sistemi günümüzde neredeyse tüm araçlarda kullanılmaktadır. CANBUS 2.0A standardı ‘standart çerçeve’ ve CANBUS 2.0B standardı ‘genişletilmiş çerçeve’ olarak bilinir. Standart çerçeve 11bit tanımlayıcı bite sahipken genişletilmiş çerçeve 29bit tanımlayıcı bite sahiptir.

Veri iletim hızı bakımından araç haberleşme protokolünü 4 gruba ayırabiliriz. 1)A sınıfı ağ, LIN (local interface network) gibi ağ hızı 10 Kbit/s den aşağıdır ve hat 16 tane düğüme kadar destekler. Dikiz aynası ve koltuk ısıtma sistemi gibi. 2) B sınıfı ağ, 10 Kbit/s ile 125 Kbit/s arasında iletim hızı olan ağdır. Gösterge panelleri ve gövde kontrol ünitesi(BCU) bu ağda yer alır. 3) C sınıfı ağ, 250 Kbit/s ile 1 Mbit/s arası iletim hızında

olan ağdır. Motor bilgileri ve sürüşle ilgili olan bilgiler bu ağda yer almaktadır. 4) C sınıfı ağ, 1 Mbit/s hız olan ağdır. Çoğunlukla güç sistemleri, güvenlik sistemleri, navigasyon sistemleri ve video sistemleri için kullanılır. Örneğin güç kontrol ünitesi(PCU), ivmelenme kontrol ünitesi(ACU) ve kilitleme önleyici fren sistemi(ABS) üniteleri bu ağdadır. Elektrikli araçlar için tasarlanmış örnek topoloji Şekil 1.1 de gösterilmektedir. (Cheng, Xiaoyu, Yongfu, & Wei, 2014)



Şekil 1.1. Elektrikli araçlar için tasarlanmış örnek topoloji.

Can hattının haberleşmesinde kullanılan protokolün belirlenmesinde uygulanan adımlar şu şekildedir:

- Mesajları gruplandır: Bir düğümün göndereceği veri sayısı az ise bunun için tek bir mesaj çerçevesi yeterlidir. Ancak düğümden gönderilecek olan veri fazla ise tek bir mesaj çerçevesi talepleri karşılamaz. Bu nedenle verileri sınıflandırarak farklı mesaj çerçevelerinde göndermek önem arz eder. Birbiri ile alakalı olan verileri aynı grupta gönderecek şekilde ayarlanmalı.
- Mesajların gönderim sıklığını belirle: Gönderim sıklığı mesajın önemine göre belirlenmektedir. Bir verinin önemine göre veri gönderim sıklığı 10ms, 20ms, 50ms, 100ms ve 1s olarak ayarlanabilir.
- Mesaj önceliğini belirle: Ayarlanabilecek 8 adet öncelik değeri vardır. 0 en yüksek 7 en düşük öncelik değeridir. Öncelik ayarı yapılmayan bütün mesajlar öncelik ayarı 3'tür.

- Dügüm adresini belirle: CAN hattında düğüm adresi eşsizdir. Bir ağda bir düğüm adresi yalnızca tek bir düğüme verilebilir.
- Mesajın gönderim yöntemini seç: Mesaj gönderimi noktadan noktaya, periyodik gönderim veya talep gönderimi olarak seçilebilir.
- Belirlenen özelliklere göre yazılımı oluştur: Bütün özellikler belirlendikten sonra ilgili mesaj yapısının yazılım tarafına geçirilmesi ve test edilmesi sağlanmalıdır. (Li & Ji, 2013)

Toyota MR2 için CAN kontrol sisteminin tasarım ve uygulaması bu testte sunulmaktadır. 312V'luk nominal DC akü gerilimi 48V DC'lik orta gerilime ve orta geriliminden 12V DC'ye düşürür. CAN kontrol alan ağı sistemde tamamen çalışır durumdadır. MR2'de seçilen yardımcı yüklerin simüle edilmesi için kullanılan tüm yükler başarıyla çalıştırıldı. (Azzeh, 2005)

Model asansörün dağıtık olarak kontrolünü gerçekleştirmek için; dört kat için birer kontrol kartı, bir adet kabin içi kontrol kartı ve bir adet de ana kontrol kartı gerçekleştirilmiştir. Her bir kontrol kartında bir PIC16F877 mikrodenetleyicisi kontrol ve haberleşmeyi sağlamak üzere kullanılmıştır. Kontrol kartlarının diğer kartlarla haberleşmesini CAN-Bus üzerinden sağlamak için her kontrol kartına ayrı ayrı bağlanmak üzere toplamda altı adet CAN modülü gerçekleştirilmiştir. Her bir CAN modülünde bir tane MCP2515 CAN-SPI ve bir tane MCP2551 CAN-Bus çevirici kullanılmıştır. Böylelikle devrede bulunan altı modülün CAN-Bus üzerinden haberleşerek dağıtık kontrol donanımı oluşturulmuştur. Sonuçta model asansörün CAN-Bus teknolojisi kullanılarak dağıtık kontrolü başarılı bir şekilde gerçekleştirilmiştir. Böylelikle CAN-Bus teknolojisinin dağıtık kontrol uygulamaları için kullanışlı bir teknoloji olduğu gözlemlenmiştir. (Dinçer, 2010)

Bir EV'nin tüm itme sistemi bileşenleri, dinamik model hesaplamalarından prototip aşamasına adım adım geliştirildi. Özellikle, kuvvetlerin araç üzerindeki etkisi ve kuvvet dağılımı incelenmiştir. Ayrıca gerekli çekiş gücü, motor gücü, motor torku ve şaft hızı değerleri elde edildi. FEA yöntemi ile bilgisayar ortamında uygun bir BLDC motoru tasarlandı ve tasarımı etkileyen faktörler incelendi. Ara sıra su soğutmasıyla motor



büyüklüğü ve kütlesi küçültülerek iyileştirme sağlandı. Tasarımın karakteristik eğrisi bir dinamometre yükleme sistemi ile incelendi. Sürüş mesafesini arttırmak için etkili rejeneratif frenleme uygulandı. Rejeneratif frenleme için, herhangi bir ek devreye gerek kalmadan motor sürücüsü güç kademesinden yararlandı. Araç için optimizasyon çalışmaları yapıldı ve ileri kontrol yazılımı testleri devam ediyor. (Üstün, ve diğerleri, 2017)

Elektrikli taşıtlarda ultrakapasitör ve bataryaların hibrit kullanımı bir enerji yönetim stratejisi ile prototip bir elektrikli araç üzerinde denenmiş ve buna ilişkin deneysel sonuçlar değerlendirilmiştir. Tasarlanan hibrit enerji depolama sisteminde bataryalar temel güç talebini karşılamakta kullanılırken, dönüştürücü üzerinden sisteme bağlı olan ultrakapasitör sisteminden ise aşırı güç taleplerinin karşılanması amacıyla yararlanılmıştır. Tasarlanan dönüştürücü testlerde prototip aracın güç ihtiyacını ultrakapasitör ile tek başına sağlayabilmiştir. Ayrıca dönüştürücü hibrit sistem testlerinde oldukça başarılı bir performans göstermiştir. Bunu yanı sıra uygulanan kontrol yöntemi sayesinde, batarya gerilim seviyesi istenen değerler arasında tutulabilmektedir. Bu şekilde, bataryadan aşırı akımların çekilmesi dolaylı olarak engellenebilmektedir. Sonuç olarak batarya ömrü ve verimliliği, buna bağlı olarak da genel sistem performansı önemli bir oranda arttırılabilmektedir.

Test sonuçlarına bakıldığında sistemin beklendiği gibi çalıştığı açıkça gözükmektedir. Saatte 55 km/h hıza ulaşan prototip araç maksimum 6kW'a kadar güçler talep etmektedir. Buna karşılık bataryadan çekilen akımlar sadece batarya ile çalışma sırasında çekilen akımlarla karşılaştırıldığında azalmıştır. Batarya gerilimi de incelendiğinde hibrit sistemde batarya (bara) geriliminin küçük sapmalar haricinde istenen aralıkta tutulduğu gözlemlenmektedir. (Uğur, 2011)

Gerçek zamanlı bir ECU'nun geliştirilmesi ve geçerliliği bu çalışmada gerçekleştirilmiştir. Modüler programlama kullanımı sayesinde, çift çekirdekli işlemcide olduğu gibi FPGA'nın paralelliğinin sağladığı tüm kaynaklardan yararlandı. Bu nedenle, sistemin getirdiği son tarihler, karmaşık bir gerçek zamanlı işletim sistemine ihtiyaç duyulmadan karşılandı. Dahası, Simulink® gibi simülasyon platformlarından complex

kontrol algoritmalarının uygulanmasına izin veren çok hızlı bir cevap veren kontrol düşük seviyeli bir dille gerçekleştirildi. Bu ECU'nun aşağıdaki gibi diğer aktif güvenlik algoritmalarının yanı sıra TCS'nin geliştirilmesinde de kullanılabileceği gösterilmiştir: kilitleme önleyici fren sistemleri (ABS), elektronik denge kontrolü (ESC) ve çarpışma önleme sistemleri. Modüler programlama Zynq-7000 gibi bir SoC tarafından sunulan veri kaynaklarından yararlanır. İşlemci çekirdeğinden biri kontrol algoritması için kullanıldı, diğeri ise yönlendirilmiş veri kaydı yaptı. FPGA, her bir kontrolör için ihtiyaç duyulan sensörlerin hem okuma hem de ön izlemesini hem HDL dilini hem de paralel olarak yumuşak işlemcileri kullanarak yaptı. Ek olarak, IMU ve GPS, bu belgede geliştirilen ECU'nun performansını artırarak, kontrol algoritmasına entegre edilebilecek olan ve aracın durumu hakkında bilgi sağlayan veri birleştirme tekniklerine izin verir. (Fernandez, Vargas, Garcia, Carrillo, & Aguilar, 2019)

Elektrikli Araçlar için sistem mimarisi ve yazılım tasarımı zorluklarına genel bir bakış sunulmaktadır. İlk olarak, pil, elektrik motoru ve elektrik güç aktarımı dikkate alındığında elektrikli araca özgü bileşenleri ve bunların kontrolleri tanıtılmaktadır. Ayrıca, telsiz ve bilgi sistemleri gibi elektrikli araçların güvenliğini ve enerji verimliliğini artırmaya yardımcı olacak teknolojiler tartışılmaktadır. Sonuç olarak, yazılım tasarımı, temel donanım mimarisini tamamen soyutlarken kontrol edilen bileşeni göz önünde bulundurarak, bütünsel bir yapıda yapılmalıdır. Bu amaçla, ara katman yazılım ve doğrulama teknolojilerinin tasarım ve test karmaşıklığının azaltılmasına nasıl yardımcı olabileceği gösterilmektedir. (Lukasiewicz, ve diğerleri, 2013)

## **1.2. Elektrikli Araçlar**

Dünyada petrol ürünlerinin kullanımının aşırı yükseldiği bu dönemde fosil yakıtların çevreye verdiği zarar, petrol rezervlerinin kısıtlı olması ve giderek azalması sebebiyle alternatif enerji arayışları sürmektedir. Güneş enerjisi, biyo enerji ve rüzgar enerjisi gibi yenilenebilir enerji kaynakları 'Temiz Enerji' olarak adlandırılıyor ve petrolün yerini alacağı öngörülüyor. Temiz enerji günümüzde çok önemli bir ihtiyaç haline gelmiştir, çoğu devlet planlamalarına yenilenebilir enerji kaynakları geliştirme planlarını ve

bunlarla ilgili olan elektrikli araçlar, batarya sistemleri ve motor sistemlerini dahil etmiştir.

Elektrikli araçlar yalnızca elektrikle beslenir. Dolayısıyla elektrikli araçlarda tahrik sistem kaynağı olarak elektrikli motorlar, uzun süre kullanılacak enerji kaynağı olarak fosil yakıtlar yerine farklı yöntemlerle elde edilebilen enerjiyi depolama özelliğine sahip bataryalar, geçici güç değişimlerini dengelemek için kapasitörler bulunmaktadır. Bataryalarda depolanacak enerji farklı yöntemlerle elde edilebilir. Bu kaynak, güneş enerjisini düşük verimle dönüştüren fotovoltaik hücreler olabileceği gibi daha yüksek verime sahip olan ancak belirli şartlar oluşmadan kullanılması zor olan rüzgar enerjisi türbinleri de olabilir. Yenilenebilir enerji kaynaklarından olan güneş ve rüzgar enerjisinin elektrikli araçlarda uygulamaları ile ilgili çalışmalar süregelmeye başlamış olmasına rağmen bu enerjilerin harekete dönüştürülmesi işleminin mobil platformlarda önemli bir kriter olan yüksek verim ve küçük hacim ihtiyacını mevcut yöntemlerle karşılayamadığı görülmüştür. Bu alanlardaki teknolojik geliştirmeler devam etmektedir. Kaynak olarak şebeke elektriğinin kullanılması yönündeki çalışmalarla birlikte elektrikli araçların yaygınlaşması hız kazanmıştır. Geleneksel araçların fiziksel özelliklerini fazla değiştirmeden araç içinde yapılabilecek modifikasyonlarla prizli elektrikli araçların elde edilmesinin mümkün olabilmesi bu yaygınlaşma da önemli rol oynamıştır. Elektrikli araçlarda fiyat verimliliği ise doğrudan elektrik sağlayıcıların fiyatlandırması ile doğrudan ilişkilidir. Çevresel etkiler bakımından içten yanmalı motorlarda kullanılan yakıtlar olmadığından emisyon oranının kontrol altına alınması yönünde elektrikli araç kullanımının büyük fayda sağlayacağı yadsınamaz bir gerçektir. (Chakraborty & Simões, 2013) Elektrikli araçlarda, frenleme sırasında içten yanmalı motorlarda kaybolan enerjinin önemli bir kısmı rejeneratif etki olarak elektrik enerjisine dönüştürülür. Bu sayede enerji verimliliği ve araç menzilin arttırılmasına imkan sağlanmış olur.

### **1.3. Çalışmanın Özgünlüğü**

Tübitak Efficiency Challenge yarışmasına katılacak olan SUBU – TETRA elektrikli aracı PLC’li bir sistem kullanılarak çalışması sağlanmaktadır. Bu araştırmada PLC yerine kullanılacak AKS tasarlanarak testleri yapılmıştır. Test aşamasında STM32F4

Discovery kartı kullanılmıştır. Oluşturulan AKS sisteminde kablolama ve sensör sinyallerinin gürültüden daha az etkilenmesi amacı ile araca giriş çıkış birimi olarak isimlendirilen birim eklenmiştir.

AKS ve giriş çıkış birimi birbirleri ile CANBUS hattı üzerinden haberleşmektedir. Aracın kontrolüne ilişkin hesaplamalar, çevre birimlerden veri alma, bu birimlerin kullanabilecekleri verileri CANBUS hattına gönderme, kalibrasyon bilgileri ve arayüz ile haberleşme kısmı AKS'nin görevidir. Giriş çıkış kartı aracın ön kısmında bulunmaktadır. Ön tarafta bulunan sensör kartının tekerlek devir, motor sıcaklık, gaz pedalı, fren pedalı, el freni, direksiyon açısı, tekerlek açısı gibi sensörlerden gelen bilgiler CAN hattı üzerinden AKS'ye göndermektedir.

CAN hattındaki bilgilerin kayıtlarının tutulması için AKS seri port üzerinden bilgisayar arayüzüne verileri göndermektedir. AKS sistemi diğer birimlerden ve kendi girişlerinden aldığı verileri değerlendirerek motor hızı, sıcaklık hesapları ve güç tüketimi gibi gerekli verilerin hesaplanması sağlanmaktadır.

Bilgisayar arayüzü sayesinde sürüş verilerinin alınması, kayıt edilmesi, analizi ve kalibrasyon işlemleri yapılabilmektedir.

Aracın dönüşlerde sağ ve sol tekerlek dönüş hızlarının farklı olduğu gözlemlenmiştir. Bu problemin çözümü olarak direksiyon açısı, tekerlek açısı ve hız değerleri göz önünde bulundurularak dönüş yapılan tarafta bulunan tekerleğe yavaşlatma katsayısı eklenmiştir.

Araç kontrol sistemi tasarımında kullanılan mikrodenetleyicinin özelliklerinden biri olan DMA kullanılmıştır. DMA frekans okuma ve analog giriş okuma kütüphanelerinde kesmelerle algoritma belirlenerek kullanılmıştır. DMA kullanımı 'DMA half complete' ve 'DMA full complete' kesmeleri ile kullanılmıştır. Sonuç her kesmeden sonra okunan değer ile bir önceki kesme sonucunun ortalaması alınarak hesaplanır. Bu sayede basit bir alçak geçiren filtre gibi davranarak analog okunan değerlerdeki gürültü etkisi azalması sağlanmıştır.

## **BÖLÜM 2. GENEL BİLGİLER**

### **2.1. Araç Kontrol Sistemi**

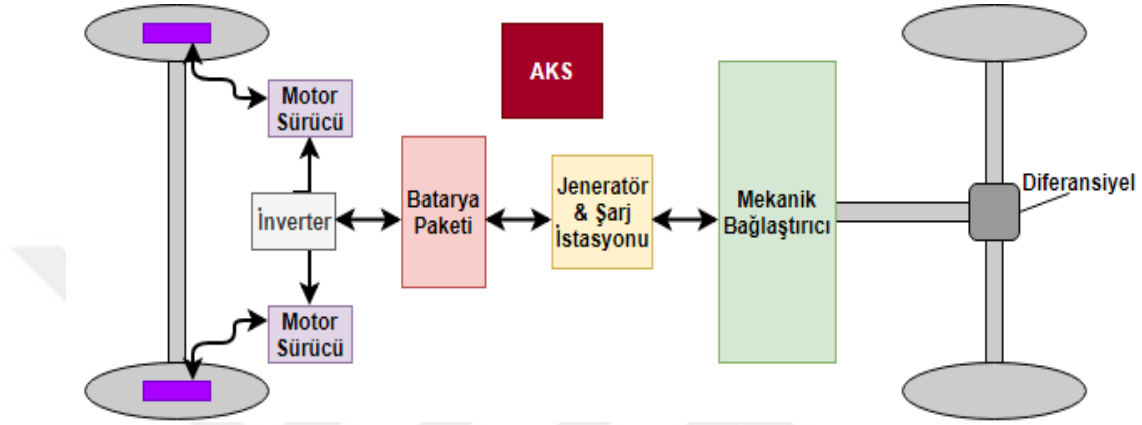
Araç kontrol sistemi araç üzerinde bulunun farklı sistemlerden ve çeşitli sensörlerden aldığı bilgileri işleyen, işlediği bilgilere göre ilgili donanımları kontrol eden ve bu bilgileri diğer cihazlar ile paylaşan ünedir. Motor kontrol sistemi, batarya yönetim sistemi, DC-DC dönüştürücü, kullanıcı ekranı, havalandırma sistemi, batarya yönetim sistemi ve gövde yönetim sistemi gibi sistemler araç kontrol sisteminin haberleştiği başlıca sistemlerdir. Araç kontrol sistemine bağlı başlıca sensörler devir sensörü, açma sensörü, buton ve anahtarlardır. Aracın topolojisi Şekil 2.1 de gösterilmektedir.

Araç kontrol sistemi yazılım ve donanım kısımlarından oluşur. Donanım kısmı mikrodenetleyici, besleme devresi, ADC giriş koruma devresi ve filtreleri, PWM çıkış koruma devresi, haberleşme alıcı-verici entegreleri gibi bölümlerden oluşur. Yazılımda ise verilerin toplanması, işlenmesi ve haberleşme gibi görevler icra edilir.

Araç kontrol sisteminde aşağıdaki fonksiyonların bulunması beklenir;

- Motor hız kontrol: Aracın hızının lineer olarak kontrolü ve ani hızlama ve yavaşlamalarda aracın sarsılmadan çalışmasının sağlanmasıdır.
- Araç enerji yönetim sistemi: Araçta fazla enerji kullanımı olduğunda enerji kullanımının sınırlandırılmasının sağlanmasıdır.
- Araç içi haberleşme sisteminin yönetimi: Araçta bulunan ünitelerin AKS ile haberleşmesi ve AKS tarafından işlenmiş verilerin üniteler ile paylaşılmasının sağlanmasıdır.
- Arıza teşhisi: Araçta bulunan arızaların sensörlerden gelen veriler ve yazılım yardımı ile belirlenmesi ve AKS nin uyarı verilmesi sağlanmasıdır.

- Araç durumunun izlenmesi: Araçta sürüş esnasında, öncesi veya sonrasında sürücü tarafından ihtiyacı olan bilgilerin gösterilmesinin sağlanmasıdır.
- Araç bilgilerinin toplanması ve uzak sisteme iletimi: Araçta toplanan ve işlenen verilerin uzakta bulunan bir birime kablosuz bir şekilde iletilmesinin sağlanmasıdır.



Şekil 2.1. Elektrikli araç topolojisi.

## 2.2. ARM İşlemci Mimarisi

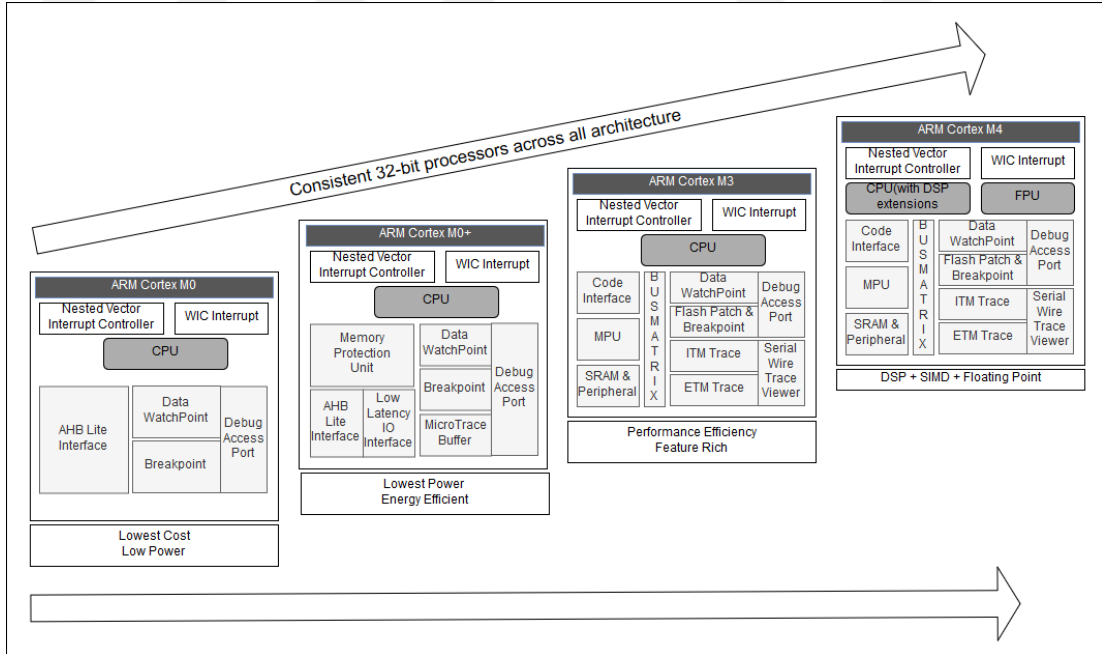
Acorn Computers Ltd. Tarafından ARM (Acorn RISC Machine) mimarisi adıyla geliştirilmeye başlanmıştır. 1985 yılında ilk defa piyasaya sürülmüş ve 1990 yılında Firma ismi Advanced RISC Machine Ltd. olarak değişmiştir. ARM Ltd. olan günümüzdeki ismini 1998 yılında almıştır. ARM firması bir yarıiletken firması değildir. 32 bitlik işlemci mimarileri üreterek bunları ST, TI, Atmel, Samsung gibi yarıiletken üreten firmalara satar.

32 bitlik işlemci olan ARM 8 bitlik işlemcilere göre daha hızlı işlem yapabilme yeteneğine sahiptir. Yüksek performansı ve düşük güç tüketimi ile günümüzde en çok kullanılan işlemci mimarisidir.

ARM firması işlemci mimarileri 3 farklı gruba ayırır:

- Klasik ARM Mimarisi: Cortex serisinden önceki işlemci mimarileri bu gruptadır. ARM7 serisi sinyal işleme ve motor kontrol uygulamalarında ağırlıklı olarak kullanılmaktadır. ARM9 ve ARM11 mimarileri daha çok mobil cihazlarda kullanılmaktadır.
- Cortex M Serisi İşlemci Mimarisi: Uygun fiyat ve yüksek performans çözümleri sunan mimari grubudur. Endüstriyel uygulamalar, IOT uygulamaları, bireysel uygulamalar gibi pek çok yerde kullanılır.
- Cortex A Serisi İşlemci Mimarisi: Yüksek performans işlemcileridir. Tablet, telefon, oyun konsolları gibi son kullanıcı ürünlerinde kullanılır. Yüksek performans ve işlem gücü ihtiyacı olan yerlerde sıklıkla kullanılır.

ARM mimarili bir işlemci ile çalışırken yazılım geliştirmenin en pratik yolu işlemci üreticilerinin uygun fiyatlara sunduğu geliştirme kartlarının üzerinde çalışmaktır. Farklı firmalar 32 bitlik ARM mimarili işlemciler üretse dahi işlemci ile çalışmadan önce bu işlemcinin mimarisini bilmek yazılım geliştirmeye büyük avantaj sağlar. Şekil 2.2’de Cortex-M işlemcilerin mimarilerini görebilirsiniz.



Şekil 2.2. ARM Cortex-M işlemci mimarileri.

## **2.3. Haberleşme Protokolleri**

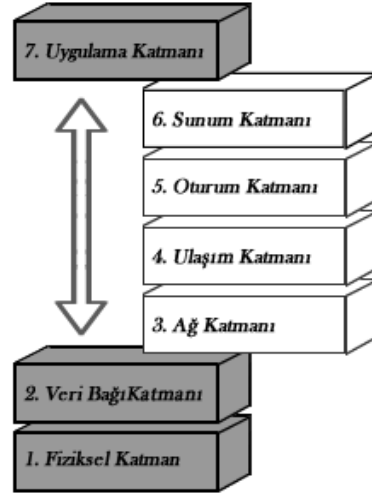
Sistemde donanımların kendi aralarında ve bilgisayar ile haberleşmesi için bazı haberleşme protokollerinin kullanılması gerekmektedir. Bu sistem için tercih ettiğimiz haberleşme yöntemleri CANBUS haberleşmesi ve UART Haberleşmesidir.

### **2.3.1. CANBUS haberleşme**

CAN bus, Robert BOSCH tarafından bulunan saniyede 1Mbit hıza ulaşabilen haberleşme sistemidir. USB ya da Ethernet gibi geleneksel ağların aksine CAN noktadan noktaya büyük boyutlarda veri blokları göndermez. Bir CAN ağında, sıcaklık ya da devir gibi birçok mesaj CAN hattında yayınlanır, bu bilgiler hattaki her bir düğüm tarafından erişilebilir olmaktadır.

CAN otomotiv endüstrisinin karmaşık kablolama sorununu tek bir haberleşme hattı ile çözmek amacıyla geliştirilmiş, ISO(International Standardization Organization) tarafından seri bir haberleşme ağı olarak tanımlanır. Elektriksel gürültü ve bozulmalara karşı yüksek dayanıklılığı, kendi hata kontrol yapısı ve hatalı verileri düzeltmesi CAN'ın otomasyon, medikal ve üretim gibi endüstrilerde kullanımının artmasında önemli bir rol almıştır. ISO -11898 standardında CAN haberleşmesinde bilginin ağ üzerinden diğer cihazlara nasıl gönderilmesi gerektiği OSI modelinde uyumlu katmanlarla tanımlanmıştır. Şekil 2.3 ile OSI katman yapısı gösterilmiştir. Fiziksel olarak bağlanmış cihazlar arasında haberleşme modelin fiziksel katmanında tanımlıdır. ISO 11898 Veri Bağı Katmanı ve Fiziksel Katman olmak üzere ISO/OSI modelinin en alttaki iki katmanını oluşturur.





Şekil 2.3. ISO standart mimarisi.

Şekil 2.3’de görülen Uygulama Katmanı CANOpen gibi haberleşmeyi üst seviye uygulamaya bağlayan özel protokolü belirler. Bu protokol uluslararası kullanıcılar ve üreticiler tarafından destekleniyor. Kvaser firmasının ‘KVASER CAN’ ve Rockwell otomasyonun ‘DeviceNet’ protokolü diğer CAN tabanlı endüstri standartlarıdır. Aynı şekilde motor uygulamalarında J1939 protokolü standar haline gelmiştir.

### 2.3.1.1. Standart ve genişletilmiş mesaj çerçevesi

CAN haberleşme protokolünde çakışma tespiti ile çoklu erişimde hat kontrolü ve mesaj önceliğine göre çakışma önleyici (CSMA/CD+AMP-Carrier Sense Multiple Access / Collision Detection + Arbitration on Message Priority) özelliğine sahiptir. Çoklu erişimde hat kontrolü (CSMA) hatta bulunan bütün cihazlar hatta mesaj göndermeden önce belirlenen periyot kadar beklemesi ile çözülür. Çakışma tespiti ve mesaj önceliğine göre çakışma önleyici, önceden öncelik programlı mesaj çerçevesindeki bitsel çarpışma önlemeye dayalıdır. Yüksek öncelikli mesaj tanımlayıcı her zaman hatta öncelikli erişim hakkını kazanır. Hattaki tüm düğümler bütün bitlerini yazdıktan sonra hatta 1 değerini gönderir ve hata çözümleyici düğüm bunu algılar.

ISO-11898:2003 standardı, 11 bitlik tanımlayıcı(standart frame) ile haberleşme hızı 125Kbit/s ile 1Mbit/s arasındadır. Bu standart daha sonraları geliştirilerek 29 bitlik tanımlayıcı(extended frame) halini aldı. Standart mesaj çerçevesi ile 2048 ( $2^{11}$ ) adet mesaj

tanımlayıcısı kullanılabilirken genişletilmiş mesaj çerçevesi 537milyon ( $2^{29}$ ) adet mesaj tanımlayıcısını destekler.

S		R	I		D				C	A	E	I
O	Tanımlayıcı(11-bit)	T	D	r0	L			8 Byte Veri	R	C	O	F
F		R	E		C				C	K	F	S

Şekil 2.4. Standart mesaj çerçevesi.

**SOF** : Mesaj çerçevesi başlangıç bitidir. Mesajın başladığını ve hatta bağlı düğümlerin senkronizasyonunu sağlar.

**Tanımlayıcı** : Standart CAN 11 bit tanımlayıcı mesajın önceliğini belirtir. Düşük değerli mesaj önceliği daha yüksektir.

**RTR** : Uzak iletim isteği bitidir. Başka bir düğümden bilgi almak istendiğinde aktif hale gelir.

**IDE** : Tanımlayıcı türünün Standart ya da Gelişmiş olduğunu belirtir.

**r0** : Rezerve bit.

**DLC** : Veri uzunluk bilgisi iletilmek istenilen datanın boyutunu barındırır.

**Data** : İletilmek istenilen veriyi 64 bite kadar barındırır.

**CRC** : Hata algılama amacıyla hesaplanan değeri bulundurur.

**ACK** : Mesajı doğru alan her cihaz bu tekrarlayan biti arttırarak orjinal mesajın içerisine yazar.

**EOF** : Mesaj çerçeve sonu. 7 bittir.

**IFS** : Sonraki mesaj frame inin buffer alanına doğru bir şekilde alınabilmesi için gereken zaman. 3 bitlik çerçeveler arası boşluk.

S		S	I		R			D		C	A	E	I
O	Tanımlayıcı	R	D	Tanımlayıcı	T	r0	r1	L	8 Byte	R	C	O	F
F	(11-bit)	R	E	(18-bit)	R			C	Veri	C	K	F	S

Şekil 2.5. Genişletilmiş mesaj çerçevesi.

**SRR** : Gelişmiş Frame de uzak iletim isteği yeni kullanılan vekil iletim isteği biti.

**IDE** : 1 bit tanımlayıcının arkasından 18 bit tanımlayıcı bitinin geldiğini belirtir.

**r1** : RTR ve r0 bitlerinden sonra gelir. Rezerve olarak ayrılmıştır.

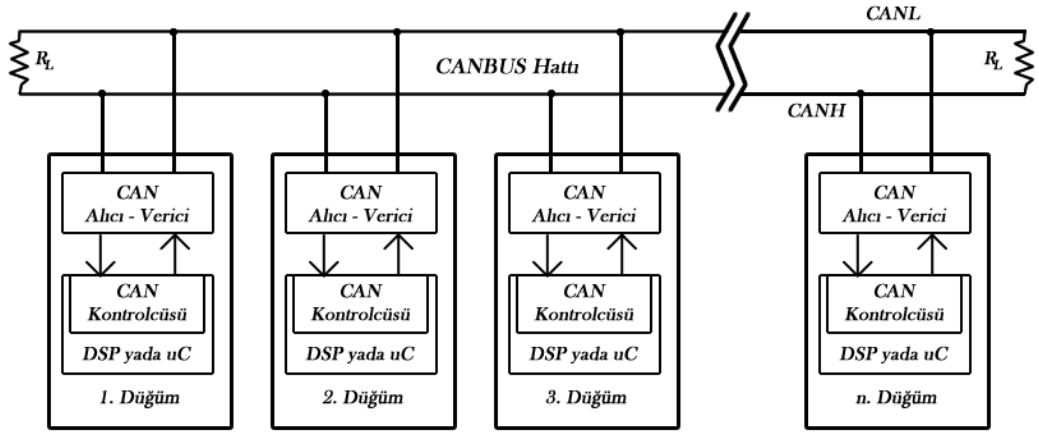
### 2.3.1.2. CANBUS mesaj çerçeve yapısı

Veri mesaj çerçevesi, uzak mesaj çerçevesi, hata çerçevesi ve aşırı yük çerçevesi olmak üzere CAN hattı mesaj tip veya çerçeveleri 4 adettir.

- Veri Mesaj Çerçevesi: Veri çerçevesi en genel kullanılan mesaj tipidir ve Çakışma önleme alanı, Veri alanı, Paket Doğrulama Kodu(CRC) ve Doğrulama(ACK) alanının bir bölümünü oluşturur. Çakışma önleme alanı 11 veya 29 bit tanımlayıcıyı ve RTR bitini içerir. Veri alanı 8 byte lık datayı içerir. Paket doğrulama kodu 16 bitlik checksum değeri hata tespiti için kullanılır.
- Uzak Mesaj Çerçevesi: Uzak çerçevenin amacı, başka bir düğümden veri iletimini talep etmektir. Uzak çerçeve, iki önemli farkla birlikte veri çerçevesine benzer. İlk olarak, bu tür bir mesaj açıkça çakışma önleme alanında RTR bit tarafından uzak bir çerçeve olarak işaretlenmiş ve ikincisi veri alanı yoktur.
- Hata Çerçevesi: Hata çerçevesi, bir CAN mesajının biçimlendirme kurallarını ihlal eden özel bir mesajdır. Bir düğüm bir mesajda bir hata tespit ettiğinde gönderilir ve hatta bulunan bütün düğümlerin hata çerçevesi yollamasına sebep olur. Orijinal verici daha sonra mesajı otomatik olarak yeniden iletir.
- Aşırı Yük Çerçevesi: Hata çerçevesine benzer. Mesajlar arasında ekstra gecikme sağlamak için kullanılır.

### 2.3.1.3. CANBUS hattı

Şekildeki gösterimde veri bağı bağlantısı ve fiziksel katmanlar görülmektedir. Dahili CAN kontrolcüsüne sahip herhangi bir denetleyicide veri bağı katmanı mevcuttur. Fiziksel ortama bağlanmak elektriksel dönüşümlerin yapılabileceği bir alıcı-verici entegresi üzerinden yapılır.



Şekil 2.6. CAN hattı gösterimi.

Sinyal çözümü CAN hattının H ve L arasında farksal olarak yapılır. Dengelenmiş farksal sinyal burgulu çift kablo üzerinden gelen sinyali azaltır ve bu sayede yüksek haberleşme hızlarına çıkabilir. Bu sayede yüksek hızlarda dahi CAN haberleşmesi gürültü bağışıklığını sağlamış olur. Dengelenmiş ile kast edilen her sinyal kablosundan geçen akım birbirine eşit fakat ters yöndedir. Farksal dengeli alıcı ve burgulu çift kablo kullanımı bir CAN hattının gürültü bağışıklığını yükseltir.

ISO 11898 Standardında belirtilenlere göre maksimum sinyal hızı 1Mbit/s olabilmesi için 30 düğümlü ve 40m olabilir. Kablo empedansı blendajlı veya blendajsız 120Ω olacak şekilde belirtilmiştir. CAN hattında hat uzunluğuna göre kullanılacak maksimum veri iletim hızları Tablo 2.1Tablo 2.1. Hat uzunluğuna bağlı ile gösterilmektedir.

Tablo 2.1. Hat uzunluğuna bağlı maksimum veri iletim hızı.

Hat Uzunluğu	Maksimum Veri İletim Hızı
30 m	1 Mbit/s
50 m	800 kbit/s
100 m	500 kbit/s
250 m	250 kbit/s
500 m	125 kbit/s
1000 m	50 kbit/s

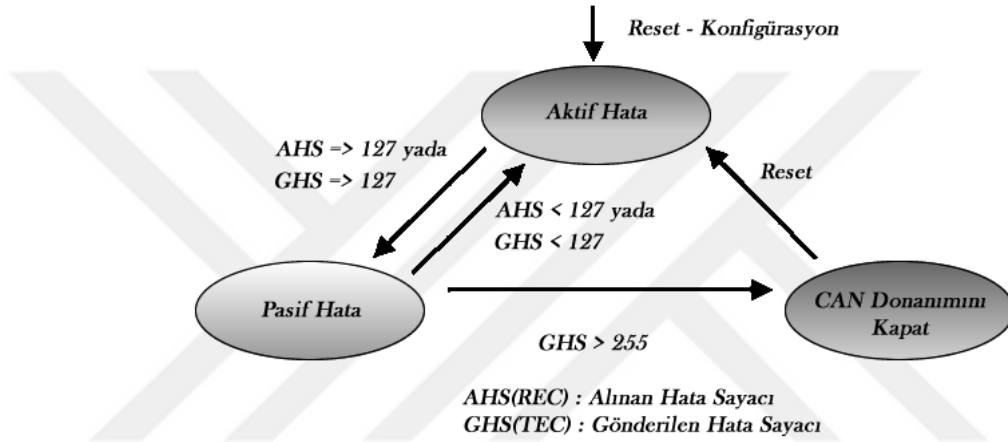
#### 2.3.1.4. CANBUS hata önleme

CAN protokolünde çerçeve format kontrolü, CRC kontrolü ve ACK(onay) kontrolü olmak üzere 3 adet çerçeve, bit kontrolü ve bit stuffing olmak üzere 2 adet bit düzeyinde hata kontrolü vardır. Mesajda hata olduğu durumda gönderici düğüm mesajı tekrar gönderir.

CAN çerçeve format kontrolü bir çerçeve alındıktan sonra çerçeve yapısına uyup uymadığını karşılaştırır. Alınan veride hata olması durumunda veri reddedilerek hata çerçevesi gönderilir. Bu sayede sistemin doğru formatta veri alması sağlanmış olur. CRC hata önleme yönteminde başlangıç bitinden CRC bitine kadar olan bitler belirli işlemlerden geçerek CRC kodu oluşturulur. Alıcı kısım çerçeveyi aldıktan sonra gelen çerçeveye göre oluşturduğu CRC kodu ile kendisine gelen CRC kodunun aynı olup olmadığını kontrol eder. Bu sayede formata uygun ancak hatalı olan mesajların önüne geçilmiş olur. CRC hatası farklı düğümlerden aynı ID ile mesaj gönderilmesi durumunda oluşur. Çerçeve düzeyinde bulunan son hata kontrolü olan ACK hata kontrolü, ACK biti alıcının mesajı aldığını onaylaması anlamına gelir. Mesaj gönderen düğüm CRC bitlerini gönderdikten sonra ACK bitini çekinik olarak gönderir ve en az bir tane alıcının ACK bitini baskın hale çekmesi beklenir. ACK biti belirli bir süre boyunca cevaplanmazsa hatta mesajın alınacağı bir düğüm olmadığı anlaşılır. Gönderici düğüm ACK onayını alana kadar aynı mesajı göndermeye devam eder.

CAN haberleşmede senkronizasyon saat darbeleri ile değil lojik değişimler ile sağlanır. Aynı lojik seviyenin 5'den fazla arka arkaya gelmesi durumunda senkronizasyonda problem olduğu anlamına gelir ve alıcıda hata oluşur. Bunun engellemesi için 5. seviyeden sonra 1 adet ters seviyeden bit gönderilerek işlem devam eder. Bu yöntem bit stuffing yöntemi denir. Bir düğüm hata mesajı göndermek istediğinde arka arkaya 6 adet baskın bit yazarak hata oluşturur. CAN haberleşmesinde gönderilen veri aynı zamanda okunur. Eğer bir düğüm veriyoluna veri yazarken gönderdiği seviyeden farklı seviye okuyorsa bit hatası meydana gelir.

CAN donanımında hata durumlarında kullanılmak üzere 2 adet hata sayacı vardır. Bu sayaçlar ile göndericide oluşan veya alınan hatalar sayılır. Bu sayıcılardan biri 127 veya daha büyük bir sayıya ulaşınca donanım 'pasif hata' moduna geçer ve gelen hata çerçevelerini cevaplar. Hata oluştuğu zaman baskın bitler yerine çekinik bitler gönderir. Gönderim Hata Sayacı 255 değerini geçerse donanım kendisini kapatır ve hattaki iletişimden etkilenmez. Bunun yapılmasının sebebi düğümün hataya düşerek hatta sürekli hatalı mesaj göndermesini engellemektir. Düğümün haberleşmeye devam edebilmesi için yeniden başlatılması veya tekrar konfigüre edilmesi gerekmektedir.



Şekil 2.7. Hata sayaçları çalışma şeması.

CAN çok sayıda küçük boyutlu mesajlar gönderme ve yüksek doğruluk ihtiyacı olan uygulamalar için uygundur. CAN adres temelli değil mesaj temelli olduğundan bir bilginin birden fazla konuma gitmesi istenen sistemler için de uygundur.

Hata sınırlama da CAN'in önemli avantajlarından biridir. CAN Hattına hatalı mesaj gönderen düğümün hatta mesaj göndermesi kesilir, bu sayede band genişliği her zaman kritik bir mesaj iletimi için uygun durumda kalmış olur. (Karaca, 2003)

### 2.3.2. UART haberleşmesi

UART haberleşmesinde iki adet cihaz birbiri ile direk haberleşir. UART haberleşmede aktarılmak istenen ver seri forma dönüştürülerek iletilir ve alıcı taraf alınan veriyi işlenebilecek forma geri çevirir. İki donanım arasında veri iletimini bu yöntemle

sağlamak için Tx ve Rx olmak üzere iki adet kabloya ihtiyaç vardır. Tx hattı veri göndermek ve Rx hattı veri almak için kullanılır. Bu haberleşmede senkronizasyon sinyali kullanılmadığı için asenkron bir haberleşmedir. Senkronizasyon sinyali yerine iletilmek istenen verinin başına ve sonuna başlangıç ve bitiş bitleri eklenir. Bu bitler sayesinde hatta veri gönderiminin ne zaman başladığını ve bittiğini belirtir.

Rx pininde başlangıç biti algılandığında, bilinen veri iletim hızında gelen bitleri okumaya başlar. Veri iletim hızı saniyedeki iletilen bit cinsinden verilir. Haberleşmenin gerçekleşebilmesi için veri iletim hızlarının iki cihaz için de aynı ayarlanması gerekmektedir. Aynı zamanda iki cihazın paket yapılarının da aynı şekilde konfigüre edilmesi gerekmektedir.

Başlangıç biti (1 bit)	Veri bitleri (5-9 bit)	Parite biti (0 ya da 1 bit)	Bitiş biti (1 yada 2 bit)
---------------------------	---------------------------	--------------------------------	------------------------------

Şekil 2.8. Seri haberleşme paket yapısı.

**Başlangıç biti:** UART veri iletiminde veri iletimi olmadığında hat voltaj seviyesi yüksek seviyededir. Veri iletimini başlatmak için hattın seviyesi bir saykıl boyunca düşük seviyeye çekilir. Yüksek seviyeden düşük seviyeye geçişi algılandığında veri iletim hızına göre gelen veriler okunmaya başlar.

**Veri çerçevesi:** Veri çerçevesi iletilmek istenen gerçek veriyi barındırır. Eğer parite biti kullanılmışsa 5 ile 8 bit arasında olabilir. Parite biti kullanılmadıysa veri çerçevesi 9 bit olabilir. Çoğu durumda veri gönderimi öncelikle düşük işaretli bitten başlar.

**Parite:** Parite teklik veya çiftliğini belirtir. Parite biti alınan verinin iletim esnasında değişip değişmediğini anlamak için bir yöntemdir. Elektromanyetik gürültü, uyumsuz veri iletim hızları ya da uzun mesafe veri iletimleri gibi sebeplerle bitler iletim esnasında değişebilmektedir. Veri alımı tamamlandıktan sonra değeri 1 olan bitleri toplayarak toplamın tek mi çift mi olduğu kontrol edilir. Eğer parite biti 0 ise toplanan bitlerin toplamı çift olmalıdır. Eğer parite biti 1 ise toplanan bitlerin toplamı tek olmalıdır. Parite biti bu değerler ile eşleşiyorsa o zaman veri aktarımı hatasız tamamlanmış demektir.

Parite bitinin bu deęerler ile eřleşmemesi durumunda veri çerçevesinde bulunan verinin deęiřtięi anlaşılır.

Bitiř Bitleri: Veri paketinin bittięini belirtmek için kullanılan bittir. Konfigürasyonda belirlenen bitiř biti sayısı süresince hattı düşük seviyeden yüksek seviyeye çeker.





## **BÖLÜM 3. MATERYAL VE YÖNTEM**

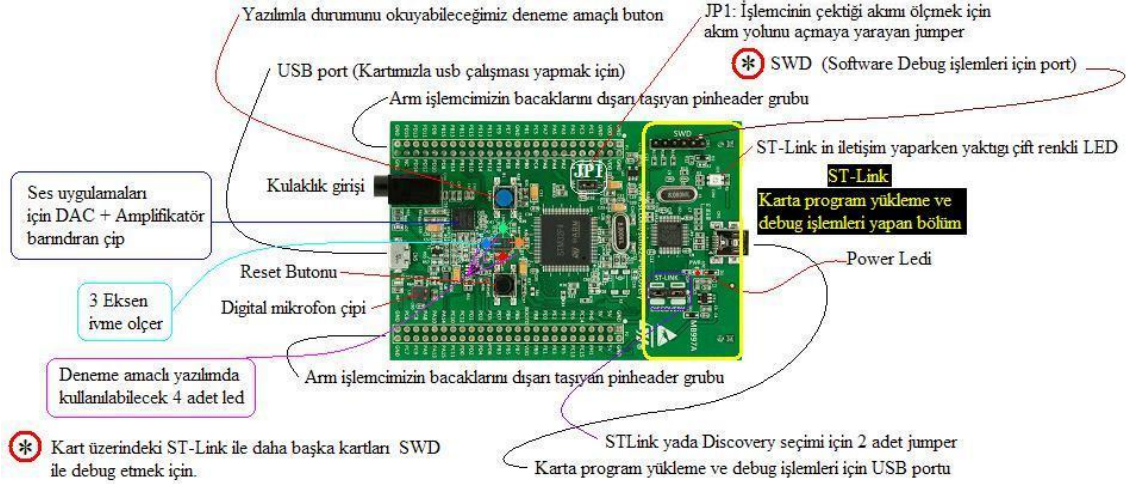
### **3.1. Materyal**

#### **3.1.1. STM32F407 Discovery geliştirme kartı**

Günümüzde ARM mimarisine olan yönelimle birlikte yüksek performanslı işlemciler daha uygun fiyatlara elde edilebilir hale geldi. Bu da teknolojik gelişmelerin daha hızlı ilerlemesini sağladı.

Kendi programlayıcısı ve giriş-çıkışların pinler halinde kitin üzerinde olması test ve hata ayıklama işlemlerinde kullanıcıya kolaylıklar sağlıyor. Bu satede hızlı prototiplemenin önü açılmış bulunmaktadır. Günümüzde birçok firma, üniversite, yeni başlangıç şirketleri ve kişisel hobiciler bu kitleri kullanmaktadır.

STM32F4 Discovery kiti üzerinde ST firmasının amiral gemisi olan STM32F407 işlemcisini bulundurur. Cortex-M4 mimarisi bulunan işlemcinin en önemli özelliklerinden birisi dahili kayar nokta birimi bulunmasıdır. Giriş çıkışlar 2 adet 2x50 erkek pin soketi üzerinden olmaktadır. Kartın üzerinde 4 adet led bulunmaktadır. Dahili MEMS mikrofonu ve kulaklık çıkışı ile ses uygulamaları yapma kolaylığı sağlar. Bu serideki kitlerin amacı hızlı ve ucuz prototipleme olduğundan kartlar düşük maliyetli üretilir. Şekil 3.1 de STM32F4-Discovery geliştirme kartının bölümleri gösterilmektedir.



Şekil 3.1. STM32F4-Discovery kitinin bölümleri.

Bu ürünlerin talep görmesinin bir diğer sebebi olarak dökümantasyon ve örnek sayısının çok olması. STM32F4 ürünü çok fazla kullanıldığı için bir projeye başlamadan önce ilgili örneklerin bulunabilmesi kolay hale geliyor. Aynı zamanda geliştirme sürecinde karşılaşılan hataların araştırılıp bulunması veya firmadan direk yardım alınması da bu işlemciler ile prototipleme ve ürün geliştirme konusunda kolaylık sağlıyor. FPU içeren 32 bit Cortex M4 işlemci mimarisi, 1Mbyte'a kadar Flash Hafıza, 192+4Kbyte'a kadar SRAM, 24 Kanal 12 bit ADC, 2 x 12 bit DAC, Genel Amaçlı DMA(Doğrudan hafıza erişimi), 12 x 16 bit, 2 x 32 bit Timer, 140 adete kadar genel amaçlı Giriş/Çıkış, 3 adete kadar I2C haberleşme arayüzü, 4 adete kadar USART/UART-LIN-IrDA arayüzü, 3 adete kadar SPI haberleşme arayüzü, 2 adete kadar CAN haberleşme arayüzü, SDIO haberleşme arayüzü, USB 2.0 High/Full Speed arayüzü, 10/100 Ethernet arayüzü STM32F407 işlemcisinin genel özellikleridir.

DMA Kontrolcüsü: 2 adet genel amaçlı DMA portu bulunmaktadır. Her bir port birbirinden bağımsız 8 adet kanala sahiptir. Bu kanalların hafızadan hafızaya, hafızadan çevresel birime ya da çevresel birimden hafızaya transfer etme yeteneği vardır.

DMA kontrolcüler dairesel tampon yönetimini destekler, bunun için ekstra kod ihtiyacı yoktur. Herbir DMA kanalı atandığı donanımın DMA isteğine bağlanır. Konfigürasyon, transfer büyüklüğü, kaynak ve hedefler yazılımsal olarak yapılır. DMA donanımı SPI,

I2C, I2S, USART, Timerlar, DAC, SDIO, Kamera arayüzü ve ADC donanımları ile kullanılabilir.

Timerlar: STM32F407 2 adet gelişmiş ve 8 adet genel amaçlı, 2 adet basit ve 2 adet watchdog timer a sahiptir. Bu timerların hepsi debug modda iken durdurulabilir. Gelişmiş Timer (TIM1, TIM8): 4 tane birbirinden bağımsız kanal Input Capture, Output Compare, PWM generation ve One-Pulse Mode Output modlarında kullanılabilir. Genel Amaçlı Timer(TIMx): 110 adet senkronize edilebilir genel amaçlı timer mevcuttur. Basit Timer (TIM6, TIM7): Bu timerlar genelde DAC tetiklemesi ve çeşitli dalga formları üretmek için kullanılır. Aynı zamanda 16 bitlik genel amaçlı timer gibi kullanılabilir.

I2C Arayüzü: 3 adete kadar I2C arayüzü efendi ve köle modun da çalışabilir. Herbiri 100KHz olan normal modu ve 400KHz olan hızlı modu destekler. DMA ile bağlantısı yapılarak haberleşme DMA modülü üzerinden yapılır.

Control Alan Ağı (CAN): Donanımların ikisi de 1Mbit/s hızı ve 2.0A ve B özelliğini destekler. Standart ve gelişmiş çerçeve mesajlarını alıp gönderebilirler. Herbir CAN donanımı 3'er adet postakutusuna, 2'ser adet 3 aşamalı FIFO'ya ve SRAM de ayrılmış 256 byte hafızaya sahiptirler.

Genel Amaçlı Giriş/Çıkış(GPIO) : Herbir GPIO pini yazılımsal olarak çıkış(push-pull ya da open-drain,pull-up ya da pull-down), giriş(floating, pull-up ya da pull-down) olarak ayarlanabilir. GPIO pinlerinin çoğu alternatif fonksiyon olarak paylaşılmış durumdadır. Tüm GPIO pinlerinin hızları iç gürültü, güç tüketimi ve elektromanyetik emisyonu azaltabilmek için yazılımsal olarak ayarlanabilir.

Analog Dijital Dönüştürücü (ADC) : 16 kanala kadar paylaşımli 3 adet 12 bitlik ADC modülü mevcuttur. ADC DMA modülü tarafından kullanılabilir. A/D çevrim ve timerları senkronize etmek için ADC ler TIM1, TIM2, TIM3, TIM4, TIM5 ya da TIM8 tarafından tetiklenmeli.

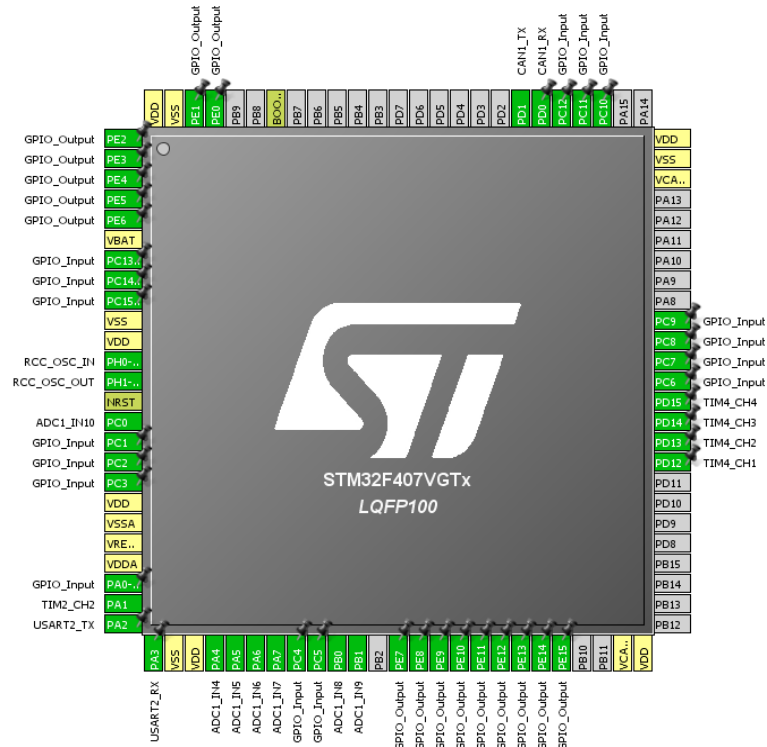
Dijital Analog Dönüştürücü (DAC): 2 adet 12 bitlik tamponlu DAC modülü 2 adet analog voltaj sinyali almak için kullanılır. DAC Üçgen dalga ve gürültü dalga üretimi yapabilir.

### **3.1.2. Geliştirme ortamı**

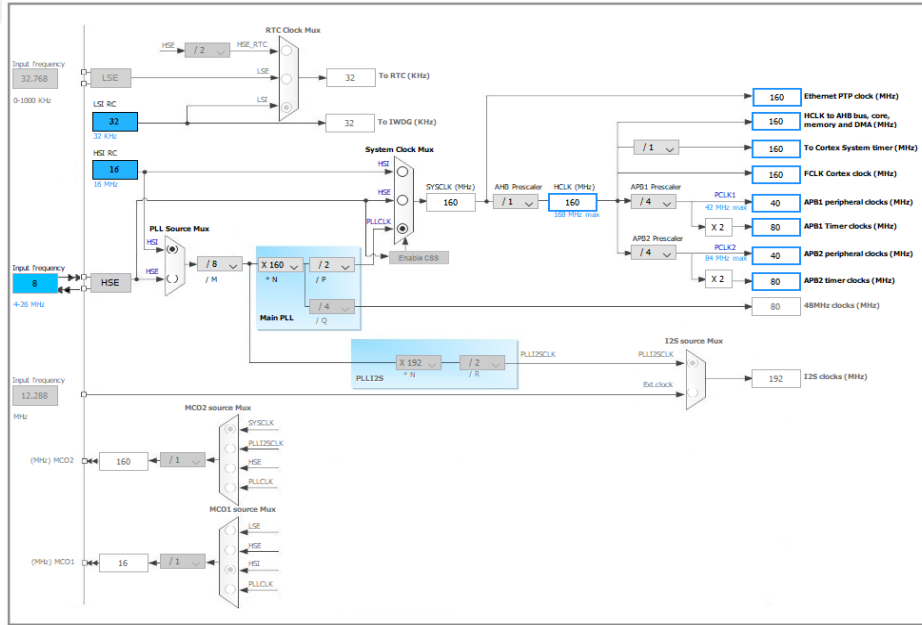
Günümüzde ARM mimarisinin ve 32 bitlik işlemcilerin kullanım alanlarının artması geliştiriciler için daha gelişmiş geliştirme ortamlarını ihtiyaç kılıyor. Karmaşıklığın az olması, proje içerisinde geçişlerin hızlı olması, derleme hızı ve kullanım kolaylığı bir geliştirme ortamından beklenen temel özelliklerin başında gelir. STM işlemcilerin programlanabileceği geliştirme ortamları çeşitli firmalar tarafından seçenek olarak sunulmaktadır. Keil, True Studio, SW4STM, ve IAR en çok kabul gören geliştirme ortamlarıdır.

Geliştirme ortamları için hata ayıklama özellikleri geliştirme ortamı seçiminde en önemli kriterlerden birisidir. Proje geliştiren yapılan değişikliklerin kontrolü için yazılımcı tarafından sürekli olarak hata ayıklama kullanılmaktadır. Bu sebeple hata ayıklama menüsü bir geliştirme ortamının en önemli gereksinimlerindedir.

ST firmasının 32 bitlik ARM mimarili işlemcilerin daha kolay geliştirilmesine olanak sağlamak için grafiksel arayüzlü STM32CubeMX programını geliştiricilere sunmuştur. Bu arayüz ile genel amaçlı giriş çıkışlar, işlemci saat ayarları, güç tüketim hesabı, ADC, UART, SPI, I2C, Timer ve CAN gibi donanımların konfigürasyonlarının daha kolay bir şekilde ayarlanabilmektedir. CubeMx HAL kütüphanelerini kullanarak projeyi oluşturur. HAL(Donanım Soyutlama Katmanı) kütüphaneleri ST firmasının kendi işlemcileri için oluşturduğu alt seviye kütüphanelerdir. STM32CubeMX pin seçimi Şekil 3.2 de gösterilmiştir.



Şekil 3.2. STM32CubeMX işlemci giriş-çıkış seçimi.



Şekil 3.3. STM32F4 clock ayarları.

Giriş-Çıkış ünitesinin seçilen pinlerin fonksiyon atamaları Tablo 3.1'de gösterilmektedir.

Tablo 3.1. STM32CubeMx Giriş-Çıkış Ünitesi pin tanımlamaları.

Pin Numarası	Pin Adı	Pin Türü	Alternatif Fonksiyon	Açıklama
1	PE2	Çıkış	GPIO_Output	Ön Far Çıkışı
2	PE3	Çıkış	GPIO_Output	Silecek Çıkışı
3	PE4	Çıkış	GPIO_Output	Korna Çıkışı
16	PC1	Giriş	GPIO_Input	Kontak Anahtarı Girişi
17	PC2	Giriş	GPIO_Input	Motor Freni Girişi
18	PC3	Giriş	GPIO_Input	Vites Girişi
24	PA1	Giriş	TIM2_CH2	Tekerlek Devir Sensörü Sağ
29	PA4	Giriş	ADC1_IN4	Gaz Pedalı Girişi
30	PA5	Giriş	ADC1_IN5	Direksiyon Dönüş Açısı Girişi
31	PA6	Giriş	ADC1_IN6	Tekerlek Dönüş Açısı Girişi
33	PC4	Giriş	GPIO_Input	Geri Vites Girişi
34	PC5	Giriş	GPIO_Input	Ayak Freni Girişi
63	PC6	Giriş	GPIO_Input	Sağ Sinyal Anahtarı Girişi
64	PC7	Giriş	GPIO_Input	Sol Sinyal Anahtarı Girişi
65	PC8	Giriş	GPIO_Input	Ön Far Anahtar Girişi
66	PC9	Giriş	GPIO_Input	Silecek Anahtar Girişi
78	PC10	Giriş	GPIO_Input	Korna Girişi
81	PD0	G/Ç	CAN1_RX	CAN L
82	PD1	G/Ç	CAN1_TX	CAN H
97	PE0	Çıkış	GPIO_Output	Sağ Sinyal Çıkışı
98	PE1	Çıkış	GPIO_Output	Sol Sinyal Çıkışı

Araç kontrol sisteminin seçilen pinlerin fonksiyon atamaları Tablo 3.2’de gösterilmektedir.

Tablo 3.2. STM32CubeMx Araç Kontrol Sistemi pin tanımlamaları.

Pin Numarası	Pin Adı	Pin Türü	Alternatif Fonksiyon	Açıklama
1	PE2	Çıkış	GPIO_Output	Geri Vites Çıkışı
2	PE3	Çıkış	GPIO_Output	Fren Lambası Çıkışı
3	PE4	Çıkış	GPIO_Output	Sesli İkaz Çıkışı
24	PA1	Giriş	TIM2_CH2	Tekerlek Devir Sensörü Sol
25	PA2	G/Ç	USART2_TX	USART Tx
26	PA3	G/Ç	USART2_RX	USART Rx
29	PA4	Giriş	ADC1_IN4	Motor Sıcaklık Sensör Girişi
30	PA5	Giriş	ADC1_IN5	Motor Sürücü Sıcaklık Sensör Girişi
31	PA6	Giriş	ADC1_IN6	Batarya Sıcaklık Sensör Girişi
32	PA7	Giriş	ADC1_IN7	Batarya Gerilim Sensör Girişi
56	PD9	Giriş	GPIO_Input	AKS-GÇU Seçim Pini (1=AKS, 0=GÇU)
59	PD12	Çıkış	TIM4_CH1	Sağ Motor PWM Çıkışı
60	PD13	Çıkış	TIM4_CH2	Sol Motor PWM Çıkışı
81	PD0	G/Ç	CAN1_RX	CAN L
82	PD1	G/Ç	CAN1_TX	CAN H
97	PE0	Çıkış	GPIO_Output	Motor Freni Çıkışı
98	PE1	Çıkış	GPIO_Output	Vites Çıkışı

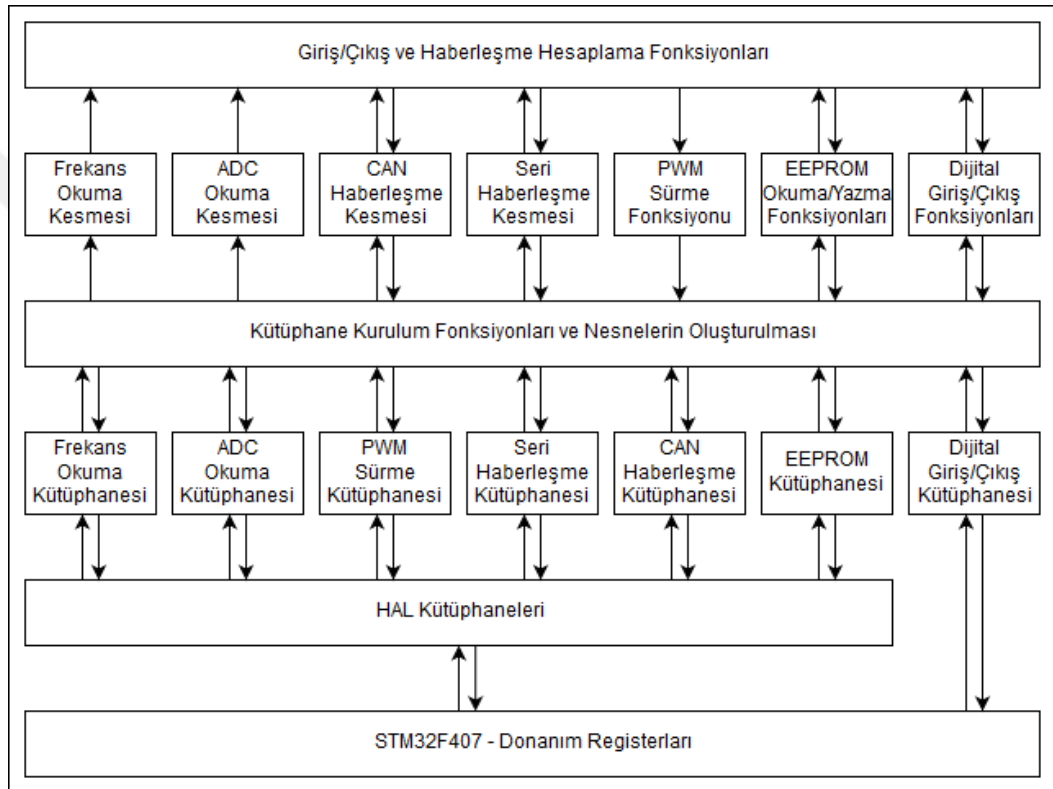
### 3.2. Yöntem ve Algoritmalar

Sistemde bulunan ünitelerden ikisinde de donanım olarak aynı konfigürasyonlar tanımlanmıştır. Sensör kartı kaynak ID’si belirlendikten sonra GÇU olarak tanımlanması durumunda aldığı bilgileri gönderme işlemi yapmaktadır. Yazılım iki farklı grupta değerlendirilmelidir. Bunlardan birincisi donanımın içerisinde çalışan gömülü yazılımdır. Diğeri ise AKS’den aldığı verileri bilgisayar ortamında toplayacak ve bilgilerin gösterilmesini sağlayacak arayüz yazılımdır.

#### 3.2.1. Gömülü yazılım

Yazılım tasarımı yapılırken STM32CubeMX ile oluşturulan proje üzerinde herhangi bir değişiklik yapmadan projeye ana çalışma fonksiyonu olarak tanımlanan YED\_MAIN()

fonksiyonunu içinde bulunduran YEDMAIN.c ve YEDMAIN.h dosyalarımın eklenmesi sağlanmıştır. main.c içerisinde main() fonksiyonu içerisinde başlangıç ayarları yapıldıktan sonar çalışma fonksiyonumuz olan YED\_MAIN() fonksiyonu çağrılmaktadır. Bu kullanımın geliştiriciye sağlayacağı avantaj STM32CubeMX ile bir değişiklik yapıldığı veya HAL kütüphaneleri güncellenmek istendiğinde projede herhangi bir değişiklik yapma ihtiyacı gerektirmemesidir. Şekil 3.4 ile yazılım mimarisi gösterilmektedir.

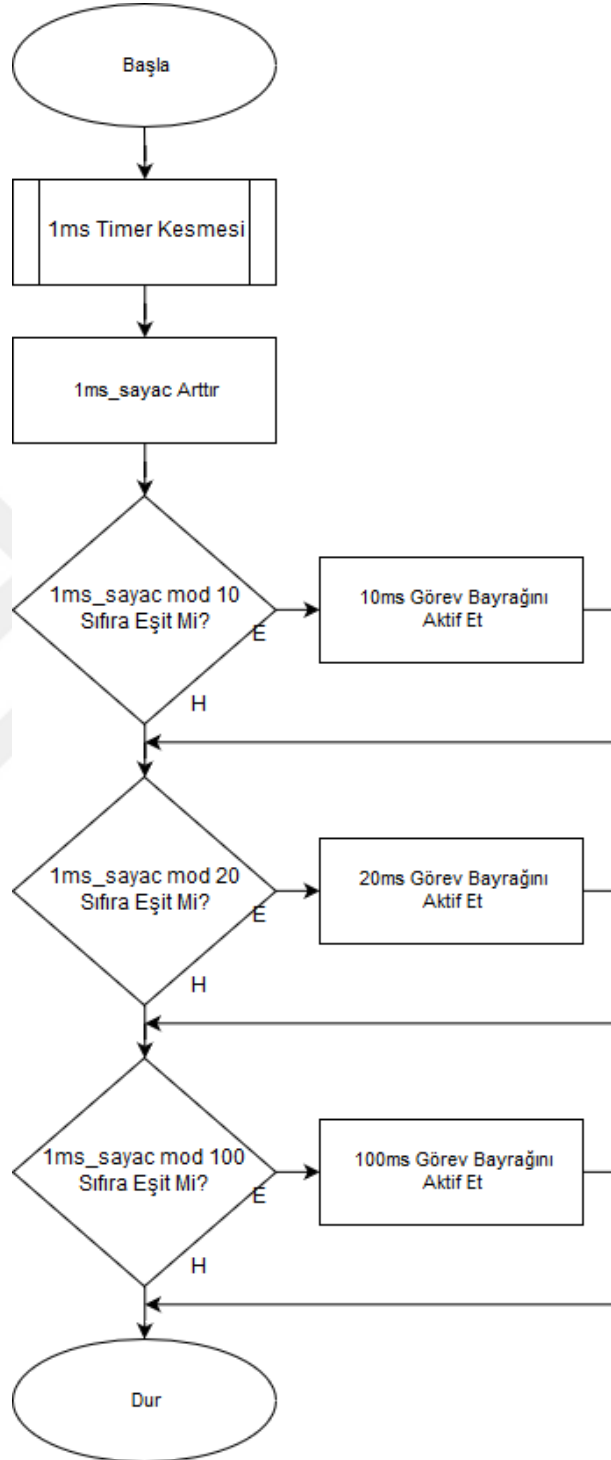


Şekil 3.4. Yazılım mimarisi.

Yazılım içerisinde her işlem belirli zaman periyotlarına bölünmüştür. Bunlar 10ms, 20ms ve 100ms olmak üzere 3 farklı değerdir. Proje oluşturulurken 1 adet timer 1ms değerine kurularak her 1ms de bir defa kesmeye girmesi sağlanmaktadır. Bu kesme içerisinde 1ms sayac artırılarak zaman değeri tutulur ve Arkasından sayacın her 10'un katları için 10ms'lik bayrak, 20ms'nin katları için 20ms'lik bayrak olacak şekilde oluşturulan bütün periyotlar için bayraklar aktif edilir. YED\_MAIN() içerisinde bulunan while(1) döngüsü içinde sırası ile 10ms bayrağının aktif edilip edilmediğini sorgulayarak eğer bayrak aktif edildi ise bayrağı pasif ettikten sonra 10ms'de yapılacak işlemler yazılır. Bu durum diğer



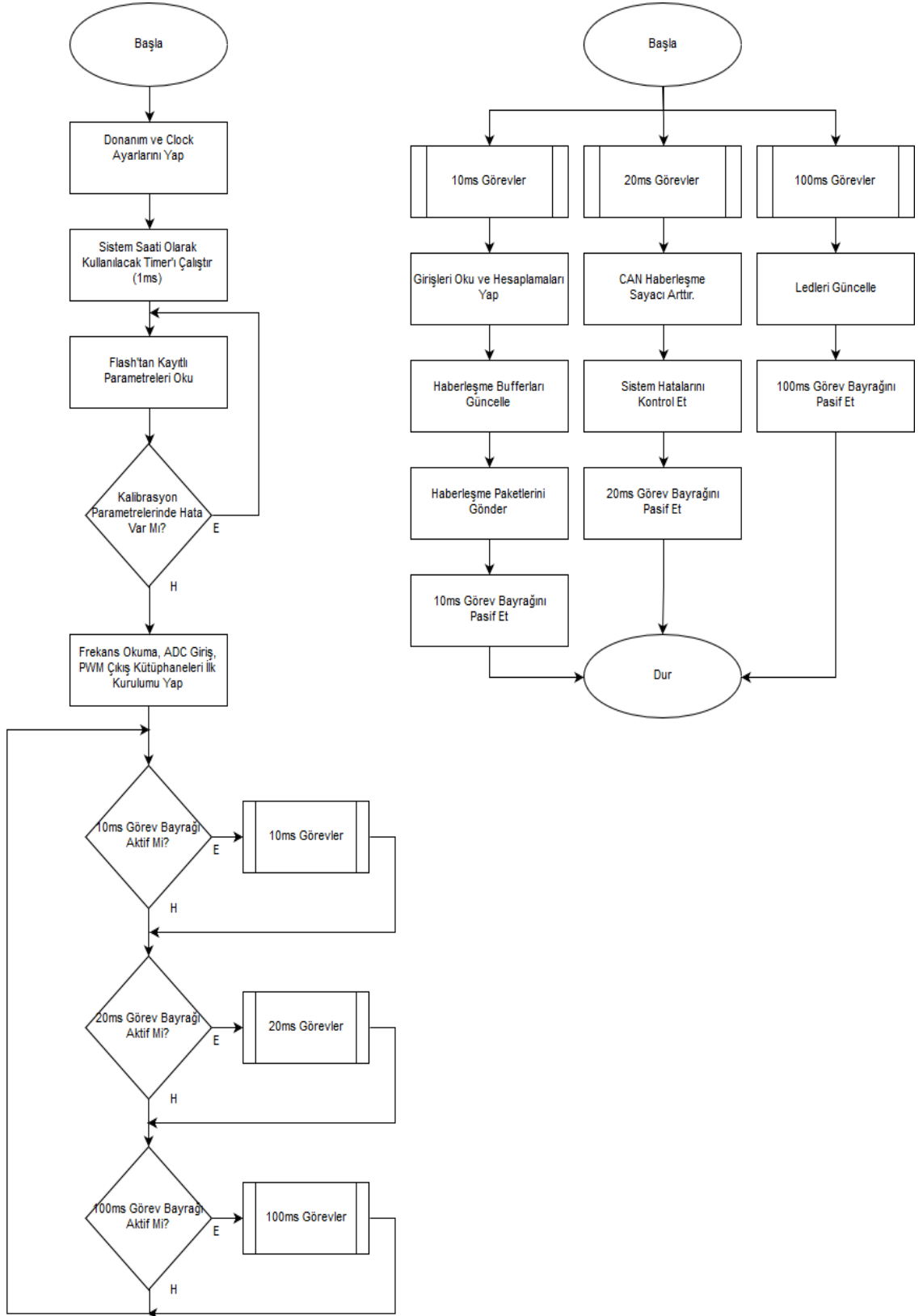
periyotlar için de aynı şekilde uygulanır. Şekil 3.5 de timer kesmesinin yazılım algoritması gösterilmektedir.



Şekil 3.5. 1ms timer kesme algoritması.

Gömülü yazılımda yapılan işlemleri ana başlıklara ayırmak gerekirse bunlar, ADC okuma, Frekans okuma, PWM çıkış, Kalibrasyon yapısı, Canbus haberleşme, Seri haberleşme olarak ayrılabilir. Şekil 3.6 ile gömülü yazılım akış diyagramı gösterilmektedir

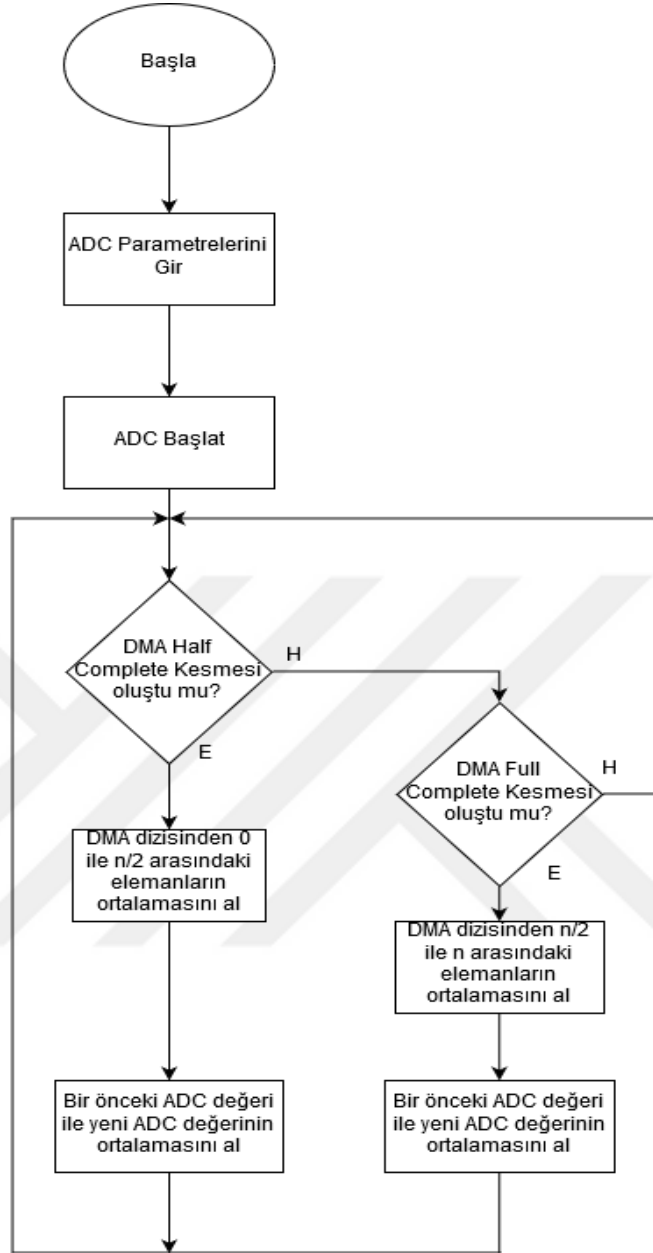




Şekil 3.6. Gömülü yazılım akış diyagramı.

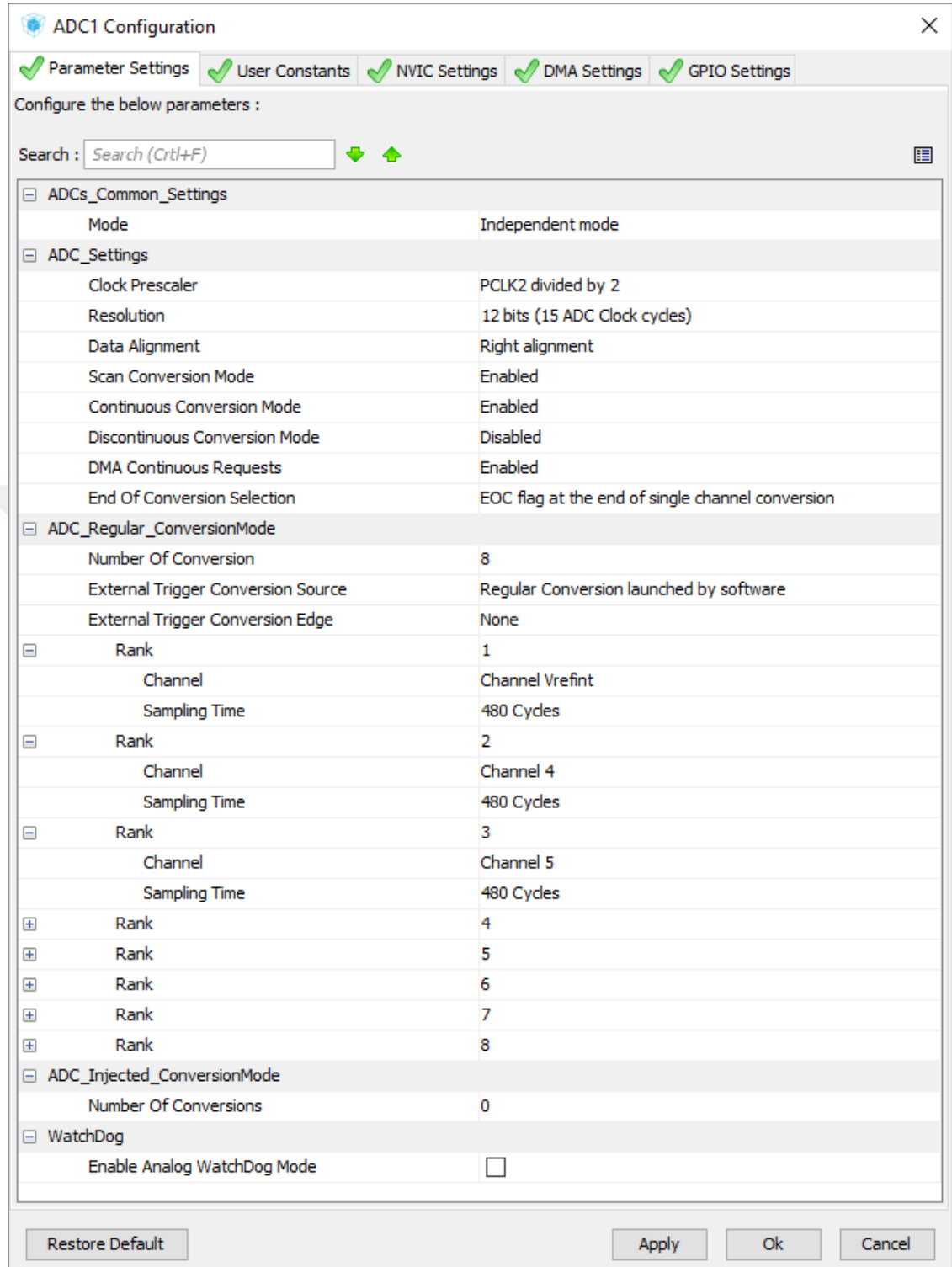
### 3.2.1.1. ADC okuma

STM32F4 12bit çözünürlüklü ADC donanımına sahiptir. Girişe uygulanan voltaj değerini okumak işlemci yükünü arttıracığından okuma işlemi DMA tarafından gerçekleştirilmektedir. ADC parametrelerinde hangi DMA modülünün hangi kanalının kullanılacağı seçilmektedir. ADC başlatma fonksiyonunda DMA'nın verileri otomatik olarak atacağı bir adres ve DMA uzunluğu girilmektedir. DMA yapısı itibari ile yarı tamamlandı(half complete) ve tamamlandı(full complete) kesmeleri üretmektedir. Bu bize DMA uzunluğu 32 olan bir ADC modülünde 16. Eleman alındığında ve 32. Eleman alındığında ayrı ayrı kesmeler oluşturur. Bu kesmelerden yarı tamamlandı kesmesi oluştuğunda dizideki ilk 16 değer en yüksek ve en düşük değerli birer tane elemanı hariç ortalaması alınır. Tamamlandı kesmesi oluştuğunda ise son 16 elemanın en yüksek ve en düşük değerli birer tane elemanı hariç ortalaması alınır. Bu durum bizi her ADC çevrim tamamlandığında kesmeye giderek programın bölünmesini engeller aynı zamanda 16 adet değer ortalamasının alınması sebebiyle basit bir filtre gibi davranır. Bu işlem tamamlandıktan sonra bir önceki hesaplanan ADC değeri ile yeni hesaplanan değer ortalaması alınarak ADC değişimlerinin daha yumuşak geçişli olması sağlanır. ADC kütüphanesi kurulurken voltaj hesabının aktif edilmesi durumunda STM32CubeMX tarafından seçilen Vrefint seçeneği hangi sıralamada olduğu bilgisi girilerek ADC girişlerinde bulunan voltaj hesaplaması mV cinsinden yapılmaktadır. ADC işleminin çalışma algoritması Şekil 3.7 te gösterilmiştir.



Şekil 3.7. ADC giriş hesaplama algoritması.

STM32CubeMX içerisinde konfigüre edilen ADC'lerin doğru ayarlarının girilmesi hesaplama açısından önemlidir. Kullanılan parametre ayarları Şekil 3.8 de gösterilmiştir.



Şekil 3.8. ADC parametre ayarları.

ADC okuma ile ilgili DMA ayarları Şekil 3.9 da gösterilmektedir.

ADC1 Configuration

Parameter Settings User Constants NVIC Settings DMA Settings GPIO Settings

DMA Request	Stream	Direction	Priority
ADC1	DMA2 Stream 0	Peripheral To Memory	Low

Add Delete

DMA Request Settings

		Peripheral	Memory
Mode	Circular	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Increment Address		<input type="checkbox"/>	<input checked="" type="checkbox"/>
Use Fifo	<input checked="" type="checkbox"/>	Threshold	Full
Data Width	Half Word		Half Word
Burst Size	Single		Single

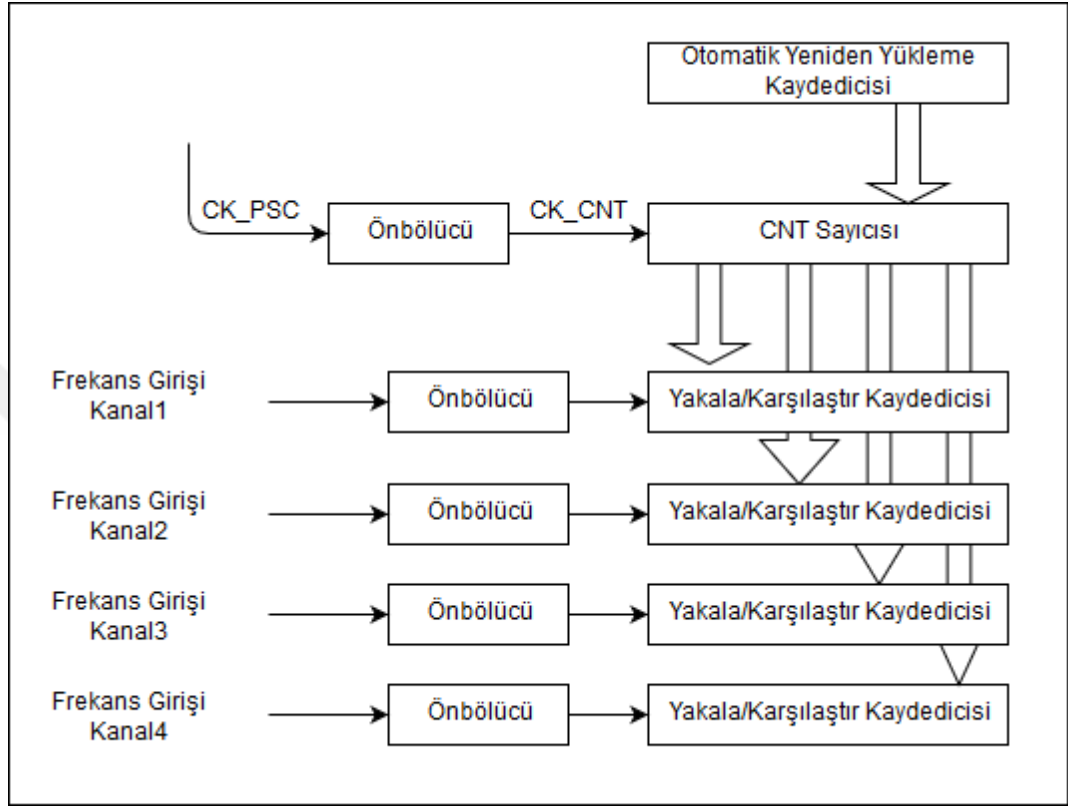
Restore Default Apply Ok Cancel

Şekil 3.9. ADC DMA ayarları.

### 3.2.1.2. Frekans okuma

STM32F4 işlemcisinde frekans okumak için en etkili yol olan Timer'ların giriş yakalama (input capture - IC) özelliği kullanılmaktadır. Giriş yakalama blok diyagramı Şekil 3.10 de gösterilmektedir. Timer girişi örnekledikten sonra kenar yakalayıcı ile konfigürasyona göre yükselen ya da alçalan kenar kontrolünü yapar. Timer'ın kurulu olduğu sayaç bizim yakalama çözünürlüğümüzü verir. Timer saymaya başladığında ilk yükselen kenar gelen değeri yakala/karşılaştır kaydedicisine kaydeder. Bir sonraki yükselen kenar giriş

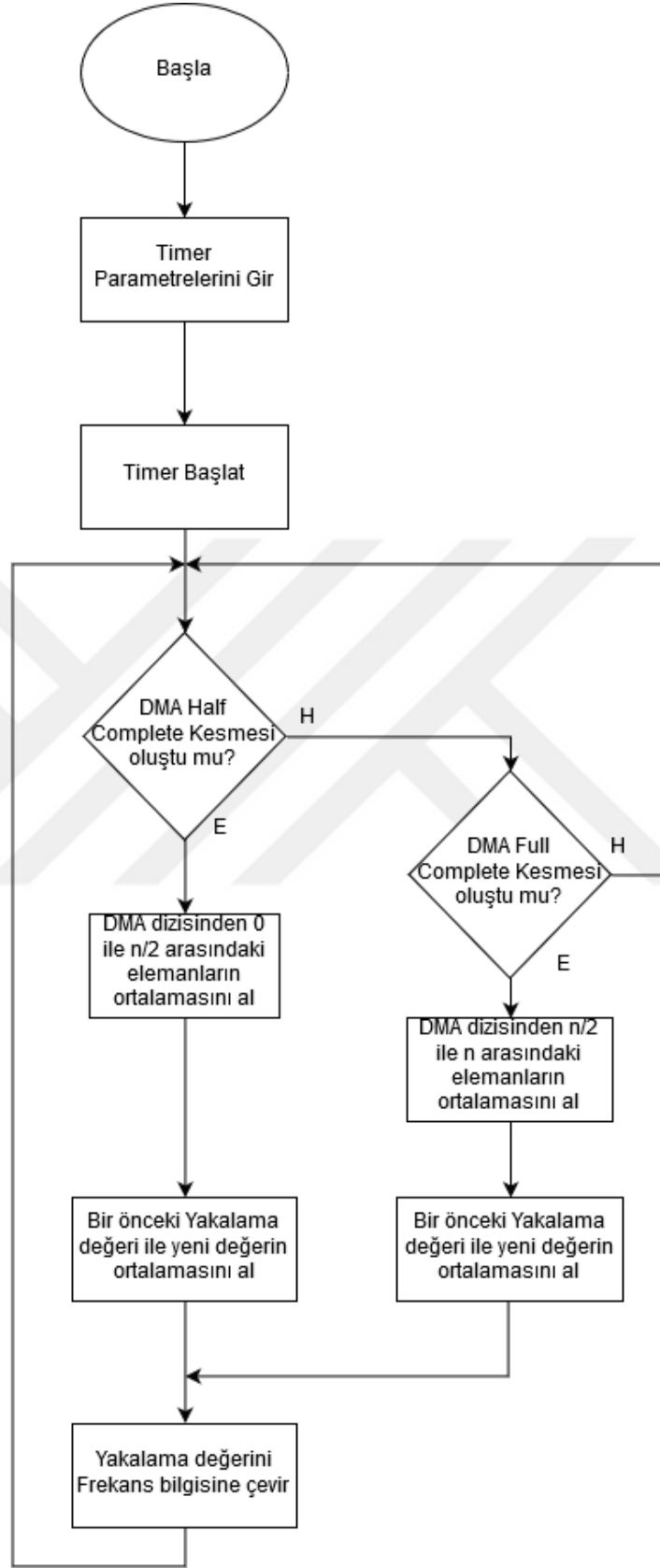
geldiğinde o andaki timer sayacının değeri alınır. Bu iki değer arasındaki fark bize periyodumuzu verir. Timer girişleri bu işlemleri donanımsal olarak yaptığı için yazılım tarafında fazladan iş yükü oluşturmaz.



Şekil 3.10. Giriş yakalama donanımı blok diyagramı.

Timer tarafından giriş frekansının yakalanmasından sonra alınan değerler DMA üzerine yazılır ve DMA'nın 'HalfComplete' ve 'FullComplete' kesmeleri kullanılarak belirlenen sayıda örnek alındıktan sonra her kesme oluştuğunda oluşan kesmeden bir önceki kesmeye kadar olan değer birbirinden çıkarılarak bölünür. Bu bize DMA uzunluğu kadar alınan yakalama değerinin ortalamasını olarak daha uygun sonuçlar elde etmemizi sağlar. Kesmeler sonucunda hesaplanan her değer bir önce oluşan kesmenin sonucu ile ortalaması alınarak sonuç kısmına yazılır. Frekans girişinin hesaplama algoritması Şekil 3.11 de gösterilmektedir.





Şekil 3.11. Frekans giriş hesaplama algoritması.

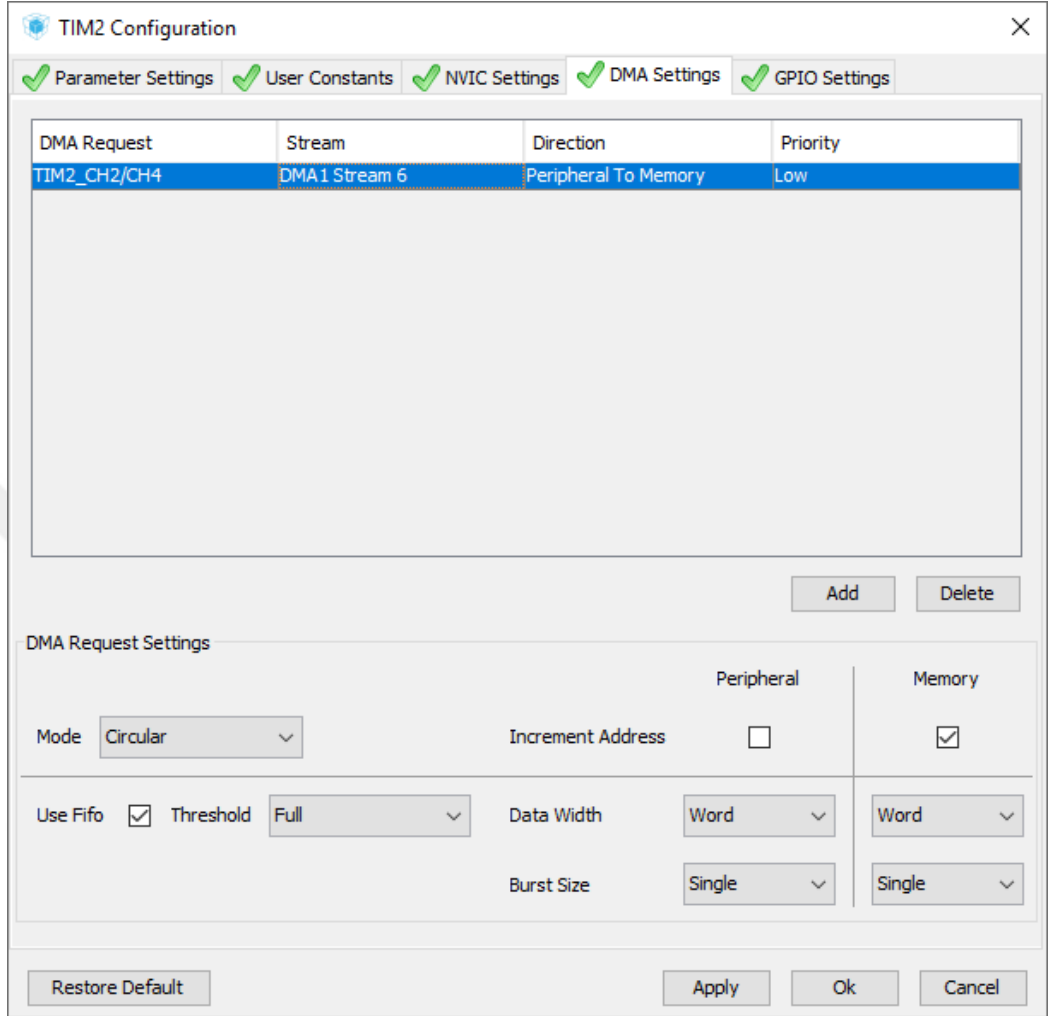
Frekans giriři için yapılan Timer2 parametre ayarları Őekil 3.12 da gsterilmektedir.

The image shows a 'TIM2 Configuration' dialog box with a search bar and several sections of parameters. The 'Counter Settings' section includes Prescaler (PSC - 16 bits value) set to 79, Counter Mode set to Up, Counter Period (AutoReload Register - 32 bits va... set to 0xFFFFFFFF, and Internal Clock Division (CKD) set to No Division. The 'Trigger Output (TRGO) Parameters' section includes Master/Slave Mode (MSM bit) set to Disable (Trigger input effect not delayed) and Trigger Event Selection set to Reset (UG bit from TIMx\_EGR). The 'Input Capture Channel 2' section includes Polarity Selection set to Rising Edge, IC Selection set to Direct, Prescaler Division Ratio set to No division, and Input Filter (4 bits value) set to 0. The dialog box has a 'Restore Default' button on the left and 'Apply', 'Ok', and 'Cancel' buttons on the right.

Section	Parameter	Value
Counter Settings	Prescaler (PSC - 16 bits value)	79
	Counter Mode	Up
	Counter Period (AutoReload Register - 32 bits va...	0xFFFFFFFF
	Internal Clock Division (CKD)	No Division
Trigger Output (TRGO) Parameters	Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
	Trigger Event Selection	Reset (UG bit from TIMx_EGR)
Input Capture Channel 2	Polarity Selection	Rising Edge
	IC Selection	Direct
	Prescaler Division Ratio	No division
	Input Filter (4 bits value)	0

Őekil 3.12. Timer2 frekans giriř parametreleri.

Frekans giriři okumak için DMA kesme ayarları Şekil 3.13 da gösterilmektedir.

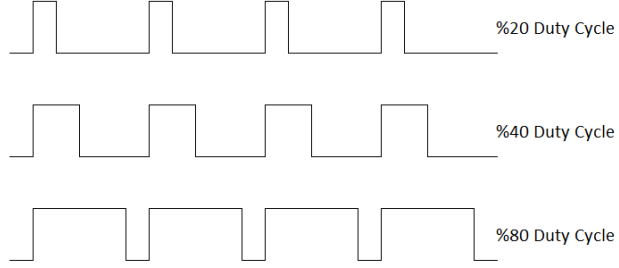


Şekil 3.13. Timer2 frekans giriř DMA ayarları.

### 3.2.1.3. PWM çıkıř

Araç control sistemi aracın motor kontrolünü motor sürücülere PWM sinyali göndererek yapmaktadır. PWM (Pulse Width Modulation – Darbe genişlik modülasyonu) genel anlamıyla elektriksel sinyal tarafından verilen ortalama gücü, ayrı ayrı parçalar halinde etkin bir şekilde keserek azaltma yöntemidir. PWM denilince bu yöntemin en önemli kavramı olan ‘Duty Cycle’ karşımıza çıkıyor. ‘Duty Cycle’, ‘görev döngüsü’ olarak tanımlanabilir ve bir sinyalin periyotta aktif kalma değerine denir. Duty Cycle ne kadar yüksek olursa aktarılan güç o kadar yüksek, ne kadar düşük olursa aktarılan güç o kadar düşük olur. Farklı Duty Cycle’lardan oluşan PWM sinyalleri Şekil 3.14 de

gösterilmektedir. Periyot pwm in kendini 1 kere tekrar etme süresine denir. Frekans ise 1 sn içerisinde kaç defa sinyalin kendisini tekrar ettiğini ifade eder.



Şekil 3.14. Farklı Duty Cycle değerlerinde PWM sinyalleri.

PWM ayar parametreleri Şekil 3.15 de gösterilmektedir.

The screenshot shows the 'TIM4 Configuration' dialog box. The 'Parameter Settings' tab is active. The 'Pulse (16 bits value)' field is highlighted in blue and set to 0. The 'Fast Mode' is set to 'Disable' and 'CH Polarity' is set to 'High'. The 'Pulse (16 bits value)' field has a tooltip that reads: 'Pulse (16 bits value) Pulse must be between 0 and 65 535.' The dialog box also includes a search bar, a list of settings, and buttons for 'Restore Default', 'Apply', 'Ok', and 'Cancel'.

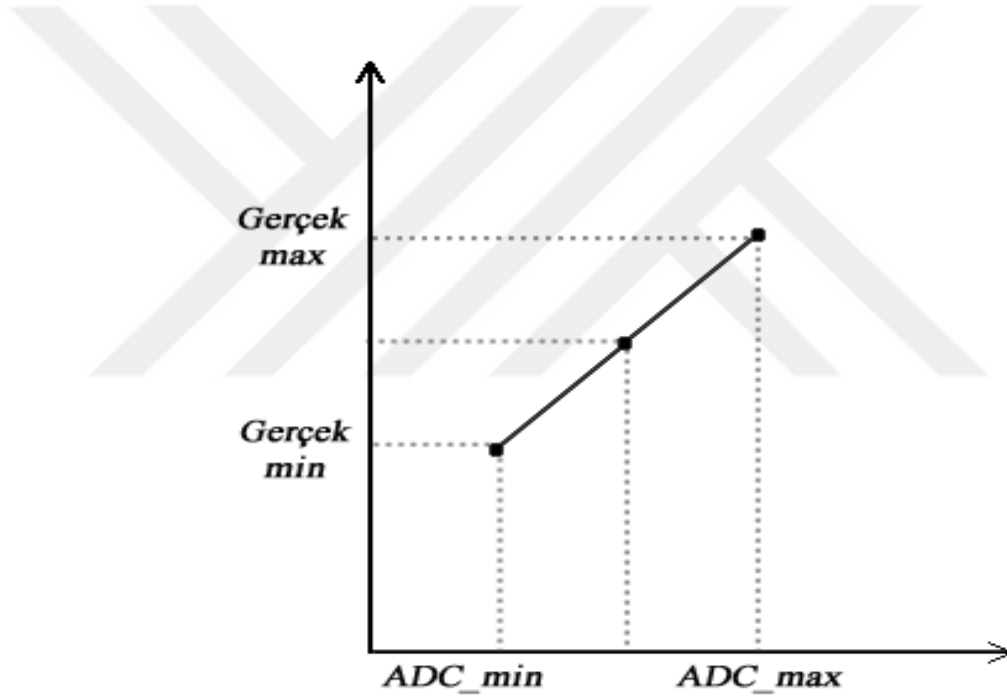
Parameter	Value
Prescaler (PSC - 16 bits value)	79
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits va...)	999
Internal Clock Division (CKD)	No Division
Master/Slave Mode (MSM bit)	Disable (Trigger input effect not delayed)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)
Mode	PWM mode 1
Pulse (16 bits value)	0
Fast Mode	Disable
CH Polarity	High

Şekil 3.15. PWM çıkış parametreleri.

#### 3.2.1.4. Kalibrasyon yapısı

Araç control sisteminde bulunan sensörlerin değerlerinin anlam kazanabilmesi için kalibrasyon yapılmalıdır. Bu kalibrasyon işleminin doğruluğu sistemin doğruluğunu direk etkiler. Sistemde kalibrasyonu yapılması gereken gaz pedalı, direksiyon açısı, tekerlek açısı, pwm çıkış gibi bazı değerlerin kalibre edilmesi gerekmektedir.

Kalibrasyon işlemi system lineer olarak kabul edildiğinden lineer interpolasyon kullanılarak uygulanmıştır. Lineer interpolasyonun grafik gösterimi Şekil 3.16 de gösterilmiştir.



Şekil 3.16. Lineer interpolasyon.

En basit kalibrasyon yöntemlerinden olan bu yöntem ile gaz pedalı minimum adc değeri durumu için  $ADC_{min}$ , gaz pedalının en basılı olduğu noktada ise maksimum ADC durumu yazılır. Gerçek min ve gerçek max değerleri ise gaz pedalının okunduktan sonra bize hangi anlamlı değeri vermek istediği aralıktır. Yani gaz pedalı için gerçek min değeri 0, gerçek max değeri 1000 olarak verilmektedir.

### 3.2.1.5. CANBUS haberleşme

CANBUS haberleşme kısmında AKS ile GÇÜ arasında haberleşmede J1939 protokolüne bağlı kalınarak geliştirme sağlanmıştır. Bu işlem için öncelikle veritabanı tasarımı yapılmalıdır. Veritabanı bizim gönderilmesini istediğimiz verilerin listesidir. Oluşturulan veritabanı Tablo 3.3’de gösterilmektedir.

Tablo 3.3. CAN hattına gönderilen bilgiler.

Veri Adı	Çarpan	Limitler	Ofset	Uzunluk
gaz pedalı	0.4%/bit	0%-100%	0	8 bit
direksiyon dönüş açısı	1°/bit	-450°- +450°	-450	16 bit
tekerlek dönüş açısı	1°/bit	-30° +45°	-30	8 bit
seçili vites	1durum/bit	0 - 4	0	8 bit
sağ ön tekerlek hızı	1 RPM/bit	0-4500 RPM	0	16 bit
sol ön tekerlek hızı	1 RPM/bit	0-4500 RPM	0	16 bit
motor sıcaklık	1°C/bit	-40°- +210°	-40	8 bit
motor sürücü sıcaklık	1°C/bit	-40°- +210°	-40	8 bit
Batarya sıcaklık	1°C/bit	-40°- +210°	-40	8 bit
Batarya gerilim	1V/bit	0-200V	0	8 bit
Sağ arka motor pwm duty	0.4%/bit	0%-100%	0	8 bit
Sol arka motor pwm duty	0.4%/bit	0%-100%	0	8 bit

Oluşturulan veritabanında mesajların gruplandırılarak PGN lerinin ve mesaj gönderim sıklığının ayarlanması gerekmektedir. Oluşturulan gruplandırma tablosu Tablo 3.4 de gösterilmektedir.

Tablo 3.4. Grup tanımlamaları.

PGN	Grup Adı	Gönderim Sıklığı
0x1010	sürücü kontrol değerleri	10ms
0x1011	tekerlek sensör değerleri	50ms
0x1012	motor sıcaklık değerleri	100ms
0x1013	batarya değerleri	100ms
0x1014	motor sürme değerleri	10ms

Mesajların öncelik ve gruplarına göre ID tanımlaması yapılıncaya ortaya çıkan mesaj ID leri Tablo 3.5 da gösterilmektedir.

Tablo 3.5. Mesaj ID tanımlamaları.

Mesaj ID	Veri	Bayt	Kaynak Cihaz
0x181010A2	gaz pedalı	1	GÇU
0x181010A2	seçili vites	3	GÇU
0x181010A2	direksiyon dönüş açısı	5 - 6	GÇU
0x181011A2	tekerlek dönüş açısı	1	GÇU
0x181011A2	sağ ön tekerlek hızı	3 - 4	GÇU
0x181011A1	sol ön tekerlek hızı	5 - 6	AKS
0x181012A1	motor sıcaklık	1	AKS
0x181012A1	motor sürücü sıcaklık	2	AKS
0x181013A1	batarya sıcaklık	1	AKS
0x181013A1	batarya gerilim	2	AKS
0x181014A1	sağ arka motor pwm duty	1	AKS
0x181014A1	sol arka motor pwm duty	3	AKS

### 3.2.1.6. Seri haberleşme

Araç kontrol sisteminin bilgisayar arayüzü ile haberleşmesi seri port üzerinden yapılmaktadır. Araç kontrol sistemi 10ms'lik aralıklarla bilgisayar arayüzüne veri paketi göndermektedir. Haberleşme paketi yapısı Şekil 3.17'de gösterilmektedir.

Paket Başlangıç	Komut	Uzunluk	Veri0	...	...	Veri Son	Doğrulama	Paket Sonu
--------------------	-------	---------	-------	-----	-----	----------	-----------	---------------

Şekil 3.17. Haberleşme paket yapısı.

Seri porttan veri gönderilirken doğru veri gönderildiğinden emin olabilmek için veri gönderildiğini belirten bir başlangıç verisi gönderilir, sonrasında paketin içeriğinin ne olduğu bilgisi gönderilir. Arkasından uzunluk bilgisi gönderilmektedir. Uzunluk bilgisine göre arkadan gelen uzunluk kadar bayt gönderilir. Paket başlangıcından sonra son veri gelene kadar olan kısımda bulunan baytlar toplanarak 255 değeri ile maskeleye yapılır. Kalan değer doğrulama değeri olarak elde edilmiş olur. Doğrulama sonrasında paketin bittiğini belirten paket sonu baytları gönderilir.

Bu yapıda paket içerisindeki herhangi bir veri bozulduğu durumda paket hatalı olacağından haberleşmenin doğruluğu sağlanmış olur. Oluşturulan sistemde tanımlanan komut açıklamaları Tablo 3.6'da gösterilmiştir.



Tablo 3.6. Komut açıklamaları.

Komut	Açıklama
0x11	Sistem durumu görüntüleme komutu (diagnostik)
0xA1	Arayüz tarafından gönderilen kalibrasyon bilgileri 1 no'lu paket
0xA2	Arayüz tarafından gönderilen kalibrasyon bilgileri 2 no'lu paket
0xA3	Arayüz tarafından gönderilen kalibrasyon bilgileri 3 no'lu paket
0xA4	Arayüz tarafından gönderilen kalibrasyon bilgileri 4 no'lu paket
0xA5	Arayüz tarafından gönderilen kalibrasyon bilgileri 5 no'lu paket
0xA6	Arayüz tarafından gönderilen kalibrasyon bilgileri 6 no'lu paket
0xA7	Arayüz tarafından gönderilen kalibrasyon bilgileri 7 no'lu paket
0xB0	Arayüz tarafından AKS ye gönderilen kalibrasyon bilgileri istek
0xB1	AKS tarafından gönderilen kalibrasyon bilgileri 1 no'lu paket
0xB2	AKS tarafından gönderilen kalibrasyon bilgileri 2 no'lu paket
0xB3	AKS tarafından gönderilen kalibrasyon bilgileri 3 no'lu paket
0xB4	AKS tarafından gönderilen kalibrasyon bilgileri 4 no'lu paket
0xB5	AKS tarafından gönderilen kalibrasyon bilgileri 5 no'lu paket
0xB6	AKS tarafından gönderilen kalibrasyon bilgileri 6 no'lu paket
0xB7	AKS tarafından gönderilen kalibrasyon bilgileri 7 no'lu paket

Her komut için paket yapısı ve veri içeriğindeki bilgiler değişiklik göstermektedir. Sistem durumu görüntüleme komut paketi için oluşturulan paket yapısı Tablo 3.7'de gösterilmektedir.

Tablo 3.7. Sistem durumu görüntüleme komut paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4	V5
0x59	0x45	0x44	0x11	26	Süre 1	Süre 2	Süre 3	Süre 4	Gaz Pedalı L	Gaz Pedalı H
V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16
Tekerlek Devir Sağ L	Tekerlek Devir Sağ H	Tekerlek Devir Sol L	Tekerlek Devir Sol H	Direksiyon Açısı L	Direksiyon Açısı H	Tekerlek Açısı L	Tekerlek Açısı H	Motor PWM Sağ L	Motor PWM Sağ H	Motor PWM Sol L
V17	V18	V19	V20	v21	V22	V23	V24	CRC	EOF1	EOF2
Motor PWM Sol H	Batarya Gerilim L	Batarya Gerilim H	Batarya Sıcaklık L	Batarya Sıcaklık H	Toplam Yol	Digital Bilgiler 1	Digital Bilgiler 2	-	0x44	0x45
EOF3										
0x60										

Arayüzde kalibrasyon başlatıldığı zaman işlemci içerisinde yüklü parametrelerin gelmesi için kalibrasyon giriş komutu gönderilir. Kalibrasyon bilgileri istek paketi Tablo 3.8’de gösterilmektedir.

Tablo 3.8. Arayüz tarafından AKS ye gönderilen kalibrasyon bilgileri istek paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	CRC	EOF1	EOF2	EOF3
0x59	0x45	0x44	0xB0	1	0xFF	-	0x44	0x45	0x60

Araç kontrol sistemi gaz pedalı kalibrasyon bilgilerini ‘0xB1’ komut numaralı paket ile göndermektedir. AKS tarafından gönderilen gaz pedalı kalibrasyon paketi Tablo3.9’de gösterilmektedir.

Tablo 3.9. AKS tarafından gönderilen gaz pedalı kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xB1	10	Gaz Pedalı Min ADC L	Gaz Pedalı Min ADC H	Gaz Pedalı Max ADC L	Gaz Pedalı Max ADC H	Gaz Pedalı Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Gaz Pedalı Min REAL H	Gaz Pedalı Max REAL L	Gaz Pedalı Max REAL H	Gaz Pedalı ADC Katsayı	Gaz Pedalı Offset	-	0x44	0x45	0x60	

Araç kontrol sistemi direksiyon açısı sensör kalibrasyon bilgilerini ‘0xB2’ komut numaralı paket ile göndermektedir. AKS tarafından gönderilen direksiyon açısı kalibrasyon paketi Tablo 3.10’de gösterilmektedir.

Tablo 3.10. AKS tarafından gönderilen direksiyon açısı kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xB2	10	Direksiyon Açısı Min ADC L	Direksiyon Açısı Min ADC H	Direksiyon Açısı Max ADC L	Direksiyon Açısı Max ADC H	Direksiyon Açısı Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Direksiyon Açısı Min REAL H	Direksiyon Açısı Max REAL L	Direksiyon Açısı Max REAL H	Direksiyon Açısı ADC Katsayı	Direksiyon Açısı Offset	-	0x44	0x45	0x60	

Araç kontrol sistemi sol tekerlek açısı sensör kalibrasyon bilgilerini ‘0xB3’ komut numaralı paket ile göndermektedir. AKS tarafından gönderilen sol tekerlek açısı kalibrasyon paketi Tablo 3.11’de gösterilmektedir.

Tablo 3.11. AKS tarafından gönderilen sol tekerlek açısı kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xB3	10	Tekerlek Açısı - Sol Min ADC L	Tekerlek Açısı - Sol Min ADC H	Tekerlek Açısı - Sol Max ADC L	Tekerlek Açısı - Sol Max ADC H	Tekerlek Açısı - Sol Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Tekerlek Açısı - Sol REAL H	Tekerlek Açısı - Sol Max REAL L	Tekerlek Açısı - Sol Max REAL H	Tekerlek Açısı - Sol ADC Katsayı	Tekerlek Açısı - Sol Offset	-	0x44	0x45	0x60	

Araç kontrol sistemi sağ tekerlek açısı sensör kalibrasyon bilgilerini ‘0xB4’ komut numaralı paket ile göndermektedir. AKS tarafından gönderilen sağ tekerlek açısı kalibrasyon paketi Tablo 3.12’de gösterilmektedir.

Tablo 3.12. AKS tarafından gönderilen sağ tekerlek açısı kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xB4	10	Tekerlek Açısı - Sağ Min ADC L	Tekerlek Açısı - Sağ Min ADC H	Tekerlek Açısı - Sağ Max ADC L	Tekerlek Açısı - Sağ Max ADC H	Tekerlek Açısı - Sağ Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Tekerlek Açısı - Sağ Min REAL H	Tekerlek Açısı - Sağ Max REAL L	Tekerlek Açısı - Sağ Max REAL H	Tekerlek Açısı - Sağ ADC Katsayı	Tekerlek Açısı - Sağ Offset	-	0x44	0x45	0x60	

Araç kontrol sistemi motor pwm kalibrasyon bilgilerini '0xB5' komut numaralı paket ile göndermektedir. AKS tarafından gönderilen motor pwm kalibrasyon paketi Tablo 3.13'de gösterilmektedir.

Tablo 3.13. AKS tarafından gönderilen PWM kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xB5	8	Sağ Motor PWM Min Duty L	Sağ Motor PWM Min Duty H	Sağ Motor PWM Max Duty L	Sağ Motor PWM Max Duty H	Sol Motor PWM Min Duty L
V5	V6	V7	CRC	EOF1	EOF2	EOF3			
Sol Motor PWM Min Duty H	Sol Motor PWM Max Duty L	Sol Motor PWM Max Duty H	-	0x44	0x45	0x60			

Araç kontrol sistemi batarya gerilim sensör kalibrasyon bilgilerini '0xB6' komut numaralı paket ile göndermektedir. AKS tarafından gönderilen batarya gerilim sensörü kalibrasyon paketi Tablo 3.14'de gösterilmektedir.

Tablo 3.14. AKS tarafından gönderilen batarya gerilim kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xB6	10	Batarya Gerilim Min ADC L	Batarya Gerilim Min ADC H	Batarya Gerilim Max ADC L	Batarya Gerilim Max ADC H	Batarya Gerilim Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Batarya Gerilim Min REAL H	Batarya Gerilim Max REAL L	Batarya Gerilim Max REAL H	Batarya Gerilim ADC Katsayı	Batarya Gerilim Offset	-	0x44	0x45	0x60	

Araç kontrol sistemi batarya sıcaklık sensör kalibrasyon bilgilerini ‘0xB7’ komut numaralı paket ile göndermektedir. AKS tarafından gönderilen batarya sıcaklık sensörü kalibrasyon paketi Tablo3.15’de gösterilmektedir.

Tablo 3.15. AKS tarafından gönderilen batarya sıcaklık kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xB7	10	Batarya Sıcaklık Min ADC L	Batarya Sıcaklık Min ADC H	Batarya Sıcaklık Max ADC L	Batarya Sıcaklık Max ADC H	Batarya Sıcaklık Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Batarya Sıcaklık Min REAL H	Batarya Sıcaklık Max REAL L	Batarya Sıcaklık Max REAL H	Batarya Sıcaklık ADC Katsayı	Batarya Sıcaklık Gerilim Offset	-	0x44	0x45	0x60	

Arayüz tarafından araç kontrol sistemine yeni gaz pedalı kalibrasyon bilgileri ‘0xA1’ komut numaralı paket ile gönderilmektedir. Paket yapısı Tablo 3.16’da gösterilmektedir.

Tablo 3.16. Arayüz tarafından gönderilen gaz pedalı kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xA1	10	Gaz Pedalı Min ADC L	Gaz Pedalı Min ADC H	Gaz Pedalı Max ADC L	Gaz Pedalı Max ADC H	Gaz Pedalı Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Gaz Pedalı Min REAL H	Gaz Pedalı Max REAL L	Gaz Pedalı Max REAL H	Gaz Pedalı ADC Katsayı	Gaz Pedalı Offset	-	0x44	0x45	0x60	

Arayüz tarafından araç kontrol sistemine yeni direksiyon açısı kalibrasyon bilgileri ‘0xA2’ komut numaralı paket ile gönderilmektedir. Paket yapısı Tablo 3.17’de gösterilmektedir.

Tablo 3.17. Arayüz tarafından gönderilen direksiyon açısı kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xA2	10	Direksiyon Açısı Min ADC L	Direksiyon Açısı Min ADC H	Direksiyon Açısı Max ADC L	Direksiyon Açısı Max ADC H	Direksiyon Açısı Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Direksiyon Açısı Min REAL H	Direksiyon Açısı Max REAL L	Direksiyon Açısı Max REAL H	Direksiyon Açısı ADC Katsayı	Direksiyon Açısı Offset	-	0x44	0x45	0x60	

Arayüz tarafından araç kontrol sistemine yeni sol tekerlek açısı kalibrasyon bilgileri ‘0xA3’ komut numaralı paket ile gönderilmektedir. Paket yapısı Tablo 3.18’de gösterilmektedir.

Tablo 3.18. Arayüz tarafından gönderilen sol tekerlek açısı kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xA3	10	Tekerlek Açısı - Sol Min ADC L	Tekerlek Açısı - Sol Min ADC H	Tekerlek Açısı - Sol Max ADC L	Tekerlek Açısı - Sol Max ADC H	Tekerlek Açısı - Sol Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Tekerlek Açısı - Sol REAL H	Tekerlek Açısı - Sol Max REAL L	Tekerlek Açısı - Sol Max REAL H	Tekerlek Açısı - Sol ADC Katsayı	Tekerlek Açısı - Sol Offset	-	0x44	0x45	0x60	

Arayüz tarafından araç kontrol sistemine yeni sağ tekerlek açısı kalibrasyon bilgileri ‘0xA4’ komut numaralı paket ile gönderilmektedir. Paket yapısı Tablo 3.19’da gösterilmektedir.

Tablo 3.19. Arayüz tarafından gönderilen sağ tekerlek açısı kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xA4	10	Tekerlek Açısı - Sağ Min ADC L	Tekerlek Açısı - Sağ Min ADC H	Tekerlek Açısı - Sağ Max ADC L	Tekerlek Açısı - Sağ Max ADC H	Tekerlek Açısı - Sağ Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Tekerlek Açısı - Sağ Min REAL H	Tekerlek Açısı - Sağ Max REAL L	Tekerlek Açısı - Sağ Max REAL H	Tekerlek Açısı - Sağ ADC Katsayı	Tekerlek Açısı - Sağ Offset	-	0x44	0x45	0x60	

Arayüz tarafından araç kontrol sistemine yeni motor pwm kalibrasyon bilgileri ‘0xA5’ komut numaralı paket ile gönderilmektedir. Paket yapısı Tablo 3.20’de gösterilmektedir.

Tablo 3.20. Arayüz tarafından gönderilen PWM kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xA5	8	Sağ Motor PWM Min Duty L	Sağ Motor PWM Min Duty H	Sağ Motor PWM Max Duty L	Sağ Motor PWM Max Duty H	Sol Motor PWM Min Duty L
V5	V6	V7	CRC	EOF1	EOF2	EOF3			
Sol Motor PWM Min Duty H	Sol Motor PWM Max Duty L	Sol Motor PWM Max Duty H	-	0x44	0x45	0x60			

Arayüz tarafından araç kontrol sistemine yeni batarya gerilim kalibrasyon bilgileri ‘0xA6’ komut numaralı paket ile gönderilmektedir. Paket yapısı Tablo 3.21’de gösterilmektedir.

Tablo 3.21. Arayüz tarafından gönderilen batarya gerilim kalibrasyon paketi.

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xA6	10	Batarya Gerilim Min ADC L	Batarya Gerilim Min ADC H	Batarya Gerilim Max ADC L	Batarya Gerilim Max ADC H	Batarya Gerilim Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Batarya Gerilim Min REAL H	Batarya Gerilim Max REAL L	Batarya Gerilim Max REAL H	Batarya Gerilim ADC Katsayı	Batarya Gerilim Offset	-	0x44	0x45	0x60	

Arayüz tarafından araç kontrol sistemine yeni batarya sıcaklık kalibrasyon bilgileri '0xA7' komut numaralı paket ile gönderilmektedir. Paket yapısı Tablo 3.22'de gösterilmektedir.

Tablo 3.22. Arayüz tarafından gönderilen batarya sıcaklık kalibrasyon paketi

SOF1	SOF2	SOF3	KMT	UZNL	V0	V1	V2	V3	V4
0x59	0x45	0x44	0xA7	10	Batarya Sıcaklık Min ADC L	Batarya Sıcaklık Min ADC H	Batarya Sıcaklık Max ADC L	Batarya Sıcaklık Max ADC H	Batarya Sıcaklık Min REAL L
V5	V6	V7	V8	V9	CRC	EOF1	EOF2	EOF3	
Batarya Sıcaklık Min REAL H	Batarya Sıcaklık Max REAL L	Batarya Sıcaklık Max REAL H	Batarya Sıcaklık ADC Katsayı	Batarya Sıcaklık Gerilim Offset	-	0x44	0x45	0x60	

Gösterilen paket yapıları projeye özel olarak oluşturulmuş olup her paketin başlangıç, bitiş ve crc baytları olması sebebi ile paket içerisinde gelen verilerin hatalı alınmasının önüne geçilmiştir.

### 3.2.2. Arayüz yazılımı

Araç kontrol sisteminin giriş-çıkış kontrolü, parametre kontrolü ve ayarları, kaydı ve daha sonra analizinin yapılabilmesi işlemlerini bilgisayar arayüzü ile yapılabilmektedir. Arayüz Visual Studio 2017 geliştirme ortamında C# programlama dili ile programlanmıştır.

Arayüz tasarlanırken araç kontrol sistemi ile haberleşme yöntemi olarak seri haberleşme seçilmiştir. Seri haberleşme tx ve rx pinleri üzerinden veri gönderip alabilmektedir. Haberleşme hızı 115200 bit/s seçilmiştir.

Arayüz yazılımı temelde seri porttan veri alma, verilerin gösterilmesi, sensör kalibrasyonları, girişlerin test edilmesi ve analiz olmak üzere 5 temel yapıya ayrılabilir. Şekil 3.19'de bilgisayar arayüzü gösterilmektedir.



### 3.2.2.1. Seri porttan veri alma

Arayüz ile haberleşmeye başlamadan önce kartın bilgisayara bağlı olduğu COM portu ve baudrate seçilerek haberleşme konfigürasyonu yapılmalıdır. Ayarlar seçildikten sonra bağlan butonuna basılarak seri port veri alma ve göndermek için açılır. ‘Verileri Kaydet’ seçeneği seçildiğinde bağlantı açıldığı zamana ait bir kayıt dosyası oluşturularak gelen paketler kaydedilir. Seri haberleşme konfigürasyon seçenekleri Şekil 3.18’de gösterilmektedir.



Şekil 3.18. Arayüz bağlantı konfigürasyon seçenekleri.

Seri porttan veri alma işlemi ‘DataReceived’ event’ı üzerinden yapılır. Seri porttan veri geldiğinde programda ‘DataReceived’ kesmesi oluşur. Oluşturulan seri port okuma buffer’ına bu kesme içerisinde her gelen veri eklenerek devam eder. Belirli bir boyutu olan bu buffer’a veri yazma en sona geldiğinde veriler en baştan buffer’a yazılmaya devam eder. Bu yapıya dairesel buffer adı verilir. Her gelen veri bu buffer’a atılır. Buffer ile birlikte okuma ve yazma sayaçları da veri alırken tutulması gereken değerleri tutmalıdır. Veriler buffer’a atıldıktan sonra en son kalan indis yazma sayacına kaydedilir. Bir timer kullanılarak buffer da yeni gelen veri olup olmadığı kontrol edilir. Eğer yeni gelen veri var ise en son okunan indisten en son yazma indisine kadar olan kısımda paket yapımızın oluşup oluşmadığı aranır. Bütün veriler işlendikten sonra okuma sayacı en son okunan indis ile eşitlenir.

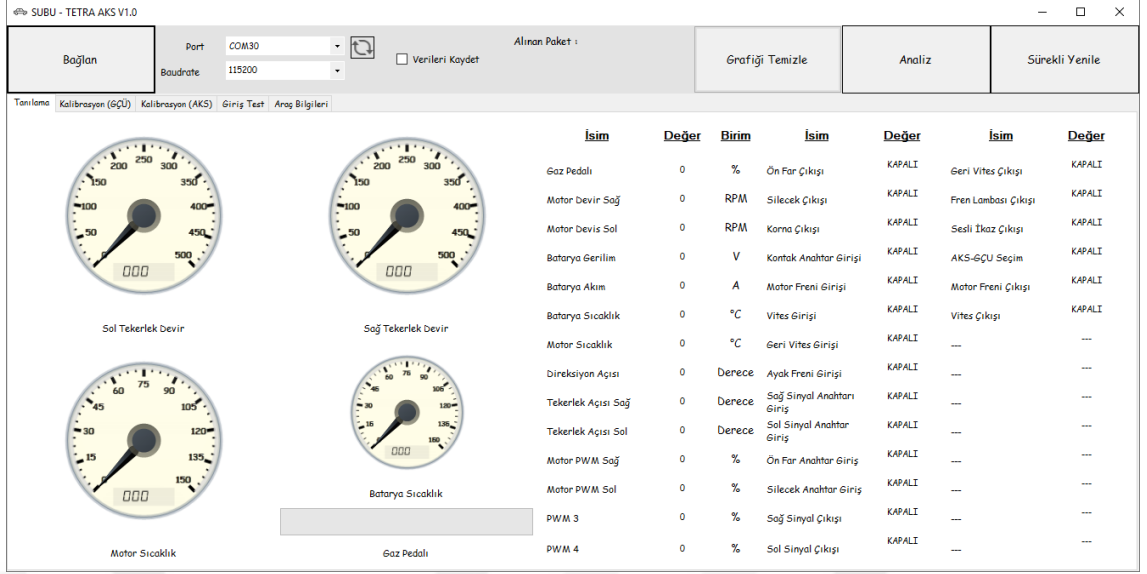
Veri yakalama işlemi öncelikli olarak alınan baytlardan ‘0x59’ bulunana kadar sırası ile sorgulanmaya başlanır. Değer bulunduktan sonra ‘0x45’ değeri aranır. Sonrasında ‘0x44’ aranır. ‘0x59’ değeri bulunduktan sonra diğer değerler hemen arkasından gelmezse tekrar başa dönülerek ‘0x59’ değeri aranmaya geri döner. Başlangıç baytları doğru geldiği durumda bir sonraki bayt paket komut numarasıdır. Bu değer ‘komut\_gelen’ isimli bir değişken içerisinde tutulabilir. Komut numarasından sonra paket içerisinde gelen veri

adedi bayt cinsinden gelir. Veri adedi bir deęişkende tutularak, bundan sonraki deęerlerin veriler isimli bir dizide tutulur. Doğru sayıda veri alındıktan sonra gelen deęer CRC deęeridir. CRC'ye kadar olan kısım toplanarak 0xFF ile maskelenir ve bu deęer gelen CRC deęerine eřit olup olmadıęı kontrol edilir. Eęer gelen CRC deęeri hesaplanan deęere eřit deęil ise bu gelen paketin hatalı olduęunu bize bildirir ve veriler iřlenmez. CRC hesaplanan deęer ile gelen deęer birbirine eřit ise paket bitiş baytlarının yakalanmasından sonra gelen paketin komut numarasına göre gelen verilerin ilgili deęişkenlere yazılması saęlanır.

Seri port baęlantısı yapıldıktan sonra AKS bilgisayara baęlanıldıęında “Verileri Kaydet” kutucuęunun yanında bulunan “Alınan Paket” yazısının yanında haberleşme durumu anlık olarak gözlemlenebilir.

#### **3.2.2.2. Verilerin gösterilmesi**

Alınan veriler ilgili deęişkenlere aktarılır. Form üzerinde bulunan göstergelerin güncellenmesi iřlemi 100ms'lik aralıklarla yapılmaktadır. Bu iřlem için oluřturulan timer nesnesi Şekil 3.19'da gösterilen “Sürekli Yenile” butonuna basıldıęında aktif edilir. Timer nesnesinin her timer\_tick olayında AKS'den gelen güncel verilere göre ‘Tanıla’ sekmesinde bulunan 4 adet döner gösterge ve labellar ilgili verilere göre güncellenmektedir.



Şekil 3.19. Bilgisayar arayüzü verilerin gösterilmesi.

### 3.2.2.3. Kalibrasyon sekmesi

Sistemin en önemli ihtiyaçlarından birisi olan sensör kalibrasyonu arayüz tarafından yapılabilmesi test ve yarışma esnasında kolaylık sağlamaktadır. Kalibrasyon işlemi analog sensörler için gerçekleştirilmektedir. Her sensör için ADC ve gerçek değerlerin girilebileceği alanlar mevcuttur. Kalibrasyon sekmeleri AKS ve GÇÜ sensörleri için farklı sekmelerdedir. Giriş-çıkış ünitesi kalibrasyon kısmı Şekil 3.20'da gösterilmiştir.

The calibration section is divided into eight panels, each corresponding to a specific sensor:

- Tekerlek Devir Sensörü - Sağ:** ADC Giriş Değeri, ADC 1, Gerçek Değer 1, ADC 1, Gerçek Değer 1, Gönder
- Gaz Pedali:** ADC Giriş Değeri, ADC 1, Gerçek Değer 1, ADC 1, Gerçek Değer 1, Gönder
- 5 Nolu ADC:** ADC Giriş Değeri, ADC 1, Gerçek Değer 1, ADC 1, Gerçek Değer 1, Gönder
- 7 Nolu ADC:** ADC Giriş Değeri, ADC 1, Gerçek Değer 1, ADC 1, Gerçek Değer 1, Gönder
- Tekerlek Dönüş Açısı:** ADC Giriş Değeri, ADC 1, Gerçek Değer 1, ADC 1, Gerçek Değer 1, Gönder
- Direksiyon Dönüş Açısı:** ADC Giriş Değeri, ADC 1, Gerçek Değer 1, ADC 1, Gerçek Değer 1, Gönder
- 6 Nolu ADC:** ADC Giriş Değeri, ADC 1, Gerçek Değer 1, ADC 1, Gerçek Değer 1, Gönder
- 8 Nolu ADC:** ADC Giriş Değeri, ADC 1, Gerçek Değer 1, ADC 1, Gerçek Değer 1, Gönder

Şekil 3.20. Kalibrasyon sekmesi GÇÜ.

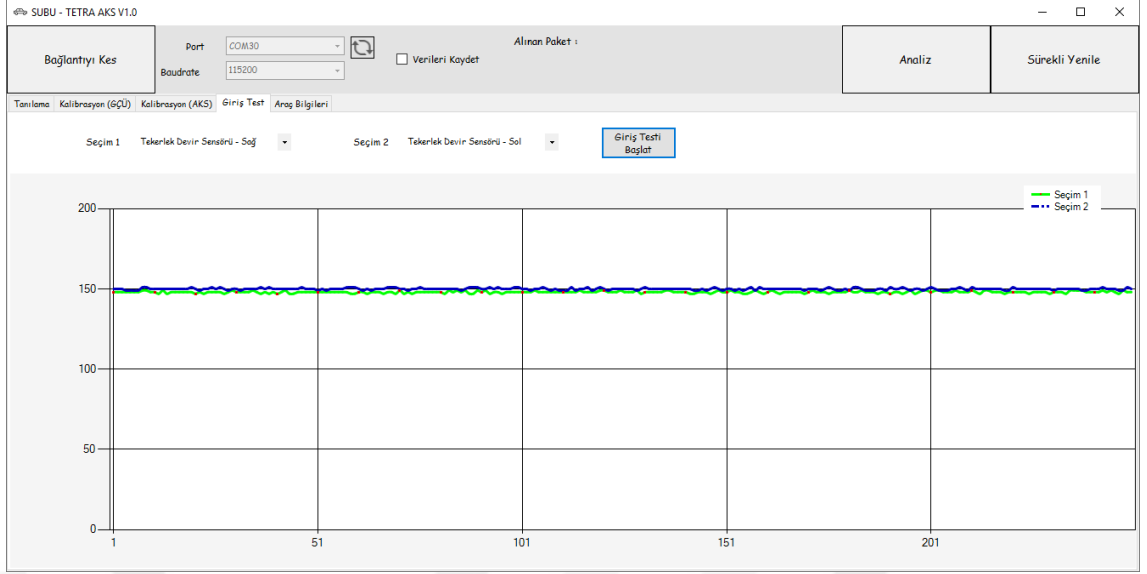
Araç kontrol sistemi sensörleri kalibrasyon sekmesi Şekil 3.21’de gösterilmektedir.

The screenshot displays the 'Kalibrasyon (AKS)' tab of the SUBU - TETRA AKS V1.0 software. The interface is organized into a top control bar and a main data entry area. The top bar includes a 'Bağlantıyı Kes' button, a 'Port' dropdown set to 'COM30', a 'Baudrate' dropdown set to '115200', a 'Verileri Kaydet' checkbox, and an 'Alınan Paket' field. On the right side of the top bar are 'Analiz' and 'Sürekli Yenile' buttons. The main area features a tabbed interface with 'Tanılama', 'Kalibrasyon (GÜ)', 'Kalibrasyon (AKS)', 'Giriş Testi', and 'Araç Bilgileri'. The 'Kalibrasyon (AKS)' tab is active, showing four columns of sensor calibration data. The columns are: 'Tekerlek Devir Sensörü - Sol', 'Motor Sürücü Sıcaklık Sensörü', 'Batarya Gerilim Sensörü', and four '.....ADC' sections. Each column contains three rows of input fields: 'ADC Giriş Değeri', 'ADC 1', and 'Gerçek Değer 1', followed by a 'Gönder' button.

Şekil 3.21. Kalibrasyon sekmesi AKS.

#### 3.2.2.4. Giriş testi

Sensörlerden gelen verilerin takibinin görsel olması hata bulma süresini kısaltır. Giriş test sekmesinde 2 ünitenin sensörleri de 2 adet açılır listede bulunmaktadır. İstenilen 1 veya 2 adet sensör seçilerek ekranda grafik şeklinde kullanıcının görmesi sağlanmaktadır. Sağ ve sol tekerlek devir sensörlerinden okunan devir bilgisi giriş test sekmesinde Şekil 3.22’de gösterilmektedir.



Şekil 3.22. Bilgisayar arayüzünde giriş test sekmesi.

### 3.2.2.5. Araç Bilgileri

Bilgisayar arayüzünde araç bilgileri sekmesinde aracın bilgileri sabit bir şekilde yer almaktadır. Aracın mekanik ve elektriksel verileri bu sekmede tablo haline getirilmiştir. Şekil 3.23'de bilgisayar arayüzünde aracın teknik bilgilerinin olduğu kısım gösterilmektedir.

Özellik	Değer	Birim
Üniversite Adı	Sakarya Uygulama Bilimleri Üniversitesi	
Takım Adı	SUBU TETRA	
Uzunluk	2714	mm
Genişlik	1215	mm
Yükseklik	1200	mm
Tekerlek Sayısı	4	adet
Gövde	Kompozit karbon fiber ve winlister reçine	malzeme
Ön Tekerlek Çapı	560	mm
Arka Tekerlek Çapı	560	mm
Motor	BLDC	-
Motor Sürücü	Hazır ürün	-
Motor Gücü	2	kW
Motor Ağırlığı	18	kg
Batarya Maksimum Voltaj	108	V
Batarya Nominal Voltaj	99	V
Batarya Nominal Güç	1980	Wh

Şekil 3.23. Bilgisayar arayüzünde aracın teknik bilgileri.

## BÖLÜM 4. ARAŞTIRMA BULGULARI VE TARTIŞMA

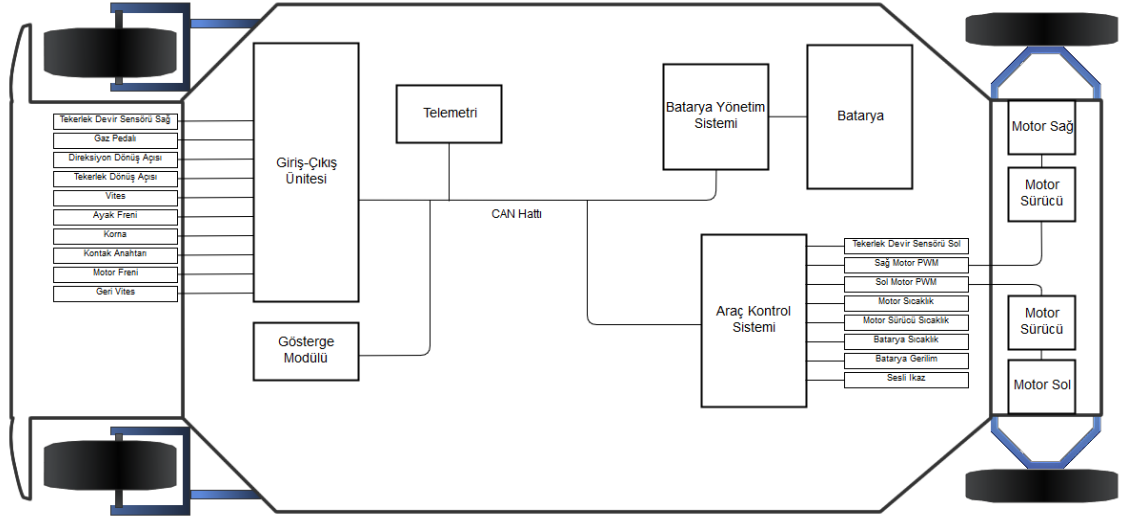
### 4.1. Test Aracı ve Sistem Bilgileri

Çalışmanın uygulanacağı Tübitak Efficiency Challenge yarışına katılacak olan SUBU-TETRA adlı elektrikli aracın genel görünümü Şekil 4.1’de gösterilmektedir.



Şekil 4.1. Test aracı genel görünüm.

Araç içerisinde, projede kullanılan malzemelerin yerleşim blok diyagramı Şekil 4.2’de gösterilmektedir. Aracın ön bölümünde bulunan giriş ve çıkışların kontrolü giriş-çıkış ünitesi tarafından yapılmıştır. Giriş-çıkış ünitesinin kontrolü tamamen araç kontrol sistemi tarafından yapılmıştır. Araç içinde bulunan donanım aralarında CANBUS haberleşme kullanarak haberleşmektedir.



Şekil 4.2. Sistemin blok diyagramı.

## 4.2. Araç Teknik Özellikleri

Araçta kullanılan üniteler ve kullanım adetleri Tablo 4.1’de gösterilmektedir.

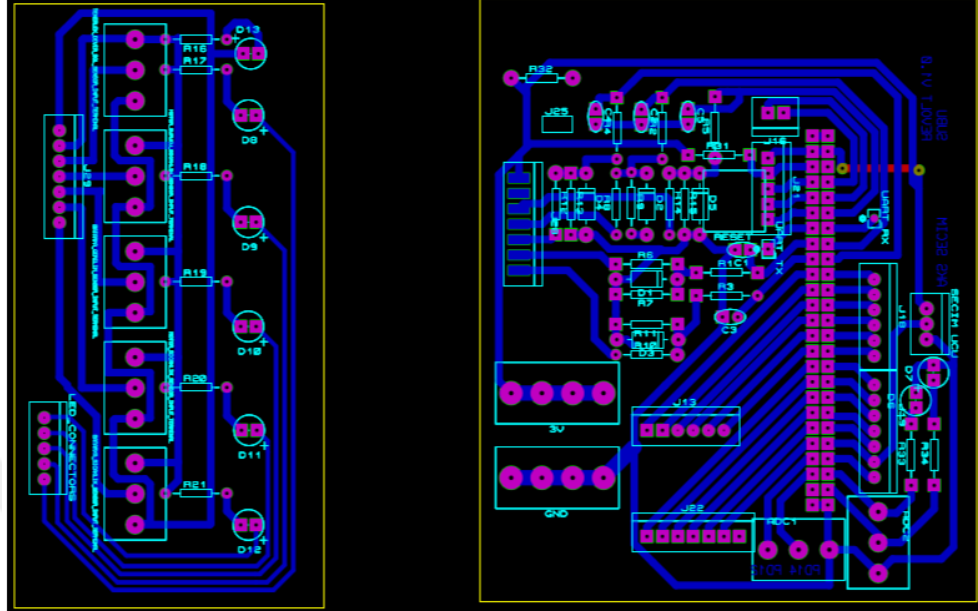
Tablo 4.1. Araçta kullanılan üniteler listesi.

Ünite Adı	Açıklama	Adet
Motor	2 Adet 72V 2kW’lık hub motor.	2
Motor Sürücü	2kW’lık motor sürücüsü	2
Batarya	72V’luk batarya paketi (20 seri 17 paralel paket)	1
Batarya Kontrol Kartı	72V’luk batarya kontrol kartı	1
Araç Kontrol Ssitemi	Geliştirilen Araç Kontrol Sistemi	1
Giriş Çıkış Ünitesi	Geliştirilen Giriş-Çıkış Ünitesi	1
Telemetri	SUBU öğrencileri tarafından geliştirilen Telemetri	1

## 4.3. Araç Kontrol Sistemi Bağlantı Devresi

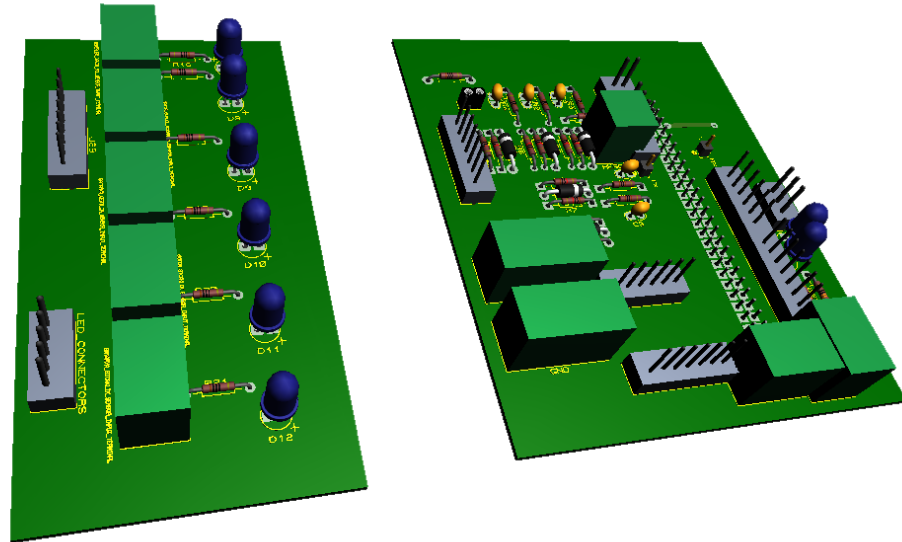
Araç kontrol sistemi ve giriş çıkış ünitesi araç içi bağlantılarının daha sağlıklı ve güvenilir yapılabilmesi için STM32F407 Discovery kartı üzerinde yerleştirilebilecek bağlantı devresi tasarımı Proteus devre çizim programı aracılığı ile çizilmiştir ve atölye ortamında

2 adet üretilmiştir. Bağlantı kartı için oluşturulan baskı devre çizimi görünüm Şekil 4.3’de gösterilmektedir.



Şekil 4.3. Bağlantı kartı baskı devre çizimi.

Bağlantı kartı devresi Şekil 4.4’ de gösterilmektedir.

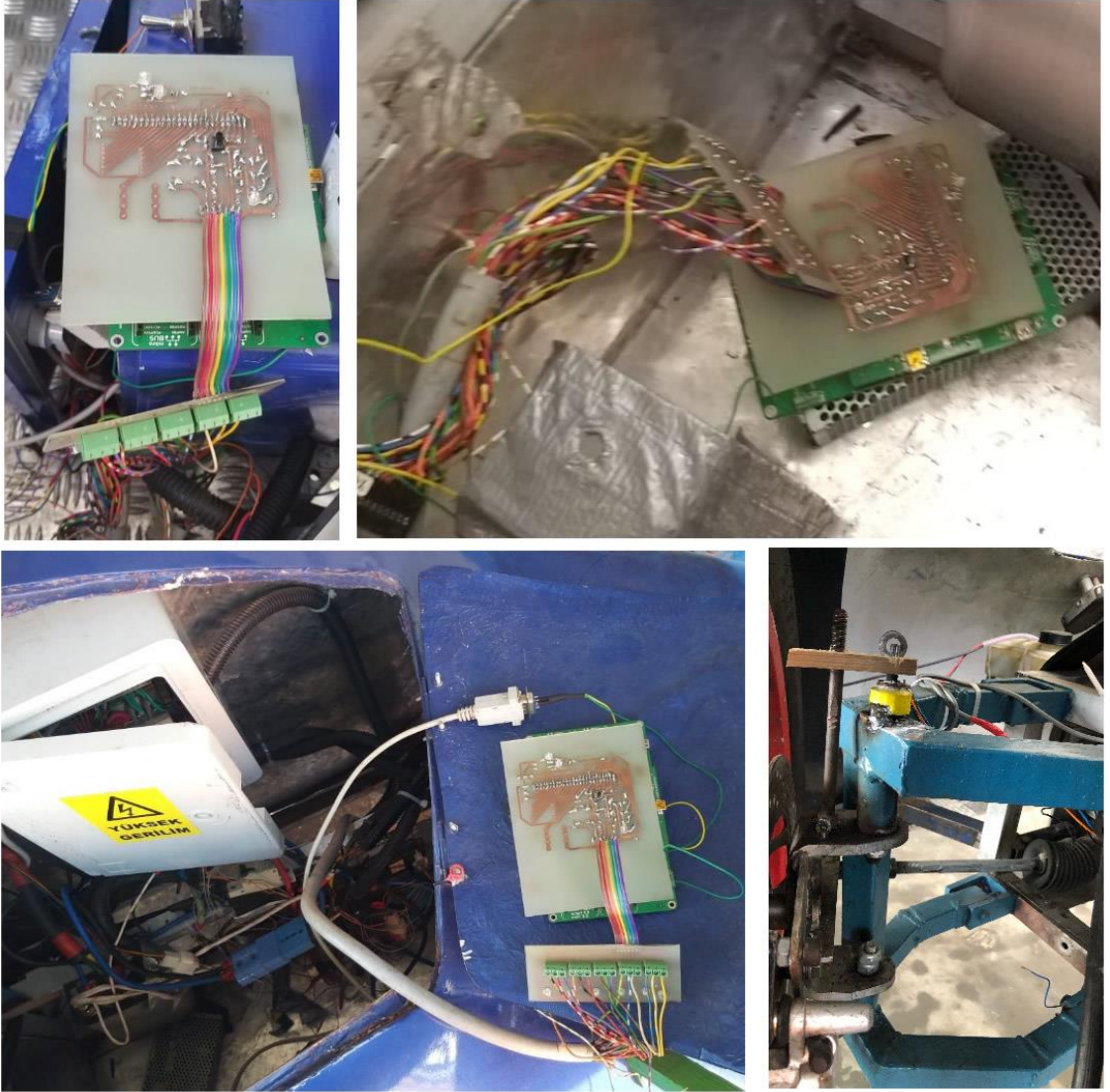


Şekil 4.4. Bağlantı kartı 3 boyutlu görünüm.



#### 4.4. Test Düzenegi

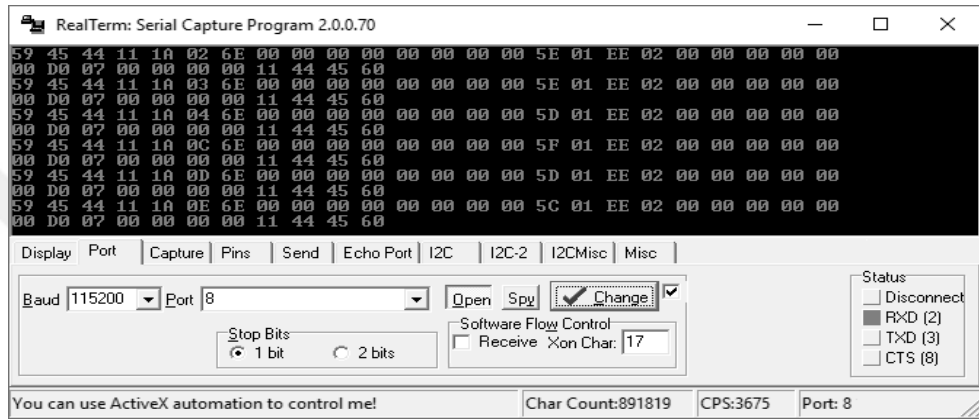
Araç sürüş testi Sakarya Uygulama Bilimleri Üniversitesi Serdivan Kampüsünde yapılmıştır. AKS ge GÇÜ donanımları üretilen bağlantı kartı ile birleştirilerek ve gerekli bağlantılar yapılarak sürüş testi yapılmıştır. Test düzenegi ile ilgili görseller Şekil 4.5’de gösterilmektedir. Araç içerisinde araç kontrol sistemi bağlantı kartı gösterimi sol üst kısımda, giriş-çıkış ünitesi yerleşimi sağ üst kısımda, araç kontrol sistemi bağlantı panosu sol alt kısımda ve tekerlek dönüş açısı bağlantı gösterimi sağ alt tarafta gösterilmektedir.



Şekil 4.5. Test aracı ve test kartı resimleri.

#### 4.5. Araç Kontrol Sistemi Kayıt Edilen Veriler

Araç kontrol sistemi verileri seri port üzerinden sürekli olarak göndermektedir. Araç kontrol sistemi tarafından gönderilen bu veriler bilgisayar ortamında arayüz tarafından alınarak kaydedilebilir ya da bağlantıların doğruluğunun kontrolü amacıyla herhangi bir seri port izleme aracı ile gözlemlenebilir. Şekil 4.6’da okunan veriler paketler halinde gösterilmiştir.



Şekil 4.6. Seri port okuma programı ile paketlerin ayrılmış gösterimi.

Araç kontrol sisteminin sürüş esnasında alınan verileri arayüz tarafından kayıt edilmektedir. Kayıt başlayabilmesi için ‘Bağlan’ butonuna basmadan önce ‘Verileri Kaydet’ seçeneği seçilmiştir. Bu kayıtlar bilgisayarda belirlenen bir klasör içerisinde bulunmaktadır. Kayıt edilen verilerin klasör içerisinde gösterimi Şekil 4.7 de gösterilmektedir.

Ad	Değiştirme tarihi	Tür	Boyut
2019_04_20_19_43_20.txt	29.04.2019 09:53	Metin Belgesi	56 KB
2019_04_20_20_34_40.txt	29.04.2019 09:53	Metin Belgesi	219 KB
2019_04_20_20_37_13.txt	29.04.2019 09:53	Metin Belgesi	4 KB
2019_04_20_20_37_27.txt	29.04.2019 09:53	Metin Belgesi	108 KB
2019_04_20_20_38_47.txt	29.04.2019 09:52	Metin Belgesi	5 KB

Şekil 4.7. Kayıtlı veriler.

Bilgisayar ile haberleşme esnasında alınan paketler gerekli kontroller yapıldıktan sonra eğer hatalı olarak gelmedi ise paket içeriğindeki veriler aralarında bir ayırıcı kullanılarak

satır satır yazılır. Şekil 4.8’de araç durma anında gelen paketlerin kayıt edilmiş hali gösterilmektedir.

```
7d$2$0$0$99$2$0$0$0$0$e6$1$ee$2$0$0$0$0$0$0$75$0$0$0$0$0$
82$2$0$0$9a$2$0$0$0$0$e6$1$ee$2$0$0$0$0$0$0$76$0$0$0$0$0$
83$2$0$0$9a$2$0$0$0$0$e6$1$ee$2$0$0$0$0$0$0$76$0$0$0$0$0$
88$2$0$0$99$2$0$0$0$0$e6$1$ee$2$0$0$0$0$0$0$75$0$0$0$0$0$
89$2$0$0$9a$2$0$0$0$0$e6$1$ee$2$0$0$0$0$0$0$74$0$0$0$0$0$
8d$2$0$0$9a$2$0$0$0$0$e6$1$ee$2$0$0$0$0$0$0$75$0$0$0$0$0$
94$2$0$0$99$2$0$0$0$0$e5$1$ee$2$0$0$0$0$0$0$74$0$0$0$0$0$
95$2$0$0$99$2$0$0$0$0$e6$1$ee$2$0$0$0$0$0$0$71$0$0$0$0$0$
96$2$0$0$9a$2$0$0$0$0$e6$1$ee$2$0$0$0$0$0$0$75$0$0$0$0$0$
99$2$0$0$9a$2$0$0$0$0$e6$1$ee$2$0$0$0$0$0$0$75$0$0$0$0$0$
9f$2$0$0$9a$2$0$0$0$0$e6$1$ee$2$0$0$0$0$0$0$75$0$0$0$0$0$
a1$2$0$0$9a$2$0$0$0$0$e5$1$ee$2$0$0$0$0$0$0$75$0$0$0$0$0$
a5$2$0$0$9a$2$0$0$0$0$e6$1$ee$2$0$0$0$0$0$0$74$0$0$0$0$0$
```

Şekil 4.8. Gelen paketlerin kayıt şekli.



Sürüş esnasında alınan bazı veriler Tablo 4.2’de gösterilmektedir

Tablo 4.2. Seri porttan alınan işlenmiş veri tablosu

	Zaman	Gaz Pedalı	Sağ Tekerlek	Direksiyon Açısı	Tekerlek Açısı	Batarya Sıcaklık
1	4860	551	155	1,8	0	32,6
2	4861	544	157	1,6	2	32,6
3	4865	538	157	1,3	2	32,6
4	4870	549	157,2	0,2	-2	32,6
5	4875	557	154	-0,1	5	32,6
6	4879	553	156	0,3	5	32,5
7	4884	557	155	1,1	6	32,6
8	4887	556	155	1,5	3	32,6
9	4890	558	157	1,8	5	32,7
10	4896	561	154	2,3	5	32,6
11	4902	560	156	2,3	-1	32,6
12	4903	559	155	2,3	1	32,6
13	4908	558	155	2,3	-5	32,4
14	4909	559	154	2,3	-2	32,7
15	4910	559	154	2,2	2	32,8
16	4914	558	154	2,1	-2	32,6
17	4915	559	153	2,1	5	32,6
18	4919	572	153	1,9	5	32,6
19	4921	569	153	1,8	6	32,7
20	4924	568	153	1,3	3	32,6
21	4927	579	152	0,7	7	32,6
22	4930	586	152	-0,2	5	32,7
23	4931	587	152	-0,5	-1	32,7
24	4936	578	151	-2,6	1	32,6
25	4942	580	151	-4,5	-5	32,6

#### 4.6. AKS ile GÇÜ Arası CANBUS Haberleşme Testi

Araç kontrol sistemi ile giriş-çıkış ünitesi arasında CANBUS haberleşmesi olduğu sürece iki ünite üzerinde de haberleşme gösterge ledi 1 saniyelik aralıklarla yanıp sönmektedir. Bağlantı kopması durumunda led 500 milisaniye aralıklarla yanıp sönmektedir. CANBUS haberleşmesi bus topolojisi kullandığı için hattaki verileri gözlemlemek amacı ile hatta ekleyeceğimiz bir cihaz haberleşmesinde sorun yaratmayacaktır.

CANBUS hattına bağlanan CAN-USB dönüştürücü ile hatta olan mesaj çerçevelerinin gözlemlenmesi sağlanmıştır. Şekil 4.9’da CAN hattında bulunan mesajlar gösterilmektedir.

CAN-ID	Length	Data	Cycle Time	Count
181010A2h	8	45 FF 02 FF F4 01 FF FF	10	3303
181011A2h	8	35 FF C2 01 FF FF FF FF	50	661
181011A1h	8	FF FF FF FF C2 01 FF FF	50	661
181012A1h	8	45 65 FF FF FF FF FF FF	100	331
181013A1h	8	40 59 FF FF FF FF FF FF	100	331
181014A1h	8	25 FF 26 FF FF FF FF FF	10	3303

Şekil 4.9. AKS ve Giriş-Çıkış ünitesi arasındaki CAN mesajları.



## **BÖLÜM 5. ÖNERİLER VE SONUÇ**

### **5.1. Sonuç**

Bu çalışmada elektrikli araç üzerinde kullanılacak, araç kontrol sistemi yazılımı geliştirilmiş ve haberleşme yapısı araçlarda standart haline gelen CANBUS haberleşmenin bir uygulaması olan J1939 protokolü yapısına uygun yapılmıştır. Arayüz tarafında araç kontrol sisteminin gönderdiği bütün verilerin kayıt edilmesi sağlanmıştır. Bu çalışmada sürüş testi esnasında gaz pedalı, direksiyon dönüş açısı, tekerlek dönüş açısı, tekerlek devir, batarya sıcaklık bilgilerine göre araç hızını kontrol eden duty cycle hesaplama işlemi yapılmıştır.

Sistemde AKS ve giriş-çıkış ünitesi olmak üzere araç hareket kontrolü için iki adet ünite haberleştirilmiştir. CANBUS haberleşme araç içerisinde yeni sensör, giriş veya çıkış ihtiyacı olması durumunda yeni bir ünite eklemeyi kolay hale getirmektedir. Sistemin bu özelliği gelecekte sistemin geliştirilmesi açısından önem kazanmaktadır.

Bilgisayar arayüzü aracılığı ile araç bilgileri bilgisayar üzerinde anlık olarak gösterilmektedir. Aynı zamanda bu veriler kayıt altına alınmaktadır. Kayıt altına alınan veriler daha sonra sürüş analizi yapmak için kullanılmaktadır. Bilgisayar arayüzü tarafından sensör kalibrasyon bilgilerinin görülmesi ve sensörlerin kalibrasyonunun yeniden yapılarak kayıt edilebilmesi sağlanmıştır.

Araca sistemin montajı yapılmış ve araç zeminden yukarı kaldırılarak giriş çıkışların kontrolü ve sensör kalibrasyon işlemleri yapılmıştır. Araç sürüş testine hazırlandıktan sonra yolda sürüş testlerine geçilmiştir.

Sonuç olarak araç kontrol sisteminin temel düzeyde çalışması sağlanmış ve aracın sürücü tarafından hatasız bir şekilde sürülmesi sağlanmış ve araç içi ünite haberleşmesinde

herhangi bir sorunla karşılaşılmamıştır. Aracın bilgilerinin gözlenmesi işlemi bilgisayar arayüzü tarafından yapılmıştır.

## **5.2. Öneriler**

Tasarlanan sistemde herhangi bir üniteye mikrodenetleyicinin kullanılmayan pinlerine yeni giriş ve çıkışlar eklenebilir. Yazılım mimarisi bu geliştirmeye açık olarak tasarlandığı ve yazılım kütüphanelerinin oluşturulmasında bu düşünce esas alındığından herhangi bir üniteye yeni sensör ve aktüatörler eklenmesi kolaylaşmıştır.

Araç kontrol sisteminde aracın sürüşü ile ilgili kontroller yapılmıştır. Projede aracın yazılımsal, donanımsal ve fonksiyonel kısımları geliştirmeye açıktır. Airbag, emniyet kemeri ikaz gibi sürücü güvenliği için gerekli olan fonksiyonlar eklenebileceği gibi otonom, yarı otonom araç gibi farklı konularda da proje gelişime açıktır.

Bilgisayar arayüzü mobil cihazlar için bir uygulama hazırlanarak bluetooth ya da internet üzerinden aynı işlemleri yapması sağlanılabilir. Web server oluşturularak herhangi bir cihazdan erişim işlemi de proje için geliştirmeye açık bir seçenektir.

## KAYNAKÇA

Azzeh, A. (2005). *CAN Control System for an Electric Vehicle*. Christchurch, New Zealand: University of Canterbury.

Bosch CAN Specification, Version 2.0. (1991). Stuttgart: Robert Bosch GmbH. <http://esd.cs.ucr.edu/webres/can20.pdf> adresinden alındı

Chakraborty, S., & Simões, M. G. (2013). *Power Electronics for*. (W. Kramer, Dü.) CO, Golden, USA: Springer. doi:10.1007/978-1-4471-5104-3

Chen , W., Round, S., & Duke, R. (2002). Design of an Auxiliary Power Distribution Network for an Electric Vehicl.. *1st IEEE International Workshop on Electronic Design, Test and Applications*, 257-261.

Cheng, A., Xiaoyu, L., Yongfu, L., & Wei, X. (2014, October). Design and application of saE J1939 communication protocol in electric vehicle. *Journal of Networks*, 2805-2812.

Didik, F. (2007, 03 17). *Electric Vehicle Companies*. [http://www.didik.com/ev\\_hist.htm](http://www.didik.com/ev_hist.htm) adresinden alındı

Dinçer, E. (2010, Şubat). *CANBUS ile Dağıtık Kontrol Uygulaması*. Niğde: Niğde Üniversitesi.

Fernandez, J. P., Vargas, M. A., Garcia, J. M., Carrillo, J. A., & Aguilar, J. J. (2019). *Low-Cost FPGA-Based Electronic Control Unit for Vehicle Control Systems*. Malaga: sensors Volume 19, Issue 8.

Instruments, T. (2018). C in Embedded Systems and the Microcontroller World, AN587.



- Karaca, A. (2003). *Denetleyici alan ağları kullanılarak (CAN) bina güvenlik uygulaması*. Sakarya: Sakarya Üniversitesi.
- Lee, K.-J., Kim, J.-W., Chang, H.-J., & Ahn, H.-S. ((2018)). MIXED HARMONIC RUNNABLE SCHEDULING FOR AUTOMOTIVE SOFTWARE ON MULTI-CORE PROCESSORS. *International Journal of Automotive Technology, Vol. 19*(No. 2), pp. 323–330. doi:DOI 10.1007/s12239–018–0031–2
- Li, Y., & Ji, X. (2013). Controller design for ISG hybrid electric vehicle based on SAE J1939 Protocol. *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)*. Paris, France: Atlantis Press.
- Lukasiewicz, M., Steinhorst, S., Andalarn, S., Sagstetter, F., Waszecki, P., Chang, W., . . . Mundhenk, P. (2013). *System Architecture and Software Design for Electric Vehicles*. Austin TX: DAC '13.
- SAE J1939 Standard. (2004). The Society of Automotive Engineers(SAE) Truck and Bus Control and Communications Subcommittee. SAE J1939 Standards Collection: <http://www.sae.org/products/j1939.htm> adresinden alındı
- Sperling, D. (2018). Electric vehicles: Approaching the tipping point. *BULLETIN OF THE ATOMIC SCIENTISTS*, 11-18.
- Tübitak. (2019, 03 11). *Kurallar*. Tübitak Efficiency Challenge Electric Vehicle: <https://challenge.tubitak.gov.tr/kurallar.html> adresinden alındı
- Uğur, E. (2011). *Prototip Bir Elektrikli Araç Üzerinde Enerji Yönetim Sisteminin Uygulanması*. İstanbul: Yıldız Teknik Üniversitesi.
- Uysal, A., & Soylu, E. (2017). *Embedded System Design and Implementation of an Intelligent Electronic Differential System for Electric Vehicles*. Karabük: International Journal of Advanced Computer Science and Applications.

Üstün, Ö., Tuncay, R. N., Mökükçü, M. S., Kıvanç, Ö. C., Tosun, G., Gökçe, C., & Çakan, M. (2017). *An integrated approach for the development of an electric vehicle powertrain: design, analysis, and implementation*. İstanbul: Turkish Journal of Electrical Engineering & Computer Sciences.

Xu, W., Chen, H., Zhao, H., & Ren, B. (2018). *Torque optimization control for electric vehicles with four in-wheel motors equipped with regenerative braking system*. Elsevier Ltd.



## ÖZGEÇMİŞ

Yunus Emre Durgun 08.01.1989 da İstanbul'da doğdu. İlk ve orta eğitimini Semiha Şakir İlköğretim Okulunda tamamladı. 2006 yılında Handan-Hayrettin Yelkikanat Anadolu Meslek Lisesinden mezun oldu. 2008 yılında başladığı Sakarya Üniversitesi Elektronik Öğretmenliği Bölümünü 2012 yılında tamamladı. 2014-2016 yılları arasında Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliğinde Mühendislik Tamamlama eğitimini tamamladı. 2013-2018 yılları arasında Rekarma Makine A.Ş firmasında araçüstü otomasyon projelerinde yazılım mühendisi olarak görev aldı. Şu anda Baykar Makine bünyesinde gömülü yazılım mühendisi olarak görev yapmaktadır.