

Elektronik Haberleşme Cihazları - Veri Hizmet Araçları ve Kayıt Birimi

Görkem OCAK

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Ocak 2022



Electronical Communication Equipments - Data Utilities and Record Unit

Görkem OCAK

MASTER OF SCIENCE THESIS

Department of Computer Engineering

January 2022

Elektronik Haberleşme Cihazları - Veri Hizmet Araçları ve Kayıt Birimi

Görkem OCAK

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Bilimleri Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Prof. Dr. İdiris DAĞ

Ocak 2022

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Prof. Dr. İdiris DAĞ danışmanlığında hazırlamış olduğum “**Elektronik Haberleşme Cihazları - Veri Hizmet Araçları ve Kayıt Birimi**” başlıklı tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 04/01/2022

Görkem OCAK

ÖZET

Endüstriyel alanlarda sıkça kullanılan elektronik cihazlar kendi içlerinde ve dış dünya ile veri iletimini sağlamak amacıyla elektronik haberleşme yollarını kullanmaktadır.

Elektronik haberleşme özellikle askeri araç, sağlık cihazları ve akıllı ev sistemlerine ait cihazlarda sıklıkla kullanılmak ile beraber elektronik haberleşmeyi sağlayan ve kolaylaştıran her türlü cihaz elektronik haberleşme cihazı olarak adlandırılır.

Günümüzde elektronik cihazlara olan ilginin artması ve elektronik cihazların her türlü sektörde insan hayatını kolaylaştırmak adına yer almasından dolayı elektronik cihaz teknolojisi her geçen gün ilerlemektedir. Elektronik cihaz teknolojisinin ilerlemesi ile elektronik cihazların gerçekleştirmesi ve sunması gereken imkânların artacağına dair beklentiler oluşmaktadır. Bu beklentileri karşılamak için elektronik cihazların olmazsa olmazı olan elektriksel verinin değeri artmak ile beraber bu verinin iletimini sağlamak için kurulan elektronik haberleşme alanına olan ilgi ve çalışmalarda artmaktadır.

Bilindiği üzere günümüz teknolojisinin temellerinde verinin anlamlandırılması yer almaktadır. Verinin anlamlandırılabilir bir platforma gelmesi için verinin doğru, eksiksiz ve güvenilir şekilde iletimi çok önemlidir. “Elektronik Haberleşme Cihazları - Veri Hizmet Araçları ve Kayıt Birimi” adlı Yüksek Lisans tezinde elektronik haberleşmeyi taklit edebilen, haberleşme verilerine dair kayıt alabilen, haberleşmeye veri dâhil edebilen elektronik cihazlar, veri hizmet araçları ve kayıt birimlerine dair akademik çalışmaların incelenmiştir. Ayrıca konuya ilişkin “EceDuruApp” adını verdiğim bir masaüstü veri hizmet aracı ve veri kayıt uygulaması ile “EceDuruDevice” adını verdiğim bir elektronik haberleşme cihazı tasarlanmıştır.

Tasarlamış olduğum elektronik haberleşme cihazı; veri iletiminde köprü görevi görmenin dışında haberleşme verilerinin dış dünyaya anlık aktarımı ve elektronik haberleşme teknolojilerine yönelik dönüştürücü rolü de üstlenebilmesi amacıyla hazırlanmıştır. Ayrıca tasarlamış olduğum masaüstü uygulaması ise veri haberleşmesine köprü, haberleşme teknolojileri arasında dönüştürücü özelliklerini barındırmanın yanı sıra veri kayıt etme ve haberleşmeye veri dâhil edebilme özelliklerine sahiptir.

Anahtar Kelimeler: Elektronik Haberleşme, Elektronik Cihaz, Veri Hizmet Aracı, Veri Kayıt

SUMMARY

Electronic devices, which are frequently used in industrial areas, use electronic communication means to provide data transmission within themselves and with the outside world.

Although electronic communication is frequently used in devices belonging to military vehicles, health devices and smart home systems, all kinds of devices that provide and facilitate electronic communication are called electronic communication devices.

Today, electronic device technology is advancing day by day due to the increasing interest in electronic devices and the fact that electronic devices are included in all sectors to facilitate human life. With the advancement of electronic device technology, there are expectations that the possibilities that electronic devices need to realize and offer will increase. In order to meet these expectations, as the value of electrical data, which is indispensable for electronic devices, increases, the interest and studies in the field of electronic communication established to ensure the transmission of this data is increasing.

As it is known, interpretation of data is the basis of today's technology. Correct, complete and reliable transmission of data is very important for the data to come to a meaningful platform. In my master's thesis titled "Electronical Communication Equipments - Data Utilities and Record Unit", academic studies on electronic devices, data utilities and record units that can imitate electronic communication, record communication data and include data in communication have been examined. In addition, a desktop data utilities and data recording application I named "EceDuruApp" and an electronic communication device called "EceDuruDevice" were designed.

Electronic communication device I designed; In addition to serving as a bridge in data transmission, it has been prepared in order to act as a converter for instant transmission of communication data to the outside world and electronic communication technologies. In addition, the desktop application that I have designed has the features of data communication as a bridge, transformer between communication technologies, as well as recording data and including data in communication.

Keywords: Electronic Communication, Electronic Device, Data Utilities, Data Record

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY	vi
İÇİNDEKİLER	vii
ŞEKİLLER DİZİNİ	viii
ÇİZELGELER DİZİNİ	ix
SİMGELER VE KISALTMALAR DİZİNİ	x
1. GİRİŞ VE AMAÇ	1
2. LİTERATÜR ARAŞTIRMASI	5
3. MATERYAL VE YÖNTEM	14
3.1. EceDuruDevice	24
3.2. EceDuruApp	34
4. BULGULAR VE TARTIŞMA	43
4.1. EceDuruDevice Yazılım Doğrulama Test Prosedürü	45
4.2. EceDuruApp Yazılım Doğrulama Test Prosedürü	56
5. SONUÇ VE ÖNERİLER	77
KAYNAKLAR DİZİNİ	81

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
3.1 DB9 Pinout	17
3.2 DB25 Pinout	18
3.3 ESP32 Geliştirme Kartı	23
3.4 EceDuruDevice Kutu İçi Görünümü	24
3.5 EceDuruDevice Akış Diagram Görseli	25



ÇİZELGELER DİZİNİ

<u>Çizelge</u>	<u>Sayfa</u>
3.1 DB9 ve DB25 Pin Eşleştirmesi	18
3.2 RS-232 ile RS-485 Karşılaştırması	20
3.3 ESP32 Geliştirme Kartı Teknik Özellikler	23



SİMGELER VE KISALTMALAR DİZİNİ

Simgeler

DC

Açıklamalar

Direct Current

Kisaltmalar

Ar-Ge

ARINC

ARM

CAN

CPU

DCE

DTE

EIA

FPGA

GPRS

GSM

IoT

IP

MIL-STD

MSSQL

MySQL

PIC

REST

RF

TCP

TTL

UART

UDP

USART

Wi-Fi

Açıklamalar

Araştırma ve Geliştirme

Aeronautical Radio Incorporated

Acorn RISC Machine

Controller Area Network

Central Process Unit

Data Communication Equipment

Data Terminal Equipment

Electronic Industries Association

Field Programmable Gate Array

General Packet Radio Service

Global System for Mobile Communications

Internet of Things

Internet Protocol

Military Standard

Microsoft Structured Query Language Server

My Structured Query Language

Peripheral Interface Controller

Representational State Transfer

Radio Frequency

Transmission Control Protocol

Transistor-Transistor Logic

Universal Asynchronous Receiver Transmitter

User Datagram Protocol

Universal Synchronous Asynchronous Receiver Transmitter

Wireless Fidelity

1. GİRİŞ VE AMAÇ

Endüstriyel alanlarda sıkça kullanılan elektronik cihazlar kendi içlerinde ve dış dünya ile veri iletimini sağlamak amacıyla elektronik haberleşme yollarını kullanmaktadır. Günümüzde elektronik cihazlara olan ilginin artması ve elektronik cihazların her türlü sektörde insan hayatını kolaylaştırmak adına yer almasından dolayı elektronik cihaz teknolojisi her geçen gün ilerlemektedir. Elektronik cihaz teknolojisinin ilerlemesi ile elektronik cihazların gerçekleştirilmesi ve sunması gereken imkânların artacağına dair beklentiler oluşmaktadır. Bu beklentileri karşılamak için elektronik cihazların olmazsa olmazı olan elektriksel verinin değeri artmak ile beraber bu verinin iletimini sağlamak için kurulan elektronik haberleşme alanı olan ilgi ve çalışmalarda artmaktadır. Bilindiği üzere günümüz teknolojisinin temellerinde verinin anlamlandırılması yer almaktadır. Verinin anlamlandırılabilir bir platforma gelmesi için verinin doğru, eksiksiz ve güvenilir şekilde iletimi çok önemlidir.

Elektronik cihazlar en basitinden en karmaşığına temel olarak elektrik enerjisi ile bir işi, bir süreci ya da bir hizmeti gerçekleştirmeyi sağlar. Elektronik haberleşme ise elektronik cihazların gerçekleştirmek istediğı amaç için kullanılan bir araçtır. Bu sebepten istenilen amaç farklılık gösterdikçe kullanılan araçta farklılık göstermektedir.

Elektronik cihazlar kullanıldığı sektör ve alana göre bazı standartları sağlama ihtiyacı vardır. Bu konuya örnek vermek gerekirse askeri alanda kullanılan elektronik cihazların belirli askeri standartları sağlaması gerekmektedir. Bu standartlar cihaza ait donanım, mekanik ve elektriksel standartlar olduğu gibi yazılım standartları olarak da karşımıza gelmektedir. Genellikle yazılım alanında yer alan standartlar içeriğinde elektronik haberleşmeye dair özellikler ve kısıtlar yer almaktadır. Bu standartların sağlandığına dair testler ve bu testlerde kullanılan elektronik haberleşme sağlayan test cihazları, test uygulamaları, veri hizmet araç ve uygulamaları ciddi bir iş yükü oluşturmaktadır.

Elektronik cihaza ait testler sadece standartları sağlanıp sağlanmadığını göstermek için değil aynı zaman da cihazın çalışabilir, faal ve kullanılabilirliğini göstermeyi de hedeflemektedir. Bu nedenle bazı elektronik cihazlar bulunduğu sistem itibari ile başka bir elektronik cihaza elektronik haberleşme yöntemi bağlıdır ve bağlı olan bu cihazında testinin tamamlanmış ve doğrulanmış olması gerekmektedir. Bu gibi durumlar için test ortamında bağlı olan cihaz yerine bağlı olan cihazı taklit eden doğrulanmış yazılım uygulamaları ya da

cihazları kullanılabilir. Bu yazılım uygulamaları ya da cihazları da bağı olan cihaz gibi çalışmayı sağlamak için bağı olan cihazın sağladığı elektronik haberleşmeyi de taklit etmelidir. Elektronik haberleşmeyi taklit edebilen bu yazılımlar, uygulamalar ya da cihazlar karışımıza veri hizmet araç ve uygulamaları olarak gelmektedir.

Veri hizmet aracı ve uygulamaları sadece test ortamlarında kullanılmamaktadır. Aynı zaman da bu cihaz ve uygulamalar elektronik haberleşme hattında yer aldıkları için kayıt alma kabiliyeti barındırıyor ise veri kayıt birimi olarak da kullanılmaktadır. Elektronik cihazların bağı olduğu veri iletim hattına bağlanabilen bu cihazlar sayesinde gerek test, gerek analiz, gerekse müdahale yapmak mümkün olabilir. Örnek vermek gerekirse bir sisteme dâhil olan iki elektronik cihaz arasında yer alan elektronik haberleşme hattına eklenen veri kayıt birimi iki cihazında verilerini ayrı ayrı ya da birlikte kayıt altında tutabilir. Bu sisteme dair arıza durumu oluştuğunda veri kayıt birimi tarafından kayıt altına alınan bu veriler incelenerek sistemdeki arızaya neden olan durum tespit edilebilir. Verilen örnekte sisteme dâhil olan cihaz sayısı arttıkça veri kayıt birimi cihazlarının önemi de artmaktadır. Bu olay örgüsünün günümüz teknolojisinde örneklerinde biri olarak uçak, tren gibi ulaşım araçlarında yer alan kara kutu cihazları örnek verilebilir.

Veri kayıt birimlerinin kullanım alanları sadece kara kutu cihazları ile sınırlamak mümkün değildir. Veri kayıt birimlerine eklenebilen özellik ve beceriler ile bazı sistemlerde önleyici ve karar verici birim rollerini de üstlenebilir. Örnek vermek gerekirse sisteme dâhil olan üç cihaz ve veri kayıt biriminin var olduğu bir senaryo da bu üç cihaz da fiziksel olarak birbirine elektronik haberleşme sağlayacak yapı da bağı değil ve üç cihaz da fiziksel olarak elektronik haberleşme sağlamak için veri kayıt birimine bağı olsun. Kısaca bu üç cihazın birbiri ile elektronik haberleşmesini sağlamak amacıyla veri kayıt birimi köprü görevi gördüğü düşünülmektedir. Veri kayıt birimi hangi cihazdan gelen veriyi hangi cihaza iletmesi gerektiğini karar vererek cihaz iletişimi sağlayabilir aynı zamanda verilen doğru karar ve yönetimi sayesinde cihazları kendi ile ilgi olmayan verilerin kendisine gelmesi ile oluşabilecek sorunları önlemiş olur.

Veri kayıt birimi veri iletim hattında olmayan ancak veri iletim hattında yer alan verileri analiz ederek ve anlamlandırarak veri iletim hattına yeni veriler ekleyebilir. Bu duruma da kısaca bir örnek vermek gerekirse; bir önceki örnek ile aynı sistemde iken bu sisteme dâhil olan iki cihazın sensör verisi oluşturup veri iletim hattı ile bu iletme görevi olsun. Bu sisteme dâhil olan diğer cihaz ise bu sensör verisine göre bir röle çıkışı açıp kapatma görevine ve veri iletim hattına röle durum bilgisini iletme görevine sahip olsun. Veri kayıt birimine iki cihazdan sensör verileri gelmekte ve bir cihazdan da röle durum bilgisi gelmekte iken veri kayıt birimi de röle açıp kapatma yeteneğine sahip olan cihaza sensör verilerine göre değerlendirme de bulunup röle aç ya da kapat görevini

gerçekleştirmesini sağlayabilir. Öylece veri kayıt birimi sadece kendine gelen verileri kayıt etme dışında kendisine röle aç ya da kapat mesajı iletilmemesine rağmen karar veren ve bu mesajı üreten rolde görev olabilir. Bu gibi görev belirleyen ileten yapılar sistemlerde görev bilgisayar ya da görev sayar olarak adlandırılır.

Elektronik cihaz haberleşmesi ile bu haberleşme yöntemlerini kullanabilen veri kayıt birimleri, veri hizmet araçları ve uygulamaları ile ilgili çalışmaların incelenmesi dışında bu konunun geniş bir alan kaplaması nedeniyle tez ile birlikte yürütülebilecek faaliyet imkânı da fazladır. Birçok alanda bu tezin çıktılarının kullanılması amacıyla elektronik cihazların genelinde en çok kullanılan elektronik haberleşmelerden biri olan seri haberleşmenin incelenmesi ve tasarlanacak masaüstü uygulama ve elektronik cihaz ile seri haberleşme örneklendirilmesi amaçlanmıştır.

Günümüzde özellikle akıllı cihaz teknolojisinin gelişmesi ile kablosuz haberleşmesinin kullanılabilirliğinin artması nedeniyle elektronik haberleşmeyi konu alan bu tezde TCP(Transmission Control Protocol)/IP(Internet Protocol) haberleşme protokülünde incelenmesi ve tasarlanacak masaüstü uygulama ve elektronik cihaz ile TCP/IP haberleşme protokolünün örneklendirilmesi amaçlanmıştır.

Elektronik haberleşme, elektronik cihaz teknolojisi için çok önemli bir konu olmak ile beraber bu haberleşme çeşitliliğinden kaynaklı bazı ihtiyaçlar doğmuştur. Konuya örnek vermek gerekirse bir firmanın ürettiği ve birbirlerine seri haberleşme ile bağlanabilen iki elektronik cihaz zaman içerisinde gerek teknolojik gelişimlerden gerek son kullanıcı isterlerinden gerekse yeni ürün yaratma ihtiyaçlarından dolayı değişime uğrayabilir. Bu değişimle birlikte seri olarak birbirlerine bağlanan bu iki cihaz artık kablosuz olarak birbirlerine bağlanması ihtiyacı oluşabilir. Ancak en az maliyet ve değişiklikle bu işlemi yapmanın bir yolu seri haberleşmenin kablosuz haberleşme teknolojisine uyumlu hale gelmesi için kablosuz haberleşme protokollerinden TCP/IP ile veri haberleşmesine devam etmesidir. Bu tarz bir dönüşüm için tasarlanmış olan dönüştürücü cihazları mevcuttur. Bir cihazın haberleşmesini seri haberleşmeden TCP/IP haberleşmesine dönüştürmek ya da tam tersi işlem yapmak günümüz teknolojisinde mümkündür. Ancak bu tarz cihazların maliyetlerin yüksekliği ve performans olarak yetersizliği en çok karşılaşılan durumlardandır. Bu tez çalışmasında tasarlanacak elektronik cihaz ve masaüstü uygulaması ile seri haberleşmeden TCP/IP haberleşmesine dönüştürmek ya da tam tersi işlemlerin yapılabilmesi amaçlanmaktadır.

Veri haberleşmesinin kaydedilmesi için veri iletim hattına bağlı olmak gerekmektedir. Elektronik cihazların veri haberleşmesini sağlamak için kablolu veya kablosuz olmak üzere birbirlerine bağlanması ile bu bağlantı üzerinden veri iletimi yapıldığı bilinmektedir. Veri iletim

hattına bağlanmak için veri iletim hattında bulunan elektronik cihazlar ile bağlantı kurmak gerekmektedir.

Genellikle veri kayıt işlemi yapan cihazlar bir sistemdeki tüm elektronik cihazlara bağlı olan, bu cihazları yönetiminden sorumlu ana sunucu, görev sayar yada master cihazlardır yada bu yapıdaki bir cihaza bağlı olan sadece kayıt görevi olan cihazlardır. Bu şekilde sistemin bir parçası olan elektronik cihazlar veri haberleşmesinin iki yönü olan veri alıcı ve veri gönderici tarafındaki verileri kayıt altında tutabilir. Ancak sistemine göre değişmek ile beraber verinin kapladığı alan arttıkça genellikle kayıt yapan cihazlar için temel bir problem olan yetersiz hafıza alanı problemi karşımıza çıkar. Bu nedenle fiziksel olarak sınırlı bir hafıza alanı ile yetinmek yerine günümüzde bulut teknolojisi ile sanal veri depolama alanları kullanılmaktadır. Ancak bazı sistemler bu sisteme geçiş yapmak için gerek sistem standartları olarak gerekse teknolojik olarak yetersizdir. Bu gibi sebeplerden dolayı veri kayıt cihazları çeşitliliği çok olan bir elektronik haberleşme cihazları olarak dikkat çekmektedir. Bu tez çalışmasında tasarlanacak masaüstü uygulaması ile veri kayıt işleminin örneklendirilmesi amaçlanmıştır.

2. LİTERATÜR ARAŞTIRMASI

Haberleşme insanlığa ait en temel ihtiyaçlardan biridir. Bu yüzden her alanda haberleşmeye ihtiyaç oluşmaktadır (Natale, 2008). Haberleşmenin bileşenlerinden verici, alıcı, veri yolu ve veri de bu nedenle her alanda karşımıza çıkmaktadır. Veri yolu ve veri haberleşmenin türünün belirlenmesi konusunda en önemli iki bileşendir. Elektronik cihazlar, elektronik haberleşme türlerini kullanarak veri alışverişini sağlarlar. Örneğin; seri haberleşme elektronik cihazlarda çok sık kullanılmaktadır (İbrahim, 2008).

Elektronik cihazların durum kontrol işlemleri elektronik haberleşme ile alınan verilere göre gerçekleştirilmektedir (Bora vd., 2014).

Elektronik cihazlar veriyi elde etmek için alt algılayıcılar ve bileşenler kullanabilir. Ar-Ge(Araştırma ve Geliştirme) çalışmalarında ve ürün kalitesinin testi sırasında elde edilen verileri toplayan sistemlere "Veri Toplama Sistemleri" denmektedir (Burunkaya vd., 2011). Elektronik cihazlar topladıkları verileri işlenerek bilgi haline getirebileceği gibi veri iletimi ile bu verinin işlenmesini başka bir ortam da yapılmasını da sağlayabilir. Veri haberleşmesi ile veri toplama, işleme ve analiz adımları gerçekleştirilmektedir ve bu işlemler birçok alanda sıklıkla kullanılmaktadır (Seyfettin vd., 2014).

(Watterson ve Heffernan, 2007) çalışmasıyla gömülü sistemlerde çalışma zamanında giriş çıkış verilerinin doğrulanma ve izlenmesine yönelik işlemlerin basitleştirilmesinden bahsedilmiştir.

(Karapınar K., 2016) çalışmasıyla gömülü sistemlerin seri haberleşmesi PIC(Peripheral Interface Controller) ve Arduinio ile hazırlanan ortamda iki elektronik cihaz haberleşmesi ile gösterilmiştir.

(Gültekin, 2005) çalışmasıyla veri toplama ve kayıt üzerine "GLİVET" adı verilen elektronik bir cihaz yapılmıştır. Çalışma da kısa sürede veri toplama ve kayıt sağlayan elektronik cihaz oluşturmaya yarayacak uygun maliyetlerde bir alt yapı sunulmaktadır.

(Tokel, 2009) çalışmasıyla masaüstü uygulaması ve PIC arasında asenkron ve senkron seri haberleşmeye yönelik bir çalışma yapılmış veri hizmet uygulamasının gerçekleştirilmesinde Visual Basic 6.0 kullanılmıştır. (Çayıroğlu ve Şimşir, 2008) çalışmasıyla PIC ve step motorla sürülen bir robotun uzaktan kamera sistemi ile kontrolüne

yönelik çalışma yapılmıştır. (N. Topaloğlu, 2012) çalışmasıyla hazırlanan web kameranın hareketinin ve verisinin kontrolü ile uzaktan sınav gözetleme çalışması yapılmıştır. Sistemde yer alan uygulama veri izleme ve veri kontrolü sağlayan bir Veri hizmet uygulamasıdır.

(Kürklü ve Çağlayan, 2007) çalışmasıyla PIC ile geliştirilen cihaz RF(Radio Frequency) kullanarak veri iletimi sağlamaktadır. Bu sistem sayesinde modern seralarda kullanılacak bir iklim kontrol sistemi geliştirilmesine yönelik bir çalışma yapılmıştır ve Delphi 6.0 ortamında hazırlanan arayüz ile veri izleme sağlanmıştır. (Benzer, 2013) çalışmasıyla akıllı veri toplama için PIC ile kızılötesi ve radyo frekansı kullanılmıştır.

(Akca, 2013) çalışmasında mera hayvancılığında hayvanlarının sağlığının gözlemlenmesi için geliştirilen cihaz ile sensör verilerinin izlenmesini ve kayıt altında tutulması için MSSQL(Microsoft Structured Query Language Server) ve C# ayrıca web arayüzü için Asp.Net kullanılmıştır.

(İnan ve Çakmak, 2008) çalışmasıyla PIC ile elde edilen ölçüm değerlerinin RS-485 aracılığıyla basic dilinde hazırlanmış masaüstü uygulamasında izlenim ve kontrolü sağlanmıştır.

(Gökozan ve Taşkın, 2012) çalışmasıyla güç kalitesi izleme amacıyla anlık veri izleme ve veri kayıt işlemini veri hizmet uygulaması hazırlanmıştır. Bu veri hizmet uygulaması LabVIEW tabanlı olarak hazırlanmıştır. LabView "National Instruments(NI)" şirketi tarafından hazırlanmış bir programdır ve program test, ölçüm, otomasyon, cihaz kontrolü, veri toplama, veri analizi gibi birçok alanda kullanılan uygulamalar için yaygın olarak kullanılmaktadır (Ertugrul, 2002).

(Divan vd., 2002) çalışmasıyla "I-Grid" adı verilen web tabanlı enerji kalitesi izleme sistemi ve DSP tabanlı "I-Sense" adı verilen güç kalitesi izleme cihazı geliştirilmiştir. (R. P. Lee vd., 2002) çalışmasında enerji kalitesi izlemek için bir cihaz ve PEGASYS yazılımı kullanılarak, web tabanlı izleme sistemi oluşturulmuş ve çalışmaya ait çıktılar Hong-Kong'ta test edilmiştir. (Atalık, 2007) çalışmasında, Mini-ITX PC tabanlı internet üzerinden haberleşen bir enerji kalitesi izleme cihazı tasarlanmıştır.

(Güngör, 2015) çalışmasında kaçak elektrik kullanımının GSM(Global System for Mobile Communications) aracılığıyla takibine yönelik Arduino ile cihaz hazırlanmıştır ve elektrik sayaçlarına bağlanarak kaçak kullanımı tespitini GSM ile bildirmektedir. Cihaz hem kaçak durumuna yönelik test cihazı özelliğinin dışında veri iletişimde köprü görevi görmektedir.

(Güçlü vd., 2018) çalışmasıyla sayaçtan Arduino UNO R3 ile okunan verilerin W5100 Ethernet Shield sunucu olarak ayarlanmasıyla birlikte uzaktan web arayüzü ile izlenmesi sağlanmıştır.

(Demir, 2006) çalışmasında GPRS(General Packet Radio Service) modem ile sayacın RS-485 ve optik portunu kullanılarak uzaktan sayaç okumaya yönelik model geliştirilmiştir. (Şahin, 2007) çalışmasında, uzaktan sayaç okumak için GPRS haberleşmesi ve web uygulama gerçekleştirilmiştir. (Süzer, 2006) çalışmasında uzaktan sayaç okuma teknikleri ve Modbus-RTU, IEC 61107 MOD C Protokoller ile veri hizmet uygulaması olarak kullanılacak bir Visual Basic 6.0 ile uygulama yazılımı gerçekleştirilmiştir.

(Yıldırım vd., 2018) çalışmasıyla elektrik enerji kalitesini çevrimiçi izlemek için FPGA(Field Programmable Gate Array) ile bir cihaz yapılmıştır ve bu cihaz UDP(User Datagram Protocol) haberleşme protokolü ile arayüz programına veri iletimi sağlamaktadır. (Xu vd., 2012) çalışmasında FPGA ile güç kalitesi izleme sistemi oluşturulmuş ve EtherCAT ve PowerPC teknolojilerinden yararlanılmıştır.

(Köksal, 2014) çalışmasıyla FPGA ile analog haberleşme verilerinin kayıt edilmesi ve kayıt edilen verilerin işlenmesine örnek verilmiştir ve seri haberleşmeye yönelik anlatımda bulunulmuştur.

(Erkol ve Demirel, 2013) çalışmasıyla FPGA ile bilgisayar ya da seri porta sahip donanımlarla seri haberleşme sağlayabilecek bir donanım hazırlanmıştır. Benzer çalışmalar (Fang ve Chen, 2011), (Fongjun vd., 2011) ve (Niu vd., 2012) çalışmalarında da yapılmıştır.

(Arıcı ve Tenruh, 2014) çalışmasıyla PIC, seri haberleşme ve LabVIEW ile veri toplama alanında bir çalışma yapılmıştır. Benzer bir çalışma (Güner, 2005) çalışmasıyla da yapılmış bu çalışmada ayrıca veri işleme ve veri izleme konuları üzerinde durulmuştur.

(A. S. Yılmaz ve Görmemiş, 2011) çalışmasıyla modbus portokolü kullanılabilen bir ölçüm cihazı için delphi ile veri analiz ve veritabanına kayıt işlemlerini gerçekleştirebilen bir veri hizmet uygulaması hazırlanmıştır ve bu uygulama enerji izleme yazılımı olarak verilmiştir.

(Erkan, 2007) çalışmasıyla PLC ünitelerinden seri haberleşme ile alınan veriler Visual Basic 6.0 ortamında geliştirilen arayüz ile dokuma makinelerinde verimliliği artırma üzerine kullanılmıştır.

(Tolga vd., 2020) ile CAN(Controller Area Network) haberleşmesi ile sensörlerden alınan veriler STM32f103C8 mikroişlemcisi ile işlenip C# ile hazırlanmış masaüstü uygulamasına RS232 seri haberleşme protokolü ile iletilerek bu uygulamayla veri izleme ve kaydı yapılmıştır.

(Tunç, 2007) çalışmasıyla analog sismometrelerden sayısal veriler seri haberleşme ile masaüstü yazılıma aktarılmasına ait örnek vardır.

(Akin, 2009) çalışmasıyla sayısal sinyal işleme ve mikro denetleyiciler kullanılarak enerji kalitesi problemlerinin belirlenmesi ve web sunucusuna aktarılması ile ilgili bir çalışma gerçekleştirilmiştir.

Elektronik cihazlar kullanıldığı her sektör ve alanda cihaza ait varsa yazılımın uygunluğu büyük önem barındırmaktadır. Yazılımın uygunluğunu belirlenmek için yapılan testlere yazılım testleri denir. Yazılım testi, yazılımdaki hataları, riskleri belirlemek ve uygulamayı hedeflenen seviyeye ulaştırmak ve doğrulamak için yapılan testlerin tamamıdır (Muller ve Friedenber, 2011).

Özellikle askeri alanda kullanılan elektronik cihazlar ciddi sorumluluklar barındırdığı için bu cihazlara ait yazılım testleri ve bu cihazlarda kullanılan elektronik haberleşme bir hayli önem arz etmektedir.

(Ilyas ve Mahgoub, 2004) çalışmasında askeri alanda kullanılmak üzere kablosuz haberleşme üzerine uygulama hazırlanmış ve askeri alan için önemine vurgu yapılmıştır.

(Karasoy ve Çinar, 2014) çalışmasıyla ULAK adı verilen farklı platformlar arasındaki haberleşmeyi sağlayabilme, uzak metot çağırısına imkan verme ve haberleşmeyi asenkron ya da senkron gerçekleştirebilme yeteneklerine sahip dağıtık sistemler için haberleşme otomasyon ara katmanı anlatılmıştır. Bu çalışmanın çıktıları askeri alanda kullanılmaktadır. Bu çalışmayı kısaca özetlemek gerekirse birden fazla cihaz, uygulama ya da aracın bağlandığı ve birbirleri ile olan haberleşmesinin yönetildiği katmandır. Bu çalışma için veri hizmet uygulamasının tüm gereksinimlerini sağlayabilecek nitelikte bir çalışma demek mümkündür. Elektronik cihaza ait testler sadece standartları sağlanıp sağlanmadığını göstermek için değil aynı zaman da cihazın çalışabilir, faal ve kullanılabilirliğini göstermeyi de hedeflemektedir. Bu nedenle bazı elektronik cihazlar bulunduğu sistem itibari ile başka bir elektronik cihaza elektronik haberleşme yöntemi bağlıdır ve bağlı olan bu cihazında testinin tamamlanmış ve doğrulanmış olması gerekmektedir. Bu gibi durumlar için test ortamında bağlı olan cihaz yerine bağlı olan cihazı taklit eden doğrulanmış yazılım uygulamaları ya da cihazları kullanılabilirliktedir.

Bu yazılım uygulamaları ya da cihazları da bağı olan cihaz gibi çalışmayı sağlamak için bağı olan cihazın sağladığı elektronik haberleşmeyi de taklit etmelidir. Elektronik haberleşmeyi taklit edebilen bu yazılımlar, uygulamalar ya da cihazlar karşımıza veri hizmet araç ve uygulamaları olarak gelmektedir.

(Senem ve Keleş, 2017) çalışmasıyla askeri alanda kullanılan gömülü ve gerçek zamanlı yazılımlarda giriş-çıkış verilerinin izlenmesine yönelik veri hizmet uygulaması yapılmıştır.

(Gevher vd., 2016) çalışmasıyla askeri alanda testlerde kullanılan simülatör yazılımlarının ortak bir altyapı ile hazırlanması amacı ile Genel Amaçlı Simülatör yazılımı (GAS) geliştirilmiştir. Çalışmada farklı arayüz ve protokolleri (TCP/IP, UDP, ARINC(Aeronautical Radio Incorporated) 429, CAN Bus, Ayrık hat, RS-232/RS-422, MIL-STD(Military Standard)-1553B) ortak bir mesaj tanımı şablonu ile tanımlayıp, dinamik olarak güncellenebilen kullanıcı arayüzü geliştirilmesini sağlayan Genel Amaçlı Simülatör Yazılımının tasarımı ve kullanımı değerlendirilmiştir. GAS yazılımı, önceden hazırlanan derlenmiş bileşenleri (*.dll) kullanabilme ve derlenmeye dahil olacak C# sınıflarının gerekirse sahada dahi güncellenebilmesine izin verme yeteneği ile, benzer görevlere sahip simülatör yazılımlarının çoğaltılmasını kolaylaştırmaktadır.

(Aykurt, 2017) çalışmasıyla askeri kara araçlarında test amaçlı kullanıcı arayüz yazılımı geliştirme konusu üzerinde durulmuş ve örnek bir çalışma yapılmıştır.

Ayrıca (Myers ve Rosson, 1992) ve (Gajos vd., 2006) çalışmalarında veri hizmet uygulaması olarak kullanılan kullanıcı arayüzlerine dair çalışmalar yer almaktadır.

(Avcıoğlu, 2015) çalışmasıyla ile seri haberleşme sağlayan C# kullanılarak masaüstü uygulamalar hazırlanmıştır. Bu uygulamalar askeri alandan kullanılmıştır. Çalışmaya ait uygulamaların veri hizmet uygulaması olması ve askeri alanda test amacıyla kullanıldığının örneklenmesi benzer çalışmalarında askeri sektörde kullanılabileceğini göstermektedir.

Yönetilebilir test hizmetlerinin otomatikleştirilmesi ile maliyette azalma, standart, güvenilir bir sistem oluşturulurken veri hizmet araç ve uygulamalarının test araçları olması sisteme güvenilirlik ve istikrar kazandırır (Whittaker vd., 2012).

Test otomasyonu; test adımlarını araç veya uygulama yardımıyla otomatik hale getirmektir. Test otomasyonlarının oluşturulmasındaki asıl hedef hataları bulmak değil süreci otomatik hali getirmektir (Dustin vd., 2009). Test otomasyon yazılımları testlerin

daha hızlı gerçekleştirilmesi sağlanabilmektedir (Alégroth ve Feldt, 2013), (Borjesson ve Feldt, 2012)

(Bayır ve Elen, 2009) çalışmasında uzaktan eğitim amaçlı bir otomatik test standının hazırlanan kullanıcı arayüz yazılımı ile veri izleme dışında tehlikeli durumlarda sistemi durdurma yeteneğine sahip olması veri hizmet araçlarına verilen yetki ile sağlanabilecek imkânları göstermektedir.

Yazılım testlerinin amacı hataları ve riskleri bulmak iken testlerin hedefi ise kaliteli ürün, müşteri memnuniyeti, maliyeti azaltma hatta can ve mal kaybını önlemektir (Goucher ve Riley, 2009).

Veri hizmet aracı ve uygulamaları test ortamlarının dışında elektronik haberleşme hattında yer almalarının avantajı ile kayıt alma işlemini gerçekleştirebiliyorsa veri kayıt birimi olarak da kullanılmaktadır.

(Grossi, 2006) çalışmasında havacılık sektöründe kullanılan kayıt cihazlarına dair bilgilendirmeler vermiştir ve kayıt cihazlarının önemini kazaları önlemede uçuş verisinin kullanımı olan ihtiyaç ile vurgulamıştır.

(Watthanawisuth vd., 2012) çalışmasında araç kazalarında kullanılabilecek kablosuz kara kutu ile ilgili çalışmalar yapılmıştır.

(Yolcu ve Şahan, 2017) çalışmasında araç arıza tespit cihazı hazırlanmıştır. Bu elektronik cihaz araç ile CAN haberleşmesi ile aldığı verileri seri haberleşme ile çalışma sırasında oluşturulan kullanıcı arayüzüne seri haberleşme ile veri iletimi sağlayarak veri izleme özelliği barındırmak dışında veri kayıt işlemi özelliği de barındırmaktadır. Bu çalışmaya benzer bir çalışmada (Jie vd., 2015) mevcuttur. (Şimşek ve Taşdelen, 2015) çalışmasıyla ile CAN haberleşme verileri Arduinio üzerinden internete ve SignalR teknolojisi ile son kullanıcıya ulaşmaktadır. Geliştirilen kart köprü görevi görmektedir.

Veri kayıt için veri iletim hattında yer alan elektronik cihazların bazıları veri iletim hattına veri dâhil edebilir. (Çavuşoğlu ve Kırmızı, 2010) çalışmasıyla seri haberleşme ile masaüstü yazılımdan gönderilen veriler alan PIC ile oluşturulmuş cihaz RF kullanımı ile DC(Direct Current) motorlu mobil bir aracı kontrolü sağlanması örneklenmiştir. Cihaz görev sayar cihazına örnek olarak verilebilir.

(Uyar, 2013) çalışmasında kablosuz veri toplama cihazına örnek oluşturmak için RF ile alıcı olarak çalışan PIC ve bu cihaza veri sağlayan başka bir PIC hazırlanmış ve bu veri

alışverişinin izlenip kayıt edilebilmesi için alıcı PIC ile seri haberleşen bir Visual Basic ile masaüstü yazılımı oluşturulmuştur. (Ateş ve Şaykol, 2015) çalışmasında nRF24L01 modülü ile verilerin gerçek zamanlı izlenmesine yönelik çalışma yapılmıştır. (Uzun, 2016) çalışmasında RF alıcı veri modülü ile daha uzak mesafelerdeki veri alış verişine ait bir çalışma yapmıştır.

(Kayıkçıoğlu ve Gangal, 2017) çalışmasıyla Arduino ile hazırlanan parmak izi okuma özelliği olan elektronik cihaz Bluetooth ile okunan veriyi sunucu yazılımını göndererek sunucu da yer alan veritabanı ile eşleştirme yaparak yoklama sistemi sağlanmıştır. Bu çalışmada hazırlanan elektronik cihaz sensör ile sunucu arasında köprü görevi görmektedir. Benzer çalışmalar (Karabulut, 2010) ve (Çınar, 2014) ile yapılmıştır.

(Umang ve Mitul, 2015) çalışmasında AMI, WIMAX, Zigbee haberleşme teknolojilerinde yaptığı kıyaslama da akıllı ölçme uygulamalarında Zigbee tercih edilmesi gerektiğini; daha fazla batarya ömrü, ucuz maliyeti ve daha güvenli olması sebepleriyle sunmuştur. (Chae vd., 2012) çalışmasında inşaat sektöründe kullanılmak üzere sensör verilerinin Zigbee ile veri izleme ortamına aktarılarak asma köprü durumunu incelenmiştir.

(Barroca vd., 2013) çalışmasında inşaat sektöründe faydalanması için beton yapı içerisindeki sıcaklık ve nem değişim verileri kablosuz haberleşme toplamaya yönelik çalışma yapmışlardır.

(López vd., 2009) çalışmasında balık çiftliğindeki pH, amonyum ve sıcaklık verilerinin kablosuz olarak takip edilebilmesi için bir çalışma yapılmıştır.

(Karnouskos ve Izmaylova, 2009) çalışmasında web servis özelliğinin uzaktan veri izlemedeki öneminden bahsetmiştir. (Üçgün vd., 2020) çalışmasında Raspberry Pi 3 kontrolcü kartı ile iç ortam hava kalitesi izleme adına bir sistem hazırlanmıştır. Bu sistemde veri aktarımı Raspberry Pi 3 kartı içerisindeki Raspbian işletim sistemine kurulan MySQL(My Structured Query Language) sunucusu ile veri tabanına kaydedilmekte ve veri tabanına kaydedilen veriler web servis aracılığıyla web arayüzüne aktarılmaktadır.

Akıllı ev sistemlerinde yer alan birçok elektriksel cihaza ait veriler internet üzerinden görüntülenebilmektedir. Ayrıca akıllı ev sistemlerinde kullanılan birçok uygulama, elektriksel cihazların sadece aç/kapa işlevini yerine getirmek ile yetinmeyip birçok özelliğin kontrolüne imkân sağlamaktadır (Nunes ve Delgado, 2000). Akıllı ev sistemlerinde sensör verilerine göre işlemler yapılması (K. C. Lee ve H.-H. Lee, 2004) çalışmasında da bahsedilmiştir.

Birçok uygulama, elektriksel cihazların sadece aç/kapa işlevini yerine getirmek ile yetinmeyip birçok özelliğin kontrolüne imkân sağlamaktadır (Baeg vd., 2007).

Sağlık sektöründe uzaktan hastanın fizyolojik değerlerini takip etmek elektronik cihazların oluşturulduğu bilinmektedir (Hardalaç vd., 2002). (Lin vd., 2004) çalışmasında sağlık izleme hizmetindeki ilerlemeler için haberleşme teknolojilerinin gelişmesinin önemine vurgu yapmıştır. (Türker, 2010) çalışmasında elektrokardiyografi sinyalini Zigbee kullanarak veri iletimini ve iletilen veri seri haberleşme ile bilgisayar arayüz yazılımına aktararak veri izlemesi sağlanmıştır.

(E. Y. Topaloğlu, 2004) çalışmada kan gazı cihazından seri haberleşme ile alınan veriler masaüstü yazılım ile analiz edilip başka bir masaüstü yazılıma gönderilmektedir. Veri iletimi için köprü görevi görmesi ve veri analizi yaparak kullanıcıya gösterebilmesi özellikleri barındıran bu uygulama veri hizmet uygulamalarına bir örnek oluşturmaktadır.

(Ha vd., 2020) çalışmasında hava kalitesi yönetimi için sensör verisinin ve veri iletiminin önemine vurgu yapılmıştır. (Jo vd., 2020) çalışmasında, "Smart-Air" isimli IoT(Internet of Things) ve bulut teknolojisini kullanan bir elektronik cihaz geliştirmişlerdir.

(Marques vd., 2019) çalışmasında ESP8266 modülünü kullanmış ve sensör verilerinin aktarımını ve depolanmasına ait çalışma yapılmıştır. Ayrıca aynı çalışma da "iAirCO2" adlı uygulama ile web ve mobilde ortamdaki CO₂ miktarının ait verilerin izlenmesi sağlanmıştır.

(Sung vd., 2019) çalışmasında Arduino Uno ile sensör verileri olarak ESP8266 modülü ile kablosuz haberleşme ile veri tabanına kaydetmişlerdir ve verilerin izlenmesini C# masaüstü uygulaması ile yapmışlardır.

(Zhao vd., 2019) çalışmasında IoT tabanlı iç ortam hava kalitesi detektörü tasarlanmıştır ve bu cihaz haberleşmesinde farklı senatyolar için RS-485, LoRa, Wi-Fi(Wireless Fidelity), GPRS ve NB-IoT iletişim arayüzlerine dair örnekler yapmışlardır.

(Kelebekler, 2019) çalışmasında nesnelerin interneti sensörlerden alınan global ışımaya, rüzgar hızı ve yönü, ortam sıcaklık ve bağıl nem verileri 32 bit Atmel SAM3X8E ARM(Acorn RISC Machine) Cortex-M3 işlemcili Arduino Due mikrodenetleyici tarafından toplanarak ESP8266 Wi-Fi modül üzerinden bulut ortamına taşınmıştır.

(Pereira vd., 2018) çalışmasında Raspberry Pi ile gerçek zamanlı bulut server üzerine veri kayıt ve web server ile veri izleme ile örnek çalışması yapılmıştır. Aslında

sistem PIC ile tasarlanmış bir cihaza ait verilerinin Raspberry Pi tarafından bulut server yapısında depolanmasına yaramaktadır. Yani bu çalışma da Raspberry Pi veri kayıt ve veri izleme için köprü görevi görmektedir.

(A. K. Yılmaz ve Acar, 2017) çalışmasıyla büyük veri mimarisi için IoT cihazlarından 3G aracılığı verinin toplanması, işlenmesi, veri tabanına aktarılması ve REST(Representational State Transfer) servisler üzerinden kullanıcıya sunulmasına yönelik çalışma yapılmıştır.

(Inner, 2017) çalışmasında veri kayıt ve izleme üzerine bir çalışma yapılmıştır. Raspberry Pi 3 kullanarak Bluetooth ile güneş panelinden alınan verileri sunucuya ileterek sunucuda yer alan veritabanına kayıt yapabildiği ve veritabanındaki verileri web servisler ile çalışma sırasında oluşturulan kullanıcı arayüz uygulaması ile izlenmesini sağlamıştır.

(Cristaldi vd., 2016) çalışmasında güneş panellerini izleme ve arıza tespit sistemi üzerinde çalışma yapılmıştır. (Shariff vd., 2015) ve (Choudhury vd., 2015) çalışmalarında güneş paneli verileri ZigBee veri haberleşmesi iletimi ve takibi sağlanmıştır. (Fuentes vd., 2014) çalışmasında fotovoltaik (PV) sistemlere ait verilerin düşük maliyetle izlenmesi Arduino ile veri toplama çalışması yapılmıştır.

3. MATERYAL VE YÖNTEM

Elektrik, madde yüklerinin varlığı ya da yük hareketleri ile oluşan enerjidir. Elektronik, elektrik kullanarak bilgi işleyen, taşıyan veya depolayan elemanları ve sistemleri inceleyen bilim dalıdır.

Veri, ham gerçek enformasyon parçasığına verilen addır. Verilerinin anlamlı halde birleştirilmesi ve işlenmesi ile bilgi elde edilir. Anlamlı bir bilginin karşılıklı alış verişine haberleşme denir. Elektronik haberleşme ise elektronik veri alış verişine denir.

Tüm haberleşme sistemleri için temel olarak veriyi gönderen kaynak (verici), veriyi ileten haberleşme kanalı (veri iletim ortamı), iletim ortamında veriyi etkileyebilecek ve bozulmalara neden olabilecek gürültü ve veriyi alan hedef (alıcı) öğeleri vardır. Bu en ilkel duman ile haberleşmeden en yenilikçi kablolu haberleşmeye kadar tüm haberleşme sistemleri için geçerlidir.

Elektronik veriler analog ve dijital veriler olarak ayrılır. Daha doğru bir ifade ile zamana göre değişen ve sürekli bir özellik gösteren veriye analog veri denir. Görülen, işitilen, hissedilen tüm fiziksel büyüklükler birer analog veridir ve değer aralıkları vardır. Dijital veriler ise orijinal verinin değerlerinin belirli zaman aralıklarıyla örneklenmesi sonucu elde edilen var ya da yok diğer bir deyişle 1 ya da 0 verisidir. Dijital veriler analog verilerin değer aralıklarına göre sınırlandırılan mantıksal verilerdir. Örneğin; bir sistemde yapılan sıcaklık ölçüm değerleri analog veriler iken bu sıcaklık değerinin belirli bir değer üstüne sıcak ve belirli bir değer altına soğuk denmesi ile elde edilen sıcak ve soğuk verileri dijital verilerdir. Bir analog değer belirlenen değerlerin üstünde dijital 1 olarak ifade edilmesi ve belirli bir değer altında dijital 0 olarak ifade edilmesine mantıksal veri denir.

Elektronik veriler gibi veriyi ileten haberleşme kanalı kılavuzlu ve kılavuzsuz iletim ortamı olarak ayrılır. Kılavuzlu iletim ortamı aslında fiziksel bağlantıyı yani bir diğer deyişle kabloyu ifade ettiği için kablolu haberleşme olarak ifade edilir. Kılavuzsuz iletim ortamı ise herhangi bir fiziksel bağlantı yerine hava, su, boşluk gibi doğal ortamları ifade ettiği için kablolu haberleşme olarak ifade edilir.

İletim ortamında veriyi etkileyebilecek ve bozulmalara neden olabilecek gürültü, verici tarafından gönderilen verinin alıcı tarafından farklı alınmasına neden olabilir. Bu farklılıklar analog veriler için sinyal kalitesinin bozulmasına, dijital veriler için bit

hatalarının oluşmasına (1 bitin 0 bite ya da 0 bitin 1 bite dönüşmesi) olarak gözlemlenebilir. Özellikle kablosuz sistemlerde elektromanyetik dalganın üretildiği kaynaktan uzaklaştıkça aradaki mesafenin karesiyle ters orantılı olarak azalır. Elektromanyetik dalgalar, alıcıdan vericiye ulaşana kırılma, saçılma, girişim ve emilim gibi nedenlerden ötürü de enerji kaybına uğrayabilirler. Zayıflamaya ilaveten elektriksel gürültü, dalga girişimi ve çoklu yol etkisi gibi faktörler de kısmen ya da tamamen bozulmalara neden olabilir. Alıcı ve gönderici tarafına yakın olan sargılı motorlar ve enerji hatları da başlıca elektriksel gürültü kaynaklarıdır.

Bilgisayar veya entegreler ile iletişim kurabilmek için seri haberleşme ve paralel haberleşme gibi yöntemler kullanılır. Paralel iletişim; her bir bilgi için ayrı bir veri iletim hattı vardır. Örneğin; 8 bitlik veri gönderimi için 8 tane veri iletim hattı gereklidir. Paralel haberleşmede tüm bitler aynı anda veri iletim yolundan gönderilir. Bu nedenle seri iletişime kıyasla çok daha hızlıdır. Fakat paralel iletişim uzun mesafe veri iletimi için uygun değildir. Çünkü mesafe uzadıkça veriyi etkileyebilecek gürültü ve veride bozulmaların görülmesi daha olasıdır. Ayrıca paralel haberleşme seri haberleşmeye göre daha pahalıdır. Bu da veri iletim yolunun fazla olmasından kaynaklıdır. Seri haberleşme paralel haberleşmeye göre yavaş olmasına rağmen, hem daha uzun mesafelere veri iletimi için uygun olmasından hem de daha ucuz olmasından dolayı tercih edilir. Seri haberleşme de veriler tek bir veri iletim hattı üzerinden gönderilip, tek bir kanal üzerinden alınır. Bilgisayar sistemlerinde, ana kartların içerisindeki mesafeler kısa olduğundan maliyeti çok fazla olmayacağı için ve daha hızlı veri iletimi sağlamak amacıyla paralel haberleşme tercih edilir.

Veri iletim hatları alıcı ve verici arasındaki iletim yönü çeşitliliğine göre Simplex, Half Duplex ve Full Duplex olarak üçe ayrılır. Simplex hatlar; tek yönlü veri iletim hattıdır. Verici tarafından veri alıcıya iletilmesi dışında başka veri iletim yönü yoktur. Örneğin; televizyon ya da radyo yayını Simplex veri iletim hatlarını kullanır. Half Duplex hatlar; tek hat üzerinden çift yönlü veri iletim hattıdır. Verici tarafından veri alıcıya iletilmesi dışında alıcı tarafından veri vericiye gönderilebilir fakat bu gönderim işlemleri aynı anda olamaz. Örneğin; polis telsizi haberleşmesi Half Duplex veri iletim hattının kullanıldığı bir yapıdır. Bu nedenle polis telsizi gibi yapılar verici veri iletimi sonrasında veri almaya hazır geldiği belirtmek için “Tamam” tarzı ifadeler kullanır. Bu tarz ifadeler kullanılmasının nedeni verici ve alıcının belirlenmesidir. Verici veri iletimi sonrasında artık alıcı konumuna geçtiği belirten ifade kullanır ve veri beklemeye başlar. Alıcı ile vericinin aynı anda veri iletmeleri mümkün olmayacağı için bu haberleşme de alıcı ve verici rolü belirleme senaryoları önemlidir. Full Duplex hatlar; verici ve alıcı arasında veri iletimi çift yönlüdür ve verici alıcıya veri gönderirken alıcıdan veri de alabilmektedir. Örneğin; telefon görüşmeleri Full Duplex veri iletim hattını kullanır.

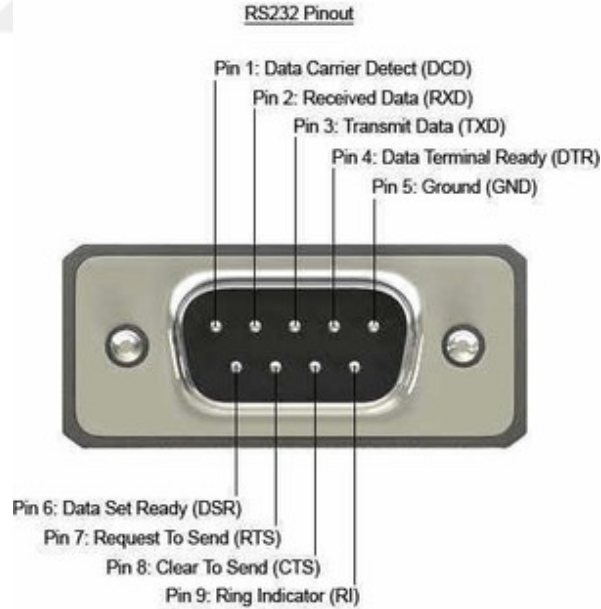
Seri veri iletim, verinin tek bir kanal üzerinden peş peşe bitler halinde gönderilmesini sağladığı için rayın üzerinde ilerleyen vagonlu lokomotifine benzer. Ray veri iletim hattı iken, her bir vagon da bit olarak değerlendirilebilir. Seri iletişimde verinin başlangıç noktasını belirlemek için “Başla” biti, iletim sırasında oluşabilecek hatayı belirlemek için “Eşlik” biti, verinin bitiş noktasını belirlemek için “Dur” biti kullanılır. Seri veri iletiminde alıcı doğru haberleşme için karakter uzunluğu, başla-bitir biti ve iletim hızını bilmek zorundadır. Paralel veri iletiminde verinin tüm bitleri aynı anda iletildiği için “Başla”, “Eşlik” ve “Dur” bitlerine ihtiyaç yoktur. Paralel veri iletimi bir kargo teslimatına benzetilebilir. Alıcı ve verici belli olduktan sonra gönderilecekler bir paketlenerek vericiden alıcıya iletilir. Paket içerisindekiler alıcıya aynı anda ulaşır. Bu nedenle paralel veri iletimin doğruluğu seri veri iletiminde daha yüksektir.

Seri veri iletişimi, eş zamanlı (senkron) ve eş zamansız (asenkron) seri iletişim olmak üzere ikiye ayrılır. Çoğu sistemde veri iletiminden önce verici veriyi iletmeden önce alıcıyı uyarması gereklidir. Verici, veri iletimi öncesinde alıcıyı uyarılmazsa alıcı gelen veri için gerekli zamanı ayıramayabilir. Zamanlanmanın doğru ayarlanamaması hatalı iletişim kurulmasına neden olabilir. Bu nedenle alıcı ve vericinin haberleşmesi için uygun zamanlamanın yapılması işlemine senkronizasyon işlemi denir. Senkronizasyon işlemi sağlamak için saat sinyali kullanılır. Adından da anlaşılacağı üzere eş zamanlı (senkron) haberleşme, zamanı referans alarak yapılan haberleşmedir. Verici ile alıcı arasında veri iletim hattı dışında referans alabilecekleri saat sinyali için saat hattı da bulunmaktadır. Saat hattına uygulanan saat sinyali alıcı ve vericinin aynı zamanda çalışmasını sağlar. Bu ortak saat hattı sayesinde senkronizasyon işlemi sağlanır. Senkronizasyonu sağlanması için fazladan bir hatta daha ihtiyaç duyulmasından dolayı ve hat sayısı arttıkça verinin bozulma ya da gürültüye uğrayabileceği yeni opsiyonların oluşması nedeniyle uzun mesafelere veri iletiminde dezavantaj sağlar. Eş zamansız (asenkron) seri iletişim saat sinyali kullanmaz. Ancak veri iletiminde saat sinyali kullanmadan da senkronizasyon sağlanması gerekmektedir ve bunun için “Başla” ve “Dur” bitlerini kullanır. Veri transferi herhangi bir zamanda başlayabileceği için burada kritik olan kısım veri iletimin başladığını ve bittiğini alıcı tespit etmelidir. Eş zamansız (asenkron) veri iletimi, eş zamanlı veri iletimine göre yavaştır. Eş zamansız (asenkron) seri veri haberleşmesi, karakter-tabanlı iletişimlerde daha yaygın olarak tercih edilirken blok-tabanlı veri iletimlerinde eş zamanlı (senkron) seri veri haberleşmesi tercih edilir.

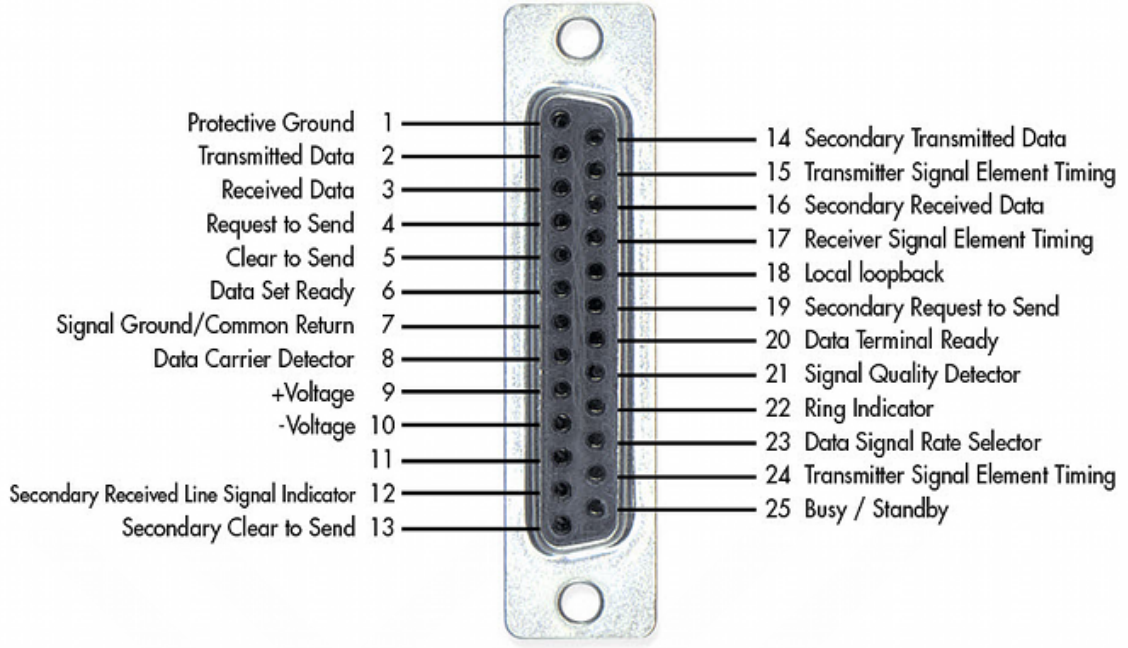
Seri veri haberleşmesinde veri iletim hızı saniyedeki bit sayısı (baudrate) olarak adlandırılır. Alıcı ve verici aynı frekansa ayarlanmalıdır aksi takdirde ilk bit alındıktan sonra alıcı bir sonraki bitin ne zaman geleceğini bilemez ve bu durumda farklı zamanda yanlış veri alır. Bu durumlarda veri iletimi hatalı olur. Bir saniye de taşınan bit sayısı yani veri iletim hızının varsayılan değeri 9600 baud olsa bile sisteme özel olarak 4800, 9600,

57600, 115200 gibi veri iletim hızları da kullanılmaktadır. 921600 çok tercih edilmeyen sadece hızlı veri iletişimi gerektiren yerlerde kullanılır.

Haberleşme protokolü; vericinin ve alıcının belirli kurallara göre haberleşmesine denir. Verinin nasıl paketleneyeceği, bir karakterdeki bit sayısı ve veri bitlerinin nasıl başlayıp biteceği gibi kurallara göre haberleşme protokolleri çeşitlilik gösterir. RS-232, eş zamansız (asenكرون) seri haberleşme protokolüdür. RS-232, EIA (Electronic Industries Association) tarafından geliştirilmiş bir standarttır. Full Duplex yapıda çalışabilmektedir. -25 Volt ile +25 Volt arasında bir değer belirleyerek 15 metreyi bulan bir haberleşme olanağı sağlar. Daha uzun mesafeler de veri kaybına neden olabilir. Günümüzde en yaygın kullanılan seri giriş çıkış arabirim protokolü olarak kullanılır. RS-232 standardında 25 pin DB25 ve 9 pin DB9 konektörleri kullanılmaktadır. Günümüzde RS-232 için en yaygın olarak kullanılan konektör tipi DB9 olarak bilinmektedir. DB25 konektörüne ait görsel Şekil 3.2 ve DB9 konektörüne ait görsel Şekil 3.1 ile verilmiştir. DB25 ve DB9 için pinlere ait eşleştirmeler Tablo 3.1 ile verilmiştir.



Şekil 3.1 DB9 Pinout



Şekil 3.2 DB25 Pinout

Çizelge 3.1 DB9 ve DB25 Pin Eşleştirmesi

DB9 Pin No	DB25 Pin No	Sinyal Adı	Görevi
1	8	Data Carrier Detect (DCD)	Veri Taşıyıcı Tanımlama
2	3	Receive Data (R*D)	Veri Al
3	2	Transmit Data (T*D)	Veri Gönder
4	20	Data Terminal Ready (DTR)	Veri Terminali Hazır
5	7	Signal Ground (GND)	Sinyal Topraklama
6	6	Data Set Ready (DSR)	Veri Paketi Hazır
7	4	Request to Send (RTS)	Veri Gönderme İsteği
8	5	Clear to Send (CTS)	Veri Göndermeye Müsait
9	22	Ring Indicator (RI)	Çevrim Göstergesi

RS-232 standardında gönderilen veri paketlerinin formatları uygulamaya göre değişiklik gösterebilir. Veri paket formatlarından en yaygını 8-N-1 formudur. Gönderilen her bayt bir “Başla” biti ile başlar ve bunu 8 adet veri biti ile “Dur” biti takip eder. 8-N-1 formatında N harfi “Eşlik” bitinin veri paketinde bulunmadığını gösterir. Diğer formatlarda “Eşlik” biti tek veya çift eşlik biti olarak verilmektedir. “Eşlik” biti adından da anlaşılacağı

üzere tek eşlik ise kendisi de dâhil veriye ait tüm bitlerdeki dijital 1 bitlerinin sayısının tek yapmaya eğer çift eşlik ise çift yapmaya çalışır. 7-E-1 formatı da veri paket formatları arasında çok sık kullanılır. 8-N-1 formatına göre eşlik biti içeren 7-E-1 formatında gönderilen her bayt “Başla” biti ile başlar ve bunu 7 adet veri biti ile “Eşlik” biti ve son olarak “Dur” biti takip eder. 7-E-1 formatında gönderilen veri tek eşlik formatında gönderilmesi gerekirken “Eşlik” biti 1, çift eşlik formatında gönderilmesi gerekirken “Eşlik” biti 0 olmalıdır. Aksi takdirde veri bozulmuş denilir. Ancak eşlik biti çok güvenli değildir. Çünkü birden fazla bozulma durumlarını tespit edemeyebilir. Örneğin; veri tek eşlik formatta gönderilmesi gerekirken gönderilmesi gereken 7 bitlik veri 1010100 iken iletilen veri 0101010 ise “Eşlik” biti yine tek sayı ve verinin iletiminde bozulma olmadığını söyler. Bu tarz bozulmaları atlama ihtimali olduğu gibi tespit ettiği hata durumlarında hata düzeltme özelliği yoktur, sadece hata tespiti yapar.

Analog verilerin dijital verilere karşılık gelmesi için yapılan çevirime TTL çevirimi denir. RS-232 protokolünde -3 Volt ile -25 Volt arasında dijital 1 olarak ve +3 Volt ile +25 volt arasında dijital 0 olarak tanımlanırken, -3 Volt ile +3 Volt arası kararsız bölge olarak yada tanımsız olarak adlandırılır. Bilgisayar üzerinde veri iletimi ikilik sistemde yapılır iken dijital 1 değeri +5 Volt değerine, dijital 0 değeri 0 Volt değerine denk gelmektedir. RS-232 için belirlenen çalışma aralığının bilgisayar için uyumlu olmaması dolayısıyla TTL dönüştürücüler kullanılarak uyum sorunu aşılır. RS-232 iletim seviyelerini TTL seviyelerine çeviren elektronik kartlar için uyumlu entegreler olduğu gibi harici olarak bilgisayarın USB hattına bağlantı yapmak için kullanılan RS-232/USB çevirici aparatları da mevcuttur. Bu tarz aparatlar yoluyla veri bilgisayar ortamına taşınabilir hale gelir. TTL seviyeli sinyallerin kullanılması güç yönünden, ısı dağılımının az olmasından, veri transferine uygun olması yönlerinden çok avantajlıdır. Ancak TTL dışarıdan gelen sinyallerden çabuk etkilenir ve bu yüzden gerilim kayıplarına neden olur. Bu gibi durumlarda genel olarak TTL için belirlenen kararsız alan -3 Volt ve +3 Volt olduğundan kararsız alana düşmeler olabilir ve bu da başarılı veri iletimini engeller. Bu nedenle uzun mesafelerde tercih edilmez.

Seri haberleşme uzun mesafelerin sorun yaratması sorunu temel alınarak RS-232 dışında başka protokollerde geliştirilmiştir. RS-232 dışında en sık kullanılan seri haberleşme protokolü RS-485 protokolüdür. RS-485 standardı 1200 metreye kadar kablo uzunluğuna izin veren çok noktalı, Half Duplex seri iletişim standardıdır. Aslında temel olarak RS-232 standardının uzun mesafeler için kullanılması olarak düşünülebilir. İletişim hızı kablo uzunluğu ve türüne göre değişkenlik gösterir. RS-232 seri haberleşme protokollüne sahip bir haberleşmeyi RS-485 seri haberleşme protokolüne çevirmek mümkün olduğu gibi tersi de mümkündür. RS-232 standardının RS-485 standardına göre; uzun mesafelerde dezavantajı olmasının dışında RS-232 iki cihazın haberleşebileceği bir protokol iken RS-485 protokolü birkaç cihaz haberleşebilir. RS-485 standardı ilk olarak

Çizelge 3.2 RS-232 ile RS-485 Karşılaştırması

Özellik	RS-232	RS-485
Diferansiyel Yapı	YOK	VAR
Maksimum alıcı sayısı	1	32
Veri İletim Hat Tipi	Half Duplex/Full duplex	Half Duplex
Maksimum Mesafe	15 metre	1200 metre
Ağ topolojisi	Noktadan noktaya	Çok noktalı
Alıcı voltaj hassasiyeti	± 3 V	± 200 mV
Alıcı voltaj aralığı	± 15 V	-7..12 V
Sürücü maksimum çıkış voltajı	± 25 V	-7..12 V
Sürücü minimum çıkış voltajı (yük ile)	± 5 V	± 1.5 V

1983 yılında bir başka seri haberleşme protokolü RS-422 standardının gelişmiş düzeyi olarak çıkmıştır. RS-485 ile RS-232 arasındaki temel fark, RS-485 iki tel arasındaki değişken voltaj metodu ile veri transferi yapmasıdır. Bu yapıya diferansiyel yapı denir. Ayrıca düz kablo yerine burgulu kablo kullanımı ile gürültü akımının şiddeti azalır. Tablo 3.2 ile RS-232 ile RS-485 farklılıklarının bazıları verilmiştir.

Amerika’da 1970’li yıllarda üniversiteler arası veri iletişimde kullanılmak amacıyla ARPA adı verilen bir iletişim sistemi oluşturulmuştur. Bu sistemle farklı şehirlerdeki bilgisayarların birbirlerine bağlanabilmeleri ve veri iletişimi sağlamaları mümkün olmuştur. 1974’de bu iletişime standart getirilmiş ve adına TCP/IP protokolü denmiştir. TCP bilgisayarlar arasındaki veri iletişimini, paketler hâlinde ve kayıpsız olarak gerçekleştirilmesini yardımcı olan bir protokoldür. Aslında TCP protokolünün en önemli özelliği kimlik doğrulaması yapması ve veriyi alıcıya gönderirken veya alıcıdan veriyi alırken verinin bütünlüğünü sağlamasıdır. IP ise Türkçe karşılığı olarak ”İnternet Protokolü” demektir. IP veri için gerekli yönlendirme kurallarını belirlemektedir, verinin içeriği, uzunluğu gibi özellikleriyle ilgilenmeden sadece iletim adreslerini belirler. Bu yapıyı kargo taşımacılığındaki mantık üzerinden örneklersek kargolanacak eşyaların paketlenmesi, alıcıya ulaştıktan sonra eşyalara ulaşılabilir olması ve kargo şirketinin alıcıya ulaştı diye vericiyi bilgilendirmesi TCP özellikleri iken verici ve alıcı adresleri ile kargonun nereden nereye iletileceğinin belirlenmesi kısmına IP denilebilir.

TCP ve IP birleşerek TCP/IP protokol yapısını oluşturur. TCP/IP katmanlı bir yapıya sahiptir. Bu katmanlar; farklı sunucular üzerinde süreç ve uygulamalar arasında iletişimin sağlandığı uygulama katmanı, bir noktadan diğer bir noktaya veri akışını sağlayan taşıma katmanı, ağlar arasında verinin kaynaktan hedefe iletilmesini sağlayan internet katmanı, ağ

ile uç nokta arasındaki veri paketlerinin dijital 1 ve 0 verilerine dönüştürülerek taşındığı ağ erişim katmanıdır.

Elektronik cihazlar haberleşme için seri haberleşme kullanmakta ise haberleşme sınıfı olarak ikiye ayrılır. Bu sınıflardan biri verinin kullanıldığı, değerlendirildiği son kullanıcı cihazlarına denilen DTE (Data Terminal Equipment) sınıfını cihazlardır ve örnek olarak bilgisayar ve terminaller verilebilir. Diğer sınıf ise temel görevi veri aktarımına yardımcı olmak olan DCE (Data Communication Equipment) sınıfı cihazlardır ve örnek olarak modemler verilebilir.

Elektronik cihaz ve elektronik sistemlerde en çok kullanılan kavramlardan mikro işlemciler ve mikrodenetleyiciler kavramları çoğu zaman karıştırılan kavramlardır. Mikro işlemciler yapısında CPU (Central Process Unit) yani Türkçe karşılığı merkezi işlem birimi, ön bellek ve giriş/çıkış birimleri bulunan devrelere denir. Aslında kısaca mikro işlemciye bilgisayar denilebilir. Mikro işlemciler bağlı oldukları sistemin, mekanizmanın kontrolünü sağlarlar bu nedenle insan vücudundaki beyine benzerler. Bu neden mikro işlemciler için elektronik yapıların beyni demek yanlış olmaz. Mikrodenetleyiciler ise mikro işlemci ve mikrodenetleyici özel kılan diğer bileşenlerden oluşan bir mikro bilgisayardır. Aslında bir insanı ele alacak olursak insanın beynine mikro işlemci demek ne kadar yanlış değilse mikrodenetleyiciye de insanın kendisi demek o kadar yanlış olmaz. Çünkü mikrodenetleyiciler bir mikro işlemcinin, hafıza ve giriş - çıkışlar, kristal osilatör, zamanlayıcılar, seri ve analog giriş çıkışlar, programlanabilir hafıza gibi bileşenlerle tek bir tümleşik devre üzerinde üretilmiş halidir. Yani insan vücudunda bulunan göz bir algılayıcı ise bu mikrodenetleyicinin bir analog girişi denebilir ve bu gözden gelen verinin değerlendirmesi insan beyninde yani mikrodenetleyicinin mikro işlemcisinde gerçekleştirilir. İşte bu nedenle mikro işlemciler elektronik cihazların temelinde yer alan beyin yapısı iken mikro işlemcinin etrafında elektronik cihazın isterlerini karşılamak için bağlantı kurulmuş entegrelerden oluşan tümleşik devreye mikrodenetleyici denir. Mikro işlemciler sadece işlemi yönetir ve işleme karar verir ancak işlemi gerçekleştirmez mikrodenetleyiciler tek başına işlemi denetler ve gerçekleştirir. Mikro işlemci, karmaşık sistemleri kontrol etme amaçlı kullanılır ancak mikrodenetleyici daha basit devrelerde tercih edilir. Mikro işlemcilerin programlanması mikrodenetleyicilere göre daha karmaşıktır. Mikro işlemci fiyatı mikrodenetleyici fiyatına göre pahalıdır. Bu yüzden maliyet kriterlerine bakılarak basit sistemlerde mikrodenetleyiciler kullanılmaktadır. Ancak mikrodenetleyicilerin yetersiz kalabileceği daha karmaşık yerlerde mikro işlemciler kullanılır.

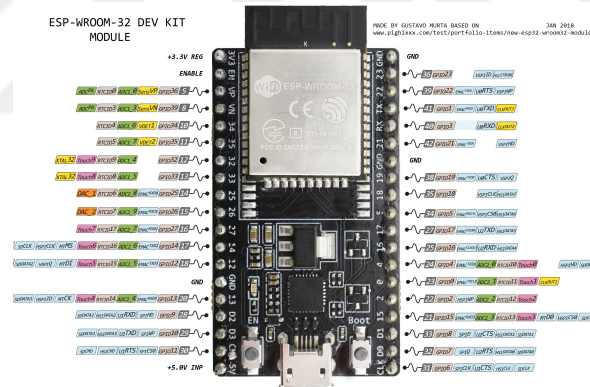
Mikrodenetleyici mikro işlemcisinin hangi işlemi yürüteceği belirlenmez ise hiçbir işe yaramayan bir plastik, metal ve temizlenmiş kum yığındır. Mikrodenetleyicinin

mikroişlemcisi tarafından ne yapılacağını belirleyen içeriğe yazılım denir ve mikrodenetleyici yazılım ile programlanarak ne işlev gerçekleştireceği belirlenir. Aslında programlama ya da diğer adıyla yazılım, bir cihazın, bir bilgisayarın ya da bir sistemin nasıl davranacağını anlatan, bu yapıya yön veren komutları, kelimeleri, aritmetik işlemleri içeren algoritmik bir yol haritasıdır. Programlama, bir programlama dilinde yapılır. Mikrodenetleyiciler aslen makine dili olan assembly dilinde programlanmıştır. Ancak C, Python ve JavaScript gibi makine dili olan assembly dilinine göre daha üst düzey programlama dilleriyle programlanabilmektedirler. Bilgisayar gibi son kullanıcıya hizmet eden daha üst seviye cihazlarda Java ve C# gibi daha üst seviye programlama dilleri kullanılarak kullanıcı arayüz yazılımları oluşturulabilmektedir. Herhangi bir sistemi akıllı hale getiren elektronik donanım ve yazılımdan oluşan entegre sisteme gömülü sistem denir. Gömülü sistemler bir ya da birden fazla mikrodenetleyici barındırabilir. Örneğin; buzdolabı gömülü bir sistemdir ve soğutucu motor kısmı ve iç aydınlatmayı sağlayan kısımlar farklı mikrodenetleyiciler olabilir. Genellikle bir elektronik cihaza ait gömülü kartın fonksiyonel olarak çalışır duruma getirilmesine olanak sağlayan yazılıma gömülü yazılım denir. Buzdolabının iç aydınlatmasını için hazırlanan gömülü karta buzdolabı kapağı açıldığında ışığı yanmasını sağlamak için gömülü yazılım oluşturup bu yazılımı gömülü karta programlamak gerekmektedir.

Gömülü yazılım oluşturmak ve gömülü kartları programlamak için birçok platform geliştirilmiştir. Bu tez çalışmasında gömülü yazılımı Arduino IDE platformunda geliştirilmiştir. Bu geliştirilen yazılım EceDuruDevice olarak adlandırılan bir elektronik cihaza ait yazılımdır. EceDuruDevice; Espressif Systems firmasının ürettiği ESP32 geliştirme kartından oluşmaktadır. Espressif Systems; Şanghai merkezli, nesnelerin interneti kapsamında çip tasarımı ve üreten firmadır ve düşük maliyetli Wi-Fi çözümleri üretmesi ile ün kazanmıştır. Wi-Fi, uzun adıya Wireless Fidelity cihazların kablosuz bir şekilde internete bağlanmasını sağlayan teknolojidir. Espressif Systems tarafından üretilen ve en çok kullanılan Wi-Fi modülü ESP8266 Wi-Fi modülüdür. Bu çalışmada kullanılan ESP32 geliştirme kartı ESP8266 Wi-Fi modülünü üstünde barındırmaktadır. Birçok mikrodenetleyici üzerinde seri iletişim birimi bulunmaktadır ve bu birimler vasıtasıyla seri iletişim yürütülür. Bu seri iletişim hatları asenkron olarak çalıştığı için herhangi bir “clock” ihtiyacı duymayan UART (Universal Asynchronous Receiver Transmitter) ve hem senkron hem de asenkron olarak çalışabilen USART (Universal Synchronous Asynchronous Receiver Transmitter) şeklindedir. USART, UART’a göre daha gelişmiş bir protokoldür. ESP32 geliştirme kartı ile seri haberleşme sağlaması için 3 UART hattı vardır. EceDuruDevice yazılımı UART hatları aracılığıyla seri haberleşmeyi kullanabilmektedir. Ayrıca EceDuruDevice ESP32 geliştirme kartında yer alan ESP8266 Wi-Fi modülü sayesinde kablosuz haberleşme de sağlayabilmektedir. ESP32 geliştirme kartına ait görsel Şekil 3.3 ile verilmiştir. ESP32 için teknik bilgiler Tablo 3.3 ile verilmiştir.

Çizelge 3.3 ESP32 Geliştirme Kartı Teknik Özellikler

Üretici	Espressif Systems
Flash	4 MB
RAM	520Kb
ROM	448 KB
Digital I/O Pini	25
PWM	16 kanal
ADC Kanal	18 (12bit)
DAC Kanal	2 (8 Bit)
GPIO (Kapasitif hassiyetli)	10
Analog Çıkış	2 kanal (8 bit)
Analog Giriş	6 Kanal
Haberleşme	3 UART, 3 SPI , I2C, Can Bus ve I2S Stereo
WIFI	802.11 b/g/n 2.4GHz



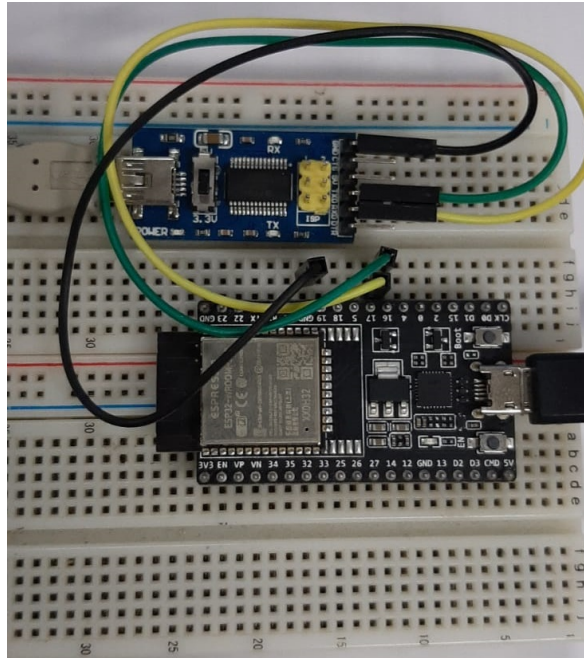
Şekil 3.3 ESP32 Geliştirme Kartı

Masaüstü arayüz uygulamaları özellikle elektronik cihaz teknolojisinin ürünlerinde son kullanıcının cihaz hakkındaki bilgi alması, cihaz özelliklerinin kullanımı ya da cihaz özelliklerinde değişiklik için sıklıkla kullanılır. Son kullanıcı dışında geliştiriciler için masaüstü arayüz uygulaması elektronik cihaz yazılımının bakım, test ve güncelleme için tercih edilir. Masaüstü uygulamalar yüksek seviyeli C++, C# ve Java gibi birçok programlama dilinde oluşturulabilmektedir. Bu programlama dillerini kullanarak masaüstü yazılım uygulamaların oluşturulabileceği birçok platform vardır. Bu tez çalışmasında hazırlanan EceDuruApp adı verilen masaüstü yazılım uygulaması Visual Studio platformunda C# diliyle oluşturulan bir WinForm uygulamasıdır.

Elektronik cihazlar ve bu cihazların seri haberleşmesinin bilgisayar arayüzünde kullanılması için gerilim seviyesinde uygun dönüştürmeler yapılması işlemleri TTL (Transistor-Transistor Logic) çevirimi işlemlerine örnektir. Bu çevirimi yapan apartlar TTL USB çevirici apartlarıdır. EceDuruDevice seri haberleşme pinlerinin bilgisayar ile bağlantısında bu apartlardan yararlanılmıştır. Ayrıca bilgisayarlarda seri bağlantı portları oluşturmak için gerçek bir fiziksel portun olması şart değildir. Fiziksel olarak var olmayan seri haberleşme portlarına Virtual Port bir diğer adıyla sanal port adı verilir. Bu tarz portlar bir yazılım vasıtasıyla oluşturulur. EceDuruApp uygulamasının testleri sırasında kullanılan sanal portlar Eltima LLC şirketinin sunduğu “Configure Virtual Serial Port Driver” uygulaması ile oluşturulmuştur.

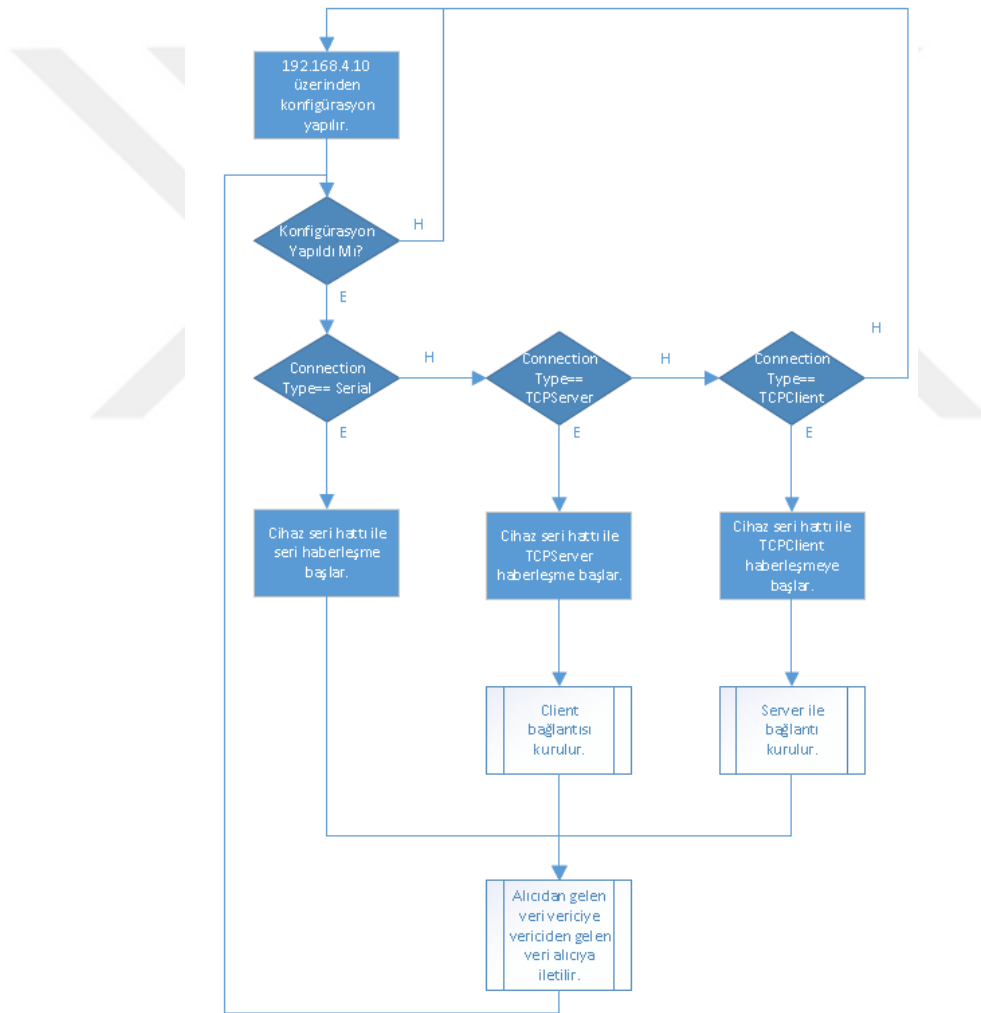
3.1 EceDuruDevice

Bu kısımda tez kapsamında hazırlanan EceDuruDevice hakkında genel tanıtım ve kullanım özelliklerini anlatmak amacıyla kullanım kılavuzu dokümanı hazırlanmıştır. Hazırlanan doküman EceDuruDevice V1.0.00 için hazırlanmış olup geliştirme ve iyileştirmeler sonrasında dokümanı etkileyen içerikler güncellenmelidir. EceDuruDevice genel özellikleri, kullanımı dışında fonksiyonel olarak çalışırılığı test edilmesi ve bu testin raporlanması planlanmıştır. EceDuruDevice kutu içi görünümü Şekil 3.4 ile verilmiştir.



Şekil 3.4 EceDuruDevice Kutu İçi Görünümü

EceDuruDevice yazılımı konfigürasyon ayarı gerektiren bir yazılımdır. Konfigürasyon ile çalışma tipi belirlenir ve konfigürasyon tamamlandığı durumu yazılıma iletilir. Eğer yazılım konfigürasyonu ayarlanmışsa yazılım çalışma moduna geçer. Çalışma modları; Serial, TCPServer ve TCPClient olarak üçe ayrılır. Çalışma modu ne olursa olsun EceDuruDevice yazılımının gerçekleştirdiği işlem aynıdır. EceDuruDevice yazılımı kurulan haberleşme üzerinden alıcının verisini vericiye vericinin verisini alıcıya iletilmesini yani köprülenmesini sağlar. Konfigürasyon değiştirilmesi için konfigürasyon sayfasının tekrar açılması yeterlidir. EceDuruDevice yazılımına ait akış diagramına ait görsel Şekil 3.5 ile verilmiştir. Kaynak kodlar tez ile CD içerisinde teslim edilecektir.



Şekil 3.5 EceDuruDevice Akış Diagram Görseli

GÖRKEM OCAK YÜKSEK LİSANS PROJESİ

ECEDURUDEVICE KULLANIM KILAVUZU DOKÜMANI

Doküman Kodu : EceDuruDevice.KKD.01
Revizyon : 1.00
Tarih : 04.06.2021

DOKÜMAN REVİZYON SAYFASI

Kod: EceDuruDevice.KKD.01

REV. NO	TARİH	SAYFA NO.	AÇIKLAMA
1.00	04.06.2021	Tümü	İlk basım

ECEDURUDEVICE KULLANIM KILAVUZU DOKÜMANI

İÇİNDEKİLER

İÇİNDEKİLER	iii
1. GENEL	4
2. KAPSAM	4
3. AÇIKLAMA	4
3.1 GENEL TANIMLAMA	4
3.2 KULLANIM ÖZELLİKLERİ	5
3.2.1 SERİ HABERLEŞME İLE SERİ HABERLEŞME ARASI KULLANIM	6
3.2.2 SERİ HABERLEŞME İLE TCP SERVER HABERLEŞME ARASI KULLANIM...	7
3.2.3 SERİ HABERLEŞME İLE TCP CLİENT HABERLEŞME ARASI KULLANIM.....	7
3.3 KULLANIM ALANLARI	8

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruDevice.KKD.01
 REV. NO : 1.00
 REV. TARİH : 04.06.2021

ECEDURUDEVICE KULLANIM KILAVUZU DOKÜMANI

1. GENEL

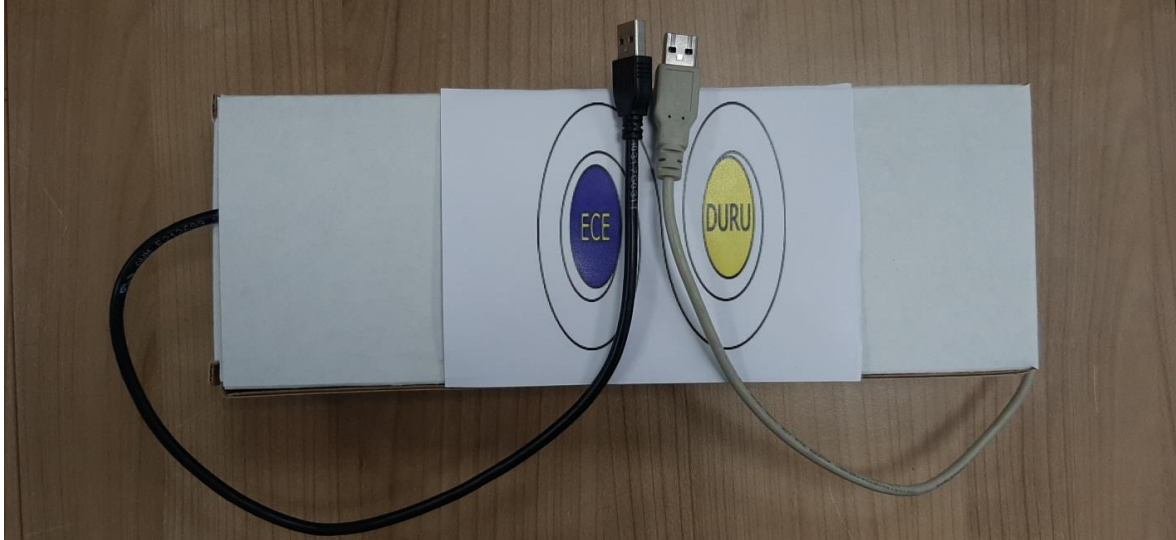
Bu doküman Eskişehir Osmangazi Üniversitesi Bilgisayar Mühendisliği Yüksek Lisans öğrencisi Görkem OCAK tarafından tez çıktısı EceDuruDevice ile ilgili genel tanımlama ve kullanım bilgisini içermektedir.

2. KAPSAM

Bu doküman Kullanım Kılavuzu kapsamında, Genel Tanımlama, Kullanım Özellikleri ve Kullanım Alanları ile ilgili detayları içermektedir.

3. AÇIKLAMA

EceDuruDevice V1.0.00 yazılımı ile tez içeriğinde bahsedilen veri haberleşmesinde köprü ve haberleşme teknolojileri arasında dönüştürücü özellikleri gerçekleştirilebilmektedir. EceDuruDevice seri haberleşmeyi iki adet USB erkek konnektör opsiyonu ile sağlanmaktadır. Cihaz seri bağlantısını USB opsiyonu olan cihazlarla yapabilmektedir ve bağlanılan cihazda seri bağlantıya ait port oluşmaktadır. EceDuruDevice V1.0.00' a ait görsel Şekil 1 ile verilmiştir.



Şekil 1 EceDuruDevice Cihaz Görünümü

3.1 GENEL TANIMLAMA

Şekil 1 ile verilen cihaz görünümünde yer alan siyah USB kablosu serialCon1 ve gri USB kablosu serialCon2 olarak adlandırılmıştır. Siyah USB kablosu cihazın güç hattının da barındırdığı için seri haberleşmeden yapılacak dönüştürme işlemleri de dahil olmak üzere tüm özellikler bu hattan yürütülür.

EceDuruDevice genel olarak seri haberleşme ile seri, TCP Server yada TCP Client haberleşmeleri arasında köprü ve dönüştürme işlemine yarayan elektronik haberleşme ekipmanıdır. Aynı zamanda kablolu seri haberleşme ile başlayan haberleşmeyi Wi-Fi ile kablolu TCP Server yada TCP Client haberleşmesine dönüştürebilme özelliğine de sahiptir.

ECE DURU DEVICE KULLANIM KILAVUZU DOKÜMANI

3.2 KULLANIM ÖZELLİKLERİ

EceDuruDevice ile 3 tip kullanım opsiyonu vardır;

1. Seri Haberleşme ile Seri Haberleşme Arası Kullanım
2. Seri Haberleşme ile TCP Server Haberleşme Arası Kullanım
3. Seri Haberleşme ile TCP Client Haberleşme Arası Kullanım



Şekil 2. EceDuruDevice Kullanım Tipleri

EceDuruDevice için kullanım tiplerinin anlatımını detaylandırmak için Şekil 2 görseli verilmiştir. Görsel de yer alan DEVICE-1 ve DEVICE-2 haberleşme kurması gereken iki cihazı temsil etmektedir. DEVICE-1 ile EceDuruDevice arası haberleşme tipine A ve DEVICE-2 ile EceDuruDevice arası haberleşme tipine B dersek;

- A ya da B haberleşmelerinden biri mutlaka seri haberleşme olmak zorundadır.
- A ya da B haberleşmelerinden seri haberleşme olarak seçilen dışındaki seri, TCP Server ya da TCP Client olarak bağlantı kurabilir.

Bu kullanım tiplerinin hangisinin kullanılacağı “Konfigürasyon Sayfası” ile belirlenir. Bu sayfaya ait görsel Şekil 3 ile verilmiştir. “Konfigürasyon Sayfası” kısmına erişmek için EceDuruDevice enerjili olması gerekmektedir. EceDuruDevice enerjili durumda iken “EceDuruDevice” adı verilen Wi-Fi erişim noktasına “12345678” şifresi ile bağlantı yapılır. Bağlantı yapan cihaz(bilgisayar, telefon vb.) ile tarayıcı ile 192.168.4.10 adresine gidilir ve “Konfigürasyon Sayfası” kısmına erişilir.

ECE DURU DEVICE

DEVICE CONFIGURATION

CONVERT CONFIGURATION

CONNECTION CONVERT :

WIFI CONFIGURATION STATUS: READY

SSID: PASSWORD:

Ethernet CONFIGURATION

IP: PORT:

CONFIGURATION STATUS: FALSE

Şekil 3 EceDuruDevice Konfigürasyon Sayfası

ECE DURU DEVICE KULLANIM KILAVUZU DOKÜMANI

“Konfigürasyon Sayfası” ile kullanıcı hangi kullanım tipini seçebilmektedir. Kullanım tipini seçmek için;

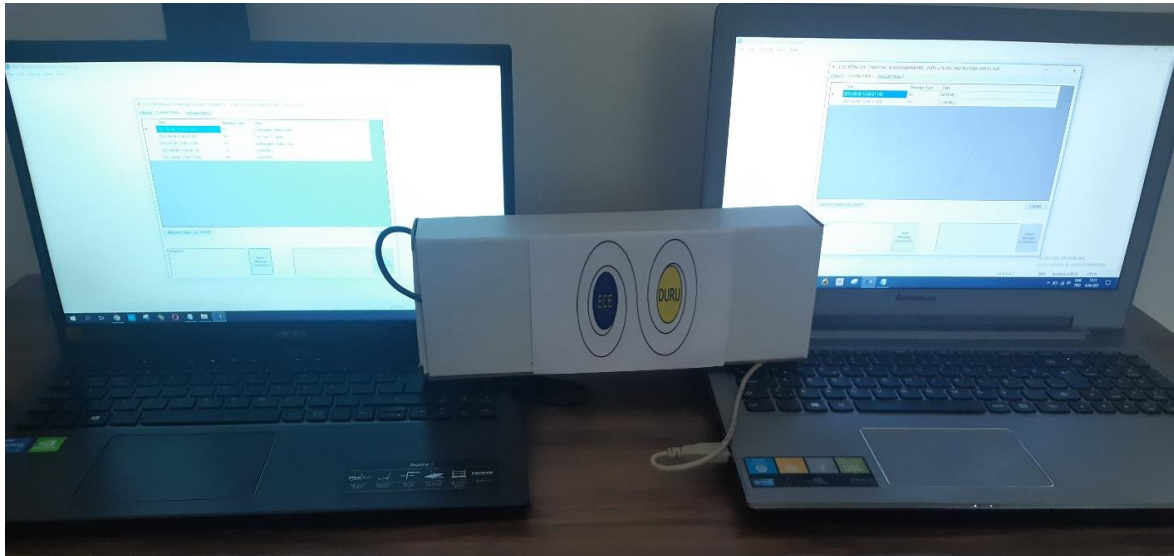
1. “Konfigürasyon Sayfası” üzerinde yer alan “CONNECTION CONVERT” kısmında tip seçimi (Serial, TCP Server yada TCP Client) olarak seçilir ve “SET CON1” butonuna basılır.
2. Seçim türüne göre “192.168.4.10/ConSelect1Serial”, “192.168.4.10/ConSelect1TCPServer” yada “192.168.4.10/ConSelect1TCPClient” adreslerinden birine gidilir. Bu adrese geçiş yapılması sonrası “SAVE CONFIGURATION” butonuna basılır.
3. “SAVE CONFIGURATION” butonuna basılması sonrası “192.168.4.10/saveConfig” sayfasına geçilir ve “CONFIGURATION STATUS: TRUE” olduğu gözlemlenir.
4. Bu adıma kadar gelinmesi durumunda seri haberleşme ile seçilen haberleşme türü arasında köprü ve dönüşüm için gerekli ayarlamalar tamamlanmıştır.

EceDuruDevice için kullanılan haberleşme ayarlarına dair notlar şu şekildedir;

- Seri haberleşme opsiyonlarının baudrate ayarları 115200 olarak sabittir.
- TCP Server opsiyonuna ait IP değeri 192.168.4.10 ve port değeri 4001 olarak sabittir. Bu değerlerden client beklenir.
- TCP Client opsiyonuna ait IP değeri 192.168.4.11 ve port değeri 4001 olarak sabittir. Bu değerlere sahip server ile bağlantı kurulur.

3.2.1 SERİ HABERLEŞME İLE SERİ HABERLEŞME ARASI KULLANIM

EceDuruDevice ile seri haberleşmeyi seri haberleşme ile köprülemek için kullanılan özelliktir. Bu işlem öncesi serialCon1 ve serialCon2 ile cihaz bağlantıları yapılması gerekmektedir. Cihaz bağlantılarının hazırlanması sonrası cihaz konfigürasyonuna seri haberleşme ile seri haberleşme arası kullanıma uygun hale getirilmesi gerekmektedir. Bu ayarlamaların tamamlanması ile bağlanılan iki seri haberleşme hattı doğrudan birbirlerine bağlanmış olur. Örnek kullanım için hazırlanan test ortamına dair görsel Şekil-3 ile verilmiştir. Görseldeki iki cihazda seri haberleşme kullanılmaktadır.



Şekil 4 EceDuruDevice Seri ile Seri Örneği

ECE DURU DEVICE KULLANIM KILAVUZU DOKÜMANI

3.2.2 SERİ HABERLEŞME İLE TCP SERVER HABERLEŞME ARASI KULLANIM

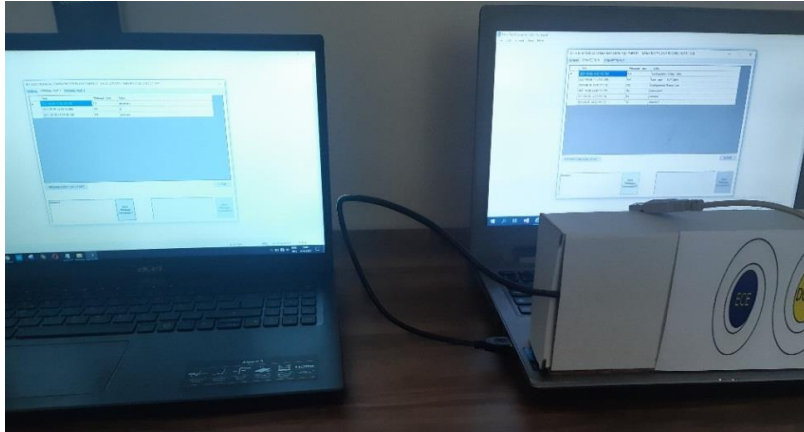
EceDuruDevice ile seri haberleşmeyi TCP Server haberleşmesine dönüştürerek bu haberleşmeler arası köprü oluşturmak için kullanılan özelliktir. Bu işlem öncesi serialCon1 ile cihaz bağlantıları yapılması gerekmektedir. Cihaz bağlantılarının hazırlanması sonrası cihaz konfigürasyonuna seri haberleşme ile TCP Server haberleşme arası kullanıma uygun hale getirilmesi gerekmektedir. Bu ayarlamaların tamamlanması ile bağlanılan iki seri haberleşme hattı doğrudan IP değeri 192.168.4.10 ve port değeri 4001 olan TCP Server haberleşme hattına bağlanmış olur. Örnek kullanım için hazırlanan test ortamına dair görsel Şekil-4 ile verilmiştir. Görselde soldaki cihaz seri sağdaki cihaz TCP haberleşmesine sahiptir.



Şekil 5 EceDuruDevice Seri ile TCP Server Örneği

3.2.3 SERİ HABERLEŞME İLE TCP CLİENT HABERLEŞME ARASI KULLANIM

EceDuruDevice ile seri haberleşmeyi TCP Client haberleşmesine dönüştürerek bu haberleşmeler arası köprü oluşturmak için kullanılan özelliktir. Bu işlem öncesi serialCon1 ile cihaz bağlantıları yapılması ve TCP Client haberleşmesinin bağlanacağı TCP Server yapısının hazır olması gerekmektedir. TCP Client haberleşmesinin bağlanacağı TCP Server haberleşmesine ait IP değeri 192.168.4.11 ve port değeri 4001 şeklindedir. Cihaz bağlantılarının hazırlanması sonrası cihaz konfigürasyonuna seri haberleşme ile TCP Client haberleşme arası kullanıma uygun hale getirilmesi gerekmektedir. Bu ayarlamaların tamamlanması ile bağlanılan iki seri haberleşme hattı doğrudan IP değeri 192.168.4.11 ve port değeri 4001 olan TCP Client haberleşme hattına bağlanmış olur. Örnek kullanım için hazırlanan test ortamına dair görsel Şekil-5 ile verilmiştir. Görselde sağdaki cihaz seri soldaki cihaz TCP haberleşmesine sahiptir.



Şekil 6 EceDuruDevice Seri ile TCP Client Örneği

ECEURUDEVICE KULLANIM KILAVUZU DOKÜMANI

3.3 KULLANIM ALANLARI

EceDuruDevice kullanım tipine göre çeşitli alanlarda kullanılabilir.

Seri haberleşme ile seri haberleşme arasında kullanım tercih edilirse;

- Seri haberleşme hattının veri kaybı olmadan kablo boyunun uzatılmasında,
- Seri haberleşme yapan cihaza ait simülatör cihazların oluşturulmasında,
- Seri haberleşme yapan cihaz verilerinin bir ortamda toplanmasında,
- Seri haberleşme verilerinin incelenerek tersine mühendislik alanlarında kullanılabilir.

İki bilgisayarın seri haberleşme kurmasını sağlamak için gerekli fiziksel altyapıya sahiptir.

Seri haberleşme ile TCP Server haberleşme arasında kullanım tercih edilirse;

- Seri haberleşmeye sahip cihazın TCP Client haberleşmeye sahip bir yada birden fazla cihaza bağlanmasında,
- Seri haberleşmeye sahip cihazın TCP Client olarak simülatör cihazları oluşturulmasında,
- TCP Client haberleşme yapan cihaz verilerinin toplanmasında,
- Seri haberleşme yapan cihaz verilerinin toplanmasında,
- Kablolu bir cihazı kablosuz bir cihaza bağlanmasında kullanılabilir.

İki bilgisayardan birine kablolü diğerine kablosuz ve master olarak bağlanmayı sağlayacak fiziksel altyapıya sahiptir.

Seri haberleşme ile TCP Client haberleşme arasında kullanım tercih edilirse;

- Seri haberleşmeye sahip cihazın TCP Server haberleşmesine sahip bir cihaza bağlanmasında,
- Seri haberleşmeye sahip cihazın TCP Server olarak simülatör cihazları oluşturulmasında,
- TCP Server haberleşme yapan cihaz verilerinin toplanmasında,
- Seri haberleşme yapan cihaz verilerinin toplanmasında,
- Kablolu bir cihazı kablosuz bir cihaza bağlanmasında kullanılabilir.

İki bilgisayardan birine kablolü diğerine kablosuz ve slave olarak bağlanmayı sağlayacak fiziksel altyapıya sahiptir.

EceDuruDevice barındırdığı bu özelliklerinin dışında geliştirilmeye açık olan yönüyle ihtiyaca uygun evrilmeye uygun bir cihazdır.

3.2 EceDuruApp

Bu kısımda tez kapsamında hazırlanan EceDuruApp hakkında genel tanıtım ve kullanım özelliklerini anlatmak amacıyla kullanım kılavuzu dokümanı hazırlanmıştır. Hazırlanan doküman EceDuruApp V1.0.00 için hazırlanmış olup geliştirme ve iyileştirmeler sonrasında dokümanı etkileyen içerikler güncellenmelidir. EceDuruApp genel özellikleri, kullanımı dışında fonksiyonel olarak çalışırılığı test edilmesi ve bu testin raporlanması planlanmıştır.

Kaynak kodlar tez ile CD içerisinde teslim edilecektir.



GÖRKEM OCAK YÜKSEK LİSANS PROJESİ

ECEDURUAPP KULLANIM KILAVUZU DOKÜMANI

Doküman Kodu : EceDuruApp.KKD.01
Revizyon : 1.00
Tarih : 04.06.2021

DOKÜMAN REVİZYON SAYFASI

Kod: EceDuruApp.KKD.01

REV. NO	TARİH	SAYFA NO.	AÇIKLAMA
1.00	04.06.2021	Tümü	İlk basım

ECEDURUAPP KULLANIM KILAVUZU DOKÜMANI

İÇİNDEKİLER

İÇİNDEKİLER	iii
1. GENEL	4
2. KAPSAM	4
3. AÇIKLAMA	4
3.1 GENEL TANIMLAMA	4
3.2 KULLANIM ÖZELLİKLERİ	4
3.2.1 CONFIG	4
3.2.2 CONNECTION 1.....	6
3.2.3 CONNECTION 2.....	7
3.3 KULLANIM ALANLARI	8

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.KKD.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP KULLANIM KILAVUZU DOKÜMANI

1. GENEL

Bu doküman Eskişehir Osmangazi Üniversitesi Bilgisayar Mühendisliği Yüksek Lisans öğrencisi Görkem OCAK tarafından tez çıktısı EceDuruApp ile ilgili genel tanımlama ve kullanım bilgisini içermektedir.

2. KAPSAM

Bu doküman Kullanım Kılavuzu kapsamında, Genel Tanımlama, Kullanım Özellikleri ve Kullanım Alanları ile ilgili detayları içermektedir.

3. AÇIKLAMA

EceDuruApp V1.0.00 yazılımı ile tez içeriğinde bahsedilen veri haberleşmesinde köprü, haberleşme teknolojileri arasında dönüştürücü, veri haberleşmesine veri dahil edebilme ve veri kayıt özellikleri gerçekleştirilebilmektedir.

3.1 GENEL TANIMLAMA

EceDuruApp genel olarak seri, , TCP Server yada TCP Client haberleşmesi ile seri, TCP Server yada TCP Client haberleşmeleri arasında köprü ve dönüştürme işlemlerine yarayan ve haberleşme hattına veri dahil edebilmenin yanısıra veri kayıt edebilme özelliklerine sahip masaüstü elektronik haberleşme veri hizmet aracı uygulamasıdır.

3.2 KULLANIM ÖZELLİKLERİ

EceDuruDevice ile 3 sayfadan oluşmaktadır;

1. CONFIG
2. CONNECTION 1
3. CONNECTION 2

3.2.1 CONFIG

EceDuruApp yazılımının çalışma senaryosuna karar verilen sayfadır. Kullanıcı bu sayfa üzerinden kurabileceği iki bağlantı noktası için tercihlerde bulunmaktadır. "Connection 1" ve "Connection 2" olarak verilen bu iki opsiyonun konfigürasyon ayarlama yapısı aynıdır. EceDuruApp üzerinden kullanıcı "Serial", "TCP Server" yada "TCP Client" haberleşme altyapılarını tercih edebilir. Bu haberleşme yapılarının tercih edilmesine yönelik adımlar şöyle sıralabilir;

1. EceDuruApp üzerinde bulunan "CONFIG" sayfasını açın.
2. Bağlantı yapmak istediğiniz bağlantı alanı belirleyin. (Connection 1 yada Connection 2)
3. Madde-2 ile belirlenen alanda yer alan "Connection Type" alanında yapmak istediğiniz bağlantı tipini seçin ve "SELECT" butonuna basınız.
4. Eğer Madde-3 ile seçtiğiniz bağlantı tipi "Serial" ise bağlantı yapmak istediğiniz COM port bilgisi ve BAUDRATE değerini giriniz. (Örneğin; COM: COM6 ve BAUDRATE:115200)
5. Eğer Madde-3 ile seçtiğiniz bağlantı tipi "TCP Server" yada "TCP Client" ise IP ve PORT alanlarına bağlantı yapmak istediğiniz değerleri girin. (Örneğin; IP: 192.168.4.10 ve PORT: 4001)
6. Bağlantı ayarları için değerleri girme işlemi sonrasında "CONNECT" butonu ile bağlantı kurun.

EceDuruApp Connection 1 ile bağlı olduğu cihazdan alınan veriyi Connection 2 ile bağlı olan cihaza ve Connection 2 ile bağlı olduğu cihazdan alınan veriyi Connection 1 ile bağlı olan cihaza otomatik aktarımına ECHO denir.

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.KKD.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP KULLANIM KILAVUZU DOKÜMANI

Kurulan iki farklı bağlantının veri haberleşmesini köprülemek için “ECHO” seçeneği aktif edilmelidir. Köprülenen haberleşmenin veri kayıt işlemini başlatmak için “RECORD ECHO LOG START” ve kayıt işlemini bitirmek için “RECORD ECHO LOG STOP” butonları kullanılır. “CONFIG” sayfasına ait örnek görsel Şekil 1 ile verilmiştir.

The screenshot shows the CONFIG window of EceDuruApp. It has two tabs: CONFIG, CONNECTION 1, and CONNECTION 2. The CONFIG tab is active. Under CONNECTION 1, the Connection Type is set to 'Serial-1', COM is 'COM21', and BAUDRATE is '115200'. There are 'DESELECT' and 'DISCONNECT' buttons. Under CONNECTION 2, the Connection Type is set to 'TCP Server-2', IP is '192.168.4.11', and PORT is '4001'. There are 'DESELECT' and 'DISCONNECT' buttons. At the bottom, the 'ECHO' checkbox is checked, and the 'RECORD ECHO LOG START' button is visible.

Şekil 1 EceDuruApp CONFIG Sayfası (Örnektir)

Veri kayıt işlemi uygulamanın yer aldığı ana dosya dizinin içinde yer alan “Logs” klasörü içerisinde “Echo_KayıtYılAyGün_KayıtSaatDakikaSaniye.txt” formatında tutulur. Örnek kayıt dosya ismi; Echo_210603_093524.txt şeklindedir. Veri kaydının tutulduğu “.txt” formatındaki dosya içeriğinde kayıt içeriği veri işlem zaman bilgisi, veri işlem türü ve verinin ASCII formattaki karşılığı yer almaktadır. Connection 1 için tutulan örnek veri kayıt dosyasının içeriği Şekil 2 ile verilmiştir.

The screenshot shows a Notepad window titled 'Echo_210603_093524.txt - Notepad'. The content of the file is as follows:

```

2021-06-03 09:35:35.660 CON2TX: H1;H2
2021-06-03 09:35:35.660 CON1RX: H1;H2
2021-06-03 09:35:40.066 CON1TX: H2;H1
2021-06-03 09:35:40.066 CON2RX: H2;H1

```

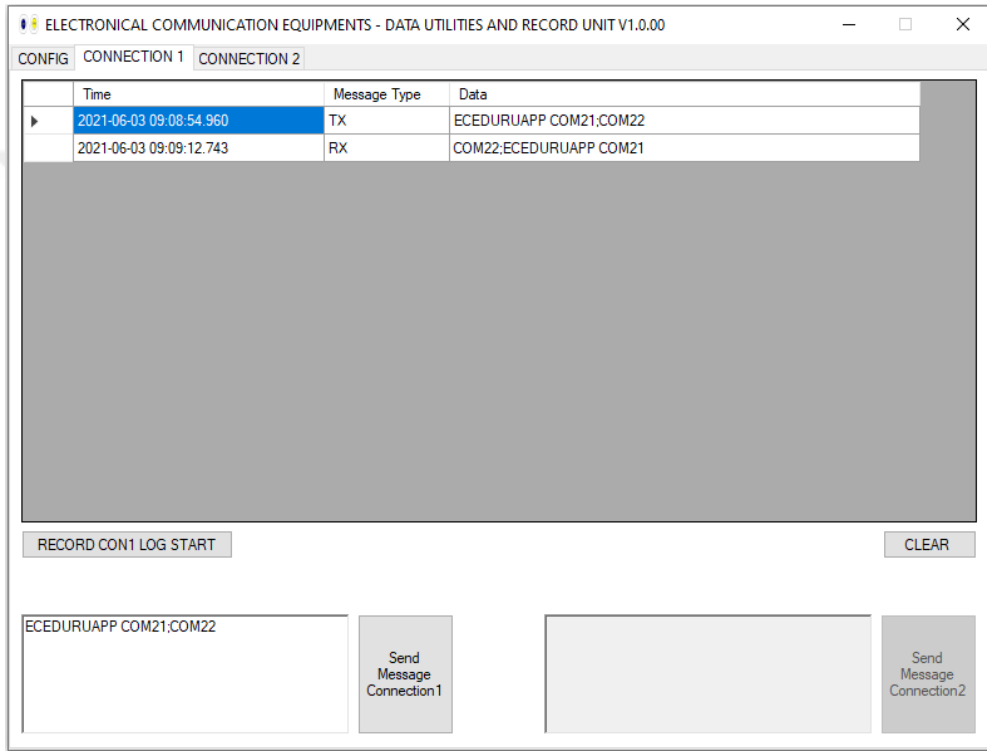
The status bar at the bottom indicates 'Ln 1, Col 1', '100%', 'Windows (CRLF)', and 'UTF-8'.

Şekil 2 EceDuruApp Echo Örnek Veri Kayıt Dosya İçeriği

ECEDURUAPP KULLANIM KILAVUZU DOKÜMANI

3.2.2 CONNECTION 1

EceDuruApp ile “CONFIG” sayfası üzerinden “Connection 1” kısmı için gerekli bağlantı ayarlarının yapılması sonra kurulan bağlantıya ait veri gönderim, veri alım ve veri kayıt işlemlerinin yapıldığı sayfadır. Bu sayfaya ait örnek görsel Şekil 3 ile verilmiştir. Bu sayfada haberleşme verilerine ait veri işlem zaman bilgisi, veri işlem türü ve verinin ASCII formattaki karşılığı veri listesinde yer almaktadır. “RECORD CON1 LOG START” butonu ile veri haberleşmesine ait kayıt işlemi başlatılırken “RECORD CON1 LOG STOP” butonu ile veri haberleşmesine ait kayıt işlemi sonlandırılır.

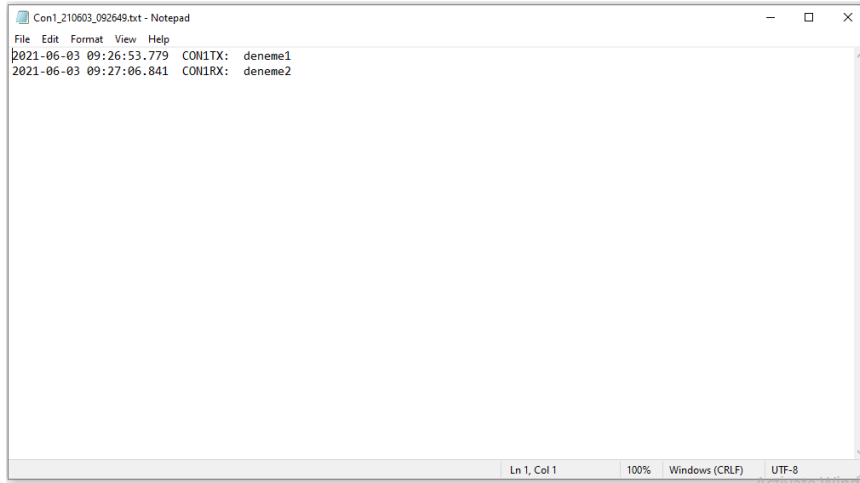


Şekil 3 EceDuruApp CONNECTION 1 Sayfası (Örnektir)

Veri kayıt işlemi uygulamanın yer aldığı ana dosya dizinin içinde yer alan “Logs” klasörü içerisinde “BağlantıOpsiyonu_KayıtYılıAyGün_KayıtSaatDakikaSaniye.txt” formatında tutulur. Örnek kayıt dosya ismi; Con1_210603_092649.txt şeklindedir.

Veri kaydının tutulduğu “.txt” formatındaki dosya içeriğinde kayıt içeriği veri işlem zaman bilgisi, veri işlem türü ve verinin ASCII formattaki karşılığı yer almaktadır. Connection 1 için tutulan örnek veri kayıt dosyasının içeriği Şekil 4 ile verilmiştir.

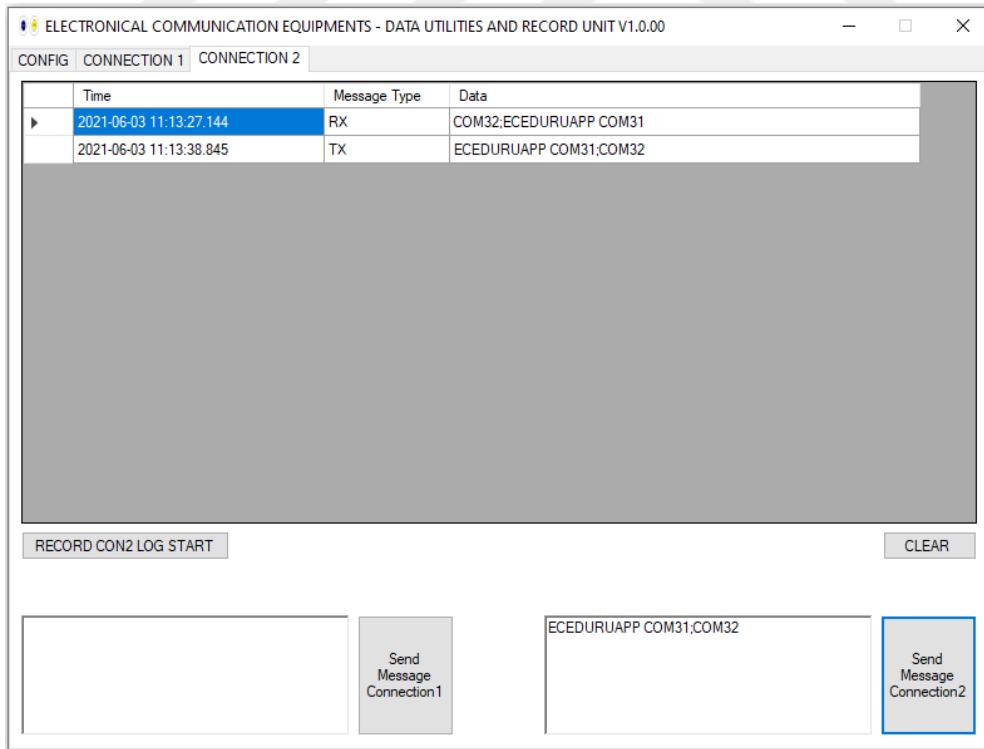
ECEDURUAPP KULLANIM KILAVUZU DOKÜMANI



Şekil 4 EceDuruApp Connection 1 Örnek Veri Kayıt Dosya İçeriği

3.2.3 CONNECTION 2

EceDuruApp ile “CONFIG” sayfası üzerinden “Connection 2” kısmı için gerekli bağlantı ayarlarının yapılması sonra kurulan bağlatıya ait veri gönderim, veri alım ve veri kayıt işlemlerinin yapıldığı sayfadır. Connection 1 için anlatılan tüm özellikler Connection 2 için de aynıdır. Örnek kayıt dosya ismi; Con2_210603_092649.txt şeklindedir. Bu sayfaya ait örnek görsel Şekil 5 ile verilmiştir.



Şekil 5 EceDuruApp CONNECTION 2 Sayfası (Örnektir)

Connection 1 ile olan bağlantı yönüne mesaj gönderimi “Send Message Connection 1” butonu ile sağlanırken Connection 2 ile olan bağlantı yönüne mesaj gönderimi “Send Message Connection 2” butonu ile sağlanır. Connection 1 ve Connection 2 yönüne mesaj iletimi sadece bu bağlantılar aktif ise gerçekleştirilir.

ECEDURUAPP KULLANIM KILAVUZU DOKÜMANI**3.3 KULLANIM ALANLARI**

EceDuruApp kullanım tipine göre çeşitli alanlarda kullanılabilir. EceDuruApp;

- Seri haberleşme kullanan cihazlarda terminal, simülatör ve test görevi görmek için,
- TCP Server haberleşme kullanan cihazlarda terminal, simülatör, veri kayıt ve test görevi görmek için,
- TCP Client haberleşme kullanan cihazlarda terminal, simülatör, veri kayıt ve test görevi görmek için,
- Seri haberleşmeyi TCP Server haberleşmesine ve TCP Server haberleşmesine seri haberleşme yapılarına dönüştürme, köprü olma, veri dahil edebilme ve veri kayıt işlemleri için,
- Seri haberleşmeyi TCP Client haberleşmesine ve TCP Client haberleşmesine seri haberleşme yapılarına dönüştürme, köprü olma, veri dahil edebilme ve veri kayıt işlemleri için,
- TCP Server haberleşmeyi TCP Client haberleşmesine ve TCP Client haberleşmesine TCP Server haberleşme yapılarına dönüştürme, köprü olma, veri dahil edebilme ve veri kayıt işlemleri için kullanılabilir.

EceDuruApp ile birbirleri ile haberleşme arayüzü olarak haberleşmesi mümkün ancak haberleşme teknolojisi olarak haberleşmesi mümkün olmayan cihazların bağlantılarının kurulmasına olanak verir.

EceDuruApp gelişmeye açık ve son kullanıcı gereksinimlerini karşılamak için evrilebilecek bir altyapıya sahip bir masaüstü yazılım uygulamasıdır.

4. BULGULAR VE TARTIŞMA

Tez çalışması sırasında yapılan literatür taramasında tez çalışması ile hazırlanmış olan EceDuruDevice ve EceDuruApp yazılımları için özgünlük araştırması yapılması da sağlanmıştır. Literatür araştırmasında EceDuruDevice gibi iki cihaz arası veri haberleşmesi yapan cihazlara rastlanmasına karşın bu cihazlar iki cihaz bağlantısı içinde ya tek tip elektronik haberleşme ya da iki farklı ancak statik haberleşme teknolojisi tercih edilmiştir. Herhangi bir cihazda EceDuruDevice tarafından sağlanan kullanıcı tercihli elektronik haberleşme teknolojileri arasında dönüşüm kullanıldığına rastlanılmamıştır. Endüstriyel pazarda ise elektronik haberleşmeleri genellikle USB hatına çıkılmasına yönelik cihazlar tasarlanmıştır. Elektronik haberleşme teknolojilerinde dönüştürücü cihaz üretmek yerine dönüştürücü entegre tasarımı tercih edilmiştir. Bu tercih sayesinde kart seviyesinde çözümler sunulmuştur. Bu tez çalışmasında genel amaçlı veri haberleşme teknolojileri ve bu teknolojilerin yazılımsal bir çalışma ile örneklendirilmesi amacıyla oluşturulan EceDuruDevice ile piyasada ve literatürde yer alan özel amaçla üretilmiş cihazlardan farklıdır. Diğer cihazlardan özgün kılan bir diğer unsur ise oluşturulma amacıdır. EceDuruDevice ticari bir ürün oluşturmak amacı taşımayan kişisel bilgi, tecrübe ve edinimlerden tez çalışması için ortaya çıkarılmış bir cihazdır. Aynı şekilde EceDuruApp yazılımı literatürde ve piyasada yer alan birçok haberleşme terminal yazılımından bazı özellikleri ile ayrılmaktadır. EceDuruApp tarafından sağlanan özelliklerin özgünlüğü tartışılabilir. Çünkü EceDuruApp tarafından sağlanan özelliklerin her biri ücretli ya da ücretsiz birçok uygulamada yer alabilmektedir. Ancak ticari bir amaç barındırmadan kişisel bilgi, tecrübe ve edinimlerden tez çalışması için ortaya çıkarılmış bu yazılım uygulaması birçok özelliği aynı anda tek bir yazılım uygulamasında toplanmasını sağlamıştır. Özellikle özgün bir özellik olarak yer alan haberleşme teknolojilerinde dönüşüm yapmayı sağlama özelliğinin başka yazılım uygulamaları ile sağlanması mümkün olsa da bu uygulamaların veri haberleşmesine veri dahil edebilme, veri haberleşme kaydı oluşturma ya da veri haberleşmesini görüntüleyebilme özelliklerinden bir arada yapamaması nedeniyle EceDuruApp yazılımından farklıdır.

Tez ile hazırlanmış olan EceDuruDevice ve EceDuruApp genel özelliklerini gerçekleştirebildiğine dair fonksiyonel olarak doğrulanması amacıyla test prosedürleri hazırlanmıştır. Hazırlanan test prosedürü dışındaki senaryolarda EceDuruDevice ve EceDuruApp yazılımlarına dair eksik olan ve iyileştirme gerektiren kısımlar belirlenmiştir. Bu iyileştirmelerden bazıları tartışmaya açık konulardır. Örneğin; EceDuruDevice şu an erişim noktası (access point) olarak çalışmaktadır. Bu özellikle cihazın kullanılabilmesi için

cihaza bağlanması zorunluluğu taşımaktadır. Bu özellik yerine ortak erişim noktasının değiştirilebilir olması özelliği eklenebilir. Ancak bu durumda cihaz için belirlenen sabit IP değerinin ortak ağda yer alan herhangi bir cihaz da yer almaması zorunluluğu gerektirmektedir. Aynı IP değeri ile bir ortak ağda yer almak mümkün değildir, bu duruma IP çakışması denir. Bu tarz durumlarda daha önce IP değerini alan cihaz etkilenmez iken alınmış bu IP değerini almak isteyen cihaz da IP çakışması hatası oluşur ve ağa dahil olamaz. Bu gibi sorunların önüne geçmek için özelleştirilmiş bir ağ olan cihaza ait erişim noktası olması avantaj iken her cihazın bu ağa bağlanması mümkün olmayacağı için dezavantajdır. Örneğin; TCP Server olarak haberleşme altyapılı bir elektronik cihazın erişim noktası sabit IP değerli bir internet adresi ise EceDuruDevice ile bu cihaza erişim için Wi-Fi ile internete çıkılmalı, daha sonra cihaz ile ortak alt ağ dizini ayarlanmalı ve cihaza bağlanmalıdır. Bu işlemlerin gerçekleştirilmesi sırasında IP çakışmasına neden olmayacak bir IP adresi seçilmelidir. Aksi halde cihaz ile bağlantı kurulamaz.

EceDuruApp yazılımında EceDuruDevice yazılımında karşılaşılan zorluklar yoktur. EceDuruApp yazılımı hem masaüstü bir uygulama olmasından hemde daha üst seviye bir programlama dili ile yazılmış olmasından dolayı EceDuruDevice yazılımına göre daha kolay oluşturulmuştur. Ancak EceDuruApp yazılımı EceDuruDevice yazılımından daha fazla özellik barındırdığı için yazılım karmaşıklığı daha fazladır. EceDuruDevice için bir bağlantı her zaman seri olmak zorundadır. Bu nedenle seri ile seri, TCP Server ve TCP Client arası köprü ve dönüşüm gerçekleştirebilmektedir. Ancak EceDuruApp ile yer alan bağlantıların her iki tarafı da seri, TCP Server ve TCP Client olabilmektedir. Bu çeşitlilik uygulamayı daha özellikli duruma getirmiştir. Ayrıca EceDuruDevice üzerinden veri kaydı yapmak mümkün değildir ancak EceDuruApp yazılımı veri kaydı mümkündür. EceDuruApp yazılımının EceDuruDevice yazılımına göre bir diğer avantajı ise donanımdan bağımsız olmasıdır. Bu özelliğin avantajını bir örnekle anlatmak daha kolaydır. Örneğin; elimizde 3 farklı haberleşme yapan elektronik cihazlar olsun ve bu cihazlara A, B ve C diyelim. Her cihaz birbirleri ile seri haberleşmelidir. A ile B, B ile C ve C ile A arasında seri haberleşmeye köprü olabilmek için 3 tane EceDuruDevice gerekmektedir. Bu 3 EceDuruDevice oluşturulması ve kurulması zaman ve maliyet gerektirir. Ancak tek bilgisayardan istenildiği kadar EceDuruApp uygulaması açılabilir olduğu ve fiziksel olarak bir donanıma bağlı olmadığı için bu örnekte aynı bilgisayardan 3 adet EceDuruApp yazılımının çalıştırılması yeterlidir.

EceDuruDevice yazılımının EceDuruApp yazılımına göre avantajlı olduğu noktalar vardır. Özellikle EceDuruDevice üzerinden kablosuz haberleşmeyi yaratabilecek altyapıya sahip olması EceDuruDevice yazılımının en önemli özelliğidir. Örneğin; bilgisayar üzerinden seri haberleşme arayüzü uygulaması olan bir elektronik cihazı ele alalım. Bu cihazın seri kabloyla bilgisayara bağlanması ile çalışabilir olma durumunu cihaz ile

EceDuruDevice arasında yapılacak seri bağlantının kurulması ve EceDuruDevice üzerinden bilgisayarlar TCP Client yada TCP Server kullanılması ile veri alışverişi mümkündür. Ancak cihazın kullandığı arayüz TCP Client ya da TCP Server altyapısına izin vermiyor ise bu dönüşüm için EceDuruApp yazılımına ihtiyaç vardır. Kısacası elektronik cihaz ile bu cihazla seri haberleşme kuran masaüstü uygulamasının seri kablo bağlantısı olmadan konuşurmak EceDuruDevice ve EceDuruApp ile mümkündür. Bu özelliği EceDuruDevice olmadan sağlamak şu şekilde mümkündür. Elektronik cihaz EceDuruApp yazılımı olan bir bilgisayara seri olarak bağlanır. Elektronik cihazın seri haberleşme sağladığı uygulamanın yüklü olduğu farklı bir bilgisayardan EceDuruApp yazılımı açılır. İki farklı bilgisayardaki EceDuruApp yazılımları kablosuz olarak bir TCP Server diğeri ise TCP Client olacak şekilde bağlanır. Bu sayede cihaz doğrudan masaüstü yazılımına bağlı olmadan çalıştırılabilir. Ancak bu özelliğin kullanılması için EceDuruApp uygulamasının kullanıldığı iki bilgisayarda ortak ağda bulunmalıdır. Ayrıca örnekte verildiği gibi cihaz ile zaten bilgisayar bağlantısı yapmak mümkünse doğrudan kendi masaüstü yazılımını kullanmak daha mantıklıdır. Ancak bazı koşullarda bu senaryo gerçekleştirilmesi gerekebilir. Bu koşullara sahip örnek şu şekildedir; masaüstü yazılımı ile seri haberleşen bir madencilik robotunu ele alalım. Maden robotunun kullanımı sırasında can kaybı olmaması için uzaktan erişim gerekmektedir. Ancak robotun bu tarz bir kabiliyeti yoktur. Madende bilgisayar ile seri bağlantı kurularak arayüzden yönlendirmeyi sağlayacak olan operatör EceDuruDevice ve EceDuruApp yardımıyla bu robotu maden dışından yönlendirebilir. Robot ile EceDuruDevice bağlantısı yapılır ve EceDuruDevice seri haberleşmeden TCP Server haberleşmesine dönüştürme özelliği aktif edilir. Yönlendirmeyi sağlayacak olan bilgisayardan açılan EceDuruApp ile robot arayüz yazılımı arasında seri haberleşme bağlantısı yapılır ve bilgisayar EceDuruDevice erişim noktasına bağlanır. Bilgisayar üzerinde yer alan EceDuruApp yazılımı artık EceDuruDevice yazılımına bağlanabilir duruma geldiği için EceDuruApp yazılımı TCP Client olarak EceDuruDevice yazılımına bağlanır. Artık sadece maden dışında yer alan operatör maden içinde yer alan robotu yönlendirmek için arayüz yazılımını kullanır. Tabi bu örnekte kablosuz haberleşmenin sağlanabileceği sinyal seviyelerinin var olduğunu düşünerek bahsedilmiştir. Kapalı alanlarda sinyal seviyelerindeki sıkıntı nedeniyle kullanılması sorun yaratabilecek olsa da kapalı ortam dışı seri haberleşme kuran diğer robot ve arayüz yazılımları senaryolarını da bu örneğe uyarlamak mümkündür. Örneğin; bomba imha robotu, virüs laboratuvarının yer alan robotlar, çevre koşul laboratuvarları vb. ortamlarda kullanım kolaylığı sağlayabilir.

4.1 EceDuruDevice Yazılım Doğrulama Test Prosedürü

Tez ile hazırlanan EceDuruDevice adlı elektronik cihaz ve bu cihaza ait testlerin gerçekleştirilmesi için yazılım doğrulama test prosedürü hazırlanmıştır. Hazırlanan test

prosedür dokümanı üzerinden 11.06.2021 tarihinde tarafımca yapılan testler sonrası hazırlanan test raporu verilmiştir.

Test raporunda herhangi bir test prosedüründe sorun gözlemlenmediği için EceDuruDevice yazılımı bahsedilen özellikleri barındırır ve fonksiyonel olarak çalışır durumdadır.



GÖRKEM OCAK YÜKSEK LİSANS PROJESİ

ECEDURUDEVICE YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

Doküman Kodu : EceDuruDevice.YDTP.01

Revizyon : 1.00

Tarih : 04.06.2021

DOKÜMAN REVİZYON SAYFASI

Kod: EceDuruDevice.YDTP.01

REV. NO	TARİH	SAYFA NO.	AÇIKLAMA
1.00	04.06.2021	Tümü	İlk basım

ECEURUDEVICE YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

İÇİNDEKİLER

İÇİNDEKİLER	iii
1. GENEL	4
2. KAPSAM	4
2.1 TEST ÖLÇÜM CİHAZLARI, ARAÇ-GEREÇLERİ VE AKSESUARLARI	4
3. AÇIKLAMA	4
3.1 TEST DÜZENEĞİ-1	5
3.2 TEST DÜZENEĞİ-2	5
3.3 TEST DÜZENEĞİ-3	5
4. PROSEDÜRLER	6
5. SONUÇ:	9

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruDevice.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUDEVICE YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

1. GENEL

Bu doküman Eskişehir Osmangazi Üniversitesi Bilgisayar Mühendisliği Yüksek Lisans öğrencisi Görkem OCAK tarafından tez çıktısı EceDuruDevice yazılımı ile ilgili detayları içermektedir.

2. KAPSAM

Bu doküman Yazılım Doğrulama Test Prosedürleri kapsamında uygulanacak, Genel Tanımlama, Test adımları ile ilgili detayları içermektedir.

2.1 TEST ÖLÇÜM CİHAZLARI, ARAÇ-GEREÇLERİ VE AKSESUARLARI

Sıra No	Ekipman Adı
1	Bilgisayar
2	Haberleşme Terminal Uygulaması (Hercules)

3. AÇIKLAMA

EceDuruDevice V1.0.00 yazılımı ile tez içeriğinde bahsedilen veri haberleşmesinde köprü ve haberleşme teknolojileri arasında dönüştürücü özellikleri gerçekleştirilebilmektedir. Bu özelliklerin doğrulama işlemleri bu doküman kapsamında gerçekleştirilecektir.

Seri haberleşmeden seri haberleşmeye veri hizmetlerinin doğrulanması **Prosedür-1** ile gerçekleştirilecektir.

Seri haberleşme ile TCP Server haberleşmesi arasındaki veri hizmetlerinin doğrulanması **Prosedür-2** ile gerçekleştirilecektir.

Seri haberleşme ile TCP Client haberleşmesi arasındaki veri hizmetlerinin doğrulanması **Prosedür-3** ile gerçekleştirilecektir.

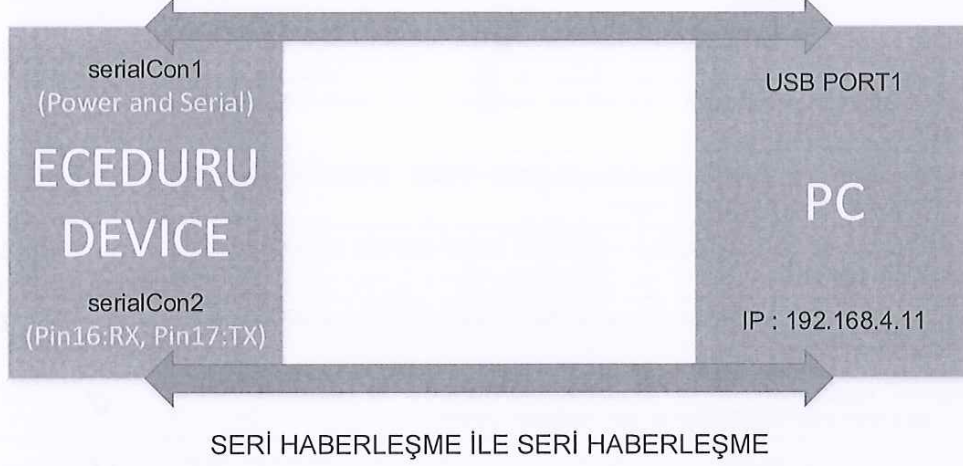
Yazılımın doğrulanması sırasında HW Group şirketi tarafından doğrulanmış Hercules uygulaması kullanılmıştır. Bu uygulamaya benzer özellikler taşıyan doğrulanmış başka terminal uygulamaları kullanımında da test prosedürleri geçerlidir.

EceDuruDevice doğrulaması sırasında sabit IP, port ve Wi-Fi ayarları kullanılmıştır.

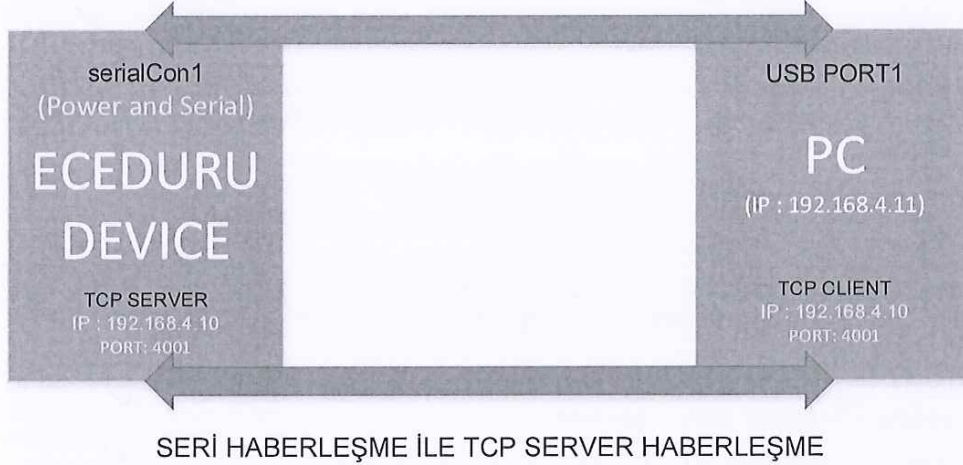
EceDuruDevice için yapılan testlerde seri olarak kendisine bağlı olan cihazın veri haberleşmesini TCP ile kendisi ile birlikte ortak bir erişim ağına bağlı olan bir cihaza dönüştürebilmektedir. Bu ortak erişim ağını sağlamak için EceDuruDevice tarafından açılan erişim noktasına bağlantı yapmak gerekmektedir.

ECEDURUDEVICE YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

3.1 TEST DÜZENEĞİ-1



3.2 TEST DÜZENEĞİ-2



3.3 TEST DÜZENEĞİ-3



Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruDevice.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUDEVICE YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

4. PROSEDÜRLER

PROSEDÜR - 1 SERİ HABERLEŞME İLE SERİ HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
1. İki adet Hercules uygulaması başlatın ve Hercules-1 ile Hercules-2 olarak belirleyin.	✓
2. Test bilgisayarı ile EceDuruDevice arasında TEST DÜZENEĞİ-1 ile verilen test düzeneği bağlantısı yapın.	✓
3. Test bilgisayarı ile EceDuruDevice arasındaki bağlantılardan serialCon1 için ayrılan portu not edin.	✓ Com3
4. Test bilgisayarı ile EceDuruDevice arasındaki bağlantılardan serialCon2 için ayrılan portu not edin.	✓ Com16
5. Hercules-1 uygulamasında Serial sayfası ile Madde-3 ile kaydedilen port adresi ve baudrate 115200 olarak seri bağlantı yapın.	✓
6. Hercules-2 uygulamasında Serial sayfası ile Madde-4 ile kaydedilen port adresi ve baudrate 115200 olarak seri bağlantı yapın.	✓
7. Test bilgisayarının Wi-Fi bağlantısını EceDuruDevice olarak seçin ve şifre olarak "12345678" ile bağlantı kurun.	✓
8. Test bilgisayarı üzerinden tarayıcı vasıtasıyla 192.168.4.10 adresine gidin. Hercules-1 uygulamasında "Configuration Status False" verisinin geldiğini doğrulayın.	✓
9. Tarayıcı ile açılan sayfada "CONNECTION CONVERT" kısmını "Serial" olarak seçili iken "SET CON1" butonuna basınız. Hercules-1 uygulamasında "Con Type 1 : Serial" verisinin geldiğini doğrulayın.	✓
10. Tarayıcı ile açılan sayfada yer alan "SAVE CONFIGURATION" butonuna basınız. Hercules-1 uygulamasında "Configuration Status True" verisinin geldiğini doğrulayın.	✓
11. Hercules-1 uygulaması vasıtasıyla EceDuruDevice üzerinden Hercules-2 uygulamasına "H1;D;H2" verisini gönderin.	✓
12. Hercules-2 uygulamasıyla "H1;D;H2" verisinin geldiğini doğrulayın.	✓
13. Hercules-2 uygulaması vasıtasıyla EceDuruDevice üzerinden Hercules-1 uygulamasına "H2;D;H1" verisini gönderin.	✓
14. Hercules-1 uygulamasıyla "H2;D;H1" verisinin geldiğini doğrulayın.	✓
15. Bu aşamaya kadar sorunsuz gelindiyse EceDuruDevice yazılımıyla birbirine fiziksel olarak bağlı olmayan iki seri portun haberleşmesini sağlamak için köprü görevi görebilmektedir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
16. Test ortamında yer alan Hercules uygulamalarını kapatın. EceDuruDevice bağlantılarını sökün.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruDevice.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUDEVICE YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 2 SERİ HABERLEŞME İLE TCP SERVER HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
1. İki adet Hercules uygulaması başlatın ve Hercules-1 ile Hercules-2 olarak belirleyin.	✓
2. Test bilgisayarı ile EceDuruDevice arasında TEST DÜZENEGİ-2 ile verilen test düzeneği bağlantısı yapın.	✓
3. Test bilgisayarı ile EceDuruDevice arasındaki bağlantılardan serialCon1 için ayrılan portu not edin.	✓ Com3
4. Hercules-1 uygulamasında Serial sayfası ile Madde-3 ile kaydedilen port adresi ve baudrate 115200 olarak seri bağlantı yapın.	✓
5. Test bilgisayarının Wi-Fi bağlantısını EceDuruDevice olarak seçin ve şifre olarak "12345678" ile bağlantı kurun.	✓
6. Test bilgisayarın IP ayarlarını aşağıda verilen gibi olmasını sağlayınız. • IPv4 Address : 192.168.4.11 • Subnet Mask : 255.255.255.0 • Default Gateway : 192.168.4.1	✓
7. Test bilgisayarı üzerinden tarayıcı vasıtasıyla 192.168.4.10 adresine gidin. Hercules-1 uygulamasında "Configuration Status False" verisinin geldiğini doğrulayın.	✓
8. Tarayıcı ile açılan sayfada "CONNECTION CONVERT" kısmını "TCP Server" olarak seçili iken "SET CON1" butonuna basınız. Hercules-1 uygulamasında "Con Type 1 : TCP Server" verisinin geldiğini doğrulayın.	✓
9. Tarayıcı ile açılan sayfada yer alan "SAVE CONFIGURATION" butonuna basınız. Hercules-1 uygulamasında "Configuration Status True" ve "Server started.Waiting Client..." verilerinin geldiğini doğrulayın.	✓
10. Hercules-2 uygulamasında TCP Client sayfası ile Ip 192.168.4.10 ve port 4001 olarak TCP/IP bağlantısı yapın. Hercules-1 uygulamasında "Client connected" verisinin geldiğini doğrulayın.	✓
11. Hercules-1 uygulaması vasıtasıyla EceDuruDevice üzerinden Hercules-2 uygulamasına "H1;D;H2" verisini gönderin.	✓
12. Hercules-2 uygulamasıyla "H1;D;H2" verisinin geldiğini doğrulayın.	✓
13. Hercules-2 uygulaması vasıtasıyla EceDuruDevice üzerinden Hercules-1 uygulamasına "H2;D;H1" verisini gönderin.	✓
14. Hercules-1 uygulamasıyla "H2;D;H1" verisinin geldiğini doğrulayın.	✓
15. Bu aşamaya kadar sorunsuz gelindiyse EceDuruDevice yazılımıyla seri haberleşme ile TCP Server haberleşmesi arası dönüştürme işlemi yapabilmektedir ayrıca köprü görevi görebilmektedir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
16. Test ortamında yer alan Hercules uygulamalarını kapatın. EceDuruDevice bağlantılarını sökün.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruDevice.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECE DURUDEVICE YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 3 SERİ HABERLEŞME İLE TCP SERVER HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
1. İki adet Hercules uygulaması başlatın ve Hercules-1 ile Hercules-2 olarak belirleyin.	✓
2. Test bilgisayarı ile EceDuruDevice arasında TEST DÜZENEĞİ-3 ile verilen test düzeneği bağlantısı yapın.	✓
3. Test bilgisayarı ile EceDuruDevice arasındaki bağlantılardan serialCon1 için ayrılan portu not edin.	✓ <i>com3</i>
4. Hercules-1 uygulamasında Serial sayfası ile Madde-3 ile kaydedilen port adresi ve baudrate 115200 olarak seri bağlantı yapın.	✓
5. Test bilgisayarının Wi-Fi bağlantısını EceDuruDevice olarak seçin ve şifre olarak "12345678" ile bağlantı kurun.	✓
6. Test bilgisayarın IP ayarlarını aşağıda verilen gibi olmasını sağlayınız. • IPv4 Address : 192.168.4.11 • Subnet Mask : 255.255.255.0 • Default Gateway : 192.168.4.1	✓
7. Hercules-2 uygulamasında TCP Server sayfası ile port 4001 olarak TCP/IP bağlantısı yapın.	✓
8. Test bilgisayarı üzerinden tarayıcı vasıtasıyla 192.168.4.10 adresine gidin. Hercules-1 uygulamasında "Configuration Status False" verisinin geldiğini doğrulayın.	✓
9. Tarayıcı ile açılan sayfada "CONNECTION CONVERT" kısmını "TCP Client" olarak seçili iken "SET CON1" butonuna basınız. Hercules-1 uygulamasında "Con Type 1 : TCP Client" verisinin geldiğini doğrulayın.	✓
10. Tarayıcı ile açılan sayfada yer alan "SAVE CONFIGURATION" butonuna basınız. Hercules-1 uygulamasında "Configuration Status True" ve "connected" verilerinin geldiğini doğrulayın.	✓
11. Hercules-1 uygulaması vasıtasıyla EceDuruDevice üzerinden Hercules-2 uygulamasına "H1;D;H2" verisini gönderin.	✓
12. Hercules-2 uygulamasıyla "H1;D;H2" verisinin geldiğini doğrulayın.	✓
13. Hercules-2 uygulaması vasıtasıyla EceDuruDevice üzerinden Hercules-1 uygulamasına "H2;D;H1" verisini gönderin.	✓
14. Hercules-1 uygulamasıyla "H2;D;H1" verisinin geldiğini doğrulayın.	✓
15. Bu aşamaya kadar sorunsuz gelindiyse EceDuruDevice yazılımıyla seri haberleşme ile TCP Client haberleşmesi arası dönüştürme işlemi yapılabilmektedir ayrıca köprü görevi görebilmektedir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
16. Test ortamında yer alan Hercules uygulamalarını kapatın. EceDuruDevice bağlantılarını sökün.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruDevice.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEURUDEVICE YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

5. SONUÇ:

Uygun	Uygun Değil	Açıklama
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Sistem çalışır ve faaldir.

TESTE KATILANLAR:

ADI SOYADI	İLETİŞİM	TARİH	İMZA
Görkem OCAK			

4.2 EceDuruApp Yazılım Doğrulama Test Prosedürü

Tez ile hazırlanan EceDuruApp adlı masaüstü yazılım uygulaması ve bu uygulamaya ait testlerin gerçekleştirilmesi için yazılım doğrulama test prosedürü hazırlanmıştır. Hazırlanan test prosedür dokümanı üzerinden 11.06.2021 tarihinde tarafımda yapılan testler sonrası hazırlanan test raporu verilmiştir.

Test raporunda herhangi bir test prosedüründe sorun gözlemlenmediği için EceDuruApp yazılımı bahsedilen özellikleri barındırır ve fonksiyonel olarak çalışır durumdadır.



GÖRKEM OCAK YÜKSEK LİSANS PROJESİ

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

Doküman Kodu : EceDuruApp.YDTP.01
Revizyon : 1.00
Tarih : 04.06.2021

DOKÜMAN REVİZYON SAYFASI

Kod: EceDuruApp.YDTP.01

REV. NO	TARİH	SAYFA NO.	AÇIKLAMA
1.00	04.06.2021	Tümü	İlk basım

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

1. GENEL

Bu doküman Eskişehir Osmangazi Üniversitesi Bilgisayar Mühendisliği Yüksek Lisans öğrencisi Görkem OCAK tarafından tez çıktısı EceDuruApp yazılımı ile ilgili detayları içermektedir.

2. KAPSAM

Bu doküman Yazılım Doğrulama Test Prosedürleri kapsamında uygulanacak, Genel Tanımlama, Test adımları ile ilgili detayları içermektedir.

2.1 TEST ÖLÇÜM CİHAZLARI, ARAÇ-GEREÇLERİ VE AKSESUARLARI

Sıra No	Ekipman Adı
1	Bilgisayar
2	Haberleşme Terminal Uygulaması (Hercules)

3. AÇIKLAMA

EceDuruApp V1.0.00 yazılımı ile tez içeriğinde bahsedilen veri haberleşmesinde köprü, haberleşme teknolojileri arasında dönüştürücü, veri kayıt edebilen ve haberleşmeye veri dahil edebilmektedir. Bu özelliklerin doğrulama işlemleri bu doküman kapsamında gerçekleştirilecektir.

Seri haberleşmeden seri haberleşmeye veri hizmetlerinin doğrulanması **Prosedür-1** ile gerçekleştirilecektir.

TCP Server haberleşmesi ile TCP Server haberleşmesi arasındaki veri hizmetlerinin doğrulanması **Prosedür-2** ile gerçekleştirilecektir.

TCP Client haberleşmesi ile TCP Client haberleşmesi arasındaki veri hizmetlerinin doğrulanması **Prosedür-3** ile gerçekleştirilecektir.

Seri haberleşme ile TCP Server haberleşmesi arasındaki veri hizmetlerinin doğrulanması **Prosedür-4** ile gerçekleştirilecektir.

Seri haberleşme ile TCP Client haberleşmesi arasındaki veri hizmetlerinin doğrulanması **Prosedür-5** ile gerçekleştirilecektir.

TCP Server haberleşmesi ile TCP Client haberleşmesi arasındaki veri hizmetlerinin doğrulanması **Prosedür-6** ile gerçekleştirilecektir.

Uygulamanın doğrulanması sırasında HW Group şirketi tarafından doğrulanmış Hercules uygulaması kullanılmıştır. Bu uygulamaya benzer özellikler taşıyan doğrulanmış başka terminal uygulamaları kullanımında da test prosedürleri geçerlidir.

Uygulamanın doğrulanması sırasında sanal seri haberleşme portları kullanılmıştır. Bu sanal seri haberleşme portları Eltima LLC şirketinin sunduğu "Configure Virtual Serial Port Driver" uygulaması ile oluşturulmuştur.

Testler sırasında kullanılan sanal seri portlar şu şekildedir;

- COM11 ile COM12 birbirlerine bağlı seri sanal port
- COM21 ile COM22 birbirlerine bağlı seri sanal port

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

İÇİNDEKİLER

İÇİNDEKİLER.....	iii
1. GENEL.....	4
2. KAPSAM	4
2.1 TEST ÖLÇÜM CİHAZLARI, ARAÇ-GEREÇLERİ VE AKSESUARLARI.....	4
3. AÇIKLAMA.....	4
4. PROSEDÜRLER.....	5
5. SONUÇ:.....	20

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

4. PROSEDÜRLER

PROSEDÜR - 1 SERİ HABERLEŞME İLE SERİ HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
1. İki adet Hercules uygulaması başlatın ve Hercules-1 ve Hercules-2 olarak belirleyin.	✓
2. Hercules-1 uygulamasında Serial sayfası ile COM11 ve baudrate 115200 olarak seri bağlantı yapın.	✓
3. Hercules-2 uygulamasında Serial sayfası ile COM21 ve baudrate 115200 olarak seri bağlantı yapın.	✓
4. EceDuruApp uygulamasını açın.	✓
5. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında Connection Type; "Serial-1" iken "SELECT" butonuna tıklayın.	✓
6. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında COM alanına COM12 ve BAUDRATE alanına 115200 yazın ve "CONNECT" butonuna tıklayın.	✓
7. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında Connection Type; "Serial-2" iken "SELECT" butonuna tıklayın.	✓
8. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında COM alanına COM22 ve BAUDRATE alanına 115200 yazın ve "CONNECT" butonuna tıklayın.	✓
9. Hercules-1 uygulaması vasıtasıyla sadece EceDuruApp Connection 1 kısmına "COM11;COM12" verisini gönderin.	✓
10. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "COM11;COM12" verisinin geldiğini doğrulayın.	✓
11. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasından "COM12;COM11" verisini "Send Message Connection 1" butonuna basarak gönderin.	✓
12. Hercules-1 uygulamasıyla "COM12;COM11" verisinin geldiğini doğrulayın.	✓
13. Hercules-2 uygulaması vasıtasıyla sadece EceDuruApp Connection 2 kısmına "COM21;COM22" verisini gönderin.	✓
14. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "COM21;COM22" verisinin geldiğini doğrulayın.	✓
15. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasından "COM22;COM21" verisini "Send Message Connection 2" butonuna basarak gönderin.	✓
16. Hercules-2 uygulamasıyla "COM22;COM21" verisinin geldiğini doğrulayın.	✓
17. Bu aşamaya kadar sorunsuz gelindiye EceDuruApp uygulaması iki bağlantı noktasından da seri haberleşme gerçekleştirebilmektedir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
18. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasında yer alan "RECORD CON1 LOG START" butonu ile log alma işlemini başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.11 09:29:48 ✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 1 SERİ HABERLEŞME İLE SERİ HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
19. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasından sırasıyla "CON1;1", "CON1;2" ve "CON1;3" verilerini "Send Message Connection 1" butonuna basarak gönderin.	✓
20. Hercules-1 uygulamasıyla "CON1;1", "CON1;2" ve "CON1;3" verilerinin geldiğini doğrulayın.	✓
21. Hercules-1 uygulaması vasıtasıyla sadece EceDuruApp Connection 1 kısmına sırayla "H1;1", "H1;2" ve "H1;3" verilerini gönderin.	✓
22. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "H1;1", "H1;2" ve "H1;3" verilerini geldiğini doğrulayın.	✓
23. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasında yer alan "RECORD CON1 LOG STOP" butonu ile log alma işlemini sonlandırın.	✓
24. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Con1_" olarak başlayan ve Madde 18 de not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 18 de saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
25. Log kaydının içerisinde "TX" tipinde "CON1;1", "CON1;2" ve "CON1;3" ve "RX" tipinde "H1;1", "H1;2" ve "H1;3" verilerinin yer aldığını doğrulayın.	✓
26. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasında yer alan "RECORD CON2 LOG START" butonu ile log alma işlemini başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.11 09:31:09 ✓
27. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasından sırasıyla "CON2;1", "CON2;2" ve "CON2;3" verilerini "Send Message Connection 2" butonuna basarak gönderin.	✓
28. Hercules-2 uygulamasıyla "CON1;1", "CON1;2" ve "CON1;3" verilerinin geldiğini doğrulayın.	✓
29. Hercules-2 uygulaması vasıtasıyla sadece EceDuruApp Connection 2 kısmına sırayla "H2;1", "H2;2" ve "H2;3" verilerini gönderin.	✓
30. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "H2;1", "H2;2" ve "H2;3" verilerini geldiğini doğrulayın.	✓
31. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasında yer alan "RECORD CON2 LOG STOP" butonu ile log alma işlemini sonlandırın.	✓
32. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Con2_" olarak başlayan ve Madde 26 da not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 26 da saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
33. Log kaydının içerisinde "TX" tipinde "CON2;1", "CON2;2" ve "CON2;3" ve "RX" tipinde "H2;1", "H2;2" ve "H2;3" verilerinin yer aldığını doğrulayın.	✓
34. Bu aşamaya kadar sorunsuz gelindiyse EceDuruApp uygulaması iki bağlantı noktasından da seri haberleşme için veri kaydı yapabilmektedir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 1 SERİ HABERLEŞME İLE SERİ HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
35. EceDuruApp uygulaması üzerinden CONFIG sayfasından ECHO kısmını aktif edin. "RECORD ECHO LOG START" butonu ile log kaydını başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.11 09:33:24 ✓
36. Hercules-1 uygulaması vasıtasıyla EceDuruApp Connection 1 kısmına "H1;CON1;CON2;H2" verisini gönderin.	✓
37. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
38. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "TX" tipinde "H1;CON1;CON2;H2" verisinin gönderildiğini doğrulayın.	✓
39. Hercules-2 uygulamasıyla "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
40. Hercules-2 uygulaması vasıtasıyla EceDuruApp Connection 2 kısmına "H2;CON2;CON1;H1" verisini gönderin.	✓
41. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
42. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "TX" tipinde "H2;CON2;CON1;H1" verisinin gönderildiğini doğrulayın.	✓
43. Hercules-1 uygulamasıyla "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
44. EceDuruApp uygulaması üzerinden CONFIG sayfasında yer alan "RECORD ECHO LOG STOP" butonu ile log alma işlemini sonlandırın.	✓
45. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Echo_" olarak başlayan ve Madde 35 de not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 35 de saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
46. Log kaydının içerisinde "CON1RX" ve "CON2TX" tipinde "H1;CON1;CON2;H2" ile "CON1TX" ve "CON2RX" tipinde "H2;CON2;CON1;H1" verilerinin yer aldığını doğrulayın.	✓
47. Bu aşamaya kadar sorunsuz gelindiye EceDuruApp uygulaması iki bağlantı noktasından da seri haberleşme için köprü görevi görebilmektedir ve köprü verilerinin veri kaydı yapılabilir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
48. Test ortamında yer alan Hercules ve EceDuruApp uygulamalarını kapatın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 2 TCP SERVER HABERLEŞME İLE TCP SERVER HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
1. Bilgisayarın bağlı olduğu IP bilgisini öğrenin ve not edin. (Windows bilgisayarlar için Command Prompt üzerinden ipconfig ile öğrenilebilmektedir.)	192.168.4.11 ✓
2. EceDuruApp uygulamasını açın.	✓
3. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında Connection Type; "TCP Server-1" iken "SELECT" butonuna tıklayın.	✓
4. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında IP alanına Madde 1 ile not edilen bilgisayar Ip değerini ve PORT alanına 4000 yazın ve "CONNECT" butonuna tıklayın.	✓
5. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında Connection Type; "TCP Server-2" iken "SELECT" butonuna tıklayın.	✓
6. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında IP alanına Madde 1 ile not edilen bilgisayar Ip değerini ve PORT alanına 4001 yazın ve "CONNECT" butonuna tıklayın.	✓
7. İki adet Hercules uygulaması başlatın ve Hercules-1 ve Hercules-2 olarak belirleyin.	✓
8. Hercules-1 uygulamasında TCP Client sayfası ile Madde 1 ile not edilen bilgisayar Ip değeri ve port 4000 olarak TCP/IP bağlantısı yapın.	✓
9. Hercules-2 uygulamasında TCP Client sayfası ile Madde 1 ile not edilen bilgisayar Ip değeri ve port 4001 olarak TCP/IP bağlantısı yapın.	✓
10. Hercules-1 uygulaması vasıtasıyla sadece EceDuruApp Connection 1 kısmına "H1;CON1" verisini gönderin.	✓
11. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "H1;CON1" verisinin geldiğini doğrulayın.	✓
12. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasından "CON1;H1" verisini "Send Message Connection 1" butonuna basarak gönderin.	✓
13. Hercules-1 uygulamasıyla "CON1;H1" verisinin geldiğini doğrulayın.	✓
14. Hercules-2 uygulaması vasıtasıyla sadece EceDuruApp Connection 2 kısmına "H2;CON2" verisini gönderin.	✓
15. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "H2;CON2" verisinin geldiğini doğrulayın.	✓
16. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasından "CON2;H2" verisini "Send Message Connection 2" butonuna basarak gönderin.	✓
17. Hercules-2 uygulamasıyla "CON2;H2" verisinin geldiğini doğrulayın.	✓
18. Bu aşamaya kadar sorunsuz gelindiye EceDuruApp uygulaması iki bağlantı noktasından da TCP Server haberleşmesi gerçekleştirilebilmektedir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
19. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasında yer alan "RECORD CON1 LOG START" butonu ile log alma işlemi başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.11 09:39:27 ✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 2 TCP SERVER HABERLEŞME İLE TCP SERVER HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
20. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasından sırasıyla "CON1;1", "CON1;2" ve "CON1;3" verilerini "Send Message Connection 1" butonuna basarak gönderin.	✓
21. Hercules-1 uygulamasıyla "CON1;1", "CON1;2" ve "CON1;3" verilerinin geldiğini doğrulayın.	✓
22. Hercules-1 uygulaması vasıtasıyla sadece EceDuruApp Connection 1 kısmına sırayla "H1;1", "H1;2" ve "H1;3" verilerini gönderin.	✓
23. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "H1;1", "H1;2" ve "H1;3" verilerini geldiğini doğrulayın.	✓
24. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasında yer alan "RECORD CON1 LOG STOP" butonu ile log alma işlemini sonlandırın.	✓
25. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Con1_" olarak başlayan ve Madde 19 da not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 19 da saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
26. Log kaydının içerisinde "TX" tipinde "CON1;1", "CON1;2" ve "CON1;3" ve "RX" tipinde "H1;1", "H1;2" ve "H1;3" verilerinin yer aldığını doğrulayın.	✓
27. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasında yer alan "RECORD CON2 LOG START" butonu ile log alma işlemini başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.11 09:42:04 ✓
28. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasından sırasıyla "CON2;1", "CON2;2" ve "CON2;3" verilerini "Send Message Connection 2" butonuna basarak gönderin.	✓
29. Hercules-2 uygulamasıyla "CON1;1", "CON1;2" ve "CON1;3" verilerinin geldiğini doğrulayın.	✓
30. Hercules-2 uygulaması vasıtasıyla sadece EceDuruApp Connection 2 kısmına sırayla "H2;1", "H2;2" ve "H2;3" verilerini gönderin.	✓
31. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "H2;1", "H2;2" ve "H2;3" verilerini geldiğini doğrulayın.	✓
32. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasında yer alan "RECORD CON2 LOG STOP" butonu ile log alma işlemini sonlandırın.	✓
33. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Con2_" olarak başlayan ve Madde 27 de not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 27 de saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
34. Log kaydının içerisinde "TX" tipinde "CON2;1", "CON2;2" ve "CON2;3" ve "RX" tipinde "H2;1", "H2;2" ve "H2;3" verilerinin yer aldığını doğrulayın.	✓
35. Bu aşamaya kadar sorunsuz gelindiyse EceDuruApp uygulaması iki bağlantı noktasından da TCP Server haberleşme için veri kaydı yapabilmektedir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 2 TCP SERVER HABERLEŞME İLE TCP SERVER HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
36. EceDuruApp uygulaması üzerinden CONFIG sayfasından ECHO kısmını aktif edin. "RECORD ECHO LOG START" butonu ile log kaydını başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.11 09:44:41 ✓
37. Hercules-1 uygulaması vasıtasıyla EceDuruApp Connection 1 kısmına "H1;CON1;CON2;H2" verisini gönderin.	✓
38. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
39. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "TX" tipinde "H1;CON1;CON2;H2" verisinin gönderildiğini doğrulayın.	✓
40. Hercules-2 uygulamasıyla "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
41. Hercules-2 uygulaması vasıtasıyla EceDuruApp Connection 2 kısmına "H2;CON2;CON1;H1" verisini gönderin.	✓
42. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
43. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "TX" tipinde "H2;CON2;CON1;H1" verisinin gönderildiğini doğrulayın.	✓
44. Hercules-1 uygulamasıyla "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
45. EceDuruApp uygulaması üzerinden CONFIG sayfasında yer alan "RECORD ECHO LOG STOP" butonu ile log alma işlemini sonlandırın.	✓
46. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Echo_" olarak başlayan ve Madde 36 da not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 36 da saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
47. Log kaydının içerisinde "CON1RX" ve "CON2TX" tipinde "H1;CON1;CON2;H2" ile "CON1TX" ve "CON2RX" tipinde "H2;CON2;CON1;H1" verilerinin yer aldığını doğrulayın.	✓
48. Bu aşamaya kadar sorunsuz gelindiyse EceDuruApp uygulaması iki bağlantı noktasından da TCP Server haberleşme için köprü görevi görebilmektedir ve köprü verilerinin veri kaydı yapılabilir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
49. Test ortamında yer alan Hercules ve EceDuruApp uygulamalarını kapatın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 3 TCP CLIENT HABERLEŞME İLE TCP CLIENT HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
1. Bilgisayarın bağlı olduğu IP bilgisini öğrenin ve not edin. (Windows bilgisayarlar için Command Prompt üzerinden ipconfig ile öğrenilebilmektedir.)	192.168.4.11 ✓
2. İki adet Hercules uygulaması başlatın ve Hercules-1 ve Hercules-2 olarak belirleyin.	✓
3. Hercules-1 uygulamasında TCP Server sayfası ile port 4000 olarak TCP/IP bağlantısı yapın.	✓
4. Hercules-2 uygulamasında TCP Server sayfası ile port 4001 olarak TCP/IP bağlantısı yapın.	✓
5. EceDuruApp uygulamasını açın.	✓
6. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında Connection Type; "TCP Client-1" iken "SELECT" butonuna tıklayın.	✓
7. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında IP alanına Madde 1 ile not edilen bilgisayar Ip değerini ve PORT alanına 4000 yazın ve "CONNECT" butonuna tıklayın.	✓
8. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında Connection Type; "TCP Client -2" iken "SELECT" butonuna tıklayın.	✓
9. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında IP alanına Madde 1 ile not edilen bilgisayar Ip değerini ve PORT alanına 4001 yazın ve "CONNECT" butonuna tıklayın.	✓
10. Hercules-1 uygulaması vasıtasıyla sadece EceDuruApp Connection 1 kısmına "H1;CON1" verisini gönderin.	✓
11. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "H1;CON1" verisinin geldiğini doğrulayın.	✓
12. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasından "CON1;H1" verisini "Send Message Connection 1" butonuna basarak gönderin.	✓
13. Hercules-1 uygulamasıyla "CON1;H1" verisinin geldiğini doğrulayın.	✓
14. Hercules-2 uygulaması vasıtasıyla sadece EceDuruApp Connection 2 kısmına "H2;CON2" verisini gönderin.	✓
15. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "H2;CON2" verisinin geldiğini doğrulayın.	✓
16. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasından "CON2;H2" verisini "Send Message Connection 2" butonuna basarak gönderin.	✓
17. Hercules-2 uygulamasıyla "CON2;H2" verisinin geldiğini doğrulayın.	✓
18. Bu aşamaya kadar sorunsuz gelindiye EceDuruApp uygulaması iki bağlantı noktasından da TCP Client haberleşmesi gerçekleştirebilmektedir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
19. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasında yer alan "RECORD CON1 LOG START" butonu ile log alma işlemini başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.11 09:50:07 ✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 3 TCP CLIENT HABERLEŞME İLE TCP CLIENT HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
20. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasından sırasıyla "CON1;1", "CON1;2" ve "CON1;3" verilerini "Send Message Connection 1" butonuna basarak gönderin.	✓
21. Hercules-1 uygulamasıyla "CON1;1", "CON1;2" ve "CON1;3" verilerinin geldiğini doğrulayın.	✓
22. Hercules-1 uygulaması vasıtasıyla sadece EceDuruApp Connection 1 kısmına sırayla "H1;1", "H1;2" ve "H1;3" verilerini gönderin.	✓
23. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "H1;1", "H1;2" ve "H1;3" verilerini geldiğini doğrulayın.	✓
24. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasında yer alan "RECORD CON1 LOG STOP" butonu ile log alma işlemini sonlandırın.	✓
25. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Con1_" olarak başlayan ve Madde 19 da not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 19 da saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
26. Log kaydının içerisinde "TX" tipinde "CON1;1", "CON1;2" ve "CON1;3" ve "RX" tipinde "H1;1", "H1;2" ve "H1;3" verilerinin yer aldığını doğrulayın.	✓
27. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasında yer alan "RECORD CON2 LOG START" butonu ile log alma işlemini başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.11 09:52:45 ✓
28. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasından sırasıyla "CON2;1", "CON2;2" ve "CON2;3" verilerini "Send Message Connection 2" butonuna basarak gönderin.	✓
29. Hercules-2 uygulamasıyla "CON1;1", "CON1;2" ve "CON1;3" verilerinin geldiğini doğrulayın.	✓
30. Hercules-2 uygulaması vasıtasıyla sadece EceDuruApp Connection 2 kısmına sırayla "H2;1", "H2;2" ve "H2;3" verilerini gönderin.	✓
31. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "H2;1", "H2;2" ve "H2;3" verilerini geldiğini doğrulayın.	✓
32. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasında yer alan "RECORD CON2 LOG STOP" butonu ile log alma işlemini sonlandırın.	✓
33. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Con2_" olarak başlayan ve Madde 27 de not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 27 de saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
34. Log kaydının içerisinde "TX" tipinde "CON2;1", "CON2;2" ve "CON2;3" ve "RX" tipinde "H2;1", "H2;2" ve "H2;3" verilerinin yer aldığını doğrulayın.	✓
35. Bu aşamaya kadar sorunsuz gelindiyse EceDuruApp uygulaması iki bağlantı noktasından da TCP Client haberleşme için veri kaydı yapabilmektedir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 3 TCP CLIENT HABERLEŞME İLE TCP CLIENT HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
36. EceDuruApp uygulaması üzerinden CONFIG sayfasından ECHO kısmını aktif edin. "RECORD ECHO LOG START" butonu ile log kaydını başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.11 09:54:01 ✓
37. Hercules-1 uygulaması vasıtasıyla EceDuruApp Connection 1 kısmına "H1;CON1;CON2;H2" verisini gönderin.	✓
38. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
39. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "TX" tipinde "H1;CON1;CON2;H2" verisinin gönderildiğini doğrulayın.	✓
40. Hercules-2 uygulamasıyla "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
41. Hercules-2 uygulaması vasıtasıyla EceDuruApp Connection 2 kısmına "H2;CON2;CON1;H1" verisini gönderin.	✓
42. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
43. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "TX" tipinde "H2;CON2;CON1;H1" verisinin gönderildiğini doğrulayın.	✓
44. Hercules-1 uygulamasıyla "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
45. EceDuruApp uygulaması üzerinden CONFIG sayfasında yer alan "RECORD ECHO LOG STOP" butonu ile log alma işlemini sonlandırın.	✓
46. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Echo_" olarak başlayan ve Madde 36 da not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 36 da saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
47. Log kaydının içerisinde "CON1RX" ve "CON2TX" tipinde "H1;CON1;CON2;H2" ile "CON1TX" ve "CON2RX" tipinde "H2;CON2;CON1;H1" verilerinin yer aldığını doğrulayın.	✓
48. Bu aşamaya kadar sorunsuz gelindiye EceDuruApp uygulaması iki bağlantı noktasından da TCP Client haberleşme için köprü görevi görebilmektedir ve köprü verilerinin veri kaydı yapılabilir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
49. Test ortamında yer alan Hercules ve EceDuruApp uygulamalarını kapatın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 4 SERİ HABERLEŞME İLE TCP SERVER HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
1. Bilgisayarın bağlı olduğu IP bilgisini öğrenin ve not edin. (Windows bilgisayarlar için Command Prompt üzerinden ipconfig ile öğrenilebilmektedir.)	192.168.4.11 ✓
2. EceDuruApp uygulamasını açın.	✓
3. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında Connection Type; "Serial-1" iken "SELECT" butonuna tıklayın.	✓
4. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında COM alanına COM12 ve BAUDRATE alanına 115200 yazın ve "CONNECT" butonuna tıklayın.	✓
5. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında Connection Type; "TCP Server-2" iken "SELECT" butonuna tıklayın.	✓
6. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında IP alanına Madde 1 ile not edilen bilgisayar Ip değerini ve PORT alanına 4001 yazın ve "CONNECT" butonuna tıklayın.	✓
7. İki adet Hercules uygulaması başlatın ve Hercules-1 ve Hercules-2 olarak belirleyin.	✓
8. Hercules-1 uygulamasında Serial sayfası ile COM11 ve baudrate 115200 olarak seri bağlantı yapın.	✓
9. Hercules-2 uygulamasında TCP Client sayfası ile Madde 1 ile not edilen bilgisayar Ip değeri ve port 4001 olarak TCP/IP bağlantısı yapın.	✓
10. EceDuruApp uygulaması üzerinden CONFIG sayfasından ECHO kısmını aktif edin. "RECORD ECHO LOG START" butonu ile log kaydını başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.11 09:58:01 ✓
11. Hercules-1 uygulaması vasıtasıyla EceDuruApp Connection 1 kısmına "H1;CON1;CON2;H2" verisini gönderin.	✓
12. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
13. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "TX" tipinde "H1;CON1;CON2;H2" verisinin gönderildiğini doğrulayın.	✓
14. Hercules-2 uygulamasıyla "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
15. Hercules-2 uygulaması vasıtasıyla EceDuruApp Connection 2 kısmına "H2;CON2;CON1;H1" verisini gönderin.	✓
16. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
17. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "TX" tipinde "H2;CON2;CON1;H1" verisinin gönderildiğini doğrulayın.	✓
18. Hercules-1 uygulamasıyla "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
19. EceDuruApp uygulaması üzerinden CONFIG sayfasında yer alan "RECORD ECHO LOG STOP" butonu ile log alma işlemini sonlandırın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 4 SERİ HABERLEŞME İLE TCP SERVER HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
20. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Echo_" olarak başlayan ve Madde 10 da not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 10 da saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
21. Log kaydının içerisinde "CON1RX" ve "CON2TX" tipinde "H1;CON1;CON2;H2" ile "CON1TX" ve "CON2RX" tipinde "H2;CON2;CON1;H1" verilerinin yer aldığını doğrulayın.	✓
22. Bu aşamaya kadar sorunsuz gelindiyse EceDuruApp uygulaması seri haberleşme ile TCP Server haberleşmesi arası dönüştürme işlemi yapabilmektedir ayrıca köprü görevi görebilmektedir ve köprü verilerinin veri kaydı yapılabilmektedir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
23. Test ortamında yer alan Hercules ve EceDuruApp uygulamalarını kapatın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 5 SERİ HABERLEŞME İLE TCP CLIENT HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
1. Bilgisayarın bağlı olduğu IP bilgisini öğrenin ve not edin. (Windows bilgisayarlar için Command Prompt üzerinden ipconfig ile öğrenilebilmektedir.)	192.168.4.11 ✓
2. İki adet Hercules uygulaması başlatın ve Hercules-1 ve Hercules-2 olarak belirleyin.	✓
3. Hercules-1 uygulamasında Serial sayfası ile COM11 ve baudrate 115200 olarak seri bağlantı yapın.	✓
4. Hercules-2 uygulamasında TCP Server sayfası ile port 4001 olarak TCP/IP bağlantısı yapın.	✓
5. EceDuruApp uygulamasını açın.	✓
6. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında Connection Type; "Serial-1" iken "SELECT" butonuna tıklayın.	✓
7. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında COM alanına COM12 ve BAUDRATE alanına 115200 yazın ve "CONNECT" butonuna tıklayın.	✓
8. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında Connection Type; "TCP Client-2" iken "SELECT" butonuna tıklayın.	✓
9. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında IP alanına Madde 1 ile not edilen bilgisayar Ip değerini ve PORT alanına 4001 yazın ve "CONNECT" butonuna tıklayın.	✓
10. EceDuruApp uygulaması üzerinden CONFIG sayfasından ECHO kısmını aktif edin. "RECORD ECHO LOG START" butonu ile log kaydını başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.11 10:02:04 ✓
11. Hercules-1 uygulaması vasıtasıyla EceDuruApp Connection 1 kısmına "H1;CON1;CON2;H2" verisini gönderin.	✓
12. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
13. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "TX" tipinde "H1;CON1;CON2;H2" verisinin gönderildiğini doğrulayın.	✓
14. Hercules-2 uygulamasıyla "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
15. Hercules-2 uygulaması vasıtasıyla EceDuruApp Connection 2 kısmına "H2;CON2;CON1;H1" verisini gönderin.	✓
16. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
17. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "TX" tipinde "H2;CON2;CON1;H1" verisinin gönderildiğini doğrulayın.	✓
18. Hercules-1 uygulamasıyla "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
19. EceDuruApp uygulaması üzerinden CONFIG sayfasında yer alan "RECORD ECHO LOG STOP" butonu ile log alma işlemini sonlandırın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 5 SERİ HABERLEŞME İLE TCP CLIENT HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
20. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Echo_" olarak başlayan ve Madde 10 da not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 10 da saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
21. Log kaydının içerisinde "CON1RX" ve "CON2TX" tipinde "H1;CON1;CON2;H2" ile "CON1TX" ve "CON2RX" tipinde "H2;CON2;CON1;H1" verilerinin yer aldığını doğrulayın.	✓
22. Bu aşamaya kadar sorunsuz gelindiyse EceDuruApp uygulaması seri haberleşme ile TCP Client haberleşmesi arası dönüştürme işlemi yapabilmektedir ayrıca köprü görevi görebilmektedir ve köprü verilerinin veri kaydı yapılabilir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
23. Test ortamında yer alan Hercules ve EceDuruApp uygulamalarını kapatın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 6 TCP SERVER HABERLEŞME İLE TCP CLIENT HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
1. Bilgisayarın bağlı olduğu IP bilgisini öğrenin ve not edin. (Windows bilgisayarlar için Command Prompt üzerinden ipconfig ile öğrenilebilmektedir.)	192.168.4.1 ✓
2. İki adet Hercules uygulaması başlatın ve Hercules-1 ve Hercules-2 olarak belirleyin.	✓
3. EceDuruApp uygulamasını açın.	✓
4. Hercules-2 uygulamasında TCP Server sayfası ile port 4001 olarak TCP/IP bağlantısı yapın.	✓
5. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında Connection Type; "TCP Server-1" iken "SELECT" butonuna tıklayın.	✓
6. EceDuruApp uygulamasında CONFIG sayfasında Connection 1 kısmında IP alanına Madde 1 ile not edilen bilgisayar Ip değerini ve PORT alanına 4000 yazın ve "CONNECT" butonuna tıklayın.	✓
7. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında Connection Type; "TCP Client-2" iken "SELECT" butonuna tıklayın.	✓
8. EceDuruApp uygulamasında CONFIG sayfasında Connection 2 kısmında IP alanına Madde 1 ile not edilen bilgisayar Ip değerini ve PORT alanına 4001 yazın ve "CONNECT" butonuna tıklayın.	✓
9. Hercules-1 uygulamasında TCP Client sayfası ile Madde 1 ile not edilen bilgisayar Ip değeri ve port 4000 olarak TCP/IP bağlantısı yapın.	✓
10. EceDuruApp uygulaması üzerinden CONFIG sayfasından ECHO kısmını aktif edin. "RECORD ECHO LOG START" butonu ile log kaydını başlatın ve buton basılması anına ait Yıl, Ay, Gün, Saat, Dakika ve Saniye bilgilerini not edin.	2021.06.4 10:07:02 ✓
11. Hercules-1 uygulaması vasıtasıyla EceDuruApp Connection 1 kısmına "H1;CON1;CON2;H2" verisini gönderin.	✓
12. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "RX" tipinde "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
13. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "TX" tipinde "H1;CON1;CON2;H2" verisinin gönderildiğini doğrulayın.	✓
14. Hercules-2 uygulamasıyla "H1;CON1;CON2;H2" verisinin geldiğini doğrulayın.	✓
15. Hercules-2 uygulaması vasıtasıyla EceDuruApp Connection 2 kısmına "H2;CON2;CON1;H1" verisini gönderin.	✓
16. EceDuruApp uygulaması üzerinden CONNECTION 2 sayfasına geçin ve "RX" tipinde "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
17. EceDuruApp uygulaması üzerinden CONNECTION 1 sayfasına geçin ve "TX" tipinde "H2;CON2;CON1;H1" verisinin gönderildiğini doğrulayın.	✓
18. Hercules-1 uygulamasıyla "H2;CON2;CON1;H1" verisinin geldiğini doğrulayın.	✓
19. EceDuruApp uygulaması üzerinden CONFIG sayfasında yer alan "RECORD ECHO LOG STOP" butonu ile log alma işlemini sonlandırın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

PROSEDÜR - 6 TCP SERVER HABERLEŞME İLE TCP CLIENT HABERLEŞME TESTİ	SONUÇ (Uygun/ Değil)
20. Bilgisayar üzerinden EceDuruApp uygulamasının çalıştırıldığı dosya dizisinde yer alan Logs klasöründe yer alan "Echo_" olarak başlayan ve Madde 10 da not edilen bilgilere ait ".txt" uzantılı log kaydını açın. (Madde 10 da saniye değerini zamanında not edilememiş olması ihtimaline karşı dikkate almayın ve diğer değerlere göre uygun olan log kaydını bulun.)	✓
21. Log kaydının içerisinde "CON1RX" ve "CON2TX" tipinde "H1;CON1;CON2;H2" ile "CON1TX" ve "CON2RX" tipinde "H2;CON2;CON1;H1" verilerinin yer aldığını doğrulayın.	✓
22. Bu aşamaya kadar sorunsuz gelindiyse EceDuruApp uygulaması TCP Server haberleşme ile TCP Client haberleşmesi arası dönüştürme işlemi yapabilmektedir ayrıca köprü görevi görebilmektedir ve köprü verilerinin veri kaydı yapılabilir. Eğer sorunla karşılaşıldıysa testi sonlandırın.	✓
23. Test ortamında yer alan Hercules ve EceDuruApp uygulamalarını kapatın.	✓

Görkem OCAK 2021 Haziran
Yüksek Lisans Tez

KOD : EceDuruApp.YDTP.01
REV. NO : 1.00
REV. TARİH : 04.06.2021

ECEDURUAPP YAZILIM DOĞRULAMA TEST PROSEDÜRÜ

5. SONUÇ:

Uygun	Uygun Değil	Açıklama
<input checked="" type="checkbox"/>	<input type="checkbox"/>	Sistem çalışır ve faaldir.

TESTE KATILANLAR:

ADI SOYADI	İLETİŞİM	TARİH	İMZA
Görkem OCAK			

5. SONUÇ VE ÖNERİLER

Bu tez çalışması ile elektronik cihaz teknolojisi yakından incelenmiştir. Elektronik haberleşme ile ilgili yapılan çalışmalara ait yapılan literatür çalışması dışında seri haberleşme ve TCP/IP haberleşme protokolü örneklendirilmiştir.

Kablolu ve kablosuz haberleşme üzerine yapılan teorik anlatımlardan bağımsız olarak kablolu bir haberleşmenin kablosuz bir haberleşmeye dönüştürülmesine dair senaryosal anlatımlarda tez çalışmasında yer almaktadır. EceDuruDevice ve EceDuruApp özellikleri ve kullanım alanlarına dair bilgilendirmelerin tez sonrası yapılan çalışmalar ile iyileştirilmesi mümkündür. Özellikle EceDuruDevice yazılımı EceDuruApp yazılımına göre biraz daha toy bir yazılımdır. EceDuruDevice yazılımının geliştirilmesine kullanıcı tercihine bağlı olarak cihaz erişim noktasının değiştirilebilir olması özelliği eklenebilir. Böylece kullanıcı istediği zaman EceDuruDevice ile açılan erişim noktasını isterse konfigürasyon sayfasından oluşturacağı Wi-Fi ayarları ile ortak erişim noktası oluşturabilir. Bu özelliğin eklenmesi ile EceDuruDevice bağlanılan cihazdan bağlanan cihaza dönüşebilir. Bu sayede bir sisteme cihaz olarak, arayüz olarak yada köprü olarak eklenebilir. Şu an EceDuruDevice etrafında bir sistem oluşturmak mümkündür. Bu sistemde sınırlıdır. Bu sistemde EceDuruDevice kesinlikle bir bağlantısını seri haberleşme yapmak zorundadır. Bu sınırın ortadan kaldırılması da yapılabilecek geliştirme çalışmalarındandır. Bu geliştirme ile seri haberleşmeyi seri, TCP Server ve TCP Client olarak dönüştürme ve köprü olma yeteneğine sahip olan EceDuruDevice seri, TCP Server ve TCP Client haberleşmelerini yine seri, TCP Server ve TCP Client haberleşmelerine dönüştürme ve köprü olma yeteneğine sahip olacaktır. Bu özelliğin eklenmesi ile EceDuruApp ile sağlanan özelliklerden birine daha sahip olması hedeflenmektedir.

EceDuruApp yazılımı haberleşme teknolojilerini dönüştürme işleminde seri, TCP Server ve TCP Client ile sınırlıdır. Bu dönüştürmelere UDP, CAN, ARINC 429, MIL-STD-1553 gibi protokollerin eklenmesi ile yazılım daha üst seviye duruma getirilebilir. Tabi bu işlemler sıralı olarak gerçekleştirilmelidir. Bu sıralama için de kullanım sıklığını dikkate almak mantıklı gözükmektedir. Bu nedenle öncelikle UDP haberleşmesini de özellikler arasına dahil etmek hem kullanışlı hemde daha kolay olacaktır. Çünkü bilindiği üzere TCP/IP haberleşmesi ile UDP haberleşmesi birbirlerine benzerlik göstermektedir. Bu benzerlikler yazılım aşamasında daha fazladır. Ayrıca EceDuruApp yazılımda TCP Server özelliği için yazılıma bağlanan Client bilgilerinin yer alması kullanıcı için önemli bir özellik olabilir. Bağlı olan kullanıcıların bağlantı zamanı, bağlantı

adres bilgileri ve bu kullanıcıların sayısı gibi bilgiler uygulamanın kullanıcı dostu olmasını sağlar. Ayrıca benzer şekilde oluşturulan terminal bazlı uygulamalarda rastlanmamış olan özgün bir yetenek olan kullanıcı bağlantısını kesme özelliği de eklenmesi fark yaratıcı unsur olabilir. Bu özellik EceDuruApp uygulamasının hazırlanması sürecinde denenmiştir ve başarılı olmuştur. Fakat kullanıma alınması için gerekli olan testler henüz tamamlanmadığı için bu versiyonda bu özelliğe yer verilmemiştir.

EceDuruDevice yazılımının köprü ve dönüştürücü özelliği dışında veri kayıt işlemini de sağlaması için çalışma yapılabilir. Bu veri kayıt özelliği fiziksel bir depolama birimine ya da sanal sunucu da yer alan bir veritabanı ile olabilir. Fiziksel depolama alanının sanal sunucu da yer alan bir veritabanına göre tercih edilmesinin en önemli nedeni güvenlik olabilir. Fiziksel olarak bir bağlantı kurulan depolama alanı cihaz tasarımında yer alması depolama alanı ile kurulacak haberleşmenin güvenliği için yeni bir çalışma yapılmasına gerek duyulmayabilir. Ancak sanal bir sunucu da yer alan veritabanında verilerin tutulması tercih edilirse verinin doğru, eksiksiz kaydedilmesi ve verinin erişilebilirliği aşamalarında güvenlik açıkları oluşabilir. Bu nedenle fiziksel bir alan tercih etmek güvenlik açısından daha kolay görülmektedir. Ancak fiziksel alan için tercih edilecek depolama alanının kapasitesi arttıkça maliyeti artacak olması ve fiziksel alana erişime ait kısıtlamalar olması nedeniyle tercih edilmeyebilir. Bu nedenle sanal bir sunucu da yer alan veritabanı ile veri kaydı yapmak kapasite, maliyet ve erişim kolaylığı açısından önemli fırsatlar sunmaktadır. EceDuruDevice yazılımı veri kaydetme özelliği kazanması durumunda kullanım alanına göre fiziksel yada sanal bir sunucu da yer alan veritabanı ile depolama seçilebilir olması cihaz kalitesini arttıracaktır.

EceDuruApp yazılımının veri kayıt işlemi uygulamanın kullanıldığı bilgisayar üzerinde ".txt" formatında veri kaydı olarak gerçekleşmektedir. Bu kayıtların anlık olarak farklı bir bilgisayardan gözükmeleri için sanal bir sunucu da yer alan veritabanına veri kayıt özelliği eklemek uygulamayı bir üst seviyeye taşıyabilir. Hatta geliştirme aşamasında veritabanının yapısında yapılacak birkaç değişiklik ile ortak erişim noktası olma zorunluluğu ortadan kaldırılabilir. Örneğin; EceDuruApp uygulamasına ait bir bilgisayar ile bir elektronik cihaz haberleşme arayüzü seri, TCP Server ya da TCP Client olarak bağlı olsun. Elektronik cihaza ait kullanıcı arayüzü ise farklı bir ülkedeki bir bilgisayarda EceDuruApp uygulamasına bağlı olsun. Eğer iki farklı bilgisayarda açık olan iki EceDuruApp uygulaması veritabanı üzerinden veri alışverişi gerçekleştirebilme özelliğine sahipse farklı bir ülkeden robotun kullanıcı arayüzü üzerinden hareketi sağlanabilir. Bu özellik haberleşme özelinde geliştirilen ve cihaz, uygulama gibi yapılardan bağımsız olduğu için kullanım alanı çok geniştir. Cihaz ve cihaza bağlı olan bir başka cihaz yada arayüz yazılımının EceDuruApp ile haberleşme kurabilmesi ile EceDuruApp tüm özellikleri

kullanılabilir şekilde gelir. Bu nedenle EceDuruApp birçok projeye rahatlıkla dahil olabilecektir.

EceDuruDevice tasarımında esas işlevi gerçekleştiren kısım yazılım olduğu için tez çalışması boyunca yazılıma yoğunlaşmıştır. Ancak cihaz ticari bir ürüne evrilmesi için gerekli cihaz donanım yani elektronik kart tasarımına ve cihaz mekanik tasarımına yönelik çalışma yapılması gerekmektedir. EceDuruDevice tez çalışmasında sadece seri haberleşmeyi seri, TCP Server ve TCP Client haberleşmelerine dönüştürme ve köprü görevi görmesinden dolayı kullanılan seri iki kanal için iki adet USB port opsiyonu vardır. TCP Server ve TCP Client özellikleri için ise ESP8266 ile sağlanan kablosuz ağ özelliği kullanılmaktadır. Ancak ürün geliştirme aşamasında seri kanallar için seri dişi/erkek konnektörleri ile ethernet soketleri eklenebilir. Bu özellik ile kablosuz özelliği olmayan ancak ethernet kablosu ile TCP Server ve TCP Client özelliği barındıran cihazlara da erişim sağlanabilir. Tez çalışmasındaki haliyle EceDuruDevice ethernet kablosu ile TCP Server ve TCP Client özelliği barındıran cihazlara bağlantı sağlamak için Wi-Fi özelliği taşıyan bilgisayar gibi ara birimlere ihtiyaç duymaktadır. Mekanik olarak yapılacak çalışma içinse cihazın kullanım alanına uygun bir tasarıma ve malzeme seçimine gidilmelidir. Çevresel koşulları ve cihazın çalışma şartları belirlemeden cihaz mekanik tasarımı yapılması kullanışsız ve maliyetli bir iş paketidir. Elektronik devre tasarımında kullanılacak malzemelerin seçimi aşamasında cihazın sağlaması gereken standartları karşılayacak minimum maliyetli malzemelere yönelim sağlanırsa cihazın ticari bir ürüne çevrilmesinde bir engel kalmaz. Cihazın ticari ürüne dönüştürülmesi için tez çalışmasında oluşturulan EceDuruDevice prototip ürün olarak kullanılabilir. Ancak cihaza yeni özellikler kazandırmak için EceDuruDevice yetersiz kalma durumuyla karşılaşılabilir. Buna karşın yine de EceDuruDevice V1.0.00 haliyle birçok projeye rahatlıkla dahil olabilecek seviyededir.

EceDuruDevice tasarımında kartın güç hattı ile 1 numaralı seri haberleşmeyi kuran hattı ortak kablodadır. Bu kablo hattında güç hattını ayırarak cihazın bağlantı kurduğu cihazdan beslenmesi gerekliliği ortadan kaldırılabilir. EceDuruDevice V1.0.00 tasarımında güç hattını pil ile enerjilendirmek için tasarımsal iyileştirme yapılabilir ancak burada seri hatlarda yer alan ground hatları ile kartın ground hatlarının ortaklanması gerekmektedir. Pil ile enerjilenen EceDuruDevice tasarımına güç hattı girişine anahtar eklenmesi yapılarak kablo sökme ve takma yapısından aç ve kapa yapısına evrilmiş olur. Bu tasarım iyileştirmesinde EceDuruDevice V1.0.00 da tasarımda yer alan 2 numaralı seri haberleşme hattı için kullanılan TTL yapısının gücünün bağlanılan cihazdan gelmesi durumu devam ettiği için bu hatta ait entegrede güç var olmaya devam edebilir. Bu nedenle 2 numaralı seri haberleşme hattında yer alan güç hattının da yeni tasarım kart üzerinden olması düşünülebilir.

Tez çalışmasının çıktıları EceDuruDevice ve EceDuruApp yazılımları prototip çalışmalardır. Bu çalışmalar ticari ürünleşme yapısına gidilmesi durumunda ürünü talep eden ticari kuruluşun özel gereksinimlerini karşılayabilecek şekilde yazılım, donanım ve mekanik tasarım açısında güncellemeye tabii tutulabilir. Bu sayede EceDuruDevice ve EceDuruApp ürünlerinin geliştirilmesi ve farklılaştırılması ile yeni ürünler oluşturulabilir. Bu ürünlerin ismi, amacı, çalışma senaryosu, görünümü yada malzemeleri değiştirilmiş olsa bile temel olarak baz aldığı mentalite EceDuruDevice ve EceDuruApp tarafından alınacaktır. Bu nedenle bu proje çalışmasına çekirdek proje demek çok da yanlış bir tanımlama olmaz. Bu kapsamda hazırlanan yeni ürünler ile ürün katalođu oluşturularak son kullanıcının ihtiyacı doğrudusunda ilgili ürüne ulaşması sağlanabilir.



KAYNAKLAR DİZİNİ

Akca, Mustafa Ali (2013), Kablosuz algılayıcı ağlar ile mera hayvancılığı üzerine bir uygulama, phdthesis. Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü.

Akın, Semavi (2009), Elektrik enerji kalitesi olayları için gömülü sistem uygulamaları, mathesis. Anadolu Üniversitesi.

Alégroth, Emil ve Feldt, Robert (2013), Transitioning manual system test suites to automated testing: An industrial case study, *2013 IEEE Sixth International Conference on Software Testing, Verification and Validation*. IEEE, 56–65.

Arıcı, İsmail ve Tenruh, Mahmut (2014), Labview Tabanlı Bir Elektronik Deney Seti Geliştirilmesi, *Akademik Bilişim*.

Atalık, T (2007), İnternet üzerinden haberleşen güç kalitesi monitör cihazı donanım tasarımı, phdthesis. Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

Ateş, Burcu ve Şaykol, Ediz (2015), KABLOSUZ ALGILAYICI AĞLARDA BULUT BİLİŞİM İLE SERVİS KALİTESİNİN ARTIRILMASI, *Beykent Üniversitesi Fen ve Mühendislik Bilimleri Dergisi*.

Avcıoğlu, Alper (2015), Bilgisayar Tabanlı Sistemlerde Test Otomatizasyonunun Tasarlanması ve Gerçeklenmesi, phdthesis. Yüksek Lisans Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

Aykurt, Cankat (2017), Askeri Kara Araçlarında Test Amaçlı Hızlı Kullanıcı Arayüzü Yazılımı Geliştirme, *11. Ulusal Yazılım Mühendisliği Sempozyumu (UYMS'17)*, pp. 217-225, Alanya, Türkiye 2017.

KAYNAKLAR DİZİNİ (devam)

Baeg, Seung-Ho, Park, Jae-Han, Koh, Jaehan, Park, Kyung-Wook ve Baeg, Moon-Hong (2007), Building a smart home environment for service robots based on RFID and sensor networks, *2007 International Conference on Control, Automation and Systems*. IEEE, 1078–1082.

Barroca, Norberto, Borges, Luís M, Velez, Fernando J, Monteiro, Filipe, Gorski, Marcin ve Castro-Gomes, João (2013), Wireless sensor networks for temperature and humidity monitoring within concrete structures, *Construction and Building Materials* 40, 1156–1166.

Bayır, R. ve Elen, A (2009), BİLGİSAYAR DESTEKLİ OTOMOTİV TEST STANDININ İNTERNET ÜZERİNDEN UZAKTAN EĞİTİM AMAÇLI KULLANIMI İÇİN KULLANICI ARAYÜZLÜ YAZILIM TASARIMI, *5. Uluslararası İleri Teknolojiler Sempozyumu, 2007-2012*.

Benzer, Volkan (2013), Marketlerde akıllı veri toplama ve analizi, mathesis. Maltepe Üniversitesi, Fen Bilimleri Enstitüsü.

Bora, Gaurav, Bora, Saurabh, Singh, Shivendra ve Arsalan, Sheikh Mohamad (2014), OSI reference model: An overview, *International Journal of Computer Trends and Technology* 7.4, 214–218.

Borjesson, Emil ve Feldt, Robert (2012), Automated system testing using visual gui testing tools: A comparative study in industry, *2012 IEEE Fifth International Conference on Software Testing, Verification and Validation*. IEEE, 350–359.

KAYNAKLAR DİZİNİ (devam)

- Burunkaya, Mustafa, Yıldız, Sadık ve Karataş, İsmail (2011), Öğrenci proje çalışmalarında kullanılabilir düşük maliyetli ve genel amaçlı bir veri toplama sisteminin (VTS) gerçekleştirilmesi, *Bilişim Teknolojileri Dergisi* 4.3, 21–28.
- Chae, MJ, Yoo, HS, Kim, JY ve Cho, MY (2012), Development of a wireless sensor network system for suspension bridge health monitoring, *Automation in Construction* 21, 237–252.
- Choudhury, Sushabhan, Kuchhal, Piyush, Singh, Rajesh, vd. (2015), ZigBee and Bluetooth network based sensory data acquisition system, *Procedia Computer Science* 48, 367–372.
- Cristaldi, Loredana, Faifer, Marco ve Lazzaroni, Massimo (2016), A cooperative monitoring and diagnostic architecture for PV systems, *2016 IEEE Sensors Applications Symposium (SAS)*. IEEE, 1–6.
- Çavuşoğlu, İ ve Kırmızı, F (2010), *Seri Port İle Haberleşebilen Uzaktan Kumandalı Kameralı Araç*.
- Çayiroğlu, İbrahim ve Şimşir, Mehmet (2008), PIC ve Step Motorla Sürülen Bir Mobil Robotun Uzaktan Kamera Sistemi ile Kontrolü, *Erciyes Üniversitesi Fen Bilimleri Enstitüsü Fen Bilimleri Dergisi* 24.1, 1–16.
- Çınar, Süleyman (2014), Mobil android ortamında parmak izi tanıma ve kimlik doğrulama sisteminin geliştirilmesi, phdthesis. Fen Bilimleri Enstitüsü.

KAYNAKLAR DİZİNİ (devam)

Demir, M (2006), Elektrik Sayaçlarının GPRS İle Uzaktan Okunması, *Afyon Kocatepe Üniversitesi Fen Bilimleri Enstitüsü Elektrik Eğitimi Anabilim Dalı Yüksek Lisans Tezi, Afyon, s85.(Yayınlanmamış).*

Divan, Deepak, Luckjiff, Glen, Brumsickle, William, Freeborg, John ve Bhadkamkar, Atul (2002), I-grid TM: infrastructure for nationwide real-time power monitoring, *Conference Record of the 2002 IEEE Industry Applications Conference. 37th IAS Annual Meeting (Cat. No. 02CH37344).* IEEE, 1740–1745.

Dustin, Elfriede, Garrett, Thom ve Gauf, Bernie (2009), *Implementing automated software testing: How to save time and lower costs while raising quality.* Pearson Education.

Erkan, Murat (2007), Bilgisayar destekli veri toplama sisteminin doküman makinelerinde uygulaması, mathesis. Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Bursa.

Erkol, Hüseyin Oktay ve Demirel, Hüseyin (2013), FPGA ile Seri Port Tasarımı ve Uygulaması, *Otomatik Kontrol Ulusal Toplantısı, TOK2013, 26-28 Eylül 2013, Malatya.*

Ertugrul, Nesimi (2002), *LabVIEW for electric circuits, machines, drives, and laboratories.* Prentice Hall Professional.

Fang, Yi-yuan ve Chen, Xue-jun (2011), Design and simulation of UART serial communication module based on VHDL, *2011 3rd International Workshop on Intelligent Systems and Applications.* IEEE, 1–4.

KAYNAKLAR DİZİNİ (devam)

Fongjun, Theerapong, Tantaworrasilp, Apicit, Kwansud, Phanuphan, Bunnun, Pished ve Theeraworn, Chonlada (2011), Automatic multi channel serial I/O interface using FPGA, *SICE Annual Conference 2011*. IEEE, 864–867.

Fuentes, M, Vivar, M, Burgos, JM, Aguilera, J ve Vacas, JA (2014), Design of an accurate, low-cost autonomous data logger for PV system monitoring using Arduino™ that complies with IEC standards, *Solar Energy materials and solar cells* 130, 529–543.

Gajos, Krzysztof Z, Czerwinski, Mary, Tan, Desney S ve Weld, Daniel S (2006), Exploring the design space for adaptive graphical user interfaces, *Proceedings of the working conference on Advanced visual interfaces*, 201–208.

Gevher, Mustafa, Tekkalmaz, Metin, Yamanyar, Hakan ve Bilgin, Omer (2016), Genel Amaçlı Haberleşme Arayüzü Simulator Yazılımı., *UYMS'16*, 295–301.

Goucher, Adam ve Riley, Tim (2009), *Beautiful Testing: Leading Professionals Reveal How They Improve Software.* ” O'Reilly Media, Inc.”.

Gökozan, Hayrettin ve Taşkın, Sezai (2012), Çok Fonksiyonlu Güç Kalitesi İzleme ve Veri Toplama Sistemi Multifunctional Power Quality Monitoring and Data Acquisition System, *ELECO '2012 Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu*, 29 Kasım - 01 Aralık 2012, pp. 784-787, Bursa.

Grossi, Dennis R (2006), Aviation recorder overview, *International Symposium On Transportation Recorders, Arlington, Virginia.*

KAYNAKLAR DİZİNİ (devam)

Güçlü, Bilal, BAYRAKÇI, Hilmi Cenk ve Süzen, Ahmet Ali (2018), ELEKTRİK SAYAÇLARININ UZAKTAN OKUNMASI VE ENERJİ ANALİZİ, *Uluslararası Teknolojik Bilimler Dergisi* 10.3, 27–39.

Gültekin, Akif Gökhan (2005), Gömülü Linux İle Dağıtık Veri Toplama Sistemi, phdthesis. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Güner, Yüksel (2005), LabVIEW programı ile veri toplama, veri işleme ve veri izlemenin e-öğrenme olarak hazırlanması, phdthesis. Marmara Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

Güngör, Okan (2015), Kaçak Elektrik Kullanımının GSM Aracılığıyla Takibi/Follow the Electricity Leakes Through the GSM Signals, *EMO Bilimsel Dergi* 4.8, 29–34.

Ha, Quang Phuc, Metia, Santanu ve Phung, Manh Duong (2020), Sensing data fusion for enhanced indoor air quality monitoring, *IEEE Sensors Journal* 20.8, 4430–4441.

Hardalaç, Fırat, Polat, Hüseyin, Barışçı, Necaattin ve Ergün, Uçman (2002), Web Based Telemedicine Remote Auscultation System on the Internet Turkish Journal of Telecommunications, *Turkish Journal of Telecommunications*, cilt.1, ss.33-39.

Ibrahim, Dogan (2008), Pic Microcontroller Projects in C, *Basic to Advanced 2*.

Ilyas, Mohammad ve Mahgoub, Imad (2004), *Handbook of sensor networks: compact wireless and wired sensing systems*. CRC press.

KAYNAKLAR DİZİNİ (devam)

Inner, Burak (2017), Data monitoring system for solar panels with Bluetooth, *2017 25th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 1–4.

İnan, Seyit Ahmet ve Çakmak, Seyfettin (2008), PIC Tabanlı Rs485 Ölçme ve Kontrol Sisteminin Tasarımı ve Uygulaması, *VII. Ulusal Ölçümbilim Kongresi*, 559-567, İzmir, 2008.

Jie, Hu, Changlin, Liu, Fuwu, Yan ve Zeyang, Zhang (2015), The development of vehicle diagnostic system based on Android platform, *2015 International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 42–47.

Jo, JunHo, Jo, ByungWan, Kim, JungHoon, Kim, SungJun ve Han, WoonYong (2020), Development of an iot-based indoor air quality monitoring platform, *Journal of Sensors* 2020.

Karabulut, Mustafa Mehmet (2010), Parmak izi tanıma temelli gerçek zamanlı öğrenci yoklama sistemi otomasyonu/Fingerprint recognition based online student attendance automation system design, phdthesis. Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.

Karapınar K., Kaya E (2016), Gömülü Sistemlerin Seri Haberleşmesi, phdthesis. Karabük Üniversitesi, Fen Bilimleri Enstitüsü, Karabük.

Karasoy, Burcu ve Çınar, Soner (2014), Dağıtık Sistemler için Haberleşme Otomasyon Ara Katmanı: ULAK., *UYMS*.

KAYNAKLAR DİZİNİ (devam)

Karnouskos, Stamatias ve Izmaylova, Anastasia (2009), Simulation of web service enabled smart meters in an event-based infrastructure, *2009 7th IEEE International Conference on Industrial Informatics*. IEEE, 125–130.

Kayıkçıođlu, Temel ve Gangal, Ali (2017), Parmak İzi Okumaya Dayalı Hızlı Yoklama Sistemi, phdthesis. Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Trabzon.

Kelebekler, Ersoy (2019), Nesnelerin İnterneti Tabanlı Meteorolojik Veri Takip Sistemi, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi* 7.1, 650–663.

Köksal, Resul (2014), Yapılandırılabilir donanım üzerindeki (FPGA) mikrobilgisayar sistemleri için analog haberleşme arayüz tasarımı, phdthesis. Bozok Üniversitesi, Fen Bilimleri Enstitüsü, Yozgat.

Kürklü, Ahmet ve Çağlayan, Nuri (2007), Mikrodenetleyici ve Radyo Frekansı Kullanılarak Alternatif İklim Kontrol Sisteminin Geliştirilmesine Yönelik Bir Çalışma, *Akdeniz Üniversitesi Ziraat Fakültesi Dergisi* 20.2, 229–239.

Lee, Kyung Chang ve Lee, Hong-Hee (2004), Network-based fire-detection system via controller area network for smart home automation, *IEEE Transactions on Consumer Electronics* 50.4, 1093–1100.

Lee, Ringo PK, Lai, LL ve Tse, Norman (2002), A Web-based multi-channel power quality monitoring system for a large network, *Proceedings of International Conference on Power System Management and Control*, 2002 p. 112 – 117.

KAYNAKLAR DİZİNİ (devam)

- Lin, Yuan-Hsiang, Jan, I-Chien, Ko, PC-I, Chen, Yen-Yu, Wong, Jau-Min ve Jan, Gwo-Jen (2004), A wireless PDA-based physiological monitoring system for patient transport, *IEEE Transactions on information technology in biomedicine* 8.4, 439–447.
- López, M, Martinez, S, Gomez, JM, Herms, A, Tort, L, Bausells, J ve Errachid, A (2009), Wireless monitoring of the pH, NH₄⁺ and temperature in a fish farm, *Procedia Chemistry* 1.1, 445–448.
- Marques, Gonçalo, Ferreira, Cristina Roque ve Pitarma, Rui (2019), Indoor air quality assessment using a CO₂ monitoring system based on internet of things, *Journal of medical systems* 43.3, 1–10.
- Muller, Tomas ve Friedenber, Debra (2011), Certified tester foundation level syllabus, *Journal of International Software Testing Qualifications Board* 36.
- Myers, Brad A ve Rosson, Mary Beth (1992), Survey on user interface programming, *Proceedings of the SIGCHI conference on Human factors in computing systems*, 195–202.
- Natale, Marco Di (2008), Understanding and using the Controller Area network, *Handout of a lecture at UC Berkeley available at <http://inst.eecs.berkeley.edu/~ee249/fa08>*.
- Niu, Huibo, Wang, Weijiang ve Liu, Zhenjuan (2012), High Speed Synchronous Serial Port and Research on Glitch Based on the DSP and FPGA, *2012 International Conference on Control Engineering and Communication Technology*. IEEE, 448–450.

KAYNAKLAR DİZİNİ (devam)

Nunes, Renato JC ve Delgado, Jose CM (2000), An Internet application for home automation, *2000 10th Mediterranean Electrotechnical Conference. Information Technology and Electrotechnology for the Mediterranean Countries. Proceedings. MeleCon 2000 (Cat. No. 00CH37099)*. IEEE, 298–301.

Pereira, Renata IS, Dupont, Ivonne M, Carvalho, Paulo CM ve Jucá, Sandro CS (2018), IoT embedded linux system based on Raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant, *Measurement* 114, 286–297.

Senem, Murat ve Keleş, Ali (2017), Gömülü ve Gerçek Zamanlı Yazılımlarda Arayüz Giriş-Çıkış İzleyici, *UYMS*.

Seyfettin, VADI, GULER, Naki ve BAYINDIR, Ramazan (2014), Endüstriyel Alanlarda Kullanılan Veri İletim Tekniklerinin Karşılaştırılması, *Gazi Üniversitesi Fen Bilimleri Dergisi Part C: Tasarım ve Teknoloji* 2.1, 181–188.

Shariff, Farihah, Abd Rahim, Nasrudin ve Hew, Wooi Ping (2015), Zigbee-based data acquisition system for online monitoring of grid-connected photovoltaic system, *Expert Systems with Applications* 42.3, 1730–1742.

Sung, Wen-Tsai, Hsiao, Sung-Jung ve Shih, Jing-An (2019), Construction of indoor thermal comfort environmental monitoring system based on the IoT architecture, *Journal of Sensors* 2019.

Süzer, Emre Sami (2006), Uzaktan sayaç okuma teknikleri ve modbus-rtu, iec 61107 mod c protokolleri ile örnek yazılım, phdthesis. İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

KAYNAKLAR DİZİNİ (devam)

Şahin, H (2007), Elektronik Sayaçlarda Kullanılmak Üzere Uzaktan Gprs Modül İle Elektronik Elektrik Sayacı Okuma Uygulaması, *Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.*

Şimşek, Mehmet Ali ve Taşdelen, Kubilay (2015), Denetleyici Alan Ağı Üzerinden Sensör Verilerinin İzlenmesi, *Akademik Bilişim*, 4–6.

Tokel, Çağlar (2009), Dört eksenli RC servo motor tahrikli bir robot manipülatörü tasarımı ve uygulaması, phdthesis. DEÜ Fen Bilimleri Enstitüsü.

Tolga, ÖZER, KIVRAK, Sinan, Yüksel, OĞUZ ve KELEK, Muhammed Mustafa (2020), STM32f103C8 Mikroişlemcisi Kullanılarak CAN Haberleşme Protokolü Tabanlı Veri Toplama Sistemi Uygulaması, *Afyon Kocatepe Üniversitesi Uluslararası Mühendislik Teknolojileri ve Uygulamalı Bilimler Dergisi* 3.1, 27–31.

Topaloğlu, Ergün Yazar (2004), KAN GAZI ANALİZ CİHAZLARINDAN AMELİYATHANELERE BİLGİSAYAR KONTROLLÜ VERİ İLETİMİ, phdthesis. ANKARA ÜNİVERSİTESİ FEN BİLİMLERİ.

Topaloğlu, Nurettin (2012), İNTERNET ÜZERİNDEN UZAKTAN WEB KAMERA KONTROLÜNÜN TASARIMI VE UYGULAMASI, *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi* 27.2.

Tunç, Süleyman (2007), Analog sismometrelerden sayısal veri transferi, mathesis. Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü.

KAYNAKLAR DİZİNİ (devam)

Türker, GF (2010), Kalp Atışının Sezilmesi ve Alınan Sinyalin Kablosuz Algılayıcı Ağlar ile İletimi, *Muğla Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Muğla.*

Umang, Patel ve Mitul, Modi (2015), A Review on Smart Meter System, *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering* 3.12, 70–73.

Uyar, Murat (2013), Kablosuz veri toplama yöntemi ile mekanik sistemlerin durum izlemesi, phdthesis. DEÜ Fen Bilimleri Enstitüsü.

Uzun, Hakan Sefa (2016), Kablosuz sensör ağ uygulamaları için düğüm modülü tasarımı, mathesis. Maltepe Üniversitesi, Fen Bilimleri Enstitüsü.

Üçgün, Hakan, GÖMBECİ, Fatmanur, YÜZGEÇ, Uğur ve YALÇIN, Nesibe (2020), IoT Tabanlı Platform ile Gerçek Zamanlı İç Ortam Hava Kalitesi İzleme Sistemi, *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi* 7.1, 370–381.

Watterson, Conal ve Heffernan, Donal (2007), Runtime verification and monitoring of embedded systems, *IET software* 1.5, 172–179.

Watthanawisuth, Natthapol, Lomas, Tanom ve Tuantranont, Adisorn (2012), Wireless black box using MEMS accelerometer and GPS tracking for accidental monitoring of vehicles, *Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics*. IEEE, 847–850.

KAYNAKLAR DİZİNİ (devam)

Whittaker, James A, Arbon, Jason ve Carollo, Jeff (2012), *How Google tests software*. Addison-Wesley.

Xu, Wanfang, Xu, Gang, Xi, Zhijiang ve Zhang, Chuanyong (2012), Distributed power quality monitoring system based on EtherCAT, *2012 China International Conference on Electricity Distribution*. IEEE, 1–5.

Yıldırım, Özal, ERİŞTİ, Hüseyin ve DEMİR, Yakup (2018), Yeni Bir Çevrimiçi Elektrik Enerji Kalitesi İzleme Cihazı, *Fırat Üniversitesi Mühendislik Bilimleri Dergisi* 30.3, 31–36.

Yılmaz, A Serdar ve Görmemiş, Murat (2011), ENERJİ ANALİZÖRLERİ İÇİN GELİŞTİRİLEN DELPHI TABANLI BİR ENERJİ İZLEME YAZILIMI, *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi* 12.3, 359–368.

Yılmaz, Arif Kamil ve Acar, Devrim Barış (2017), IoT Cihaz Verileri İçin Gerçek Zamanlı ve Ölçeklenebilir Büyük Veri Mimarisi, *UYMS*.

Yolcu, Cihan ve Şahan, Burak (2017), Araç İçi Durum Takibi ve Arıza Tespit Cihazı Vehicle Status Monitoring and Diagnostic, *EMO*.

Zhao, Liang, Wu, Wenyan ve Li, Shengming (2019), Design and implementation of an iot-based indoor air quality detector with multiple communication interfaces, *IEEE Internet of Things Journal* 6.6, 9621–9632.