**T.C.**

**FIRAT ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**TELEMETRİ SİSTEMİ**

**BİTİRME TEZİ**

**HAZIRLAYAN**

**Kazım BAŞLAK**

**Harun AĞCA**

**TEZ DANIŞMANI**

**Prof. Dr. Mehmet KARAKÖSE**

**ELAZIĞ -2020**

**T.C.**

**FIRAT ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**TELEMETRİ SİSTEMİ**

**BİTİRME TEZİ**

**Kazım BAŞLAK**

**Harun AĞCA**

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Bu bitirme ,………. tarihinde, aşağıda belirtilen jüri tarafından oybirliği/ oyçokluğu ile başarılı/ başarısız olarak değerlendirilmiştir.

**Danışman Üye Üye**

**Prof. Dr. Mehmet KARAKÖSE**

**ELAZIĞ -2020**

# ÖZGÜNLÜK BİLDİRİMİ

Bu çalışma başka kaynaklardan yapılan tüm alıntıların, ilgili kaynaklar referans gösterilerek açıkça belirtildiğini, alıntılar dışımdaki bölümlerin, özellikle projenin ana konusunu oluşturan teorik çalışmaların ve yazılım / donanımın tarafımızdan yapıldığını bildiririz.

Fırat Üniversitesi

Bilgisayar Mühendisliği

23119 ELAZIĞ

…./…./…. …./…./…./

Kazım BAŞLAK Harun AĞCA

# BENZERLİK BİLDİRİMİ

# 

# TEŞEKKÜR

Lisans eğitim boyunca mühendislik altyapısını en iyi şekilde bizlere aktaran değerli hocamız, Sayın Prof. Dr. Yetkin Tatar’a, tez çalışmamız boyunca bize danışmanlık yapan Sayın Prof. Dr. Mehmet KARAKÖSE hocamıza sonsuz teşekkürlerimizi sunarız.

Kazım BAŞLAK Harun AĞCA

# İÇİNDEKİLER

[ÖZGÜNLÜK BİLDİRİMİ I](#_Toc45274811)

[BENZERLİK BİLDİRİMİ II](#_Toc45274812)

[TEŞEKKÜR III](#_Toc45274813)

[İÇİNDEKİLER IV](#_Toc45274814)

[ŞEKİLLER LİSTESİ V](#_Toc45274815)

[TABLOLAR LİSTESİ VI](#_Toc45274816)

[KISALTMALAR LİSTESİ VII](#_Toc45274817)

[ÖZET VIII](#_Toc45274818)

[ABSTRACT IX](#_Toc45274819)

[1.GİRİŞ 1](#_Toc45274820)

[2. KONTROL VE İLETİM MERKEZİ 1](#_Toc45274821)

[2.1. Motor Hız Kontrolü 2](#_Toc45274822)

[2.1.1. Hall Effect 5](#_Toc45274823)

[2.2. Cihazlar Arası Haberleşme 5](#_Toc45274824)

[2.2.1. İletişim Protokolleri 6](#_Toc45274825)

[2.2.1.1. I2C (Inter-Integrated Circuit) 7](#_Toc45274826)

[2.2.1.2. UART (Universal Asynchronous Receiver Transmitter) 7](#_Toc45274827)

[2.2.2. Kablosuz Haberleşme 8](#_Toc45274828)

[2.2.2.1. ISM (Industrial Scientific Medical) Bandı 8](#_Toc45274829)

[2.2.2.2. İletişimde Güvenlik ve Veri Bütünlüğü 8](#_Toc45274830)

[2.2.2.3. HC-12 Modülü 11](#_Toc45274831)

[2.3. Sensörler 14](#_Toc45274832)

[2.4. Batarya 15](#_Toc45274833)

[2.5. Gösterim 16](#_Toc45274834)

[3. İZLEME MERKEZİ 18](#_Toc45274835)

[3.1 Genel Yapı 18](#_Toc45274836)

[3.2 Alıcı Birimi 19](#_Toc45274837)

[3.3 İzleme Birimi 21](#_Toc45274838)

[3.3.1 İzleme Biriminde Kullanılacak Teknolojiler 22](#_Toc45274839)

[3.3.2 SOLID Prensipleri ve İzleme Birimi Mimarisi 24](#_Toc45274840)

[3.3.3 Veri Tabanı Tasarımı ve Entity Framework 26](#_Toc45274841)

[3.3.4 İzleme Biriminin Çalışma Sistemi 27](#_Toc45274842)

[3.3.5 Kullanıcı Arayüzü 28](#_Toc45274843)

[SONUÇ 30](#_Toc45274844)

[KAYNAKÇA 31](#_Toc45274845)

[ÖZGEÇMİŞ 1 32](#_Toc45274846)

[ÖZGEÇMİŞ 2 33](#_Toc45274847)

# ŞEKİLLER **LİSTESİ**

[**Şekil 1.0** Telemetri şeması 1](#_Toc45272417)

[**Şekil 2.0** Hız hesap algoritması. 2](#_Toc45272418)

[**Şekil 2. 1** Sayıcı devre 3](#_Toc45272234)

[**Şekil 2. 2** Hall etkisi 5](#_Toc45272235)

[**Şekil 2. 3** Senkron iletişim 6](#_Toc45272236)

[**Şekil 2. 4** Asenkron iletişim 6](#_Toc45272237)

[**Şekil 2. 5** I2C protokolü 7](#_Toc45272238)

[**Şekil 2. 6** UART protokolü 8](#_Toc45272239)

[**Şekil 2. 7** Encoder algoritması 10](#_Toc45272240)

[**Şekil 2. 8** Decoder algoritması 11](#_Toc45272241)

[**Şekil 2. 9** ISM band aralığı 11](#_Toc45272242)

[**Şekil 2. 10** HC-12 bağlantı şeması 14](#_Toc45272243)

[**Şekil 2. 11** LM35 bağlantı şeması 15](#_Toc45272244)

[**Şekil 2. 12** LCD bilgi ekranı 1 16](#_Toc45272245)

[**Şekil 2. 13** LCD bilgi ekranı 2 17](#_Toc45272246)

[**Şekil 2. 14** LCD bilgi ekranı 3 17](#_Toc45272247)

[**Şekil 3. 1** Telemetri İzleme Merkezi 19](#_Toc45274798)

[**Şekil 3. 2** SignalR İletişim Seçim Sırası 24](#_Toc45274799)

[**Şekil 3. 3** İzleme Biriminin Mimarisi 25](#_Toc45274800)

[**Şekil 3. 4** Veri Tabanı ER Diyagramı 27](#_Toc45274801)

[**Şekil 3. 5** Kullanıcı Arayüzü 29](#_Toc45274802)

# TABLOLAR LİSTESİ

[**Tablo 1** ISM bandı 8](#_Toc45272403)

[**Tablo 2.1** ISM bandı 8](#_Toc45274495)

[**Tablo 2.2** Baud oranları 12](#_Toc45274496)

[**Tablo 2.3** Baud oranı ve alıcı hassasiyeti ilişkisi 12](#_Toc45274497)

[**Tablo 2.4** HC-12 modları 12](#_Toc45274498)

[**Tablo 2.5** AT komutları 13](#_Toc45274499)

[**Tablo 3.1** 21](#_Toc45274773)

[**Tablo 3.2** 21](#_Toc45274774)

# 

# KISALTMALAR LİSTESİ

**RF** Radyo Frekansı

**ACK** Acknowledgement

**UART** Universal Asynchronous Receiver Transmitter

**USART** Universal Synchronous Asynchronous Receiver Transmitter

**TX** Transmitter

**RX** Receiver

**GND** Ground

**ISM** Industrial Scientific Medical

**ASP** Active Server Pages, Aktif Sunucu Sayfaları

**MVC** Model View Controller

**VB** Visual Basic

**ORM** Object Relational Mapping

**USB** Universal Serial Bus

# ÖZET

Araçlarda, verilerin toplanıp işlenmesi, kullanıcıya gösterilmesi gibi işlemler, birer ihtiyaç halini almıştır. Ayrıca araç dışında, bir cihaza verilerin nakledilmesi ve gerekli analizlerin, uzak cihazda yapılması işlemleri oldukça yaygınlaşmaktadır. Bu faaliyetlerin yerine getirilebilmesi için telemetri sistemine ihtiyaç duyulmaktadır. Telemetri genel olarak, verilerin bir yerden bir başka yere aktarılmasından sorumlu cihaz veya cihazların toplamıdır. Bu proje, bir araç içerisinde, verilerin toplanması ve aktarılması işlemlerindeki tüm problemlerin çözülerek, faal bir sistemin kurulmasını amaçlamaktadır.

# ABSTRACT

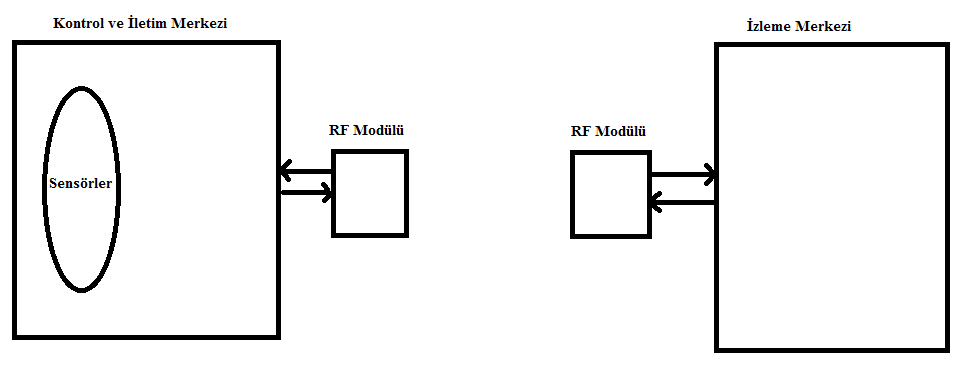
Operations such as data collection, processing and display to the user have become a necessity. In addition, outside of the vehicle, the process of transferring data to a device and performing the necessary analysis on the remote device is very common. Telemetry system is needed to carry out these activities. Telemetry is generally the sum of the device or devices responsible for transferring data from one place to another. This project aims to set up an active system in a vehicle by solving all problems in data collection and transfer.

# 1.GİRİŞ

Veri, 21. yüzyılın en büyük güç çarpanlarından biridir. Tüm dünyada sensörler hızla yayılmakta, cihazların olmazsa olmazları haline gelmektedir. Elektronik araçların duyu organları olan sensörler, dış ortamdan istenilen bilgilerin alınabilmesi için olmazsa olmazlardır. Verilerin toplanması, gerektiğinde işlenmesi ve iletilmesinden sorumlu olan telemetri sistemine ihtiyaç, her geçen gün artmaktadır. Bu proje bünyesi içerisinde, bir araç için telemetri sistemi inşa etmek amaçlanmıştır.

Araç bünyesinde bulunan telemetri sisteminin görevleri, motor hızı hesaplanması, sıcaklık ölçümü, batarya anlık gerilim, akım değerleri ölçümü, batarya kapasitesi verilerinin alınması ve tüm bu bilgilerin araçta bulunan ekrana, aynı zamanda uzak bir cihaza aktarılmasını sağlamaktır.

Uzak cihazın görevi ise verilerin gösterilmesi, veritbanında kayıtların tutulması ve gerektiğinde analizlerin yapılarak bir çıktı üretilmesini sağlamaktır. Sistemin genel gösterimi şekil 1.0’da gösterilmiştir.



**Şekil 1.0** Telemetri şeması

# 2. KONTROL VE İLETİM MERKEZİ

Telemetri sisteminin iki aşamasından biri olan kontrol ve iletim merkezi; batarya kapasitesi, anlık batarya akım ve gerilimi, araç içerisindeki diğer sensör verilerini almak, işleyerek anlamlı hale getirmek ve son olarak iletilmesi gereken cihazlara iletmekle sorumludur.

Bu bölümde, motor hız kontrolü, radyo sinyalleri, batarya sistemi, grafik ekrandan ve geliştirilen algoritmalardan bahsedilecektir.

## 2.1. Motor Hız Kontrolü

Elektrikli aracımızda kullandığımız motor, 54 adımdan oluşmaktadır. Her adım için bir darbe üretecek olan motor, tam bir dönüş için 54 kare dalga üretmektedir. Burada yapılacak olan temel işlev, mikrodenetleyiciye gelen sinyallerin aralarında geçen süre üzerinden anlık hız ölçümü yapılmasıdır.

Interrupt Pin

Adım Uzunluğu / Geçen Süre

k

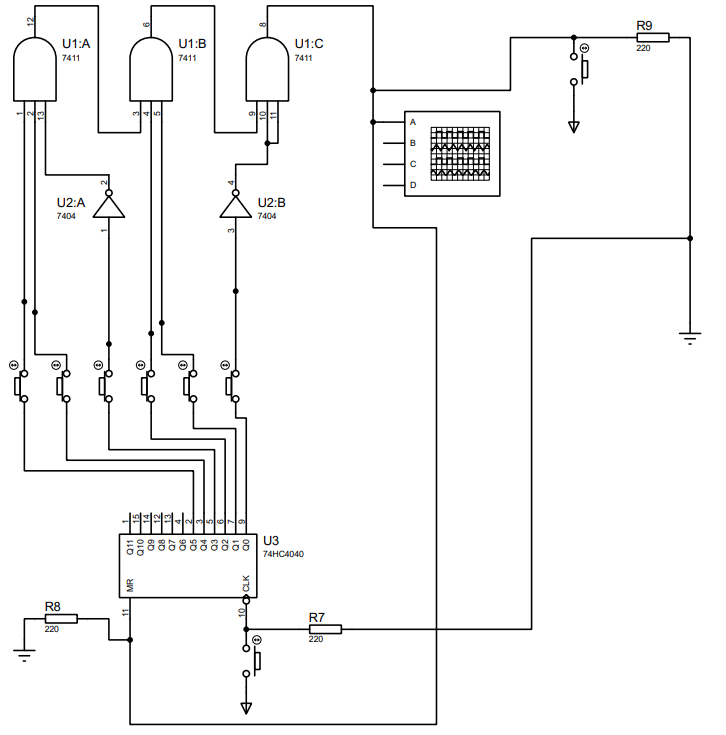
Birim Dönüşümleri

k

Sonuç

**Şekil 2.0** Hız hesap algoritması.

Tabi her bir kare dalgada hesaplama yapmak, işlem gücü açısından maliyetlidir. İşlem maliyetini azaltmak için, sayıcı entegre(HC4040) kullanılabilir. QuineMcClusky algoritması kullanılarak, motordan gelen 54 darbenin bir darbeye düşürülmesi için, gerekli devre aşağıda verilmiştir. Şekil 2.1’deki devre şemasında, sayıcı entegre her darbeye karşılık bir artmakta ve 11 bitlik çıkışındaki lojik kapılar, 54 darbenin bire düşürülmesini sağlamaktadır. Lojik kapıların oluşumu, a en anlamlı bit olmak üzere, “ a.b.c'.d.e.f ' ” şeklinde olmalıdır.



**Şekil 2. 1** Sayıcı devre

Bu projede hassasiyet, işlem maliyetinden daha önemli olduğundan ilk yöntem tercih edilmiştir.

Tekerin ne kadar mesafe ilerlediğini hesaplamak için, tabi ki teker yarıçapı gibi temel bilgiler bilinmelidir. Anlık hız hesaplamaları için gerekli denklemler aşağıda verilmiştir.

Yukarıdaki denklemler uygulandığında, motor yavaş dönerken sorun olmamakla birlikte, motor hızlandığında hesaplama sıklığı artacağından, mikrodenetleyici boş yere fazladan koşturulmuş olacaktır. Sorunun giderilmesi için, hesaplama sıklığının denetlenmesi gerekmektedir. Yani motor yavaşladığında da, hızlandığında da hesaplama sıklığı istenilen hassasiyette olmalıdır.

Sıklığın kontrol edilmesi için gerekli denklem aşağıda verilmiştir.

Hesaplama sıklığı, kaç kare dalga geldiğinde hesaplama yapılacağını göstermektedir. Örneğin, istenilen sıklık 200 ms verilirse, saniyede 5 defa hesaplama yapılır ve bu, hızın anlık görüntüsü için yeterlidir.

Bir sonraki aşama olan test aşamasında, hesaplamaların doğruluğunu teyit etmek için kare dalga üreterek sisteme sokmak ve çıktıları kontrol etmek gerekir. Bu aşamada üretilecek kare dalganın frekansı aşağıdaki denklemlere göre belirlenir.

1000: Metreye çevirme

100: santimetreye çevirme

60: Dakikaya çevirme

60: Saniyeye çevirme

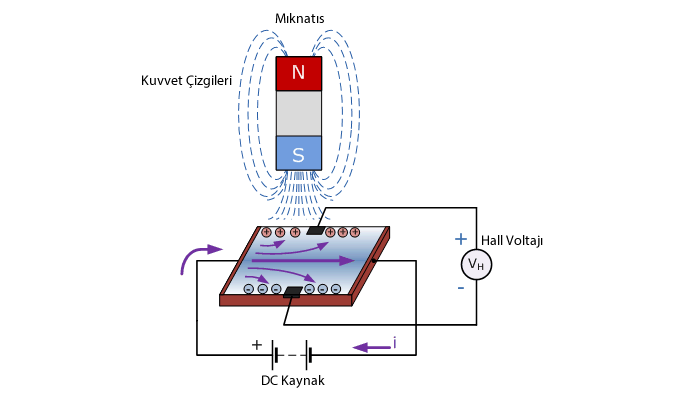
94.2: Tekerin çevresi

54: Motor sargı sayısı

Tabi mikrodenetleyicide, iş yükünden tasarruf etmek için denklem sadeleştirilerek uygulanmalıdır.

### 2.1.1. Hall Effect

Edwin Herbert Hall tarafından bulunan hall etkisi, iki ucu arasında akım akan bir iletken veya yarı iletken cisme dik olarak bir manyetik alan uygulandığında, cismin iki ucunda bir voltaj meydana gelmesidir. Şekil 2.2’de görsel olarak anlatılmıştır.



**Şekil 2. 2** Hall etkisi

Bu proje bünyesinde motor hızını ölçmek için hall effect sensörlerinden yararlanılmıştır. Hall effect sensörleri, bu gerilim farkını kontrol eder ve sayısal veya analog olarak çıkış verir.

## 2.2. Cihazlar Arası Haberleşme

Bu proje bünyesi içerisinde, kablolu ve kablosuz olmak üzere iki fiziksel ortamda, I2C ve UART iletişim protokolleri kullanılmıştır. Haberleşme hızı, mikrodenetleyiciye göre oldukça düşük olduğundan, en az byte bilgi ile iletişim kurulmaya çalışılmalıdır. Önemli bir diğer husus da, doğru(iletilmek istenen) verilerin hedefe ulaştığından emin olmaktır. Güvenlik en önemli başlıklardan bir tanesidir.

### 2.2.1. İletişim Protokolleri

İletişim protokolleri, mikrodenetleyicilerin kapasitesi, hangi protokolleri desteklediği, yapılacak işin gerektirdiği hız, kablo karmaşıklığı, fiziksel ortam gibi birçok faktör esas alınarak seçilir. Örneğin hızdan feragat edip kablo karmaşıklığı ve maliyetinden tasarruf etmek için araçlarda CAN BUS protokolü kullanılmaktadır.

İletişim senkron ve asenkron olmak üzere iki şekilde gerçekleştirilir. Senkron haberleşme, bir clock pulse sinyalini esas alarak, düşen veya yükselen kenar da veriyi alır. Asenkron haberleşmede ise, belirlenmiş baud rate değerine göre iki tarafta kendini ayarlar ve iletişimi gerçekleştirir. İletişim şeması aşağıda verilmiştir.

Data



Clock

Data



Clock

**Şekil 2. 3** Senkron iletişim

Data

Data

**Şekil 2. 4** Asenkron iletişim

Bu projede kullanılan I2C, senkron ve UART, asenkron bir iletişimdir.

#### 2.2.1.1. I2C (Inter-Integrated Circuit)

I2C senkron bir haberleşme protokolüdür. Toprak hattı dışında SCL(Serial Clock) ve SDA(Serial Data) hattı bulunmaktadır. Kısa mesafeler ve düşük hızlar için kullanılır. Master ve slave olmak üzere iki modu vardır. Her ağda bir master ve en az bir slave bulunmalıdır. Şekil 2.2’de de gösterildiği üzere cihazlar clock sinyaline göre veriyi alırlar ve bu sinyali master cihaz üretir. I2C’nin en temel özelliği, tek clock ve data hattıyla azami 127 farklı cihazla iletişimi mümkün kılmasıdır. Her cihaz, 7 bitlik bir id alır ve bu id üzerinden hangi cihazla iletişime geçileceği belirlenir.

VCC

GND

SCL

SDA

Slave 1

Slave 2

Master

**Şekil 2. 5** I2C protokolü

#### 2.2.1.2. UART (Universal Asynchronous Receiver Transmitter)

UART, asenkron bir iletişim protokolüdür. Toprak hattı dışında tx ve rx olmak üzere data hattı bulunmaktadır. Bağlantı şeması şekil 2.5’teki gibidir. Sadece, iki cihaz arasında baud oranı belirlenerek iletişim gerçekleştirilebilir.

Tx

Rx

GND

Tx

Rx

GND

**Şekil 2. 6** UART protokolü

### 2.2.2. Kablosuz Haberleşme

Kablosuz iletişim, en az iki cihaz arasında, farklı karakteristikteki sinyallerle sağlanabilir. Burada bahsedilecek olan sinyal türü, radyo sinyalleridir.

Elektromanyetik spektrum içerisinde yer alan radyo sinyalleri, 9 KHz ile 300 GHz frekans aralığındadır. Ancak bu aralıktaki her frekans kullanıma açık değildir. Frekanslar yetkili kurumlarca kişi ya da kurumlara tahsis edilirler. Bunun istisnası ISM(Industrial Scientific Medical) bandıdır. ISM bandının kullanımı, aynı ortamdaki ISM bandı kullanan farklı cihazların olması durumunda bazı sorunlar doğurmaktadır. Bu sorunlardan aşağıda bahsedilecektir.

#### 2.2.2.1. ISM (Industrial Scientific Medical) Bandı

Herkesin kullanımına açık olan ISM bandı, birçok cihaz tarafından kullanılmaktadır. IEEE 802.15.4 standardındaki duyarga ağlar, IEEE 802.11 protokolleri gibi protokolleri kullanan cihazlar da ISM bandını kullanmaktadır. ISM frekans aralıkları aşağıdaki tabloda verilmiştir.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Başlangıç Frekansı** | **Bitiş Frekansı** | **Kanal Sayısı** |
| **868 MHz** | **868 MHz** | **868.6 MHz** | **1** |
| **915 MHz** | **902 MHz** | **928 MHz** | **10** |
| **2.4 GHz** | **2.4 GHz** | **2.483 GHz** | **16** |

**Tablo 2.1** ISM bandı

#### 2.2.2.2. İletişimde Güvenlik ve Veri Bütünlüğü

Veri transferinde, mutlaka ortama göre çok veya az olmak üzere bilgi kaybı olmaktadır. Verinin doğru şekilde ve bozulmadan ulaştığından emin olunmalıdır. Kablosuz iletişimde, frekansların çakışması durumunda veriler bozulabilir ve hatta yanlış bilgi alınabilir. Örneğin aynı frekansta iki verici ve tek alıcılı bir sistemde, yüksek gerilimli vericinin gönderileri alıcıyı domine edecektir. Bu nedenle yazılım ve donanım unsurları kullanılarak, gerekli güvenlik önlemleri alınmalıdır.

Fiziksel olarak ortamda kullanılan frekanslar analiz edilmeli, kullanılmayan ve en az etki altındaki kanallar seçilmelidir. Bir frekansın, özellikle de ISM bandı içerisindeki bir frekansın, tek ortamda kullanılmıyor olması çok zor olduğundan, yanlış vericiden data almamak için gönderilen veriler bir şifreleme algoritmasından geçirilmelidir. Vericide encoder çalıştırılmalı ve alıcıda da decoder çalıştırılarak verinin istenilen gönderici tarafından gönderildiği tespit edilmelidir. İletişim bytelar halinde yapıldığından, her byteın güvenli şekilde ulaştığından emin olunmalıdır. Alıcı, veriyi aldıktan sonra bir ACK(Acknowledgement) gönderilmesi kayıpsız iletimi mümkün kılacaktır.

Bu proje bünyesinde, güvenlik ve veri bütünlüğünü sağlamak için, her gönderilen byte ikiye bölündü. Ağırlıklı 4 biti, daha önce belirlenen 0-15 arasında bir sayıya dönüştürüldü ve diğer 4 bit, verimizin 4 biti şeklinde ayarlandı. Gerçek verinin anlamlı 4 biti için verilen 4 bitlik özel kod ile, anlamsız 4 biti için verilen kod farlı olarak ayarlandı. Bu da alıcıda, verinin anlamlı kısmı mı, yoksa anlamsız kısmı mı alındı sorusunun cevabını vermektedir. Sonuç bir byte ham veri, iki byte ile ulaştırılmış oldu. Yani zaman maliyeti 2 katına çıkmış oldu. Vericiden çıkarak alıcıya ulaşan veri, decode işleminden geçtikten sonra, alıcı, vericiye bir ACK mesajı yollar. ACK, bu projede 255 olarak belirlenmiştir. Alıcıya ACK mesajı ulaşırsa, sonraki veri gönderilir, ulaşmazsa, belli bir süre beklendikten sonra aynı byte tekrar gönderilir. Algoritmanın akış şeması aşağıdaki gibidir.

ACK: 0xFF => 11111111

Anlamlı 4 bit kodu(0-15): 0xF => 1111

Anlamsız 4 bit kodu(0-15): 0xA => 1010

##### Gönderici

Gönderilen Byte

10110010

İlk Byte

11111011

İkinci Byte

11110010

Gönder

Gönder

Yeni Data Getir

Evet

Hayır

ACK Geldi mi?

**Şekil 2. 7** Encoder algoritması

##### Alıcı

Evet

Data doğru mu?

Hayır

Gelen Veri

11111011

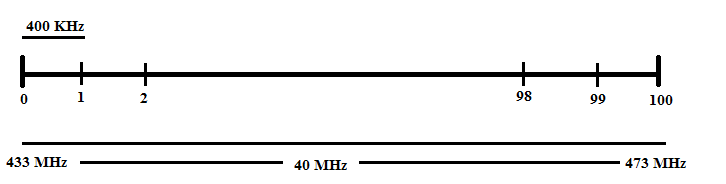
Dinle

ACK Gönder

**Şekil 2. 8** Decoder algoritması

#### 2.2.2.3. HC-12 Modülü

Hc-12 433 MHz ISM bandında çalışmaktadır. UART iletişim arayüzüne sahip olması mikrodenetleyici tarafında işleri kolaylaştırmaktadır. 433 MHz frekansta başlamak üzere, 400 KHz bant genişliğinde 473 MHz frekansa kadar 100 kanada iletişim kurma kabiliyetine sahiptir. Bitişik her beş kanal arasında çakışmalar ve iletişimde parazitler meydana gelmektedir. Bu nedenle iletişimde, 5 kanal atlamak suretiyle kanal seçimi yapılmalıdır.

**Şekil 2. 9** ISM band aralığı

* 5000 bps baud hızında 1000 m menzile ulaşmaktadır.
* Maksimum 100 mW (20dBm) iletim güce ulaşabilmekte ve 8 kademede ayarlanabilmektedir.

Modülün seri baud hızı ile havadaki hızları tablo 1.1’de verilmiştir.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Seri Port Baud Oranı** | **1,200** | **2,400** | **4,800** | **9,600** | **19,200** | **38,400** | **57,600** | **115,200** |
| **(bps)** |  |  |  |  |  |  |  |  |
| **Havadaki Baud Oranı** | **5,000bps** | | **15,000bps** | | **58,000bps** | | **236,000bps** | |
|  |  |  |  |  |  |  |  |  |

**Tablo 2.2** Baud oranları

Havadaki baud oranı değiştikçe alıcıda ki hassasiyette değişmektedir. Değerler tablo 1.2’de gösterilmiştir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Havadaki Baud Oranı** | **5,000bps** | **15,000bps** | **58,000bps** | **236,000bps** |
|  |  |  |  |  |
| **Kablosuz Alıcı** | **-117dBm** | **-112dBm** | **-107dBm** | **-100dBm** |
| **Hassasiyeti** |  |  |  |  |

**Tablo 2.3** Baud oranı ve alıcı hassasiyeti ilişkisi

HC-12’nin 3 farklı modu bulunmaktadır. Bu modlar güç tüketimi ve menzili etkilemekte olup, kullanım alanına göre seçilmelidir. Tablo 1.3’te ayrıntılar verilmiştir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mod** | **FU1** | **FU2** | **FU3** |  |
| **Boştaki Akım** | **3.6mA** | **80μA** | **16mA** | **Ortalama Değer** |
| **İletim Zaman** | **15-25mS** | **500mS** | **4-80mS** | **Bir Byte Gönderme** |
| **Gecikmesi** |  |  |  |  |

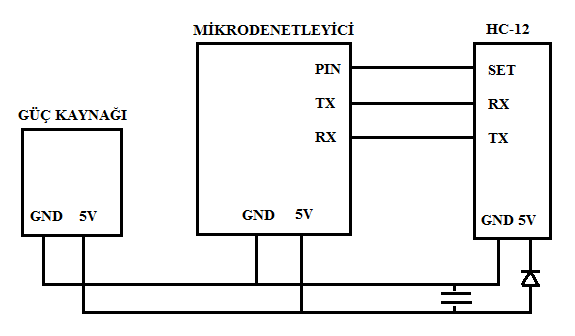
**Tablo 2.4** HC-12 modları

Modülün kullanımı, gerekli konfigürasyon ayarlarının yapılabilmesi için AT komutları mevcuttur. Ayarlama moduna girmek içi Şekil 1.10’daki şemada da görülen SET pinini sıfıra çekmek gerekmektedir. Komutlar Tablo 1.4’te verilmiştir.

|  |  |  |  |
| --- | --- | --- | --- |
| **Komut** | **Açıklama** | **Örnek** | **Cevap** |
| AT | Test komutudur. | AT | OK |
| AT B\*\*\*\* | Baud hızını değiştirir. | AT+B19200 | OK+B19200 |
| AT C\*\*\*\* | Kanalı 0-100 arasında değiştirir. | AT+C021 | COK+C021 |
| AT+FU\* | Modu değiştirir. | AT+FU1 | AT+0K |
| AT+P\* | İletim gücünü 1-8 arasında değiştirir | AT+P5 | OK+P5 |
| AT+Ry | Modül bilgilerini verir.(B,C,F,P) Değerlerini alır | AT+RB | OK+B9600 |
| AT+RX | Modülün tüm parametrelerini verir. | AT+RX | OK+FU3\r\n  OK+B9600\r\n OK+C001\r\n OK+RP: +20dBm\r\n |
| AT+U\*\*\* | Veri biti, kontrol biti, durdurma biti ayarlarını yapar. | AT+U8O1 | OK+U8O1 |
| AT+V | Modül sürümünü verir. | AT+V | HC-12\_V1.1 |
| AT+SLEEP | Uyku moduna geçer. Yaklaşık 22uA akım tüketir. | AT+SLEEP | OK+SLEEP |
| AT+DEFAULT | Standart ayarlara yani FU3 modu ayarlarına döner. | AT+DEFAULT | OK+DEFAULT |

**Tablo 2.5** AT komutları

HC-12 ile mikrodenetleyici arasındaki bağlantı şeması aşağıda verilmiştir. GND 5V arasında kondansatör kullanılması zorunlu değildir. Ancak gerilimde gürültüyü önler.

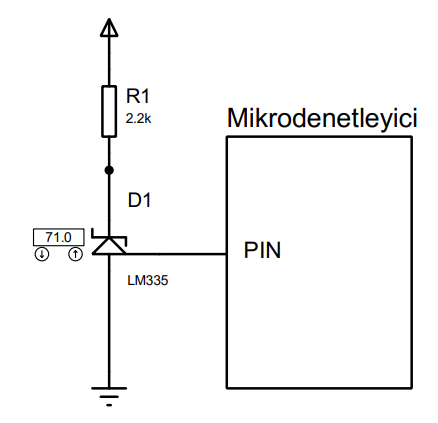


**Şekil 2. 10** HC-12 bağlantı şeması

## 2.3. Sensörler

Sensörler, otomatik kontrol cihazlarının duyu organları gibidir. Kontrol cihazları, dış ortamdan her şeyi duyargalar sayesinde algılarlar. Sayısal ve analog olmak üzere iki şekilde haberleşen sensörler mevcuttur. Araç içerisinde çeşitli sensörler bulunabilir. Bu proje dahilinde hall effect sensörleri ve sıcaklık sensörü kullanılmıştır.

Sıcaklık sensörü olarak, analog çıkış veren LM335 kullanılmıştır. Bağlantı şeması ve sıcaklık ölçüm formülü aşağıda verilmiştir.



**Şekil 2. 11** LM35 bağlantı şeması

## 2.4. Batarya

Araçtaki batarya, birbirlerine paralel ve seri bağlanmış küçük pillerden oluşmaktadır. Telemetrinin bu noktadaki görevi, bataryadan anlık akım, gerilim değerleri ve kapasitesini bir iletişim protokolüyle almaktır. Projede kullanılan batarya RS232 protokolüyle iletişim kurmaktadır. RS232, UART protokolü gibi çalışmakla birlikte, farkı olarak 12V gerilimle iletişim kurmaktadır. Bu nedenle mikrodenetleyicinin şekil 2.6’da ki gibi UART protokolüne uygun bağlanması, yalnız araya RS232 dönüştürücü koymak gerekmektedir.

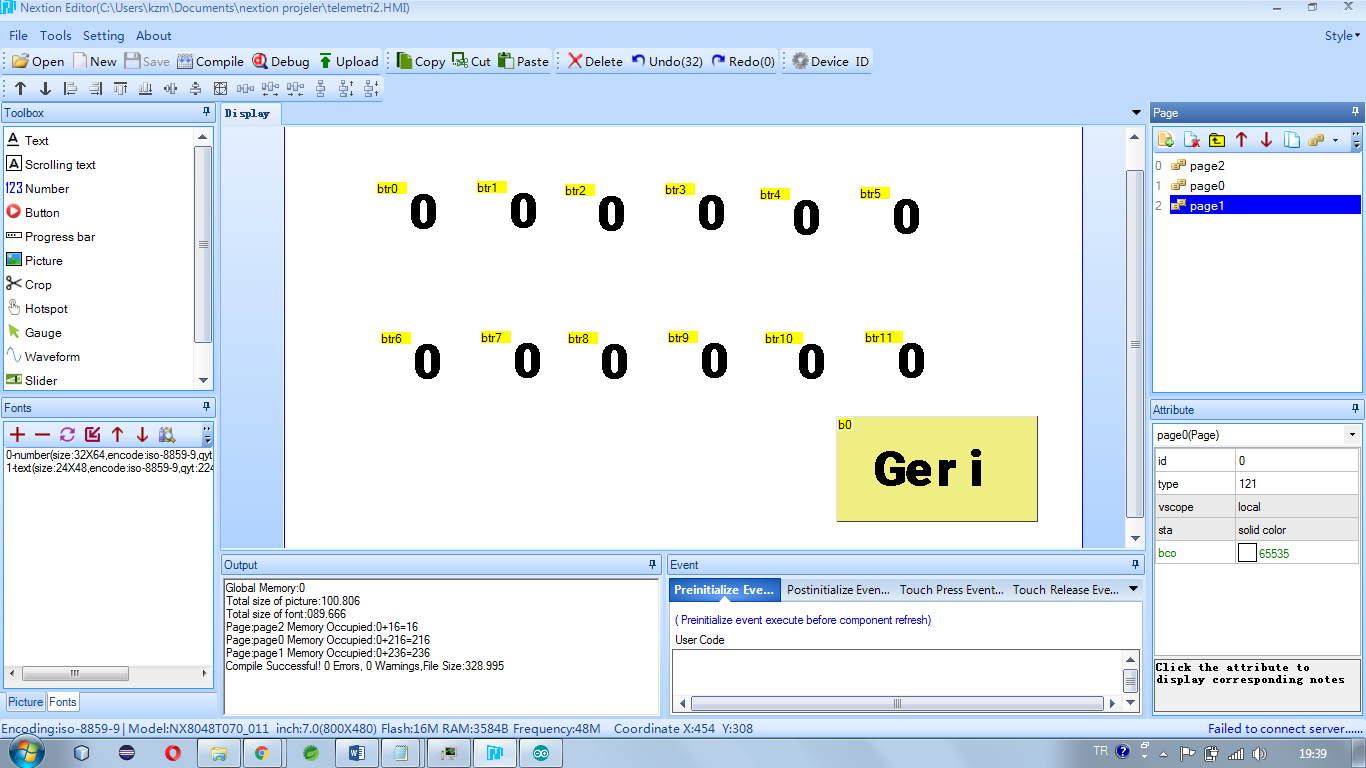
## 2.5. Gösterim

Verilerin toplanması, işlenmesi işlemleri yapıldıktan sonra bir de insanların ulaşabilmesi için görsel hale getirilmesi gerekmektedir. Bunun için birçok yöntem olmasına karşın, bu projede araç içerisine yerleştirilecek bir ekran kullanılmasına karar verilmiştir.

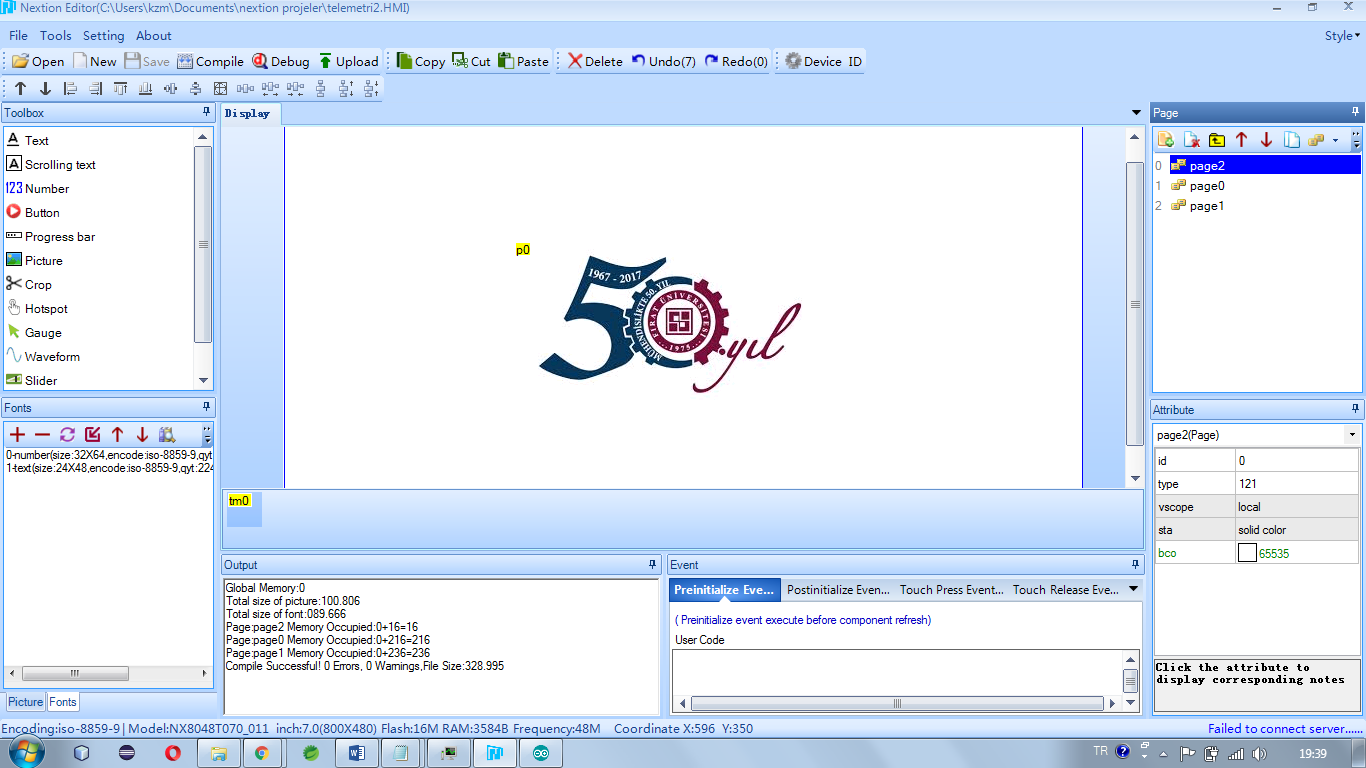
Kullanılan ekran, Nextion HMI dokunmatik TFT’dir. Bu ekranın editör programı olması ve ayrıca birçok mikrodenetleyici için kütüphanesinin bulunması kolaylık bakımından önemlidir. Editör programında, sayı, metin alanı, resim, bar gibi birçok eleman eklenebilmekte ve programlanabilmektedir. 8 Mb hafızası bulunmakta ve UART ile haberleşmektedir. Aşağıda örnek sayfalar verilmiştir.



**Şekil 2. 12** LCD bilgi ekranı 1



**Şekil 2. 13** LCD bilgi ekranı 2



**Şekil 2. 14** LCD bilgi ekranı

# 3. İZLEME MERKEZİ

Telemetri sisteminin izleme merkezi kısmı, iletim merkezi tarafından gönderilen verilerin alınmasında, kullanıcıya gösterilmesinde ve verilere daha sonra ulaşabilmek amacıyla kaydedilmesinde görev alır.

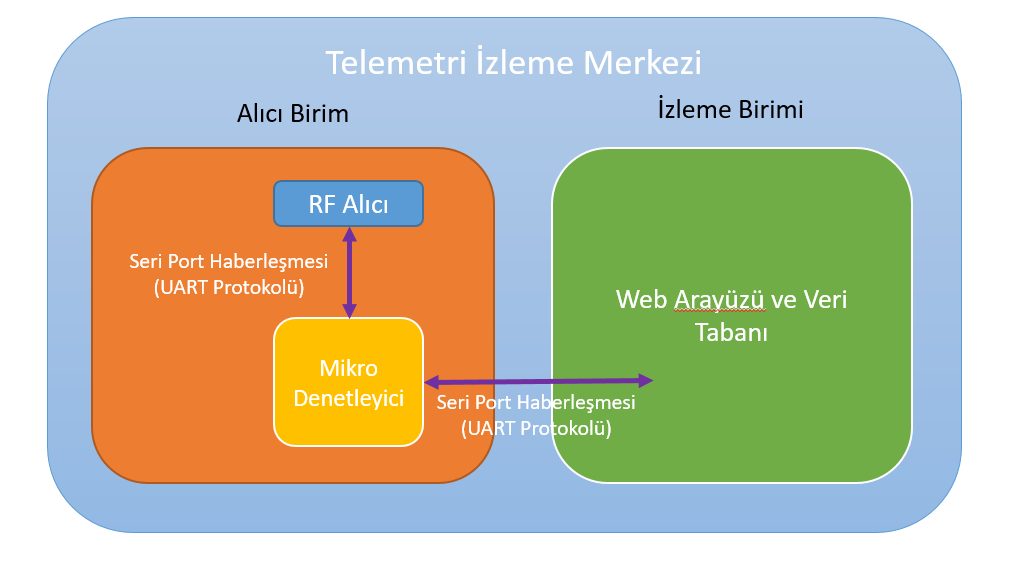
Bu bölümde izleme merkezinin genel yapısı, yazılım geliştirilirken kullanılan mimari ve kullanılan teknolojiler gibi konular ele alınacaktır.

## 3.1 Genel Yapı

İzleme merkezi genel olarak iki birimden oluşmaktadır. İlk birim iletim merkezinden gelen şifreli mesajı almak, alınan şifreli mesajı çözerek orijinal mesajı elde etmek ve elde edilen mesajın izleme birimine aktarmak ile görevli “alıcı birimi”dir. İkinci birim ise verilerin bir kullanıcı arayüzü vasıtasıyla kullanıcıya gösterileceği ve gelen verilerin bir veri tabanına kayıt edilmesinden sorumlu “izleme birimi”dir.

Alıcı birimi, bir RF alıcı modülü ve bir kontrolcüden oluşmaktadır. RF alıcı modülünün görevi iletim merkezinden radyo dalgaları vasıtasıyla gelen sinyalleri anteni ile yakalamaktır. Alıcı birimdeki kontrolcü ise RF modülünü kontrol etmekle, gelen verileri işleyerek şifreli mesajı çözmek ve çözülen mesajı izleme birimine iletmekle görevlidir.

İzleme biriminde ise bir kişisel bilgisayar bulunmakta olup verilerin kullanıcıya gösterilmesini ve verilerin kaydedilmesini sağlayacak bir yazılım geliştirilerek kişisel bilgisayar üzerinde çalıştırılmaktadır. Geliştirilen yazılım bir web uygulaması olup verilerin kaydedilmesi için de veri tabanı kullanılmaktadır. İzleme merkezinin genel yapısı Şekil 3.1’de verilmiştir. Genel yapısı anlatılan birimler kendi konu başlığı altında detaylı olarak incelenecektir.



**Şekil 3. 1** Telemetri İzleme Merkezi

## 3.2 Alıcı Birimi

İletim merkezinden verilerin iletimi radyo sinyalleri vasıtasıyla olmaktadır. Bu nedenle iletilen sinyallerin alınması için de bir radyo frekansı alıcısına ihtiyaç vardır. Bu projede alıcı olarak HC-12 modülü kullanılmaktadır.

RF alıcının kontrolü, alınan şifreli mesajın çözülmesi ve izleme birimine gönderilmesi için bir mikro denetleyiciye ihtiyaç duyulur. Bu işlemler ağır hesaplama yükü getirmediğinden dolayı ATmega328P mikro denetleyicisini kullanan Arduino UNO kartı bu iş için yeterli olacaktır. Arduino kartının RF modülünü kontrol etmesi ve radyo dalgaları ile gelen verilerin RF modülden Arduino kartına iletilmesi için aralarında bir iletişim olması gerekmektedir. Elektronik cihazlar arası iletişim ise protokoller kullanmayı gerektirir. Bunun nedeni iletişime taraf olanların ortak bir dil konuşmasını sağlamaktır. Bu projede İletişimin sağlanması için RF modülü ile Arduino kartı arasındaki iletişim bir asenkron seri haberleşme türü olan UART protokolü ile gerçekleştirilmektedir.

İletim merkezinden gelen sinyaller RF modülü ile yakalanıp Arduino ile işlendikten sonra elde edilen veriler izleme birimdeki bilgisayara iletilmelidir. Bu nedenle Arduino ile izleme birimindeki bilgisayar arasında da bir iletişim ve dolayısıyla bir protokol gereklidir.

Bir kişisel bilgisayar ile Arduino kartını haberleştirmenin en kolay yollarından birisi USB portu kullanılarak gerçekleştirilen seri haberleşmedir. Bu nedenle Alıcı Birimindeki Arduino ile izleme birimindeki kişisel bilgisayarın haberleşmesi de USB kablo üzerinden UART protokolü ile gerçekleştirilmektedir.

İletim merkezinden gelen veriler güvenlik ve veri bütünlüğü açısından şifrelenerek gönderildiği için alıcı biriminde bu şifrelerin çözülmesi gerekmektedir. Gelen veri ancak bu şekilde anlamlı bir şekilde alınmış olur. Aksi halde alınan verilerden bir anlam çıkarmak mümkün olmayacaktır.

Şifreleme ve şifre çözme işlemlerinin nasıl gerçekleştiği bölüm 2.2.2.2’de detaylı olarak anlatıldığından burada bu kadar bilgi vermek yeterli olacaktır.

Şifreleme kullanılarak veri güvenliği ve bütünlüğünün sağlanmasının ardından asıl gönderilmek istenen mesajın da belirli bir formatta olması gerekir ki iletişim kuran taraflar aldıkları verilerden bir bilgi elde edebilsinler. Aksi takdirde sağlıklı bir iletişim gerçekleştirilemeyecektir.

Veriler, iletim merkezi ile alıcı birimi ve alıcı birimi ile izleme birimi arasında olmak üzere iki yönlü olarak gerçekleştirilmektedir. Verilerin hangi cihazdan geldiğini tespit etmek için fazladan bir uğraş içerisine girmemek için her bir taraf için farklı bir format belirlemekte fayda olacaktır.

Bu doğrultuda iki farklı format belirlenmiş olup Tablo 3.1’de formatlar ve ne anlama geldikleri gösterilmiştir. Tablo 3.2’de ise örnek mesajlar ve anlamları verilmiştir.

Formatlar ve Anlamları

|  |  |  |  |
| --- | --- | --- | --- |
| Format | Nereden | Nereye | Anlamı |
| [BÜYÜKHARFLERLEMESAJ]# | İzleme Birimi | Alıcı Birimi | İzleme Birimi ile Alıcı Birim arasındaki bağlantı ve iletişim kontrolü |
| [BÜYÜKHARFLERLEMESAJ]# | Alıcı Birimi | İzleme Birimi | Alıcı Birim ile İzleme Birimi arasındaki bağlantı ve iletişim kontrolü |
| $[Sensör Id]:[Sensör Değeri]# | İletim Merkezi | Alıcı Birimi üzerinden İzleme Birimi | İletim merkezinde ölçülen bir sensör değerinin, Sensör Id No ve Sensör Değer çifti olarak alıcı birimine iletilmesi(Gelen değerler bit düzeyinde şifreli olarak gelip çözüldükten sonra bu formata gelmektedir. Şifre çözme işlemi tamamlandıktan sonra mesaj, izleme birimine alıcı birim tarafından iletilir.) |

**Tablo 3.1**

Örnek Mesajlar ve Anlamları

|  |  |  |  |
| --- | --- | --- | --- |
| Mesaj | Nereden | Nereye | Anlamı |
| CONNECT# | İzleme Birimi | Alıcı Birimi | Seri port haberleşmesi için bağlantı kurma isteği |
| CONNECTED# | Alıcı Birim | İzleme Birimi | Seri port haberleşmesi için bağlantı kuruldu. |
| $1:85# | İletim Merkezi | Alıcı Birim üzerinden İzleme Birimi | Id numarası 1 olan sensörün değeri 85 |

**Tablo 3.2**

## 3.3 İzleme Birimi

İzleme birimi kişisel bilgisayar üzerinde çalıştırılan, alıcı biriminden verileri okuyup arayüzde gösterirken asenkron olarak da gelen verileri veri tabanına kaydetmektedir.

### 3.3.1 İzleme Biriminde Kullanılacak Teknolojiler

Yazılım geliştirmek için birçok alternatif geliştirme ortamı vardır. Bir yazılım geliştirilirken bu alternatiflerden birisi veya birkaçı kullanılarak sonuca gidilir. Burada seçim yapılırken geliştirilecek yazılımın özelliklerine göre bir platform seçilebilir. İzleme birimi için geliştirilecek yazılım bir web projesi olarak seçilmiştir. Neden web projesi olarak seçildiği ilerleyen kısımlarda anlatılacaktır. Bu durumda web geliştirme için çeşitli seçenekler mevcuttur. Bu proje için web geliştirmede önemli bir yer tutan Microsoft firmasının geliştirdiği .NET Core platformu kullanılacaktır.

.NET Core platformu açık kaynak kodlu, cross platform bir yazılım geliştirme ortamıdır. Projenin backend kısmında .NET Core Class Library olarak adlandırılan paketler kullanılacak olup web arayüzü için de ASP.NET Core paketi kullanılacaktır. Programlama dili olarak C# ve verilerin kaydedilmesi için kullanılacak veri tabanı olarak MS SQL Server kullanılmasına karar verilmiştir.

Proje web tabanlı olacağından dolayı sunucu-istemci mimarisi uygulanacaktır. Lakin sunucu da istemci de aynı yerel bilgisayara kurularak bir masaüstü yazılım gibi kullanılacaktır. Neden böyle bir yol tercih edildiği aşağıda açıklanmıştır.

Arayüz olarak web arayüzü tercih edilmiştir. Web arayüzünün tercih edilmesinin sebepleri ise şu şekildedir:

* Web arayüz geliştirirken ulaşılabilecek kaynağın diğer arayüz geliştirme platformlarına göre oldukça fazla olması
* Bir problem ile karşılaşılması durumunda forumlar vasıtasıyla (örneğin Stack Overflow gibi) yardım alınabilecek kişilerin sayıca oldukça fazla olması
* Birçok açık kaynak bileşenin mevcut olması ve bunların fazladan efor sarf etmeden projeye entegre edilerek bir yandan zaman tasarrufu sağlarken bir yandan da daha profesyonel bir görünüm elde etme imkanı sağlamasıdır.

Nitekim bu projenin arayüz geliştirme aşamasında da açık kaynak projelerden yararlanılmıştır.

Yararlanılan kaynaklar kaynakça kısmında belirtilmiştir.

Web projesi kullanıldığından dolayı arayüz ile iletişime geçmek için HTTP protokolü kullanılmaktadır. HTTP protokolü ise istek-cevap(request-response) şeklinde çalıştığından dolayı gerçek zamanlı uygulamalar için farklı yöntemler uygulamak gerekmektedir. Örneğin gerçek zamanlı veri iletimi için;

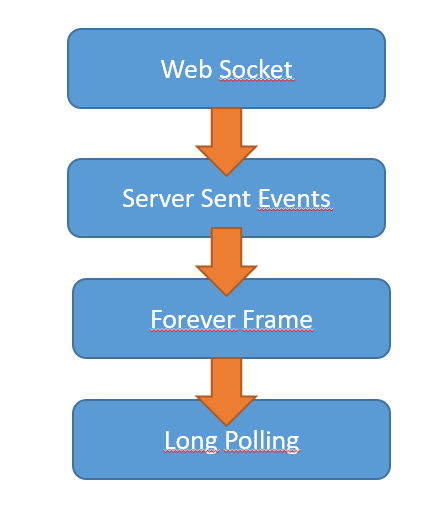
Bu yöntemlerden en basiti belirli aralıklarla suncuya istekte bulunup yeni veri olup olmadığını sorgulamaktır. Bu işlem AJAX ile gerçekleştirilebilmesine rağmen daha optimum çözümler aranmalıdır. Zira verinin gelip gelmediğini sorgulamak için istek frekansı arttırılırsa çok fazla ağ trafiği oluşacaktır. Yeni verilerin seyrek zaman aralıkları veya düzensiz zaman aralıkları ile geldiği düşünüldüğünde istek birçok kez cevapsız kalacaktır. Bu da boşuna trafik oluşturma anlamına gelir. Bu gibi durumlardan ötürü yeni yöntemler aranmış ve alternatif birkaç yöntem sunulmuştur. Bunlar Long Polling, Server Sent Events, Web Soket gibi yöntemlerdir.

Bu yöntemler kısaca şu şekilde çalışmaktadır;

1. Long Polling: Bu yöntemde istemci sunucuya yeni veri olup olmadığı hakkında istek gönderir. Sunucu ise yeni bir veri gelene kadar bekler. Yeni veri geldiğinde istemciye cevap verir.
2. Forever Frame: Bir IFrame elementi üzerinden çalışan scripler tek yönlü olarak sunucudan istemciye bilgi aktarımı gerçekleşir.
3. Server Sent Events: İstemci tarafından sunucuya bir istek yapılır. Sunucu içeriği text/event-stream olarak ayarlar ve cevap verir. O andan itibaren sunucu tarafında bir veri geldikçe istemciye talep gelmeden de veri gönderebilir.
4. Web Socket: Tarayıcının bu yöntemi desteklemesi durumunda istemci sunucuca bir istek gönderir ve arada sürekli bir bağlantı oluşur. Bu yöntemde çift yönlü olarak asenkron iletişim gerçek zamanlı yapılabilir.

.NET Core platformunda bu yöntemlerden en uygununu seçerek gerçek zamanlı web uygulamaları geliştirmeyi sağlayan SignalR adında bir kütüphane mevcuttur. Bu kütüphane en uygun iletişim tekniğini bularak gerçek zamanlı iletişimi sağlar. Kolay kullanımının yanı sıra bağlantı kopması durumunda tekrar bağlantı kurma özelliği ile de kolaylık sağlamaktadır. SignalR yukarıdaki adı geçen en uygun yöntemi bulmak için Şekil 3.2’deki sırayı izler.

Bu nedenle izleme birimine yeni veri geldiğinde arayüzü güncellemek için SignalR kütüphanesi kullanılmıştır.



**Şekil 3. 2** SignalR İletişim Seçim Sırası

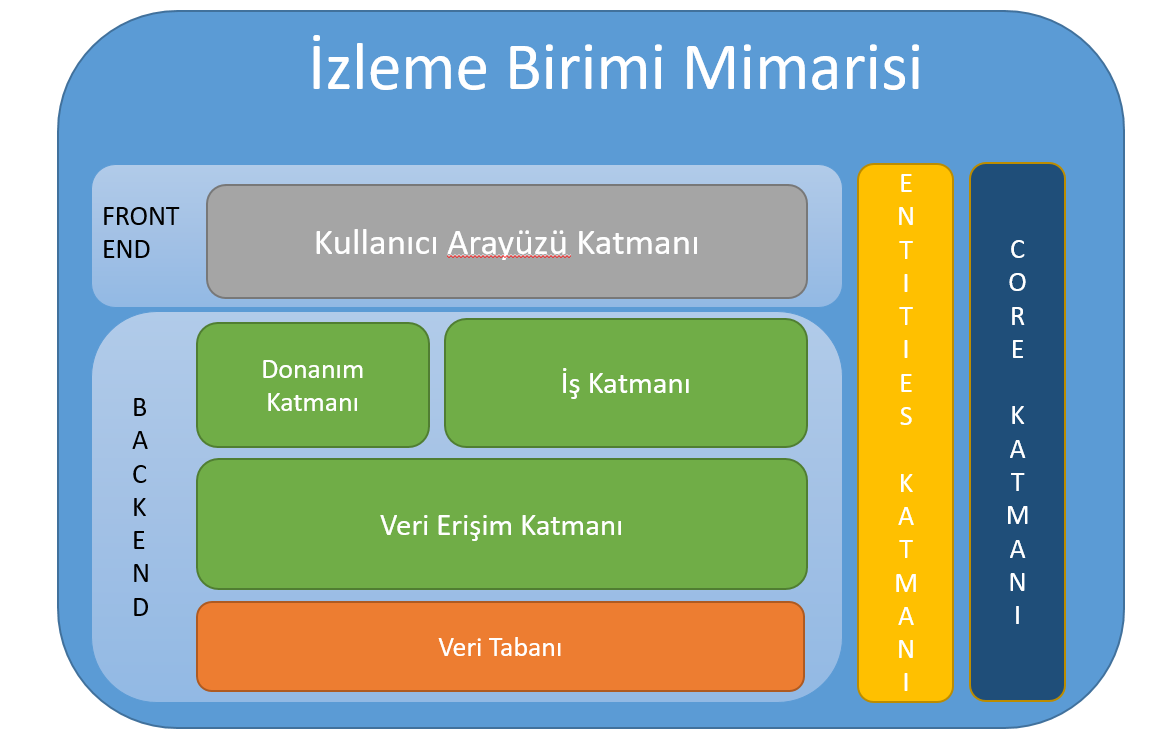
### 3.3.2 SOLID Prensipleri ve İzleme Birimi Mimarisi

Yazılım geliştirme bir süreçtir. Bu süreçte geliştirme yapılırken bazı prensiplere uygun olarak geliştirme yapmak yazılım sisteminin yönetimi açısından oldukça faydalıdır. Bu prensiplerin en başında SOLID olarak adlandırılan ve 5 tane prensibin baş harfinden oluşan prensipler bütünü yer almaktadır. SOLID prensiplerinin isimleri ve ne anlama geldikleri aşağıda belirtilmiştir.

* Single Responsibility Principle(Tek Sorumluluk Prensibi):Her bir birimin tek bir görevinin olması gerektiğini belirten prensiptir. Örneğin; Her bir class her bir fonksiyonun tek bir işi yerine getirmesi gibi.
* Open-Closed Principle(Açık-Kapalı Prensibi): Sisteme yeni bir özellik eklendiğinde mevcut sistemde değişiklik yapmaya kapalı, sistemi genişletmeye, yeni özellik eklemeye açık olması gerektiğini savunan prensiptir.
* Liskov’s Substitution Principle(Liskov’un Yerine Geçme Prensibi):Üst sınıftan türetilen sınıfların üst sınıfların yerine geçebilmesi gerektiğini belirten prensiptir.
* Interface Segregation Principle(Arayüz Ayrımı Prensibi):Sorumlulukların tek bir arayüzde toplamak yerine, uygun bir şekilde ayrı ayrı arayüzlere bölünmesi gerektiğini belirten prensiptir.
* Dependency Inversion Principle(Bağımlılıkların Terslenmesi Prensibi):Üst seviye bir sınıfın alt seviyeli bir sınıfa bağımlı olmaması gerektiğini belirten prensitptir.

Bu projede yazılımda soyutlamaya ve Tek Sorumluluk İlkesi’ne uygun olarak katmanlı mimari kullanılmıştır. Kullanıcı arayüzü hariç tüm katmanlar .Net Core platformunun 3.0 versiyonu “Class Library” projesi şeklinde oluşturulmuştur. Kullanıcı arayüzü katmanı ise bir web projesi olan ASP.NET Core MVC ile gerçekleştirilmiştir.

İzleme biriminin mimari yapısını gösteren görsel Şekil 3.3’de görüldüğü gibidir.



**Şekil 3. 3** İzleme Biriminin Mimarisi

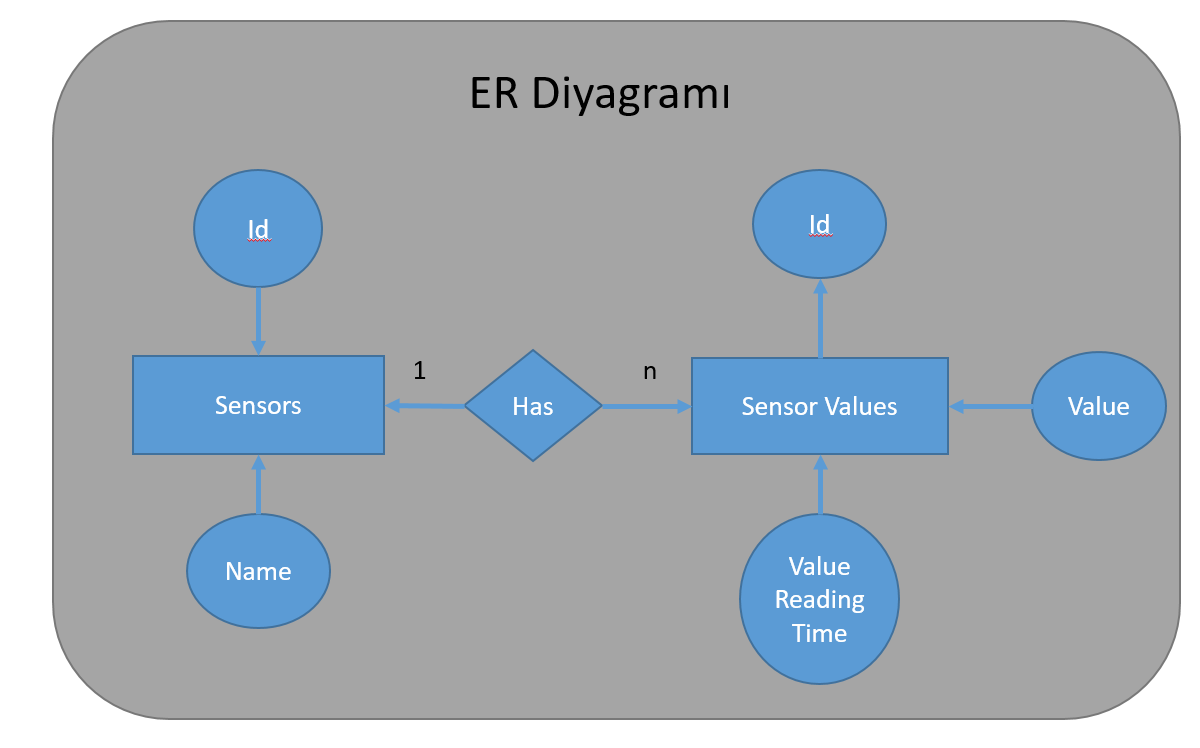
Şekilde verilen mimarideki katmanların görevleri aşağıdaki gibidir.

* Entities Katmanı, veri tabanı nesnelerine karşılık gelen sınıfları barındırır. Entities katmanındaki sınıflar projenin birçok katmanda kullanılmaktadır.
* Core Katmanı, Çoğu projede ortak olarak yapılması gereken işlemlerin parametrik olarak kodlarının yazılmasıyla oluşmuş C# dilindeki karşılığı ile Generic sınıflar ve fonksiyonlar olan kodların bulunduğu katmandır. Örneğin Entity Framework ORM’si kullanılarak veri tabanına ekleme işleminin gerçekleştirilmesi işlemi bu katmanda gerçekleştirilmektedir.
* Veri erişim katmanı, uygulamada verilerin eklenmesi, silinmesi gibi veri tabanı işlemlerinin yapıldığı katmandır. Bu katmanda “Core “ katmanından da faydalanılarak veri tabanı işlemleri gerçekleştirilmektedir.
* İş katmanı, uygulama boyunca veri işlemlerinin yapılmadan önce gerekli koşulların kontrolünün yapıldığı katmandır.
* Donanım katmanı, alıcı birim ile iletişim kurmak için gerekli kodlamaların yapıldığı katmandır.
* Kullanıcı arayüzü katmanı, kullanıcı ile iletişim kurmak için oluşturulacak olan kullanıcı arayüzünün bulunduğu katmandır. Bu projede arayüz olarak ASP.NET Core MVC Web arayüzü kullanılmıştır.

### 3.3.3 Veri Tabanı Tasarımı ve Entity Framework

İletim merkezinden alınan sensör değerlerine daha sonra ulaşabilmek için okunan değerlerin kaydedilmesi gerekmektedir. Verilerin yapısal bir şekilde kaydedileceği depolama alanları veri tabanlarıdır. Veri tabanlarında verilerin birbirleri ile ilişkisel bir şekilde tutulduğu yapılar ilişkisel veri tabanı olarak adlandırılır. İlişkisel veri tabanlarında veriler tablolarda tutulur. Bu projede gelen veriler sensör verileri olduğundan dolayı sensörlerin verilerinin tutulacağı bir tablo ve sensörlerin id numarası ile eşleşen sensör adlarının bulunduğu bir tablo tasarımı ER diyagramı ile temsil edilmiştir. Temsil edilen yapı .NET Core platformunda C# programlama dili ile sınıflar(class) olarak tanımlanmıştır. Kodda yazılmış sınıfları veri tabanı tabloları ile eşleyen ve veri tabanı işlemlerini programlama dili ile yapılmasını sağlayan araçlar vardır. Bunlara ORM(Object Relational Mapper) adı verilmektedir. Bu projede bir ORM olan Entity Framework kullanılarak veri tabanı işlemleri C# sınıfları vasıtasıyla yapılmıştır.

Veri tabanının tasarımı Şekil 3.4’de ER diyagramı şeklinde verilmiştir.



**Şekil 3. 4** Veri Tabanı ER Diyagramı

### 3.3.4 İzleme Biriminin Çalışma Sistemi

İzleme biriminin mimarisi anlatılırken katmanların görevlerinden bahsedilmişti. Bu katmanların içerisinde bulunan C# kodları ile her bir katman kendi görevini icra etmektedir. Sistemin çalışma yapısı katmanlar üzerinden ele alınırsa;

Proje ilk çalıştırıldığında arayüz katmanında bulunan kodlar çalıştırılarak ekrana bir web sayfası oluşturur. Ekranda görünen arayüzden alıcı birimi ile iletişim kurulacak seri port ayarı için gerekli olan seri port adı ve iletişim hızı seçilir. Bu ayar yapıldıktan sonra başka bir sayfaya yönlendirme yapılır. Yönlendirme yapılan sayfada sensörden gelen verilerin ekranda gösterileceği grafiksel alanlar mevcuttur. Ekranın üst kısmında bulunan butonlar vasıtasıyla alıcı birim ile iletişim gerçekleştirilir. Bu kısımda bağlantı isteği, bağlantının test edilmesi, verilerin alınması gibi kontroller mevcuttur. Butonların üzerinde bulunan mavi alan ise bilgilendirme amaçlı uyarılar için kullanılmıştır.

Arayüzdeki butonlar vasıtasıyla bağlantı başlatılıp veriler talep edildikten sonraki işlemler şu şekildedir:

* Arayüzde seçilen port ayarlarına göre iletişim için gerekli olan nesneler donanım katmanında üretilir.
* Üretilen nesneler vasıtasıyla seri port, dinleme moduna geçer.
* Veriler geldikçe donanım katmanında işlenir.
* Gelen verilerin işlenmesi tamamlanınca donanım katmanında bir olay(Event) gerçekleşir.
* Oluşan olaya daha önceden abone olmuş olan iş katmanına ve arayüz katmanına işlenmiş veri ile beraber bir bildirim gider.
* Bildirim ve gelen işlenmiş veriyi alan iş katmanı veri erişim katmanı ile iletişim kurarak gelen verinin veri tabanına eklenmesi sağlar.
* Arayüz kısmının sunucuda çalışan kısmına bildirim geldiğinde SignalR kütüphanesinin sunucu tarafındaki kısmı verileri istemciye ulaştırır.
* İstemcide bulunan SignalR kütüphanesi gelen verilere göre arayüzü günceller.

Veri erişim katmanında kullanılan Entitiy Framework ORM’si veri tabanı ile entities katamanında bulunan sınıfları eşleştirerek veri tabanını eşitler.

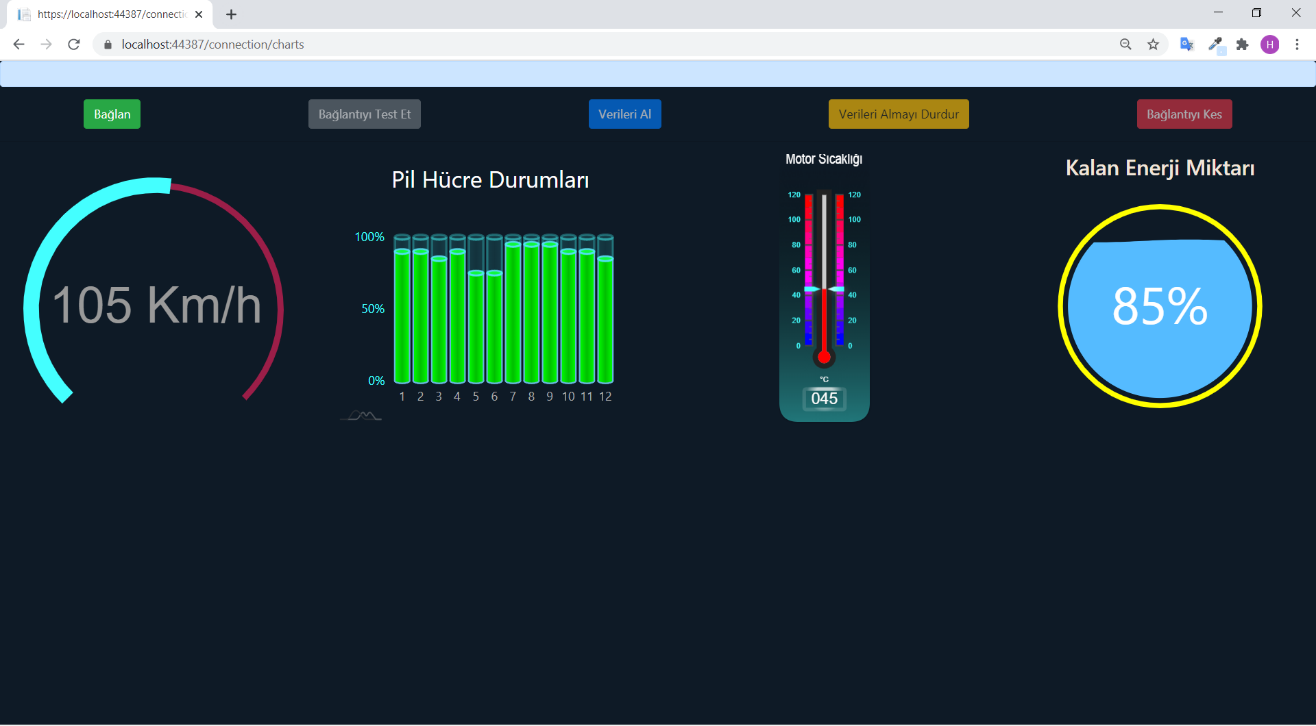
### 3.3.5 Kullanıcı Arayüzü

Kullanıcı arayüzünün tasarımı yapılırken HTML CSS ve JavaScript kullanılmıştır.

Bunlara ek olarak kütüphaneler de kullanılmıştır. Örneğin butonun stilini belirlemek için, web sayfasının yerleşimi için BootStrap kütüphanesi, kullanılmıştır.

Sensör göstergeleri için Github gibi kod paylaşım sitelerinde paylaşılan JavaScript kütüphaneleri kullanılmış olup kullanılan kütüphaneleri linkleri kaynakça kısmında belirtilmiştir.

Bu işlemler yapıldıktan sonra kullanıcı arayüzünün son halinin ekran görüntüsü Şekil 3.5’de gösterilmiştir.



**Şekil 3. 5** Kullanıcı Arayüzü

# SONUÇ

Telemetri projesi kapsamında yapılan çalışmalar, mikrodenetleyiciler arasında kablolu, kablosuz iletişim, sensörlerden verilerin toplanması, web uygulamaları, soket işlemleri gibi konular üzerine tecrübe kazandırmıştır. Kablosuz iletişimde ortaya çıkmış olan, veri bütünlüğü ve güvenlik konularındaki zafiyetlerin giderilmesi için gerekli algoritmalar geliştirilmiş, menzil baud oranları incelenerek ihtiyaç doğrultusunda seçimler yapılmıştır. Motor hız ölçümü için alternatif yöntemler düşünülmüş optimum sonuca gidilmeye çalışılmıştır. Veriler lcd ekrana gönderilerek kullanıcının erişimine sunulmuş, ayrıca uzak cihaza da aktarılmıştır.

Uzak cihazda, web uygulaması geliştirilmiş, soket programlama ile güncel verilerin hızla alınması sağlanmıştır. Asenkron olarak veritabanı kayıt işlemleri yapılarak, bekleme süreleri minimuma indirilmiş, sistem performansı optimum noktaya ulaştırılmaya çalışılmıştır.

Neticede, araç içindeki kontrol ve iletim merkezi sensör verilerini toplayarak kısıtlı işleme tabi tutmuş, bu verileri ekrana ve uzak cihaza göndermiştir. Uzak cihaz ise veritabanı kayıt işlemlerini yaparak, web üzerindeki görsel arayüzde verileri kullanıcılara sunmuştur.

# KAYNAKÇA

1. BALOĞLU A, Akıllı Şehirlerde Kablosuz Haberleşme Teknolojileri ve Doğru Teknoloji Seçimi, İstanbul Sabahattin Zaim Üniversitesi Fen Bilimleri Enstitüsü Dergisi, İstanbul 34408, Türkiye, 22-27,2019
2. SOYTÜRK M, ALTILAR D, Telsiz, tasarsız ve duyarga ağlar için kaynak-başlatmalı coğrafi veri akış yöntemi, İtü Dergisi, İTÜ Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Programı, 31-32, 2008
3. AYAV T, IEEE 802.11 Kablosuz Ağları için Bir Gerçek Zamanlı İletişim Protokolü, İzmir Yüksek Teknoloji Enstitüsü, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, 5-11, 2017
4. https://gelecegiyazanlar.turkcell.com.tr
5. www.elektrikce.com
6. https://nextion.tech

# ÖZGEÇMİŞ 1

****

KİŞİSEL BİLGİLER

Adı Soyadı : Kazım BAŞLAK

Doğum Tarihi : 28.08.1996

Uyruk : T.C.

İLETİŞİM BİLGİLERİ

Adres Bilgileri : Zeyrek Mah. İhtiyar Sok. No:1/5 Fatih / İstanbul

Cep Tel : 0539 389 04 29

E-posta : kzmbaslak@gmail.com

EĞİTİM BİLGİLERİ

Üniversite : Fırat Üniversitesi Bilgisayar Mühendisliği

Lise : İstanbul / Galata Mesleki ve Teknik Anadolu Lisesi

İlkokul : İstanbul / Şeyhülislam Hayri Efendi İÖO

# ÖZGEÇMİŞ 2

****

KİŞİSEL BİLGİLER

Adı Soyadı : Harun AĞCA

Doğum Tarihi : 10.10.1997

Uyruk : T.C.

İLETİŞİM BİLGİLERİ

Adres Bilgileri : Göllüce Mah. 25095 Sok. No:2/32 Şehitkamil / Gaziantep

Cep Tel : 0541 880 13 00

E-posta : harun39117@gmail.com

EĞİTİM BİLGİLERİ

Üniversite : Fırat Üniversitesi Bilgisayar Mühendisliği

Lise : Gaziantep / Necip Fazıl Kısakürek Anadolu Lisesi

İlkokul : Gaziantep / Ömer Güzel İlkokulu İÖO