**Nesnelerin İnterneti Temelli**

**Akıllı Ev Sistemi**

**2024**

**LİSANS TEZİ**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**

**İlyas ARSLAN**

**Nesnelerin İnterneti Temelli**

**Akıllı Ev Sistemi**

**İlyas ARSLAN**

**Karabük Üniversitesi**

**Mühendislik Fakültesi**

**Elektrik-Elektronik Mühendisliği Bölümünde**

**Lisans Tezi**

**Olarak Hazırlanmıştır**

**KARABÜK**

**Mayıs 2024**

İlyas Arslan tarafından hazırlanan “Nesnelerin İnterneti Temelli

Akıllı Ev Sistemi” başlıklı bu tezin Lisans Tezi olarak uygun olduğunu onaylarım.

..../..../2024

Dr. Öğr. Üyesi YUNUS EMRE KARASU

..…….………...

Tez Danışmanı,

Karabük Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü

Ünvan. Ad SOYAD ..…….………...

Karabük Üniversitesi Elektrik Elektronik Mühendisliği Bölümü

Ünvan. Ad SOYAD ..…….………...

Karabük Üniversitesi Elektrik Elektronik Mühendisliği Bölümü

# ÖZET

**Lisans Tezi**

**Nesnelerin İnterneti Tabanlı Akıllı Ev Sistemi**

**İlyas ARSLAN**

**Karabük Üniversitesi**

**Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü**

**Tez Danışmanı**

**Dr. Öğr. Üyesi YUNUS EMRE KARASU**

**Haziran 2024, 27 sayfa**

Bu çalışmada erişilebilirliğini durum şartlarına göre ayarlayabilen bir Nesnelerin İnterneti Temelli Akıllı Ev uygulaması geliştirilmiştir. Akıllı ev uygulamalarında amaç cihazların birbiri ile haberleşerek daha az insan müdahalesi ile çalışmaya devam edebilmeleridir. Bu da bize enerji, zaman, erişilebilirlik açısından faydalar sağlamaktadır. Aynı zamanda daha önce yapılmış çalışmalar incelendiğinde, beklenmedik bağlantı kesintileri durumunda erişilebilirliğini kaybeden sistemler gözlemlenmiştir. Genelde tek bir bulut hizmeti veya iletişim protokolü izleyen çalışmalar bu konuda başarısız kalmıştır. Öne sürülen bu çalışmada Nesnelerin İnterneti Temelli Akıllı Ev Sistemi bu konuda iyileştirmeler içermektedir. Beklenmedik bağlantı kayıpları durumlarında sistem ve cihazların verdiği hizmet kesilmeyecek, kullanıcı cihazlara erişimini kaybetmeyecek. Sistem bağlantısı devam ettiği sürece kullanıcı evin sıcaklık ve nem değerleri hakkında bilgi alabilecek, ışıklandırmaları kontrol edebilecek, havalandırma ve kapı için kullanılan, dc ve servo motorların kontrolünü sağlanmaktadır. Gerçekleştirdiğim çalışmada; 32-bit ESP32 modülü ve Blynk Bulut geliştiricisi kullanarak, ev yaşantısı daha güvenli hale getirirken aynı zamanda konforlu hale de getirecek gerçek zamanlı veri izleme ve kontrol uygulaması hayata geçirilmiştir. Blynk Bulut servisiyle tasarlanan internet bağlantısına sahip mobil veyahut diğer cihazlarla iletişime geçebilen sistem sayesinde kullanışlı bir arayüz ile sisteme erişim sağlanmaktadır. İnternet kesintisi gibi bir durum yaşanırsa akıllı-ev sistemi yerel ağda kendisine bir istasyon seçip yeni bir kullanıcı arayüzü oluşturmaktadır, ve bu arayüz üzerinden kullanıcı tekrar kontrol sağlayabilecek. Evdeki modemin bozulması veya erişilebilir olmaması gibi bir durum yaşanırsa sistem ağa ulaşamayacağı için bu sefer bir istasyon değil, kablosuz erişim noktası gibi davranacak ve bağlantı sağladığınız takdirde tekrardan aynı arayüz ile erişilebilir olmaktadır.

**Anahtar Kelimeler :** IoT, Blynk, ESP32, Akıllı ev.

# ABSTRACT

**B. Sc. Thesis**

**Internet of Things Based**

**Smart Home System**

**İlyas ARSLAN**

**Karabük University**

**Faculty of Engineering Department of Electrical-Electronics Engineering**

**Thesis Advisor**

**Asst. Prof. Dr. YUNUS EMRE KARASU**

**June 2024, XX pages**

In this study, an Internet of Things (IoT)-Based Smart Home application has been developed, which can adjust its accessibility according to the conditions. The aim of smart home applications is for devices to communicate with each other and continue to operate with less human intervention. This provides us with benefits in terms of energy, time, and accessibility. At the same time, previous studies have shown that systems lose their accessibility in the event of unexpected connection interruptions. Studies that generally follow a single cloud service or communication protocol have failed in this regard. The proposed study includes improvements in the IoT-Based Smart Home System in this context. In the event of unexpected connection losses, the services provided by the system and devices will not be interrupted, and the user will not lose access to the devices. As long as the system connection continues, the user will be able to obtain information about the temperature and humidity values of the house, control the lighting, and manage the DC and servo motors used for ventilation and doors. In the work I have carried out, using the 32-bit ESP32 module and Blynk Cloud developer, a real-time data monitoring and control application has been implemented, which makes home life safer and more comfortable. Thanks to the system designed with the Blynk Cloud service, which can communicate with mobile or other devices with internet connectivity, access to the system is provided with a user-friendly interface. In the event of an internet outage, the smart home system will select a station on the local network and create a new user interface, allowing the user to regain control through this interface. If the home modem is broken or not accessible, the system will act as a wireless access point instead of a station, and once connected, the user will again be able to access the same interface.

**Key Words :** IoT, Blynk Cloud, ESP32, smart home.

# TEŞEKKÜR

Bu tezin hazırlanmasında ve tamamlanmasında değerli katkılarını sağlayan, adım adım çalışmamı gözlemleyip yardımcı olan değerli hocam Dr. Yunus Emre KARASU’ ya en içten dileklerle teşekkür ederim. Ayrıca, tüm eğitim hayatım boyunca arkamda duran binbir emekle eğitim hayatımı destekleyen aile fertlerime anneme ve babama teşekkürler ederim. Onların sevgisi ve inancı beni bu yolculukta daha azimli kılmıştır. Bu tez, onların katkıları ve destekleriyle ortaya çıkmıştır. Hepinize minnettarım.

Saygılarımla

*“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim.”*

İlyas ARSLAN

# İÇİNDEKİLER

**Sayfa**

[ÖZET iii](#_gjdgxs)

[ABSTRACT v](#_30j0zll)

[TEŞEKKÜR vii](#_1fob9te)

[İÇİNDEKİLER vii](#_3znysh7)

[ŞEKİLLERİN LİSTESİ x](#_2et92p0)

[BÖLÜM 1 1](#_tyjcwt)

[1.](#_3dy6vkm) GİRİŞ 1

[1.1.](#_1t3h5sf) Nesmelerin İnterneti 2

[1.2.](#_1t3h5sf) Nesmelerin İnterneti Tarihçesi 2

[1.3.](#_4d34og8) Uygulama Alanları 3

[1.4.](#_2s8eyo1) Nesnelerin İnterneti Katmanları 3

[1.5. Akıllı Ev Sistemleri 4](#_2s8eyo1)

[1.6. Akıllı Ev Sistemleri Kaynak Araştırması 4](#_2s8eyo1)

[1.6.1. İnternet Tabanlı Sistemler 5](#_17dp8vu)

[1.6.1. Bluetooth Tabanlı Sistemler 5](#_17dp8vu)

[1.6.1. GSM Tabanlı Sistemler 5](#_17dp8vu)

[1.6.1. Zigbee Tabanlı Sistemler 6](#_17dp8vu)

[1.6.1. RFID Tabanlı Sistemler 6](#_17dp8vu)

[BÖLÜM 2 8](#_3rdcrjn)

[2.](#_26in1rg) MATERYAL VE METOT 8

[2.1.](#_lnxbz9) Kullanılan Cihaz ve Sensörler 8

[2.1.1. ESP-32 8](#_17dp8vu)

[2.1.2. MG905 Mikro Servo Motor 10](#_17dp8vu)

[2.1.3. DHT22 Sensörü 11](#_17dp8vu)

[2.1.4. Diğer Materyaller 12](#_17dp8vu)

[2.2. Metot](#_35nkun2) 13

[3.2.1 Sistemin Bağlantı Şeması 13](#_17dp8vu)

[3.2.2 Sistemin Besleme ve Şeması 14](#_17dp8vu)

[3.2.3 ESP-32 Yazılımı ve Kod Derleyici 14](#_17dp8vu)

[3.2.3 Sistemin Erişilebilirliği ve Kullanışı Üzerine 16](#_17dp8vu)

[3.](#_26in1rg) DEĞERLENDİRME VE SONUÇ 19

[4. ÖNERİLER 20](#_26in1rg)

# ŞEKİLLERİN LİSTESİ

| **Şekil** | **Sayfa** |
| --- | --- |

[Şekil 1.](#_44sinio) Nesnelerin İnterneti 3

[Şekil 2.](#_2jxsxqh) Nesnlerin İnterneti Katmanlar 3

[Şekil 3.](#_2jxsxqh) ESP-32 Pin Diagramı 9

[Şekil 4.](#_z337ya) Servo Motor (5v) 11

[Şekil 5.](#_3j2qqm3) Sıcaklık Sensörü 11

[Şekil 6.](#_1y810tw) Diğer Materyaller 12

[Şekil 7.](#_4i7ojhp) Sistem Bağlantı Şeması 13

[Şekil 8.](#_2xcytpi) Blynk İletişim Şeması 16

[Şekil 9.](#_1ci93xb) Blynk Arayüzü 16

[Şekil 10.](#_3whwml4) Web Server İletişim Şeması 18

[Şekil 11.](#_2bn6wsx) Web Serve Kullanıcı Arayüzü 18

[Şekil 12.](#_qsh70q) Erişim Noktası İletişim Şeması 18

[Şekil 12.](#_qsh70q) Erişim Noktası 18

BÖLÜM 1

# GİRİŞ

Nesnelerin İnterneti terimi genellikle, ağ bağlantısı ve hesaplama kapasitesinin normalde bilgisayar olarak kabul edilmemiş nesneler, sensörler ve günlük eşyalar tarafından gerçekleştirilmesi ve bu cihazların minimum insan müdahalesi ile veri üretmesine, değiştirmesine ve tüketmesine olanak sağlayan sistemler anlamına gelir[1]. Hayatımızın her adımında bu cihazlarla elde edilen verileri toplayabilir, değiştirebilir ve kullanabiliriz. Nİ cihazlarının büyük ölçekte kullanılması yaşam tarzımızın birçok yönünü değiştiriyor. Tüketiciler için, İnternet destekli cihazlar, ev otomasyon bileşenleri ve enerji yönetimi aygıtları gibi yeni Nİ ürünleri bizi “akıllı ev” vizyonuna doğru yönlendiriyor [2]. Akıllı ev sistemleri ve Nİ teknolojisi bir çok şekilde hayatımızı kolaylaştırmakta güvenlik, erişilebilirlik ve kontrol. Bu fonksiyonlarını yerine getirilebilmesi için Akıllı ev mikro bilgisayarları kullanmak gerekecektir. Akıllı Ev Mikro Bilgisayarları (AEMB), tüm akıllı ev sistemini otomatikleştirmek ve kontrol etmek için diğer cihazlara bağlı olan küçük boyutlu bilgisayarlardır. Bunlar, programlamayı ve diğer devreye entegre edilmesini kolaylaştıran tamamlayıcı bileşenlerle bir mikro kontrolörden oluşur[3]. Akıllı ev sistemlerinin ortaya çıkmasında, evlerimizde değilken de ev içerisindeki cihazlara müdahale edebilme kaygısı da etkili olmuştur. Aynı zamanda farklı iletişim sistemleri kullanarak, mikro kontrolör sayesinde cihazlarımızı her nerede olursak olalım uzaktan kontrol edebilme şansı bulacağız. Güncel uygulamalardan bir kaçı şu şekilde; Uğur Yüzgeç ve Ömer Aba tarafından[4], Raspberry Pi mikrodenetleyici tabanlı bir akıllı ev uygulaması gerçekleştirilmiştir. Bu çalışma doğrultusunda kablosuz ağ bağlantısı üzerinden akıllı ev sisteminin iletişimini sağlanmıştır, bu sayede kapı, garaja otomatik giriş, yangın/gaz kaçağı durumu ve su baskını ikazı, otomatik aydınlatma, hırsız alarmı, deprem durumu gibi senaryolar akıllı ev sistemi ile birlikte başarılı şekilde kontrol sağlanmıştır.

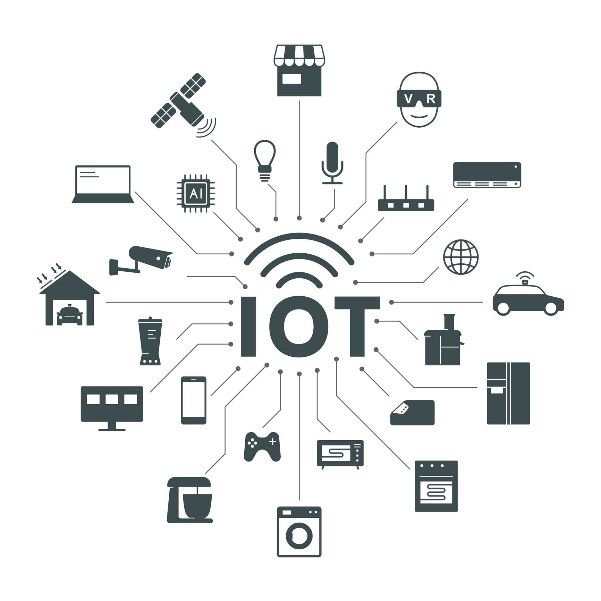
## 1.1. Nesnelerin İnterneti (IoT)

Günlük hayatta kullandığımız nesnelerden oluşan, devre, yazılım, sensör benzeri bileşenler içeren, bu bileşenler sayesinde veri toplayabilen, değiştirebilen hatta kontrol edebilmeye olanak sağlayan bir ağ olarak düşünebiliriz. Temelde insan müdahalesi olmaksızın etkileşime geçebilmeleri gayesiyle gelişmiş, gelişmeye devam etmektedir. Nİ gerçek hayatın ve fiziksel aktivitelerin sanal dünyayla etkileşimi sağlar [5]. Öncü uygulamalar sağlık, ulaşım ve otomotiv sektörlerinde gerçekleşmiştir ve git gide hayatımızda çok daha fazla yer edinmeyi başarmıştır. Bugün günümüzde milyarlarca cihaz birbiriyle iletişime geçebilmekte ve fiziksel olarak hayatımıza müdahale edebilmekte. Nesnelerin interneti kavramının günümüzde ulaştığı noktaya vakıf olmak adına, yıllar içerisinde yaşadığı değişimi ve aynı zamanda mimarı yapısını da incelemek gerekli olacaktır. Nİ sistemlerin gelişimi altyapı, iletişim, arayüzler, protokoller ve standartlar gibi birçok konuyu da içermektedir [6].

## 1.2. Nesnelerin İnterneti Tarihçesi

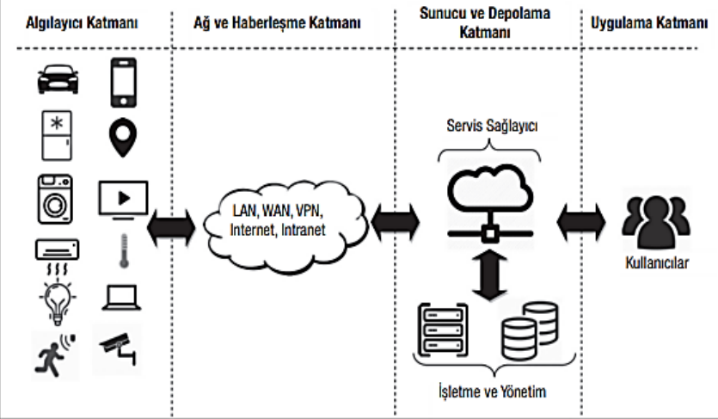
Öncelikle hayatımızdaki ilk örnekleri ele almakta fayda olacağını düşünmekteyim. Her ne kadar karmaşık gibi gözükse de temelde hayatımızın içinde oldukça yer bulmuştur internet etkileşimli nesneler, hatta ilk örnek olarak kabul edilebilecek olan nesnelerin interneti öğreği bir içecek dolabıdır. İçerisindeki içeceğin stoğunu ve soğukluk derecesini öğrenebilmek adına Carnegie Mellon Üniversitesi öğrencileri tarafından 1982 yılında tasarlanmıştı, sadece sensörler yardımıyla veri alabilmekteydiler [7]. 1990 yılında ise John Romkey, internete bağlanan ilk cihazı tasarladı bu cihaz internet üzerinden açılıp kapatılabilen bir tost makinesiydi ardından 1994 yılında ise neredeyse gerçek zamanlı veri iletimi sağlayan WearCam geliştirilmişti[8]. Tanımın ilk dile getirildiği yer ise MIT, AutoIDCentre direktörü Kevin Ashton tarafından dile getirilmiştir. Aynı yıl içerisinde, küresel RFID tabanlı bir ürün tanımlama sistemi geliştirildi[9]. Tüm bu gelişmelerin ardından nesnelerin interneti ticarileşmeye başladı ve ardı arkası kesilmeyecek şekilde gün be gün daha da gelişerek günümüze kadar varlığını sürdürdü.

## 1.3. Uygulama Alanları

İnternet ve mobil cihaz teknolojisinin hızla gelişmesi ve her yere uygum sağlayabilme kabiliyeti sayesinde nesnelerin interneti de hayatımız içerisinde çok fazla uygulama alanına sahip olmuştur. Bu şekilde hızlı bir uygulama kabiliyetine sahip olan nesnelerin interneti akıllı ev, akıllı şehir, akıllı park, akıllı sokak, akıllı tarım, akıllı hayvancılık, akıllı tedarik uygulamaları, akıllı sağlık hizmetleri, akıllı mağazalar, akıllı askeri kullanımı gibi birçok alanda yaygın bir kullanıma sahiptir .



## 1.4. Nesnelerin İnterneti Katmanları

Iot mimarisi genel olarak şu şekilde kabul görmektedir; 4 katmandan oluştuğu kabul etmekteyiz bunlar sırasıyla Algılama, Ağ, Sunucu ve Uygulama katmanlarıdır. Şekilde gösterilmiştir.

Algılama katmanı aynı zamanda cihaz katmanı olarak da bilinmektedir[10]. Bu katmanda barkod etiketleri, RFID etiketleri, kamera, sensör, terminaller ve sensör ağı ve benzerleri gibi nesneler bulunmaktadır. Algılama katmanında bahsettiğimiz bu nesneler aslında beş duyu organı gibi de düşünülebilir. Amaç nesneleri tanımlamak veyahut bilgi toplamaktır[11]. Ağ katmanı sisteme bağlı tüm nesnelerin birbirleriyle iletişim kurdukları katmandır, bu katman aracılığıyla nesneler birbirleriyle veri paylaşmasını sağlar. Temel fonksiyon veri iletmek ve işlemektir. Bunun gerçekleşmesi için güçlü bir ağ kullanmak gerekmektedir. Algılama katmanında elde edilen bilgilerin, verilerin iletimi bu katmanda gerçekleşmekte. Bu işlem farklı metotlarla sağlanabilir; mobil ağlar, wi-fi, bluetooth, kızılötesi, zigbee, lorawan vb. gibi. Uygulama katmanı, kullanıcıların ihtiyaçlarına göre hizmetler sağlar.

## Akıllı Ev Sistemleri

Akıllı evler, evlerini git gide çok daha konforlu, işlevsel ve güvenli hale getirmek isteyen insanlar tarafından talep edildiği sürece geliştirilmiş ve geliştirilmeye devam etmektedir. Talep arttıkça pazar gelişmiş ve yeni üreticiler, girişimler ortaya çıkmıştır. Talebin bu denli artmasının sebebi tabi ki de uçsuz bucaksız fonksiyonlar; evinize gitmeden sıcaklığı hakkında veri alabilmek ve uzaktan kontrol ederek istediğiniz sıcaklığa getirebilecek olmak ve hatta bunları yaparken yapar zeka sayesinde tasarruf da edecek olmanız.

## Akıllı Ev Sistemleri Literatür Araştırması

İhtiyaçlara ve taleplere göre değişik fonksiyonlara sahip birbirinden farlı bir sürü sistemle karşılaşmak da mümkündür. Uzaktan etkileşime geçebilenler, yapay zeka hizmeti sunanlar, kumanda gibi bileşenler ile kontrol edilen sistemler, her ne kadar farklı gözükseler de sundukları şey “konfor”. Bu bölümde farklı iletişim protokolleri doğrultusunda tasarlanmış akıllı ev sistemlerini inceleyecek, bir literatür taraması yapacağız.

### İnternet Tabanlı Sistemler

İnternete bağlanabilen telefon, bilgisayar, tablet gibi cihazların hali hazırda gelişen teknoloji ile birlikte popülerlikleri de arttı. Neredeyse hepimizin bunlara erişim sağlaması oldukça olağan. Zaten hali hazırda bunlara erişimimiz varken akıllı ev sistemlerinin kontrollerini de bu tabanda hazırlamak erişilebilirlik açısından daha doğru gibi. Örneklere bakacak olursak; Mehmet Taştan[12], hazırladığı çalışmada Android uygulama üzerinden internet bağlantısı kullanarak Blink bulut sistemine bağlanmış ve tasarladığı sistem içerisindeki cihazların kontrolünü sağlamıştır. Nesnelerin internete bağlanabilmesi için ESP-32 mikro kontrolör kullanmıştır. Kendisi yaptığı bu tercihin sebebi olarak; mikro usb port sayesinde rahatlıkla programlanabilir ve WiFi, bluetooth, CAN, Ethernet, sıcaklık sensörü, manyetik alan sensörü ve touch sensör gibi bir çok gömülü özelliğe sahip olmasını gösteriyor.

### Bluetooth Tabanlı Sistemler

Bluetooth çok daha kısıtlı bir çevre için kullanılması mantıklı olan iletişim protokolüdür zira öyle uzun mesafeler katetemez. Örnek olarak bir çalışma inceleyelim; M. Asadullah ve K. Ullah[13], hazırladıkları ev sisteminde kullandıkları geliştirme kartı ile telefonları arasındaki bağlantıyı bluetooth ile sağlamışlar. Bluetooth modülü olarak HC-06 ve geliştirme kartı olarak Arduino Uno tercih edilmiştir, kullandıkları ultrasonik sensör ile su seviyesi tespiti ve otomatik sulama sistemi inşa edebilmişlerdir. Çalışmanın sonunda bu yöntemle sadece kısa mesafede cihaz kontrolü sağlanabildiğinden de bahsedilmiştir.

### GSM Tabanlı Sistemler

Gsm, mobil iletişim sistemi internet bağlantımızın mümkün olmadığı durumlarda yine bize oldukça düşük maaliyetlere sebep olacak olan ses iletimi ve hücresel veri aktarımın sağlanabileceği bir iletişim yöntemdir. Mesafe fark etmeksizin bu işlemi gerçekleştirebilecek olmamız da önemli bir etmendir. Örnek inceleyecek olursak; Yıldırım ve Börekçi[14], yaptıkları çalışmada SIM800 GMS modemi kullanarak iletişimi sağlamış ve başarılı bir akıllı ev sistemi kurmuştur. Modem için kullanacağı sim kartı mevcut hizmet veren firmalardan abonelik karşılığı temin edilmiş, çekim gücünü arttırmak için harici bir de GSM anteni temin edilmiştir. Fiyat performans açısından da ülkemizde imei kayıtlı en uygun modem olduğu için tercih edildiğine de değinilmiştir.

### Zigbee Tabanlı Sistemler

Zigbee teknolojisi wi-fi ile birlikte en yaygın kullanılan veri iletişim protokolleridir. Wi-Fi’dan daha yavaş bir iletim hızına sahiptir, buna nazaran güç tüketimi de daha azdır. Zigee ekipmanların aktiviteleri çok kısıtlıdır. Ekipmanlar genellikle uyku modu diye de bilinen düşük güç tüketimi modunda çalışır. Bundan dolayı Zigbee ekipmanları yıllarca batarya değişikliğine ihtiyaç duymadan çalışma yeteneğine sahiptir[15]. En önemli özelliği tüm işlemin örgü ağ içerisinde gerçekleşmesidir. Veri büyük bir ağda hedefe ulaşana kadar cihazdan cihaza iletilir. Koordinatör, yönlendirici ve zigbee standartlarına uyumlu cihazlar ile çalışmaktadır. Zigbee kullanılarak hazırlanmış örneğe göz atalım; Tarek H. [16], hazırladığı Zigbee ağı üzerinden birbirlerine bağlanan ve akıllı düğümlere dayanan tasarımla birlikte, evin akım, gerilim değerlerini takip edebilmiş, yangın algılama, duman algılama gibi fonksiyonları sağlamıştır. Sorumlu sunucu olarak ise Raspberry Pi kullanarak sistemin buluta bağlantısını gerçekleştirmiş ve bu şekilde istediğimiz yerden evin durumu hakkında bilgi sahibi olabilme şartı sağlanmıştır.

### RFID Tabanlı Sistemler

RFID temelde bir nesnenin üzerine fiziksel bir etiket yerleştirilmesi şeklindedir, bu etiketin üzerinde bir mikroçip ve anten bulunmaktadır. Bu etiketlerden veri okunması radyo frekansları ile mümkündür. Genelde lojistik, üretim, sağlık gibi sektörlerde stok kontrol için kullanılmakta. RFID teknolojisi ilk olarak, 1945 yılında Sovyetler Birliği için radyo dalgalarını sesli bilgilerle yeniden ileten bir casus aracı olarak ortaya çıkmıştır [17]. Evlerimizde kullanacağımız bir yapıya pekte sahip değil, literatür tarama sırasında bir örneği ile karşılaşmadım, belki buzdolaplarında güncel kullandığımız ürünlerin stok durumu hakkında bilgi edilmek için bir sistem düşünülebilirdi. Farklı bir örnek verecek olursak; Günümüzde işyerlerinde, spor salonunda kullandığımız üyelik kartı ve personel kartı da RFID tabanlı çalışır, kimliklerimiz karta tanımlanmıştır ve okuyucuya bağlı işlemci kartımızı okuttuğumuzda bizleri tanımaktadır.

BÖLÜM 2

# Materyal ve method

Bu bolümde bir maket ev üzerinde tasarlanmış akıllı ev sistemini ve işleyişini gözlemleyeceğiz. Kullanılan materyaller ve çalışma prensipleri hakkında, aralarındaki iletişim protokolü detayları, yazılım geliştirme ortamı, programlama dilleri ve kullanılan bulut sistemi hakkında bilgi verecek, yapılan iyileştirmeler ve tercihlerin sebeplerine değineceğim.

## Kullanılan Cihaz ve Sensörler

***Kullanılan Cihaz ve Sensör Listesi:***

| *ESP32 Mikro Kontrolör* |
| --- |
| *MG90S Mikro Servo Motor* |
| *DHT22 Sıcaklık ve Nem Kontrolü Sensörü* |
| *DC Motor Pervane 3-6V* |
| *Standart 5V Led x2* |
| *2x 3.7V Li-ion Pil ve Pil Yatağı* |
| *L298N Motor Sürücü Kartı* |
| *Bread Board* |

### ESP-32

Düşük güç tüketimi ve birçok bağlantı pinine sahip olması sebebiyle oldukça sık tercih edilmekte olan bir kontrolcü. Bir micro usb bağlantısına sahiptir, programlanabilmesi için kolaylık sağlar ve isteğe bağlı olarak 5v besleme de bu girişle sağlanabilir. Gömülü Wifi ve Bluetooth modüllerine sahip olması da tercihen diğer kontrolörlerin önüne geçmesini sağlamakta. ESP32’ ye ait pin yapısı bir sonraki sayfada verilmiştir.

ekran görüntüsü, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

Şekil 3 ESP-32 Pin Diagramı

* GPIO PİNLERİ

Dijital giriş ve çıkış işlevlerini yerine getiren pinlerdir, aynı zamanda analog giriş pini olarak kullanmak da mümkündür.

* SENSOR VP/VN

Pozitif voltaj ve negatif voltaj anlamına gelmektedir analog sensör bağlantıları için kullanılan pinlerdir, Analog sensör genelde belirli bir voltaj aralığınki verileri iletecek şekilde tasarlanır. Bu pinler de sıkıntısız çalışması için gereken referans voltajı ayarlamak içindir.

* XTAL PİNLERİ

Bir kristal osilatör bağlamak istediğimizde kullanmamız gereken pinlerdir.

* TOUCH PİNLERİ

Fiziksel etkileşimi algılamak için kullanılabilecek pinlerdir. Dokunmatik duyarlılık algılayan sensörlerin bağlantıları için tasarlanmıştır.

* HSPI & VSPI PİNLERİ

Yüksek hız gerektiren uygulamalarda kullanmak için SPI protokolünü destekleyen pinlerdir.

* TXD & RXD PİNLERİ

İki cihaz arasında seri iletişim için kullanılır. TXD veri gönderen, RXD ise veri alan pindir.

* DAC & ADC PİNLERİ

Analog ve dijital sinyaller arasında dönüşüm yapan pinlerdir, analog çıkış veren 2 pin bulunurken, çoğu pin dijital çıkış vermektedir.

* DİĞER DETAYLAR

Aynı zamanda üzerinde bulundurduğu her pin bir pwm sinyali üretebilmektedir, bu sinyaller analog sinyal çıkışı sağlamak için kullanılır. Bir motorun hızını, açısını veya bir ledin parlaklığını ayarlamak mümkündür. Üzerinde 5V besleme için uygun olan bir Vin pini bulundurur, aynı zamanda 3.3V içinde pin mevcuttur. En sol üstte kalan EN pini ise aktif hale getirmek veyahut sıfırlamak için kullanılan pindir aynı zamanda microusb kablo girişinin solunda kalan buton da aynı görevi görmektedir. Bunların dışında 2 adet de GND pini bulunmaktadır.

### MG905 Mikro Servo Motor

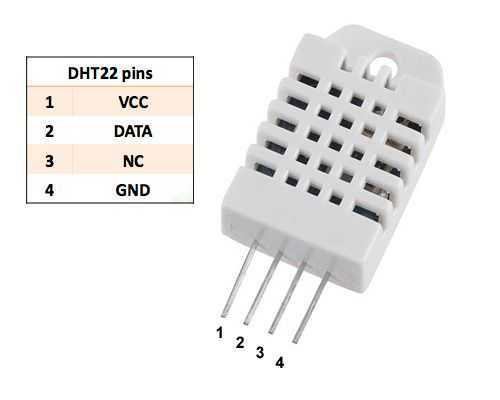
Küçük çaplı sistemlerde nesnelere hareket fonksiyonu kazandırmak için kullanılabilecek kendisine oldukça ufak olan bir motordur. 180 derece çalışma açısına sahip olduğu bilgisine sahibiz ancak yine de tam olarak bu dereceyi yakalayamıyor. 0’dan 180dereceye ulaşması sadece 0.3sn sürüyor. Uygulayabileceği tork kuvveti 1.8 kgf.cm , çalışma voltajı ise 5V.



Bağlantısı için kullanılan kablolar ise şu şekildedir; kırmızı kablolar artı (+), kahverengi ise eksi (-), turuncu ise bizim üzerinde kontrole sahip olmamızı sağlayan PWM sinyali için kullanılmakta. Bir PWM döngüsü 20ms sürmektedir, yani 50Hz bir frekansla çalışmakta.

### DHT22 Sensör

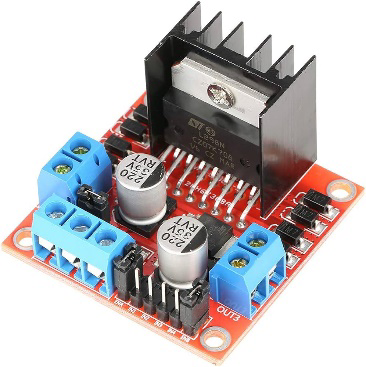
DHT22 piyasa bulabileceğimiz en temel, düşük maliyetli sıcaklık ve nem sensörüdür. Dijital bir sinyal oluşturmaktadır, yaklaşık 2sn’lik döngülerle veri oluşturur. İçerisinde bir mikrodenetleyici(8bit) bulunurdur. -40/80 değerleri arası sıcaklık, 0/100% değerleri arası nem bilgisini algılayabilmekte. 5V ile çalışmaya uygundur. Pin şeması şekildeki gibidir.



### Diğer Materyaller

Hazırlayacağım maket akıllı ev sisteminde kullanacağım diğer materyaller şu şekilde;

Bir adet DC motor, 3-6V arasında gerilimlerle çalışabilen evin havalandırma sistemi gibi değerlendirilecek, motora bağlı pervane ile ortamda hava sirkülasyonu oluşturulması amaçlandı.



Şekil 6 Diğer Elemanlar

Yine 5V çalışma geriliminde kullanılabilecek , sıradan 2 farklı led tercih edildi. Maket evin 2 farklı konumunda bulunacak bu ledlerin aynı anda veya ayrı ayrı yanabilmelerini gözlemek amaçlandı.

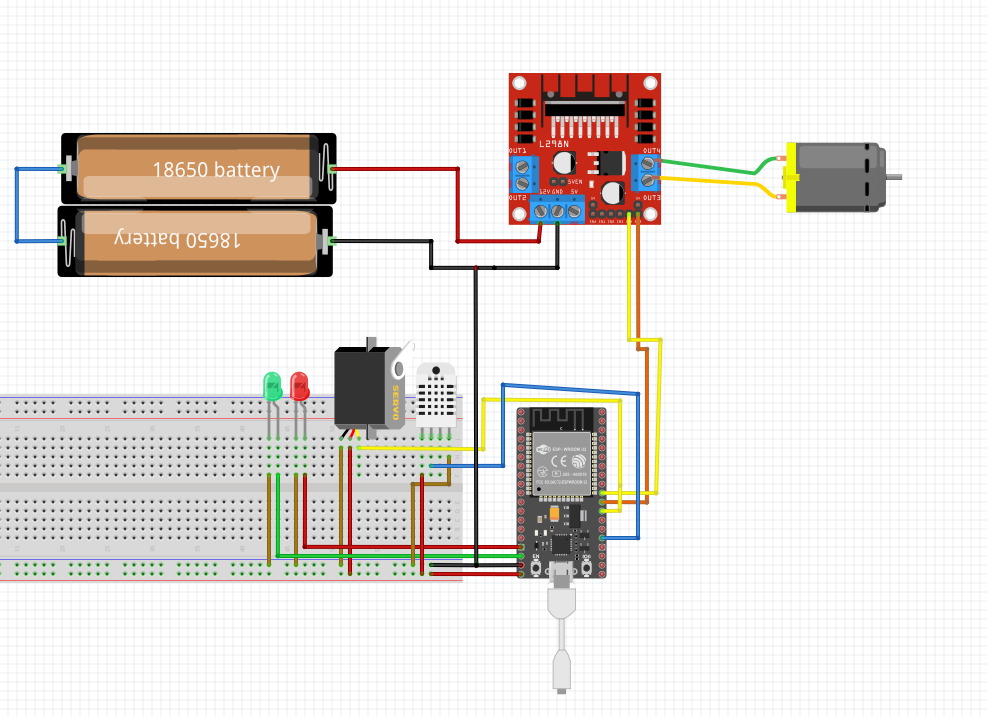
Sistemin beslemesini sağlamak içinse 5V micro-usb bağlantısı kullanılacak ve sistemdeki çoğu nesne 5v ile çalışmaya uygun olduğundan ötürü bir de voltaj regülatörü kullanmak gerekmeyecektir. Sistemin bütün beslemesini micro-usb ile yapmak mümkün olmayacaktır, gerekli olan akımı sağlamakta yetersiz kalacaktır. Bu sebepten ötürü dc motor kontrolü ve beslemesi için ise 3.7V gerilime ve 4800 Mah lityum-ion piller kullanılması kararlaştırılmıştır.

## METOT

Bu bölümde Nesnelerin İnterneti Temelli Akıllı Ev Sistemi projesi uygulama metodundan bahsedilmiştir. Sırasıyla şu şekilde ilerlemektedir; cihazların verimli şekilde çalışabilmesi gerekli bağlantılarının yapılması, sistemin güç kaynağı tercihi ve bağlantısının yapılması, mikrokontrolörün efektif şekilde çalışıp gerekli veriyi alıp, iletebilecek şekilde yazılımının geliştirilmesi ve sistemin erişilebilirliği, kullanışı.

### Sistemin Bağlantı Şeması

Şekilde sistemin bağlantı şeması detaylıca aktarılmıştır, ESP-32 üzerinde seçilen bağlantı şekillerini inceleyebilirsiniz.



Esp32 üzerindeki Vin pini üzerinden ledler, servo motor, ve dht22 sıcaklık ve nem sensörünün beslemeleri gerçekleştirilmiştir. ledlerin kontrolü GPIO 12 ve 13 pinleri, servo motor GPIO 5 pini, dc motor ise GPIO 18,19 ve 21 pinleri kullanılarak kontrol edilmiştir. Motorlar bu pinler üzerinden gelecek olan pwm sinyalleri doğrultusunda açıları, hızları ve yönleri kontrol edilebilecektir. Dht22 GPIO 4 pini üzerinden topladığı sıcaklık ve nem verilerini mikro kontrolör ile paylaşacak, işlenen veri ardından kullanıcının hizmetine sunulacak.

### Sistemin Besleme ve Şeması

Sistemin beslemesi micro-usb kablo bağlantısı ile sağlanmıştır ve sistemin neredeyse tüm bileşenleri 5v ile çalışmaya uygun olduğu düşünülürse bu tercih tutarlı olacaktır. Aynı zamanda dc motor kontrolü de amaçlandığı senaryoda yeterli akım değerlerine kullanılan besleme metodu ile sağlanamayacağı için bunu Li-on piller yardımıyla yapmak daha verimli olacaktır. Aynı zamanda bu pillerin bağlantılarını motor sürücü kartı ile sağlayarak, ESP32 mikro kontrolcüsünden gelecek olan pwm sinyalleri doğrultusunda ise dc motorun hızı ve yönü kontrol edilebilecektir. Pwm sinyallerinin başarılı şekilde iletilmesi için motor sürücü kartının toprak hattı ile esp32 toprak hattı bağlantısı gerçekleştirilmiştir.

### ESP-32 Yazılımı ve Kod Derleyici

ESP-32 ve diğer sık kullanılan Arduino, Raspberry Pi gibi mikro kontrolörlerin efektif şekilde kullanabilmesi için en sık tercih edilen dil, C++ yazılım geliştirme dilidir. Derleyici olarak da birçok program kullanılabilmektedir, ancak kütüphane ve kart kurulumlarının erişimin rahatlığı sebebiyle Arduino Ide derleyicisi kullanılması tercih edilmiştir. Gömülü yazılım sektöründeki gelişmeler sayesinde, küçük ölçekteki tasarımlarda kod geliştirmek daha kolay bir hal alıyor. Bunun başlıca sebebi kod derleyici uygulamaların veya belirli kartlar için üretilmiş yazılımların ve kütüphanelerin açık kaynak olarak paylaşılıyor oluşu. Geliştirici kartına eklemek istenen fonksiyonlar için birçok güncel geçerli kütüphane ve kod dizisi bulunabilmesi mümkün. Arduino Ide derleyicisi kendi bünyesinde bulundurduğu kütüphane erişimleri ile birlikte bu imkanı sağlamaktadır. Yine de bir çok farklı fonksiyonu bir araya getirecek sistemler için öğrenilmesi gereken temel c++ yazılım dili becerileri bulunmaktadır.

Nesnelerin İnterneti Temelli Akıllı Ev Sistemi projesinde kullandığımız kütüphaneler ve fonksiyonları şu şekilde;

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Blynksimpleesp32 kütüphanesi, esp32’nin blynk sistemi ile kablosuz ağ noktasını kullanarak iletişime geçebilmesi için gerekli olan fonksiyonların bulunduğu kütüphanedir. Wifi kütüphanesi ise çevredeki kablosuz ağ noktalarını tarayabilmeyi ve bağlanabilmeyi sağlayan kod dizilerini içermektedir. Web Server ve Wifi Client kütüphaneleri yardımıyla bağlanılan ağ üzerinde belirlenen portlar üzerinde server oluşturulabilir veya kendi kablosuz bağlantı noktası yaratılabilir. Adafruit Sensor ve DHT kütüphaneleri ise sıcaklık, nem sensörünün verilerinin derlenebilmesi ve kullanıma hazır hale gelebilmesi için gerekli kütüphanelerdir. Servo kütüphanesi ise servo motorun çalışma açılarını tanımlamak için gerek duyulan kütüphanedir.

Bu kütüphaneleri kullanarak yazılım geliştirmek oldukça hızlı ilerleyen bir süreç, ancak eklenen her kütüphanenin birbirleriyle sıkıntısız çalışması her zaman mümkün değildir. Bu doğrultuda gerekli araştırmalar yapıldıktan sonra uygun kütüphane tercihleri gerçekleştirilmiştir. Hata çıkardığı tespit edilen kütüphane dosyaları üzerinde değişiklikler yapılmıştır.

### Sistemin Erişilebilirliği ve Kullanışı Üzerine

Çalışmanın başında da bahsedildiği gibi sistem üzerindeki erişimi sürekli olarak sağlayabilmek amaçlanmaktadır. Erişimin sürekli sağlanabilmesi için üç farklı yöntem senkronize şekilde çalışabilecek şekilde oluşturulmuştur. İletişim ilk aşamada ücretsiz şekilde hizmet sağlayan bulut uygulamalarından biri olan Blynk sistemi üzerinden gerçekleştirildi.

elektronik donanım, elektronik cihaz, küçük alet, mobil telefon içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, daire, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Esp32’nin wifi modülü sayesinde kullanılabilir bir kablosuz ağ noktası ve internet bağlantısı yardımıyla blynk sisteminde tanımlanan cihaz adını kullanarak serverlarına bağlantı sağlandı ve bulut sistemi üzerinden veri alışverişi sağlandı. Kullanışlı arayüzü ve diğer kütüphanelerle etkin şekilde çalışabilecek olması sebebiyle tercih edildi. Bu şekilde sistemin kontrolü internet bağlantısına sahip olan her cihazdan rahatça kontrol edilebilir hale geldi. Daha önce bahsedilen bazı uygulamalarda da kullanıldığı gözlemlenebilir.

Blynk sunucularına bağlanmak için gerekli olan kod dizileri:metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu aşamada tespit edilen en büyük sıkıntı, bir internet kesintisi anında blynk sistemi geri dönülemez şekilde, fiziksel bir müdahale olmadan tekrar bağlantı kurmayı başaramamakta. Tekrar bağlantı kurmaya çalışması için gerekli yazılım dizilerini ekleyebiliriz ancak bu da internet bağlantısı sağlanamadığı sürece hiçbir işe yaramayacaktır. Önerilen çözüm internet bağlantısı olmadığı durumlarda dahi kontrol edebilecek şekilde sistemin yapılandırılmasıdır. İlk aşamada ön görülen uygulama evimizde bulunan yerel ağ üzerinde bir istasyonda yönetici paneli oluşturarak kontrole devam edebilmektedir, bu süreçte sistem hem hizmet vermeye hem de tekrardan blynk sunucularına bağlanmayı deneyecektir.

Blynk sisteminin çalışıp çalışmama durumlarına göre etki eden kod dizisi ve istasyonda sunucu oluşturan kod dizisi aşağıda verilmiştir.

metin, yazı tipi, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, yazı tipi, ekran görüntüsü içeren bir resim

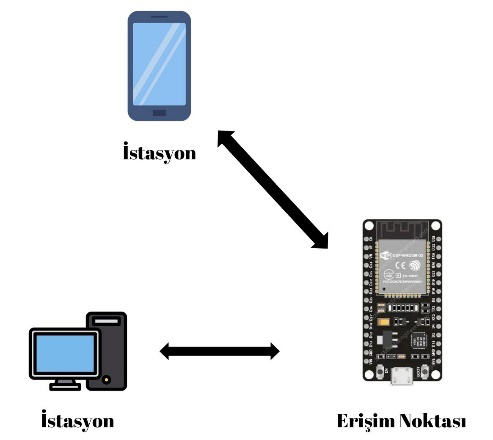
Açıklama otomatik olarak oluşturuldu

Blynk işleyişini engellenmemesi adına blynk bağlantı statüsü yerine istasyon üzerindeki sunucunun durumu kontrol edilmiş, sunucu çalışmadığı sürece blynk sisteminin sıkıntısız şekilde çalışması için gerekli kodlar koşturulmuştur. Ardından bir bağlantı kesintisi tespiti için gerekli kod koşturulmuş duruma göre sunucu oluşturulmuştur.

metin, ekran görüntüsü, ofis malzemesi, alet içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Aynı yerel ağa bağlı olunduğu sürece istenen cihazdan, farklı bir arayüz tasarımı aracılığıyla yine tüm nesneler üzerinde kontrol sağlanabilecektir. Bu aşamada erişilebilirlik sağlanmış olmasına rağmen hala sıkıntılar ile karşı karşıya kalınabilir. Örneğin, erişim noktasına erişemezse cihazlar üzerinde herhangi bir kontrole sahip olunamaz. Erişim noktasının işlevini yitirdiği noktada yeni bir erişim noktasına ihtiyaç duyulacaktır. Bu durumda sistem kendisini bir erişim noktası gibi tanımlayacak ve cihazlar üzerinden herhangi farklı bir ağa bağlanmaksızın direkt olarak sistemin oluşturduğu erişim noktasına bağlantı sağlanabilecek.metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Esp32 wifi modülü ve Wifi kütüphanesi yardımıyla oluşturulan erişim noktasında, kablosuz ağ bağlantısı kurabilecek her cihaz yine sistem üzerinde kontrole sahip olabilecektir. Bu bağlantı şeklinde de kullanıcı için aynı arayüz aktif olacaktır. Önceki erişim noktasının durumunun kontrolü ve esp32’nin erişim noktası olarak yapılandırılması için gerekli kod dizisi aşağıdaki gibidir.

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

İlk erişim noktasında oluşturulan sunucu ve daha sonra esp32’nin erişim noktasında yaratılan sunucu tamamen aynı fonksiyonlara sahiptir, tamamen yeniden bir sistem tanımlanmamış sadece sistemin erişilebilir olduğu konum değiştirilmiştir.

# DEĞERLENDİRME VE SONUÇ

Bu bölümde Nesnelerin İnterneti Temelli Akıllı Ev Sistemi projesi dahilinde hem nesnelerin interneti hem de akıllı ev teknolojileri hakkında edinilen bilgiler doğrultusunda, değerlendirmeleri içermektedir.

Hazırlanan çalışmada akıllı ev sistemleri üzerinde tespit edilen problemler giderilmeye çalışıldı. Sistem üzerinde yapılan her tercih erişilebilirlik ve maaliyet konuları dikkate alınarak gerçekleştirildi. Temel sorun olarak belirlenen erişilebilirliğin çözümü için senkron şekilde çalışabilecek 3 farklı erişim metodu dizayn edildi. Yapılan testler doğrultusunda bu metotlar arası geçişlerin rahatlıkla sağlanabilmesi için kod dizileri oluşturuldu.

# ÖNERMELER

Bu kısımda hazırlanan Nesnelerin İnterneti Temelli Akıllı Ev Sistemi başta olmak üzere bir çok Nesnelerin İnterneti çalışmasında mikrokontrolör kullanımı ve iletişim metotları üzerinden geleceğe önerilere yer verilmiştir. Hazırlanan çalışmada durum şartlarına adapte olan ve iletişimini bu duruma göre dizayn eden bir mikrodenetleyici kullanımı öne sürdür, bunu yaparken veri tabanı ve sunucu hizmetlerini dış kaynaklı şekilde sağlandı. Tüm bu hizmetleri çalışma bünyesinde de inşa etmek mümkündür. Mikrokontrolörün erişebileceği bir sunucu ve veri tabanı sayesinde bu hizmetlerde de bağımsızlık sağlanabilir. Bu gelişme ardından birden fazla mikrokontrolör ve bağlantılı nesneleri bu sunucu üzerinden kontrol edebilmek mümkün olacaktır. Bir diğer öngörülebilir yenilikse, akıllı ev sistemi projelerinin şuan olduğundan çok daha farklı şekillerde erişilebilir olacağını düşünülürse, bu sistemlerin aynı zamanda deprem gibi doğal afet anlarında arama kurtarma çalışmalarına destek verebilecek şekilde düzenlenebilir. Gerekli mekanik ve dayanıklılık çalışmalarıyla birlikte deprem, yangın anlarında kötü şartlarda kalacak kullanıcının arama kurtarma ekipleriyle iletişim kurabilmesine imkan sağlayabilir. Bu işlemi gerek manuel gerek sensörlerden alacağı verilerle otomatik olarak gerçekleştirebilir, manuel kullanabilme ihtimali dahilinde arama kurtarma ekipleriyle bizzat yazılı, sesli ve görüntülü iletişim kurulabilir. Bu sayede doğal afetlerin yol açacağı bir çok ölüm önlenebilir.

## KAYNAKLAR

[1] Rose, K., Eldridge, S., & Chapin, L. (2015). The internet of things: An overview. *The internet society (ISOC)*, *80*(15), 1.

[2] Rose, K., Eldridge, S., & Chapin, L. (2015). The internet of things: An overview. *The internet society (ISOC)*, *80*(15), 4.

[3] Lobaccaro, G., Carlucci, S., & Löfström, E. (2016). A review of systems and technologies for smart homes and smart grids. *Energies*, *9*(5), 17.

[4] Yüzgeç, Uğur, and Ömer Aba. "Raspberry pi kullanılarak bir akıllı ev uygulaması geliştirilmesi." *Bilecik Şeyh Edebali Üniversitesi Fen Bilimleri Dergisi* 4.1 (2017): 21-29.

[5] Y. Huang and G. Li, “A Semantic Analysis for Internet of Things,” in International Conference on Intelligent Computation Technology and Automation (ICICTA), May 2010.

[6] Gokhale, P., Bhat, O., & Bhat, S. (2018). Introduction to IOT. *International Advanced Research Journal in Science, Engineering and Technology*, *5*(1), 41-44.

[7] "The "Only" Coke Machine on the Internet". Carnegie Mellon University. Retrieved 10 November 2014.

[8] Suresh, P., Daniel, J. V., Parthasarathy, V., & Aswathy, R. H. (2014, November). A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment. In *2014 International conference on science engineering and management research (ICSEMR)* (pp. 1-8). IEEE.

[9] Ashton, K. (2009). That ‘internet of things’ thing. *RFID journal*, *22*(7), 97-114.

[10] Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012, December). Future internet: the internet of things architecture, possible applications and key challenges. In *2012 10th international conference on frontiers of information technology* (pp. 257-260). IEEE.

[11] Wu, M., Lu, T. J., Ling, F. Y., Sun, J., & Du, H. Y. (2010, August). Research on the architecture of Internet of Things. In *2010 3rd international conference on advanced computer theory and engineering (ICACTE)* (Vol. 5, pp. V5-484). IEEE.

[12] Taştan, M., & Gökozan, H. (2018). An Internet of Things based air conditioning and lighting control system for smart home. *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, *50*(1), 181-189.

[13] Asadullah, M., & Ullah, K. (2017, April). Smart home automation system using Bluetooth technology. In *2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT)* (pp. 1-6). IEEE.

[14] YILDIRIM, C., & BÖCEKÇİ, V. (2023). Düşük Maliyetli GSM Tabanlı Ev Güvenlik Sistemi. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, *11*(4), 1946-1957.

[15] Aziz Ertan,(2019). Zigbee Teknolojisinin Araştırılması ve Biyomedikal İşaretlerin İletimine İlişkin Örnek Bir Uygulama, Dicle Üniversitesi Fen Bilimleri Enstitüsü, (1-138).

[16] Tarek H. Ali Adeb, (2021). Zıgbee ve Kullanarak Ev Otomasyon Sisteminin Geliştirilmesi, Kastamonu Üniversitesi Fen Bilimleri Enstitüsü, (1-132).

[17] . Jia, X., Feng, Q., Fan, T., Lei, Q. (2012, April). RFID technology and its applications in Internet of Things (IoT). In 2012 2nd international conference on consumer electronics, communications and networks (CECNet) (pp. 1282-1285). IEEE.

## EKLER

#include <WiFi.h>

#include <WiFiClient.h>

#define BLYNK\_TEMPLATE\_ID "TMPL6gzwWKyCw"

#define BLYNK\_TEMPLATE\_NAME "esp32"

#define BLYNK\_PRINT Serial

#include <BlynkSimpleEsp32.h>

#include <WiFi.h>

#include <WebServer.h>

#include <Adafruit\_Sensor.h>

#include <DHT.h>

#include <ESP32Servo.h>

#define DHTPIN 4         // DHT22 sensörünün bağlı olduğu pin

#define DHTTYPE DHT22    // DHT tipi (DHT11, DHT22, DHT21)

int ENB = 18; // PWM sinyali için

int IN4 = 19; // Yön kontrolü için

#define ENB 18 // PWM sinyali için

#define IN4 19 // Yön kontrolü için

WebServer server(80);

DHT dht(DHTPIN, DHTTYPE);

Servo myservo;

char auth[] = "QL0C5XHx3OQrk9anq\_m3-UNwFQZXu6ci";

const char\* ssid = "proje";     //FiberHGW\_ZT63JY       proje

const char\* password = "12345679";    //YXPhsu4TkfTH          12345679

const char\* ap\_ssid = "ESP32"; // AP adı

const char\* ap\_password = "12345679"; // AP şifresi

int GREEN\_LED\_PIN = 12;

int RED\_LED\_PIN = 13;

#define LED\_PIN\_1 13 // ESP32'deki GPIO pin numarası

#define LED\_PIN\_2 12 // ESP32'deki GPIO pin numarası

#define SERVO\_PIN 5 // ESP32'deki GPIO pin numarası

String ledStateGreen = "OFF";

String ledStateRed = "OFF";

int servoAngle = 0; // Servo motorun açısı

bool apStarted = false; // AP modunun başlatılıp başlatılmadığını takip etmek için bayrak

bool serverStarted = false;

void setup() {

  pinMode(ENB, OUTPUT);

  pinMode(IN4, OUTPUT);

  pinMode(GREEN\_LED\_PIN, OUTPUT);

  pinMode(RED\_LED\_PIN, OUTPUT);

  myservo.attach(5); // Servo motorun bağlı olduğu pin

  dht.begin();

  Serial.begin(9600);

  delay(2000);

  connectToWifi();

  if (WiFi.status() == WL\_CONNECTED){

    Blynk.begin(auth, ssid, password);

  }

}

void loop() {

  if(!serverStarted) {    // server çalışmadıysa

    Blynk.run();

    sendSensor();

    if (!Blynk.connected()) {

    beginServer();

    serverStarted = true;

  }

  } else {

       server.handleClient();

  }

  if (!apStarted) { // AP modu başlatılmadıysa, normal çalışmaya devam et

    server.handleClient();

    if (WiFi.status() != WL\_CONNECTED) {

      // WiFi bağlantısı yoksa AP modunu başlat

      startAPMode();

    }

  } else {

    server.handleClient();

  }

}

void connectToWifi() {

  int attempts = 0;

  while (attempts < 10) {

    WiFi.begin(ssid, password);

    Serial.println("WiFi bağlantısı kuruluyor...");

    unsigned long startTime = millis();

    while (millis() - startTime < 3000) { // Her bir deneme arasında 3 saniye bekleyin

      if (WiFi.status() == WL\_CONNECTED) {

        Serial.println("");

        Serial.println("WiFi bağlandı");

        Serial.println("IP adresi: ");

        Serial.println(WiFi.localIP());

        return;

      }

      delay(100);

    }

    attempts++;

    if (attempts >= 10) {

      Serial.println("\nFailed to connect to WiFi after 5 attempts. Starting AP mode.");

      startAPMode();

      return;

    }

  }

}

void sendSensor() {

  float h = dht.readHumidity();

  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {

    Serial.println("Failed to read from DHT sensor!");

    return;

  }

  // Blynk uygulamasındaki sanal pinlere nem ve sıcaklık değerlerini yazdırın

  Blynk.virtualWrite(V4, h); // V4 pinine nem değerini yazdırın (Humidity)

  Blynk.virtualWrite(V3, t); // V3 pinine sıcaklık değerini yazdırın (Temperature)

}

BLYNK\_WRITE(V0) {

  int led1State = param.asInt();

  digitalWrite(LED\_PIN\_1, led1State);

}

BLYNK\_WRITE(V1) {

  int led2State = param.asInt();

  digitalWrite(LED\_PIN\_2, led2State);

}

BLYNK\_WRITE(V5) {

  int sliderValue = param.asInt(); // Slider'dan gelen değeri oku (0-255)

  analogWrite(ENB, sliderValue); // Motor hızını belirleyen PWM sinyalini gönderir

  if (sliderValue == 0) {

    digitalWrite(IN4, LOW); // Motoru durdur

  } else {

    digitalWrite(IN4, HIGH); // Motoru çalıştır

  }

}

BLYNK\_WRITE(V2) {

  int servoState = param.asInt();

  if (servoState == 1) {

    myservo.write(85); // Servo 90 dereceye dönüyor

  } else {

    myservo.write(5);  // Servo 0 dereceye dönüyor

  }

}

void beginServer() {

  server.on("/", handleRoot);

  server.on("/setservo", HTTP\_POST, handleSetServo); // Servo açısı ayarlama endpoint'i

  server.on("/controlled", handleControlled); // LED kontrolü endpoint'i

  server.on("/motorcontrol", handleMotorControl);

  server.begin();

  Serial.println("HTTP server started");

}

void handleRoot() {

  String page = "<!DOCTYPE html><html lang=en-EN><head><meta charset='UTF-8'><meta name='viewport' content='width=device-width, initial-scale=1.0'>";

  page += "<title>ESP32 WebServer - İlyas Arslan</title>";

  page += "<style> body { background-color: #f2f2f2; font-family: Arial, Helvetica, sans-serif; text-align: center; }</style>";

  page += "<style> .container { display: flex; flex-direction: column; align-items: center; justify-content: center; height: 100vh; }</style>";

  page += "<style> button { margin: 10px; padding: 10px 20px; border: none; border-radius: 5px; cursor: pointer; }</style>";

  page += "</head><body><div class='container'>";

  page += "<h1>ESP32 WebServer - İlyas ARSLAN</h1>";

  page += "<h3>Yeşil LED: " + ledStateGreen + "</h3>";

  page += "<form action='/controlled' method='POST'>";

  page += "<button name='green' value='1' style='background-color: #4CAF50; color: white;' >AÇ</button>";

  page += "<button name='green' value='0' style='background-color: #f44336; color: white;' >KAPA</button>";

  page += "</form>";

  page += "<h3>Kırmızı LED: " + ledStateRed + "</h3>";

  page += "<form action='/controlled' method='POST'>";

  page += "<button name='red' value='1' style='background-color: #4CAF50; color: white;' >AÇ</button>";

  page += "<button name='red' value='0' style='background-color: #f44336; color: white;' >KAPA</button>";

  page += "</form>";

  page += "<h3>Sıcaklık: " + String(dht.readTemperature()) + " °C</h3>";

  page += "<h3>Nem: " + String(dht.readHumidity()) + " %</h3>";

  page += "<h3>Servo Motor Açısı: " + String(servoAngle) + " °</h3>";

  page += "<form action='/setservo' method='POST'>";

  page += "<button name='servo' value='85'>AÇ</button>";

  page += "<button name='servo' value='5'>KAPA</button>";

  page += "</form>";

  page += "<h3>Motor Kontrolü:</h3>";

  page += "<form action='/motorcontrol' method='POST'>";

  page += "<button name='speed' value='0'>Dur</button>";

  page += "<button name='speed' value='1'>Düşük Hız</button>";

  page += "<button name='speed' value='2'>Orta Hız</button>";

  page += "<button name='speed' value='3'>Yüksek Hız</button>";

  page += "</form>";

  page += "</div></body></html>";

  server.send(200, "text/html", page);

}

void handleControlled() {

  if (server.hasArg("green")) {

    String greenValue = server.arg("green");

    if (greenValue == "1") {

      digitalWrite(GREEN\_LED\_PIN, HIGH);

      ledStateGreen = "ON";

    } else {

      digitalWrite(GREEN\_LED\_PIN, LOW);

      ledStateGreen = "OFF";

    }

  }

  if (server.hasArg("red")) {

    String redValue = server.arg("red");

    if (redValue == "1") {

      digitalWrite(RED\_LED\_PIN, HIGH);

      ledStateRed = "ON";

    } else {

      digitalWrite(RED\_LED\_PIN, LOW);

      ledStateRed = "OFF";

    }

  }

  server.sendHeader("Location", "/");

  server.send(303);

}

void handleMotorControl() {

  if (server.hasArg("speed")) {

    int motorSpeed = server.arg("speed").toInt();

    int pwmValue;

if (motorSpeed == 0) {

  pwmValue = 0; // Dur

} else if (motorSpeed == 1) {

  pwmValue = 84; // Düşük Hız

} else if (motorSpeed == 2) {

  pwmValue = 168; // Orta Hız

} else if (motorSpeed == 3) {

  pwmValue = 255; // Yüksek Hız

}

    analogWrite(ENB, pwmValue); // Motor hızını belirleyen PWM sinyalini gönderir

    if (motorSpeed == 0) {

      digitalWrite(IN4, LOW); // Motoru durdur

    } else {

      digitalWrite(IN4, HIGH); // Motoru çalıştır

    }

  }

  server.sendHeader("Location", "/");

  server.send(303);

}

void handleSetServo() {

  if (server.hasArg("servo")) {

    String servoValue = server.arg("servo");

    servoAngle = servoValue.toInt();

    myservo.write(servoAngle);

  }

  server.sendHeader("Location", "/");

  server.send(303);

}

void startAPMode() {

  WiFi.mode(WIFI\_AP);

  WiFi.softAP(ap\_ssid, ap\_password);

  Serial.println("AP mode started");

  Serial.print("AP IP address: ");

  Serial.println(WiFi.softAPIP());

  apStarted = true; // AP modu başlatıldı

  beginServer();

}

# ÖZGEÇMİŞ

**Kişisel Bilgiler**

Soyadı, adı : Arslan, İlyas

Uyruğu : T.C

Doğum tarihi ve yeri : 27.06.2000 Balıkesir

Telefon : +905432291216

e-mail : ilyasars53@gmail.com

**Eğitim**

| **Derece** | **Eğitim Birimi** | **Mezuniyet Tarihi** |
| --- | --- | --- |

| Lise | Şehit Cavit Köroğlu Anadolu Lisesi | 08/06/2018 |
| --- | --- | --- |

**Yabancı Dil :** İngilizce