

Filtre Yazılımları Kullanılarak Sıcaklık Değerlerinin IoT Tabanlı Sistemde Gösterilmesi

Tarık ÜNLÜ^{1*}, Resul BÜTÜNER², Zeki KÜÇÜKKARA³

^{1*} Adnan Akgül Özel Eğitim Meslek Lisesi, Bilişim Teknolojileri Alanı, Konya, Türkiye, (ORCID: 0000-0001-6543-2320), tarik1945@gmail.com

² Beypazarı Fatih Mesleki ve Teknik Anadolu Lisesi, Bilişim Teknolojileri Alanı, Ankara, Türkiye, (ORCID: 0000-0002-9778-2349), rbutuner@gmail.com

³ Adil Karaağaç Mesleki ve Teknik Anadolu Lisesi, Bilişim Teknolojileri Alanı, Konya, Türkiye, (ORCID: 0000-0002-5204-0819), zekikucukkara@gmail.com

(İlk Geliş Tarihi 14.07.2021 ve Kabul Tarihi 26.01.2022)

(DOI: 10.35354/tbed.971237)

ATIF/REFERENCE: Ünlü, T., Bütüner, R., Küçükkara, Z. (2022). Filtre Yazılımları Kullanılarak Sıcaklık Değerlerinin IoT Tabanlı Sistemde Gösterilmesi. *Teknik Bilimler Dergisi*, 12 (1), 1-7.

Öz

Filtre yazılımları ses, görüntü ve verilerdeki gürültüyü azaltmak veya bastırmak amacıyla kullanılır. Bu yazılımlar ile anlamlı sonuçlar çıkarılması amacıyla istenmeyen değerler çıkışa verilmemektedir. Böylelikle, herhangi bir problem için kurulan mekatronik sistemin, filtre yazılımları uygulanarak optimize edilmiş şekilde çalışması sağlanacaktır. Bu çalışmada sıcaklık sensörü kullanılarak gerçek zamanlı sistem tasarımı gerçekleştirilmiştir. IoT tabanlı oluşturulan sistemde 3 farklı filtre yazılımı uygulanarak sırası ile ham sıcaklık değerleri ile karşılaştırma yapılmıştır. Median, ortalamalar ve kalman filtreleri kullanılmıştır. Grafikselsel olarak bu filtre yazılımlarının karşılaştırılması gerçekleştirilmiştir. Çalışma ile bir mekatronik sistemin filtre yazılımları sayesinde daha verimli çalışması sağlanmıştır.

Anahtar Kelimeler: IoT, Median, Kalman, Hareketli ortalamalar, Gerçek zamanlı haberleşme, Node MCU.

Displaying Temperature Values in IoT Based System Using Filter Software

Abstract

Filter software is used to reduce or suppress noise in audio, video and data. With these software, undesirable values are not output for the purpose of obtaining meaningful results. Thus, the mechatronic system established for any problem will be optimized by applying filter software. In this study, real-time system design was carried out using a temperature sensor. In the system created based on IoT, 3 different filter software were applied and compared with the raw temperature values, respectively. Median, means, and kalman filters were used. omparison of this filter software has been done graphically. With the study, a mechatronic system was enabled to work more efficiently thanks to filter software.

Keywords: IoT, Median, Kalman, Moving averages, Real-time communication, Node MCU

1. Giriş

Hava kirliliği, atmosferde bulunan yabancı maddelerin, canlıların sağlığına zarar vermesi, maddi hasar oluşturması veya ekolojik dengeye zarar vermesi olarak tanımlanabilir. Bu nedenle hava kirliliği insan hayatını direk etkilemekte ve devamlı bir şekilde maruz kalındığında ise ciddi hastalıklara neden olabilmektedir [1]. 2019 yılında hava kirliliği üzerine yapılan araştırmada, hava kirliliği sağlık için en büyük çevresel faktör olarak kabul edilmiştir. Hava kirliliğinde biyokütle yakımı, trafik yoğunluğu, araç ve fabrikaların kontrolsüz CO yayması vb. durumlar çok etkili olmuş ve devamlı artış göstermiştir [2].

Nesnelerin interneti (IoT) teknolojisi, gerçek dünyadaki bulunan nesnelerin internet vasıtasıyla birbirleri ile haberleşmesinin sağlanması olarak tanımlanır [3]. Bu teknoloji sayesinde, cihazlar, insan müdahalesine gerek kalmadan birbirlerini algılayabilir, tanımlayabilir, birbirleriyle iletişim kurarak bilgi transferinde bulunabilirler [4]. Genel olarak, IoT teknolojisi veri toplamak ya da veri transferi yapmak için donanım bileşenleri, sensörler, aktüatörler, akıllı cihazlar ve yazılımları içerir. İç ve dış ortam hava kalitesinin izlenmesi, akıllı ev, güvenlik ve yaşlı bakım sistemleri, binalardaki enerji yönetimi ve mikro iklim izleme sistemleri [5] gibi pek çok IoT tabanlı uygulamalar günümüzde kullanılmaktadır. Endüstri 4.0'ın gelişile bu teknolojilerin kullanımı giderek artmaktadır.

Bu alanda yapılan çalışmalar incelenmiş ve literatür taraması yapılmıştır. Literatüre göre Lai ve arkadaşları (2019) hava kirliliğine sebep olan altı kimyasal maddenin (SO₂, NO₂, CO, O₃, PM_{2.5} ve PM₁₀) yoğunluğunu yüksek doğrulukta tespit etmek için yüksek hassasiyetli sensör kullanmanın IOT ağında her düğüm için yüksek maliyet gerektirdiğini dile getirmişler, bu durumu düşük maliyet ile gerçekleştirmek için düşük maliyetli sensör ve Kalman Filtresi (KF) algoritmasını kullanmışlar ve doğruluğunu %27 oranında arttırmayı başarmışlardır. Hareketli ortalama (SMA-Simple Moving Average), Üstel Ağırlıklı Hareketli Ortalama (EWMA- Exponentially Weighted Moving Average), Bütünleşik Otoregresif Hareketli Ortalama (ARIMA-Autoregressive Integrated Moving Average) olmak üzere üç yaygın tahmin algoritması ile KF algoritma karşılaştırma deneylerinde KF algoritması ile optimum tahmin sonuçlarını elde edebilmişler ve ortalama kare hatasını (MSE-Mean squared error) yaklaşık %68 oranında azaltmışlardır[6].

Üçgün ve arkadaşları (2020) iç ortamın hava kalitesini izlemek için IoT tabanlı izleme sistemi uygulamışlardır. Oluşturdukları bu sistemde, nem, ışık, gaz ve sıcaklık sensörlerinin bağlı olduğu Raspberry Pi 3 kontrol kartı kullanmışlardır. Sensörlerden gelen iç ortam verileri kontrol kartı ile işlenmiş ve elde edilen veriler veri tabanına kaydedilmiştir. Geliştirmiş oldukları hava kalitesi formülü sayesinde iç ortama ait hava kalitesi hesaplanmıştır. Hazırlanmış oldukları web arayüzünde iç ortama ait yaşam koşullarını içeren anlık bilgilerin takibini yapmışlardır[7].

Monteiro ve arkadaşları (2018) basit bir IoT senaryosu ile buzdolabı sıcaklık tahminlerini gerçekleştirmek için hem harici hem de dâhili sıcaklık sensöründen sağlanan veriyi kullanarak basit doğrusal korelasyondan, Bütünleşik Otoregresif Hareketli Ortalama modellerine kadar altı farklı farklı algoritmaların kullanımı karşılaştırmışlardır. Bir buzdolabından beş ay boyunca topladıkları gerçek verileri kullanarak hassasiyeti ve hesaplama maliyetlerini analiz etmişlerdir. Analiz sonucunda ARIMA

modelini kullanarak ortalama kare hatasını i0,09 C'ye kadar düşürmüşlerdir [8].

Ayele ve Mehta (2018) DHT11 sensöründen aldığı nem ve sıcaklık bilgisini ESP8266 IoT cihazını kullanarak bilgiyi bir server aracılığıyla Tekrarlayan Sinir Ağı (RNN- Recurrent Neural Network) sınıfının derin öğrenme alanında yer alan Uzun Kısa Süreli Bellek (LSTM- Long Short Term Memory) mimarisini kullanarak atmosferik hava sıcaklığını doğru tahmin etmeye çalışmışlardır [9].

Taştan (2019) hava kalitesini hesaplamak için gaz, toz, sıcaklık ve nem sensörlerinden gelen CO₂, CO, PM₁₀, NO₂, sıcaklık ve nem gibi parametrelerin kullanılmasıyla içerideki mekânın hava kalitesini anlık izlenmesini sağlayan IoT tabanlı bir sistem geliştirmiştir. Sistem tasarımında, ESP32 denetleyicisi ile ölçümü yapılan parametrelerin eşik değerini aşması durumunda kullanıcıyı uyaran bir yapı oluşturmuştur[10].

Ahiska ve arkadaşları (2008) tarafından yapılan çalışmada, Dünya Sağlık Örgütü'nün belirlediği standarta uygun, kanın depolanması amacıyla ısıtma ve soğutma işlemlerinin yapılabilmesini sağlayan Mikrodenetleyici Kontrollü Taşınabilir Termoelektrik Tıp Kiti cihazına ait elektronik kontrol devrelerini geliştirilmiştir. Böylece kitin iç sıcaklık kontrolünü analog sensörle ve dijital yöntemle olmak üzere iki farklı yöntemle yapabilen devre oluşturulmuş ve elde edilen kontrol sonuçlarını karşılaştırmışlar. Karşılaştırma sonucuna göre dijital sıcaklık sensörü ile oluşturulmuş devrenin yapmış olduğu soğutma ve ısıtma işleminde daha fazla enerji tasarrufu sağlandığı ve dijital sıcaklık sensörünün kalibrasyon probleminin olmadığı gözlemlenmiştir[11].

Altun ve Allahverdi (2007) parmak izi görüntüsünü iyileştirme ve netleştirme için iki yöntem önermişlerdir. Başlangıçta lokal olarak histogram eşitleme ile gürültü azaltma filtrelerini kullanmışlardır.

Gürültünün azaltılması bölümünde doğrusal, medyan ve uyarlamalı filtreleri kullanılmışlar. Diğer öneri de ise dalgacık dönüşümü ve iki boyutlu dalgacık dönüşümünün yeni bir versiyonu olan çevrimsel dönüşüm uygulamışlardır. Bu iki yöntem performans açısından karşılaştırmışlardır[12].

Pamadi ve Nickerson (2015) kablosuz sensör ağlarının işleyişi, veri yolu protokollerinin işleyişi, bir telli "DS18B20 Temperature Sensor" protokol ve arayüzüne ilişkin cihazlar üzerine araştırma yapmışlardır. DS18B20 sıcaklık sensörünün parazit modunda çalışırken bir veri ve bir toprak hattı kullanarak yapılandırılabilceği, çok düşük güç tükettiği ve gerçek zamanlı verileri nispeten yüksek hızlarda verimli bir şekilde sunduğunu belirtmişlerdir[13].

Bütüner ve Calp (2021) Covid19 salgınından korunmak ve hastaların herhangi bir sağlık kuruluşuna gitmeden nabız, kandaki oksijen ve vücut sıcaklığı değerlerinin takibinin yapılması, gerekli önlemlerin önceden alınması için mekatronik bir sistem geliştirmiştir. IoT teknolojisi ile ev ortamında iken hastaların nabız değerleri, kandaki oksijen miktarları ve vücut sıcaklığı değerleri ölçülüp veri tabanına kaydedilebilmektedir. Bu ölçülen değerleri hastanın kimlik numarasına göre internet ortamında yayınlanarak, ilgili doktor ve hemşireler tarafından takip edilebilmeleri sağlamışlardır[14].

Dijital ses, dijital görüntü ve verilerin içerdiği detayları anlaşılabilir hale getirmek için çeşitli görüntü, ses ve veri işleme yöntemleri kullanılır. Filtre yazılımları da bunlardan birisidir. Bu

kısımda, çalışmada kullanılan Orta Değer, Ortalama Değer ve Kalman filtrelerinden bahsedilmiştir.

2. Materyal ve Metot

Bu bölümde, sistemin donanım ve yazılım yapısı detaylı bir şekilde anlatılmıştır. Yazılım yapısında, sistemde kullanılan filtrelerden bahsedilmiştir. Donanımda ise sistemin çalışması için gerekli donanımların teknik özellikleri anlatılmıştır. IoT teknolojisi kullanılarak oluşturulan sistemde gömülü mimari ve WiFi modülünü içinde barındıran NodeMCU mikrodenetleyici kartı kullanılmıştır. Sıcaklık bilgileri DS18B20 sıcaklık sensörü tarafından alınmıştır.

2.1. Sistemin yazılım yapısı

2.1.1. Ortanca filtresi (median filter)

Bu filtre için sıklıkla kullanılan isimler, Ortanca Filtresi (Median filtresi) (Median Filtering) ve Sıralama Filtresi (Rank Filtering)'dir. Bu filtre, gürültüyü temizlemek amacıyla görüntü ve sinyal işleme konularında kullanılan tekniklerden biridir. Bir matris veya sayı dizisine bakılarak, ortalama değerler alınır ve kalan diğer değerler elenerek büyük farklı değerler temizlenmiş olur. Hareketli Ortalamalar Filtresine göre bu filtre lineer değildir. Ortanca Filtresi, normalinde mean filtresi gibi çalışarak görüntü ve sinyal işlemede gürültüyü azaltmak amacıyla kullanılır. Fakat görüntü işlemede detayların kaybolmaması konusunda Ortalama Filtresinden çok daha başarılı sonuç elde edilir. Devamlı veri okunan bir sistemde belirli zaman aralıkları ile sisteme sürekli veri girişi olur. Bu veriler içerisinde istenmeyen değerler de mevcuttur. Başlangıçta bir ara bellek (buffer) alanı oluşturulur. Ortadaki değeri alacağımızdan dolayı bu sayı tek olmak zorundadır. Örneğin boyutuna 5 değeri vermiş olalım. İlk 5 değer okunur ve devamında bu 5 değer küçükten büyüğe doğru dizi halinde sıralanır. Ortadaki değer çıkış olarak verilir. Diziyi ilk haline getirerek, sağa kaydırma işlemi yapılır. Bu işlemde eski değer atılarak yeni değer alınır. Aynı işlem gelen yeni durumlara da uygulanır. Sistem bu şekilde devam eder ve çalışır [15].

Aşamalar (5 elemanlı ara bellek dolduktan sonra):

Dizimiz şu şekilde olsun: $G = \{1,9,5,3,2\}$

Dizimizi küçükten büyüğe sıralıyoruz;

$\Rightarrow G = \{1,2,3,5,9\}$

Ortadaki değeri çıkışa verilir $\Rightarrow \text{Ç} = 3$

Diziyi eski haline getirilir $\Rightarrow G = \{1,9,5,3,2\}$

Sağa kaydırma yapılır ve yeni değeri eklenir (Yeni değer 4 olsun) $\Rightarrow G = \{4,1,9,5,3\}$

Aynı işlemler bu dizi için de yapılır.

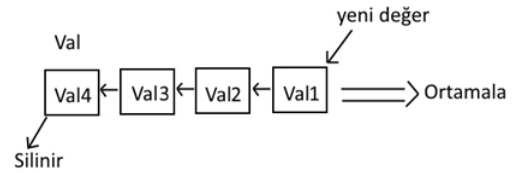
Ortanca Filtrenin diğer filtrelerden farkı sadece var olan değerleri kullanarak düzeltme yapabmesidir. Yani çıkışta görülecek her değer aslında bir şekilde girişte görülen değerdir.

Dizi içerisindeki ekstrem değerler örnekte görüldüğü üzere filtre içinde hiçbir zaman ortalama değer olmadığından çıkışa verilmemektedir. Burada dikkat edilecek husus ara bellek (buffer) büyüklüğüdür.

2.1.2. Hareketli ortalamalar filtresi (moving average filter)

Bu filtre için sıklıkla kullanılan isimler, Ortalama Filtresi (Mean Filtering), Yumuşatma (Smoothing), Ortalama (Averaging), Kutu Filtreleme (Box Filtering)'dir. Ortalama Filtresi, görüntü ve sinyal işleme konusunda yumuşatma işleminin basit ve uygulanması kolay bir tekniğidir. Diğer bir ifade ile bir piksel ve diğerleri arasındaki değişim miktarını azaltma işlemi yapmaktadır. Çoğunlukla gürültüyü azaltma amacıyla kullanılır.

Hareketli Ortalamalar Filtresi sinyal işlemede sıklıkla kullanılan bir yöntemdir. Bu filtre daha çok "Moving Average Filter" ismiyle kullanılmaktadır. Filtre isminden de anlaşıldığı gibi devamlı ortalamalar alan bir filtredir. Her yeni değer alındığında en eski değer bırakılır, yeni gelen değer ile ortalama tekrar alınır ve bu çıkış olarak verilir. Yani her alınan değerde ortalama değer güncellenmektedir [16]. Şekil 1'de bu durum daha net görülmektedir.



Şekil 1. Ortalama hesaplama

2.1.3. Kalman Filtresi (Kalman Filter)

Kalman Filtresi modellenmiş bir sistemin bir önceki giriş çıkış verilerine göre sistemin durumlarını tahmin eden bir filtredir. Bu yapı, bir filtreden ziyade tahmin tekniği olarak söylenebilir. Eş zamanlı bir şekilde kullanılmasıyla, kullanılan algoritma ile sisteme tanımlanmış olan hata modelleri ve gürültüler sayesinde çok daha başarılı sonuçlar elde edilir. Birçok sisteme uygun geçerli bir formül yoktur. Bundan dolayı sisteme göre bir modelin oluşturulmasıyla, bu modele uygun parametreler belirlenir ve bu parametrelere göre de algoritma oluşturulması gerekir. Veri sayısı ile doğru orantılı olarak tahmin başarısı da artacaktır. Bu farklılığı, onu diğer filtreleme tekniklerinden ayırılmaktadır. Bu filtre Rudolf Kalman tarafından geliştirilerek, 1960'lı yıllarda uzay programı da dâhil birçok farklı alanlarda kullanılmıştır. Bazılarına göre 20. yüzyılın en büyük buluş tekniklerinden biridir [17].

Kalman filtresinin matematiksel hesabı ise en basit şekilde aşağıdaki gibidir.

$$\hat{X}_k = K_k \cdot Z_k + (1 - K_k) \cdot \hat{X}_{k-1}$$

X_k = Hesaplanan kalman değeri

K_k = Kalman kazancı

Z_k = Ölçüm sonucu elde edilen değer

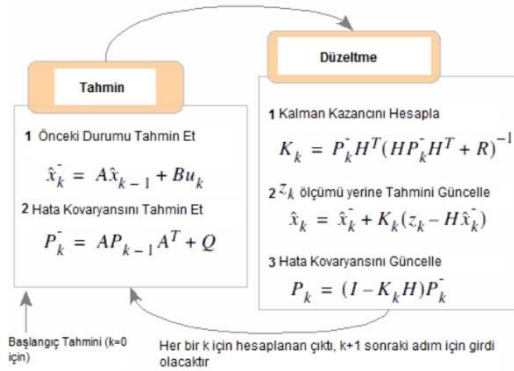
X_{k-1} = Bir önceki hesaplanan Kalman değeri

Burada K_k değeri dışındaki değerler bilinen değerlerdir.

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}$$

$$Z_k = Hx_k + v_k$$

Burada $w(k-1)$ terimi önceki işleme ait gürültüdür. vk ise ölçülen işleme ait gürültüdür. U_k kontrol sinyali olarak bilinir ve birçok uygulamada kullanılmaz. Aslında hesaplanan her değer bir önceki değer, kontrol sinyalinin ve bir önceki gürültü değerinin toplamından oluşmaktadır. Ölçülen değer bu değerle birlikte yeni gürültünün toplamını ifade eder. Formülde önemli bölüm A, B ve H parametreleridir. Matris katsayısı olarak ifade edilse bile çoğu zaman nümerik olarak bulunurlar. Burada $w(k-1)$ ve vk gauss gürültüsü olarak alınabilir. Bu teknik doğada birçok gürültüyü modellemek amacıyla oldukça sık kullanılan bir yöntemdir. Kalman filtresi uygulanırken iki aşama bulunur. Bu aşamalar tahmin ve düzeltmedir. Bu iki aşama aynı zaman diliminde ifade etmek gerekirse "k" değerinde hesaplanır (Her k değeri için). Formülleri aşağıdaki gibidir [17].



Şekil 2. Kalman filtre aşamaları

Hata Varyansı, Kalman Kazancını hesaplamak için kullanılan bir değerdir. Elde edilen değerlerden bağımsız bir şekilde güncellenerek sistemin modellenmesi aşamasında belirlenen R ve Q değerleri ile hesaplanır.

$Q > R$ durumunda; modelde hata daha baskındır.

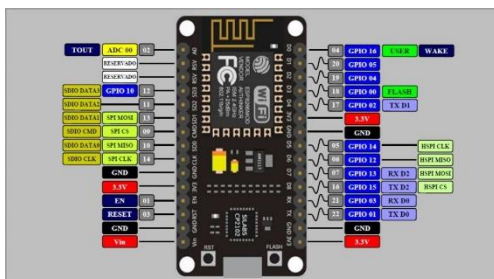
$R > Q$ durumunda; ölçümlerde hata daha baskındır.

$Q = R$ durumunda; hangisinin baskın olduğunun bilinmediği durumlarda kullanılır.

2.2. Sistemin donanım yapısı

Mekatronik sistemin çalışması için kullanılan araç gereçler bu kısımda anlatılmıştır.

NodeMCU: Üzerinde WiFi özelliğine sahip olması ile birlikte modüler yapıya sahiptir. Programlanması sayesinde Nesnelerin İnterneti uygulamaları düşük maliyet ile yapılabilmektedir. Analog giriş, dijital giriş/çıkış birimleri, PWM çıkış ve haberleşme desteği sağlar. NodeMCU mikrodenetleyiciye ait pin bağlantı yapısı Şekil 3'de verilmiştir.



Şekil 3. NodeMCU

NodeMCU sıklıkla kullanılan ve dahili WiFi modüle sahip bir esnelerin İnterneti denetleyici kartıdır. 160 MHz'de çalışan 32 bitlik bir RISC Tensilica LX106 mikrodenetleyicisi içermektedir.

NodeMCU'yu programlama işlemi için C++ dili ve Arduino IDE derleyicisi kullanılmıştır. Tablo 1'de denetleyiciye ait teknik özellikler görülmektedir.

Tablo 1. NodeMCU teknik özellikleri

Özellikler	Değer
MCU	32 bit Tensilica L106
İşlemci Frekansı	80/160 MHz
Input/Output	13xDIO
ADC Pin	1x10 bit (1V)
Çalışma Gerilimi	3.0 - 3.6 V
Çalışma Akımı	12-200 mA
Program Hafızası	4MB
WiFi	IEEE 802.11 b/g/n
Sleep Mode Akım	<10uA
Standby Mode Akım	<10mA

DS18B20 Sıcaklık Sensörü: Mikrodenetleyici ile 1 Wire arayüzünü kullanarak tek hat üzerinde haberleşir. ROM hafızasında üretim esnasında belirlenen ve tek olan 64 bitlik seri kod vardır ve bu kod ile birden fazla sensör aynı hat üzerinden haberleşebilir. Şekil 4' de DS18B20 sensörü, Tablo 2' de ise sensöre ait teknik özellikler verilmiştir.



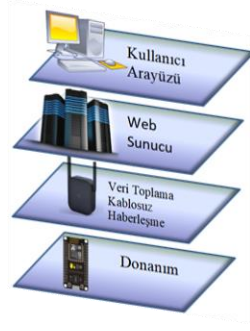
Şekil 4. DS18B20 sensörü

Tablo 2. DS18B20 teknik özellikleri

Çalışma Gerilimi	3V ~ 5.5V
Sıcaklık Algılama Aralığı (°C)	-55 ~ +125
Sensör Sıcaklık Hassasiyeti	±0.5 °C.

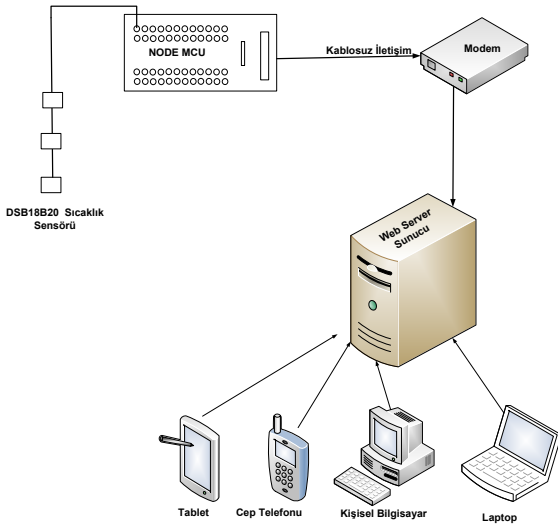
3. Bulgular

Oluşturulan prototip; donanım, C++ ile yazılan program parçası ile sensör verilerinin toplanması ve kablosuz haberleşme, web sunucu ve kullanıcı arayüzü olmak üzere dört katmanlı olarak tasarlanmıştır. Tasarım katmanları Şekil 5'de gösterilmiştir.



Şekil 5. Tasarım katmanları

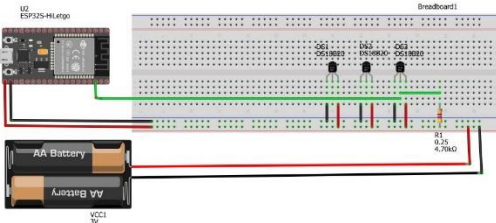
Belirlenen metodoloji çerçevesinde oluşturulan IoT tabanlı prototip sisteme ait blok diyagramı Şekil 6'da gösterilmiştir.



Şekil 6. Blok diyagramı

Donanım katmanında sıcaklık sensörü kullanılmıştır. Bu sensörlerden gelen verilerin sensör düğümü olarak kullanılan NodeMCU IoT nesnesine aktarılması sağlanmıştır. İkinci katmanda elde edilen sıcaklık verilerinin toplanması ve kablosuz teknoloji kullanılarak sunucu bilgisayara aktarılması sağlanmıştır. Üçüncü katmanda web sunucusu kullanılarak mobil platformlar dâhil verilere erişim için altyapı oluşturulmuştur. Son katmanda ise sunucu üzerinde çalıştırılan web tabanlı ara yüz yazılımı vasıtasıyla kullanıcının verileri izleyebilmesi sağlanmıştır. Bu veriler aynı zamanda sistemin veri tabanına kaydedilmektedir.

3.1. Sistemin aşamaları



Şekil 7. Oluşturulan sistem

Şekil 7'de görüldüğü üzere yapılan sistemin aşamaları şu şekildedir:

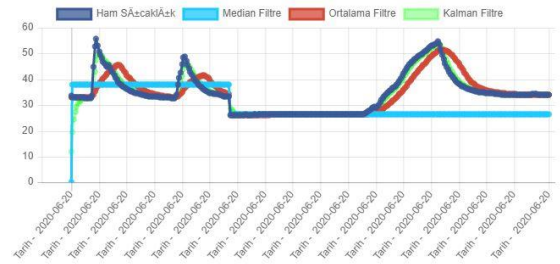
Donanım: Sıcaklığı ölçmek amacıyla üç adet DS18B20 sıcaklık sensörü kullanılmıştır. Pin tablosuna göre sensörlerden gelen güç ve input/output kabloları NodeMCU IoT nesnesine bağlanır. Sensör verileri yazılan C++ kodu ile elde edilmiştir.

Haberleşme: NodeMCU ile tümleşik WiFi modülü kablosuz haberleşme için kullanılarak WiFi modülünün izleme bilgisayarı ile haberleşmesi sağlanır.

Web Sunucu: Yerel Makinada php web Sunucu kullanılmıştır.

Arayüz Programı: Sensörlerden gelen verilerin kullanıcı tarafından görülebilmesi amacıyla web tabanlı bir arayüz uygulaması yazılmıştır. Yazılan uygulamanın internet erişimine açık izleyici bilgisayarından erişimi sağlanmıştır. Oluşturulan arayüzde sensörlerden gelen veriler ise 10 dakikada bir yenilenir. Oluşturulan arayüz programı Şekil 8'de gösterilmiştir.

Filtre Yazılımı İzleme



Şekil 8. İzleme programı

Veri Kontrolü: Oluşturulan prototipte her bir sıcaklık sensörü ölçülüp filtreleme yazılımları ile uygulanarak değerlerin veri tabanına kayıt edilmesi sağlanmıştır. Sıcaklık değerlerine müdahale edilerek sıcaklık değerlerinde değişim sağlanarak filtre yazılımlarının doğru çalışıp çalışmadığı kontrol edilmiştir ve web sayfasında grafiksel olarak gösterilmiştir.

```

1.i ← 0, j ← 0, sayacM ← 0, Buffer ← 0, HamVM ← Ham Veri Dizisi
2.Eğer sayacM değeri = 0 ise
3.   döngü()
4.   Sensör sıcaklık değeri oku
5.   HamVM dizisine okunan değeri ekle
6.   i ← i+1
7.   sayacM ← 1
8.   döngü sonu
9.eğer son
10.Eğer sayacM=1 ise
11.  döngü()
12.  döngü()
13.  Eğer HamV[i] değeri HamV[j] büyük ise
14.  Ölçülen 20 tane sensör değerini sırala
15.  eğer son
16.  j++
17.  döngü sonu
18.  j++
19.  döngü sonu
20.eğer son
21.Sıralanan sensör değerlerinin ortanca değeri alınır tempout=HamVM[10]
22.Bitir
    
```

Şekil 9. Median filtre yazılım sözde kodu

Şekil 9'da median filtre yazılımında, 20 boyutlu bir dizi tanımlanmıştır. sayacM değişkeni ile kontrol edilip ilk olarak 20 adet veri dizi içerisine atandıktan sonra sayacM değişkeni 1

yapılarak, sıralama yapılmıştır ve orta değeri alınarak sonuç olarak alınmıştır.

```

1. i ← 0, j ← 0, sayac ← 0, toplam ← 0, HamV ← Ham Veri Dizisi
2. Eğer sayac değeri = 0 ise
3.   döngü()
4.   Sensör sıcaklık değeri oku
5.   HamV dizisine okunan değeri ekle
6.   toplam=toplam+HamV [i]
7.   i ← i+1
8.   sayac ← 1
9.   döngü sonu
10.Eğer son
11. Eğer sayac=1 ise
12.   toplam=toplam+HamV[0]
13.   Satirbasi++
14.   döngü()
15.   HamV[j]=HamV[j+satirbasi]
16.   j++
17.   döngü sonu
18. Eğer son
19.Sensörlerin ortalaması ortalama=toplam/20
20. Bitir
    
```

Şekil 10. Hareketli ortalamalar filtre yazılım sözde kodu

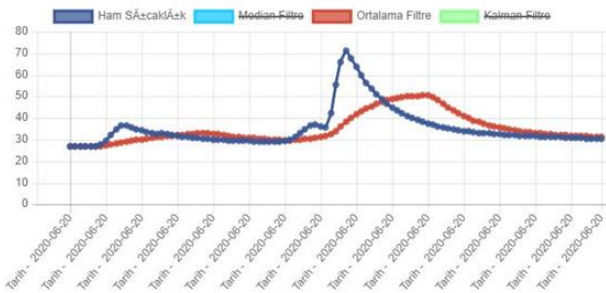
Şekil 10'da hareketli ortalamalar filtresinde 20 boyutlu bir dizi tanımlanmıştır. İlk olarak sensörden gelen 20 değer bu diziyeye atılmıştır,daha sonra hareketli ortalamalar filtre mantığına göre en eski değer bu diziden silinir ve yeni değer eklenir ardından ortalaması alınarak veri tabanına kayıt olarak gönderilir. Şekil 11'de Kalman filtresi sözde kodu verilmiştir.

```

1.kalmannew ← kalmanold, covnew← covold + 0,5,
2.kalmangain ← covnew/(covnew+0,9)
3.kalmancalculated←kalmannew+(kalmangain*(input-kalmannew))
4.covnew=(1-kalmangain)*covold
5.covold=covnew
6.kalmanold=kalmancalculated
7.return kalmancalculated
8.Bitir
    
```

Şekil 11. Kalman filtresi sözde kodu

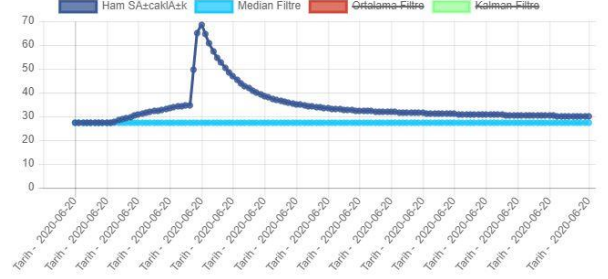
Filtre Yazılımı İzleme



Şekil 12. Ortalamalar filtre grafiği

Şekil 12'de ortalamalar filtresi ile yapılan test grafiği gösterilmektedir. Yaklaşık 5 dk sistem çalıştırılmış ve 105 tane örnek veri alınmıştır. Mavi renkli olarak gösterilen ham sıcaklık verisi, kırmızı olarak gösterilen ise ortalama filtre verisidir. Sıcaklık değerleri anlık yükseltilecek filtrenin tepkisi ölçülmüştür.

Filtre Yazılımı İzleme



Şekil 13. Median filtre grafiği

Şekil 13'de Median Filtresi ile yapılan test grafiği gösterilmektedir. Yaklaşık 5 dk sistem çalıştırılmış ve 137 tane örnek veri alınmıştır. Mavi renkli olarak gösterilen ham sıcaklık verisi, açık mavi renk olarak gösterilen ise Median Filtre verisidir. Sıcaklık değerleri anlık yükseltilecek filtrenin tepkisi ölçülmüştür.

Filtre Yazılımı İzleme



Şekil 14. Kalman filtre grafiği

Şekil 14'de kalman filtresi ile yapılan test grafiği gösterilmektedir. Yaklaşık 5 dakika sistem çalıştırılmış ve 115 tane örnek veri alınmıştır. Mavi renkli olarak gösterilen ham sıcaklık verisi, yeşil renk olarak gösterilen ise kalman filtre verisidir. Sıcaklık değerleri anlık yükseltilecek filtrenin tepkisi ölçülmüştür.

4. Sonuç

Bu çalışmada IOT tabanlı gerçek zamanlı sistem tasarımı gerçekleştirilmiştir. Oluşturulan sistemde median, ortalamalar ve kalman olmak üzere 3 farklı filtre yazılımı uygulanmıştır. Ortamın sıcaklık değerleri ölçülmüş ve ölçülen sıcaklık değerlerine median, ortalamalar ve kalman filtresi uygulanmıştır. Ortam sıcaklığına uygulanan her bir filtre grafiksel olarak gösterilmiştir. Anlık olarak sıcaklık değerleri değiştirilerek uygulanan filtrelerin etkisi grafiksel olarak analiz edilmiştir. Filtre uygulanmayan sıcaklık değeri ile uygulanan sıcaklık değerleri kıyaslanmıştır.

Kaynakça

- [1] Hammami, A. 2019. Smart Environment Data Monitoring. 2019 International Conference on Computer and Information Sciences (ICCIS), 3-4 April, Sakaka, Saudi Arabia, pp. 1-6.
- [2] Ghosh, R., Gardner, J.W., Guha, P.K. 2019. Air Pollution Monitoring Using Near Room Temperature Resistive Gas Sensors: A Review. IEEE Transactions on Electron Devices, 66(8), 3254-3264.
- [3] Hammami, A. 2019. Smart Environment Data Monitoring. 2019 International Conference on Computer and Information Sciences (ICCIS), 3-4 April, Sakaka, Saudi Arabia, pp. 1-6.
- [4] Majumder, A.J.A., Mcwhorter, T.M., Ni, Y., Nie, H., Iarve, J., Ucci, D.R. 2019. sEmoD: A Personalized Emotion Detection Using a Smart Holistic Embedded IoT System. 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), 15-19 July, Milwaukee, WI, USA, pp. 850-859.
- [5] Shah J., Mishra, B. 2016. Customized IoT enabled wireless sensing and monitoring platform for smart buildings. Procedia Technology, 23, 256-263.
- [6] Lai, X., Yang, T., Wang, Z., & Chen, P. 201). IoT Implementation of Kalman Filter to Improve Accuracy of Air Quality Monitoring and Prediction. Applied Sciences, 1-23.
- [7] Üçgün, H., Gömbeci, F., Yüzgeç, U., & Yalçın, N. 2020. IoT Tabanlı Platform ile Gerçek Zamanlı İç Ortam Hava Kalitesi İzleme Sistemi. BŞEÜ Fen Bilimleri Dergisi, 370-381.
- [8] Monteiro, P., Zanin, M., Ruiz, E., Pimentão, J., & Sousa, P. 2018. Indoor Temperature Prediction in an IoT Scenario. Sensors, 1-18.
- [9] Ayele, T., & Mehta, R. 2018. Real time temperature prediction using IoT. 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018) (s. 114-1117). Coimbatore, India: IEEE.
- [10] Taştan, M. 2019. An IoT Based Air Quality Measurement and Warning System for Ambient Assisted Living. Avrupa Bilim ve Teknoloji Dergisi, 960-968.
- [11] Ahıska, R., Fidan, U., & Dişlitaş, S. 2008. Farklı Sıcaklık Kontrol Sistemlerinin Termoelektrik Tıp Kitinin Performansına Etkisi. Gazi Üniv. Müh. Mim. Fak. Der., 441-447.
- [12] Altun, A., & Allahverdi, N. 2007. Filtreleme Teknikleri İle İyileştirilmiş Parmakizlerini Yapay Sinir Ağları İle Tanımada Yeni Bir Yaklaşım. Gazi Üniv. Müh. Mim. Fak. Der., 227-236.
- [13] Pamadi V., & Nickerson B. G. 2015. Getting Started With 1-Wire Bus Devices, Faculty of Computer Science University of New Brunswick Fredericton, N.B. E3B 5A3 Canada.
- [14] Bütüner, R., & Calp, M. 2021. Covide Pandemisinde Hasta Takip Sistemi. Lale Oraloğlu International Congress On Culture, Arts And Multidisciplinary Studies (s. 355-360). Online: Babil Yayınevi.
- [15] Memiş, S., & Erkan, U. 2021. Different Adaptive Modified Riesz Mean Filter For High-Density Salt-and-Pepper Noise Removal in Grayscale Images. Avrupa Bilim ve Teknoloji Dergisi, (23), 359-367.
- [16] Kaya, S. B., & Alkar, A. Z. 2014. Location estimation improvement by signal adaptive RSSI filtering. In 2014 22nd Signal Processing and Communications Applications Conference (SIU) (pp. 1183-1186). IEEE.
- [17] Orderud, F. Doğrusal Olmayan Ölçümlü Durum Uzay Modelleri için Kalman Filtresi Kestirimi Yaklaşımlarının Karşılaştırılması.