



T.C.
DÜZCE ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ARDUINO İLE AKILLI EV SİSTEMİ

MAHMUT CAN KURT

LİSANS BİTİRME TEZİ

DANIŞMAN
DOÇ. DR. YUSUF ALTUN

DÜZCE, 2021

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

4 Haziran 2021

Mahmut Can Kurt

TEŞEKKÜR

Lisans öğrenimimde ve bu proje ve tezin hazırlanmasında gösterdiği her türlü destek ve yardımından dolayı çok değerli hocam Doç. Dr. Yusuf ALTUN'a en içten dileklerimle teşekkür ederim.

Çalışmalarım boyunca değerli katkılarını esirgemeyen eş danışmanım Dr. Öğr. Üyesi Sultan ZAVRAK'a da şükranlarımı sunarım.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen sevgili aileme ve arkadaşlarıma sonsuz teşekkürlerimi sunarım.

4 Haziran 2021

Mahmut Can KURT

İÇİNDEKİLER

ŞEKİL LİSTESİ	V
ÖZET	VI
SUMMARY	VII
1. GİRİŞ	1
1.1. PROJENİN TANIMI	1
1.2. PROJENİN AMACI	2
2. GENEL KISIMLAR	3
2.1. AKILLI EV SİSTEMLERİNİN TARİHÇESİ	3
2.2. AKILLI EV SİSTEMLERİNİN TEMEL ÖZELLİKLERİ	4
3. KULLANILAN ARAÇ VE YÖNTEMLER	5
3.1. ARDUINO UNO	5
3.1.1. Teknik Özellikler	5
3.1.2. Pin Fonksiyonları	6
3.2. HC-06 BLUETOOTH MODÜLÜ	8
3.2.1. Teknik Özellikler.....	8
3.2.2. Pin Fonksiyonları	9
3.3. LDR (IŞIK BAĞIMLI DİRENÇ) SENSÖRÜ	10
3.4. DHT11 SICAKLIK VE NEM SENSÖRÜ	11
3.4.1. Teknik Özellikler	11
3.5. DÖRT KANALLI RÖLE KONTROL KARTI	12
3.5.1. Teknik Özellikler	12
3.6. ARDUINO IDE	14
3.6.1. Kod Yazma	14
3.6.2. Üçüncü Taraf Yazılımlar	15
3.7. PROJE GELİŞTİRME AŞAMALARI	16
3.7.1. Devre Şeması	16

3.7.2. Blok Diyagramı	17
3.7.3. Programlama Aşaması	17
3.7.3.1. Arduino Uno Üzerindeki Geliştirme Aşamaları	17
3.7.3.2. Android Uygulama Geliştirme Aşamaları	20
4. TARTIŞMA, SONUÇ VE ÖNERİLER	22
KAYNAKLAR	23
EKLER	25
Ek – A	25
Ek – B	33
ÖZGEÇMİŞ	40

ŞEKİL LİSTESİ

Şekil 1 Arduino Uno	7
Şekil 2 HC-06 Bluetooth Modülü	9
Şekil 3 LDR Sensör	10
Şekil 4 DHT11 Sıcaklık ve Nem Sensörü	11
Şekil 5 Dört Kanallı Röle Kontrol Kartı	13
Şekil 6 Arduino IDE Genel Görünümü	15
Şekil 7 Arduino Uno ve HC-06 Bluetooth Modüllü Akıllı Ev Sisteminin Devre Şeması	16
Şekil 8 Arduino Uno ve HC-06 Bluetooth Modüllü Akıllı Ev Sisteminin Blok Diyagramı	17
Şekil 9 Akıllı Ev Sistemi Yönetim Paneli	21
Şekil 10 Bluetooth Adaptörlerinin Listelenmesi	21

ÖZET

ARDUINO İLE AKILLI EV SİSTEMİ

Akıllı ev sistemleri, son zamanlarda geliştirilmeye başlanan ve gelecekte oldukça yayılacağı düşünülen yeni bir tasarım geliştirmesidir. Ayrıca akıllı ev sistemleri, yalnızca cihazları açıp kapatmakla kalmaz, aynı zamanda iç ortamdaki anlık faaliyetleri izleyebilirler. Aynı zamanda akıllı ev teknolojisi, evdeki birçok özelliğin otomatikleştirildiği bir teknoloji olmakla beraber cihazların birbirleri arasında iletişim kurabildiği bir ortam oluşturur. Bunun yanında akıllı ev sistemleri enerji tasarrufu sağlayarak kullanıcıları yüklü bir masraftan korumaktadır.

Günümüzde ev içerisindeki lambalar, fanlar ve diğer elektrikli araçlar da akıllı ev otomasyonları sayesinde kullanılabilir. Bu tezde, bluetooth kullanılarak Arduino geliştirme kartı tabanlı bir akıllı ev sistemi projesi anlatılmıştır. Uygun maliyeti ve kolay kullanımı sayesinde kullanıcılar tarafından oldukça tercih edilebilecek bir projedir. Kullanıcının akıllı telefonu kullanılarak elektronik cihazların tümü yönetilebilmektedir.

Bu tezde, çeşitli proje fikirlerinde kullanılan HC-06 Arduino Bluetooth Modül ve Arduino Uno R3 geliştirme kartı kullanılmıştır. Ek olarak ışık sensörü olarak kullanılan LDR ve DHT11 sıcaklık ve nem sensörü entegre edilerek ortamdaki ışık, sıcaklık ve nem durumlarına göre tepkimelerde bulunulması sağlanmıştır.

SUMMARY

SMART HOME SYSTEM WITH ARDUINO

Smart home systems are a new design development that has been started to be developed recently and is expected to spread widely in the future. In addition, smart home systems can not only turn devices on and off, but also monitor instantaneous activities in the indoor environment. At the same time, smart home technology is a technology where many features in the home are automated and creates an environment where devices can communicate with each other. In addition, smart home systems save energy and protect users from a heavy expense.

Today, lamps, fans and other electrical tools in the home can also be used thanks to smart home automations. In this thesis, an Arduino development board based smart home system project using bluetooth is explained. It is a project that can be highly preferred by users thanks to its affordable cost and easy use. All electronic devices can be managed using the user's smartphone.

In this thesis, HC-06 Arduino Bluetooth Module and Arduino Uno R3 development board used in various project ideas. In addition, LDR and DHT11 temperature and humidity sensors, which are used as light sensors, are integrated to provide reactions according to light, temperature and humidity conditions in the environment.

1. GİRİŞ

Bu proje, insanların hayatını kolaylaştırmak ve enerji tasarrufu sağlamak için kullanılabilecek bir akıllı ev sistemi projesidir. Akıllı ev sistemleri entegre bir yapıdan oluşmaktadır ve evin çevresini sararak evin her noktasıyla etkileşime geçilebilmektedir. Akıllı ev sistemleri aydınlatma, güvenlik, ısı ve enerji konularında oldukça kullanışlı, hayat kurtarıcı ve tasarrufludur. Bunun yanında akıllı ev sistemleri Android ve IOS işletim sistemine sahip mobil cihazlardan veya bilgisayar üzerinden rahatlıkla kontrol edilebilirler.

Bu projede, akıllı ev sistemlerinin bluetooth ile yönetilen bir örneği üzerine çalışılmıştır. Ortam aydınlatması, LDR ışık sensörüyle ortamın ışık seviyesinin azalıp artma durumunun ölçülmesinin sonucuna göre geliştirilen cihaz aracılığıyla düzenlenebilmektedir. Bununla beraber, ortamdaki oda sayısına göre aydınlatıcılar artırılıp azaltılabilmektedir. Buna ek olarak geliştirilen cihaz, projede yer alan DHT11 sıcaklık ve nem sensörü aracılığı ile ortam sıcaklığı azaldığı veya arttığı durumda kullanılacak olan fan veya klima benzeri bir cihazı çalıştırarak ortam sıcaklığını da düzenleyebilmektedir. Böylece açık unutulan lamba ve klimalardan fazla enerji harcanması engellenmiş olacaktır.

Projede geliştirilen cihaz, üzerindeki sensörler ile beraber çalışabilmesinin yanında kullanıcıya ait Android veya IOS işletim sistemine sahip bir cihazda bulunan uygulama üzerinden de kontrol edilebilmektedir. Kullanıcı, dilediği zaman ortamda bulunan ışık seviyesini düşürebilir veya artırabilir, dilediği zaman ortam ısısı için çalışacak olan cihazı da devre dışı bırakıp çalıştırabilir. Bu sayede, Arduino ile Akıllı Ev Sistemi projesi kullanıcıya konfor kazandırmanın yanında kullanıldığı ortama da değer katmaktadır.

1.1. PROJENİN TANIMI

Proje, insanların hayatını kolaylaştıran ve basit, anlaşılır ve kafa karıştırıcı olmayan, kullanıcı ve çevre dostu bir devreye sahip bluetooth modülü içeren bir akıllı ev sistemidir. Basit bir kullanıma sahiptir. 13 yaş ve üzeri olan tüm yaş gruplarını hedeflemektedir. Özellikle 50 yaş ve üzeri yaşlı kesimin hayatını kolaylaştıran bir sistemdir. Bu sebeple tasarlanan mobil uygulama basit bir arayüze sahiptir ve her butonun işlevi rahatlıkla görülebilir. Söz konusu sistemin elektronik devresinde; 1 adet Arduino Uno, 1 adet HC-06 Arduino Bluetooth Modülü, 1 adet LDR ışık sensörü, 1 adet DHT11 Sıcaklık ve Nem Sensörü, 1 adet 4 kanallı röle bulunmaktadır. Devrede

bulunan LDR sensörü ile ışık şiddeti ölçülür ve bu şiddetin belirlenen düzeyden az olması durumunda röleye bağlı olan ampul yanarak ortamı aydınlatır. Eğer ışık şiddeti belirlenen düzeyden fazla ise bağlı rölede bulunan ampul söndürülerek enerji tasarrufu sağlanır. Aynı zamanda söz konusu ampul, mobil uygulamadan kullanıcının gönderdiği sinyal ile açılmış ise kullanıcı tekrar kapatana kadar yanmaya devam eder. Bunun yanında, DHT11 sensörü ile ortam sıcaklığı belirlenen seviyenin üzerindeyse röleye bağlı fan çalışır ve ortam sıcaklığını ideal seviyeye getirir. Ortam sıcaklığı ideal seviyenin altına düştüğünde ise fan çalışmayı durdurur ve bu sayede sıcaklığın ideal seviyede tutulması sağlanır.

Projeye ait mobil uygulamada, 1 adet lamba ve 1 adet fan butonu bulunmaktadır. Eğer kullanıcı mobil cihazıyla sistemde bulunan bluetooth modülüne bağlandıysa bu butonlar aracılığıyla belirlenen ortamdaki ampulü ve/veya fanı çalıştırabilmektedir.

Aynı zamanda sistemin taşınabilir olması sayesinde kullanıcı, ortamını değiştirdiğinde, evinden taşındığında ya da işyerini değiştirdiğinde yanında götürebilir ve sistemi yeni evinde veya işyerinde kullanmaya devam edebilir.

1.2. PROJENİN AMACI

Projenin birden fazla amacı bulunmaktadır. Bunlardan ilki ve en önemli olanı enerji tasarrufu sağlanarak hem israftan kaçınılmasını hem de maddi açıdan kullanıcının rahatlamasını sağlamaktır. Bu sayede kullanıcı bahsedilen projeyi kullanarak hem yaşamış olduğumuz dünyadaki enerji kaynaklarının daha az tükenmesine hem de maddi açıdan rahatlayıp diğer ihtiyaçlarına daha fazla para ayırabilmesine vesile olur.

Diğer bir amacı ise insanların hayatını kolaylaştırmak olan proje sayesinde insanlar vakit kaybetmekten kurtulmaktadır. Günümüzde aktif olarak, oldukça sık kullanılan akıllı mobil cihazlarımız sayesinde yerimizden dahi kalkmadan ışığı açıp kapatabilmek, havalandırma, ısıtma veya soğutma sistemini çalıştırabilmek gibi imkanlar sunulmaktadır. Bu imkanlar dahilinde kullanıcılar işlerinden veya meşgul oldukları şeylerden uzaklaşmadan, oturdukları yerden rahatlıkla bulundukları ortama etki edebilmektedirler.

2. GENEL KISIMLAR

2.1. AKILLI EV SİSTEMLERİNİN TARİHÇESİ

Akıllı ev sistemlerine ilk olarak 1950 yılında Ray Bradburry'nin "The Veldt" isimli kitabında anlatılan kısa hikâyede ev işlerinin ev tarafından kontrol edilmesi konu alınarak değinilmiştir [1]. İlk akıllı ev fikri olarak 1975 yılında genel amaçlı ev otomasyon sistemi olan X10 geliştirilmiştir. Geliştirilen X10 sistemi elektrik iletim kablolarını kullanan bir sistem olarak piyasaya sunulmuştur [2]. Bunun yanında Türkiye'de 1984 yılında ilk akıllı ev sistemi çalışmalarına başlanmıştır. İlk çıkan ürünler daha çok firmalar ve üst kesime hitap ediyordu ancak günümüzde bireysel kullanıma hitap eden teknolojinin gelişmesi sayesinde oldukça yaygınlaşmaya başlamıştır [3]. 2012 senesine kadar, Amerika Birleşik Devletleri'nde yapılan bir araştırmaya göre, 1,5 milyon eve otomasyon sistemi kurulmuştur [2].

1993 yılında hayata geçirilen akıllı ev sistemi çalışmasında ev otomasyon ürünleri sınıflandırılmıştır ve CEBus nokta mimarisi verilerek bunun detaylı açıklaması yapılmıştır. Ardından 1998 yılında ofis ve ev ortamlarında kullanılmak üzere telefonla uzaktan kontrol sistemi geliştirilmiştir. Geliştirilen bu sistem DTMF (Dual Tone Multi Frequency yani Çift Tonlu Çoklu Frekans) sistemine dayalıdır. 6 sene sonra 2004 yılında, DTMF alıcı analizi için Fourier Transform algoritması kullanılmıştır. Sonrasında, 2006 senesinde, evde bulunan aydınlatma ve perde sistemleri vücut hareketleriyle kontrolü üzerine çalışmalara başlanmıştır [4].

Günümüzde kullanımı oldukça yaygınlaşan bu ürünlerde faydası büyük olan Amazon firmasının 2010 yılında geliştirmeye başlayıp Kasım 2014'te çıkarmış olduğu Amazon Echo [5] isimli ürün Alexa sesli asistanı yardımıyla hava durumu, trafik durumu, müzik çalma gibi etkinlikleri gerçekleştirebiliyordu. Ayrıca Google firmasının ürettiği Google Home, yeni adıyla Google Nest, akıllı hoparlörü ise Mayıs 2016'da ilan edilmiş ve ilk olarak Kasım 2016'da ABD'de kullanılmaya başlanmıştır [6]. Google Nest, yine Google'ın geliştirmiş olduğu Google Asistant isimli akıllı kişisel asistanı aracılığıyla sesli komutları tanımlama ve o komutlara dönüt vermek için geliştirilmiştir. Geliştirilen bu akıllı hoparlör sayesinde ev içerisindeki diğer elektronik cihazlara dokunmadan müdahale edilebilmekte ve bunun yanında internet bağlantısı sayesinde video, müzik, hava durumu gibi tüm hizmetleri kullanıcıya sunabilmektedir. Bu geliştirmelerin yanında seyirci kalmayan Apple şirketinin çıkardığı Apple HomePod ise 5 Haziran 2017'de ilan edilmiş ve Aralık 2017'de kullanıcıyla buluşmuştur [7]. Apple HomePod, yine Apple tarafından geliştirilen Siri sesli

asistanını kullanmaktadır ve aynı Google Nest gibi ev içerisinde bulunan diğer cihazlara bağlanarak kullanıcının hayatını kolaylaştırmaktadır.

2.2. AKILLI EV SİSTEMLERİNİN TEMEL ÖZELLİKLERİ

Akıllı ev sistemleri genellikle aydınlatma, ısıtma, güvenlik ve ev içerisindeki elektronik cihazların yönetilmesini ve kontrolünü sağlayan otomasyon sistemleridir. Akıllı ev sistemleri akıllı mobil cihazlara bağlanarak ev içerisinde veya ortam dışında iken sistemin yönetilmesini ve kontrolünü sağlar. Bu sistemler, gelişen teknoloji ve yeni imkanlar sayesinde daha fazla gelişebilmektedir.

Enerji tasarrufu anlamında oldukça etken bir sistem olan akıllı ev otomasyonları; evin kullanılmayan bölgelerinin ısıtılmasını, kullanılmayan aydınlatma sistemlerinin çalışmasını ve gereğinden fazla enerji harcayan ısıtma ve soğutma sistemlerini engelleyerek görevini başarılı bir şekilde yerine getirir.

Bu sistemler sayesinde; ısı enerjisinde %5-25 tasarruf sağlanırken, gereksiz aydınlatmaların otomatik olarak söndürülmesi, çalışan aydınlatma sistemlerinin %90 parlaklıkta çalışması, çalışan cihazların ucuz tarife zamanlarına göre programlanması gibi yöntemler ile elektrik enerjisinin tüketiminde %30'a varan azalma sağlar [8]. Akıllı ev sistemlerinin en büyük kolaylığı senaryolaştırma imkanını sağlamasıdır. Örneğin; kullanıcı uyumadan önce tüm perdeleri kapatıyorsa belirli saatler arasında ışıklar kısılabilir, alarm otomatik olarak devreye girebilir, televizyon ve bilgisayar otomatik olarak kapatılabilir ve oda sıcaklığı istenilen duruma göre ayarlanabilir.

3. KULLANILAN ARAÇLAR VE YÖNTEMLER

3.1. ARDUINO UNO

Projenin yönetim merkezi olan Arduino Uno; Arduino.cc tarafından 2010 yılında geliştirilmiş olan açık kaynak, ATmega328P mikro işlemciye sahip bir mikrodenetleyici kartıdır. Arduino Uno, farklı kartlara ve devrelere bağlanabilmesi için dijital ve analog giriş ve çıkış (I/O) pinleriyle donatılmıştır. Kart üzerinde 14 dijital, 6 analog giriş ve çıkış pini bulunmaktadır. Kendisine ait USB kablosu ile bilgisayara bağlanarak ve yine Arduino.cc firmasının ürettiği Arduino IDE geliştirme ortamı kullanılarak programlanabilmektedir [9].

Arduino mikro denetleyicileri, programların işlemci üzerindeki flash belleğe yüklenmesini kolaylaştıran bir önyükleyici ile önceden programlanmıştır. Arduino Uno'nun varsayılan önyükleyicisi "optiboot" önyükleyicisidir. Bu kartlar, başka bilgisayarlara seri bağlantı ile program koduyla yüklenmektedir. Bazı Arduino kartları, RS232 lojik seviyeleri ile TTL seviye sinyalleri arasında dönüştürmek için bir seviye değiştirici içermektedir. Kullanılan Arduino kartları, FTDI FT232 gibi USB-to-Serial adaptörleri kullanılarak uygulanan USB aracılığıyla programlanmaktadır. Arduino Mini ve Boarduino gibi üretilen diğer kartlar, sonradan takılabilir bir USB-to-Serial adaptörleri, Bluetooth veya diğer yöntemleri kullanabilmektedir. Ayrıca, Arduino Uno üzerinde bulunan 14 dijital I/O pini 2.54 mm boyutundadır ve dişi başlıklar ile kart üzerinde yer edinmiştir [10].

3.1.1. Teknik Özellikler

- Mikrodenetleyici: ATmega328P
- Çalışma Gerilimi: 5 Volt
- Besleme Gerilimi: 7-20 Volt
- Dijital I/O Pin Sayısı: 14 (6 tanesi ile PWM çıkışı sağlanabilir)
- UART: 1
- I2C: 1
- SPPI: 1
- Analog Giriş Pinleri: 6

- I/O Pin Başına DC Akımı: 20 mA
- 3.3V Pin için DC Akımı: 50 mA
- Flash Bellek: 0,5 KB'si bootloader tarafından kullanılan 32KB
- SRAM: 2KB
- EEPROM: 1KB
- Saat Hızı: 16 MHz
- Uzunluk: 68,6 mm
- Genişlik: 53,4 mm
- Ağırlık: 25 gr

3.1.2. Pin Fonksiyonları

- LED: Kart üzerinde dijital pin 13 tarafından yönetilen dahili bir led bulunmaktadır. Pin değeri yüksek olduğunda led yanarken, değer düşük olduğunda söner.
- VIN: Bu pin üzerinden gerilim verilebilmektedir. Aynı zamanda güç girişi üzerinden gerilim veriliyorsa bu pin üzerinden erişilebilmektedir.
- 5V: Bu pin, kart üzerindeki regülatörden regüle edilmiş bir 5V çıkış vermektedir.
- 3V3: Yerleşik regülatör tarafından 3.3 Voltluk bir besleme pinidir.
- GND: Topraklama için kullanılan “ground” pinidir.
- Reset: Kartı yeniden başlatmak için kullanılmaktadır.



Şekil 1 Arduino Uno

3.2. HC-06 BLUETOOTH MODÜLÜ

HC-06 Bluetooth Modülü, kablosuz seri iletişim için tasarlanmış ve oldukça sık kullanılan bir bluetooth modülüdür. Bu modüle bağlanmak için özel herhangi bir programa gerek duyulmamaktadır. Akıllı telefondaki bir uygulama tarafından modüle girişler yönlendirilir ve ardından bunlar Arduino'ya aktarılır. Arduino gelen mesaja karşılık, üzerlerinde bulunan kodda belirtildiği şekilde dönüt vermektedir. HC-06 Bluetooth Modülü, Cambridge Silicon Radio BC417 2.4 GHz Bluetooth Radyo çipine dayanmaktadır. Bu çip, harici 8Mbit flash bellek kullanan karmaşık bir çiptir. 3 cm boyutunda olan bu küçük modül, 3.3V güç ile çalışmaktadır. Bu modülün iki çalışma modu bulunmaktadır. Bu modlar, AT komutları gönderebileceğiniz komut modu ve başka bir bluetooth modülüne veri gönderip aldığı bir veri modudur. Kullanımı kolaylaştıran çıkış panoları mevcuttur. Entegre olduğu kartların çoğu, 5V güçte çalışmaktadır ve bazı seviye kaydırma teknikleri ile 5V Arduino sinyal seviyelerine arayüz sağlamaktadır.

HC-06 Bluetooth Modülü bir “Slave” yani “Köle” modüldür. Bu bağımlı modül, başka bir Bluetooth cihazıyla bağlantı başlatamamaktadır, ancak bağlantıları kabul edip onlarla beraber çalışabilmektedir [11].

3.2.1. Teknik Özellikler

- Bluetooth Protokolü: Bluetooth v2.0
- Güç Seviyesi: Class2 (+6 dBm)
- Bant: 2.40 GHz – 2.48 GHz, ISM Bandı
- Alıcı Hassasiyeti: -85 dBm
- USB Protokolü: USB v1.1/2.0
- Modülasyon Modu: Gauss Frequency Shift Keying
- Güvenlik Sistemi: Kimlik doğrulama ve şifreleme
- Çalışma Gerilimi: 3.3 – 6 Volt
- Çalışma Sıcaklığı: -20 °C / +55 °C
- Çalışma Akımı: 40 mA

3.2.2. Pin Fonksiyonları

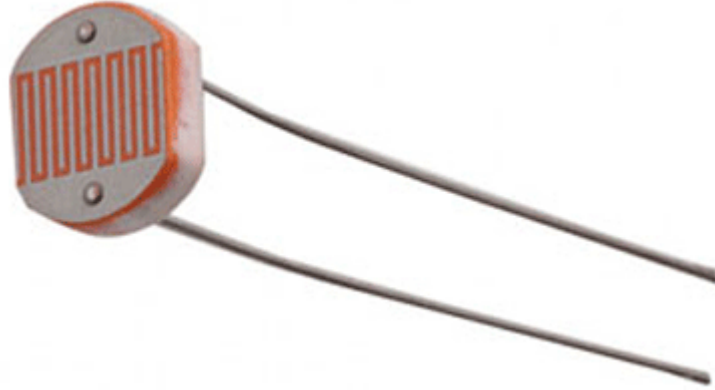
- Key: Pin durumu, modülün AT komut modunda veya normal modda çalışmasını belirler. Pin durumu yüksek ise AT komut modda, düşük ise normal modda çalışmaktadır.
- VCC: Modülün güç girişidir, +5V besleme ile çalışır.
- GND: Topraklama pinidir.
- TXD: Seri veriler, modül tarafından bu pin üzerinden iletilmektedir.
- RXD: Seri veriler, modül tarafından bu pin üzerinden alınmaktadır.
- State: Bu pin modülün durumunu görüntülemek için kart üzerindeki LED'e bağlanır.



Şekil 2 HC-06 Bluetooth Modülü

3.3. LDR (IŞIK BAĞIMLI DİRENÇ) SENSÖRÜ

LDR, özel bir direnç türüdür. Direnç olması sebebiyle kutupsuzdur ve bu sayede herhangi bir yönde bağlanabilmektedir. Breadboard üzerinde kolaylıkla kullanılabilmektedir. Bulunduğu devre üzerinde değişen direnç değerleri ile bir çıkış sağlamaktadır ama bu çıkışı dış ortamdan aldığı bir fiziksel değişim ile gerçekleştirmesinden ötürü bir sensör görevi görmüş olur. LDR'ler üzerlerine düşen ışık şiddeti ile ters orantılı çalışmaktadır. Işık şiddeti arttıkça direnç azalmakta, ışık şiddeti azaldığında da direnç değeri artmaktadır. LDR'nin en duyarlı olduğu ışık yeşil ışıktır. Küçük, ucuz ve çokça bulunabilen bir sensör çeşididir. PG5, PG5-MP, PG12, PG12-MP, PG20 ve PG20-MP serileri bulunmaktadır. Gündüz ve gece algılama projelerinde, sokak aydınlatmalarında, otomatik yatak odası ışıklarında çokça kullanılmaktadır [12].



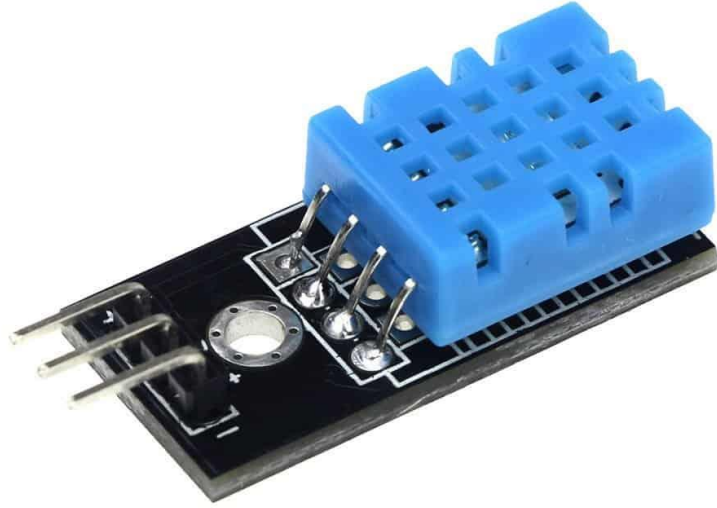
Şekil 3 LDR Sensör

3.4. DHT11 SICAKLIK VE NEM SENSÖRÜ

DHT11 Sıcaklık ve Nem Sensörü, öncesinde kalibrasyonu yapılmış bir dijital çıkış sağlamaktadır. Sensör, ortamda bulunan bağıl nemi ölçerken, sensör üzerindeki negatif sıcaklık katsayısı termistörü ile ortam sıcaklığı ölçülmektedir. Hata oranı oldukça düşüktür fakat 0 santigrat derecenin altında çalışmamaktadır. Kendine has dijital sinyal toplama tekniğini ve sıcaklık ve nem algılama teknolojisini kullanarak, yüksek güvenilirlik ve uzun vadede kararlılık sağlar [13].

3.4.1. Teknik Özellikler

- Çalışma Gerilimi: 3.3 – 5 Volt
- Nem Ölçüm Aralığı: %20 – 90 RH
- Sıcaklık Ölçüm Aralığı: 0 – 50 °C
- Nem Ölçümü Hata Payı: \pm %1 RH
- Sıcaklık Ölçümü Hata Payı: \pm 1 °C
- Sinyal Toplama Süresi: Ortalama 2 Saniye



Şekil 4 DHT11 Sıcaklık ve Nem Sensörü

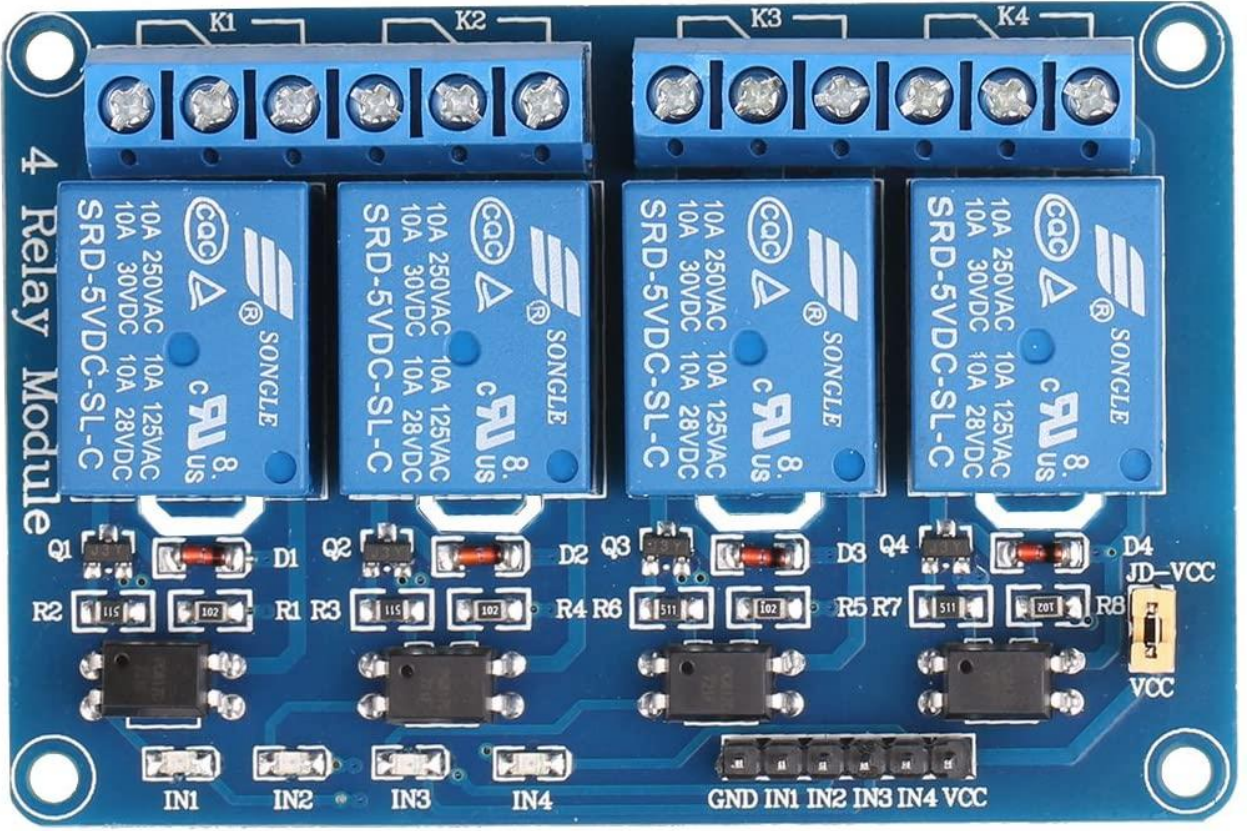
3.5. DÖRT KANALLI RÖLE KONTROL KARTI

Dört kanallı röle kontrol kartı, 5V güç ile bağlantıların kontrol edilebildiği ve Arduino veya benzer başka mikro denetleyicilerle beraber kullanılabilir. Mikro denetleyici üzerinden sinyal alındığında 20mA'lık bir akım ile çalışmaktadır. Endüstriyel ve robotik projelerinde oldukça sık kullanılmaktadır. 30V DC veya 220V AC gerilimde 10A'e kadar enerjiyle kullanılabilir. Üzerinde her röle için kontrol ledleri bulunmaktadır. Aynı zamanda röleler lojik 0 ile tetiklenmektedir. Her bir röle için NC, NO ve COM pinleri bulunmaktadır. Bu sayede tetiklenme durumunda kısa devre veya tetikleme durumunda açık devre olması istenilen durumlarda da kullanılabilir [14].

Röleler, bir anahtarı mekanik olarak çalıştırmak için elektromıknatısları kullanır. Ayrıca, bir devrenin düşük güç sinyaliyle veya birden fazla devrenin tek sinyal ile kontrol edilmesinin gerekli olduğu durumlarda kullanılır. Röleler ilk olarak, uzak mesafeli telgraf devrelerinde yükseltici olarak kullanılmıştır. Ardından telefon santrallerinde ve bilgisayarlarda lojik işlemlerin gerçekleştirilmesi için kullanılmaya başlandı.

3.5.1. Teknik Özellikler

- Röle sayısı: 4
- Çalışma Gerilimi: 5V / 12V / 24V DC
- Kontrol Sinyali: TTL Seviyesi
 - 5V Röle: 0 – 2V Açma, 3 – 5V Kapama
 - 12V Röle: 0 – 4V Açma, 5 – 12V Kapama
 - 24V Röle: 0 – 8V Açma, 9 – 24V Kapama
- Nominal Yük: 7A / 250VAC, 10A / 125VAC, 10A / 28VDC
- Bağlantı Gecikme Süresi: 5 – 10 ms
- Akım:
 - 5V Röle: Kanal başına 52 mA±
 - 12V Röle: Kanal başına 32 mA±
 - 24V Röle: Kanal Başına 20 mA±



Şekil 5 Dört Kanallı Röle Kontrol Kartı

3.6. ARDUINO IDE

Arduino IDE (Integrated Development Environment – Entegre Geliştirme Ortamı), Arduino kartlar için kodların yazılıp derlenmesi ve doğrudan Arduino kartlara yüklenmesi için kullanılan yazılım geliştirme platformudur [15]. Bu platform, cihazların tasarlanma sürecini kolaylaştırmak için geliştirilmiş ve oldukça popüler olan bir elektronik yazılım geliştirme platformudur. Akıllı ev sistemleri, robotik, giyilebilir bilgisayarlar gibi uygulama alanlarında oldukça sık kullanılmaktadır.

Arduino IDE, tek bir program üzerinden geniş bir yelpazede geliştirilme yapılmasına imkân tanır. Bir proje üzerinde bulunan birden fazla dosya takip edilerek programcılar tarafından proje yönetiminin karmaşık hale gelmesini de aynı zamanda engelleyebilir. Arduino IDE, kodun kendisinin derlenmesini, hata ayıklamasının yapılmasını, kodun direkt olarak Arduino kartlara iletilmesini ve çalıştırıldıktan sonra anlık veri ve işlem takibinin yapılmasını sağlar.

3.6.1. Kod Yazma

Arduino IDE, kendine has kurallar barındırarak C ve C++ dillerini destekleyebilmektedir. Kullanıcı tarafından yazılacak olan kod metin düzenleyici üzerinde yazılır ve .ino dosya uzantısı ile kullanılan işletim sistemi ortamına kaydedilir. Yazılan kodda bulunan herhangi bir hata veya sistem hakkında diğer mesaj bilgileri, programda alt kısımda bulunan konsol penceresinde görüntülenmektedir.

Şekil 6’da görüldüğü üzere Arduino IDE programının genel görünümü oldukça basit ve kullanışlıdır. Program üzerinde bulunan butonların anlamları ve amaçları aşağıdaki gibidir;



Verify – Doğrula: Kodu derleyerek kod üzerinde herhangi bir hata durumunu kontrol eder.



Upload – Yükle: Kodu derleyip bilgisayara bağlı olan geliştirme kartına yüklemeye yarar.



New – Yeni: Yeni ve temiz bir .ino dosyası açılır.



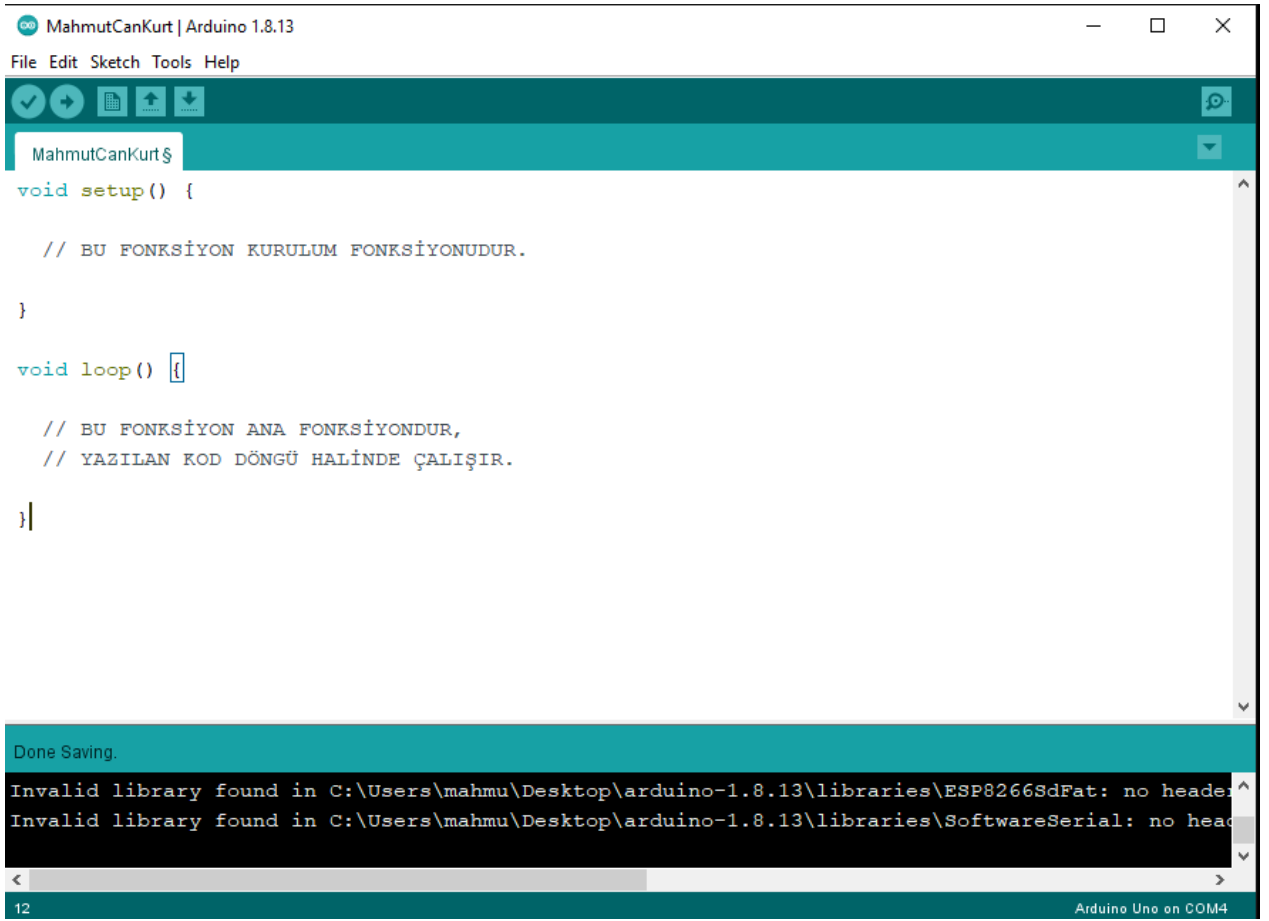
Open – Aç: Bilgisayardaki tüm dosyalar üzerinden .ino uzantılı bir dosyayı açmaya yarar.



Save – Kaydet: Yazılan kodu kaydetmeye yarar.



Serial Monitor – Seri Monitör: Yazılan koddaki girdi ve çıktıları anlık olarak gösterebilen bir ekran olan Seri Monitörü çalıştırır.



Şekil 6 Arduino IDE Genel Görünümü

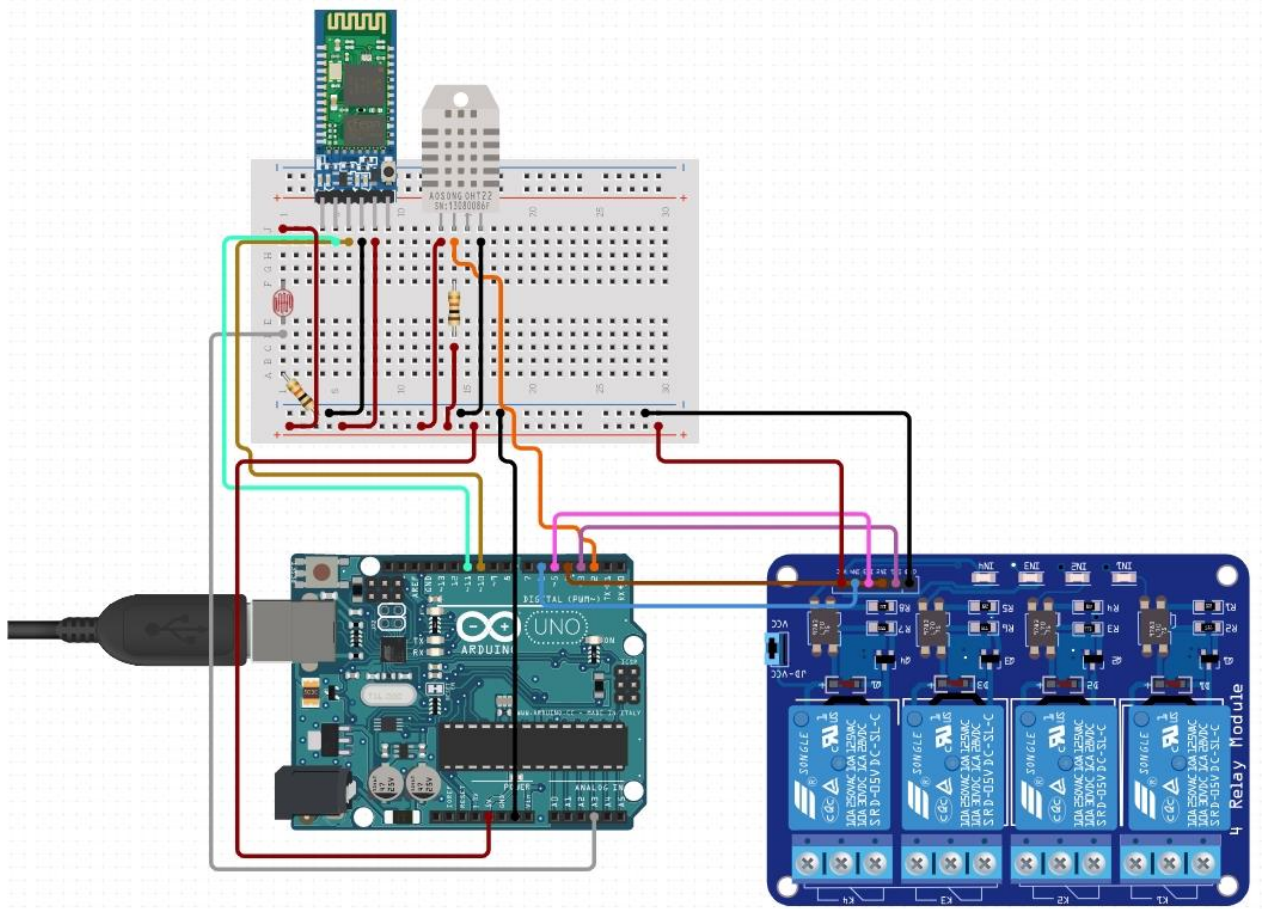
3.6.2. Üçüncü Taraf Yazılımlar

Üçüncü taraf yazılım kullanımına da imkân tanıyan Arduino IDE sayesinde yalnızca Arduino kartlarda değil, geliştirilebilen ek donanımlar üzerinde de geliştirilme yapılmasına yardımcı olur. Aynı zamanda kullanılan giriş sensörleri ve çıkış cihazlarının da kütüphaneleri yazılan koda eklenerek, kullanılması elzem olan ve geliştirme aşamasında yardımcı olan fonksiyonların kullanılması konusunda da yardımcı olmaktadır.

Üçüncü taraf yazılımların proje dosyasına eklenebilmesi için Tools → Manage Libraries kısmından kütüphane yöneticisine ulaşılabilir, çevrimiçi dünyada bulunan ve topluluk tarafından geliştirilen tüm kütüphaneler projeye eklenebilir. Eklenen kütüphaneler sayesinde geliştirilecek olan projenin geliştirme aşaması oldukça hızlı gerçekleşmektedir.

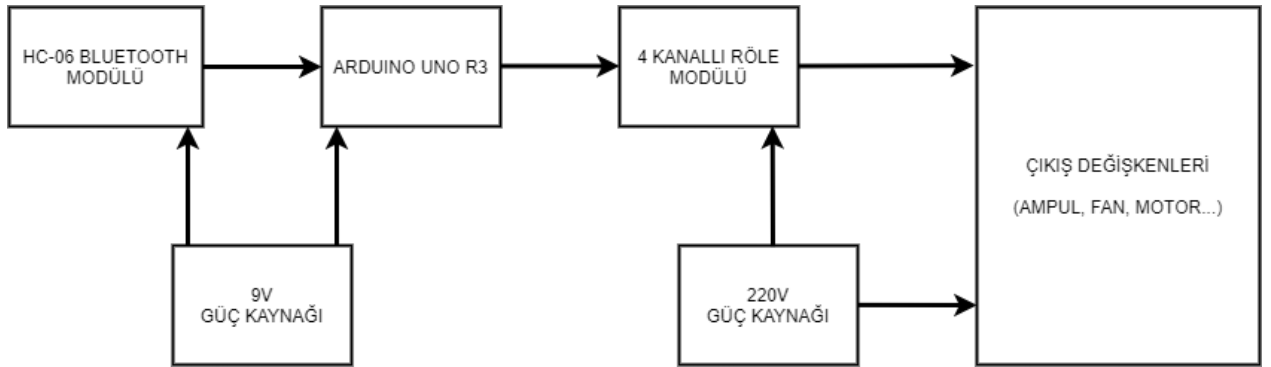
3.7. PROJE GELİŞTİRME AŞAMALARI

3.7.1. Devre Şeması



Şekil 7 Arduino Uno ve HC-06 Bluetooth Modüllü Akıllı Ev Sisteminin Devre Şeması

3.7.2. Blok Diyagramı



Şekil 8 Arduino Uno ve HC-06 Bluetooth Modüllü Akıllı Ev Sisteminin Blok Diyagramı

3.7.3. Programlama Aşaması

HC-06 Bluetooth Modülü ve Arduino ile Akıllı Ev Sistemi projesinin programlanma aşamasında aşağıdaki işlemler uygulanmıştır.

3.7.3.1. Arduino Uno Üzerindeki Geliştirme Aşamaları

Arduino Uno mikroişlemcisinde kodlamaya başlamak için öncelikle kütüphane tanımlamaları yapılır. Seri monitörü kullanabilmek için gerekli olan SoftwareSerial ve DHT11 sensörünü kullanabilmek için gerekli olan DHT kütüphaneleri eklenir.

```
#include <SoftwareSerial.h>
#include <DHT.h>;
```

Ardından HC-06 Bluetooth Modülünden gelecek olan sinyalin dinlenmesi için bir flag değişkeni ve 4 kanallı rölenin kullanımda olan 3 çıkış pini tanımlanır. Bunun yanında DHT11 sensöründen gelen nem ve sıcaklık değerlerinin tanımlanması işlemi yapılır.

```
String flag;
int relay1 = 4;
int relay2 = 5;
int relay3 = 6;
float hum;
float temp;
```

Sonrasında, bluetooth modülünün giriş çıkış pinleri, LDR ve DHT11 sensörleri için data pinleri ve DHT11 sensörünün kendisinin tanımlanması yapılır.

```
SoftwareSerial blue(10,11);
#define LDR 3
#define DHTPIN 7
#define DHTTYPE DHT11
DHT dht(DHTPIN, DHTTYPE);
```

Setup fonksiyonu oluşturulur. Setup fonksiyonu içerisinde; seri monitör ve bluetooth modülünün baud rate değerleri, DHT11 sensörünün başlama emri ve pinlerin giriş, çıkış durumları tanımlanır.

```
void setup() {
  Serial.begin(9600);
  blue.begin(9600);
  dht.begin();
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
  pinMode(relay3, OUTPUT);
}
```

Loop fonksiyonu oluşturulur. Loop fonksiyonu içerisinde; LDR sensörünün pini ve koşullar belirtilmiştir. Bluetooth modülünden ve LDR, DHT11 sensörlerinden gelen veriler sürekli olarak dinlenmektedir. Aşağıda loop fonksiyonunun içerisinde bulunan örnek bir koşul verilmiştir. Bu fonksiyon da dahil olmak üzere bütün kodları tez dosyasının sonunda bulunan Ek-A kısmında bulabilirsiniz. Aşağıda bulunan kod şu işlevi yapmaktadır; eğer bluetooth modülüne gelen değer “1 ON” ise relay1 olarak tanımlanan 1. röleye HIGH değerini göndermektedir. Bu değere karşılık 1 numaralı röleye bağlı bulunan çıkış cihazı çalışır. Ardından seri monitöre nem, sıcaklık ve röle durumları yazdırılır. Bunun yanında bluetooth ile bağlanılan mobil cihazda da ekrana aynı değerler yazdırılır.

```
if (flag == "1 ON")
{
    digitalWrite(relay1, HIGH);
    Serial.println("\nRELAY 1 ON");
    delay(2000);
    hum = dht.readHumidity();
    temp = dht.readTemperature();
    Serial.print("Nem: ");
    Serial.print(hum);
    Serial.print(" %, Sıcaklık: ");
    Serial.print(temp);
    Serial.print(" Celsius");
    blue.print("RELAY 1 ON ");
    blue.print("Nem: ");
    blue.print(hum);
    blue.print(" %, Sıcaklık: ");
    blue.print(temp);
    blue.println(" Celsius \n");
}
```

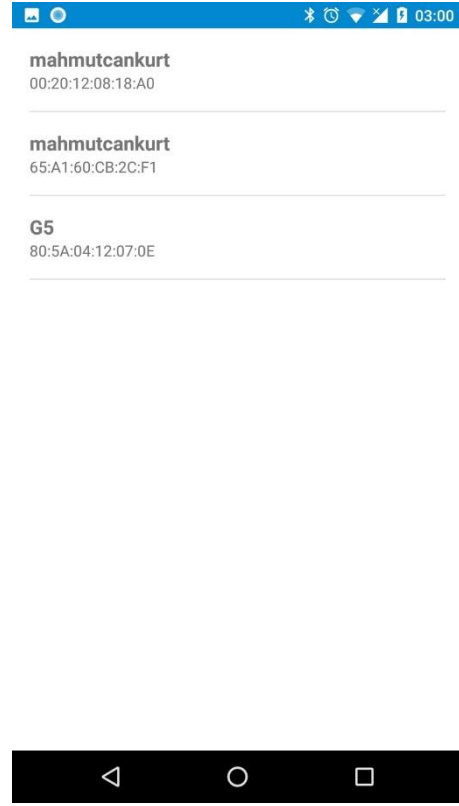
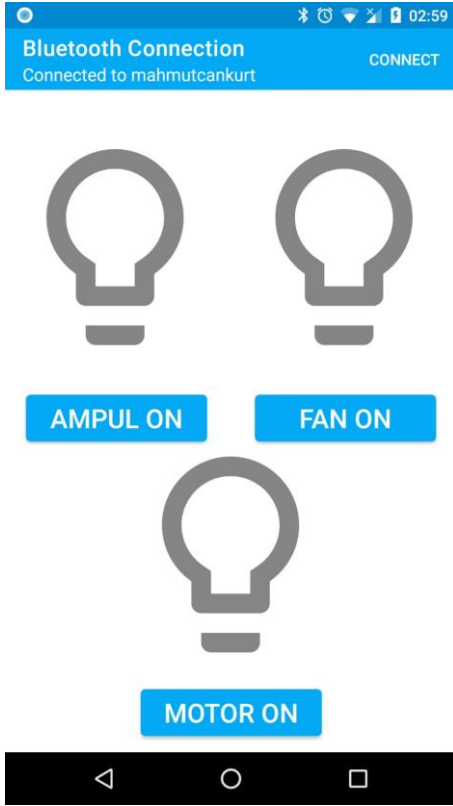
3.7.3.2. *Android Uygulama Geliştirme Aşamaları*

Arduino Uno ile Akıllı Ev Sistemi projesinin yönetilmesi için bir mobil uygulamaya ihtiyaç duyulmaktadır. Bu uygulama, HC-06 Bluetooth modülüne bağlı olan Arduino Uno kartıyla bir bluetooth bağlantısı oluşturur.

Öncelikle mobil cihazda varsayılan bluetooth adaptörü başlatılır. Ardından MAC adresi bağlanılan uzak cihazdan alınır. Edinilen MAC adresi kullanılarak bir bağlantı başlatmak için kod içerisinde ayrı bir metot oluşturulur. Bu metot, başarılı bir bağlantı kurulumu ve başarısız bir bağlantı kurulumu durumlarını yönetirken aynı zamanda bluetooth bağlantısının kapatılması durumunu da kontrol eder. Başarılı bir bağlantı oluşturulduğunda, oluşturulan yeni bir metot, veri alışverişini yönetir. Bu metot, gelen verileri okur ve uygulama tarafından oluşturulan komutu iletir.

Bu uygulama için 2 aktivite dosyası ve 2 sınıf oluşturulmuştur. Oluşturulan MainActivity dosyası etkileşimleri yönetirken SelectDeviceActivity dosyası bağlanılan bluetooth cihazına ait kullanıcı arayüzünü görüntüler. DeviceListAdapter sınıfı, daha önce bağlanılan ve o an cihaza bağlı bulunan bluetooth adaptörlerini listeler. Bu liste SelectDeviceActivity’de görüntülenir. Son olarak DeviceInfoModel sınıfı uzak cihaz bilgileri için yer tutucu görevi görmektedir. Bahsedilen aktivite ve sınıf dosyalarının içeriğindeki bazı tanımlamaları Ek-B kısmında bulabilirsiniz.

Aşağıda geliştirilen Android uygulamasının kullanıcı arayüzlerinden alınmış ekran görüntüleri bulunmaktadır. Şekil 9 yönetim panelini, Şekil 10 bluetooth adaptörlerinin listelenmesi durumunu göstermektedir.



Şekil 9 Akıllı Ev Sistemi Yönetim Paneli Şekil 10 Bluetooth Adaptörlerinin Listelenmesi

4. TARTIřMA, SONUÇ VE ÖNERİLER

Bu tez çalışmasında, mobil bir cihaz kullanılarak istenilen zamanda evde bulunan sistemlerin açılıp kapatılabilmesi veya kontrol edilebilmesi sağlanırken aynı zamanda ortam hakkında detaylı bilgiye erişilebilecek bir sistem geliştirilmiştir.

Akıllı ev teknolojileri üzerine yapılan akademik çalışmaların sayısı günümüzde oldukça fazladır. Fakat bu çalışmalar akıllı ev sistemlerinin birçok farklı yönünü ele alırken birbirleri arasında büyük ayrımlar görülmemektedir. Bu tez çalışmasında ele alınan proje de diğer akıllı ev sistemleri ve otomasyonları ile ilgili akademik çalışmalarla eşdeğer bir çalışma olmuştur.

Akıllı Ev Sistemleri aydınlatma, ısıtma, soğutma ve ortam sıcaklığı ve nem durumu hakkında detaylı bilgi vermesi sayesinde kullanıcının ev, işyeri gibi ortamlarında rahat ve huzurlu bir şekilde yaşamasını sağlamaktadır. Geliştirilen bu sistem ile enerji tasarrufu sağlanarak kullanıcının maddi yönden de rahatlamasına ve dünyadaki enerji kaynaklarının azalmasına büyük oranda ters etki sağlamaktadır.

Projenin geliştirilmesi için yapılabilecek çalışmalarda, yeni iletişim teknolojileri kullanılarak sistemin daha uzak mesafelerden kontrolünün sağlanması yapılabilir. Aynı zamanda, yalnızca elektrik tasarrufu değil, su, doğalgaz gibi değerli kaynakların da tasarrufu sağlanabilir. Ayrıca güvenlik sistemleriyle beraber çalışarak kullanıcının ve ortamın güvenliği de sağlanabilir.

KAYNAKLAR

- [1] Wikipedia. *The Veldt (Short Story)* [Online]. Web adresi: [https://en.wikipedia.org/wiki/The_Veldt_\(short_story\)](https://en.wikipedia.org/wiki/The_Veldt_(short_story)) [Ziyaret Tarihi: 17 Mayıs 2021]
- [2] Wikipedia. *Home Automation* [Online]. Web adresi: https://en.wikipedia.org/wiki/Home_automation [Ziyaret Tarihi: 17 Mayıs 2021]
- [3] Sektörüm Dergisi. 2019, *Akıllı Evler ve Akıllı Ev Sistemleri Hakkında Bilmeniz Gerekenler* [Online]. <https://www.sektorumdergisi.com/akilli-ev-nedir-akilli-ev-sistemleri-nelerdir/> [Ziyaret Tarihi: 17 Mayıs 2021]
- [4] GÜL, FATMA. 2010, *Akıllı Ev Sistemleri*, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Gazi Üniversitesi.
- [5] Wikipedia. *Amazon Echo* [Online]. Web adresi: https://en.wikipedia.org/wiki/Amazon_Echo [Ziyaret Tarihi: 17 Mayıs 2021]
- [6] Wikipedia. *Google Nest (Smart Speaker)* [Online]. Web adresi: [https://en.wikipedia.org/wiki/Google_Nest_\(smart_speakers\)](https://en.wikipedia.org/wiki/Google_Nest_(smart_speakers)) [Ziyaret Tarihi: 17 Mayıs 2021]
- [7] Wikipedia. *HomePod* [Online]. Web adresi: <https://en.wikipedia.org/wiki/HomePod> [Ziyaret Tarihi: 17 Mayıs 2021]
- [8] FORD, R., PRITONI, M., SANGUINETTI, A., KARLIN, B. (2017). Categories and Functionality of Smart Home Technology for Energy Management, *Building and Environment*, 127, 543 – 554.
- [9] SEMİZ, T.Y., 2018, *Arduino Uno Nedir? Özellikleri ve Projeleri* [Online]. Web adresi: <https://maker.robotistan.com/arduino-uno/> [Ziyaret Tarihi: 25 Mayıs 2021]
- [10] Wikipedia. *Arduino Uno* [Online]. Web Adresi: https://en.wikipedia.org/wiki/Arduino_Uno [Ziyaret Tarihi: 25 Mayıs 2021]
- [11] GÜL, MURAT, 2017, *Bluetooth Modüller HC-05 ve HC-06* [Online]. Web adresi: <https://www.muhendisbeyinler.net/bluetooth-moduller-hc-05-ve-hc-06/> [Ziyaret Tarihi: 26 Mayıs 2021]
- [12] SEMİZ, T.Y., 2018, *LDR Nedir? Foto Dirençlerin Çalışma Mantığı* [Online]. Web adresi: <https://maker.robotistan.com/ldr/> [Ziyaret Tarihi: 26 Mayıs 2021]

[13] Waveshare, 2016, *DHT11 Temperature – Humidity Sensor* [Online]. Web adresi: https://www.waveshare.com/wiki/DHT11_Temperature-Humidity_Sensor [Ziyaret Tarihi: 27 Mayıs 2021]

[14] Sunfounder, 2017, *4 Channel 5V Relay Module* [Online]. Web adresi: http://wiki.sunfounder.cc/index.php?title=4_Channel_5V_Relay_Module [Ziyaret Tarihi: 27 Mayıs 2021]

[15] Arduino cc, 2015, *Arduino Software (IDE)* [Online]. Web adresi: <https://www.arduino.cc/en/guide/environment> [Ziyaret Tarihi: 30 Mayıs 2021]

EKLER

Ek – A

Arduino IDE üzerinde yazılan, Arduino Uno mikroişlemcisine yüklenen kodlar aşağıdaki gibidir:

```
#include <SoftwareSerial.h> // Haberleşme için gerekli olan kütüphane.

#include <DHT.h>;          // DHT11 sensörü için gerekli olan kütüphane.

String flag;               // Bluetooth modülünden gelecek sinyalin dinlenmesi için değişken.

int relay1 = 4;            // 1 numaralı röleye ait pin.

int relay2 = 5;            // 2 numaralı röleye ait pin.

int relay3 = 6;            // 3 numaralı röleye ait pin.

float hum;                 // Nem değişkeni.

float temp;                // Sıcaklık değişkeni.


SoftwareSerial blue(10,11); // Bluetooth modülünün TX / RX pinleri.

#define LDR 3              // LDR'nin kullanacağı pin.

#define DHTPIN 7           // DHT11'in kullanacağı pin.

#define DHTTYPE DHT11      // DHT sensörünün tanımlanması.

DHT dht(DHTPIN, DHTTYPE); // DHT sensörünün nesne olarak oluşturulması.


void setup() {             // Kurulum fonksiyonu.

  Serial.begin(9600);      // Seri monitör baud rate tanımlaması.

  blue.begin(9600);        // Bluetooth modülü baud rate tanımlaması.

  dht.begin();             // DHT11 sensörü çalışma emri.

  pinMode(relay1, OUTPUT); // 1 numaralı röle çalışma şekli tanımı.

  pinMode(relay2, OUTPUT); // 2 numaralı röle çalışma şekli tanımı.
```

```
pinMode(relay3, OUTPUT); // 3 numaralı röle çalışma şekli tanımı.
}
```

```
void loop() { // Döngü ve çalışma fonksiyonu.

    int ldr = analogRead(A0); // LDR sensörü pin tanımlaması.

    while (Serial.available()) { // Bluetooth sensörü dinlenmesi.

        delay(3);

        char c = Serial.read();

        flag += c;

    }

    if (flag == "1 ON") // 1 numaralı rölenin çalışma koşulu.

    {

        digitalWrite(relay1, HIGH); // 1 numaralı rölenin çalışma emri.

        Serial.println("\nRELAY 1 ON"); // Seri monitöre 1 numaralı rölenin açık olduğu bilgisinin yazılması.

        delay(2000);

        hum = dht.readHumidity(); // Nem değerinin okunması.

        temp = dht.readTemperature(); // Sıcaklık değerinin okunması.

        Serial.print("Nem: "); // Nem değerinin yazdırılması.

        Serial.print(hum); // Nem değeri.

        Serial.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

        Serial.print(temp); // Sıcaklık değeri.

        Serial.print(" Celsius"); // Sıcaklık birimi.

        blue.print("RELAY 1 ON "); // Bluetooth adaptörüyle bağlanılan cihaza bilginin yazdırılması.

        blue.print("Nem: "); // Nem değerinin yazdırılması.
```

```

blue.print(hum);           // Nem değeri.

blue.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

blue.print(temp);         // Sıcaklık değeri.

blue.println(" Celsius \n"); // Sıcaklık birimi.

}

if (ldr < 850)             // LDR'den gelen veriye göre 1 numaralı rölenin çalışma koşulu.
{
    digitalWrite(relay1, HIGH);

    Serial.println("\nRELAY 1 ON");

    delay(10000);

    hum = dht.readHumidity();

    temp = dht.readTemperature();

    Serial.print("Nem: ");

    Serial.print(hum);

    Serial.print(" %, Sıcaklık: ");

    Serial.print(temp);

    Serial.print(" Celsius");

    Serial.print("Ortam Aydınlatması Yetersiz.");

    blue.print("RELAY 1 ON ");

    blue.print("Nem: ");

    blue.print(hum);

    blue.print(" %, Sıcaklık: ");

    blue.print(temp);

    blue.println(" Celsius \n");

```

```

blue.print("Ortam Aydınlatması Yetersiz.");

}

if (flag == "1 OFF")          // 1 numaralı rölenin kapanma koşulu.
{
    digitalWrite(relay1, LOW);    // 1 numaralı rölenin kapanma emri.

    Serial.println("\nRELAY 1 OFF"); // Seri monitöre 1 numaralı rölenin açık olduğu bilgisinin yazılması.

    delay(2000);

    hum = dht.readHumidity();    // Nem değerinin okunması.

    temp = dht.readTemperature(); // Sıcaklık değerinin okunması.

    Serial.print("Nem: ");        // Nem değerinin yazdırılması.

    Serial.print(hum);            // Nem değeri.

    Serial.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

    Serial.print(temp);          // Sıcaklık değeri.

    Serial.print(" Celsius");    // Sıcaklık birimi.

    blue.print("RELAY 1 OFF "); // Bluetooth adaptörüyle bağlanılan cihaza bilginin yazdırılması.

    blue.print("Nem: ");          // Nem değerinin yazdırılması.

    blue.print(hum);              // Nem değeri.

    blue.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

    blue.print(temp);            // Sıcaklık değeri.

    blue.println(" Celsius \n"); // Sıcaklık birimi.

}

if (ldr > 900) // LDR'den gelen veriye göre 1 numaralı rölenin kapanma koşulu.
{

```

```

digitalWrite(relay1, LOW);

Serial.println("\nRELAY 1 OFF");

delay(10000);

hum = dht.readHumidity();           // Nem değerinin okunması.

temp = dht.readTemperature();       // Sıcaklık değerinin okunması.

Serial.print("Nem: ");               // Nem değerinin yazdırılması.

Serial.print(hum);                   // Nem değeri.

Serial.print(" %, Sıcaklık: ");     // Sıcaklık değerinin yazdırılması.

Serial.print(temp);                  // Sıcaklık değeri.

Serial.print(" Celsius");            // Sıcaklık birimi.

Serial.print("Ortam Aydınlatması Güzel."); // Seri monitöre ortam aydınlatmasının yazılması.

blue.print("RELAY 1 OFF ");          // Bluetooth adaptörüyle bağlanılan cihaza bilginin yazdırılması.

blue.print("Nem: ");                 // Nem değerinin yazdırılması.

blue.print(hum);                     // Nem değeri.

blue.print(" %, Sıcaklık: ");       // Sıcaklık değerinin yazdırılması.

blue.print(temp);                    // Sıcaklık değeri.

blue.println(" Celsius \n");         // Sıcaklık birimi.

blue.print("Ortam Aydınlatması Güzel."); // Bluetooth adaptörüyle bağlanılan cihaza ortam
aydınlatmasının yazdırılması.

}

if (flag == "2 ON")                  // 2 numaralı rölenin çalışma koşulu.

{

digitalWrite(relay2, HIGH);

Serial.println("\nRELAY 2 ON");

```

```

delay(2000);

hum = dht.readHumidity();    // Nem değerinin okunması.

temp = dht.readTemperature(); // Sıcaklık değerinin okunması.

Serial.print("Nem: ");      // Nem değerinin yazdırılması.

Serial.print(hum);          // Nem değeri.

Serial.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

Serial.print(temp);         // Sıcaklık değeri.

Serial.print(" Celsius");   // Sıcaklık birimi.

blue.print("RELAY 2 ON ");  // Bluetooth adaptörüyle bağlanılan cihaza bilginin yazdırılması.

blue.print("Nem: ");        // Nem değerinin yazdırılması.

blue.print(hum);            // Nem değeri.

blue.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

blue.print(temp);          // Sıcaklık değeri.

blue.println(" Celsius \n"); // Sıcaklık birimi.

}

if (flag == "2 OFF")        // 2 numaralı rölenin kapanma koşulu.

{

    digitalWrite(relay2, LOW);

    Serial.println("\nRELAY 2 OFF");

    delay(2000);

    hum = dht.readHumidity();    // Nem değerinin okunması.

    temp = dht.readTemperature(); // Sıcaklık değerinin okunması.

    Serial.print("Nem: ");      // Nem değerinin yazdırılması.

    Serial.print(hum);          // Nem değeri.

```

```

Serial.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

Serial.print(temp);           // Sıcaklık değeri.

Serial.print(" Celsius");     // Sıcaklık birimi.

blue.print("RELAY 2 OFF ");   // Bluetooth adaptörüyle bağlanılan cihaza bilginin yazdırılması.

blue.print("Nem: ");          // Nem değerinin yazdırılması.

blue.print(hum);              // Nem değeri.

blue.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

blue.print(temp);             // Sıcaklık değeri.

blue.println(" Celsius \n");  // Sıcaklık birimi.

}

if (flag == "3 ON")           // 3 numaralı rölenin çalışma koşulu.
{

    digitalWrite(relay3, HIGH);

    Serial.println("\nRELAY 3 ON");

    delay(2000);

    hum = dht.readHumidity();   // Nem değerinin okunması.

    temp = dht.readTemperature(); // Sıcaklık değerinin okunması.

    Serial.print("Nem: ");      // Nem değerinin yazdırılması.

    Serial.print(hum);          // Nem değeri.

    Serial.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

    Serial.print(temp);         // Sıcaklık değeri.

    Serial.print(" Celsius");   // Sıcaklık birimi.

    blue.print("RELAY 3 ON ");  // Bluetooth adaptörüyle bağlanılan cihaza bilginin yazdırılması.

    blue.print("Nem: ");        // Nem değerinin yazdırılması.

    blue.print(hum);            // Nem değeri.

```

```

blue.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

blue.print(temp);           // Sıcaklık değeri.

blue.println(" Celsius \n"); // Sıcaklık birimi.

}

if (flag == "3 OFF")        // 3 numaralı rölenin kapanma koşulu.
{

    digitalWrite(relay3, LOW);

    Serial.println("\nRELAY 3 OFF");

    delay(2000);

    hum = dht.readHumidity(); // Nem değerinin okunması.

    temp = dht.readTemperature(); // Sıcaklık değerinin okunması.

    Serial.print("Nem: ");     // Nem değerinin yazdırılması.

    Serial.print(hum);        // Nem değeri.

    Serial.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

    Serial.print(temp);       // Sıcaklık değeri.

    Serial.print(" Celsius"); // Sıcaklık birimi.

    blue.print("RELAY 3 OFF "); // Bluetooth adaptörüyle bağlanılan cihaza bilginin yazdırılması.

    blue.print("Nem: ");      // Nem değerinin yazdırılması.

    blue.print(hum);         // Nem değeri.

    blue.print(" %, Sıcaklık: "); // Sıcaklık değerinin yazdırılması.

    blue.print(temp);        // Sıcaklık değeri.

    blue.println(" Celsius \n"); // Sıcaklık birimi.

}

flag = "";

}

```


Ek – B

HC-06 Bluetooth Modülüyle iletişimin sağlanması ve oluşturulan akıllı ev sisteminin kontrolü için kullanılacak olan Android uygulamasına ait kodlar aşağıdaki gibidir.

- DeviceInfoModel.java

```
package com.example.mahmutbt;
```

```
public class DeviceInfoModel {
```

```
    private String deviceName, deviceHardwareAddress;
```

```
    public DeviceInfoModel(){ }
```

```
    public DeviceInfoModel(String deviceName, String deviceHardwareAddress){
```

```
        this.deviceName = deviceName;
```

```
        this.deviceHardwareAddress = deviceHardwareAddress;
```

```
    }
```

```
    public String getDeviceName(){return deviceName;}
```

```
    public String getDeviceHardwareAddress(){return deviceHardwareAddress;}
```

```
}
```

- DeviceListAdapter.java

```

package com.example.mahmutbt;

import android.content.Intent;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;
import androidx.recyclerview.widget.RecyclerView;
import java.util.List;

public class DeviceListAdapter extends RecyclerView.Adapter<RecyclerView.ViewHolder> {
    private Context context;
    private List<Object> deviceList;
    public static class ViewHolder extends RecyclerView.ViewHolder {
        TextView textName, textAddress;
        LinearLayout linearLayout;
        public ViewHolder(View v) {
            super(v);
            textName = v.findViewById(R.id.textViewDeviceName);
            textAddress = v.findViewById(R.id.textViewDeviceAddress);
            linearLayout = v.findViewById(R.id.linearLayoutDeviceInfo);
        }
    }

    public DeviceListAdapter(Context context, List<Object> deviceList) {
        this.context = context;
        this.deviceList = deviceList;
    }
}

```

```

public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
    View v = LayoutInflater.from(parent.getContext()).inflate(R.layout.device_info_layout, parent,
false);
    ViewHolder vh = new ViewHolder(v);
    return vh;
}

@Override
public void onBindViewHolder(final RecyclerView.ViewHolder holder, final int position) {
    ViewHolder itemHolder = (ViewHolder) holder;

    final DeviceInfoModel deviceInfoModel = (DeviceInfoModel) deviceList.get(position);
    itemHolder.textName.setText(deviceInfoModel.getDeviceName());
    itemHolder.textAddress.setText(deviceInfoModel.getDeviceHardwareAddress());

    // When a device is selected
    itemHolder.linearLayout.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View view) {
            Intent intent = new Intent(context,MainActivity.class);

            // Send device details to the MainActivity
            intent.putExtra("deviceName", deviceInfoModel.getDeviceName());
            intent.putExtra("deviceAddress",deviceInfoModel.getDeviceHardwareAddress());

            // Call MainActivity
            context.startActivity(intent);
        }
    });
}

@Override
public int getItemCount() {
    int dataCount = deviceList.size();
    return dataCount;
}
}

```

- **SelectDeviceActivity.java**

```
package com.example.mahmutbt;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.DefaultItemAnimator;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.os.Bundle;
import android.view.View;

import com.google.android.material.snackbar.Snackbar;

import java.util.ArrayList;
import java.util.List;
import java.util.Set;

public class SelectDeviceActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_select_device);

        // Bluetooth Setup
        BluetoothAdapter bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

        // Get List of Paired Bluetooth Device
```

```

Set<BluetoothDevice> pairedDevices = bluetoothAdapter.getBondedDevices();
List<Object> deviceList = new ArrayList<>();
if (pairedDevices.size() > 0) {
    // There are paired devices. Get the name and address of each paired device.
    for (BluetoothDevice device : pairedDevices) {
        String deviceName = device.getName();
        String deviceHardwareAddress = device.getAddress(); // MAC address

        DeviceInfoModel deviceInfoModel = new
DeviceInfoModel(deviceName,deviceHardwareAddress);

        deviceList.add(deviceInfoModel);
    }
    // Display paired device using recyclerView
    RecyclerView recyclerView = findViewById(R.id.recyclerViewDevice);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
    DeviceListAdapter deviceListAdapter = new DeviceListAdapter(this,deviceList);
    recyclerView.setAdapter(deviceListAdapter);
    recyclerView.setItemAnimator(new DefaultItemAnimator());
} else {
    View view = findViewById(R.id.recyclerViewDevice);

    Snackbar snackbar = Snackbar.make(view, "Activate Bluetooth or pair a Bluetooth device",
Snackbar.LENGTH_INDEFINITE);

    snackbar.setAction("OK", new View.OnClickListener() {
        @Override
        public void onClick(View view) { }
    });
    snackbar.show();
}

}
}

```

- MainActivity.java

Geliştirilen Android uygulamasının ana işlevlerini yerine getiren aktivite dosyasıdır. Aşağıda bulunan kod bloğu, buton aktivitelerini ve gönderilecek olan mesajın emrinin verilmesini sağlamaktadır.

```
buttonToggle.setOnClickListener(new View.OnClickListener() {
    @SuppressWarnings("SetTextI18n")
    @Override
    public void onClick(View view) {
        String cmdText = null;
        if(!isButtonToggleOn){
            buttonToggle.setText("1 OFF");
            cmdText = "1 ON"; // Arduino üzerinden röle 1'i çalıştırmak için.
        }
        else{
            buttonToggle.setText("1 ON");
            cmdText = "1 OFF"; // Arduino üzerinden röle 1'i kapatmak için.
        }
        isButtonToggleOn = !isButtonToggleOn;
        connectedThread.write(cmdText);
    }
});

buttonToggle2.setOnClickListener(new View.OnClickListener() {
    @SuppressWarnings("SetTextI18n")
    @Override
    public void onClick(View view) {
        String cmdText = null;
        if(!isButtonToggle2On){
            buttonToggle2.setText("2 OFF");
            cmdText = "2 ON"; // Arduino üzerinden röle 2'yi çalıştırmak için.
```

```

    }
    else{
        buttonToggle2.setText("2 ON");
        cmdText = "2 OFF"; // Arduino üzerinden röle 2'yi kapatmak için.
    }
    isButtonToggle2On = !isButtonToggle2On;
    connectedThread.write(cmdText);
}

});

buttonToggle3.setOnClickListener(new View.OnClickListener() {
    @SuppressWarnings("SetTextI18n")
    @Override
    public void onClick(View view) {
        String cmdText = null;
        if(!isButtonToggle3On){
            buttonToggle3.setText("3 OFF");
            cmdText = "3 ON"; // Arduino üzerinden röle 3'ü çalıştırmak için.
        }
        else{
            buttonToggle2.setText("3 ON");
            cmdText = "3 OFF"; // Arduino üzerinden röle 3'ü kapatmak için.
        }
        isButtonToggle3On = !isButtonToggle3On;
        connectedThread.write(cmdText);
    }
});

```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Mahmut Can KURT
 Doğum Tarihi ve Yeri : 01.04.1998, BAKIRKÖY
 Yabancı Dili : İngilizce
 E-Posta : is.mahmutcankurt@gmail.com



ÖĞRENİM DURUMU

Derece	Alan	Okul / Üniversite	Mezuniyet Yılı
Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2021
Lise	Sayısal	Pendik Fatih A.L.	2016

İŞ / STAJ DENEYİMLERİ

Yaz Stajı (Haziran 2020 – Ağustos 2020)

KOBISI, İstanbul Teknik Üniversitesi, Türkiye

Gönüllü Staj (Ekim 2019 – Aralık 2019)

CARPEDU, Düzce Üniversitesi, Türkiye

Yaz Stajı (Ağustos 2019 – Eylül 2019)

TOSIATECH, İstanbul, Türkiye

Gönüllü Staj (Ağustos 2018 – Eylül 2018)

TOSIATECH, İstanbul, Türkiye