

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

**YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**BATARYA YÖNETİM SİSTEMİNİN GELİŞTİRİLMESİ VE
CANBUS VERİ TRAFİĞİNİN İNCELENMESİ**

Muhammet Ahmet KOYUNCU

**Danışman
Dr. Öğr. Üyesi Kubilay TAŞDELEN**

ISPARTA - 2022



© 2022 [Muhammet Ahmet KOYUNCU]

İÇİNDEKİLER

	Sayfa
İÇİNDEKİLER	i
ÖZET	ii
ABSTRACT	iii
TEŞEKKÜR	iv
ŞEKİLLER DİZİNİ	v
ÇİZELGELER DİZİNİ	vii
SİMGELER VE KISALTMALAR DİZİNİ	viii
1. GİRİŞ	1
2. KAYNAK ÖZETLERİ	2
3. MATERYAL VE YÖNTEM	6
3.1. Batarya Yönetim Sistemi	6
3.1.1. Batarya yönetim sisteminin görevleri	6
3.1.1.1. Bataryayı izleme görevleri	7
3.1.1.2. Bataryayı koruma görevleri	7
3.1.1.3. Teşhis ve yaklaşım görevleri	8
3.1.1.3.1. Bataryanın şarj durumu (SoC)	8
3.1.1.3.2. Bataryanın sağlık durumu (SoH)	8
3.1.1.4. Şarj ve deşarj yönetimi	8
3.1.1.5. Batarya hücre gerilimlerinin dengelenmesi	9
3.2. Denetleyici Alan Ağı (CAN) Haberleşmesi	10
3.2.1. CAN topolojisi	12
3.2.2. CAN denetim mekanizması	12
3.2.3. CAN hızı	13
3.2.4. CAN mesaj çerçeveleri	13
3.2.4.1. Veri çerçevesi	14
3.2.4.2. Uzak çerçeve	16
3.2.4.3. Hata çerçevesi	17
3.2.4.5. Ayırıcı boşluk	17
3.2.5. CAN protokolü hata mekanizması	17
3.2.6. CAN protokolü hat yükü	19
3.3. Tasarlanan Deney Yüzeyi	19
3.3.1. CAN analizi	23
3.3.2. CAN bit doldurma işlemi	24
3.3.3. Batarya paketi ön şarj sistemi	25
3.3.4. Batarya paketi ısı sensorü	26
4. BULGULAR	29
4.1. CAN Hızı 125 kbit/s Analizi	30
4.2. CAN Hızı 500 kbit/s Analizi	40
4.3. CAN Hızı 1000 kbit/s Analizi	50
4.4. Tüm Hızlardaki Genel Analiz	60
5. TARTIŞMA VE SONUÇ	62
KAYNAKLAR	64
ÖZGEÇMİŞ	68

ÖZET

Yüksek Lisans Tezi

BATARYA YÖNETİM SİSTEMİNİN GELİŞTİRİLMESİ VE CANBUS VERİ TRAFİĞİNİN İNCELENMESİ

Muhammet Ahmet KOYUNCU

Isparta Uygulamalı Bilimler Üniversitesi
Lisansüstü Eğitim Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Kubilay TAŞDELEN

Son zamanlarda elektrikli araçların kullanımı ve üretimi artmaktadır. Bu artışla paralel olarak elektrikli araç teknolojileri sürekli olarak gelişmektedir. Elektrikli araçların kullanımlarının artmasıyla bataryaların şarj ve deşarj döngülerini kontrol eden BYS(Batarya Yönetim Sistemin)'nin de önemi artmakta ve verimlerinin artırılması için araştırmalar yapılmaktadır. Elektrikli araçlarda kullanılan enerji yoğunluğu yüksek bataryaların, sürekli olarak kontrol altında tutulmaları gerekmektedir. Elektrikli araçların ihtiyaç duyduğu yüksek gerilim ve kapasitenin elde edilmesi için çok fazla sayıda hücre kullanılmaktadır. Her bir hücrenin kontrol altında tutulması için batarya yönetim sistemine ihtiyacı vardır. BYS, araç ile haberleşmesinde CAN (Controller Area Network- Denetleyici Alan Ağı) kullanılır.

Bu tez çalışmasında, günümüz elektrikli araçlarında kullanılan BYS'nin CAN hattındaki veri trafiğinin analizi yapılmıştır. Yapılan bu analizde batarya paketinin anlık verileri CAN hattı üzerinden gönderilmektedir. Sistem içerisinde BYS, şarj cihazı ve VCU (Vehicle Control Unit- Araç Kontrol Ünitesi) bulunmaktadır.

BYS, VCU ve şarj cihazı düğümlerinin bulunduğu bir deney düzeneği hazırlanmıştır ve bu deney düzeneği üzerinde hat yükü hesaplaması yapılmıştır. Hat yükü hesaplama işleminde üç farklı yöntem kullanılmıştır ve bu yöntemlerin birbiri ile ilişkisi karşılaştırılmıştır. Test aşamalarında farklı CAN hızları kullanılmıştır ve mesaj gönderme sıklıkları değiştirilerek hat yükü analizi yapılmıştır. Hazırlanan deney düzeneğinden elde edilen verilere göre CAN hattındaki mesajların yükü %1.97 ile %19.58 arasında olduğu görülmüştür.

Anahtar Kelimeler: Denetleyici alan ağı, Batarya yönetim sistemi, Hat yükü analizi

2022, 68 sayfa

ABSTRACT

Master's Thesis

DEVELOPMENT OF BATTERY MANAGEMENT SYSTEM AND INVESTIGATION OF CANBUS DATA TRAFFIC

Muhammet Ahmet KOYUNCU

**Isparta University of Applied Sciences
The Institute of Graduate Education
Department of Electrical Electronics Engineering**

Supervisor: Asst. Prof. Dr. Kubilay TAŞDELEN

In these days, the usage and production of electric vehicles has been increasing. In parallel with this increase, electric vehicle technologies are developing constantly. With the increase in the usage of electric vehicles, the importance of BMS (Battery Management System), which controls the charge and discharge cycles of batteries, is increasing and researches are being carried out to increase their efficiency. Batteries with high energy density used in electric vehicles need to be kept under constant control. A large number of cells are used to obtain the high voltage and capacity required by electric vehicles. Each cell needs a battery management system to keep it under control. CAN (Controller Area Network) is used in BMS communication with the vehicle.

In this thesis, the data traffic on the CAN line of the BMS used in today's electric vehicles has been analyzed. In this analysis, instant data of the battery pack is sent over the CAN line. The system includes BMS, charger and VCU (Vehicle Control Unit).

An experimental setup with BMS, VCU and charger nodes was prepared and line load calculation was made on this experimental setup. Three different methods were used in the line load calculation process and the relationship of these methods with each other was compared. Different CAN speeds were used in the test stages and line load analysis was performed by changing the message sending frequencies. According to the data obtained from the prepared experimental setup, it was seen that the load of the messages on the CAN line was between 1.97% and 19.58%.

Key Words: Control area network, Battery management systems, Line load analysis

2022, 68 pages

TEŐEKKÜR

Tezimin y¼r¼t¼lmesinde desteęini ve emeęini hiębir zaman esirgemeyen tez danıŐmanım sayın Dr. Öğr. Üyesi Kubilay TAŐDELEN'e ęalıŐma s¼resince bana desteklerinden dolayı teŐekk¼rlerimi sunarım.

Tezimin her aŐamasında beni yalnız bırakmayan aileme sonsuz sevgi ve saygılarımı sunarım.

Muhammet Ahmet KOYUNCU
ISPARTA, 2022



ŞEKİLLER DİZİNİ

	Sayfa
Şekil 3.1. Lityum iyon pilin güvenli çalışma bölgesi.....	6
Şekil 3.2. LIR18650 pilin şarj ve deşarj eğrisi.....	9
Şekil 3.3. Hücre dengeleme yöntemleri	10
Şekil 3.4. CAN-H ve CAN-L hatlarına ait voltaj seviyeleri	11
Şekil 3.5. CAN protokolünde kullanılan topolojiler	12
Şekil 3.6. CAN hata çerçevesi biçimleri	17
Şekil 3.7. TEC ve REC sayaçlarının durumu.....	19
Şekil 3.8. CAN hattı üzerindeki bulunan düğümler.....	20
Şekil 3.9. Deney düzeneği.....	21
Şekil 3.10. BYS düğümünün gönderdiği mesajlar	22
Şekil 3.11. Şarj cihazı düğümünün gönderdiği mesaj.....	22
Şekil 3.12. VCU düğümünün gönderdiği mesajlar	23
Şekil 3.13. Vector VN1630A cihazı	23
Şekil 3.14. Düğümler ve mesaj kimlikleri	24
Şekil 3.15. Bit doldurma işlemi	25
Şekil 3.16. Ön şarj devresi	26
Şekil 3.17. Ön şarj işlem süreci.....	26
Şekil 3.18. Sıcaklık-Kapasite eğrisi	27
Şekil 3.19. Isı sensörü katsayıların belirlenmesi.....	27
Şekil 3.20. NTC ısı sensörü	28
Şekil 4.1. Deney düzeneği 125 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü	30
Şekil 4.2. Deney düzeneği 125 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri.....	31
Şekil 4.3. Simülasyon 125 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü ..	32
Şekil 4.4. Simülasyon 125 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri.....	33
Şekil 4.5. Deney düzeneği 125 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü ..	35
Şekil 4.6. Deney düzeneği 125 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri.....	36
Şekil 4.7. Simülasyon 125 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü	37
Şekil 4.8. Simülasyon 125 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri.....	38
Şekil 4.9. Deney düzeneği 500 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü	40
Şekil 4.10. Deney düzeneği 500 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri	41
Şekil 4.11. Simülasyon 500 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü	42
Şekil 4.12. Simülasyon 500 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri.....	43
Şekil 4.13. Deney düzeneği 500 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü	45
Şekil 4.14. Deney düzeneği 500 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri.....	46
Şekil 4.15. Simülasyon 500 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü	47
Şekil 4.16. Simülasyon 500 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri.....	48

Şekil 4.17. Deney düzeneği 1000 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü	50
Şekil 4.18. Deney düzeneği 1000 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri.....	51
Şekil 4.19. Simülasyon 1000 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü	52
Şekil 4.20. Simülasyon 1000 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri	53
Şekil 4.21. Deney düzeneği 1000 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü	55
Şekil 4.22. Deney düzeneği 1000 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri	56
Şekil 4.23. Simülasyon 1000 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü	57
Şekil 4.24. Simülasyon 1000 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri	58
Şekil 4.25. Toplam hat yükleri çubuk grafiği	61



ÇİZELGELER DİZİNİ

	Sayfa
Çizelge 3.1. Hat uzunluğuna bağlı hız değişimi	11
Çizelge 3.2. Çerçeve tipleri ve görevleri.....	14
Çizelge 3.3. Veri uzunluk kodu ve bayt sayısı.....	15
Çizelge 4.1. Hesaplama 125 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü	34
Çizelge 4.2. Hesaplama 125 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü.....	39
Çizelge 4.3. Hesaplama 500 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü	44
Çizelge 4.4. Hesaplama 500 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü.....	49
Çizelge 4.5. Hesaplama 1000 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü	54
Çizelge 4.6. Hesaplama 1000 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü.....	59
Çizelge 4.7. Toplam hat yükleri.....	60

SİMGELER VE KISALTMALAR DİZİNİ

ACK	Acknowledge
BYS	Batarya Yönetim Sistemi
CAN	Controller Area Network
CRC	Cyclic Redundancy Check
Kbit /s	Kilobit / saniye
LIN	Local Interconnect Network
ms	Milisaniye
NTC	Negative Temperature Coefficient
OSI	Open System Interconnect
REC	Receiver Error Counter
RTR	Remote Transmission Request
s	Saniye
SoC	State of Charge
SoH	State of Health
TEC	Transmitter Error Counter
V	Voltaj
VCU	Vehicle Control Unit
.dbc	Database file
Ω	ohm

1. GİRİŞ

Fosil yakıtlarla çalışan araçların yakıt rezervlerinin sınırlı olması ve bu araçların çevreye verdikleri zararlı gazlar sebebiyle elektrikli araçlara olan ilgi zamanla artmaktadır. Elektrikli araçların hareket edebilmek için ihtiyaç duyduğu güç elektrik motorları tarafından sağlanmaktadır. Elektrik motorları, elektrik enerjisini mekanik enerjiye çevirirler. Bu işlemi gerçekleştirirken gerekli olan enerjiyi batarya paketlerinden alırlar. Bundan dolayı elektrikli araçlarda kullanılan bataryalar araç için çok büyük önem taşımaktadır. Elektrikli araçların verimli ve güvenilir bir şekilde kullanılması bataryanın doğru bir şekilde kullanımıyla ilişkilidir. Dolayısıyla, daha verimli ve güvenilir bir çalışma için, BYS kullanılmaktadır. Batarya paketleri içerisinde seri bağlı hücreler arasında voltaj farkları oluşabilir. Bu koşullar altında BYS'nin bu voltaj farkını ortadan kaldırması beklenmektedir. Hücre dengeleme işlevi aktif ve pasif olmak üzere iki grupta incelenebilir. Otomotiv sektöründe kullanılan BYS tasarımında maliyetten dolayı pasif dengeleme tercih edilmektedir.

BYS, şarj edilebilir bir batarya paketinin veya hücrelerinin akım, gerilim, sıcaklık, yaşam döngüsü, kapasitesi gibi fiziksel parametrelerini izleyen ve bu değerleri kullanıcıya genelde CAN hattı üzerinden raporlayan cihazlardır. BYS, bir batarya paketinden aldıkları verileri genelde CAN hattı üzerinden diğer düğümlere gönderir. CAN protokolü 1983 yılında otomotiv sektöründe kullanılmak üzere Robert Bosch tarafından geliştirilmeye başlanmıştır ve resmi olarak 1986 yılında kullanılmaya başlanmıştır (Lawrenz, 1995).

Gelişen teknolojiyle beraber CAN hattına bağlanan cihazların sayısında artmaktadır. Bu cihazların artmasıyla beraber CAN hattındaki yoğunluğun artmasında beraberinde getirecektir. Bu cihazların gönderdiği mesajların CAN hattı üzerindeki etkisini görmek için hesaplama yapılmalı ve kontrol edilmelidir. BYS, CAN hattında bulunması gereken en önemli düğümlerden biridir. Bu düğümün diğer düğümlerdeki mesaj kimlikleri ile aynı olmaması gerekmektedir. Her cihazın bir kimliği bulunmaktadır ve verilerini bu kimlikle CAN hattına gönderirler. Bu tezdeki yapılmış olan çalışmayla bir BYS'nin CAN hattına olan yoğunluğu incelenmiştir.

2. KAYNAK ÖZETLERİ

Gerek batarya alanında yapılan çalışmalar gerekse de CAN hattı üzerinde yapılan çalışmalar gün geçtikçe artmaktadır. Tez süresi boyunca yapılan bu araştırmalar doğrultusunda gerekli bilgilere ulaşılmıştır.

Li vd. (2008) yaptıkları çalışmada, CAN hattının görüntülenmesi için bir cihaz geliştirmişlerdir. Bu cihaz ile uyumlu bilgisayar arayüzü yapmışlardır. Geliştirdikleri bu donanım USB portu sayesinde direkt olarak bilgisayara programı ile uyumludur. Ayrıca bu CAN hattındaki verileri kayıt ederek daha iyi analiz edilmesini sağlamışlardır. Yaptıkları çalışmaları test etmişler ve başarılı bir sonuç almışlardır.

Li vd. (2008) yapmış oldukları çalışmada, CAN hattının analizini yapmışlardır. Bu çalışmada bir karşılaştırma yapılmıştır. İki ayrı analiz aracını kullanarak bunlar arasındaki farklılıkları incelemişlerdir. Bu programlardan birisi ticari olarak da kullanılan CANoe aracıdır. Diğeri ise Matlab/Simulink 'in sunduğu analiz modelidir. Yaptığı testler sonucunda iki programın vermiş olduğu sonuçlar birbirine yakın olduğunu doğrulamıştır.

Aldoğan (2012) tez çalışmasında, BYS tasarımı yapmıştır. Yapmış olduğu bu tasarım CAN ve LIN haberleşme protokolüne sahiptir. Yaptığı bu donanım basit olarak denenmiş ve gerekli testleri yaparak başarılı bir şekilde çalıştığını göstermiştir.

Altun (2012) yaptığı BYS çalışmasında, hem ana kart için hem de slave kartlar için yazılım algoritmaları geliştirmiştir. Batarya paketinin daha verimli kullanılması için Enerji Koruma Modu algoritması geliştirmiştir. Yaptığı çalışmalar sonucunda tecrübelerini paylaşarak bir sonraki araştırmalarda kullanılmak için eksikliklerini söylemiştir.

Al Hayani (2012) yapmış olduğu tezinde, J1939 protokolüne dayanan Scania CAN veriyolu ağlarında, hattın yükünü, verimini, ağların arıza noktalarını ve durum gecikmesini hesaplayan, modelleyen bir kullanıcı arayüzü geliştirmişlerdir.

Nafiz (2014) tez çalışmasında, elektrikli otobüslerde kullanılacak BYS geliştirmiştir. Batarya sayısının değişimine uyum sağlayacak olan bu BYS, bataryalardan aldığı sıcaklık, gerilim ve akım gibi bilgileri RS232, RS485 ve CANBus ile rapor edebilmektedir. Bataryaları ise aktif şarj dengeleme metodu ile dengelemiştir. Yaptığı çalışmasını yüksek kapasiteli piller ile test etmiştir.

Kelkar ve Kamal (2014) yapmış oldukları çalışmada, uyarlanabilir hata teşhis algoritması (AFDCAN) adlı yeni bir algoritma önermektedirler. Düşük maliyetli kaynak kısıtlı dağıtılmış gömülü sistemler için tasarlanmıştır. Önerilen bu algoritma, CAN üzerindeki tüm hatalı düğümleri tespit eder. Bir teşhis döngüsü sırasında yeni düğüm girişine ve onarılan hatalı düğümlerin yeniden girişine izin verir. AFDCAN' in yüksek hata toleransı sağladığı ve güvenilir iletişim sağladığı bulunmuştur.

Murway ve Groza (2014) yaptıkları çalışmada, CAN hattındaki veriyolunun her düğüm tarafından yerleştirilen çerçevelerin benzersiz fiziksel özelliklerine kadar kimlik doğrulaması yapmaya çalışmışlardır. Bunun için her potansiyel göndericiyi benzersiz bir şekilde tanımlamak için voltaj ölçümleri olarak, sinyali filtreleyerek, ortalama kare hatalarını ve konvolüsyonları inceleyerek çerçeveleri analiz ettiler. Deneysel sonuçları, belirli düğümler arasında ayırım yapmanın açıkça mümkün olduğunu ve alıcı-vericilerin ve çerçeve kimliklerinin akıllıca seçimleriyle her mesajın tam olarak göndericisine bağlanabileceğini göstermişlerdir.

Yılmaz (2017) yapmış olduğu yüksek lisans tezinde, tasarladığı batarya yönetim sistemine bluetooth ve Wifi özelliği ekleyerek geliştirdiği sistemin internete bağlanmasını sağlamıştır.

Gül (2018) yaptığı tez çalışmasında, elektrikli araçlarda kullanıma uygun olmak üzere bir batarya yönetim sistemi geliştirmiştir. Geliştirdiği sistem master-slave topolojisine dayanmaktadır. 12 adet lityum iyon hücreden oluşan bir batarya paketinde testlerini gerçekleştirmiştir. Batarya paketinden gerekli bilgileri almak için Linear Technology firmasının ürettiği LTC6804 entegre devresini kullanmıştır. Bu sayede ölçüm sonuçları olarak en doğru yaklaşımlar yapılmıştır.

Yavuz vd. (2018), SocketCAN ile bir CAN ağı tasarlanmıştır. Tasarlanan bu sistemde Rate Monotonic (RM) Zamanlama Algoritması kullanılmıştır. Böylelikle optimum statik mesaj zamanlaması gerçekleştirilmiştir. Bu RM zamanlama politikasına göre, iletilecek olan mesajların öncelikleri iletim başlamadan önce belirlenir ve herhangi bir mesajın son teslim süresi, periyoduna eşittir. Yaptıkları testlerden sonra mesajların iletimlerinde herhangi bir bozulmanın olmadığını kanıtlamışlardır.

Tosun (2019) tez çalışmasında, bir BYS geliştirmiş ve bu sistemi elektrik bisiklet batarya paketine uygulamıştır. Bu batarya paketinin şarj ve deşarj performanslarını hesaplamıştır. Geliştirmiş olduğu bu sistemin koruma algoritmalarını hazırlamıştır ve yaptığı testler sonucunda batarya paketinin yeterli bir şekilde çalıştığını göstermiştir.

Rohrer vd. (2019) çalışmasında, traktörler için gerçek zamanlı bir CAN hattı veri analizi yapmıştır. Öncelikle diğer programlar ile verileri analiz etmiştir. Bu programlar Kvaser CanKing ve Vector CANalyzer programlarıdır. Analiz ettiği bu araçlardan sonra kendisi basit yapıda bir analiz programı geliştirmiştir.

Baygüneş (2019) yapmış olduğu bir tez çalışmasında, yirmi adet seri bağlı bir batarya hücresinin kontrolünü sağlamak amacıyla BYS tasarlamıştır. Bataryalarda pasif dengeleme yöntemi kullanmıştır. Tasarladığı sistemde bataryanın sağlık durumu (SOH) kestirimi yapmıştır. Tasarım testlerden geçirilerek etkin bir şekilde çalıştığı ortaya koymuştur.

Ekici (2019) yapmış olduğu bir yüksek lisans tezinde, farklı kimyaya sahip olan ve bir elektrikli araç kullanarak BYS tasarımı yapmıştır. Geliştirdiği sistemin şarj ve deşarj işlemlerini yapmıştır. Bu çalışma sonucunda elektrik araçlarda farklı kimyasal özelliklere sahip pillerin bir batarya yönetim sistemi ile kontrol edilmesi üzerinde çalışılmıştır.

Aktaş vd. (2020) yapmış olduğu araştırma makalesinde, elektrikli araçlar için düşük üretim maliyetli bir batarya yönetim sistemi tasarımı gerçekleştirilmiştir. Bu çalışmada ana işlemci olarak STM32F103C8 serisi kullanılmıştır. Sistem tasarımında lityum iyon piller kullanılarak yirmi adet seri bağlı batarya hücresi kullanılmıştır. Bu pillerin

dengelemesi pasif yöntem kullanılarak yapılmıştır. Ayrıca pillerin SoC ve SoH hesaplamasını yapmışlardır.

Kıvrak vd. (2020) yapmış olduğu araştırma makalesinde, dört hücreli bir batarya paketinin pasif dengeleme metodu kullanılarak bir BYS tasarlamışlardır. Bu BYS master-slave mantığına dayalı olarak geliştirilmiştir. Slave kartlar pillerden ölçtükleri verileri master karta CAN hattı üzerinden göndermiştir.

Bozdal vd. (2020) yapmış oldukları araştırma makalesinde, modern araçlarda CAN protokolü siber saldırıların karmaşık ve gelişen doğasına karşı iyi korunmamış olduğunu ve bunun şifreleme eksikliğinden kaynaklandığını söylemiştir. Buna karşılık giriş tespit sistemi (IDS) ile tam olarak güvenlik sağlamasa bile kabul edilebilir bir güvenlik yöntemi olduğunu göstermiştir.

Baki (2021) yaptığı tez çalışmasında, korsanların araç kontrol ünitelerini kolayca ele geçirebileceğini ve bunun sonucunda meydana gelebilecek risklerin ciddi olabileceğini savunmuştur. Buna önlem almak amacıyla bir Korsan Belirleme Ünitesi (KBÜ) kurmuştur. Bu sistem korsan varlığını belirleyip elektronik kontrol ünitelerinin birden fazla yoldan basit şifreli olarak haberleşmesi sağlanarak saldırıları savuşturmayı başarmıştır.

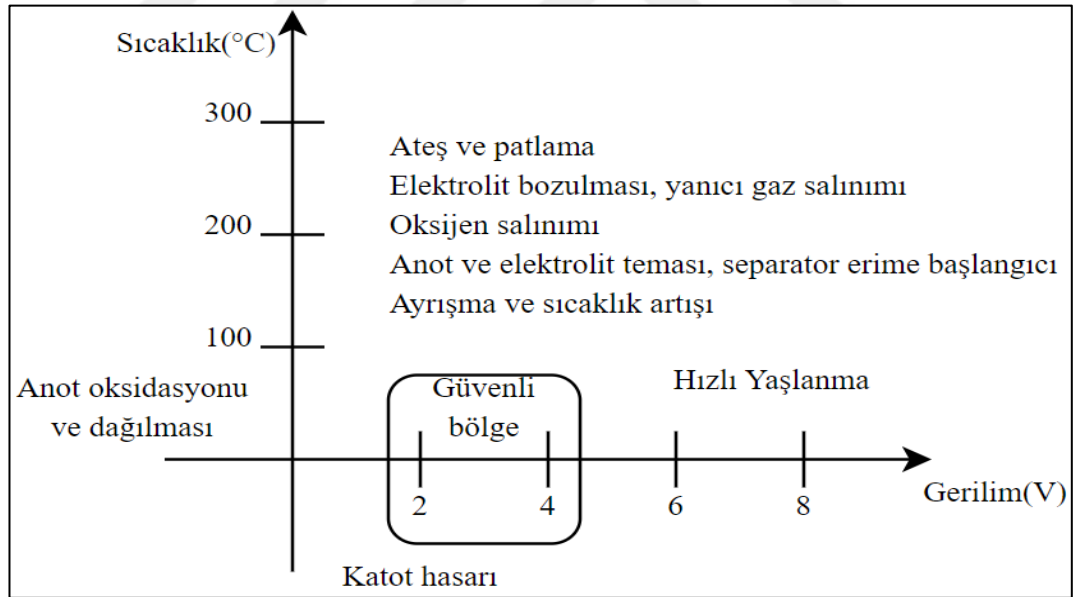
3. MATERYAL VE YÖNTEM

3.1. Batarya Yönetim Sistemi

Batarya yönetim sistemleri, batarya paketinin denetlenmesi, şarj ve deşarj işlemi esnasında gerilim, akım ve sıcaklık gibi önemli parametrelerinin kontrol altında tutulmasıdır.

Lityum iyon hücreler, yerine geçtikleri batarya teknolojilerine göre daha yüksek performanslar sunsalar da güvenlik dezavantajları vardır. Bu hücreler uygun çalışma bölgelerinde kullanılmazlarsa, kalıcı performans kaybı yaşayabilirler, ısınma ve patlama gibi ciddi güvenlik riskleri oluşturabilirler.

Şekil 3.1’de tipik bir lityum iyon hücrenin güvenli çalışma bölgesi görülmektedir. Bataryadaki tüm hücrelerin bu bölgede tutulması ise BYS’nin görevidir (Gül, 2018).



Şekil 3.1. Lityum iyon pilin güvenli çalışma bölgesi (Lu vd., 2012)

3.1.1. Batarya yönetim sisteminin görevleri

Modern batarya yönetim sistemleri, hücre sayısına ve kullanıldığı alana göre onlarca farklı görevi yerine getirmelidir. Elektrikli araçlarda kullanılan BYS’ler ise

teknolojinin tüm imkânlarından yararlanan, araç içerisindeki diğer elektronik kontrol birimlerine benzer şekilde birçok farklı görevi yerine getiren sistemlerdir (Gül, 2018).

3.1.1.1. Bataryayı izleme görevleri

Bir batarya yönetim sisteminin tüm görevlerinin başında bataryayı izleme görevleri gelir. BYS yaptığı ölçümler ile diğer görevlerini yürütebilir. Modern bir BYS'nin batarya içerisindeki yaptığı ölçümler aşağıda listelenmiştir.

- Tüm hücrelerin gerilimleri
- Batarya paketinin toplam gerilimi
- Hücre sıcaklıkları
- Bataryanın izolasyon direncinin değeri
- Hücreler arası haberleşmenin durumunu
- Konnektörlerin ve kapakların hangi konumda olduğu

3.1.1.2. Bataryayı koruma görevleri

Batarya yönetim sisteminin görevlerinden birisi olan koruma görevi, çeşitli korumaları içerir. Bunlar akım koruması ve gerilim koruması olarak belirtilebilir. Akım koruma fonksiyonu olarak, batarya hücresine takılan veya kendi içerisinde barındıran bir akım sensörü ile sağlanır. Batarya yönetim sistemi batarya paketinin ne kadar akım çekeceği veya ne kadar akım verebileceğini bilir. Bu sayede akım sensöründen alınan bilgiye göre fazla şarj veya deşarj durumlarında batarya paketinin enerjisini kapatır.

BYS'nin en önemli görevlerinden biri de hücrelerin sıcaklıklarını ölçmesidir. Devrede herhangi bir algılayıcının sıcaklık değeri, sıcaklık sınırını aşarsa BYS, soğutmak için fan sistemi açılacaktır. Açılacak fan sistemi tarafından sıcaklığın daha çok yükselmesinin önüne geçmek istenilmektedir. Fan sisteminin devreye girmesine rağmen sıcaklık yükselmesinin önüne geçilemezse, BYS acil röle açma sinyali göndererek şarj/deşarj işlemini durduracak ve hücre gerilim dengelenmesi işlemine son verecektir (Baygüneş, 2019).

3.1.1.3. Teşhis ve yaklaşım görevleri

Batarya yönetim sistemi ölçtüğü gerilim, akım ve sıcaklık verilerinden yola çıkarak batarya hakkında bazı yaklaşımlar yapmak zorundadır. Bunlar bataryanın sağlık durumu (SoH) ve batarya paketinin şarj durumudur (SoC).

3.1.1.3.1. Bataryanın şarj durumu (SoC)

Bataryanın şarj durumu basit bir ifadeyle, bataryanın içinde olan enerji miktarının toplam kullanılabilir enerji miktarına oranıdır. SoC, bataryadan doğrudan ölçülebilecek bir parametre değildir bu yüzden batarya gerilimi gibi ölçülebilen parametreler ile de doğrudan ilişkili değildir. Bu sebeple elde edilen veriler ile bir kestirim yapmak gerekmektedir.

SoC'yi hesaplamak karmaşık bir durum ve gelişmiş algoritmalar gerektirir. Kaynaklara göre SoC tahmin yöntemlerinin sınıflandırılması farklılık göstermektedir. Bir kaynağa göre SoC hesaplama yöntemi dört bölüme ayrılmıştır. Bu yöntemler doğrudan ölçüm, defter tutma, uyarlamalı sistemler ve melez yöntemlerdir (Prajapati vd., 2011).

3.1.1.3.2. Bataryanın sağlık durumu (SoH)

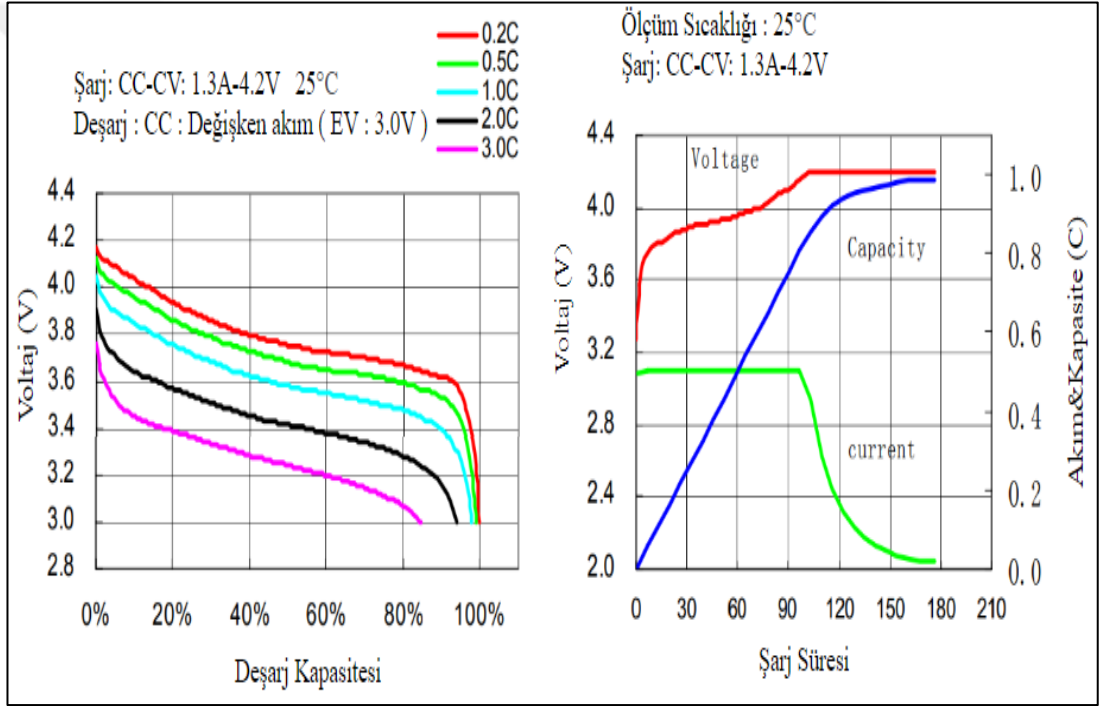
Bataryanın sağlık durumu, batarya yönetim sisteminin göstermesi gereken başka bir önemli ölçüttür. Bataryanın sağlık durumu, SoC değeri gibi doğrudan bataryanın terminal uçlarından ölçülebilen bir parametre değildir. Elektrikli araçlarda pilin sağlık durumu çok önemlidir ve bu SoH tahmini için hala çalışmalar devam etmektedir (Rahimi-Eichi vd., 2013). SoH, pilin kullanılmaya başladığı zamandan, yeni bir pile göre durumunun bir göstergesidir.

3.1.1.4. Şarj ve deşarj yönetimi

Şarj sırasında batarya yönetim sistemi şarj cihazı ile haberleşir ve gerekli voltaj ve akım bilgisini cihaza genellikle CAN hattı ile gönderir. Böylelikle şarj cihazının buna uygun olarak batarya paketini şarj eder. Deşarj işlemi ise batarya yönetim sistemi

motor sürücüsüne pilin güvenli sınırlar içinde ne kadar akım verebileceğini bildirir ve motor sürücüsü de buna uygun olarak batarya paketini kullanır. Tüm bu işlemleri güvenli bir şekilde yapmak batarya yönetim sisteminin görevidir.

Şekil 3.2’de bir LIR18650 pilin şarj ve deşarj eğrisi gösterilmektedir. Batarya yönetim sistemi en kötü senaryolarda bir tepki verebilmelidir. Bu senaryolar çeşitli şekilde olabilir. Örnek olarak, batarya yönetim sistemi şarj işlemi sırasında istediği akımı şarj cihazına gönderir ve eğer şarj cihazı bu akım sınırlarına uymazsa, pilin güvenliğini tehlikeye atacak bir şekilde şarj ediliyorsa batarya yönetim sistemi bu durumu kontaktörlerini açarak şarj cihazı ile olan bağlantısını kapatması gerekmektedir.

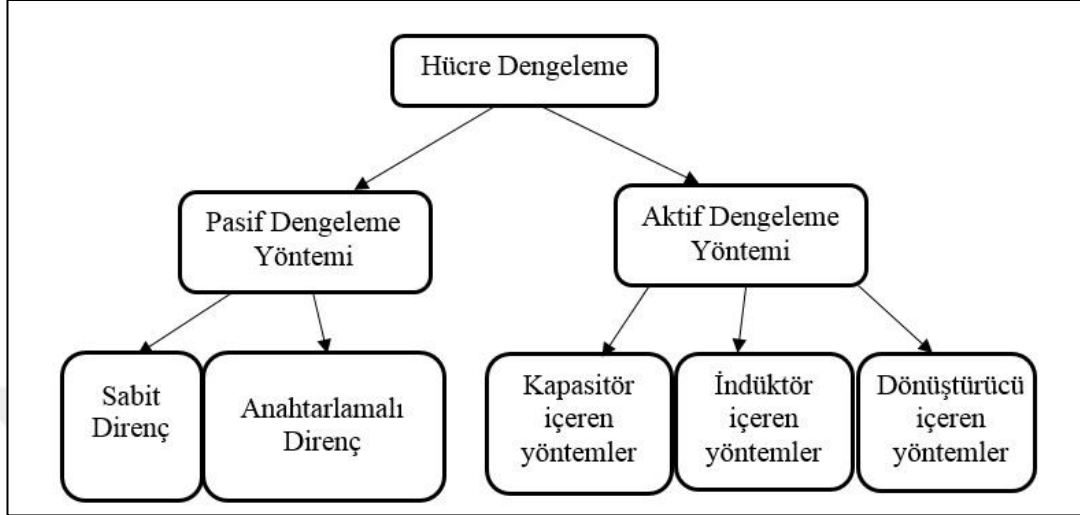


Şekil 3.2. LIR18650 pilin şarj ve deşarj eğrisi (Anonim, 2010)

3.1.1.5. Batarya hücre gerilimlerinin dengelenmesi

Batarya paketini oluşturan hücreler her zaman özdeş değildir. Bunun sebepleri arasında üretim tarihi, pilin iç kimyası gibi etmenler oluşturmaktadır. Bu sebeplerden dolayı batarya yönetim sisteminin en önemli görevlerden birisi de bu gerilim farklılıklarını ortadan kaldırmaktır (Daowd vd., 2011).

Batarya yönetim sistemi tasarımlarında hücre dengelemesi yapılması için donanımsal tasarımlar yapılır ve bazı yöntemler geliştirilmiştir. Şekil 3.3'te hücre dengeleme yöntemleri gösterilmiştir.



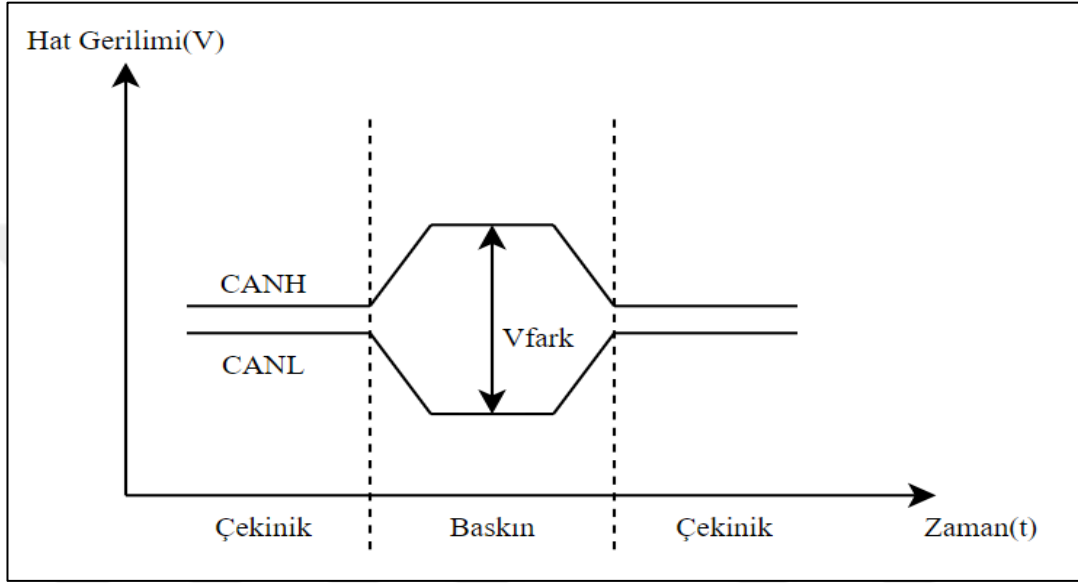
Şekil 3.3. Hücre dengeleme yöntemleri

3.2. Denetleyici Alan Ağı (CAN) Haberleşmesi

Denetleyici Alan Ağı (CAN) haberleşmesi, Bosch firması tarafından otomotiv sektöründeki haberleşme altyapısı ihtiyaçlarını karşılamak üzere 1986 yılında ortaya çıkarılmıştır. 1993 yılında ISO protokolü uluslararası bir standart olarak kabul edilmiştir (Kiencke, 1994). Protokolün kabul edilmesinden sonra Intel ve Philips firmaları tarafından ilk CAN çipleri üretilerek endüstride kullanılmaya başlanmıştır. 1991 yılı itibari ile Mercedes firması CAN haberleşme protokolünü seri üretilen araçlarında kullanan ilk otomotiv firması olmuştur (Anonim, 2011).

CAN standart protokolü, OSI (Open System Interconnect – Açık Sistem Bağlantısı) referans modelinin 1. ve 2. katmanlarına karşılık gelmektedir. Her ne kadar başlangıçta yalnızca otomotiv uygulamaları için geliştirilmiş olsa da küçük boyut, düşük maliyet, yüksek güvenilirlik ve yüksek hız gibi özelliklerinden dolayı birçok dağıtık endüstriyel kontrol uygulamalarında yaygın olarak kullanılmaktadır. Güvenliğin çok önemli olduğu gerçek zamanlı uygulamalarda da kullanılır. Çünkü yapılan istatistiksel olasılık hesapları doğrultusunda bir asırda bir tane tespit edilemeyen mesaj hatası yapabileceği tespit edilmiştir (Lawrenz, 1995; Özçelik vd., 2001).

Şekil 3.4'te CAN-H ve CAN-L hatlarına ait voltaj seviyeleri gösterilmiştir. Buna şekile göre iki hat arasındaki voltaj farkı (CAN-High – CAN-Low) belli bir eşik değerin üzerinde ise bu zaman dilimindeki sinyal baskın olarak adlandırılır ve dijital olarak değeri lojik 0'dır. Bunun haricinde eğer iki hat arasındaki voltaj farkı belli bir eşik değerin altında ise, bu zaman dilimindeki sinyal çekinik olarak tanımlanır ve dijital değeri lojik 1'dir.



Şekil 3.4. CAN-H ve CAN-L hatlarına ait voltaj seviyeleri (Anonim, 2015)

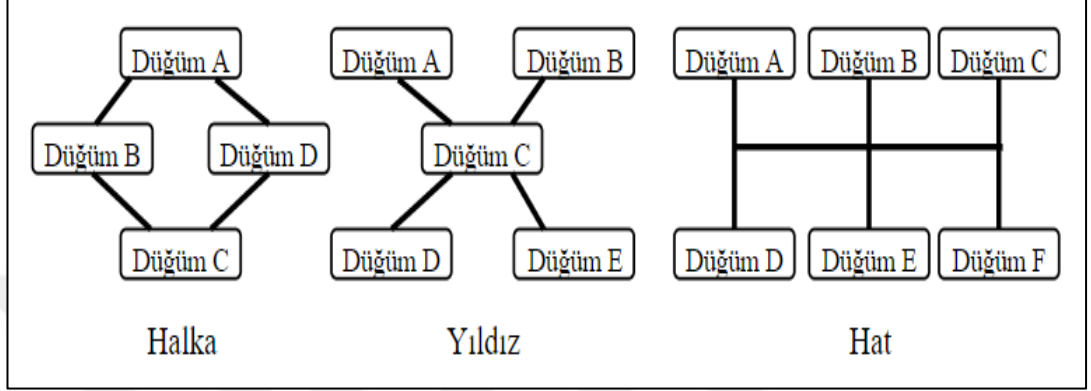
CAN haberleşme protokolün veri iletim hızları arasında mesafe faktörü bulunmaktadır. Veri iletişim hızlarına karşılık gelen mesafeler Çizelge 3.1'de gösterilmiştir (Murphy, 2016).

Çizelge 3.1. Hat uzunluğuna bağlı hız değişimi (Murphy, 2016)

Hat Uzunluğu (metre)	Maksimum Hız
40	1 Mbit/s
100	500 kbit/s
200	250 kbit/s
500	125 kbit/s
1000	40 kbit/s

3.2.1. CAN topolojisi

CAN haberleşme protokolünün topolojisi bir ağ üzerindeki düğümlerin birbirine bağlayan kablonun yapısıdır. Bunlar Halka, Yıldız veya Hat topolojilerinden birisi olabilir. Şekil 3.5'te bu topolojilerin yapısı gösterilmiştir.



Şekil 3.5. CAN protokolünde kullanılan topolojiler

Bu topolojiler arasında en yaygın olan topoloji, hat topolojisidir (Richardson vd., 2001). Bu topolojinin üstün tutulmasının nedeni, her modül ağ içerisindeki mesajlaşmayı sağlayabilmek için tek bir seri kablo ile diğer modüllere bağlıdır. Düğümler arasında CSMA/CD+CR erişim kuralları kullanılır. Bu kurallar, iki düğüm aynı anda mesaj göndermek istediğinde ortaya çıkan çakışmaları engellemek için kullanılır (Pazul, 1999; Ünal, 2006).

3.2.2. CAN denetim mekanizması

Ağ sistemlerinde veriyoluna erişim mekanizmaları, sistemlerin gerçek zamanlı uygulamaları destekleme yetenekleri ve erişim mekanizmasından kaynaklanan mesaj gecikmelerinin sistem performansına etkilerinden dolayı çok önemlidir. CAN çarpışmayı çözme ve gerçek zamanlı uygulamaları gerçekleştirme amacıyla tanıtıcı alan üzerinde öncelik esasına dayalı olan taşıyıcı duyarlı çoklu erişim / mesaj öncelik denetimli çarpışma sezme (Carrier Sense Multiple Access/Collision Detection with Arbitration on Message Priority, CSMA/CD+AMP) protokolünü ortam erişim metodu olarak kullanır (Lawrenz, 1997; Ekiz, 1997).

Bit seviyesinde uzlařtırma iřleminin gerekleřebilmesi iin iki nemli durumun oluřması gerekmektedir. Birinci durum, sistemdeki lojik durumların baskın (dominant) ve ekirik (recessive) olarak tanımlanması gerekir (Tindell vd., 1994). İkinci durum ise, CAN hattı üzerinde gerek olarak grnen gnderme giriřimlerinin lojik durumlarının nasıl olduėunun mesaj gnderen modl tarafından gzetlenebilmesidir. CAN protokol, lojik 0 seviyesini baskın, lojik 1 seviyesini ise ekirik olarak tanımlamaktadır. Baskın bit durumu, her zaman ekirik bit durumunun zerinde uzlařtırma iřlemini kazanacaktır. CAN mesaj erevesi (Frame) ierisinde bulunan mesaj belirleyicideki en dřk deėerden dolayı, gnderilecek olan mesaj en yksek ncelikteki mesaj olacaktır. Dřk ncelikli bir mesaj, bir noktadan sonra ekirik bir bit gndermeyi deneyecektir. Bu noktada, bu biti gnderen modl uzlařtırma iřlemini kaybedecek ve aniden mesaj gnderme iřlemini durduracaktır. Yksek nceliėe sahip mesaj, gnderme iřlemi bitene kadar hat zerinde istenen yerlere gidecektir (Kutlu, 2004).

3.2.3. CAN hızı

CAN haberleřme protokolnde birden fazla veri transfer hızı bulunmaktadır. Bunlar arasında en ok kullanılanları 125 kbit/s, 250 kbit/s ve 500 kbit/s olarak belirlenebilir. Bunlarla birlikte CAN protokolnde haberleřme hızının st limiti olarak 1000 kbit/s'de nadir olarak kullanılmaktadır (Bosch, 1991). Bu hız seimi genelde modllere yazılım yklenmesi gibi nadir durumlarda iřlem hızının yksek olması iin kullanılır.

3.2.4. CAN mesaj erevesi

CAN haberleřme protokol istasyonlara baėlı deėil mesaja dayalı bir haberleřme protokoldr. Bu zelliėi sayesinde btn dėmlerin btn iletimlerini fark edebileceėi anlamına gelir. Sadece zel bir dėme mesaj gnderilmez, btn dėmler hattaki trafiėin tamamını kontrol eder. CAN dėmlerinde filtreleme yapılabilir, bu sayede o dėm sadece kendi istediėi mesaj kimliklerini alır ve bořuna iřlemciye yk binmez (Kara, 2009). Dėmler arasındaki haberleřme 5 ereve tipi kullanılarak yapılır. izelge 3.2'de ereve tipleri ve bunların grevleri gsterilmiřtir.

Çizelge 3.2. Çerçeve tipleri ve görevleri (Kara, 2009)

Çerçeve Tipi	Görevi	Kullanıcı Ayarı
Veri Çerçevesi	Bu çerçeve düğümdeki gönderilen mesaj tarafından kullanılır. Hattaki alıcı düğümlere gerekli veriler gönderilir.	Gerekli
Uzak Çerçeve	Bu çerçevede uzaktaki bir düğümden veri isteğinde bulunmak için gönderilir. Uzaktaki bir düğüm bu isteği aldığı anda aynı ID ile mesaj göndererek cevap verir.	Gerekli
Hata Çerçevesi	CAN veri yolunda bir hata algılandığı zaman diğer ünitelere bu durumu haber etmek için kullanılır.	Gerekli Değil
Aşırı Yük Çerçevesi	Bu çerçevede alıcı düğüm tarafından üretilir ve alıcı düğüm veriyi almaya hazır olmadığını belirtir.	Gerekli Değil
Ayrımcı Boşluk	Veri çerçevesi veya uzak çerçeveyi kendisinden önce gelen çerçeveyi ayırt etmek için kullanılır.	Gerekli Değil

3.2.4.1. Veri çerçevesi

CAN protokolünde standart ve genişletilmiş çerçeve olmak üzere 2 tipte veri çerçevesi bulunmaktadır. Bunlar arasındaki farklılık mesaj kimliklerinin uzunluğudur. Veri çerçevesi bir mesajı transfer düğümünden alıcı düğüme göndermek için kullanılan bir çerçevedir. CAN hattının en önemli çerçevesidir. Bu çerçeve 7 adet alandan oluşur.

- Başlangıç biti (Start of Frame)
- Öncelik karar alanı (Arbitration Field)
- Kontrol alanı (Control Field)
- Veri alanı (Data Field)
- CRC alanı (Cyclic Redundancy Check Field)
- ACK alanı (Acknowledge Field)
- Çerçeve sonu alanı (End of Frame)

Standart veri çerçevesi içerisinde 12 bitlik bir uzlaştırma alanı bulunmaktadır. Bu 12 bitten 11 tanesi mesajın kimliğidir. Diğer kalan 1 bit ise RTR (Uzaktan İletim Talebi) bitidir. Bu bit veri çerçevesinin uzaktaki bir düğümden istekte bulunması için kullanılır (Zuberi ve Shin, 1995).

Genişletilmiş veri çerçevesinde 32 bitlik bir uzlaştırma alanı bulunmaktadır. Bu 32 bitten 29 tanesi mesajın kimliğidir. 1 bit IDE (Belirleyici) bitidir. Bu bit veri çerçevesinin genişletilmiş veri çerçevesinde olduğunu göstermek için kullanılır. 1 bit SRR biti bulunur ve bu bit kullanılmaz. 1 bit ise RTR bitidir. Bu RTR biti standart veri çerçevesi ile aynı amaç için kullanılır (Zuberi ve Shin, 1995).

Başlangıç biti: Veri alanının başlangıcında yer alır. 1 bit uzunluğundadır ve CAN mesajının başlangıcını belirtir (Bayılmış, 2006). Bu bit her zaman baskın (0) bittir. Bu alan standart ve genişletilmiş format için ortak bir alandır.

Öncelik karar alanı: Bu alan standart veri çerçevesinde 12 bit, genişletilmiş çerçevede ise 32 bit olarak tanımlanır. Aynı zamanda bu alan verinin önceliğini belirtir. Standart veri çerçevesinde 12 bitin 1 bitini RTR biti oluşturur. Genişletilmiş çerçevede de bu durum aynıdır 1 bit RTR bitidir.

Kontrol alanı: Bu alanda 6 bit bulunmaktadır. Gönderilecek mesajdaki verilerin kaç bayt olduğunu belirtir. Bu alan standart ve genişletilmiş CAN formatı için ortak bir alandır. Bu alanda rezerve edilmiş r0 ve r1 bitleri bulunmaktadır. Bu 2 bit gönderilirken baskın (lojik 0) olma zorunluluğu bulunmaktadır. Standart formatta bulunan IDE biti lojik 0 ise gönderilecek herhangi bir tanıtıcı bilgisi olmadığını göstermektedir. Bu alandaki diğer kalan 4 bit ise veri uzunluk kodu (DLC) olarak adlandırılmaktadır. Gönderilecek olan verinin uzunluğu buradaki bitlere bakılarak karar verilir (Kara, 2009). Bu karar verme mekanizmasındaki gerekli bit değerleri Çizelge 3.3'te gösterilmiştir.

Çizelge 3.3. Veri uzunluk kodu ve bayt sayısı (Kara, 2009)

Veri Bayt Sayısı	Veri uzunluk kodu			
	DLC3	DLC2	DLC1	DLC0
0	D	D	D	D
1	D	D	D	R
2	D	D	R	D
3	D	D	R	R
4	D	R	D	D
5	D	R	D	R
6	D	R	R	D
7	D	R	R	R
8	R	D ya da R	D ya da R	D ya da R

Veri Alanı: Bu alan bir düğümden CAN veri yoluna gönderdiği verinin içeriğini belirtir. Sıfır ile sekiz bayt arasında olabilir. Bu gönderilecek bayt uzunluğu DLC alanındaki gibi olmalıdır. Bu alanın en düşük adresinde bulunan bayt ilk önce gönderilecektir (Lawrenz, 1997). Bu alan standart ve genişletilmiş CAN protokolünde aynıdır.

CRC Alanı: Bu alan standart ve genişletilmiş CAN haberleşme protokolü için aynıdır. 16 bitten oluşan bu alan 15 biti CRC alanı ve son bit ise ayırma biti olarak tanımlanır. CAN hattında bir düğüm mesaj gönderirken veya başka bir düğüm bir veriyi alırken bu alandaki CRC değeri karşılaştırılır, aynı değilse bir hata mesajı üretilir (Kara, 2009).

ACK Alanı: Bu alan 2 bitlik uzunluğundadır. Bir biti ACK slot diğer biti ise ACK belirteçidir. Bu alanda mesajın alınıp alınmadığını ve düğümler arasında bir hatanın olup olmadığı hakkında gönderici düğümü bilgilendirir. Mesajı gönderen düğüm veri çerçevesinin ACK alanının bitlerini lojik 1 olarak gönderir. Bu mesajı alan düğüm ise mesajı doğru bir şekilde aldığı anda ACK slot alanını gönderici düğümüne lojik 0 olarak gönderir. Gönderici düğüm CAN hattını dinlediğinden dolayı diğer düğümün bu değişikliğini algılar ve en az bir tane düğümün mesajı tam ve doğru bir şekilde aldığı anlar (Lawrenz, 1997). Bu özellik CAN protokolünün güvenilirliği açısından önemlidir.

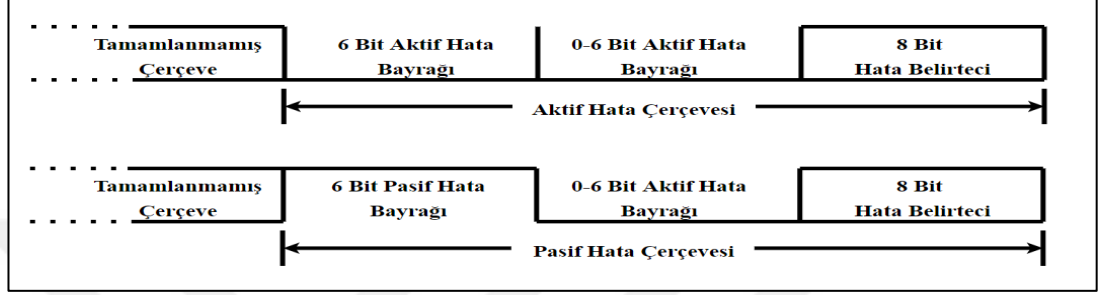
Çerçeve sonu alanı: Bu alan standart ve genişletilmiş CAN için ortak bir alan olup çerçevenin sonu olduğunu bildirir. 7 bitlik lojik 1 bitinden oluşur.

3.2.4.2. Uzak çerçeve

CAN haberleşmede veri çerçevesi içerisinde bulunan RTR biti, bir düğümün başka bir düğümden bilgi almak istediği durumlarda kullanılır. Bu çerçevede veri alanı bulunmamaktadır. Bir düğüm, uzak çerçeve ile mesaj gönderirken istekte bulunduğu veri çerçevesinin mesaj kimliğini koymalı ve RTR bitini lojik 1 olarak göndermesi gerekmektedir (Lawrenz, 1997).

3.2.4.3. Hata çerçevesi

CAN haberleşme protokolünde hata yönetim mekanizması sayesinde hatalı veya bozulmuş bir düğümden gönderilen bir hata çerçevesi CAN hattını tamamen işgal etmesi önlenmiş olur (Bayılmış, 2006). Şekil 3.6'da hata çerçevesi biçimleri gösterilmiştir.



Şekil 3.6. CAN hata çerçevesi biçimleri (Bayılmış, 2006)

3.2.4.4. Aşırı yük çerçevesi

Aşırı yük çerçevesi bir düğümden bir sonraki mesajı almadan önce daha önceki aldığı veriyi işleyebilmesi için yeterli bir zamana sahip olmadığını belirtmek için kullanılır. Bu çerçeve ile alıcı düğüm gönderici düğümlerden verilerini almasını geciktirmek ister (Bayılmış, 2006).

3.2.4.5. Ayırıcı boşluk

Bu çerçeve, veri çerçevesi ve uzak çerçeveyi birbirinden ayırır. Veri çerçevesi veya uzak çerçevesi, düğümler arasındaki veri yolundan hangi çerçeve gelirse gelsin ayırıcı boşluk ile ayrılırlar. Fakat hata ya da aşırı yük çerçeveleri bu ayırıcı boşluğu kullanmazlar (Kara, 2009).

3.2.5. CAN protokolü hata mekanizması

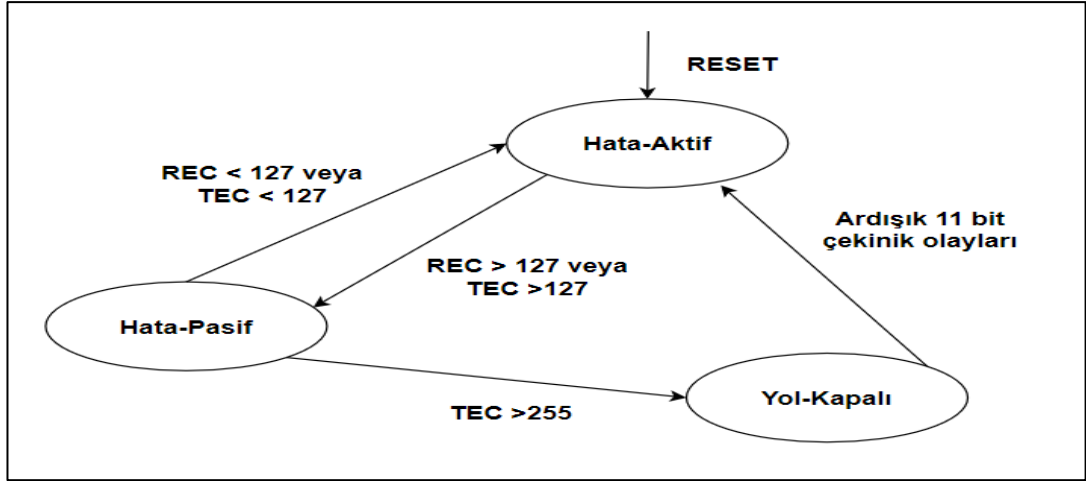
CAN protokolünde bir hata meydana geldiği zaman düğümlerin oluşturduğu bir hata çerçevesi bulunmaktadır. Bu çerçeve özel bir yapıdadır ve bit stuffing kuralını ihlal eder. Bu kuralı ihlal edildiğini diğer düğümler anlar ve buna göre davranırlar.

Hata yönetim birimi tarafından kontrol edilen bu süreçte genelde 5 farklı hata çeşiti bulunmaktadır. Tespit edilen bu hata yapıları aşağıda sıralanmıştır.

- CRC Hatası
- Teyit Hatası
- Şekil Hatası
- Bit Hatası
- Dolgu Hatası

Düğümler arasında alınan CRC dizisi ile hesaplanan CRC değerinin birbirine eşit olmadığı durumda CRC hatası ortaya çıkar. Veri çerçevesi içerisinde bulunan ACK slot bitini 0 olarak gören düğüm teyit hatasını oluşturur. CAN mesaj çerçevesi içerisinde belirli pozisyonlarda gönderilmesi gereken önceden tanımlanan bit değerleri vardır. Alıcı düğümün bu alanlarda geçersiz bir bit algılaması durumunda şekil hatası meydana gelir. CAN veri yoluna bit gönderen bir düğüm aynı anda veri yolunu izler. İzleme işlemi bit bit olarak yapılır. İzlenen bu bit seviyesi, gönderilen bit seviyesinden farklı olduğunda bir bit hatası meydana gelir. Mesaj gönderiminde bulunan bir düğüm normal şartlarda beş adet ardışık lojik 1 seviyesinde bittten sonra lojik 0 seviyeli bir bit veya ardışık lojik 0 seviyeli bittten sonra lojik 1 seviyeli bir bit ekler. Bu yapılan işleme bit dolgulama işlemi denmektedir. Eğer ki alıcı bir düğüm altı adet ardışık eşit bit seviyesi alırsa, bit dolgu hatası meydana gelir.

Şekil 3.7’de görüldüğü gibi veriyolu üzerindeki bir düğüm resetlendiği zaman aktif hata moduna geçer. TEC veya REC sayacı 127 değerinin üzerine çıkarsa pasif hatalı moda geçer. Eğer sayaç azalır tekrar aktif hata durumuna geçer. Bunun yanında TEC sayıcısı 255 değerinin üzerine çıktığında bu düğüm hattan çıkar ve tekrar veriyoluna katılması için yazılımsal ya da donanımsal olarak sıfırlanması gerekmektedir (Bayılmış, 2006).



Şekil 3.7. TEC ve REC sayaçlarının durumu

3.2.6. CAN protokolü hat yükü

CAN haberleşme protokolünde gönderilen ve alınan mesajlardan dolayı düğümler, hatta belirli bir yük bindirirler. CAN veri yolunun belirli bir kapasitesi vardır. Bu kapasite bir saniyede kaç bitlik veri transferi yapabileceğinin ölçüsüdür. Düğümden gönderilen bir mesajın kaç bitlik veri taşıdığı ve ilgili çerçevenin kaç bitten oluştuğu belli olduğu için bu yoğunluk hesaplanabilir (Tindell vd., 1995).

CAN hattının yoğunluğu belirli bir değerin üzerine çıkarsa düşük öncelikli mesajlar sürekli baskılanacağından dolayı veri yoluna mesaj gönderemezler. Böyle bir durum ise CAN protokolünün güvenilirliğini düşürecektir. Hat yoğunluğunu hesaplamak için düğümdaki her bir mesajın hatta oluşturduğu yükü ayrı olarak hesaplamak gereklidir (Di Natale, 2000). Daha sonra hattaki yoğunluk bu değerlerin toplamına eşit olacaktır. Veri yolundaki bir adet mesajı X tane düğümün alacağını düşündüğümüzde tek bir mesajın hatta oluşturduğu yoğunluk mesajın basılması ve dinleyen her bir düğümün buna vereceği yanıt bir araya gelerek hesaplanır.

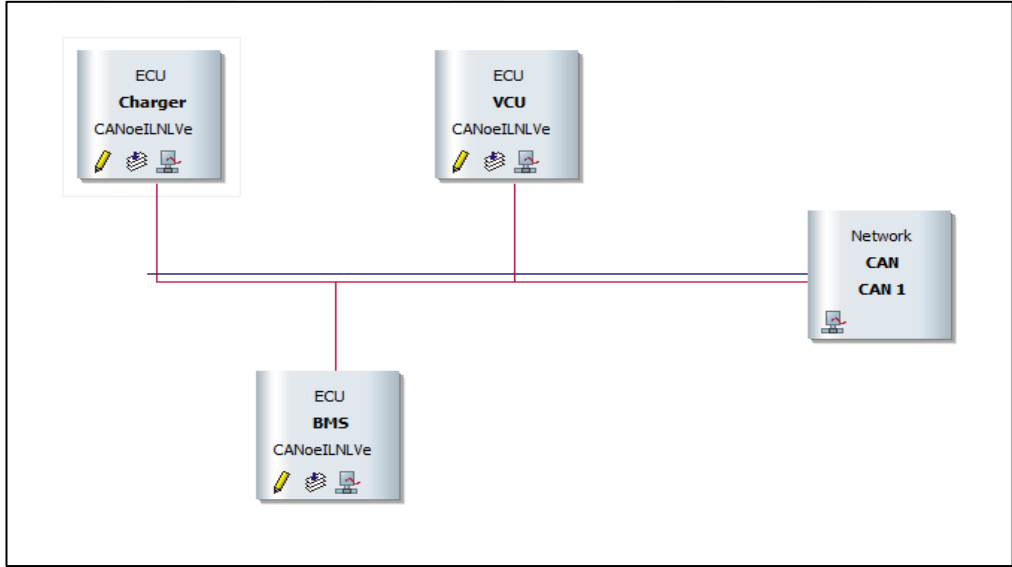
3.3. Tasarlanan Deney Yüzeyi

BYS bataryanın sıcaklık, voltaj, akım gibi önemli verileri ölçerek şarj ve deşarj işlemlerini yöneten yazılım ve donanım tabanlı bir sistemdir. Bununla beraber ölçtüğü bu verileri toplayarak belirli algoritmalarla pilin SoC ve SoH değerlerini hesaplar. Pili

tehlikeye girdiği durumlarda kullanıcıyı bilgilendirir ve gerekirse sistemi devre dışı bırakabilir.

BYS elektrikli araçlarda diğer donanımlarla haberleşmelidir. Bütün donanımların birbirleriyle haberleşmesini sağlayan ve otomotiv sektöründe en çok kullanılan haberleşme protokolü CAN haberleşmesidir.

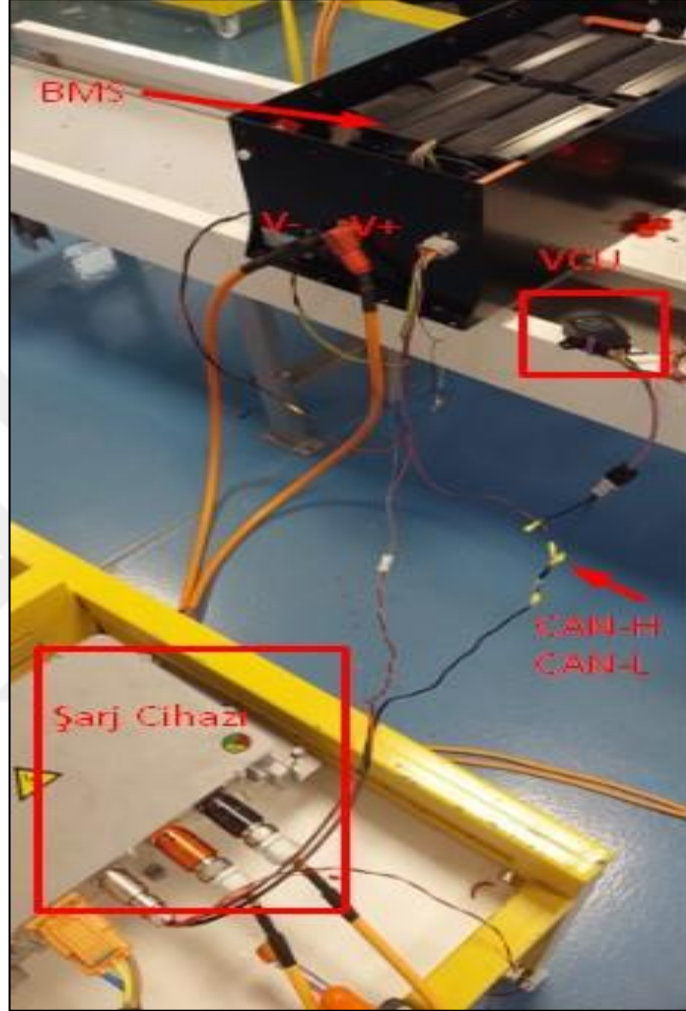
Hazırlanan deney düzeneğinde CAN hattında 3 tane düğüm bulunmaktadır. Bunlar VCU, BYS ve şarj cihazıdır. Bütün bu düğümler birbirleriyle iletişim içerisinde olmalıdır. Her düğüm diğer düğümleri kontrol etmeli ve olası bir arıza durumunda pilleri tehlikeye atmadan sistemi kapatmalıdır. Birbirleriyle iletişim içerisinde olan her bir düğümün kendine ait belirli CAN mesaj kimlikleri bulunmaktadır. Bu mesaj kimlikleri birbirlerinden bağımsız olmalıdır. CAN hattının düzgün çalışabilmesi için mesaj kimlikleri birbirleriyle karıştırılmamalıdır. Sistem üzerindeki bulunan düğümler Şekil 3.8’de gösterilmiştir. Bu düğümler, CANalyzer (CANalyzer.J1939/pro 9.0.86, Vector Informatik GmbH, Stuttgart, Germany) programından alınan görüntüdür.



Şekil 3.8. CAN hattı üzerindeki bulunan düğümler

Şekil 3.9’da deney düzeneği gösterilmektedir. Deney düzeneğinde nominal 48V bir batarya paketi bulunmaktadır. Bu batarya paketinin kontrolü BYS üzerinden yapılmaktadır. BYS sisteminin kontrolü ise VCU ile yapılmaktadır. Deney düzeneğinde şarj cihazı enerjisini şebeke üzerinden almaktadır ve DC gerilim üretmektedir. Ürettiği bu gerilim ile batarya paketini şarj etmektedir.

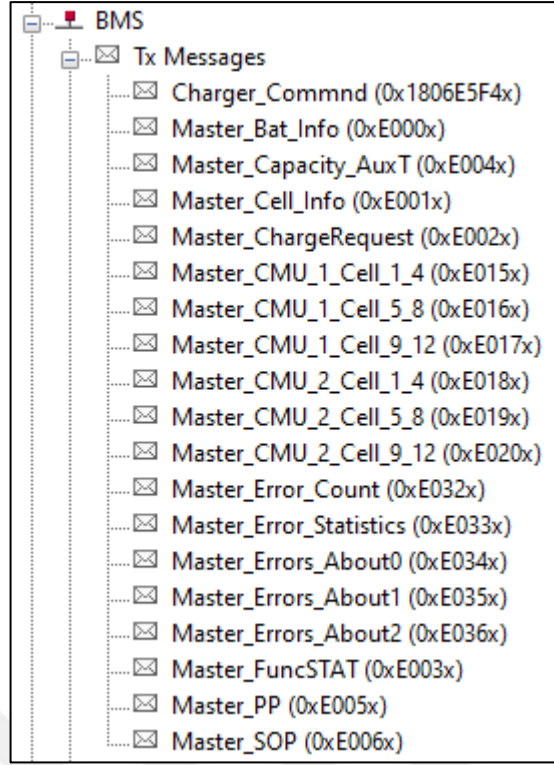
Bütün bu sistemin kontrolü CAN hattı üzerinden yapılmaktadır. Şekil 3.9’da görülen CAN-H ve CAN-L kabloları ile birbirine bağlanmaktadır. Batarya paketi istediği akımı şarj cihazına CAN hattı üzerinden göndererek pilin güvenli bir şekilde şarj olmasını sağlamaktadır.



Şekil 3.9. Deney düzeneği

Deney düzeneğinde kullanılan sistemde her bir düğümün göndermiş olduğu ve diğer düğümlerden alacak olduğu mesajlar vardır. Alınan bu mesajlara göre diğer düğümler birbirleriyle iletişim halindedirler. Deney düzeneğinde bu bağlantı CAN-H ve CAN-L kabloları ile birbirlerine bağlanmıştır ve gerekli olan 120 Ω direnç hatta paralel olarak bağlanmıştır. Bu sayede hat üzerindeki gürültüyü en aza indirmiş oluruz.

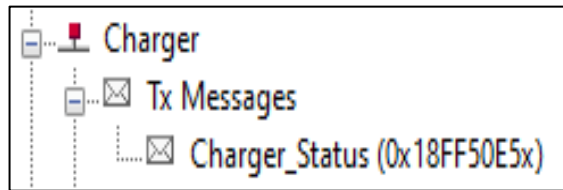
Deney düzeneği ve simülasyon programında bulunan düğümlerin her birinin CAN hattına göndermiş olduğu mesajlar bulunmaktadır. Şekil 3.10’da BYS düğümünün göndermiş olduğu mesajlar gösterilmiştir.



Şekil 3.10. BYS düğümünün gönderdiği mesajlar

Burada BYS düğümünün batarya hakkında göndermiş olduğu mesajlar yer almaktadır. Deneysel düzeneğinde burada gösterilen 0xE000 ve 0xE001 mesaj kimliğine sahip olan mesajların gönderme sıklıkları 100 ms ve 1 s olarak değiştirilecektir. Bu mesaj kimliklerinin seçilmesinde herhangi bir özel durum yoktur. CAN hattında oluşacak olan yoğunluğun değişimini tespit etmek için rastgele seçilen mesajlardır.

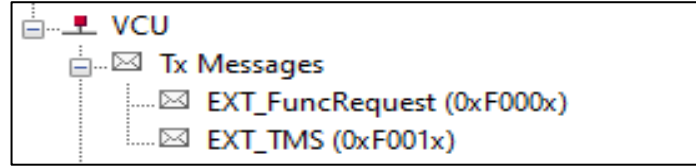
Deneysel düzeneğinde bulunan bir diğer düğüm ise şarj cihazıdır. Bu düğüm BYS kontrolünde olan pillere gerekli olan akımı sağlayarak şarj etmektedir. Şekil 3.11’de bu şarj cihazının göndermiş olduğu mesaj gösterilmektedir.



Şekil 3.11. Şarj cihazı düğümünün gönderdiği mesaj

CAN hattı yoğunluğu değişimi için yapılacak olan testlerde bu mesaj kimliğinin gönderme sıklığı ile ilgili bir değişim yapılmayacaktır. Şarj cihazının göndermiş olduğu mesajda anlık olarak batarya paketine giren akımı göstermektedir.

Deney düzeneğinde bulunan son düğüm ise VCU'dur. Normalde VCU sistemi tüm aracı kontrol edebilen bir düğümdür. Ancak bu deney düzeneğinde sadece BY5 düğümüyle olan iletişimi esas alınmıştır. VCU düğümünün göndermiş olduğu mesajlar Şekil 3.12'de gösterilmiştir.



Şekil 3.12. VCU düğümünün gönderdiği mesajlar

VCU düğümünün göndermiş olduğu bu iki mesaj ile batarya paketini kontrol edebilmektedir. Bu mesajları alan BY5 düğümü bataryayı aktif veya pasif hale getirir. VCU düğümünün göndermiş olduğu mesajların gönderme sıklıkları 100 ms olarak ayarlanmıştır ve deney düzeneğinde bu sıklık değiştirilmemiştir.

Bu deney düzeneğinde tüm bu mesajların bulunduğu CAN hattında her bir mesajın hat üzerindeki yoğunluğu tespit edilecek ve gönderme sıklığı değiştirilen mesajların da hat üzerindeki yoğunluk değişimi bulunacaktır.

3.3.1. CAN analizi

Deney düzeneğinde 3 tane düğüm bulunmaktadır. Bu düğümlerin her biri diğer düğümlere mesaj göndermektedir. Birbirleriyle iletişim içerisinde bulunan bu düğümler CAN hattına bir yük oluşturmaktadır. Oluşan bu trafiği incelemek için sisteme VN1630A (Vector Informatik GmbH, Stuttgart, Germany) cihazı ve CANalyzer yazılımı ile deney düzeneğine bağlantı yapılmıştır. Şekil 3.13'te sisteme bağlanan VN1630A cihazı gösterilmektedir.



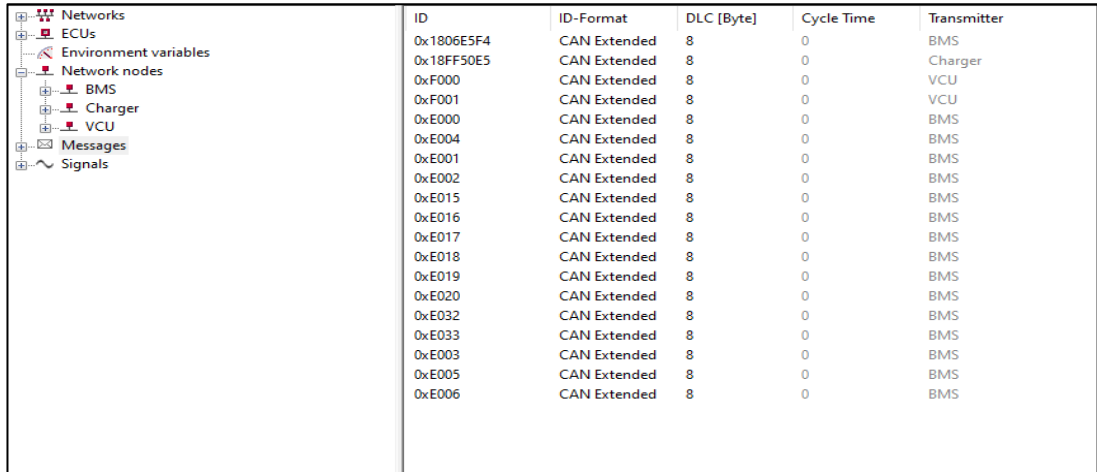
Şekil 3.13. Vector VN1630A cihazı

Bu program ve cihaz aracılığı ile CAN hattında bulunan düğümlerin her birinin hat üzerine ne kadar yük oluşturduğu görülebilmektedir. Ayrıca düğümlerdeki mesajları rahatlıkla görüp analiz edebilmek için database file (.dbc) programa dahil edilebilmektedir.

CAN hattı üzerinde bulunan düğümlerin gönderdiği mesajların sıklığına göre hat üzerinde oluşturduğu yoğunluk artmaktadır. Ayrıca CAN hattının hızı da bu yoğunluğu etkilemektedir. Sistem üzerinde bu yoğunluğu analiz etmek ve en uygun düzeyde tutmak için 6 farklı test yapılmıştır.

CAN hızını 125 kbit/s, 500 kbit/s ve 1000 kbit/s olmak üzere 3 farklı hızlarda sistem yoğunluğu hesaplanmıştır. Bu hızların her birinde 0xE000 ve 0xE001 adreslerine sahip mesajlarının gönderme sıklıkları 100 ms ve 1 s olarak değiştirilmiştir. Her bir mesajın hat üzerinde ne kadar yük bindirdiği yaklaşık olarak hesaplanabilir. Bu hesaplama diğer bölümde detaylı olarak anlatılacaktır.

Şekil 3.14’te CAN hattı üzerindeki bütün düğümler ve mesaj kimliklerinin bulunduğu database file (.dbc) dosyası gösterilmektedir.



ID	ID-Format	DLC [Byte]	Cycle Time	Transmitter
0x1806E5F4	CAN Extended	8	0	BMS
0x18FF50E5	CAN Extended	8	0	Charger
0xF000	CAN Extended	8	0	VCU
0xF001	CAN Extended	8	0	VCU
0xE000	CAN Extended	8	0	BMS
0xE004	CAN Extended	8	0	BMS
0xE001	CAN Extended	8	0	BMS
0xE002	CAN Extended	8	0	BMS
0xE015	CAN Extended	8	0	BMS
0xE016	CAN Extended	8	0	BMS
0xE017	CAN Extended	8	0	BMS
0xE018	CAN Extended	8	0	BMS
0xE019	CAN Extended	8	0	BMS
0xE020	CAN Extended	8	0	BMS
0xE032	CAN Extended	8	0	BMS
0xE033	CAN Extended	8	0	BMS
0xE003	CAN Extended	8	0	BMS
0xE005	CAN Extended	8	0	BMS
0xE006	CAN Extended	8	0	BMS

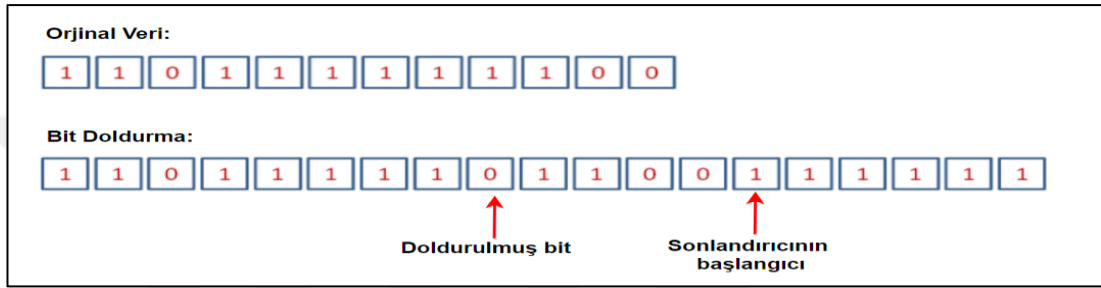
Şekil 3.14. Düğümler ve mesaj kimlikleri

3.3.2. CAN bit doldurma işlemi

Senkronizasyonu sürdürmek için yeterli geçişi sağlamak için, aynı polaritenin ardışık beş bitinden sonra bir miktar zıt polarite eklenir. Bu uygulamaya bit doldurma denir.

Doldurulmuş veri çerçeveleri alıcı tarafından boşaltılır. Çerçevedeki tüm alanlar, sabit bir boyutta olan ve doldurulmayan CRC sınırlayıcı, ACK alanı ve çerçeve sonu dışında doldurulur.

Bit doldurmanın kullanıldığı alanlarda, aynı polariteye sahip (111111 veya 000000) altı ardışık bit hata olarak kabul edilir. Bir hata tespit edildiğinde bir düğüm tarafından aktif bir hata bayrağı iletilebilir. Etkin hata bayrağı, ardışık altı baskın bitten oluşur ve bit doldurma kuralını ihlal eder. Şekil 3.15'te bit doldurma işlemi gösterilmiştir.



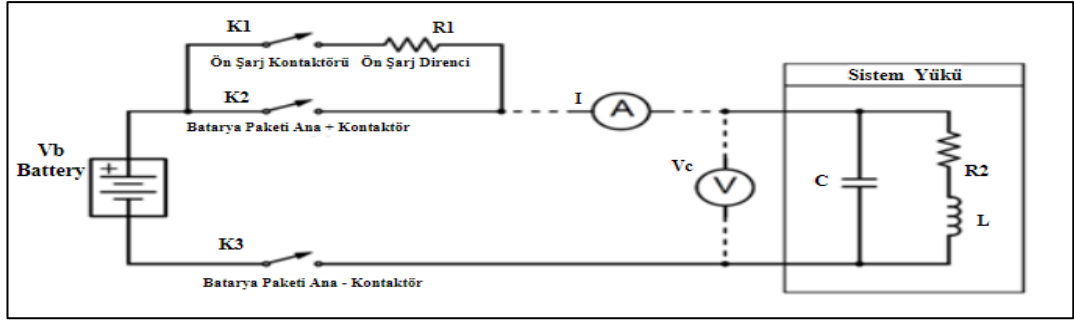
Şekil 3.15. Bit doldurma işlemi

3.3.3. Batarya paketi ön şarj sistemi

BYS, batarya paketini devreye alması ve devreden çıkarması için kontaktörler kullanır. Bu kontaktörleri kontrol ederek hem sisteme enerji verir hem de olası bir kritik durum anında sistemi kapatır. VCU tarafından alınan istek üzerine BYS sisteme enerji vermek için gerekli aşamaları yapmalıdır. Bunlardan biriside ön şarj sistemidir.

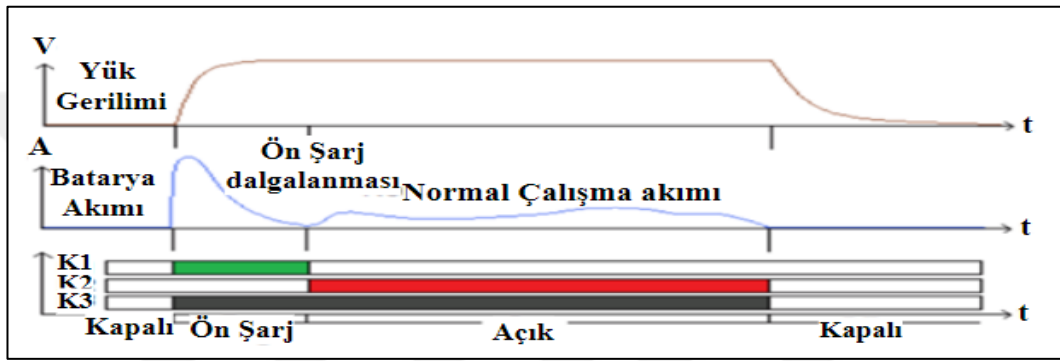
Deney düzeneğindeki şarj cihazı ve invertörler kapasitif yüklü cihazlardır. Pili başlangıçta sisteme bağlarken yük kapasitansı pil voltajına kadar şarj edildiğinden ani bir akım oluşur. Bu ani akım, yük kapasitansına bağlı olarak çok yüksek değerlere ulaşabilir. Bu akım, kontaktörlerden geçerken bir ark oluşturur ve kontaktörlerin yapışmasına ve tekrar kullanılamaz hale getirmesine yol açar. Bu tür olaylardan kaçınmak için batarya paketine bir ön şarj işlemi yapılır.

Şekil 3.16'da gösterildiği gibi, ön şarj devresi pilin artı kutbundaki kontaktöre paralel bağlanır.



Şekil 3.16. Ön şarj devresi

Şekil 3.17’de ön şarj ileminin süreci gösterilmektedir.

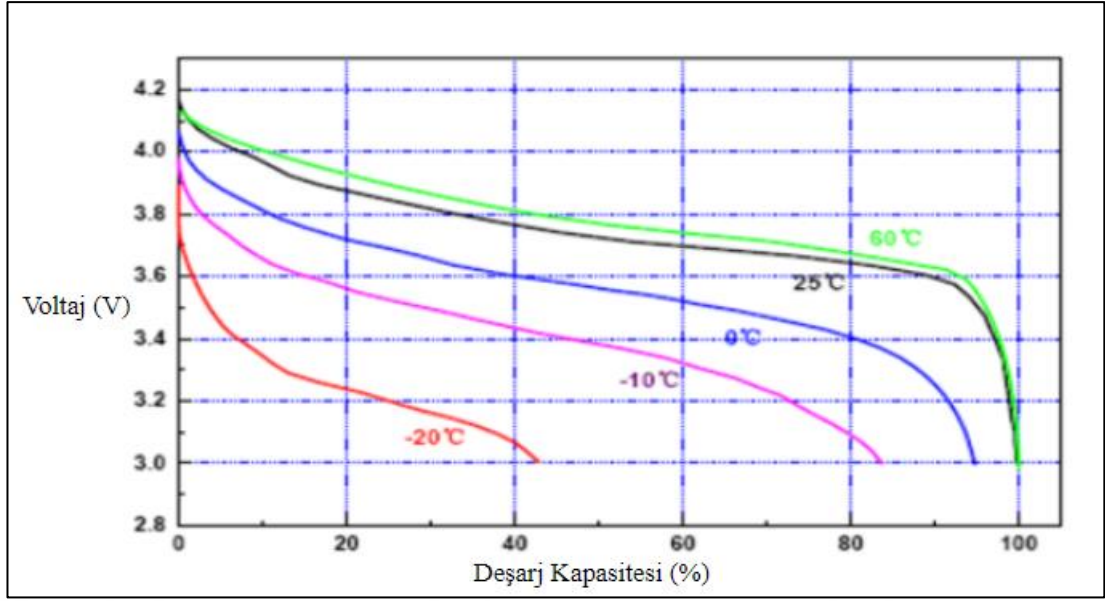


Şekil 3.17. Ön şarj işlem süreci

Genel olarak ön şarj devresi şu şekilde çalıştırılır: sistem kapalıyken tüm kontaktörler kapalıdır. Sistem ilk açıldığında, ani akım düşene kadar yükü önceden şarj etmek için K1 ve K3 kontaktörleri açılır. BYS sisteminde yazılımsal olarak bu akım limiti kontrol edilir ve belirli bir akımın üzerindeyse sistem devreye girmez ve arıza verir. Eğer limit aşılmadı ise ön şarj işleminden sonra K2 kontaktörü açılır ve K1 kontaktörü kapatılır. Böylelikle oluşabilecek ani akımlar nedeniyle gelebilecek zararlar önlenmiş olur.

3.3.4. Batarya paketi ısı sensorü

Batarya yönetim sisteminin kontrol etmesi gereken görevlerden biriside pillerin sıcaklıklarıdır. Sıcaklık, piller için çok önemli bir parametredir ve bu verinin sürekli olarak takip edilmesi gerekmektedir. Sıcaklık değerlerine göre belirli kritik limitlere ulaşıldığında batarya yönetim sistemi pillerden akım çekilmesinin önüne geçmelidir veya sıcaklık duruma göre belirli bir akım limiti belirlemelidir. Şekil 3.18’de Lithium Ion 18650 pilin sıcaklığa göre kapasitesindeki değişim gösterilmiştir.



Şekil 3.18. Sıcaklık-Kapasite eğrisi (Anonim, 2009)

Yapılan batarya yönetim sisteminde 25°C’de 10K Ω değerinde olan ısı sensörü kullanılmıştır. Kullanılan ısı sensörü sıcaklığa bağlı olarak sürekli değişen bir dirence sahiptir. Bu direnç değeri batarya yönetim sistemi tarafından ölçülür ve Steinhart – Hart denklemi ile sıcaklık değeri bulunur.

$$\frac{1}{T} = A + B [\ln(R)] + C [\ln(R)]^2 + D [\ln(R)]^3 \quad (3.1)$$

Bu denklemdeki A, B, C ve D değerleri katsayılarıdır. T ise sıcaklık değeridir. Bu katsayılar ısı sensörünün verilerine göre belirlenirler. Kullanılan ısı sensörüne göre değerler Şekil 3.19’da gösterilmiştir.

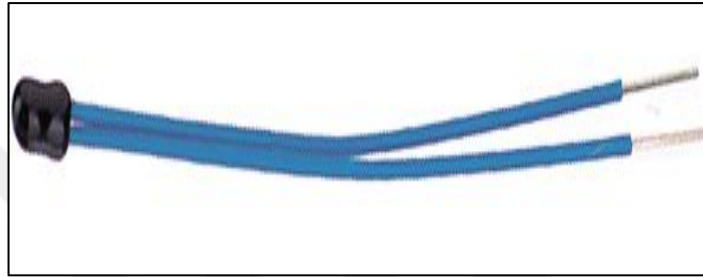
Sıcaklık (°C)		Direnç(Ω)	
T1	0	R1	28665,5
T2	25	R2	10000,0
T3	50	R3	4054,50
T4	70	R4	2118,30

A	B	C	D
0,002106565417113	-0,00015395818333	4,597330645185E-05	-1,579992450164E-06

Şekil 3.19. Isı sensörü katsayılarının belirlenmesi

Steinhart – Hart denklemindeki parametreler belirlendikten sonra sensörden alınan direnç deęerleri ile sıcaklıklar bulunmaktadır. Batarya paketinde aynı sensörden altı tane kullanılmıştır. Bu sensörler pil tasarımında farklı noktalar üzerine yerleştirilmiştir. Bu sayede her bir noktadaki sıcaklıklar sürekli olarak takip edilmiştir.

Katsayıların belirlenmesinde kullanılan referans sıcaklık deęerleri, kullanılan ısı sensörünün veri sayfasından alınmaktadır. Şekil 3.20’de batarya paketinde kullanılan ısı sensörü gösterilmektedir.



Şekil 3.20. NTC ısı sensörü

4. BULGULAR

CAN hattı üzerinde 3 tane düğüm bulunmaktadır. Bu düğümler BYS, VCU ve şarj cihazıdır. Bu düğümlerin göndermiş olduğu mesajlar ve bunların sıklığı, CAN hattı üzerinde bir yük oluşturmaktadır.

Sistem üzerinde altı farklı test yapılmıştır. Bunlar şu şekildedir.

- CAN Hat hızı = 125 kbit/s, 0xE000 ve 0xE001 mesajları 100 ms ve 1 s
- CAN Hat hızı = 500 kbit/s, 0xE000 ve 0xE001 mesajları 100 ms ve 1 s
- CAN Hat hızı = 1000 kbit/s, 0xE000 ve 0xE001 mesajları 100 ms ve 1 s

CAN hattındaki bu yükü bulmak için 3 farklı yöntem kullanılmıştır ve bu yöntemlerden çıkan sonuçlar birbirleriyle karşılaştırılmıştır. Öncelikle CAN hattındaki bu yük için hesaplama yapılmıştır. Hesaplama işlemi yapılırken bit doldurma işlemi yok sayılmıştır ve CAN mesaj çerçevesi bit uzunluğu 128 bit olarak alınmıştır.

Her bir mesajın CAN hattı üzerindeki yoğunluğu yaklaşık olarak şu şekilde hesaplanabilir:

t_{bz} = Bir bitin gönderilme süresi (ms)

V_{bh} = Bit hızı (kbit/s)

t_{mg} = Mesajın gönderilme süresi (ms)

t_{mgs} = Mesajın gönderilme sıklığı (ms)

S_{mb} = Mesaj bit sayısı

M_{bl} = Mesajın hat üzerindeki yoğunluğu (%)

$$t_{bz} = 1 / V_{bh} \quad (4.1)$$

$$t_{mg} = t_{bz} * S_{mb} \quad (4.2)$$

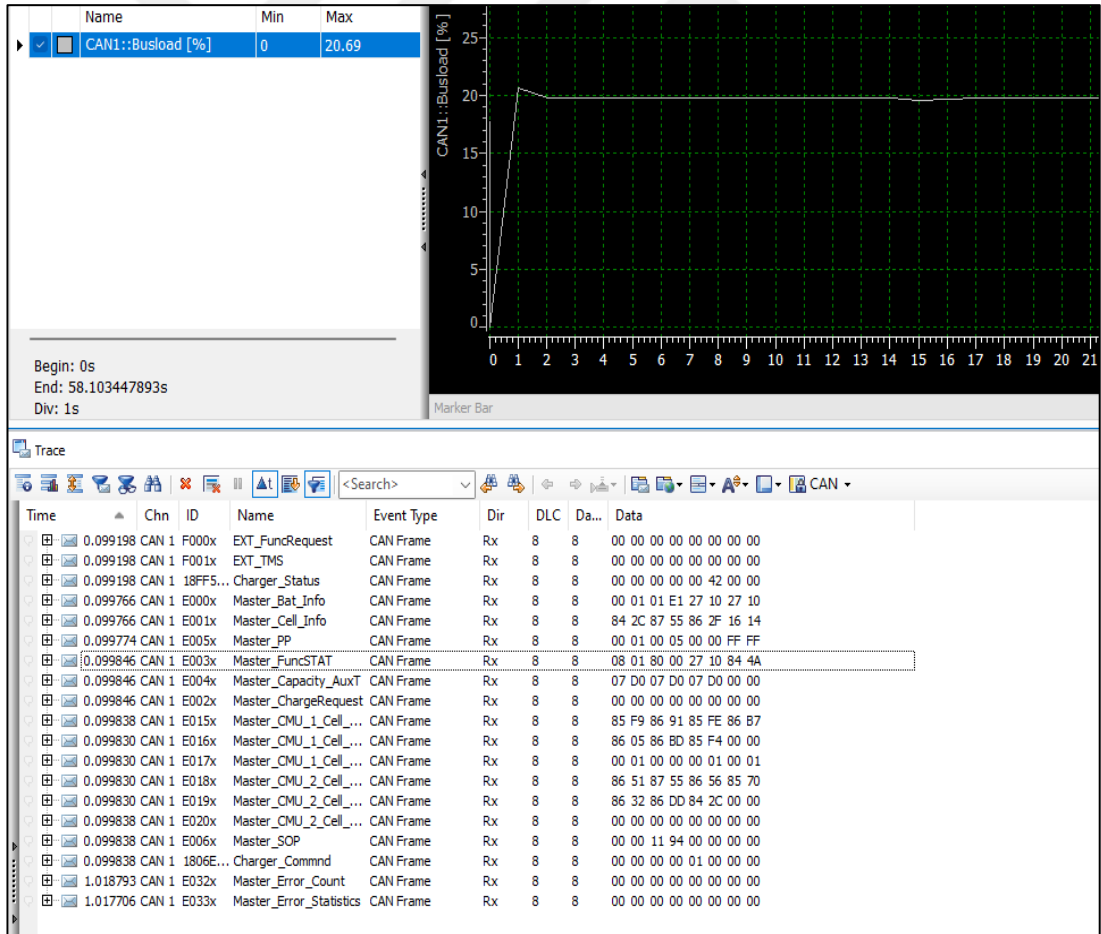
$$M_{bl} = (t_{mg} / t_{mgs}) * 100 \quad (4.3)$$

Burada gösterilen 4.1 denklemini bir bitin gönderilme süresini, 4.2 mesajın gönderilme süresini ve 4.3 denklemini ise mesajın hat üzerindeki yoğunluğunu göstermektedir.

4.1. CAN Hızı 125 kbit/s Analizi

CAN hızının 125 kbit/s olduğu deney düzeneği üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı BYS yazılımında 100 ms olarak değiştirilmiştir.

Şekil 4.1’de deney düzeneği ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 0.099 s olduğu görülmektedir. Yani bu mesajlar 100 ms sıklıkla hat üzerinde mesaj göndermektedirler. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms’dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.1. Deney düzeneği 125 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü

Şekil 4.2’de 125 kbit/s ve 0xE000 ve 0xE001 mesajlarının 100 ms ile gönderildiği deney düzeneğindeki her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %1.15, VCU düğümünün %2.38 ve BYS düğümünün oluşturduğu hat yükü ise %17.22 olduğu saptanmıştır.

Yazılım ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %1.14 ve 0xE001 mesajın hatta olan yoğunluğu ise %1.09 olduğu görülmektedir. Aynı gönderme sıklığına sahip olmasına rağmen birbirlerinden farklı sonuç vermesinin temel sebebi göndermiş olduğu verilerin birbirlerinden farklı olması ve bit doldurma işlemine göre mesaj bit uzunluğunun değişmesidir.

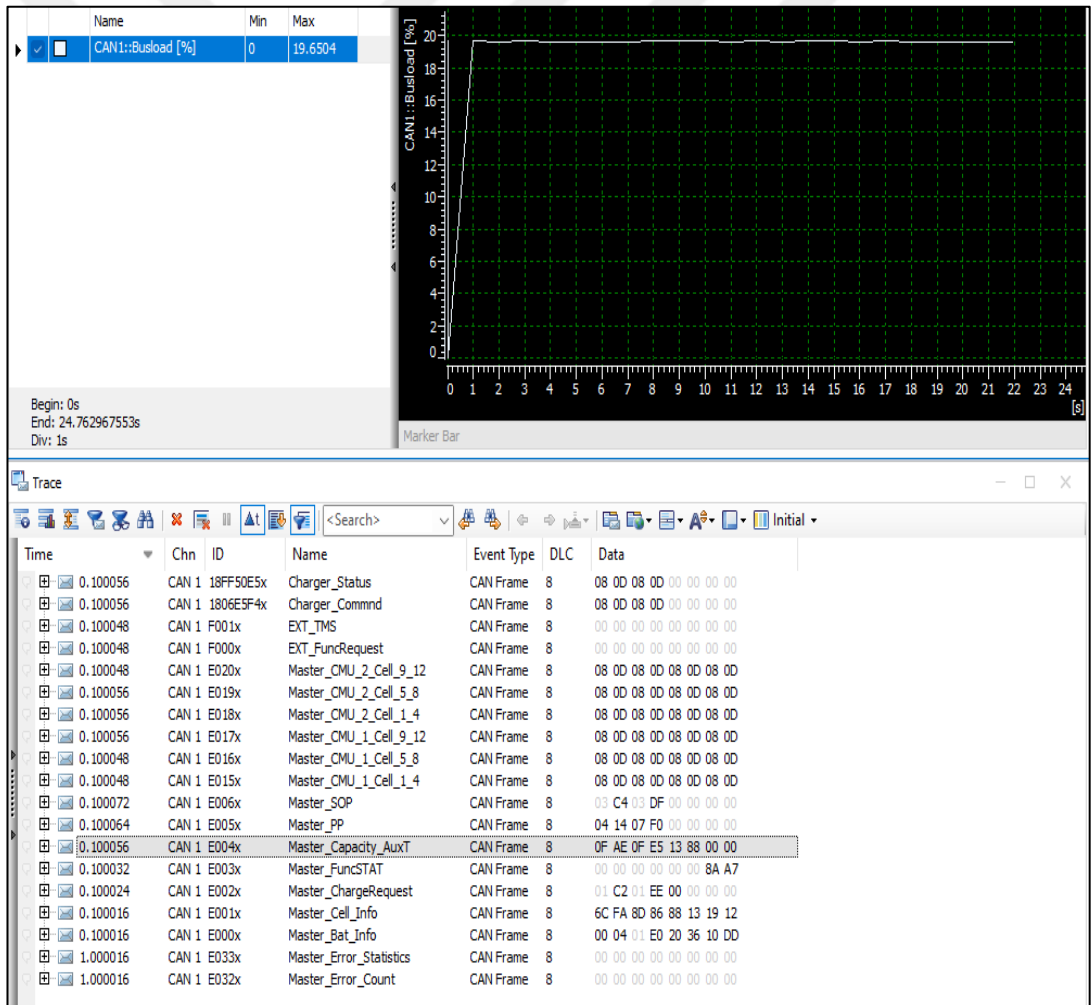
Bu testte elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %20.69 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	19.77	19.58	20.69	19.78
Charger	1.15	1.13	1.15	1.15
Charger_Status (0x18FF50E5x)	1.15	1.13	1.15	1.15
VCU	2.37	2.33	2.38	2.37
EXT_FuncRequest (0xF000x)	1.19	1.17	1.19	1.19
EXT_TMS (0xF001x)	1.18	1.16	1.18	1.18
BMS	16.24	16.05	17.22	16.27
Charger_Commnd (0x1806E5F4x)	1.17	1.17	1.26	1.17
Master_Bat_Info (0xE000x)	1.14	1.12	1.14	1.14
Master_Capacity_AuxT (0xE004x)	1.18	1.18	1.27	1.18
Master_Cell_Info (0xE001x)	1.09	1.07	1.09	1.09
Master_ChargeRequest (0xE002x)	1.19	1.19	1.29	1.19
Master_CMU_1_Cell_1_4 (0xE015x)	1.10	1.09	1.19	1.10
Master_CMU_1_Cell_5_8 (0xE016x)	1.12	1.11	1.20	1.12
Master_CMU_1_Cell_9_12 (0xE017x)	1.18	1.17	1.27	1.18
Master_CMU_2_Cell_1_4 (0xE018x)	1.09	1.08	1.17	1.09
Master_CMU_2_Cell_5_8 (0xE019x)	1.10	1.10	1.19	1.10
Master_CMU_2_Cell_9_12 (0xE020x)	1.19	1.19	1.29	1.19
Master_Error_Count (0xE032x)	0.12	0.05	0.12	0.11
Master_Error_Statistics (0xE033x)	0.12	0.00	0.12	0.12
Master_Errors_About0 (0xE034x)	0.00	0.00	0.00	0.00
Master_Errors_About1 (0xE035x)	0.00	0.00	0.00	0.00
Master_Errors_About2 (0xE036x)	0.00	0.00	0.00	0.00
Master_FuncSTAT (0xE003x)	1.13	1.13	1.23	1.13
Master_PP (0xE005x)	1.18	1.16	1.18	1.18
Master_SOP (0xE006x)	1.17	1.17	1.26	1.17

Şekil 4.2. Deney düzeneği 125 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri

CAN hızının 125 kbit/s olduğu simülasyon programı üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı simülasyon program üzerinden 100 ms ile değiştirilmiştir.

Şekil 4.3'te simülasyon programı ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 0.100 s olduğu görülmektedir. Yani bu mesajlar 100 ms sıklıkla hat üzerinde mesaj göndermektedirler. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms'dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.3. Simülasyon 125 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü

Şekil 4.4'te 125 kbit/s ve 0xE000 ve 0xE001 mesajlarının 100 ms ile gönderildiği simülasyon programında her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %1.14, VCU düğümünün %2.38 ve BMS düğümünün oluşturduğu hat yükü ise %16.14 olduğu saptanmıştır.

Simülasyon programı ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %1.14 ve 0xE001 mesajın hatta olan yoğunluğu ise %1.10 olduğu görülmektedir. Aynı gönderme sıklığına sahip olmasına rağmen birbirlerinden farklı sonuç vermesinin temel sebebi göndermiş olduğu verilerin birbirlerinden farklı olması ve bit doldurma işlemine göre mesaj bit uzunluğunun değişmesidir.

Bu testte elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %19.65 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	19.63	19.62	19.65	19.63
Charger	1.14	1.14	1.14	1.14
Charger_Status (0x18FF50E5x)	1.14	1.14	1.14	1.14
VCU	2.38	2.38	2.38	2.38
EXT_FuncRequest (0xF000x)	1.19	1.19	1.19	1.19
EXT_TMS (0xF001x)	1.18	1.18	1.18	1.18
BMS	16.11	16.10	16.14	16.12
Charger_Commnd (0x1806E5F4x)	1.15	1.15	1.15	1.15
Master_Bat_Info (0xE000x)	1.12	1.12	1.14	1.13
Master_Capacity_AuxT (0xE004x)	1.15	1.15	1.16	1.15
Master_Cell_Info (0xE001x)	1.10	1.09	1.10	1.10
Master_ChargeRequest (0xE002x)	1.16	1.16	1.16	1.16
Master_CMU_1_Cell_1_4 (0xE015x)	1.12	1.12	1.12	1.12
Master_CMU_1_Cell_5_8 (0xE016x)	1.12	1.12	1.12	1.12
Master_CMU_1_Cell_9_12 (0xE017x)	1.12	1.12	1.12	1.12
Master_CMU_2_Cell_1_4 (0xE018x)	1.12	1.12	1.12	1.12
Master_CMU_2_Cell_5_8 (0xE019x)	1.12	1.12	1.12	1.12
Master_CMU_2_Cell_9_12 (0xE020x)	1.12	1.12	1.12	1.12
Master_Error_Count (0xE032x)	0.12	0.12	0.12	0.12
Master_Error_Statistics (0xE033x)	0.12	0.12	0.12	0.12
Master_FuncSTAT (0xE003x)	1.17	1.16	1.17	1.17
Master_PP (0xE005x)	1.16	1.16	1.17	1.16
Master_SOP (0xE006x)	1.15	1.15	1.16	1.15

Şekil 4.4. Simülasyon 125 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri

Hat üzerinde bulunan mesajların kimlik bilgileri, gönderme sıklıkları ve hat üzerindeki yoğunlukları gibi bilgiler Çizelge 4.1’de gösterilmiştir.

Çizelge 4.1. Hesaplama 125 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü

Gönderen Düğüm	Mesaj ID	Gönderme Sıklığı(ms)	Bit Uzunluğu	Hat Yüğü (%)
Şarj Cihazı	0x18FF50E5	100	128	1.024
VCU	0xF000	100	128	1.024
VCU	0xF001	100	128	1.024
BYS	0xE000	100	128	1.024
BYS	0xE001	100	128	1.024
BYS	0xE002	100	128	1.024
BYS	0xE003	100	128	1.024
BYS	0xE004	100	128	1.024
BYS	0xE005	100	128	1.024
BYS	0xE006	100	128	1.024
BYS	0xE015	100	128	1.024
BYS	0xE016	100	128	1.024
BYS	0xE017	100	128	1.024
BYS	0xE018	100	128	1.024
BYS	0xE019	100	128	1.024
BYS	0xE020	100	128	1.024
BYS	0xE032	1000	128	0.1024
BYS	0xE033	1000	128	0.1024
BYS	0x1806E5F4	100	128	1.024
			Toplam Hat Yüğü:	17.6128

0xE000 mesajının hatta olan yoğunluğu şu şekilde hesaplanabilir:

$$t_{bz} = 1 / 125 = 0.008 \text{ ms} \quad (4.4)$$

$$t_{mg} = 0.008 * 128 = 1.024 \text{ ms} \quad (4.5)$$

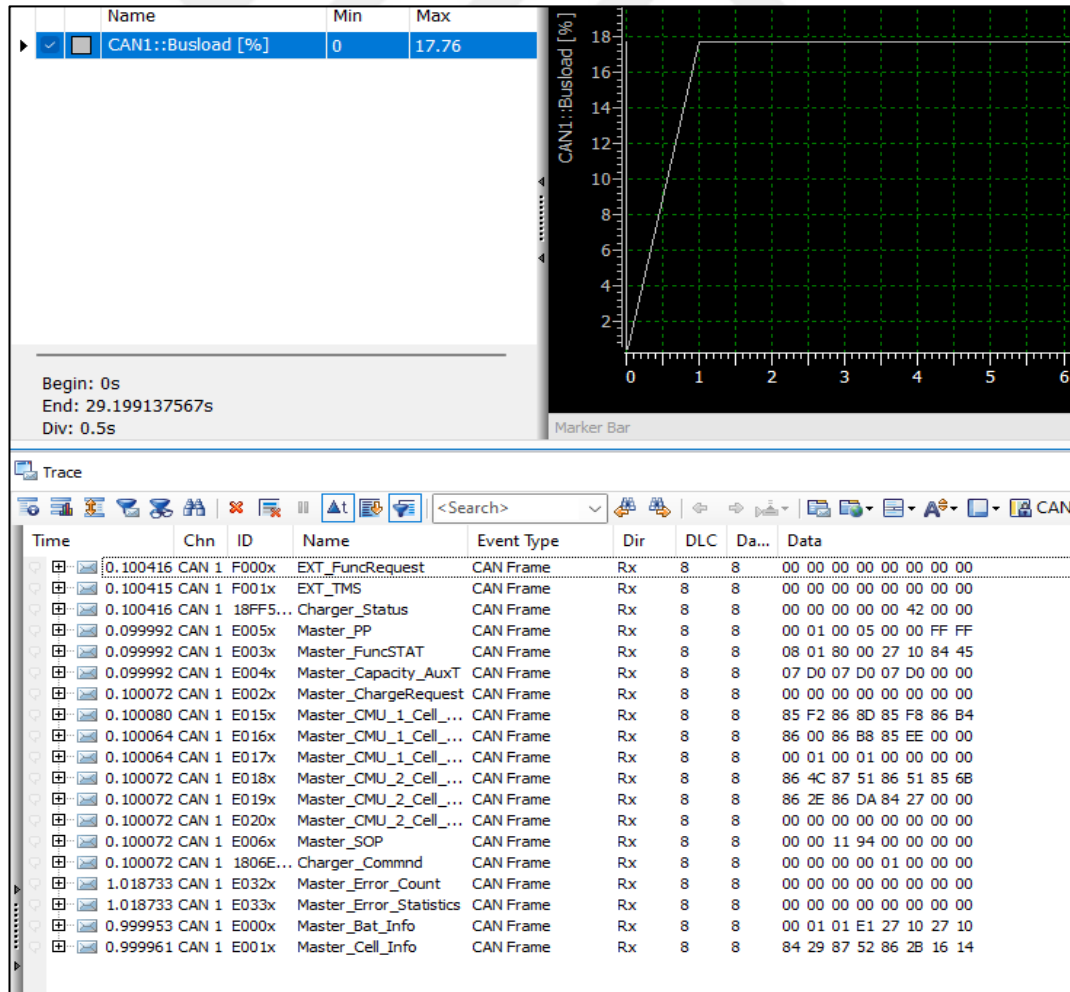
$$M_{bt} = (1.024 / 100) * 100 = \% 1.024 \quad (4.6)$$

Buradaki yapılan hesaplamada, 0xE000 ve 0xE001 mesaj kimliklerine sahip olan mesajların gönderilme sıklıkları 100 ms olarak ayarlanmıştır. Bit uzunlukları sabit 128 olarak alınmıştır. Yapılan hesaplamalara göre şarj cihazının hat üzerindeki yoğunluğu %1.024, araç kontrol ünitesinin yoğunluğu %2.048 ve batarya yönetim sisteminin hat

üzerindeki yoğunluğu ise %14.54 olarak bulunmuştur. Tüm düğümlerin hat üzerindeki yoğunluğu ise %17.61 olarak bulunmuştur.

CAN hızının 125 kbit/s olduğu deney düzeneği üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı BYS yazılımında 1 s olarak değiştirilmiştir.

Şekil 4.5'te deney düzeneği ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 0.99 s olduğu görülmektedir. Yani bu mesajlar 1 s sıklıkla hat üzerinde mesaj göndermektedirler. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms'dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.5. Deney düzeneği 125 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü

Şekil 4.6’da 125 kbit/s ve 0xE000 ve 0xE001 mesajlarının 1 s ile gönderildiği deney düzeneğindeki her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %1.15, VCU düğümünün %2.37 ve BMS düğümünün oluşturduğu hat yükü ise %14.23 olduğu saptanmıştır.

Yazılım ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %0.11 ve 0xE001 mesajın hatta olan yoğunluğu ise %0.11 olduğu görülmektedir. 100 ms ile gönderilmiş oldukları test düzeneğinde bu değerler birbirlerinden çok düşük bir oran ile farklıydı, bunun sebebi o an göndermiş oldukları veriler ile ilgilidir.

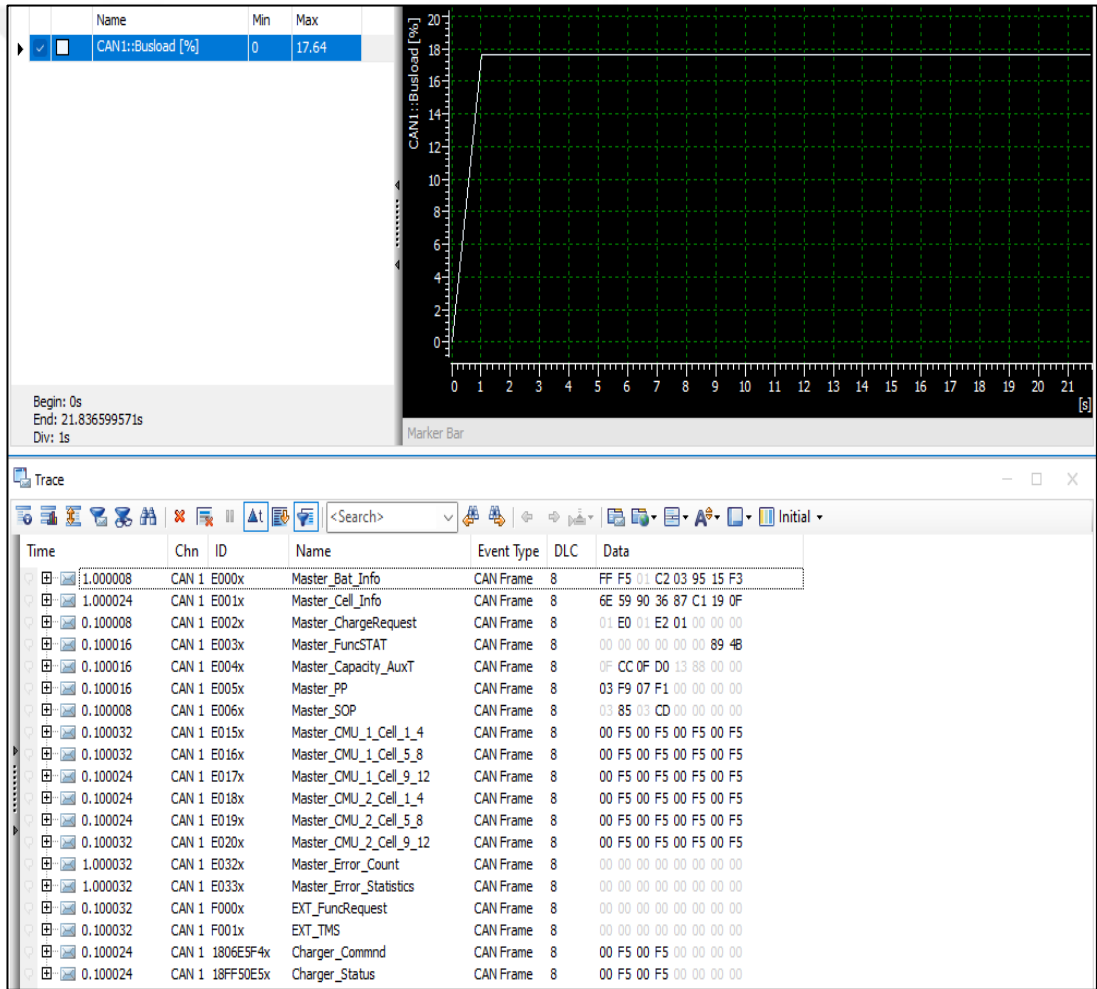
Bu teste elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %17.76 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	17.74	17.73	17.76	17.75
Charger	1.15	1.15	1.15	1.15
Charger_Status (0x18FF50E5x)	1.15	1.15	1.15	1.15
VCU	2.37	2.37	2.37	2.37
EXT_FuncRequest (0xF000x)	1.19	1.19	1.19	1.19
EXT_TMS (0xF001x)	1.18	1.18	1.18	1.18
BMS	14.21	14.20	14.23	14.23
Charger_Commnd (0x1806E5F4x)	1.17	1.17	1.17	1.17
Master_Bat_Info (0xE000x)	0.11	0.11	0.11	0.11
Master_Capacity_AuxT (0xE004x)	1.18	1.17	1.18	1.18
Master_Cell_Info (0xE001x)	0.11	0.11	0.11	0.11
Master_ChargeRequest (0xE002x)	1.19	1.19	1.19	1.19
Master_CMU_1_Cell_1_4 (0xE015x)	1.09	1.09	1.09	1.09
Master_CMU_1_Cell_5_8 (0xE016x)	1.12	1.11	1.12	1.12
Master_CMU_1_Cell_9_12 (0xE017x)	1.17	1.17	1.18	1.17
Master_CMU_2_Cell_1_4 (0xE018x)	1.09	1.09	1.09	1.09
Master_CMU_2_Cell_5_8 (0xE019x)	1.10	1.10	1.11	1.10
Master_CMU_2_Cell_9_12 (0xE020x)	1.19	1.19	1.19	1.19
Master_Error_Count (0xE032x)	0.12	0.12	0.12	0.12
Master_Error_Statistics (0xE033x)	0.12	0.12	0.12	0.12
Master_Errors_About0 (0xE034x)	0.00	0.00	0.00	0.00
Master_Errors_About1 (0xE035x)	0.00	0.00	0.00	0.00
Master_Errors_About2 (0xE036x)	0.00	0.00	0.00	0.00
Master_FuncSTAT (0xE003x)	1.13	1.13	1.13	1.13
Master_PP (0xE005x)	1.18	1.17	1.18	1.18
Master_SOP (0xE006x)	1.17	1.17	1.17	1.17

Şekil 4.6. Deney düzeneği 125 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri

CAN hızının 125 kbit/s olduğu simülasyon programı üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı simülasyon program üzerinden 1 s ile değiştirilmiştir.

Şekil 4.7’de simülasyon programı ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 1 s olduğu görülmektedir. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms’dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.7. Simülasyon 125 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü

Şekil 4.8’de 125 kbit/s ve 0xE000 ve 0xE001 mesajlarının 1 s ile gönderildiği simülasyon programında her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %1.14,

VCU düğümünün %2.38 ve BYS düğümünün oluşturduğu hat yükü ise %14.12 olduğu saptanmıştır.

Simülasyon programı ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %0.11 ve 0xE001 mesajın hatta olan yoğunluğu ise %0.11 olduğu görülmektedir.

Bu testte elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %17.64 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	17.63	17.62	17.64	17.63
Charger	1.14	1.14	1.14	1.14
Charger_Status (0x18FF50E5x)	1.14	1.14	1.14	1.14
VCU	2.38	2.38	2.38	2.38
EXT_FuncRequest (0xF000x)	1.19	1.19	1.19	1.19
EXT_TMS (0xF001x)	1.18	1.18	1.18	1.18
BMS	14.12	14.10	14.12	14.11
Charger_Commnd (0x1806E5F4x)	1.15	1.15	1.15	1.15
Master_Bat_Info (0xE000x)	0.11	0.11	0.11	0.11
Master_Capacity_AuxT (0xE004x)	1.15	1.15	1.16	1.15
Master_Cell_Info (0xE001x)	0.11	0.11	0.11	0.11
Master_ChargeRequest (0xE002x)	1.16	1.16	1.17	1.16
Master_CMU_1_Cell_1_4 (0xE015x)	1.12	1.12	1.12	1.12
Master_CMU_1_Cell_5_8 (0xE016x)	1.12	1.12	1.12	1.12
Master_CMU_1_Cell_9_12 (0xE017x)	1.12	1.12	1.12	1.12
Master_CMU_2_Cell_1_4 (0xE018x)	1.12	1.12	1.12	1.12
Master_CMU_2_Cell_5_8 (0xE019x)	1.12	1.12	1.12	1.12
Master_CMU_2_Cell_9_12 (0xE020x)	1.12	1.12	1.12	1.12
Master_Error_Count (0xE032x)	0.12	0.12	0.12	0.12
Master_Error_Statistics (0xE033x)	0.12	0.12	0.12	0.12
Master_FuncSTAT (0xE003x)	1.16	1.16	1.17	1.16
Master_PP (0xE005x)	1.16	1.16	1.16	1.16
Master_SOP (0xE006x)	1.15	1.15	1.16	1.15

Şekil 4.8. Simülasyon 125 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri

Hat üzerinde bulunan mesajların kimlik bilgileri, gönderme sıklıkları ve hat üzerindeki yoğunlukları gibi bilgiler Çizelge 4.2’de gösterilmiştir.

Çizelge 4.2. Hesaplama 125 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü

Gönderen Düğüm	Mesaj ID	Gönderme Sıklığı(ms)	Bit Uzunluğu	Hat Yüğü (%)
Şarj Cihazı	0x18FF50E5	100	128	1.024
VCU	0xF000	100	128	1.024
VCU	0xF001	100	128	1.024
BYS	0xE000	1000	128	0.1024
BYS	0xE001	1000	128	0.1024
BYS	0xE002	100	128	1.024
BYS	0xE003	100	128	1.024
BYS	0xE004	100	128	1.024
BYS	0xE005	100	128	1.024
BYS	0xE006	100	128	1.024
BYS	0xE015	100	128	1.024
BYS	0xE016	100	128	1.024
BYS	0xE017	100	128	1.024
BYS	0xE018	100	128	1.024
BYS	0xE019	100	128	1.024
BYS	0xE020	100	128	1.024
BYS	0xE032	1000	128	0.1024
BYS	0xE033	1000	128	0.1024
BYS	0x1806E5F4	100	128	1.024
			Toplam Hat Yüğü:	15.7696

0xE000 mesajının hatta olan yoğunluğu şu şekilde hesaplanabilir:

$$t_{bz} = 1 / 125 = 0.008 \text{ ms} \quad (4.7)$$

$$t_{mg} = 0.008 * 128 = 1.024 \text{ ms} \quad (4.8)$$

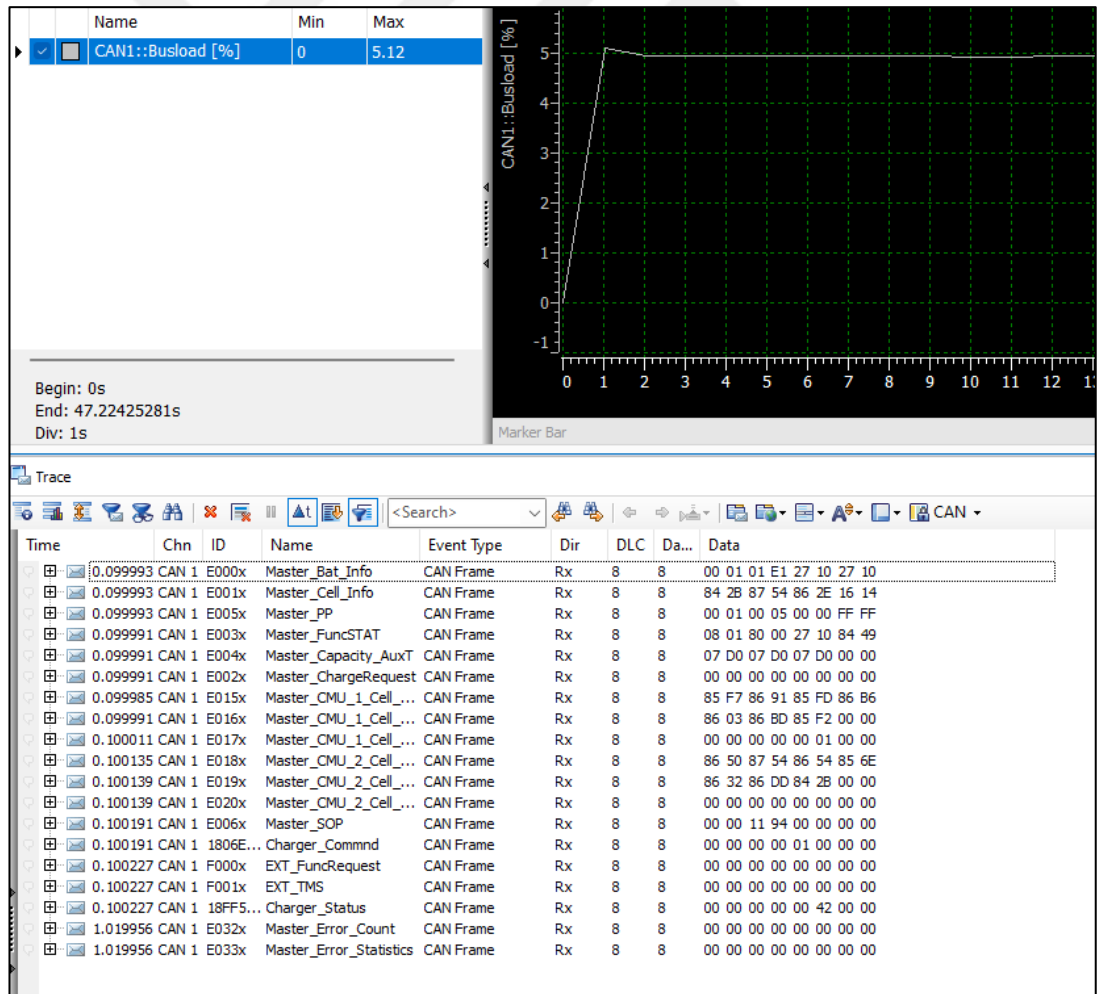
$$M_{bl} = (1.024 / 1000) * 100 = \% 0.1024 \quad (4.9)$$

Buradaki yapılan hesaplamada, 0xE000 ve 0xE001 mesaj kimliklerine sahip olan mesajların gönderilme sıklıkları 1 s olarak ayarlanmıştır. Bit uzunlukları sabit 128 olarak alınmıştır. Yapılan hesaplamara göre şarj cihazının hat üzerindeki yoğunluğu %1.024, araç kontrol ünitesinin yoğunluğu %2.048 ve batarya yönetim sisteminin hat üzerindeki yoğunluğu ise %12.69 olarak bulunmuştur. Tüm düğümlerin hat üzerindeki yoğunluğu ise %15.76 olarak bulunmuştur.

4.2. CAN Hızı 500 kbit/s Analizi

CAN hızının 500 kbit/s olduğu deney düzeneği üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı BYS yazılımında 100 ms olarak değiştirilmiştir.

Şekil 4.9’da deney düzeneği ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 0.099 s olduğu görülmektedir. Yani bu mesajlar 100 ms sıklıkla hat üzerinde mesaj göndermektedirler. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms’dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.9. Deney düzeneği 500 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü

Şekil 4.10'da 500 kbit/s ve 0xE000 ve 0xE001 mesajlarının 100 ms ile gönderildiği deney düzeneğindeki her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %0.29, VCU düğümünün %0.59 ve BYS düğümünün oluşturduğu hat yükü ise %4.27 olduğu saptanmıştır.

Yazılım ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %0.30 ve 0xE001 mesajın hatta olan yoğunluğu ise %0.29 olduğu görülmektedir. Aynı gönderme sıklığına sahip olmasına rağmen birbirlerinden farklı sonuç vermesinin temel sebebi göndermiş olduğu verilerin birbirlerinden farklı olması ve bit doldurma işlemine göre mesaj bit uzunluğunun değişmesidir.

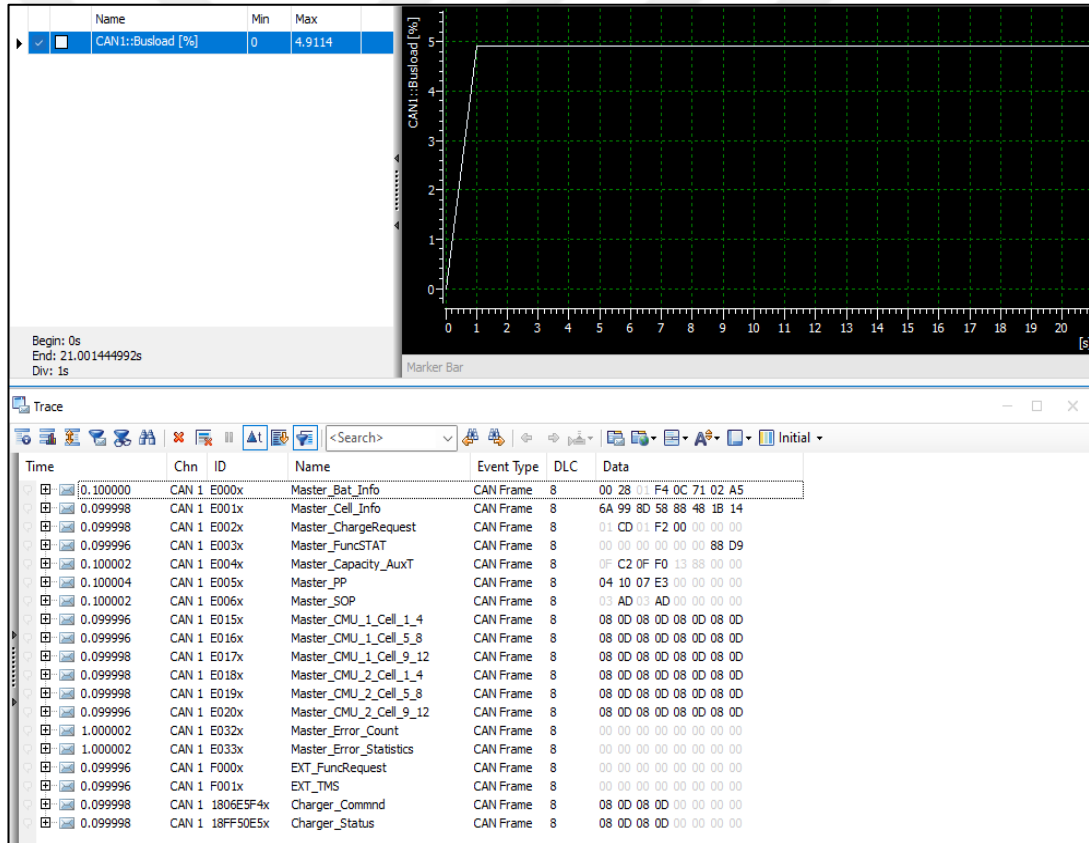
Bu testte elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %5.12 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	4.94	4.88	5.12	4.94
Charger	0.29	0.28	0.29	0.29
Charger_Status (0x18FF50E5x)	0.29	0.28	0.29	0.29
VCU	0.59	0.57	0.59	0.59
EXT_FuncRequest (0xF000x)	0.30	0.29	0.30	0.30
EXT_TMS (0xF001x)	0.30	0.28	0.30	0.30
BMS	4.06	4.00	4.27	4.06
Charger_Commnd (0x1806E5F4x)	0.29	0.29	0.31	0.29
Master_Bat_Info (0xE000x)	0.29	0.29	0.30	0.29
Master_Capacity_AuxT (0xE004x)	0.29	0.29	0.31	0.29
Master_Cell_Info (0xE001x)	0.27	0.27	0.29	0.27
Master_ChargeRequest (0xE002x)	0.30	0.30	0.31	0.30
Master_CMU_1_Cell_1_4 (0xE015x)	0.27	0.27	0.29	0.27
Master_CMU_1_Cell_5_8 (0xE016x)	0.28	0.28	0.30	0.28
Master_CMU_1_Cell_9_12 (0xE017x)	0.29	0.29	0.31	0.29
Master_CMU_2_Cell_1_4 (0xE018x)	0.27	0.27	0.29	0.27
Master_CMU_2_Cell_5_8 (0xE019x)	0.28	0.27	0.29	0.28
Master_CMU_2_Cell_9_12 (0xE020x)	0.30	0.30	0.31	0.30
Master_Error_Count (0xE032x)	0.03	0.00	0.03	0.03
Master_Error_Statistics (0xE033x)	0.03	0.00	0.03	0.03
Master_Errors_About0 (0xE034x)	0.00	0.00	0.00	0.00
Master_Errors_About1 (0xE035x)	0.00	0.00	0.00	0.00
Master_Errors_About2 (0xE036x)	0.00	0.00	0.00	0.00
Master_FuncSTAT (0xE003x)	0.28	0.28	0.30	0.28
Master_PP (0xE005x)	0.29	0.29	0.31	0.29
Master_SOP (0xE006x)	0.29	0.29	0.31	0.29

Şekil 4.10. Deney düzeneği 500 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri

CAN hızının 500 kbit/s olduğu simülasyon programı üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı simülasyon program üzerinden 100 ms ile değiştirilmiştir.

Şekil 4.11’de simülasyon programı ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 0.100 s olduğu görülmektedir. Yani bu mesajlar 100 ms sıklıkla hat üzerinde mesaj göndermektedirler. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms’dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.11. Simülasyon 500 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü

Şekil 4.12’de 500 kbit/s ve 0xE000 ve 0xE001 mesajlarının 100 ms ile gönderildiği simülasyon programında her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %0.28, VCU düğümünün %0.59 ve BYS düğümünün oluşturduğu hat yükü ise %4.03 olduğu saptanmıştır.

Simülasyon programı ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %0.28 ve 0xE001 mesajın hatta olan yoğunluğu ise %0.28 olduğu görülmektedir.

Bu testte elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %4.91 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	4.91	4.91	4.91	4.91
Charger	0.28	0.28	0.28	0.28
Charger_Status (0x18FF50E5x)	0.28	0.28	0.28	0.28
VCU	0.59	0.59	0.59	0.59
EXT_FuncRequest (0xF000x)	0.30	0.30	0.30	0.30
EXT_TMS (0xF001x)	0.30	0.30	0.30	0.30
BMS	4.03	4.03	4.03	4.03
Charger_Commnd (0x1806E5F4x)	0.29	0.29	0.29	0.29
Master_Bat_Info (0xE000x)	0.28	0.28	0.28	0.28
Master_Capacity_AuxT (0xE004x)	0.29	0.29	0.29	0.29
Master_Cell_Info (0xE001x)	0.27	0.27	0.28	0.27
Master_ChargeRequest (0xE002x)	0.29	0.29	0.29	0.29
Master_CMU_1_Cell_1_4 (0xE015x)	0.28	0.28	0.28	0.28
Master_CMU_1_Cell_5_8 (0xE016x)	0.28	0.28	0.28	0.28
Master_CMU_1_Cell_9_12 (0xE017x)	0.28	0.28	0.28	0.28
Master_CMU_2_Cell_1_4 (0xE018x)	0.28	0.28	0.28	0.28
Master_CMU_2_Cell_5_8 (0xE019x)	0.28	0.28	0.28	0.28
Master_CMU_2_Cell_9_12 (0xE020x)	0.28	0.28	0.28	0.28
Master_Error_Count (0xE032x)	0.03	0.03	0.03	0.03
Master_Error_Statistics (0xE033x)	0.03	0.03	0.03	0.03
Master_FuncSTAT (0xE003x)	0.29	0.29	0.29	0.29
Master_PP (0xE005x)	0.29	0.29	0.29	0.29
Master_SOP (0xE006x)	0.29	0.29	0.29	0.29

Şekil 4.12. Simülasyon 500 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri

Hat üzerinde bulunan mesajların kimlik bilgileri, gönderme sıklıkları ve hat üzerindeki yoğunlukları gibi bilgiler Çizelge 4.3'te gösterilmiştir.

Çizelge 4.3. Hesaplama 500 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü

Gönderen Düğüm	Mesaj ID	Gönderme Sıklığı(ms)	Bit Uzunluğu	Hat Yüğü (%)
Şarj Cihazı	0x18FF50E5	100	128	0.256
VCU	0xF000	100	128	0.256
VCU	0xF001	100	128	0.256
BYS	0xE000	100	128	0.256
BYS	0xE001	100	128	0.256
BYS	0xE002	100	128	0.256
BYS	0xE003	100	128	0.256
BYS	0xE004	100	128	0.256
BYS	0xE005	100	128	0.256
BYS	0xE006	100	128	0.256
BYS	0xE015	100	128	0.256
BYS	0xE016	100	128	0.256
BYS	0xE017	100	128	0.256
BYS	0xE018	100	128	0.256
BYS	0xE019	100	128	0.256
BYS	0xE020	100	128	0.256
BYS	0xE032	1000	128	0.0256
BYS	0xE033	1000	128	0.0256
BYS	0x1806E5F4	100	128	0.256
			Toplam Hat Yüğü:	4.4032

0xE000 mesajının hatta olan yoğunluğu şu şekilde hesaplanabilir:

$$t_{bz} = 1 / 500 = 0.002 \text{ ms} \quad (4.10)$$

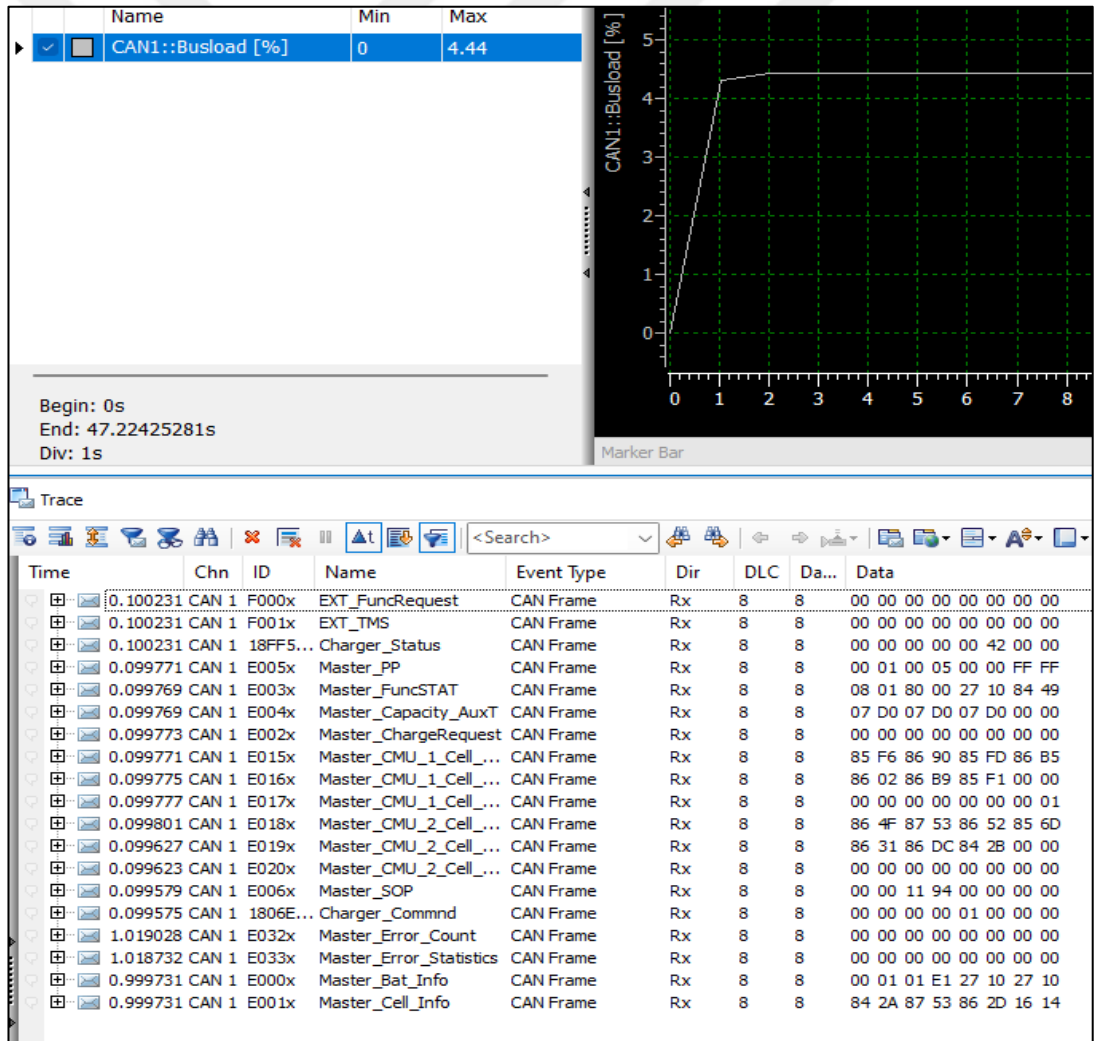
$$t_{mg} = 0.002 * 128 = 0.256 \text{ ms} \quad (4.11)$$

$$M_{bl} = (0.256 / 100) * 100 = \% 0.256 \quad (4.12)$$

Buradaki yapılan hesaplamada, 0xE000 ve 0xE001 mesaj kimliklerine sahip olan mesajların gönderilme sıklıkları 100 ms olarak ayarlanmıştır. Bit uzunlukları sabit 128 olarak alınmıştır. Yapılan hesaplamara göre şarj cihazının hat üzerindeki yoğunluğu %0.256, araç kontrol ünitesinin yoğunluğu %0.512 ve batarya yönetim sisteminin hat üzerindeki yoğunluğu ise %3.635 olarak bulunmuştur. Tüm düğümlerin hat üzerindeki yoğunluğu ise %4.40 olarak bulunmuştur.

CAN hızının 500 kbit/s olduğu deney düzeneği üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı BYS yazılımında 1 s olarak değiştirilmiştir.

Şekil 4.13'te deney düzeneği ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 0.99 s olduğu görülmektedir. Yani bu mesajlar 1 s sıklıkla hat üzerinde mesaj göndermektedirler. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms'dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.13. Deney düzeneği 500 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü

Şekil 4.14'te 500 kbit/s ve 0xE000 ve 0xE001 mesajlarının 1 s ile gönderildiği deney düzeneğindeki her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %0.29, VCU düğümünün %0.59 ve BMS düğümünün oluşturduğu hat yükü ise %3.56 olduğu saptanmıştır.

Yazılım ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %0.03 ve 0xE001 mesajın hatta olan yoğunluğu ise %0.03 olduğu görülmektedir.

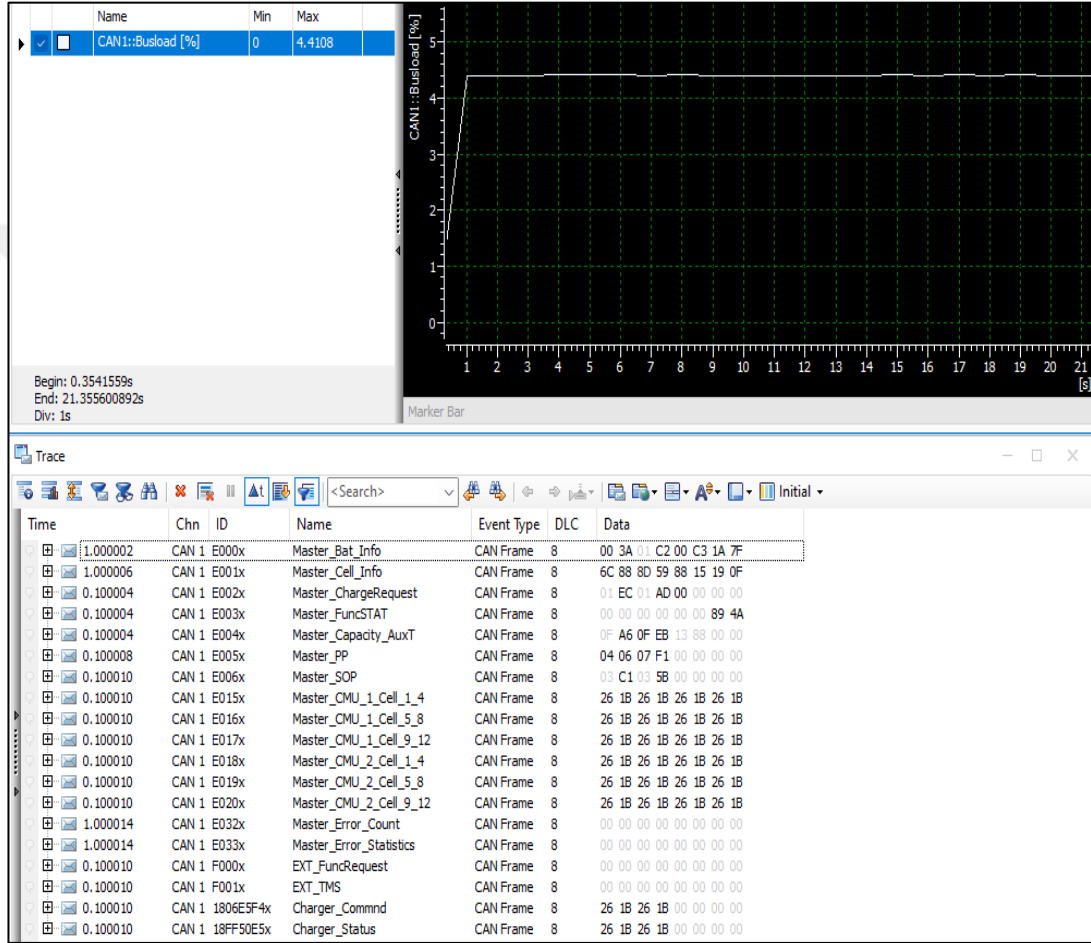
Bu testte elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %4.44 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	4.44	4.31	4.44	4.42
Charger	0.29	0.28	0.29	0.29
Charger_Status (0x18FF50E5x)	0.29	0.28	0.29	0.29
VCU	0.59	0.58	0.59	0.59
EXT_FuncRequest (0xF000x)	0.30	0.29	0.30	0.30
EXT_TMS (0xF001x)	0.30	0.29	0.30	0.29
BMS	3.56	3.45	3.56	3.55
Charger_Commnd (0x1806E5F4x)	0.29	0.28	0.29	0.29
Master_Bat_Info (0xE000x)	0.03	0.03	0.03	0.03
Master_Capacity_AuxT (0xE004x)	0.29	0.29	0.29	0.29
Master_Cell_Info (0xE001x)	0.03	0.03	0.03	0.03
Master_ChargeRequest (0xE002x)	0.30	0.29	0.30	0.30
Master_CMU_1_Cell_1_4 (0xE015x)	0.27	0.27	0.28	0.27
Master_CMU_1_Cell_5_8 (0xE016x)	0.28	0.27	0.28	0.28
Master_CMU_1_Cell_9_12 (0xE017x)	0.29	0.29	0.29	0.29
Master_CMU_2_Cell_1_4 (0xE018x)	0.27	0.27	0.27	0.27
Master_CMU_2_Cell_5_8 (0xE019x)	0.27	0.27	0.28	0.27
Master_CMU_2_Cell_9_12 (0xE020x)	0.30	0.29	0.30	0.30
Master_Error_Count (0xE032x)	0.03	0.03	0.03	0.03
Master_Error_Statistics (0xE033x)	0.03	0.03	0.03	0.03
Master_Errors_About0 (0xE034x)	0.00	0.00	0.00	0.00
Master_Errors_About1 (0xE035x)	0.00	0.00	0.00	0.00
Master_Errors_About2 (0xE036x)	0.00	0.00	0.00	0.00
Master_FuncSTAT (0xE003x)	0.28	0.27	0.28	0.28
Master_PP (0xE005x)	0.29	0.29	0.29	0.29
Master_SOP (0xE006x)	0.29	0.28	0.29	0.29

Şekil 4.14. Deney düzeneği 500 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri

CAN hızının 500 kbit/s olduğu simülasyon programı üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı simülasyon program üzerinden 1 s ile değiştirilmiştir.

Şekil 4.15'te simülasyon programı ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 1.00 s olduğu görülmektedir. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms'dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.15. Simülasyon 500 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü

Şekil 4.16'da 500 kbit/s ve 0xE000 ve 0xE001 mesajlarının 1 s ile gönderildiği simülasyon programında her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %0.28, VCU düğümünün %0.59 ve BYS düğümünün oluşturduğu hat yükü ise %3.53 olduğu saptanmıştır.

Simülasyon programı ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %0.03 ve 0xE001 mesajın hatta olan yoğunluğu ise %0.03 olduğu görülmektedir.

Bu testte elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %4.41 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	4.41	4.40	4.41	4.41
Charger	0.28	0.28	0.28	0.28
Charger_Status (0x18FF50E5x)	0.28	0.28	0.28	0.28
VCU	0.59	0.59	0.59	0.59
EXT_FuncRequest (0xF000x)	0.30	0.30	0.30	0.30
EXT_TMS (0xF001x)	0.30	0.30	0.30	0.30
BMS	3.53	3.53	3.53	3.53
Charger_Commnd (0x1806E5F4x)	0.29	0.29	0.29	0.29
Master_Bat_Info (0xE000x)	0.03	0.03	0.03	0.03
Master_Capacity_AuxT (0xE004x)	0.29	0.29	0.29	0.29
Master_Cell_Info (0xE001x)	0.03	0.03	0.03	0.03
Master_ChargeRequest (0xE002x)	0.29	0.29	0.29	0.29
Master_CMU_1_Cell_1_4 (0xE015x)	0.28	0.28	0.28	0.28
Master_CMU_1_Cell_5_8 (0xE016x)	0.28	0.28	0.28	0.28
Master_CMU_1_Cell_9_12 (0xE017x)	0.28	0.28	0.28	0.28
Master_CMU_2_Cell_1_4 (0xE018x)	0.28	0.28	0.28	0.28
Master_CMU_2_Cell_5_8 (0xE019x)	0.28	0.28	0.28	0.28
Master_CMU_2_Cell_9_12 (0xE020x)	0.28	0.28	0.28	0.28
Master_Error_Count (0xE032x)	0.03	0.03	0.03	0.03
Master_Error_Statistics (0xE033x)	0.03	0.03	0.03	0.03
Master_FuncSTAT (0xE003x)	0.29	0.29	0.29	0.29
Master_PP (0xE005x)	0.29	0.29	0.29	0.29
Master_SOP (0xE006x)	0.29	0.29	0.29	0.29

Şekil 4.16. Simülasyon 500 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri

Hat üzerinde bulunan mesajların kimlik bilgileri, gönderme sıklıkları ve hat üzerindeki yoğunlukları gibi bilgiler Çizelge 4.4'te gösterilmiştir.

Çizelge 4.4. Hesaplama 500 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü

Gönderen Düğüm	Mesaj ID	Gönderme Sıklığı(ms)	Bit Uzunluğu	Hat Yüğü (%)
Şarj Cihazı	0x18FF50E5	100	128	0.256
VCU	0xF000	100	128	0.256
VCU	0xF001	100	128	0.256
BYS	0xE000	1000	128	0.0256
BYS	0xE001	1000	128	0.0256
BYS	0xE002	100	128	0.256
BYS	0xE003	100	128	0.256
BYS	0xE004	100	128	0.256
BYS	0xE005	100	128	0.256
BYS	0xE006	100	128	0.256
BYS	0xE015	100	128	0.256
BYS	0xE016	100	128	0.256
BYS	0xE017	100	128	0.256
BYS	0xE018	100	128	0.256
BYS	0xE019	100	128	0.256
BYS	0xE020	100	128	0.256
BYS	0xE032	1000	128	0.0256
BYS	0xE033	1000	128	0.0256
BYS	0x1806E5F4	100	128	0.256
			Toplam Hat Yüğü:	3.9424

0xE000 mesajının hatta olan yoğunluğu şu şekilde hesaplanabilir:

$$t_{bz} = 1 / 500 = 0.002 \text{ ms} \quad (4.13)$$

$$t_{mg} = 0.002 * 128 = 0.256 \text{ ms} \quad (4.14)$$

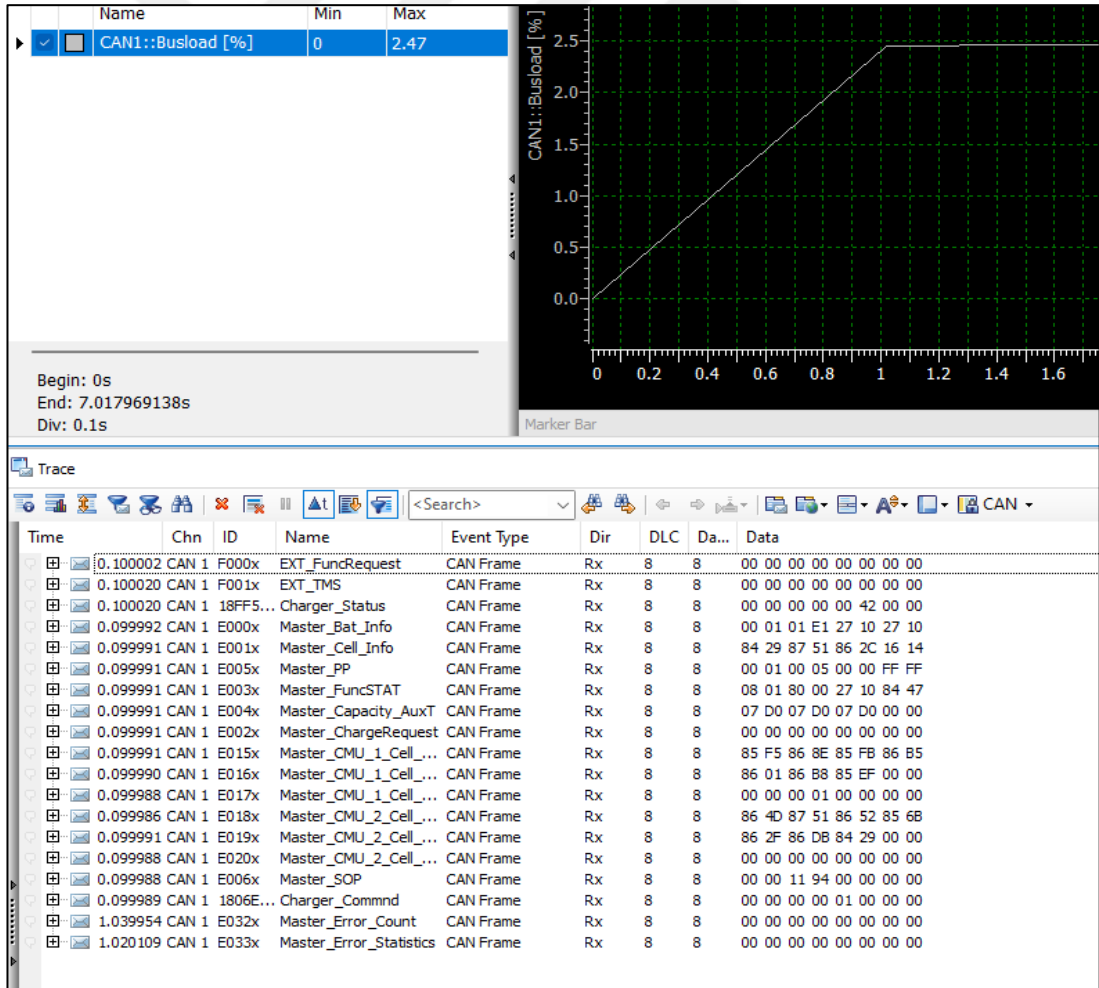
$$M_{bl} = (0.256 / 1000) * 100 = \% 0.0256 \quad (4.15)$$

Buradaki yapılan hesaplamada, 0xE000 ve 0xE001 mesaj kimliklerine sahip olan mesajların gönderilme sıklıkları 1 s olarak ayarlanmıştır. Bit uzunlukları sabit 128 olarak alınmıştır. Yapılan hesaplamara göre şarj cihazının hat üzerindeki yoğunluğu %0.256, araç kontrol ünitesinin yoğunluğu %0.512 ve batarya yönetim sisteminin hat üzerindeki yoğunluğu ise %3.174 olarak bulunmuştur. Tüm düğümlerin hat üzerindeki yoğunluğu ise %3.94 olarak bulunmuştur.

4.3. CAN Hızı 1000 kbit/s Analizi

CAN hızının 1000 kbit/s olduğu deney düzeneği üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı BYS yazılımında 100 ms olarak değiştirilmiştir.

Şekil 4.17’de deney düzeneği ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 0.099 s olduğu görülmektedir. Yani bu mesajlar 100 ms sıklıkla hat üzerinde mesaj göndermektedirler. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms’dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.17. Deney düzeneği 1000 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü

Şekil 4.18’de 1000 kbit/s ve 0xE000 ve 0xE001 mesajlarının 100 ms ile gönderildiği deney düzeneğindeki her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %0.14, VCU düğümünün %0.30 ve BMS düğümünün oluşturduğu hat yükü ise %2.03 olduğu saptanmıştır.

Yazılım ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %0.14 ve 0xE001 mesajın hatta olan yoğunluğu ise %0.14 olduğu görülmektedir.

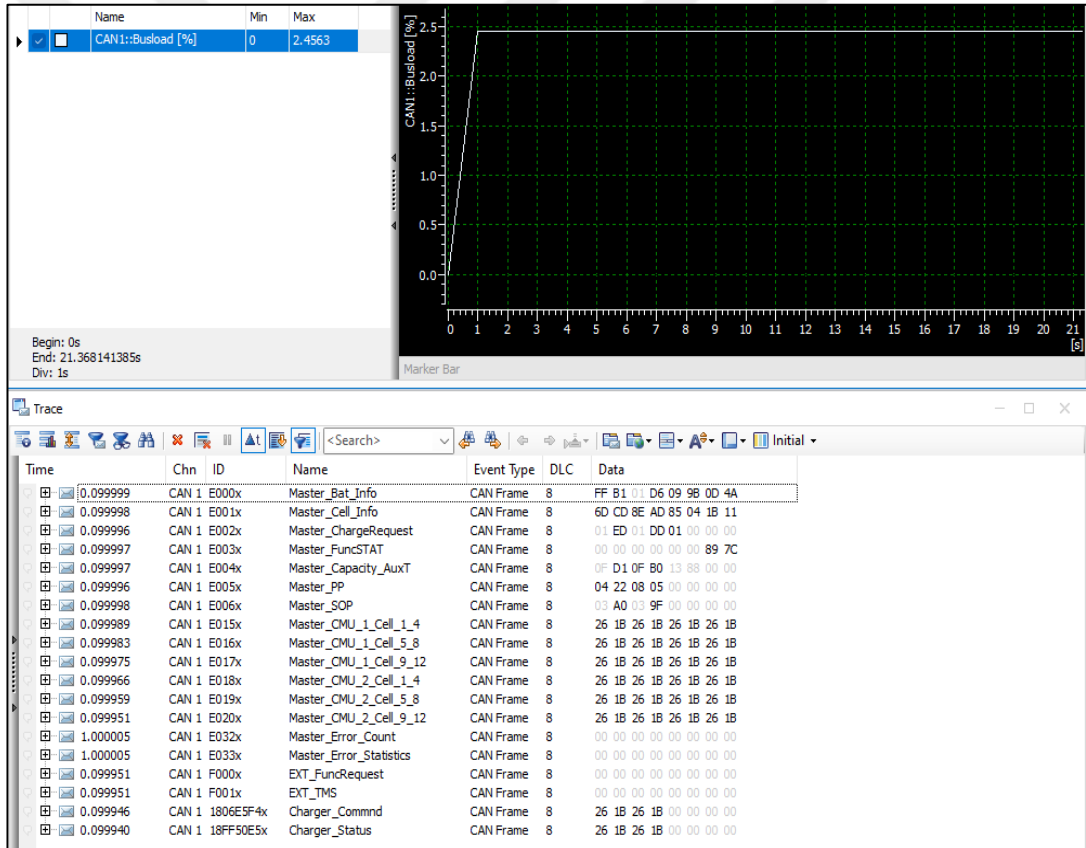
Bu testte elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %2.47 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	2.47	2.45	2.47	2.47
Charger	0.14	0.14	0.14	0.14
Charger_Status (0x18FF50E5x)	0.14	0.14	0.14	0.14
VCU	0.30	0.29	0.30	0.30
EXT_FuncRequest (0xF000x)	0.15	0.15	0.15	0.15
EXT_TMS (0xF001x)	0.15	0.15	0.15	0.15
BMS	2.03	2.01	2.03	2.03
Charger_Commnd (0x1806E5F4x)	0.15	0.14	0.15	0.15
Master_Bat_Info (0xE000x)	0.14	0.14	0.14	0.14
Master_Capacity_AuxT (0xE004x)	0.15	0.15	0.15	0.15
Master_Cell_Info (0xE001x)	0.14	0.13	0.14	0.14
Master_ChargeRequest (0xE002x)	0.15	0.15	0.15	0.15
Master_CMU_1_Cell_1_4 (0xE015x)	0.14	0.14	0.14	0.14
Master_CMU_1_Cell_5_8 (0xE016x)	0.14	0.14	0.14	0.14
Master_CMU_1_Cell_9_12 (0xE017x)	0.15	0.15	0.15	0.15
Master_CMU_2_Cell_1_4 (0xE018x)	0.14	0.13	0.14	0.14
Master_CMU_2_Cell_5_8 (0xE019x)	0.14	0.14	0.14	0.14
Master_CMU_2_Cell_9_12 (0xE020x)	0.15	0.15	0.15	0.15
Master_Error_Count (0xE032x)	0.01	0.01	0.01	0.01
Master_Error_Statistics (0xE033x)	0.01	0.01	0.01	0.01
Master_Errors_About0 (0xE034x)	0.00	0.00	0.00	0.00
Master_Errors_About1 (0xE035x)	0.00	0.00	0.00	0.00
Master_Errors_About2 (0xE036x)	0.00	0.00	0.00	0.00
Master_FuncSTAT (0xE003x)	0.14	0.14	0.14	0.14
Master_PP (0xE005x)	0.15	0.15	0.15	0.15
Master_SOP (0xE006x)	0.15	0.14	0.15	0.15

Şekil 4.18. Deney düzeneği 1000 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri

CAN hızının 1000 kbit/s olduğu simülasyon programı üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı simülasyon program üzerinden 100 ms ile değiştirilmiştir.

Şekil 4.19’da simülasyon programı ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 0.100 s olduğu görülmektedir. Yani bu mesajlar 100 ms sıklıkla hat üzerinde mesaj göndermektedirler. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms’dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.19. Simülasyon 1000 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü

Şekil 4.20’de 125 kbit/s ve 0xE000 ve 0xE001 mesajlarının 100 ms ile gönderildiği simülasyon programında her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %0.14, VCU düğümünün %0.30 ve BYS düğümünün oluşturduğu hat yükü ise %2.02 olduğu saptanmıştır.

Simülasyon programı ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %0.14 ve 0xE001 mesajın hatta olan yoğunluğu ise %0.14 olduğu görülmektedir.

Bu testte elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %2.46 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	2.45	2.45	2.46	2.45
Charger	0.14	0.14	0.14	0.14
Charger_Status (0x18FF50E5x)	0.14	0.14	0.14	0.14
VCU	0.30	0.30	0.30	0.30
EXT_FuncRequest (0xF000x)	0.15	0.15	0.15	0.15
EXT_TMS (0xF001x)	0.15	0.15	0.15	0.15
BMS	2.02	2.01	2.02	2.01
Charger_Commnd (0x1806E5F4x)	0.14	0.14	0.14	0.14
Master_Bat_Info (0xE000x)	0.14	0.14	0.14	0.14
Master_Capacity_AuxT (0xE004x)	0.14	0.14	0.14	0.14
Master_Cell_Info (0xE001x)	0.14	0.14	0.14	0.14
Master_ChargeRequest (0xE002x)	0.14	0.14	0.15	0.14
Master_CMU_1_Cell_1_4 (0xE015x)	0.14	0.14	0.14	0.14
Master_CMU_1_Cell_5_8 (0xE016x)	0.14	0.14	0.14	0.14
Master_CMU_1_Cell_9_12 (0xE017x)	0.14	0.14	0.14	0.14
Master_CMU_2_Cell_1_4 (0xE018x)	0.14	0.14	0.14	0.14
Master_CMU_2_Cell_5_8 (0xE019x)	0.14	0.14	0.14	0.14
Master_CMU_2_Cell_9_12 (0xE020x)	0.14	0.14	0.14	0.14
Master_Error_Count (0xE032x)	0.01	0.01	0.01	0.01
Master_Error_Statistics (0xE033x)	0.01	0.01	0.01	0.01
Master_FuncSTAT (0xE003x)	0.15	0.15	0.15	0.15
Master_PP (0xE005x)	0.15	0.14	0.15	0.14
Master_SOP (0xE006x)	0.14	0.14	0.14	0.14

Şekil 4.20. Simülasyon 1000 kbit/s ve 100 ms gönderme sıklığı ile düğümlerin hat yükleri

Hat üzerinde bulunan mesajların kimlik bilgileri, gönderme sıklıkları ve hat üzerindeki yoğunlukları gibi bilgiler Çizelge 4.5'te gösterilmiştir.

Çizelge 4.5. Hesaplama 1000 kbit/s ve 100 ms gönderme sıklığı ile toplam hat yükü

Gönderen Düğüm	Mesaj ID	Gönderme Sıklığı(ms)	Bit Uzunluğu	Hat Yüğü (%)
Şarj Cihazı	0x18FF50E5	100	128	0.128
VCU	0xF000	100	128	0.128
VCU	0xF001	100	128	0.128
BYS	0xE000	100	128	0.128
BYS	0xE001	100	128	0.128
BYS	0xE002	100	128	0.128
BYS	0xE003	100	128	0.128
BYS	0xE004	100	128	0.128
BYS	0xE005	100	128	0.128
BYS	0xE006	100	128	0.128
BYS	0xE015	100	128	0.128
BYS	0xE016	100	128	0.128
BYS	0xE017	100	128	0.128
BYS	0xE018	100	128	0.128
BYS	0xE019	100	128	0.128
BYS	0xE020	100	128	0.128
BYS	0xE032	1000	128	0.0128
BYS	0xE033	1000	128	0.0128
BYS	0x1806E5F4	100	128	0.128
			Toplam Hat Yüğü:	2.2016

0xE000 mesajının hatta olan yoğunluğu şu şekilde hesaplanabilir:

$$t_{bz} = 1 / 1000 = 0.001 \text{ ms} \quad (4.16)$$

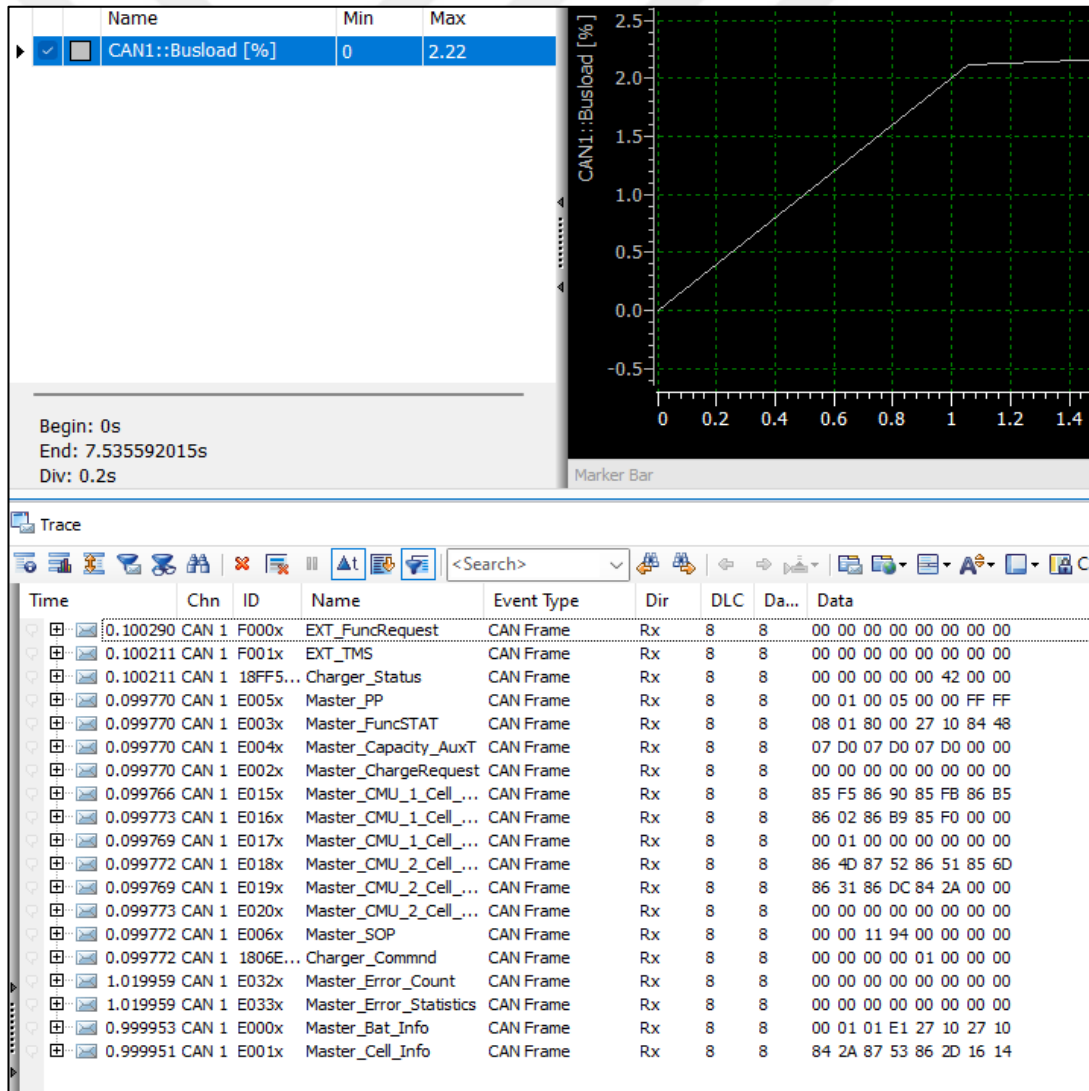
$$t_{mg} = 0.001 * 128 = 0.128 \text{ ms} \quad (4.17)$$

$$M_{bl} = (0.128 / 100) * 100 = \% 0.128 \quad (4.18)$$

Buradaki yapılan hesaplamada, 0xE000 ve 0xE001 mesaj kimliklerine sahip olan mesajların gönderilme sıklıkları 100 ms olarak ayarlanmıştır. Bit uzunlukları sabit 128 olarak alınmıştır. Yapılan hesaplamara göre şarj cihazının hat üzerindeki yoğunluğu %0.128, araç kontrol ünitesinin yoğunluğu %0.256 ve batarya yönetim sisteminin hat üzerindeki yoğunluğu ise %1.8176 olarak bulunmuştur. Tüm düğümlerin hat üzerindeki yoğunluğu ise %2.20 olarak bulunmuştur.

CAN hızının 1000 kbit/s olduğu deney düzeneği üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı BYS yazılımında 1 s olarak değiştirilmiştir.

Şekil 4.21’de deney düzeneği ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 0.99 s olduğu görülmektedir. Yani bu mesajlar 1 s sıklıkla hat üzerinde mesaj göndermektedirler. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms’dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.21. Deney düzeneği 1000 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü

Şekil 4.22’de 1000 kbit/s ve 0xE000 ve 0xE001 mesajlarının 1 s ile gönderildiği deney düzeneğindeki her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %0.14, VCU düğümünün %0.30 ve BMS düğümünün oluşturduğu hat yükü ise %1.78 olduğu saptanmıştır.

Yazılım ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %0.01 ve 0xE001 mesajın hatta olan yoğunluğu ise %0.01 olduğu görülmektedir.

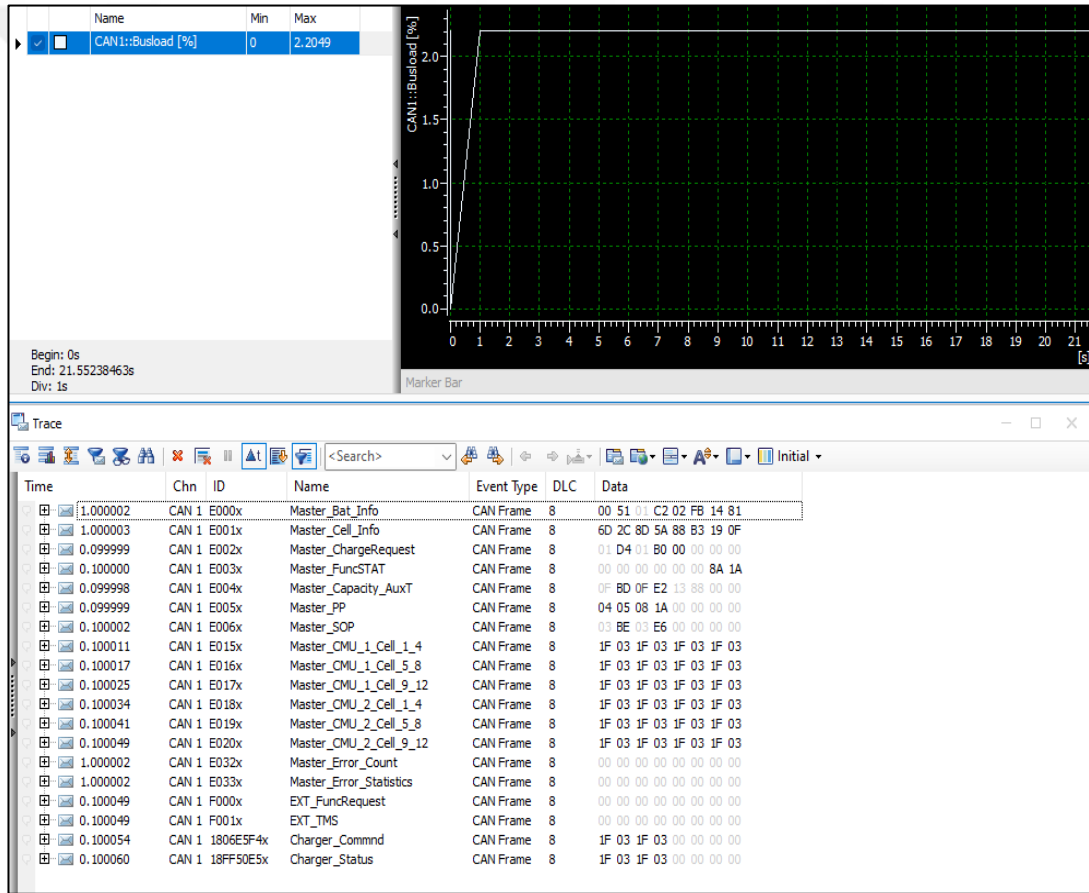
Bu testte elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %2.22 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	2.22	2.12	2.22	2.21
Charger	0.14	0.14	0.14	0.14
Charger_Status (0x18FF50E5x)	0.14	0.14	0.14	0.14
VCU	0.30	0.28	0.30	0.29
EXT_FuncRequest (0xF000x)	0.15	0.14	0.15	0.15
EXT_TMS (0xF001x)	0.15	0.14	0.15	0.15
BMS	1.78	1.70	1.78	1.77
Charger_Commnd (0x1806E5F4x)	0.15	0.14	0.15	0.15
Master_Bat_Info (0xE000x)	0.01	0.01	0.01	0.01
Master_Capacity_AuxT (0xE004x)	0.15	0.14	0.15	0.15
Master_Cell_Info (0xE001x)	0.01	0.01	0.01	0.01
Master_ChargeRequest (0xE002x)	0.15	0.14	0.15	0.15
Master_CMU_1_Cell_1_4 (0xE015x)	0.14	0.13	0.14	0.14
Master_CMU_1_Cell_5_8 (0xE016x)	0.14	0.13	0.14	0.14
Master_CMU_1_Cell_9_12 (0xE017x)	0.15	0.14	0.15	0.15
Master_CMU_2_Cell_1_4 (0xE018x)	0.14	0.13	0.14	0.13
Master_CMU_2_Cell_5_8 (0xE019x)	0.14	0.13	0.14	0.14
Master_CMU_2_Cell_9_12 (0xE020x)	0.15	0.14	0.15	0.15
Master_Error_Count (0xE032x)	0.01	0.01	0.01	0.01
Master_Error_Statistics (0xE033x)	0.01	0.01	0.01	0.01
Master_Errors_About0 (0xE034x)	0.00	0.00	0.00	0.00
Master_Errors_About1 (0xE035x)	0.00	0.00	0.00	0.00
Master_Errors_About2 (0xE036x)	0.00	0.00	0.00	0.00
Master_FuncSTAT (0xE003x)	0.14	0.14	0.14	0.14
Master_PP (0xE005x)	0.15	0.14	0.15	0.15
Master_SOP (0xE006x)	0.15	0.14	0.15	0.15

Şekil 4.22. Deney düzeneği 1000 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri

CAN hızının 1000 kbit/s olduğu simülasyon programı üzerinde yapılan testte CAN hattına gönderilen 0xE000 ve 0xE001 mesaj kimliklerine sahip mesajların gönderilme sıklığı simülasyon program üzerinden 1 s ile değiştirilmiştir.

Şekil 4.23'te simülasyon programı ile hazırlanan sistemin CAN hattı üzerindeki oluşturduğu hat yoğunluğunun grafiği gösterilmektedir. Buradaki şekilde Time olarak adlandırılan sütunda 0xE000 ve 0xE001 mesajlarının gönderilme sıklığı 1.00 s olduğu görülmektedir. Diğer düğümlerin göndermiş olduğu mesaj sıklıkları ise 100 ms'dir. Sadece 0xE032 ve 0xE033 mesajları 1 s olarak göndermektedir. Bu mesajların gönderilme sıklıkları her zaman aynıdır ve hiçbir testte değiştirilmemiştir.



Şekil 4.23. Simülasyon 1000 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü

Şekil 4.24'te 1000 kbit/s ve 0xE000 ve 0xE001 mesajlarının 1 s ile gönderildiği simülasyon programında her bir mesajın ayrı ayrı hat üzerinde oluşturduğu yükler gösterilmektedir. Şarj cihazının hat üzerinde oluşturduğu maksimum hat yükü %0.14, VCU düğümünün %0.30 ve BYS düğümünün oluşturduğu hat yükü ise %1.77 olduğu saptanmıştır.

Simülasyon programı ile gönderme sıklığı değiştirilen 0xE000 mesajının hat üzerindeki yoğunluğu %0.01 ve 0xE001 mesajın hatta olan yoğunluğu ise %0.01 olduğu görülmektedir.

Bu testte elde edilen verilere göre tüm düğümlerin CAN hattında oluşturduğu toplam hat yükü maksimum %2.20 olduğu görülmüştür.

Statistic	Current / Last	Min	Max	Avg
Busload [%]	2.20	2.20	2.20	2.20
Charger	0.14	0.14	0.14	0.14
Charger_Status (0x18FF50E5x)	0.14	0.14	0.14	0.14
VCU	0.30	0.30	0.30	0.30
EXT_FuncRequest (0xF000x)	0.15	0.15	0.15	0.15
EXT_TMS (0xF001x)	0.15	0.15	0.15	0.15
BMS	1.77	1.76	1.77	1.76
Charger_Commnd (0x1806E5F4x)	0.14	0.14	0.14	0.14
Master_Bat_Info (0xE000x)	0.01	0.01	0.01	0.01
Master_Capacity_AuxT (0xE004x)	0.14	0.14	0.14	0.14
Master_Cell_Info (0xE001x)	0.01	0.01	0.01	0.01
Master_ChargeRequest (0xE002x)	0.15	0.14	0.15	0.15
Master_CMU_1_Cell_1_4 (0xE015x)	0.14	0.14	0.14	0.14
Master_CMU_1_Cell_5_8 (0xE016x)	0.14	0.14	0.14	0.14
Master_CMU_1_Cell_9_12 (0xE017x)	0.14	0.14	0.14	0.14
Master_CMU_2_Cell_1_4 (0xE018x)	0.14	0.14	0.14	0.14
Master_CMU_2_Cell_5_8 (0xE019x)	0.14	0.14	0.14	0.14
Master_CMU_2_Cell_9_12 (0xE020x)	0.14	0.14	0.14	0.14
Master_Error_Count (0xE032x)	0.01	0.01	0.01	0.01
Master_Error_Statistics (0xE033x)	0.01	0.01	0.01	0.01
Master_FuncSTAT (0xE003x)	0.15	0.15	0.15	0.15
Master_PP (0xE005x)	0.15	0.14	0.15	0.15
Master_SOP (0xE006x)	0.14	0.14	0.14	0.14

Şekil 4.24. Simülasyon 1000 kbit/s ve 1 s gönderme sıklığı ile düğümlerin hat yükleri

Hat üzerinde bulunan mesajların kimlik bilgileri, gönderme sıklıkları ve hat üzerindeki yoğunlukları gibi bilgiler Çizelge 4.6’da gösterilmiştir.

Çizelge 4.6. Hesaplama 1000 kbit/s ve 1 s gönderme sıklığı ile toplam hat yükü

Gönderen Düğüm	Mesaj ID	Gönderme Sıklığı(ms)	Bit Uzunluğu	Hat Yüğü (%)
Şarj Cihazı	0x18FF50E5	100	128	0.128
VCU	0xF000	100	128	0.128
VCU	0xF001	100	128	0.128
BYS	0xE000	1000	128	0.0128
BYS	0xE001	1000	128	0.0128
BYS	0xE002	100	128	0.128
BYS	0xE003	100	128	0.128
BYS	0xE004	100	128	0.128
BYS	0xE005	100	128	0.128
BYS	0xE006	100	128	0.128
BYS	0xE015	100	128	0.128
BYS	0xE016	100	128	0.128
BYS	0xE017	100	128	0.128
BYS	0xE018	100	128	0.128
BYS	0xE019	100	128	0.128
BYS	0xE020	100	128	0.128
BYS	0xE032	1000	128	0.0128
BYS	0xE033	1000	128	0.0128
BYS	0x1806E5F4	100	128	0.128
			Toplam Hat Yüğü:	1.9712

0xE000 mesajının hatta olan yoğunluğu şu şekilde hesaplanabilir:

$$t_{bz} = 1 / 1000 = 0.001 \text{ ms} \quad (4.19)$$

$$t_{mg} = 0.001 * 128 = 0.128 \text{ ms} \quad (4.20)$$

$$M_{bl} = (0.128 / 1000) * 100 = \% 0.0128 \quad (4.21)$$

Buradaki yapılan hesaplamada, 0xE000 ve 0xE001 mesaj kimliklerine sahip olan mesajların gönderilme sıklıkları 100 ms olarak ayarlanmıştır. Bit uzunlukları sabit 128 olarak alınmıştır. Yapılan hesaplamara göre şarj cihazının hat üzerindeki yoğunluğu %0.128, araç kontrol ünitesinin yoğunluğu %0.256 ve batarya yönetim sisteminin hat üzerindeki yoğunluğu ise %1.5872 olarak bulunmuştur. Tüm düğümlerin hat üzerindeki yoğunluğu ise %1.97 olarak bulunmuştur.

4.4. Tüm Hızlardaki Genel Analiz

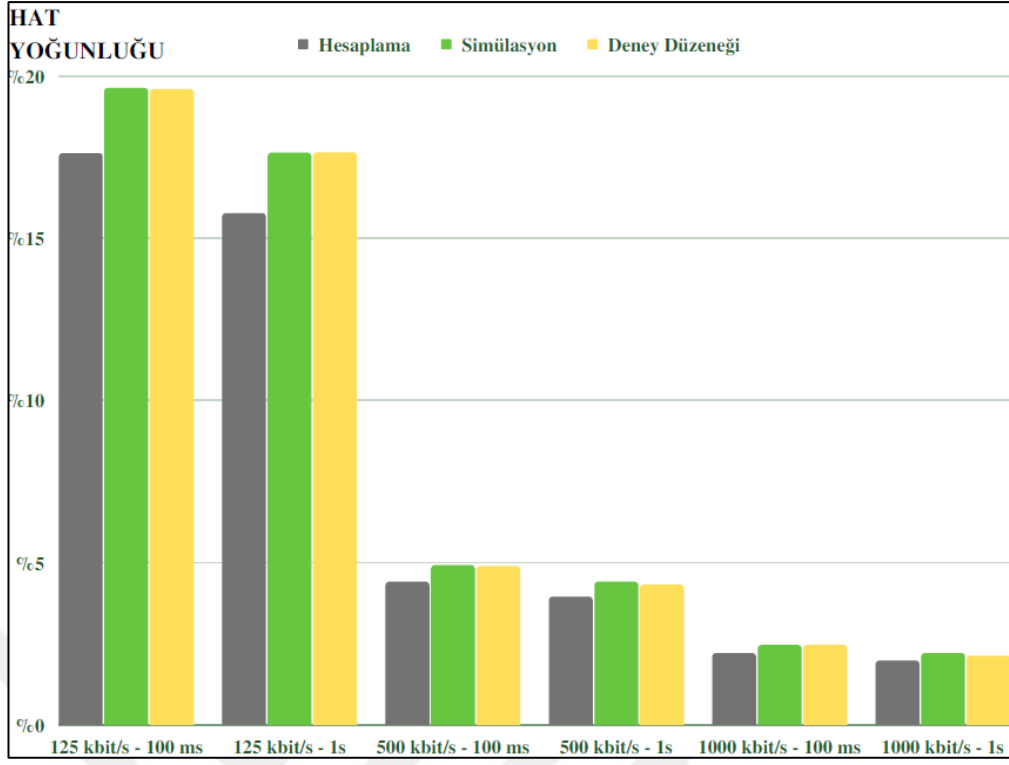
Farklı CAN hızlarında ve farklı mesaj gönderme sıklıklarında yapılan testlerin sonuçları Çizelge 4.7’de ve Şekil 4.25’te gösterilmiştir.

Çizelge 4.7. Toplam hat yükleri

CAN hızı	Gönderilme sıklığı	Toplam hat yükü		
		Hesaplama	Simülasyon	Deney Düzeneği
125 kbit/s	100 ms	%17.61	%19.62	%19.58
	1 s	%15.76	%17.62	%17.73
500 kbit/s	100 ms	%4.40	%4.91	%4.88
	1 s	%3.94	%4.40	%4.31
1000 kbit/s	100 ms	%2.20	%2.45	%2.45
	1 s	%1.97	%2.20	%2.12

Hesaplama işlemi yapılırken bit doldurma işlemi yok sayılmıştır ve mesaj bit uzunluğu 128 bit olarak alınmıştır. Bundan dolayı yapılan hesaplama işleminde elde edilen sonuç minimum değerdir. Çizelge 4.7’deki veriler, tüm yapılan testlerden elde edilen sonuçların minimum değerleridir.

Hesaplama yöntemiyle yapılan sonuçlardan elde edilen verilere göre 125 kbit/s, 100 ms mesaj gönderme sıklığında hattın toplam yoğunluğu %17.61’dir. Simülasyon sonucunda bu değer %19.62, deney düzeneğinde ise %19.58 olarak bulunmuştur.



Şekil 4.25. Toplam hat yükleri çubuk grafiđi

Hat yoğunluđu, 3 farklı CAN hızlarında ve mesaj sıklıkları deđiştirilerek yapılan test sonuçlarından elde edilen veriler Şekil 4.25'te gösterilmiştir.

Farklı renklerle gösterilen çubuk grafiđinde, gri renk ile gösterilen hesaplama yöntemini, yeşil renk simülasyon sonuçlarını ve sarı renk ile gösterilen veriler ise deneş düzeneđinden elde edilen verileri göstermektedir.

5. TARTIŞMA VE SONUÇ

Yapılan bu çalışmada batarya yönetim sistemi, şarj cihazı ve araç kontrol ünitesinin bulunduğu CAN hattına bağlantı sağlanmıştır. Bu CAN hattında bulunan düğümler birbirlerine belirli bir mesajlar gönderirler ve böylelikle birbirleri ile iletişim halinde olurlar. Yapılan analizlerle beraber farklı CAN hızlarında verilerin hatta olan yoğunluğu incelenmiştir. 3 farklı hızlarda belirlenen bu deney düzeneğinde her bir hızda farklı bir yoğunluk bulunmuştur ve bunun hesaplaması yapılmıştır.

Bütün yapılan testlerde sadece iki tane mesajın gönderilme sıklığı 100 ms ve 1 s olarak değiştirilmiştir. Diğer mesajların gönderilme sıklıkları sabit tutulmuştur. Hat yükü hesabı yapılırken her bir mesajın hat üzerindeki yoğunluğu hesaplanır ve bunların hepsinin toplamı genel hat yükü olarak kabul edilir. Hesaplama işleminde bulunan hat yükü, simülasyon ve deney düzeneği üzerinde bulunan hat yükü ile farklı sonuç çıkmaktadır. Bunun en önemli nedenlerinden birisi CAN protokolünün bit doldurma işlemini yapmasıdır. Gelen her verinin değişmesine göre bit uzunluğu değişmektedir. Bu yüzden hesaplama işlemi yapılırken mesajların bit uzunluğu sabit 128 olarak alınmıştır. Simülasyon ve deney düzeneği üzerindeki yoğunluk birbirine yakın çıkmaktadır. Bunun sebebi ise simülasyon ortamında yapılan testlerde bit doldurma işlemi program üzerinden hesaplanmıştır. Ancak buna rağmen tamamen aynı sonuç çıkması beklenemez çünkü deney düzeneği üzerindeki mesajların göndermiş olduğu veriler sürekli olarak değişmektedir. Bunun sonucunda ise bit doldurma işlemi sürekli olarak değişmektedir.

CAN hızının belirlenmesinde önemli olan faktör genel olarak düğümlerinin birbirine olan uzaklıklarıdır. CAN hızı seçildikten sonra mesajların gönderilme sıklıkları verilerin önem sırasına göre ayarlanabilir. Yapılmış olan deney düzeneğinde BYS düğümünün hata mesajları saniyede bir defa göndermektedir. Diğer veriler 100 ms ile diğer düğümlere gönderilmektedir. Yapılan testlerde 0xE000 ve 0xE001 adreslerinden gelen sinyallerin gönderilme sıklıkları değiştirilmiştir ve hatta olan yükleri analiz edilmiştir. CAN hattının yoğunluğu maksimum %50 veya %60 civarlarında olması kabul edilebilir. Tasarlanan bu deney düzeneğinde 125 kbit/s hızında maksimum %20.69 olmuştur ve bu yoğunlukta kabul edilebilir bir yüzdendir. Ancak düğümlerin

fazlaşması veya CAN hattına gönderilen mesajların artmasıyla bu yüzde deęiřecektir.

CAN hattının yoğunluęu ve BYS ile ilgili farklı alıřmalar bulunmaktadır. Bu alıřmalarda ya sadece batarya ynetim sistemi geliřtirilmiř veya sadece CAN hattının yoğunluęu hesaplanmıřtır. Hesaplanan bu hat yoğunlukları belirli bir sistem zerindeki yapılan hesaplamalardır. Yani CAN hattının hızlarının deęiřtirilmedięi veya mesajların gönderilme sıklıklarının sabit olduęu alıřmalardır. Ancak yapılmıř olan bu tezde, bataryanın gvenli bir řekilde alıřabileceęi bir batarya ynetim sistemi tasarlanıp CAN hattı zerindeki her bir mesajın yoğunluęu hesaplanmıřtır.



KAYNAKLAR

- Aktaş, M., Baygüneş, B., Kıvrak, S., Çavuş, B. & Sözen, F. (2020). Elektrikli Araç İçin Düşük Maliyetli Bir Batarya Yönetim Sistemi Tasarımı ve Gerçekleştirilmesi. *Avrupa Bilim ve Teknoloji Dergisi, Özel Sayı*, 227-238. doi.org/10.31590/ejosat.779720
- Al Hayani, M. (2012). Modeling Bus Load on CAN. <http://urn.kb.se/resolve?urn=urn:nbn:se:hh:diva-19140> (Son erişim tarihi: 05.05.2021)
- Aldoğan, E. (2012). *Elektrikli Araçlar İçin Batarya Yönetim Sistemi Donanımı Tasarımı*. (Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü)
- Altun, Ş. (2012). *Elektrikli Araçlar İçin Batarya Yönetim Sistemi Algoritması Tasarımı ve Uygulanması*. (Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü Mühendislik ve Fen Bilimleri Enstitüsü)
- Anonim, (2009). TENERGY 18650 2200mAh Li-Ion Cell. <https://datasheetspdf.com/pdf-file/1408721/TENERGY/18650/1> (Son erişim tarihi: 03.05.2021)
- Anonim, (2010). Lithium-ion Battery Datasheet LIR18650 2600mAh. <https://www.ineltro.ch/media/downloads/SAItem/45/45958/36e3e7f3-2049-4adb-a2a7-79c654d92915.pdf> (Son erişim tarihi: 03.05.2021)
- Anonim, (2011). Controllor Area Network. https://en.wikipedia.org/wiki/CAN_bus (Son erişim tarihi: 05.05.2021)
- Anonim, (2015). What do CAN Bus Signals Look Like? https://e2e.ti.com/blogs_/b/industrial_strength/archive/2015/06/04/what-docan-bus-signals-look-like (Son erişim tarihi: 05.05.2021)
- Baki, S. (2021). *Otomotiv Haberleşmesinde Denetleyici Alan Ağı (CAN) için Hibrit Bir Saldırı Savuşturma Uygulaması*. (Yüksek Lisans Tezi, İstanbul Aydın Üniversitesi Lisansüstü Eğitim Enstitüsü)
- Baygüneş, B. (2019). *Elektrikli Araçlarda Batarya Yönetim Sistemi*. (Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü)
- Bayılmış, C. (2006). IEEE 802.11B KLAN Kullanarak CAN Segmentleri Genişleten Arabağlaşım Birimi Tasarımı. (Doktora Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü)
- Bosch, (1991). *CAN Specification, Version 2.0*, Robert Bosch GmbH, Stuttgart.
- Bozdal, M., Samie, M., Aslam, S. & Jennions, I. (2020). Evaluation of can bus security challenges. *Sensors*, 20(8), 2364.

- Daowd, M., Omar, N., Van Den Bossche, P. & Van Mierlo, J., (2011). *A Review of Passive and Active Battery Balancing based on MATLAB/Simulink (in press). International Review of Electrical Engineering, 6(7), 2974-2989.*
- Di Natale, M. (2000). *Scheduling the CAN bus with earliest deadline techniques.* In Proceedings 21st IEEE Real-Time Systems Symposium. 27-30 November, Orlando, FL, USA, 259-268.
- Ekici, Y.E. (2019). *Batarya Yönetim Sistemleri.* (Yüksek Lisans Tezi, İnönü Üniversitesi Fen Bilimleri Enstitüsü)
- Ekiz, H. (1997). *Design, Implementation, and Performance Analysis of CAN/CAN and CAN/ethernet Bridges.* (Doctoral Dissertation, University of Sussex)
- Gül, H.S. (2018). *Elektrikli Araçlar İçin Batarya Yönetim Sistemi Tasarımı.* (Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü)
- Kara, İ. (2009). *CAN Haberleşme Protokolünün İncelenmesi ve Bir Sıcaklık Kontrol Sistemine Uygulanması.* (Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü)
- Kelkar, S. & Kamal, R. (2014). Adaptive fault diagnosis algorithm for controller area network. *IEEE transactions on Industrial Electronics, 61(10), 5527-5537.*
- Kıvrak, S., Özer, T. & Oğuz, Y. (2020). STM32f103C8 Mikroişlemcisi kullanılarak elektrikli araçlar için pasif dengeleme metodu tabanlı batarya yönetim sistemi tasarımı ve uygulaması. *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi, 20(3), 426-433.* doi.org/10.35414/akufemubid.691456
- Kiencke, U. (1994). *Controller Area Network - From Concept to Reality.* In Proceedings of 1st International CAN Conference, Can in Automation (CIA). 04-07 October, Erlangen, 11-20.
- Kutlu, A. (2004). MicroLab: a web-based multi-user remote microcontroller laboratory for engineering education. *International Journal of Engineering Education, 20(5), 879-885.*
- Lawrenz, W. (1995). *Worldwide Status of CAN—Present and Future.* In Proceedings of 2nd International CAN Conference, CAN in Automation (CIA), Erlangen. 13-14 October, London. 2-16.
- Lawrenz, W. (1997). *CAN System Engineering: From Theory to Practical Applications.* Springer, New York.
- Li, F., Wang, L. & Liao, C. (2008). *CAN (Controller Area Network) Bus Communication System Based on Matlab/Simulink.* In 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing. 12-14 October, Dalian, China, 1-4.

- Li, R., Liu, C. & Luo, F. (2008). *A Design for Automotive CAN Bus Monitoring System*. In 2008 IEEE Vehicle Power and Propulsion Conference. 03-05 September, Harbin, China, 1-5.
- Lu, L., Han, X., Li, J., Hua, J. & Ouyang, M. (2012). A review on the key issues for lithium-ion battery management in electric vehicles. *Journal of Power Sources*, 226, 272-288.
- Murphy, N. (2016). A short trip on the CAN bus, *Embedded Systems Design*. http://www.embedded.com/columns/murphyslaw/13000304?_requestid=639577 (Son erişim tarihi: 05.05.2021)
- Murway, P.S. & Groza, B. (2014). Source identification using signal characteristics in controller area networks. *IEEE Signal Processing Letters*, 21(4), 395-399. doi: 10.1109/LSP.2014.2304139.
- Nafiz, S. (2014). *Elektrikli Otobüsler İçin Batarya Yönetim Sistemi Tasarımı*. (Yüksek Lisans Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü)
- Özçelik, İ., Ertürk, İ. & Ekiz, H. (2001). CAN – *Ethernet Uyumlu Köprü Tasarımı ve Uygulaması*. EEBM 9. Ulusal Kongresi. 17-22 Eylül, İzmit, 264-268.
- Pazul, K. (1999). *Controller Area Network (CAN) Basics*. Microchip Technology Inc.
- Prajapati, V., Hess, H., William, E. J., Gupta, V., Huff, M., Manic, M., Rufus, F., Thakker, A. & Govar, J. (2011). *A Literature Review of State of-Charge Estimation Techniques Applicable to Lithium Poly-Carbon Monofluoride (Li/CFx) Battery*. India International Conference on Power Electronics 2010 (IICPE2010). 28-30 January, New Delhi, India, 1-8.
- Rahimi-Eichi, H., Ojha, U., Baronti, F. & Chow, M. Y. (2013). Battery management system: An overview of its application in the smart grid and electric vehicles. *IEEE Industrial Electronics Magazine*, 7(2), 4-16.
- Richardson, P., Seih, L. & Haniak, P. (2001). *A Real-Time Control Network Protocol for Embedded Systems Using Controller Area Network (CAN)*. In IEEE Electronics and Information Technology Conference. 06-08 June, Oakland University, 1-15.
- Rohrer, R.A., Pitla, S.K. & Luck, J.D. (2019). Tractor CAN bus interface tools and application development for real-time data analysis. *Computers and Electronics in Agriculture*, 163, 104847.
- Tindell, K., Burns, A. & Wellings, A.J. (1995). Calculating controller area network (CAN) message response times. *Control Engineering Practice*, 3(8), 1163-1169.
- Tindell, K., Hanssmon, H. & Wellings, A.J. (1994). *Analysing Real-Time Communications: Controller Area Network (CAN)*. 1994 Proceedings Real-Time Systems Symposium. 07-09 December, San Juan, PR, USA, 259-263.

- Tosun, G. (2019). *Batarya Yönetim Sistemi Tasarımı ve Uygulanması*. (Yüksek Lisans Tezi, İstanbul Okan Üniversitesi Fen ve Mühendislik Enstitüsü)
- Ünal, İ. (2006). *CAN (Control Area Network) Üzerinden PIC Programlama*. (Doktora Tezi, Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü)
- Yavuz, E., Sarıca, S. & Artuç, E. (2018). *Kontrol alan ağları için optimum statik mesaj zamanlaması*. *Mühendislik Bilimleri ve Tasarım Dergisi*, 6(3), 532-540. doi: 10.21923/jesd.442915
- Yılmaz, E. (2017). *Internet of Things Based Battery Management Systems Applications*. (Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü)
- Zuberi, K.M. & Shin, K.G. (1995). *Non – Preemptive Scheduling of Messages on Control Area Network for Real – Time Applications*, In Proc. Of the IEEE Real Time Applications Symposium. 15-17 May, Chicago, IL, USA, 240–249.

ÖZGEÇMİŞ

