



Mikrodenetleyici ile CanBus Haberleşme Ağı Üzerinden Veri Depolama Sistemi Uygulaması ve Performans Analizi

Sercan Ozan NUR^{1*} , Ahmet KARACA²

¹ Lisansüstü Eğitim Enstitüsü, Sakarya Uygulamalı Bilimler Üniversitesi, Türkiye. y215004004@subu.edu.tr

² Teknoloji Fakültesi, Sakarya Uygulamalı Bilimler Üniversitesi, Türkiye. akaraca@subu.edu.tr

ÖZ

Otomotiv sektörü günümüz teknolojisinin en büyük fırsatlarından biridir. Otomobiller her zaman hayatımızın vazgeçilmez ekipmanlarının başındaki yerini koruyor. Bu sistemler hem mekanik hem de elektronik olarak çok büyük bir altyapıya ve tasarıma sahiptir. İnsanlar için ön planda olan bu ekipmanların gereksinimlerinde güvenlik önceliklidir. Bu nedenle her bir algılayıcının doğru çalışması ve sistem içerisinde birbirleriyle çok hassas bir şekilde iletişim kurmaları gerekir. Otomobillerde¹ ki bu güvenli haberleşme CanBus haberleşme sistemi ile yapılmaktadır. Bu makalenin amacı, otomobillerdeki CanBus iletişim sistemleri üzerinden algılayıcı verilerine erişmektir. Verilerin kaydedilmesi ve gerektiğinde erişilerek görselleştirilmesidir.

Bu çalışmada, arıza tespiti amacıyla otomobillerdeki CanBus üzerinden yayınlanan algılayıcı verilerinin gerçek zamanlı olarak okunması için Arduino Uno R3 ile bir Veri Kayıt Sistemi tasarlanmıştır. Araçlardaki CanBus ağını deneysel olarak oluşturmak amacıyla da yine Arduino Uno R3 ile bir Motor Kontrol Ünitesi Simülatörü oluşturulmuştur. Motor Kontrol Ünitesi Simülatörü tarafından üretilen algılayıcı verilerinin, Veri Kayıt Sistemi tarafından CanBus üzerinden okunarak bir SD kartta depolanması sağlanmıştır. SD kartta kaydedilen verilerin grafik ortamına aktararak görselleştirilmesi için Visual Studio 2019 üzerinden C# programlama dili ile arayüz uygulaması yazılmıştır. Görselleştirilen veriler grafik ortamında incelenerek verilerde oluşabilecek hatalar incelenebilmektedir.

Anahtar Kelimeler: İletişim, CanBus, kontrolör alan ağı, can protokolü, veri toplama, veri depolama sistem uygulaması, mikrodenetleyici.

Data Storage System Application and Performance Analysis over CanBus Communication Network with Microcontroller

ABSTRACT

The automotive sector is one of the biggest opportunities of today's technology. Automobiles always keep is one of the irreplaceable equipment of our lives. These systems have a very large

*Sorumlu yazar: y215004004@subu.edu.tr

infrastructure and design, both mechanically and electronically. Safety is a priority in the requirements of these equipment, which is at the forefront for people. For this reason, each sensor must work correctly and communicate with one another in a very sensitive way within the system. This secure communication in automobiles is made with the CanBus communication system. The goal of this essay is to access sensor data via CanBus communication systems in automobiles. It is the recording of data and visualization by accessing it when necessary.

In this article, a Data Recording System is designed with Arduino Uno R3 for real-time reading of sensor data broadcast over CanBus in cars for fault detection. In order to create the CanBus network in the vehicles experimentally, an Engine Control Unit Simulator was created with the Arduino Uno R3. The sensor data produced by the Engine Control Unit Simulator is read by the Data Recording System over CanBus and stored on an SD card. In order to visualize the data recorded on the SD card by transferring it to the graphical environment, an interface application was written with the C # programming language over Visual Studio 2019. The visualized data can be examined in the graphical environment and the mistakes that may arise in the information can be examined.

Keywords: Communication, CanBus, controller area network, can protocol, data acquisition, data storage system application, microcontroller.

1 Giriş

Elektronik sistemler, konfor ve güvenlik amacıyla son yıllarda giderek büyüyen bir şekilde araç içi eski mekanik ve hidrolik muadil parçaların yerini almaktadır [1]. Örneğin, günümüzün modern otomobillerinde kilitlemeyen fren sistemi (ABS), elektronik hidrolik direksiyon (EPS), kayma önleyici düzenleme (ASR), elektronik denge programı (ESP), elektronik fren gücü dağılımı (EBD) ve ayarlanabilir hız sabitleyici (ACC) gibi elektronik kontrol sistemleri daha fazla konfor ve güvenlik sağlayarak sürücünün kontrolü elinde tutmasına yardımcı olur. Benzer şekilde, bir otomobilin gövde alanına ait cihazlar (ön panel, farlar, silecekler, kapılar ve camlar) ile iletişim ve eğlence ekipmanları (radyo, navigasyon sistemleri, cep telefonları) elektronik sistemler tarafından kontrol edilmektedir [1].

Günümüzün modern araçları, frenleme, vites değiştirme ve direksiyon gibi birçok temel işlevi yerine getiren düzinelerce Elektronik Kontrol Ünitesi (ECU) içerir. Ayrıca otonom sürüş, Araç Çevre Bağlantısı (V2I) ve Araçlar Arası İletişim (V2V) gibi işlevlerin de ECU'lar tarafından yerine getirilmesi beklenmektedir. Araçtaki ECU'ları bağlamak için Yerel Ara Bağlantı Ağı (LIN) [2], Denetleyici Alan Ağı (CAN) [3] ve FlexRay [4] gibi çeşitli protokoller tanımlanmış ve uygulanmıştır. CAN, otomotiv endüstrisinde en yaygın kullanılan araç içi iletişim protokolüdür.

CAN protokolü, 1980'lerde otomotiv sistem üreticilerinden biri olan Bosch tarafından geliştirilmiştir. Haberleşmedeki güvenilirliğinden dolayı otomotiv sistemlerinde kullanılmaktadır [5]. CAN hatlı sistemler, iletim hattı üzerinden merkezi olarak idare edilmesi prensibinin çalışmasına dayanır. Merkezi yönetici pozisyonundaki CPU'ya iletilen verilere göre sonuç mekanizması bu CPU tarafından işletilmektedir [6]. CAN protokolü olmadan merkezi olarak kontrol edilmek istenen elektronik sistemler için çok sayıda kablo kullanılması gerekmektedir [7]. Can protokolünün kullanılması ile tüm sistemler çift sarmal kablo yardımıyla can hattına bağlanarak kablo karmaşası ortadan kaldırılabilir. Böylece tüm sistemler birbiriyle haberleşebilir [8].

Ünal, 2006 senesinde yazdığı yüksek lisans tezinde “CAN Üzerinden PIC Programlama” çalışmasıyla sanayilerde elektronik yapılara müdahale için kullanılan mikrodenetleyicilerin denetimini ve bilgisayar ile programlanabilmesini yapan bir sistem gerçekleştirilmiştir. Bu tez çalışmasında CAN ağı üzerinden bir mikrodenetleyicinin kod belleği değiştirilerek, mikrodenetleyici üzerinde çalışan programın güncellenmesi sağlanmıştır [9].

Kara, 2009 senesinde yazdığı yüksek lisans tezinde "CAN Haberleşme Protokolünün İncelenmesi Ve Bir Sıcaklık Kontrol Sistemine Uygulanması" çalışmasıyla CAN ağının öteki haberleşme ağlarına göre üstün olduğu durumları ve üstün olmadığı durumların belirlenmesi için CAN protokolü ile bir sıcaklık kontrol uygulaması gerçekleştirmiştir. Bu çalışmada PID tabanlı sıcaklık kontrol sistemi gerçekleştirilmiş ve bu sistemin CAN ağı üzerinden kontrolü sağlanmıştır [10].

Dinçer, 2010 senesinde yazdığı yüksek lisans tezinde "CanBus İle Dağıtık Kontrol Uygulaması" çalışmasıyla model asansörün CanBus teknolojisi kullanılarak dağıtık kontrolü uygulaması işlemlerini gerçekleştirmiştir. Bu tezde model asansörün CanBus ile yaparak dağıtık denetimini uygun olarak sonuçlandırmıştır. Böylece CanBus ağının dağıtık denetim çalışmalarında uygun bir sistem olduğu belirlenmiştir [11].

Öztürk, 2010 senesinde yazdığı yüksek lisans tezinde "Hidrojen Hibrit Otobüslerde CanBus Sistemleri Ve Uygulaması" çalışmasıyla hibrit otobüsler için üç tane algılayıcı kullanılarak CanBus ağından aracın kapısının açık yada kapalı olup olmadığını, motor sıcaklığının belirlenmesini ve kapıdan binen kişi miktarının belirlenmesi sağlanmıştır. Çalışmanın sonunda Mercedes Citoro model otobüslerin CanBus ağıyla ELFA sistem ekipmanları aracılığıyla bir Hidrojen Hibrit otobüs projelendirilmiştir. Ayrıca motor yağ derecesinin belirlenmesi ile fan başlatma ya da kesme devresi ile giren kişilerin miktarlarının belirlenmesi devresi imalat edilerek CanBus ile iletişim kurdurulup neticenin ekranda gösterilmesi işlemi yapılmıştır [12].

Derse, 2017 senesinde yazdığı yüksek lisans tezinde "Yeni Bir CanBus İletişim Tabanlı Otomotiv Güç Dağıtım Modülü Uygulaması Ve Sistemin Güvenirlilik Analizi: eaPDM" çalışmasıyla kuvvet yayılım modülü imalat sürecinden ve sistemde gerçekleştirilen güvenlik deneyimlerinden bahsedilmiştir. CAN ağında çalışan yeni bir modülün tam anlamıyla hata analizini yapabilen ve bütün giriş ile çıkış değerlerini sisteme işleyebilen bir modelleme yapılmıştır [13].

Karanfil, 2020 senesinde yazdığı yüksek lisans tezinde "OBD II Protokollerini Kullanan Bir Arayüz Yazılımı Tasarım" çalışmasıyla ECU'ya iletilen verileri yazma, okuma ve tanı koyabilen bir yazılımı ve tasarımı işlemlerini gerçekleştirmiştir. Çalışmasında ELM327 makinesi ile SAE J1850, ISO 14230-4 ve ISO 9141-2 iletişim protokolleri aracılığıyla iletişim ağındaki bilgileri okuyup bu verilerin tanısını yapan programlama ve tasarım işlemi yapılmıştır. Bu çalışmada LabVIEW programı ile arayüz tasarımı gerçekleştirilmiştir. LabVIEW programı ile yapılan bu arayüz yazılımının Mobydic4910 simülatörü ile kontrolü sağlanmıştır [14].

Topuz, 2021 senesinde yazdığı yüksek lisans tezinde "Hibrit Araçlar İçin CanBus Haberleşme Birimi İle Araç Kontrol Sistemi Ve Linux Tabanlı GUI Tasarımı" çalışması yapmıştır. Günümüzde çok fazla tercih edilmeye başlanan elektrikli ya da hibrit taşıtlar için oluşturulan taşıt içi ağ ile algılayıcılardan CanBus aracılığıyla edinilen verinin, dizayn edilen gösterge paneli aracılığıyla kullanan kişilere aktarılması ve işletilmesi işlemleri gerçekleştirilmiştir [15].

Şahin, 2022 senesinde yazdığı yüksek lisans tezinde "Araçlar İçin Can Veri Yolu İzleme Sisteminin Tasarımı" çalışması gerçekleştirmiştir. Bu çalışmada deneme aracında transmisyon ve motor elektronik denetleme kısımlarına, veri ağı ile ulaşılarak, desteklenen bütün OBD II servisleri işletilmiştir. Bu servisler ile ilişkili olup iyileştirmeler yapılmıştır. Elde edilen sonuçlardaki bilgilerin önceden oluşabilecek hatanın iyileştirmesi için yapılacak bakım çalışması gibi benzeri çalışma alanlarında kullanılabileceğinin tespiti yapılmıştır [16].

Bu çalışmada ise otomobil CAN haberleşme sistemi üzerinde arıza tespitine yönelik bir veri toplama sistemi uygulaması yapılmıştır. Bu amaçla araç üzerindeki sıcaklık, motor devri, yakıt seviyesi gibi

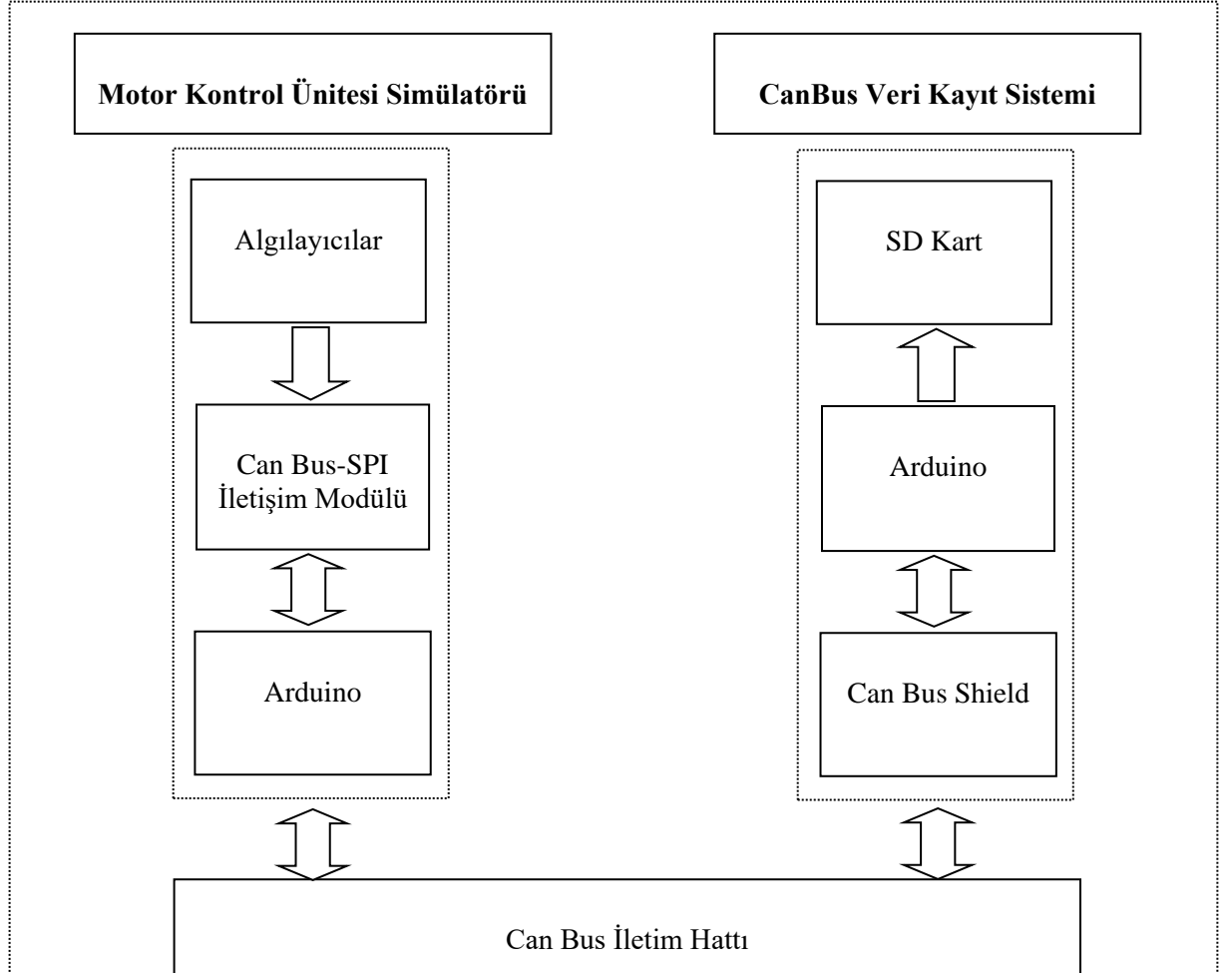
bilgilerin gerçeğe uygun üretildiği bir benzetim ortamı oluşturulmuştur. Geliştirdiğimiz cihaz bu ağa bağlanarak, belirlediğimiz verileri gerçek zamanlı olarak bir SD karta yazabilmektedir. SD karta yazılan bu veriler bilgisayar ortamına aktarılarak görselleştirilebilmektedir. Ayrıca sistemin gerçek zamanlı olarak çalışabildiğinin analizi yapılmıştır.

Bu çalışmada öncelikle Motor Kontrol Ünitesi Simülâtörü yapılmıştır. Bu simülâtör sayesinde motor sıcaklığı, motor devri, yakıt seviyesi, ortam sıcaklığı ve gaz pedalı konumu verileri OBD II standardına uygun olarak üretilmiştir. Üretilen bu verilerin de CanBus üzerinden aktarılması sağlanmıştır. Ayrıca geliştirilen CanBus Veri Kayıt Sistemi ile CanBus üzerindeki istenilen verilerin alınması ve bir SD karta yazdırılması sağlanmıştır. SD karta kaydedilen bu verilerin geliştirilen ara yüz sayesinde bilgisayar ekranında grafiğe dönüştürülmesi ve hata denetimi amacıyla incelenebilmesi sağlanmıştır. Bu çalışma ile geliştirilen sistem ile herhangi bir araca ya da veriye yönelik kayıt ve izleme yapılması sağlanmıştır.

Bu çalışmanın ikinci bölümünde metodoloji bilgisi verilir sistemin çalışması anlatılmıştır. Üçüncü bölümde CanBus hakkında genel bilgi verilir OBD II soketi üzerinden alınan parametrelerin hesaplanması işlemi anlatılmıştır. Ayrıca geliştirilen Motor Kontrol Ünitesi Simülâtörü ve Veri Kayıt Sistemi de bu bölümde anlatılmıştır. Dördüncü bölümde veri görselleştirme uygulaması anlatılmıştır. Son bölümümüzde ise çalışmamızın sonucu hakkında detaylı bilgi verilmiştir.

2 Metodoloji

Bu çalışmada geliştirilen Veri Depolama Sistemi Uygulaması, araç CanBus hattını kullanarak bu hatta oluşan algılayıcı verilerinin toplanması sağlanmıştır. Toplanan bu verilerin SD karta kaydedilmesi ve sonrasında bu verilerin bilgisayar ortamında görselleştirilmesi sağlanmıştır. Bu amaçla tasarlanan ve iki ana kısımdan meydana gelen elektronik sistemlerin blok diyagramı Şekil 1’de verilmektedir.



Şekil 1: Veri Depolama Sistemi Uygulaması Blok Diyagramı

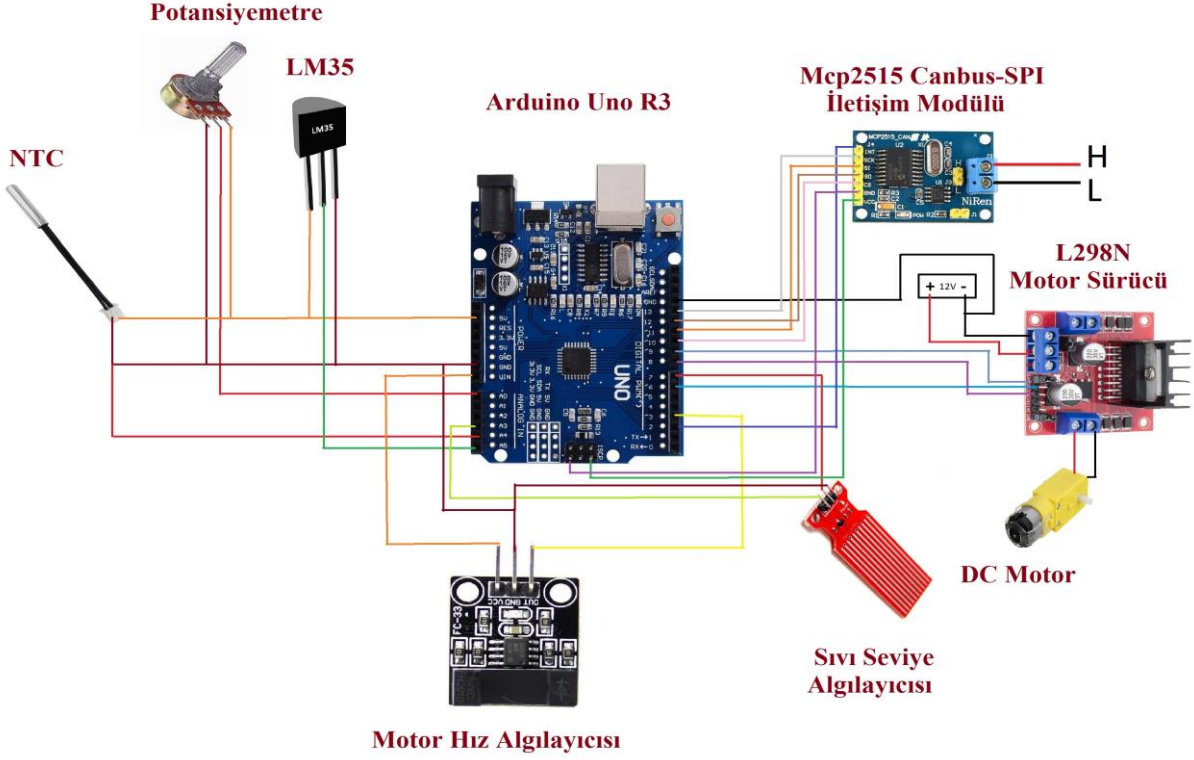
Birinci kısım, araç motor ECU'sunun çalışmasını simüle eden ve algılayıcı verilerini CanBus aracılığıyla gönderen Motor Kontrol Ünitesi simülatörüdür. Veri Depolama Sistemi çalışmasında bir aracın CanBus hattında bulunan algılayıcı verilerinin depolanması hedeflendiği için bu sistemi bir araca bağlamamız veya bu verileri üreten bir simülatör kullanmamız gerekmektedir. Çalışmayı laboratuvar ortamında gerçekleştirebilmek ve çalışma kolaylığı açısından burada bir simülatör tercih edilmiştir. Tasarlanan bu simülatör bir otomobilin motor sıcaklığı, yakıt seviyesi, gaz pedalı, ortam sıcaklığı ve motor devri verilerini bir araç üzerinde bulunan CanBus hattına uygun olarak üretmektedir. Bu kısımda algılayıcılardan alınan ham veriler Arduino ile ölçülerek CanBus aracılığıyla gönderilmektedir. Bu bölüm 3 ana kısımdan meydana gelmektedir. Algılayıcılar, Arduino ve Arduino kartının CAN hattı üzerinden iletişim kurmasını sağlayan CanBus modülünden meydana gelmektedir.

Bunun için Arduino kartı ve MCP2515 Canbus-SPI haberleşme modülü kullanılmıştır. Algılayıcı verilerinin oluşturulması için Tablo 1'deki malzemeler kullanılmaktadır:

İstenilen Parametre	Kullanılan Malzeme
Motor Sıcaklığı	LM35
Yakıt Seviyesi	Su Seviye Algılayıcısı
Gaz Pedalı	Kodlayıcı
Ortam Sıcaklığı	NTC
Motor Devri	Potansiyometre

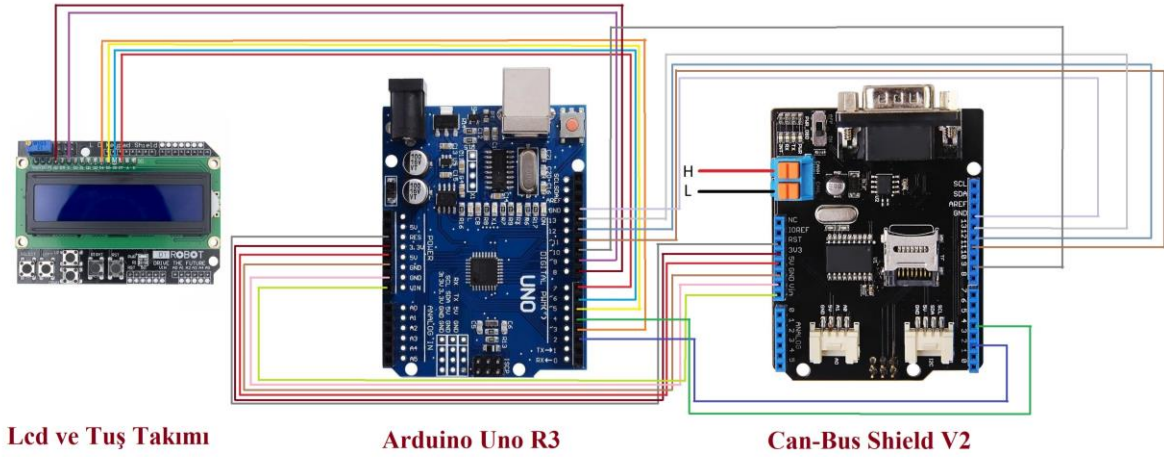
Tablo 1: Algılayıcılar İçin Kullanılan Malzemeler

Şekil 2'de Motor Kontrol Ünitesi Simülatörü için kullanılan malzemeler ve malzemeler arası bağlantıları gösterilmiştir.



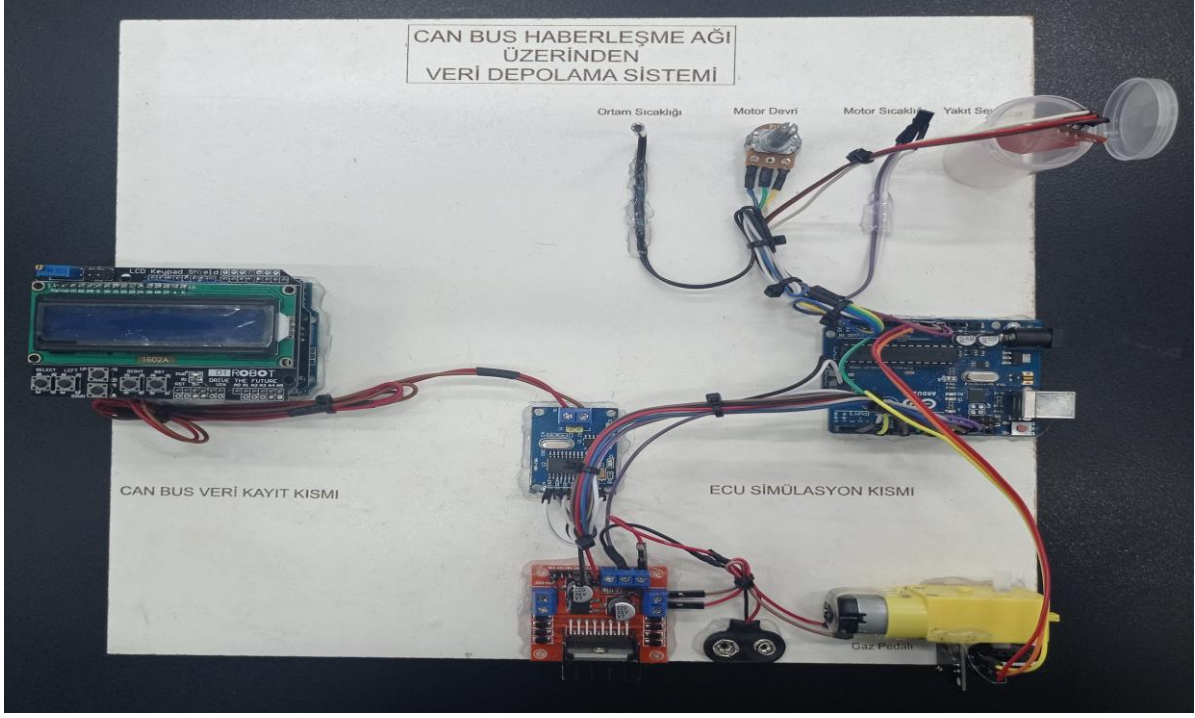
Şekil 2: Motor Kontrol Ünitesi Simülasyon Kısım

İkinci kısım ise CanBus Veri Kayıt Sistemidir. Bu kısımda motor kontrol ünitesi benzetiminden istenen bilgiler alınarak, CanBus Shield üzerindeki SD karta OBD II protokolü ile kaydedilir. Daha sonra bu veriler Arduino yardımıyla Bölüm 3'te anlatılan hesaplama yöntemiyle algılayıcı verisine dönüştürülmektedir. Verilerin alınıp SD karta kaydedildiği CanBus bölümü Şekil 3'de gösterilmiştir. Arduino Uno, Can-Bus Shield V2, LCD ve tuş takımı ile birbirine bağlanmış olup ilişkileri Şekil 3'de şemada gösterilmiştir.



Şekil 3: CanBus Veri Kayıt Sistem Kısım

Tüm projenin birleştirilmiş hali Şekil 4'de gösterilmiştir.



Şekil 4: Projeye Genel Bakış

Projede kullanılan malzemeler Arduino Uno R3, CAN-BUS Shield V2, Lcd ve Tuş Takımı, Mcp2515 Canbus-SPI İletişim Modülü, L298N Motor Sürücüsü, Enkoder Sayacı (Motor Devri Algılayıcısı), Lm35, Sıvı Seviye Algılayıcısı, Ntc, Potansiyometre, DC Motor'dur. Bunların açıklaması ve kullanım yerleri aşağıda verilmiştir.

Arduino Uno R3: Özgün ve basit kodlama dili ile programlanabilen bir geliştirme kartı olarak üretilmiştir. USB girişi ile adaptör girişi bulunmaktadır. Usb girişi ile bilgisayara bağlantı kurup kolayca kodlayabilir veya bilgisayar ile iletişim kurmasını sağlayabiliriz. Projede Arduino Uno R3 kartları CanBus hattından veri alan Can-Bus Shieldi ve simülatör kısmındaki algılayıcıları kontrol etmek için kullanılmaktadırlar.

Can-Bus Shield V2: CanBus, genellikle otomotiv endüstrisinde kullanılan bir seri haberleşme protokolüdür. Araç bilgisayarlarından araç hakkında birçok bilgi alabilir, araç destekliyorsa komutlar verebilirsiniz. Can-Bus Shield V2, Arduino ve araçlar ile iletişim sağlamak için bir köprü vazifesi görür. Arduino üzerinden aracınızın ECU'su arasındaki iletişimi sağlar. Bu sayede motor sıcaklığı, yakıt seviyesi, ortam sıcaklığı gibi birçok bilgiye Arduino kullanılarak anında ulaşılabilmektedir. Projede Can-Bus Shield V2 CAN hattından gelen veriyi SD karta kaydetmek için kullanılmaktadır.

LCD ve Tuş Takımı: LCD ve Tuş Takımı, Arduino Uno ve benzeri kartlarla kullanabileceğiniz, tüm bağlantıları yapılmış, 16x2 LCD ekranlı ve 5 programlanabilir butonlu bir Arduino Shield kartıdır. LCD ekranın kontrastını ayarlamak için yerleşik bir trimpot vardır. Basmalı butonlar A0 pinine bağlanır. Butonlara basıldığında A0 pininden okunan değer farklıdır. Her düğme için okunan değer farklıdır. Bu sayede hangi tuşa basıldığı anlaşılır. Projede LCD ve Tuş Takımı, Arduino üzerindeki komutları kontrol etmek için kullanılmaktadır.

Mcp2515 CanBus-SPI İletişim Modülü: Can-Bus Shield, Arduino kartlarının CAN hattına dâhil olmasını sağlayan bir iletişim kartıdır. Ayrıyeten, birden fazla Arduino kartının CAN hattı üzerinden kolayca iletişim kurmasını sağlar. SPI arayüzlü MCP2515 CAN denetleyicisine ve TJA1050 CAN

alıcı-vericisine dayanmaktadır. OBD II kütüphanesiyle kolay bir şekilde kullanılabilir. Projede Mcp2515 Canbus-SPI iletişim modülü, simülatör kısmındaki CAN verisini Can-Bus Shield V2 kısmına aktarmak için kullanılmaktadır.

L298N Motor Sürücüsü: 5 Volt ile 46 Volt arasındaki motorları sürmek için üretilmiş, L298N motor sürücüsü ile bütünleşmiş çalışan bir karttır. İki motorun aynı anda ayrı olarak yönetilmesini de sağlar. L298N Motor Sürücüsü projenin simülatör kısmında bulunan motoru sürebilmek için kullanılmaktadır.

Enkoder Sayacı (Motor Hız Algılayıcısı): Motor hız algılayıcısı, motor hızını ölçen bir algılayıcıdır. Motor hızı, yönü, pozisyon limiti, pulse sayısı vb. uygulamalarda rahatlıkla kullanılmaktadır. Hız, bir MCU tarafından darbe çıkışı okunarak belirlenebilir. Dijital çıkış vermenin yanı sıra opto-çift prensibi ile de çalışır. Projenin simülatör kısmında motor devrinin dönüş hızından gaz pedalı verisini elde etmek için kullanılmaktadır.

LM35: Sıcaklık algılayıcıları, çıkışında sıcaklıkla doğru orantılı bir voltaj verir. LM35i kullanırken ölçümleme devresi kurmamıza gerek yok. Ufak bir akım çeker, yaklaşık 60 μ A civarındadır. Bu sebepten dolayı, çalışma sırasında üretilen dâhili ısı azdır. Entegrenin ortam sıcaklığının yalnızca 0,1 $^{\circ}$ C üzerindedir. Çıkış empedansı düşük olduğundan, voltajı ölçerken hata payı küçüktür. Projenin simülatör kısmında motor sıcaklığı algılayıcısı olarak kullanılmaktadır.

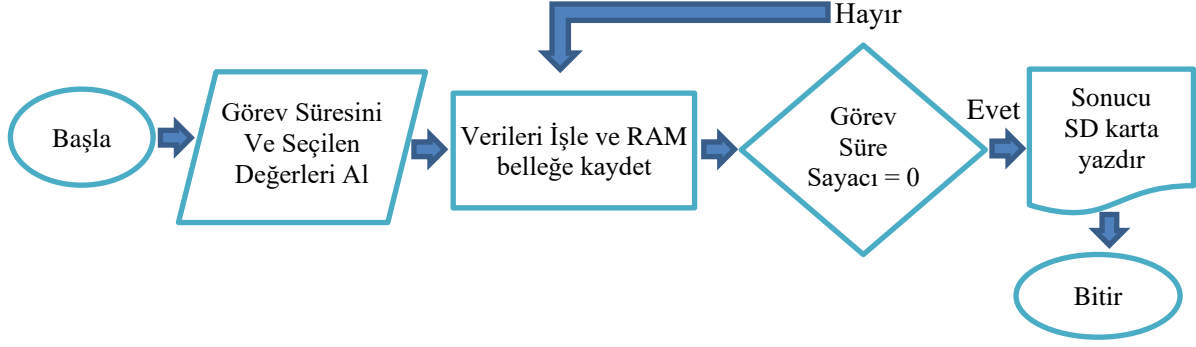
Sıvı Seviye Algılayıcısı: Sıvı seviye algılayıcısı analog çıkış veren bir algılayıcıdır. Algılayıcı çıkışını Arduino kartının analog pini üzerinden bağlantı kurup buradaki veriyi almamız yeterli olur. Aldığımız veri sıfırda hiçbir sıvı yok demektir. Eğer altı yüz otuz ile altı yüz atmış aralığında bir veri alıyorsak seviyemiz 1-2 cm aralığında demek oluyor. Ayıyeten sıvının tuzlu ya da saf olarak bulunması %10-15 aralığında değişken verilere ulaşılmasına sebep olabilir. Seviye algılayıcısı yağmur algılayıcısı olarak da kullanılabilir. Algılayıcının kullanımı çok kolaydır. Giriş voltajı ile toprak bağlantısı sağlanarak, algılayıcının çıkış bacağından okuma yapılabilir. Projenin simülatör kısmında sıvı seviye algılayıcısı yakıt seviye algılayıcısı olarak kullanılmaktadır.

NTC: 10K NTC prob ucundaki sıcaklığa göre 0-10K ohm arasında direnç gösteren ısıya duyarlı direnç çeşitlerindedir. Projenin simülatör kısmında NTC ortam sıcaklığı algılayıcısı olarak kullanılmaktadır.

Potansiyometre: Ayarlı bir direnç türüdür. Üstünde bulunan mil döndükçe değeri değişir. Genellikle farklı direnç değerleri kullanılması gereken devrelerde kullanılır. Yan çıkışlar ile orta çıkış arasında direnç değeri değişir. İki değişik sistemin birden kontrolü sağlanabilmektedir. Projenin simülatör kısmında potansiyometre motor devrinin dönüş hız verisini elde etmek için kullanılmaktadır.

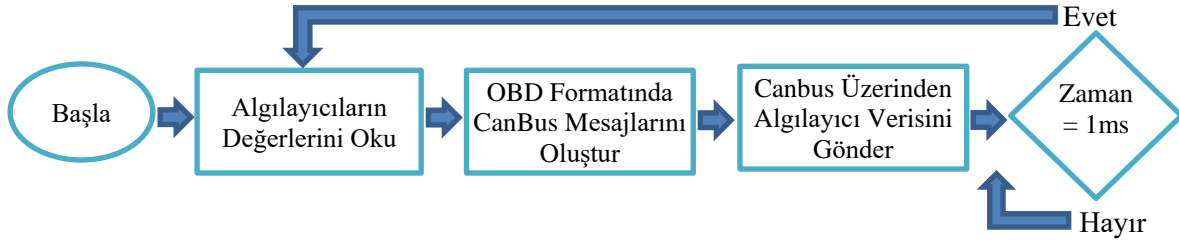
DC Motor: 12 Volt çalışma voltajına sahiptir. 250 Rpm'lik yüksek dönüş hızı ile dönebilmektedir. L redüktörlü motor tipine sahiptir. Projenin simülatör kısmında potansiyometre ile motor rpm verisini elde etmek için kullanılmaktadır.

Şekil 5'deki algoritmada gösterildiği gibi öncelikle LCD üzerinden seçilen veri toplama süresi (1 dakika, 5 dakika veya 10 dakika) ve algılayıcı verisi (motor devri, motor sıcaklığı, yakıt seviyesi, gaz pedalı, ortam sıcaklığı) CanBus Shield V2 üzerinden gelen veriler arasından seçilerek Arduino Uno R3'ün RAM belleğine kaydedilir. SD kart yazma hızının CanBus haberleşme hızından düşük olduğundan CanBus hattından gelen verilerin kayırılmaması amacıyla bu verilerin öncelikle RAM belleğe kaydedilmesi tercih edilmiştir. Daha sonra seçilen süre bittiğinde bu veriler RAM bellekten, SD karta kaydedilir.



Şekil 5: CanBus Veri Kayıt Kısımının Algoritması

Şekil 6'daki algoritmada gösterildiği gibi sistem çalıştırdıktan sonra her 1ms'de sürekli olarak algılayıcılardan değerleri okur ve bu değerleri Arduino Uno R3 ile OBD II formatına uygun olarak hesaplayarak CanBus Shiled V2e aktarır.



Şekil 6: Motor Kontrol Ünitesi Simülatörünün Algoritması

3 OBD Sistemi

Yerleşik teşhis (OBD); SAE tarafından araç iletişim yapısına bağlantı sağlamak için geliştirilmiş bir standarttır. Protokoller, OBD araçta kullanılan ECU'ya ve algılayıcı verilerine erişmek için değişik türdeki araç türlerinde kullanılmak için dizayn edilmiştir. OBD, aracın arızalarını teşhis etmesini ve rapor etmesini sağlayan bir otomotiv terimi olarak tanımlanır [17].

OBD standardının geliştirmiş hali OBD II'dir. OBD II'nin üç ana kuralı vardır. Motor kontrol ünitesi ile iletişim standartları, parametre kimlikleri ve teşhis hata kodudur [18].

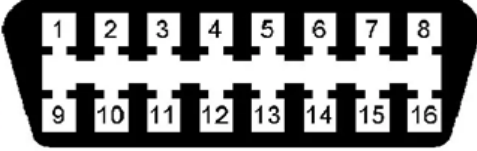
OBD II, araçlarla iletişim kurmak için 5 farklı iletişim protokolünü destekler [19]. Bunlar ISO 15765-4-CAN, ISO 14230, J1850 VPW, ISO 9141-2 ve J1850 PWM'dir.

OBD II soketi genellikle sürücü koltuğu ve yolcu bölmesi çevresinde bulunmaktadır. Şekil 7'de Renault Clio aracın OBD soket yeri gösterilmiştir.



Şekil 7: Renault Clio OBD Soket Yeri

Bazı markalarda farklılık gösterse de OBD II soketi ve bağlantıları genellikle Tablo 2'deki gösterildiği gibidir.

			
PIN	TANIM	PIN	TANIM
1	Seçenek	9	Seçenek
2	J1850 Bus +	10	J1850 Bus -
3	Seçenek	11	Seçenek
4	Şase	12	Seçenek
5	Şase	13	Seçenek
6	Can High	14	Can Low
7	K Hattı	15	L Hattı
8	Seçenek	16	Güç Girişi

Tablo 2: OBD II Soketi Ve Bağlantı Şeması

OBD II üzerinden veri alırken CAN hattı üzerinden Parametre ID(PID)'ler sorgulanır. Alınan ID kodları şu şekilde olmaktadır:

ID: 7e8 Data Len: 8 Byte0 Byte1 Byte2 Byte3 Byte4 B Byte5 B Byte6 B Byte7

Bu ID kodunu otomatik hesaplama işlemi internetteki birçok web sitesi üzerinden yapılabilmektedir. Şekil 8'deki hesaplama tablosu da internet üzerindeki bir web sayfası üzerinden alınmıştır [20].

OBD2 PID Overview [Lookup/Converter Tool, Table, CSV, DBC]

PID	Name	Bit start	Bit length	Scale	Offset	Min Max	Unit			
0C	Engine speed	31	16	0.25	0	0 16384	rpm			
		CAN ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Request		7DF	02	01	0C	AA	AA	AA	AA	AA
Response (example)		7E8	04	41	0C	2	8c	AA	AA	AA
Physical value (DEC)	=	0	+	0.25	*	652	=	163		rpm

Şekil 8: OBD II PID'ye Genel Bakış [Arama/Dönüştürücü Aracı, Tablo, CSV, DBC]

Örnek olarak OBD II soketi üzerinden alınan ID: 7e8 Data Len: 8 BYTE 04 41 0C 2 8C AA AA AA parametreleri web sayfası üzerinden Motor Sıcaklığı (0C) seçilerek Byte3 ve Byte4'ün değerleri girilir. 2 ve 8C girildiğinde sistem otomatik olarak motorun rpm değerini 163 rpm olarak vermektedir.

28C 16 sisteminde karşılık gelen sayıdır. Bu değer ondalık sisteme çevrildiğinde 652 olur. Dönüşüm denklem (1)'de gösterilmiştir.

$$(28C)_{16} = (652)_{10} \quad (1)$$

OBD II çevrim formülü denklem (2)'de uygulanmıştır.

$$652 \times 0.25 = 163 \text{ rpm} \quad (2)$$

Gelen algılayıcı verilerinde, Byte2 nin aldığı değerlere göre Gaz pedalı verisi $(11)_{16}$, yakıt seviyesi $(2F)_{16}$, ortam sıcaklığı $(46)_{16}$, motor sıcaklığı $(5C)_{16}$ parametresi ile temsil edilmektedir. Alınan 5 adet algılayıcı verisi için farklı formül işlemleri uygulanmaktadır. Alınan verilerdeki PID'lerin eşdeğerleri ve formül hesaplamaları denklem (3), (4), (5), (6) ve (7)'de verilmiştir.

$$\text{Motor Devri} = ((\text{Byte3} * 256) + \text{Byte4}) \times 0.25 \quad (3)$$

$$\text{Gaz Pedalı} = \text{Byte3} \times 100/255 \quad (4)$$

$$\text{Yakıt Seviyesi} = \text{Byte3} \times 100/255 \quad (5)$$

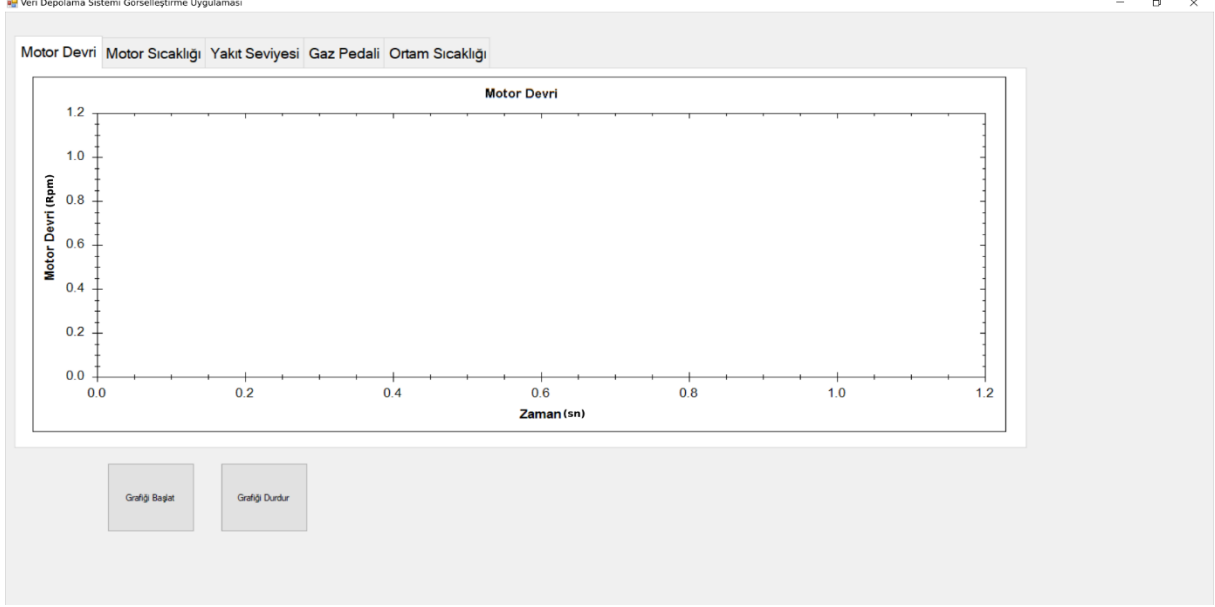
$$\text{Ortam Sıcaklığı} = \text{Byte3} - 40 \quad (6)$$

$$\text{Motor Sıcaklığı} = \text{Byte3} - 40 \quad (7)$$

4 Veri Görselleştirilme Uygulaması

Veri görselleştirme uygulaması Visual Studio 2019 üzerinden C# programlama dili ile kodlanmıştır. Masaüstü uygulamasının görevi, CanBus hattından alınan veriyi görselleştirmektir. Geliştirilen

CanBus veri kayıt sistemi tarafından alınan veriler SD karta kaydedilmektedir. Daha sonra bu veriler SD karttan bilgisayara kopyalanmaktadır. Veri görselleştirme uygulaması da bilgisayara kaydedilen veri dosyasını açarak istenilen algılayıcı verilerinin görselleştirilmesini sağlamaktadır. Devre üzerinden seçilen süreye göre sonuçlar zaman grafiğine yansımaktadır. Görselleştirilmek istenen algılayıcı verileri arayüzün üst kısmında bulunan sekmeler aracılığıyla seçilir. Şekil 9’da geliştirilen arayüz görülmektedir.



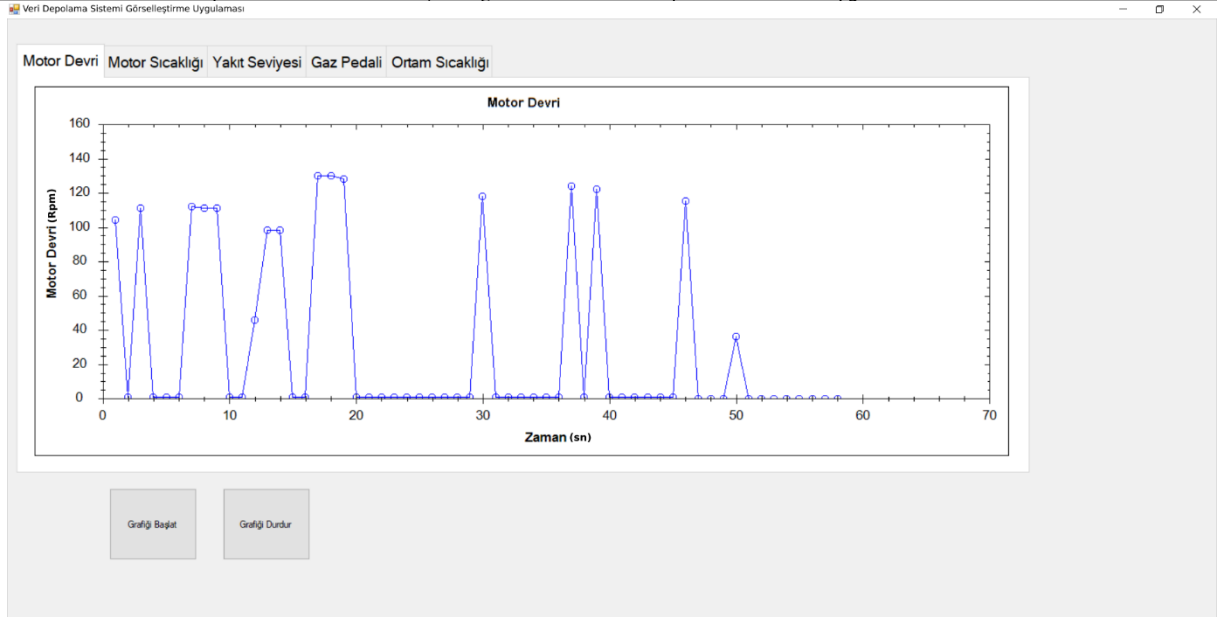
Şekil 9: Geliştirilen Arayüz Çalışması

Sistemin test aşamasında ECU benzetim sistemine bağlı algılayıcılar da rastgele değişimler oluşturulmuştur. Bu değişimler her 1 msn’de sürekli olarak ECU benzetim sistemi tarafından PID formatına çevrilip CanBus üzerinden yayınlanmaktadır. Bu veriler CanBus üzerinden alınarak veri kayıt sistemi tarafından SD karta kaydedilmektedir. SD karta kaydedilen veriler PID formatında olup, sonrasında bu veriler görselleştirme uygulaması tarafından bölüm üçte verilen formüller kullanılarak tekrar algılayıcı verisine dönüştürülmektedir. Elde edilen parametrelerin değişim aralıkları ve birimleri Tablo 3’de gösterilmiştir.

Parametre	Değer Aralığı	Birimi
Motor Devri	0 - 140	Rpm
Motor Sıcaklığı	15 - 32	⁰ C
Yakıt Seviyesi	0 - 90	%
Gaz Pedalı	10 - 65	%
Ortam Sıcaklığı	21 - 30	⁰ C

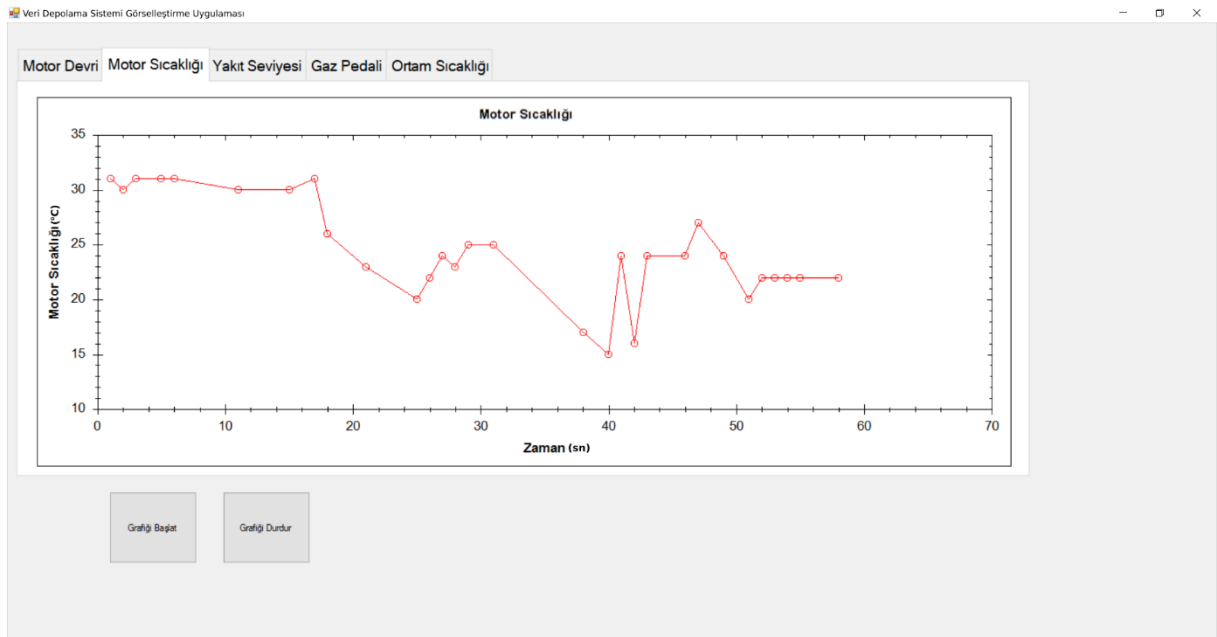
Tablo 3: Parametre Değerleri

Şekil 10’da veri görselleştirme uygulaması ile elde edilen motor devri algılayıcı verileri görülmektedir. Motor devri için 1 dakika içerisinde 0 rpm ile 140 rpm arasında değişen değerlerde veri alınmıştır. Aralarda özellikle devir kesilmiş ve sonrasında çalıştırılıp devir değerlerinin hızlanma süreleri kayıt altına alınmıştır.



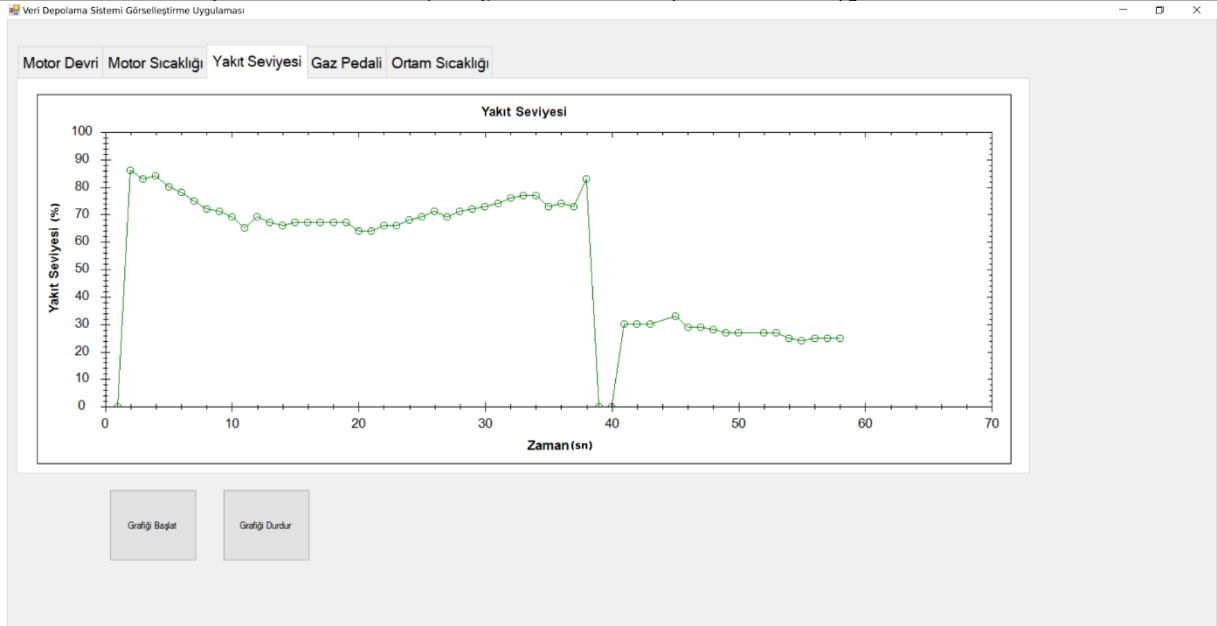
Şekil 10: Motor Devri Görselleştirme Uygulaması

Şekil 11’de veri görselleştirme uygulaması ile elde edilen motor sıcaklığı algılayıcı verileri görülmektedir. Motor sıcaklığı için 1 dakika içerisinde 15°C ile 32°C arasında değişen değerlerde veri alınmıştır. Özellikle 52. saniyeden sonra stabil olarak tutulmuş, verinin doğruluğu kanıtlanmıştır.



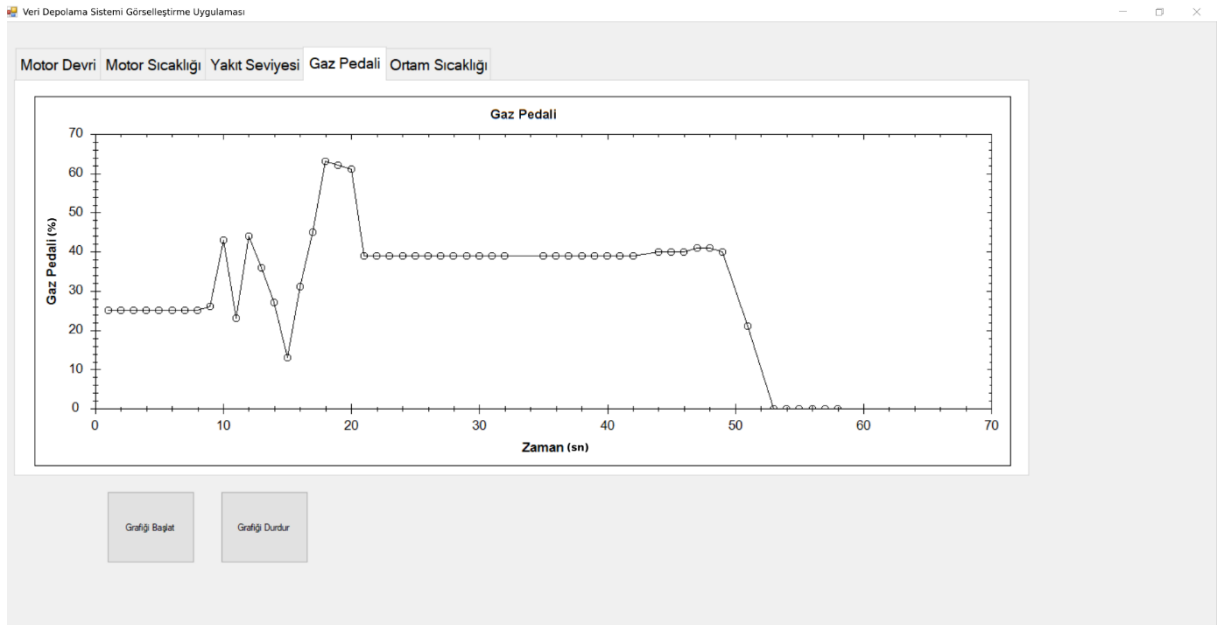
Şekil 11: Motor Sıcaklığı Görselleştirme Uygulaması

Şekil 12’de veri görselleştirme uygulaması ile elde edilen yakıt seviyesi algılayıcı verileri görülmektedir. Yakıt seviyesi için 1 dakika içerisinde %0 ile %90 arasında değişen değerlerde veri alınmıştır. Burada algılayıcıdaki tam dolu yakıt seviyemiz %100 olarak ayarlıdır. İlk olarak depo boş iken birden dolu seviyeye getirilmiştir. 39. saniyede tekrar boş konuma getirilip 40. saniye tekrar %30 civarında doldurulmuştur.



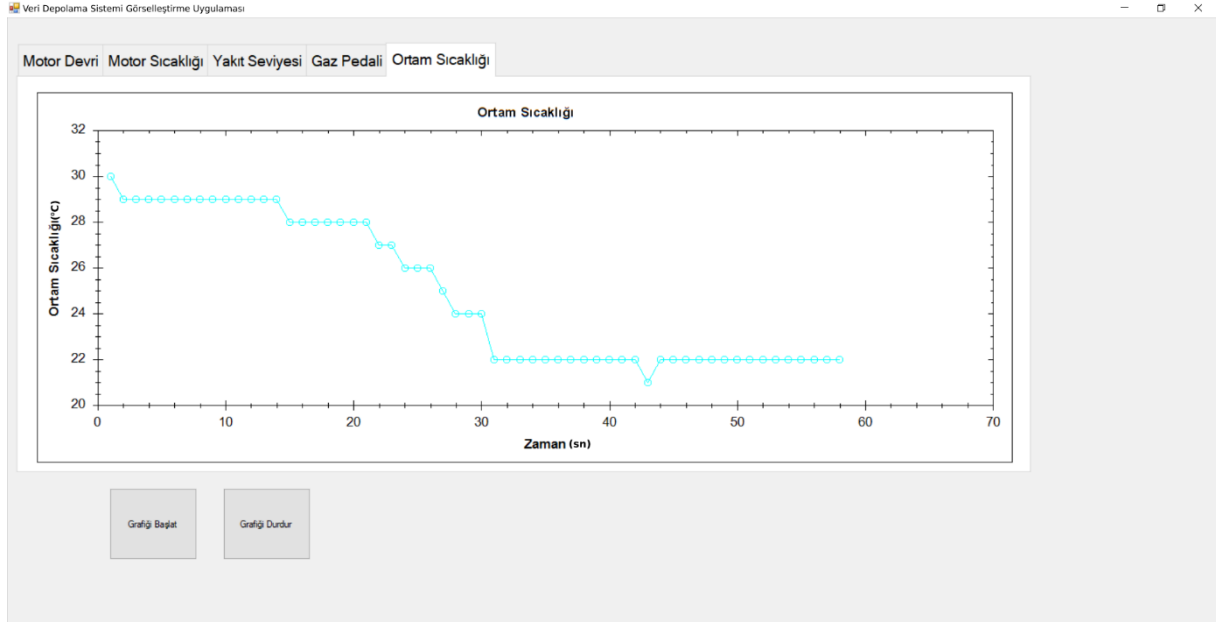
Şekil 12: Yakıt Seviyesi Görselleştirme Uygulaması

Şekil 13’de veri görselleştirme uygulaması ile elde edilen gaz pedalı algılayıcı verileri görülmektedir. Gaz pedalı için 1 dakika içerisinde 10 ile 65 devir arasında değişen değerlerde veri alınmıştır. Gaz pedalı en son 7 saniye kala kesilmiştir.



Şekil 13: Gaz Pedalı Görselleştirme Uygulaması

Şekil 14’de veri görselleştirme uygulaması ile elde edilen ortam sıcaklığı algılayıcı verileri görülmektedir. Ortam sıcaklığı için 1 dakika içerisinde 21⁰C ile 30⁰C arasında değişen değerlerde veri alınmıştır. Sıcaklık gittikçe düşmüştür.



Şekil 14: Ortam Sıcaklığı Görselleştirme Uygulaması

Edinilen bu veriler ile Motor Kontrol Ünitesi Simülör sisteminde üretilen veriler karşılaştırılarak, algılayıcı verilerinin arayüz programında doğru olarak elde edildiği görülmüştür.

5 Sonuçlar

Bu çalışmada CanBus iletişim hattı üzerinden araç verilerini toplayan ve bunları bilgisayar ortamında görselleştiren veri depolama sistemi gerçekleştirilmiştir. Geliştirilen Motor Kontrol Ünitesi Simülörü sayesinde algılayıcı verileri OBD II protokolüne uygun olarak üretilmiştir. Bu nedenle CanBus Veri Kayıt Sistemi bir aracın OBD II soketine bağlandığında da doğru olarak çalışacaktır. Motor Kontrol Ünitesi Simülörünün ürettiği algılayıcı verileri CanBus Veri Kayıt Sistemi ile düzgün bir şekilde SD karta yazılmış ve geliştirilen arayüz sayesinde Bölüm 4'te verilen grafikler çizdirilerek görselleştirilmiştir.

Diğer çalışmalardan farklı olarak hazır bir simülör kullanılmamış ve istenilen algılayıcı verilerinin üretilmesi için Motor Kontrol Ünitesi Simülörü yapılabileceği gösterilmiştir. Bu simülör oluşturulurken CanBus hattında algılayıcı verilerinin üretilmesi için Bölüm 3'te verilen OBD II PID tablosundan faydalanılmıştır. Diğer çalışmalarda pek bahsedilmeyen OBD II PID hesaplamaları ile algılayıcı verilerinin OBD II protokolüne uygun olarak üretilebileceği gösterilmiştir.

Bu çalışma toplanılan algılayıcı verilerinin SD karta kaydedilmesi açısından da diğer çalışmalara göre farklılık sağlamıştır. Bu çalışmada araç arıza kodlarının okunmasına odaklanmak yerine anlık algılayıcı verilerinin kaydedilip görselleştirilmesi amaçlanmıştır. Anlık verilerin görselleştirilerek grafik üzerinden takip edilmesi Ar-Ge ya da eğitim amaçlı çalışmalarda kullanılmasına olanak sağlayacaktır. SD kart verileri MATLAB gibi programlara aktararak istenirse sistem analiz çalışmaları da yapılabilir. Aynı zamanda birbirini etkileyen karmaşık arızaların tespitinde de fayda sağlayacaktır.

Gelecek çalışmalarda geliştirilen Motor Kontrol Ünitesi Simülöründe kullanılan algılayıcı sayısı artırılabilir. Senaryo bazlı olarak simülör yazılımı geliştirilerek, bazı algılayıcı verilerinin aracın çalışmasına uygun olarak otomatik değiştirilmesi sağlanabilir. Oluşturulacak senaryolara göre de hata kodları üreten bir simülör gerçekleştirilebilir. Veri kayıt sisteminde ise kablosuz haberleşme ile veriler kaydedilirken gerçek zamanlı olarak bilgisayar ortamına aktarılıp grafiğinin çizdirilmesi sağlanabilir. Veri kayıt sistemi ile en fazla 10 dakikalık kayıt yapılabilmektedir. Bu süre arttırılarak veriler üzerinde daha uzun incelemeler yapılabilir.

6 Beyanname

6.1 Çalışma Sınırları

Çalışmada herhangi bir sınır bulunmamaktadır.

6.2 Teşekkür

Yazarlar, bu çalışmanın kalitesini artıran yapıcı tavsiyeler veren anonim hakemlere teşekkürlerini sunar.

6.3 Finansman Kaynağı

Çalışmada herhangi bir finans kaynağı kullanılmamıştır.

6.4 Rakip Çıkarlar

Çalışma rakip çıkarlar içermemektedir.

6.5 Yazar Katkıları

Sercan Ozan NUR: Makale için bilimsel yayın araştırması, deneylerin çalışmalarının yapılması, verilerin düzenlenmesi ve raporlanması, makalenin oluşturulması, bilgisayar ortamına aktarılması ve sonuçların açıklanması aşamasında katkı sağlama.

Ahmet KARACA: Makale için hipotezin oluşturulması, neticeye ulaşmak için planlama, neticelerin açıklanması için yükümlülük alma, makalenin oluşturulması aşamasında katkı sağlama.

Kaynakça

- [1] N.Navet, Y.Song, F.Simonot-Lion, and C.Wilwert, "Otomotiv İletişim Sistemlerinde Trendler", IEEE Bildirileri, Cilt 93, No.6, 1204-1224, 2005.
- [2] Uluslararası Standardizasyon Örgütü, Karayolu Araçları, Local Interconnect Network (LIN), Bölüm 1: Genel Bilgiler Ve Kullanım Durumu Tanımı, ISO/DIS 17987-1, 2006.
- [3] Uluslararası Standardizasyon Örgütü, Karayolu Araçları, Controller Area Network (CAN), Bölüm 1: Veri Bağlantı Katmanı Ve Fiziksel Sinyalleşme, ISO11898-1, 2015.
- [4] Uluslararası Standardizasyon Örgütü, Karayolu Araçları, Communication on FlexRay, Bölüm 1: Genel Bilgiler Ve Kullanım Durumu Tanımı, ISO 10681-1, 2010.
- [5] W.Voss, "J1939 İçin Kapsamlı Bir Kılavuz", Copperhill Technologies Corporation, 2008.
- [6] S.Corrigan, "Controller Area Network'e Giriş", Texas Instruments Uygulama Raporu, Temmuz 2008.
- [7] O.González, M. Rodríguez, A. Ayala, J. Hernández and S. Rodríguez, "PIC'lerin Ve Mikrodenetleyicilerin Endüstride Parametrelerin Ölçümü Ve Kontrolünde Uygulanması", Proc of the International Journal of Electrical Engineering Education 41/3, pp.265-274, Şubat 2001.
- [8] Microchip Technology Inc. DS41159E: PIC18FXX8 Veri Sayfası Pat Richards, "CAN Fiziksel Katman Tartışması", Microchip Technology Inc, Temmuz 2003.
- [9] I.Unal, "Can Üzerinden Pic Programlama", Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 2006.
- [10] I.Kara, "Can Haberleşme Protokolünün İncelenmesi Ve Bir Sıcaklık Kontrol Sistemine Uygulanması", Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 2009.

- [11] E.Dinçer, “CanBus İle Dağıtık Kontrol Uygulaması”, Gebze Yüksek Teknoloji Enstitüsü Mühendislik Ve Fen Bilimleri Enstitüsü, Kocaeli, 2010
- [12] M.Öztürk, “Hidrojen Hibrit Otobüslerde CanBus Sistemleri Ve Uygulaması”, Gebze Yüksek Teknoloji Enstitüsü Mühendislik Ve Fen Bilimleri Enstitüsü, Kocaeli, 2010.
- [13] C.Derse, “Yeni Bir CanBus İletişim Tabanlı Otomotiv Güç Dağıtım Modülü Uygulaması Ve Sistemin Güvenirlilik Analizi: eaPDM”, Celal Bayar Üniversitesi Fen Bilimleri Enstitüsü, Manisa, 2017.
- [14] G.Karanfil, “OBDII Protokollerini Kullanan Bir Arayüz Yazılımı Tasarım”, Uludağ Üniversitesi Fen Bilimleri Enstitüsü, Manisa, 2020.
- [15] M.Topuz, “Hibrit Araçlar İçin CanBus Haberleşme Birimi İle Araç Kontrol Sistemi Ve Linux Tabanlı GUI Tasarımı”, Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü, Konya, 2021.
- [16] M.A.Şahin, “Araçlar İçin Can Veri Yolu İzleme Sisteminin Tasarımı”, Uludağ Üniversitesi Fen Bilimleri Enstitüsü, Bursa, 2022.
- [17] I.Dogan, “Mikrodenetleyici tabanlı sıcaklık izleme ve kontrolü”, ISBN: 0750655569, Elsevier Science & Technology Books, 2002.
- [18] A.Sim and B.Sitohang, “OBD-II Standart Araba Motor Teşhis Yazılımı Geliştirme”, Uluslararası Veri ve Yazılım Mühendisliği Konferansı, Kasım 2014.
- [19] K.H.Johansson, M.Törngren, and L.Nielsen, “Kontrolör Alan Ağı Araç Uygulaması”, Ağ Bağlantılı ve Gömülü Kontrol Sistemleri Kontrol Mühendisliği El Kitabı, VI, pp.741-76, 2005.
- [20] CSS Electronics, OBD2 PID Table, Ocak 2023. <https://www.csselectronics.com/pages/obd2-pid-table-on-board-diagnostics-j1979>



© 2020 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).