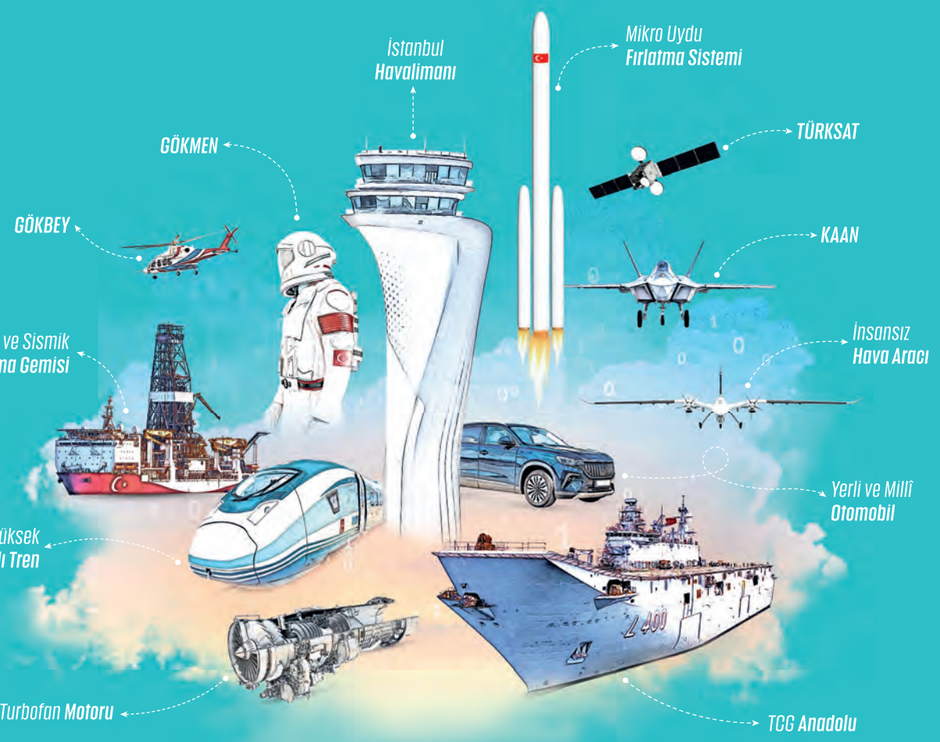
211650_Nesnelerin Interneti_11-12 Kapak.pdf

Bu kitaba sığmayan daha neler var!

eba
www.eba.gov.tr

Karekodu okut, bu kitapla ilgili EBA içeriklerine ulaş!

Teknoloji Yolculuğumuz



BU DERS KİTABI MİLLÎ EĞİTİM BAKANLIĞINCA ÜCRETSİZ OLARAK VERİLMİŞTİR.
PARA İLE SATILAMAZ.

ISBN: 978-975-11-7125-2

Bandrol Uygulamasına İlişkin Usul ve Esaslar Hakkında Yönetmeliğin Beşinci Maddesinin İkinci Fıkrası Çerçevesinde Bandrol Taşıması Zorunlu Değildir.

BİLİŞİM TEKNOLOJİLERİ ALANI
NESNELERİN İNTERNETİ 11-12 DERS MATERYALİ

MESLEKİ VE TEKNİK ANADOLU LİSESİ

BİLİŞİM TEKNOLOJİLERİ ALANI

11-12 DERS MATERYALİ

NESNELERİN İNTERNETİ

MESLEKİ VE TEKNİK ANADOLU LİSESİ

BİLİŞİM TEKNOLOJİLERİ ALANI

NESNELERİN İNTERNETİ

11-12

DERS MATERYALİ

Yazarlar

Ahmet KAHRAMAN

Ali GÖKDEMİR

Atilla ÇETİN

Murat KARATAŞ

Mustafa ÖZER

Tarık ÜNLÜ

MİLLÎ EĞİTİM BAKANLIĞI YAYINLARI.....: 8349
YARDIMCI VE KAYNAK KİTAPLAR DİZİSİ.....: 2241

Her hakkı saklıdır ve Millî Eğitim Bakanlığına aittir. Ders materyalinin metin, soru ve şekilleri kısmen de olsa hiçbir surette alınıp yayımlanamaz.

HAZIRLAYANLAR

Dil Uzmanı
Osman Nuri GÜVEN

Program Geliştirme Uzmanı
Zeki BİLGİLİ

Ölçme ve Değerlendirme Uzmanı
Tülay ENGİN

Rehberlik Uzmanı
Gülşen YALIN

Görsel Tasarım Uzmanı
Tuğba SANCI

ISBN: 978-975-11-7125-2

Millî Eğitim Bakanlığının 24.12.2020 gün ve 18433886 sayılı oluru ve Meslekî ve Teknik Eğitim Genel Müdürlüğünce ders materyali olarak hazırlanmıştır.

BASKI
Ayrıntı Basım Yayım ve Mat. Hiz. Sanayi Ticaret A.Ş.
Saray Mahallesi 126. Cadde No: 20
Kahramankazan/Ankara
Sertifika No: 49599

İSTİKLÂL MARŞI

Korkma, sönmez bu şafaklarda yüzen al sancak;
Sönmeden yurdumun üstünde tüten en son ocak.
O benim milletimin yıldızıdır, parlayacak;
O benimdir, o benim milletimindir ancak.

Çatma, kurban olayım, çehreni ey nazlı hilâl!
Kahraman ırkıma bir gül! Ne bu şiddet, bu celâl?
Sana olmaz dökülen kanlarımız sonra helâl.
Hakkıdır Hakk'a tapan milletimin istiklâl.

Ben ezelden beridir hür yaşadım, hür yaşarım.
Hangi çılgın bana zincir vuracakmış? Şaşarım!
Kükremiş sel gibiyim, bendimi çiğner, aşarım.
Yırtarım dağları, enginlere sığmam, taşarım.

Garbın âfâkını sarmışsa çelik zırhlı duvar,
Benim iman dolu göğsüm gibi serhaddim var.
Ulusun, korkma! Nasıl böyle bir imanı boğar,
Medeniyyet dediğin tek dişi kalmış canavar?

Arkadaş, yurduma alçakları uğratma sakın;
Siper et gövdeni, dursun bu hayâsızca akın.
Doğacaktır sana va'dettiği günler Hakk'ın;
Kim bilir, belki yarın, belki yarından da yakın.

Bastığın yerleri toprak diyerek geçme, tanı:
Düşün altındaki binlerce kefensiz yatanı.
Sen şehit oğlusun, incitme, yazıktır, atanı:
Verme, dünyaları alsan da bu cennet vatanı.

Kim bu cennet vatanın uğruna olmaz ki feda?
Şüheda fışkıracak toprağı sıksan, şüheda!
Cânı, cânânı, bütün varımı alsın da Huda,
Etmesin tek vatanımdan beni dünyada cüda.

Ruhumun senden İlâhî, şudur ancak emeli:
Değmesin mabedimin göğsüne nâmahrem eli.
Bu ezanlar -ki şehadetleri dinin temeli-
Ebedî yurdumun üstünde benim inlemeli.

O zaman vecd ile bin secde eder -varsa- taşım,
Her cerîhamdan İlâhî, boşanıp kanlı yaşım,
Fışkırır ruh-ı mücerret gibi yerden na'şım;
O zaman yükselerek arşa değer belki başım.

Dalgalan sen de şafaklar gibi ey şanlı hilâl!
Olsun artık dökülen kanlarımın hepsi helâl.
Ebediyyen sana yok, ırkıma yok izmihlâl;
Hakkıdır hür yaşamış bayrağımın hürriyyet;
Hakkıdır Hakk'a tapan milletimin istiklâl!

Mehmet Âkif Ersoy

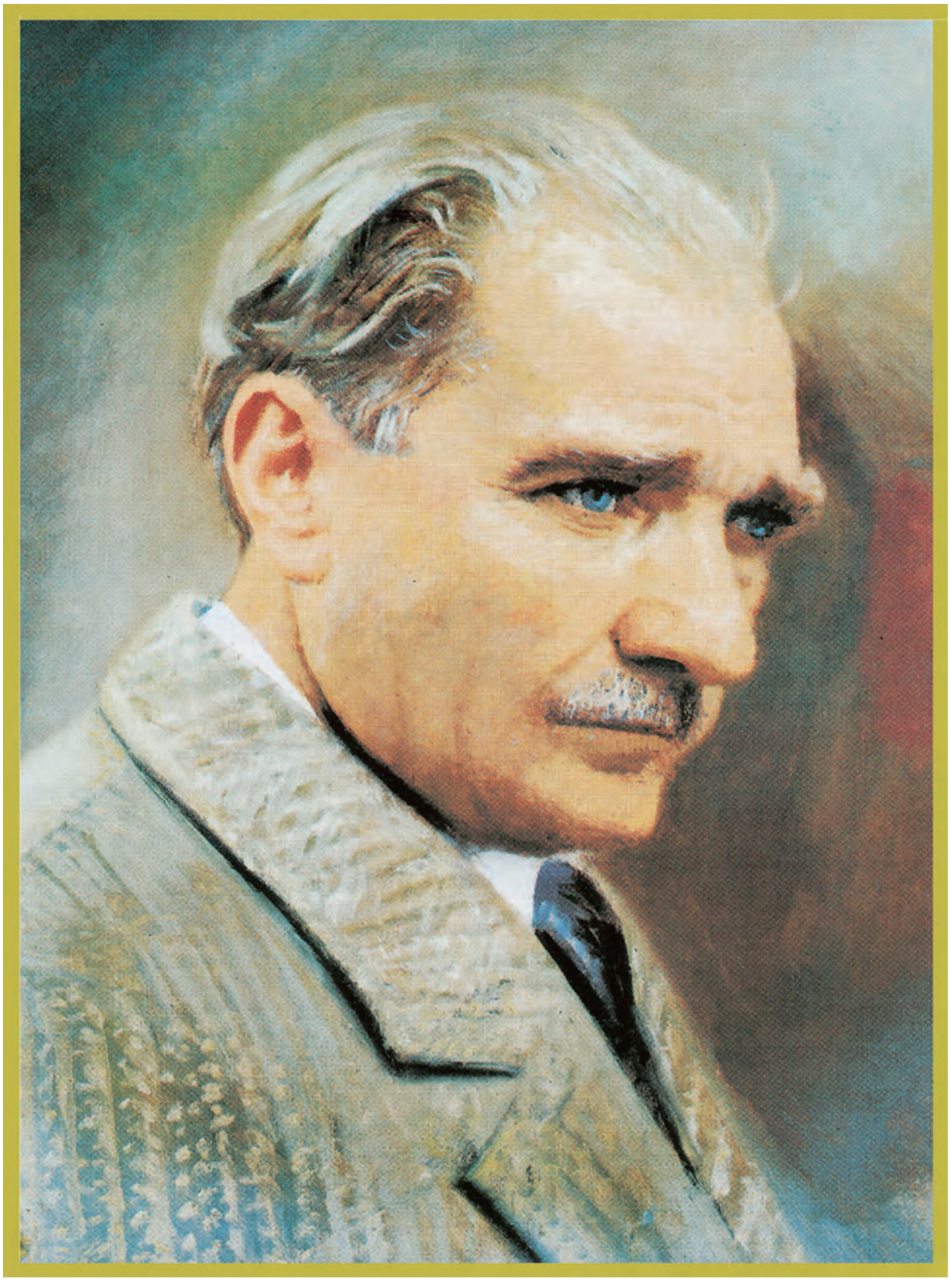
GENÇLİĞE HİTABE

Ey Türk gençliği! Birinci vazifen, Türk istiklâlini, Türk Cumhuriyetini, ilelebet muhafaza ve müdafaa etmektir.

Mevcudiyetinin ve istikbalinin yegâne temeli budur. Bu temel, senin en kıymetli hazinendir. İstikbalde dahi, seni bu hazineden mahrum etmek isteyecek dâhilî ve hâricî bedhahların olacaktır. Bir gün, istiklâl ve cumhuriyeti müdafaa mecburiyetine düşersen, vazifeye atılmak için, içinde bulunacağın vaziyetin imkân ve şeraitini düşünmeyeceksin! Bu imkân ve şerait, çok namüsait bir mahiyette tezahür edebilir. İstiklâl ve cumhuriyetine kastedecek düşmanlar, bütün dünyada emsali görülmemiş bir galibiyetin mümessili olabilirler. Cebren ve hile ile aziz vatanın bütün kaleleri zapt edilmiş, bütün tersanelerine girilmiş, bütün orduları dağıtılmış ve memleketin her köşesi bilfiil işgal edilmiş olabilir. Bütün bu şeraitten daha elîm ve daha vahim olmak üzere, memleketin dâhilinde iktidara sahip olanlar gaflet ve dalâlet ve hattâ hıyanet içinde bulunabilirler. Hattâ bu iktidar sahipleri şahsî menfaatlerini, müstevlîlerin siyasî emelleriyle tevhit edebilirler. Millet, fakr u zaruret içinde harap ve bîtap düşmüş olabilir.

Ey Türk istikbalinin evlâdı! İşte, bu ahval ve şerait içinde dahi vazifen, Türk istiklâl ve cumhuriyetini kurtarmaktır. Muhtaç olduğun kudret, damarlarındaki asil kanda mevcuttur.

Mustafa Kemal Atatürk



MUSTAFA KEMAL ATATÜRK

İÇİNDEKİLER

DERS MATERYALİNİN TANITIMI 14

1.ÖĞRENME BİRİMİ

1. NESNELER VE BAĞLANTILAR 16
1.1. NESNELERİN İNTERNETİ (IoT) 18
1.1.1. Nesnelerin İnternetinin Kullanıldığı Alanlar 19
1.1.2. Nesnelerin İnternetinin Kullanıldığı Ürünler 20
1.1.3. Nesnelerin İnternetinin Avantajları ve Dezavantajları 20
1.2. IoT BİLEŞENLERİ 21
1.2.1. Sensörler 21
1.2.2. Aktüatörler 21
1.2.3. Kontrolörler 21
1.2.4. Bağlantı 21
1.2.5. Analiz (Veri İşleme) 32
1.2.6. Kullanıcı Arayüzü 32
1.2.7. Bulut Bilişim 32
1.3. İLETİŞİM MODELLERİ 34
1.3.1. Katmanlı Ağ Modelleri 34
1.3.1.1. OSI ve TCP/IP Modelleri 34
1.3.1.2. IoT Referans Modeli 35
1.3.2. Bağlantı Seviyelerine Dayalı Model 35
1.3.3. İletişim Türlerine Dayalı Model 35
1.3.3.1. İstek ve Yanıt Modeli 36
1.3.3.2. Yayın ve Abone Modeli 36
1.3.3.3. İtme ve Çekme Modeli 36
1.3.3.4. Özel Çift Modeli 37
1.3.4. Üç Katmanlı IoT Mimari Modeli 37
1.4. VERİ GİZLİLİĞİ 38
1.4.1. Metadata 38
1.4.2. IoT Cihazlarının Gizliliğe Etkisi 42
1.4.3. IoT Cihazlarında Güvenlik 42
ÖLÇME VE DEĞERLENDİRME 44

2.ÖĞRENME BİRİMİ

2. DEVRE ELEMANLARI, MİKRODENETLEYİCİLER VE SENSÖRLER 46
2.1. DEVRE ELEMANLARI 48
2.1.1. Direnç 48
2.1.2. LED 50
2.1.2.1. Buzzer 55
2.1.2.2. Transistör 55
2.1.2.3. Röle 56
2.1.2.4. Motor Sürücü Devreleri 57
2.1.2.5. Optokuplör 58

2.1.3. Buton 60
2.1.4. Anahtar 60
2.2. BREADBOARD KULLANIMI 62
2.3. MİKRODENETLEYİCİLER 64
2.3.1. Nesnelerin İnterneti Uygulamalarında Kullanılan Mikrodenetleyici Devre Kartları 65
2.3.1.1. Arduino UNO Mikrodenetleyici Kartı 65
2.3.1.2. Arduino UNO Wi-Fi Mikrodenetleyici Kartı 67
2.3.1.3. NodeMCU Mikrodenetleyici Kart 68
2.3.1.4. ESP32 Mikrodenetleyici Kart 68
2.4. SENSÖRLER 70
2.4.1. Çıkış Türüne Göre Sensörler 70
2.4.1.1. Sıcaklık Sensörü 70
2.4.1.2. Ses Sensörü 76
2.4.1.3. Işık Seviye Sensörü 80
2.4.1.4. Mesafe Sensörü 80
2.4.1.5. Gaz Sensörü 81
2.4.1.6. Alev Sensörü 84
2.4.1.7. Su Taşkını Sensörü 87
2.4.1.8. Manyetik Alan Sensörü 89
ÖLÇME VE DEĞERLENDİRME 96

3.ÖĞRENME BİRİMİ

3. NESNELERİN İNTERNETİNDE PROGRAMLAMA 98
3.1. BLOK TEMELLİ PROGRAMLAMA 100
3.1.1. Blok Temelli Uygulama Aracı 100
3.1.2. Blok Programlama 101
3.2. PHYTON İLE PROGRAMLAMA 108
3.3. VERİ İŞLEME SÜREÇLERİ 111
3.4. API'LER 112
3.4.1. API'lerin Çalışması 112
3.4.2. SOAP API 112
3.4.3. REST API 113
3.4.4. HTTP Durum Kodları 118
3.4.5. RESTful API 118
3.5. KOD GÜVENLİĞİ 119
3.5.1. HTML İnjeksiyon Zafiyeti 121
3.6. RASPBERRY PI KULLANIMI 123
3.6.1. Raspberry Pi Donanım Özellikleri 123
3.6.2. Raspberry Pi Donanımına İşletim Sistemi Kurulumu 125
3.6.3. Raspberry Pi Donanımında Temel Linux Komutlar 127
3.6.4. Raspberry Pi Donanımında Phyton Diliyle Uygulamalar 130
3.6.5. Raspberry Pi Donanımıyla Ev Otomasyonu Projesi 144
3.7. SİMÜLASYON ARACI 146
ÖLÇME VE DEĞERLENDİRME 153

4.ÖĞRENME BİRİMİ

4. BİLGİSAYAR AĞLARI, SİS VE BULUT BİLİŞİM 156
4.1. YEREL VE GENEL ALAN AĞLARI 158
4.1.1. IoT Ekosistemi 158
4.1.2. LAN 159
4.1.3. WAN 159
4.1.4. Diğer Ağ KAvramları 160
4.2. KABLOLU VE KABLOSUZ ORTAMLAR 160
4.3. AĞ PROTOKOLLERİ 161
4.3.1. İnternet Protokolü 161
4.3.2. TCP ve UDP Protokolleri 162
4.3.3. Diğer Protokoller ve Teknolojiler 163
4.4. IoT KABLOSUZ İLETİŞİM TEKNOLOJİLERİ 166
4.5. IoT PROTOKOLLERİ 169
4.5.1. Altyapı Protokolleri 170
4.5.2. Servis Keşif Protokolleri 170
4.5.3. Uygulama Protokolleri 171
4.5.3.1. Message Queuing Telemetry Transport 171
4.5.3.2. Constrained Application Protocol 185
4.5.3.3. Advanced Message Queuing Protocol 186
4.5.3.4. Extensible Messaging and Presence Protocol 187
4.6. SIS VE BULUT BİLİŞİM 187
4.6.1. Bulut Bilişim Modeli 187
4.6.1.1. Bulut Bilişim Erişim Modelleri 188
4.6.1.2. Bulut Bilişim Hizmet Modelleri 189
4.6.1.3. IoT Bulut Bilişim Hizmetleri 190
4.6.2. Sis Bilişim Modeli 192
4.7. BÜYÜK VERİ 193
4.7.1. Büyük Veri Kullanım Örnekleri 197
4.7.2. Büyük Veri Depolama Ortamları 198
4.7.3. Veri Görselleştirme Araçları 198
4.8. BULUT BİLİŞİMDE GÜVENLİK 199
ÖLÇME VE DEĞERLENDİRME 200

5.ÖĞRENME BİRİMİ

5. NESNELERİN İNTERNETİNDE GÜVENLİK 202
5.1. IoT'TA GÜVENLİK RİSKLERİ 204
5.2. IoT SİSTEM MİMARİLERİ 206
5.2.1. Uygulama Katmanı 206
5.2.2. İletişim Katmanı 206
5.2.3. Donanım Katmanı 207
5.3. IoT DONANIM KATMANI GÜVENLİĞİ 207
5.3.1. Fiziksel Katmanda Karşılaşılabilecek Güvenlik Riskleri ve Alınacak Önlemler 207
5.3.1.1. Düğümü Ele Geçirme Saldırısı 207

5.3.1.2. Yayın Bozma Saldırısı 207
5.3.1.3. Zararlı Kod Aşılama Saldırısı 208
5.3.1.4. Uykudan Yoksun Bırakma Saldırısı 208
5.4. IoT İLETİŞİM KATMANI GÜVENLİĞİ 208
5.4.1. İletişim Katmanında Karşılaşılabilecek Güvenlik Riskleri ve Alınacak Önlemler 208
5.4.1.1. Gider Deliği Saldırısı 208
5.4.1.2. Karadelik Saldırısı 209
5.4.1.3. Solucan Deliği Saldırısı 209
5.4.1.4. Sybil Saldırısı 210
5.4.1.5. Merhaba Seli Saldırısı 210
5.5. UYGULAMA KATMANI GÜVENLİĞİ 210
5.5.1. Uygulama Katmanında Karşılaşılabilecek Güvenlik Riskleri ve Alınacak Önlemler 210
5.5.1.1. Aradaki Adam Saldırıları 210
5.5.1.2. SQL Injection Saldırısı 211
5.5.1.3. Kopyalama Saldırısı 211
5.5.1.4. Yeniden Programlama Saldırısı 211
5.5.1.5. Yol Tabanlı Servis Yalanlaması Saldırısı 211
5.5.1.6. DoS Saldırısı 212
5.5.1.7. Algılayıcı Düğümün Boğulması Saldırısı 212
ÖLÇME VE DEĞERLENDİRME 213

6.ÖĞRENME BİRİMİ

6. IoT UYGULAMALARI 214
6.1. ENDÜSTRİYEL IoT UYGULAMALARI 216
6.1.1. Enerji Sektöründe IoT 217
6.1.2. Sağlık Sektöründe IoT 217
6.1.3. Akıllı Şehirlerde IoT 218
6.1.4. Tarım Uygulamalarında IoT 219
6.2. IoT GÜVENLİĞİ 219
6.2.1. Güvenli Cihaz Geliştirme İlkesi 220
6.2.2. Güvenli Bağlantı Oluşturma İlkesi 220
6.2.3. Ağ Güvenliğini Sağlama İlkesi 221
6.2.4. Veri Depolama Güvenliği İlkesi 221
6.3. IoT SİSTEM UYGULAMALARI 222
6.3.1. Sağlık Alanında IoT Uygulaması 222
6.4. IoT'TA MAKİNE ÖĞRENMESİ VE YAPAY ZEKÂ 242
ÖLÇME VE DEĞERLENDİRME 243

7.ÖĞRENME BİRİMİ

7. IoT ÇÖZÜMLERİ GELİŞTİRME 244
7.1. IoT UYGULAMA TASARLAMA 246
7.2. IoT PROTOTİPİ 280
ÖLÇME VE DEĞERLENDİRME 290

CEVAP ANAHTARLARI 292
KAYNAKÇA 294

DERS MATERYALİNİN TANITIMI

Öğrenme biriminin temel kavramlarını gösterir.

Öğrenme birimi adını gösterir.

Öğrenme biriminde neler öğrenileceğini gösterir.

Öğrenme birimi konularını gösterir.

Öğrenme birimi numarasını gösterir.

Öğrenme birimi görselini gösterir.

Öğrenme biriminin başında yapılacak ön çalışmaları gösterir.

Öğrenme birimi karekodunu gösterir.

Öğrenme birimi linkini gösterir.

DERS MATERYALİNİN TANITIMI

Konu ile ilgili görselleri gösterir.

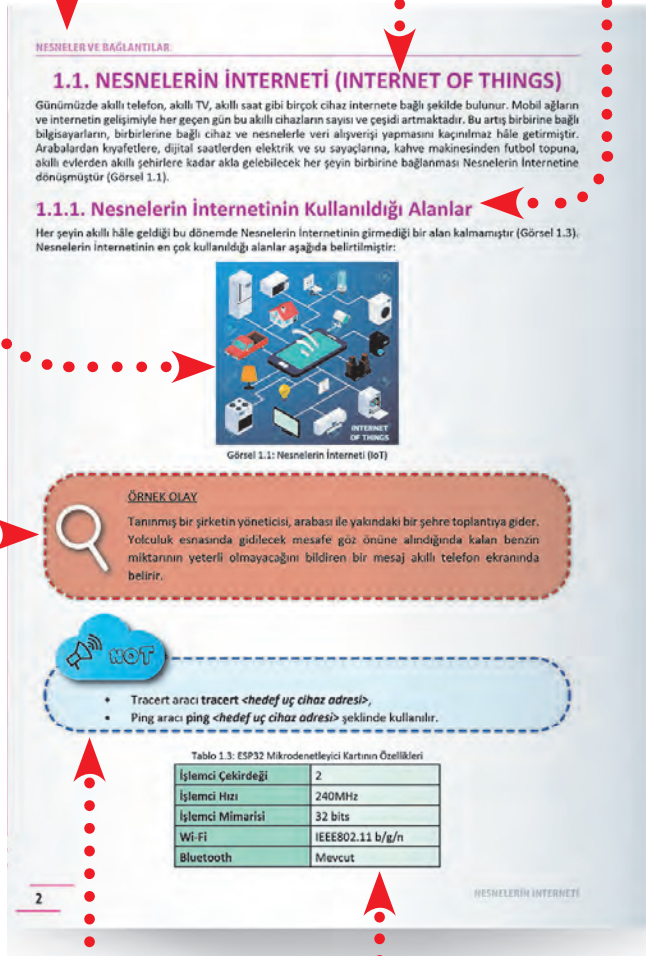
Uygulama etkinliğini gösterir.

Öğrenme birimi adını gösterir.

Konu başlıklarını gösterir.

Araştırma etkinliğini gösterir.

Sıra sizde etkinliğini gösterir.


NESNELER VE BAĞLANTILAR

1.1. NESNELERİN İNTERNETİ (INTERNET OF THINGS)

Günümüzde akıllı telefon, akıllı TV, akıllı saat gibi birçok cihaz internete bağlı şekilde bulunur. Mobil ağların ve internetin gelişimiyle her geçen gün bu akıllı cihazların sayısı ve çeşidi artmaktadır. Bu artış birbirine bağlı bilgisayarların, birbirlerine bağlı cihaz ve nesnelerle veri alışverişi yapmasını kaçınılmaz hâle getirmiştir. Arabalardan kıyafetlere, dijital saatlerden elektrik ve su sayaçlarına, kahve makinesinden futbol topuna, akıllı evlerden akıllı şehirlere kadar akla gelebilecek her şeyin birbirine bağlanması Nesnelerin İnternetine dönüşmüştür (Görsel 1.1).

1.1.1. Nesnelerin İnternetinin Kullanıldığı Alanlar

Her şeyin akıllı hâle geldiği bu dönemde Nesnelerin İnternetinin girmediği bir alan kalmamıştır (Görsel 1.3). Nesnelerin İnternetinin en çok kullanıldığı alanlar aşağıda belirtilmiştir:

Görsel 1.1: Nesnelerin İnterneti (IoT)

ÖRNEK OLAY

Tanınmış bir şirketin yöneticisi, arabası ile yakındaki bir şehre toplantıya gider. Yolculuk esnasında gidilecek mesafe göz önüne alındığında kalan benzin miktarının yeterli olmayacağını bildiren bir mesaj akıllı telefon ekranında belirir.

IoT

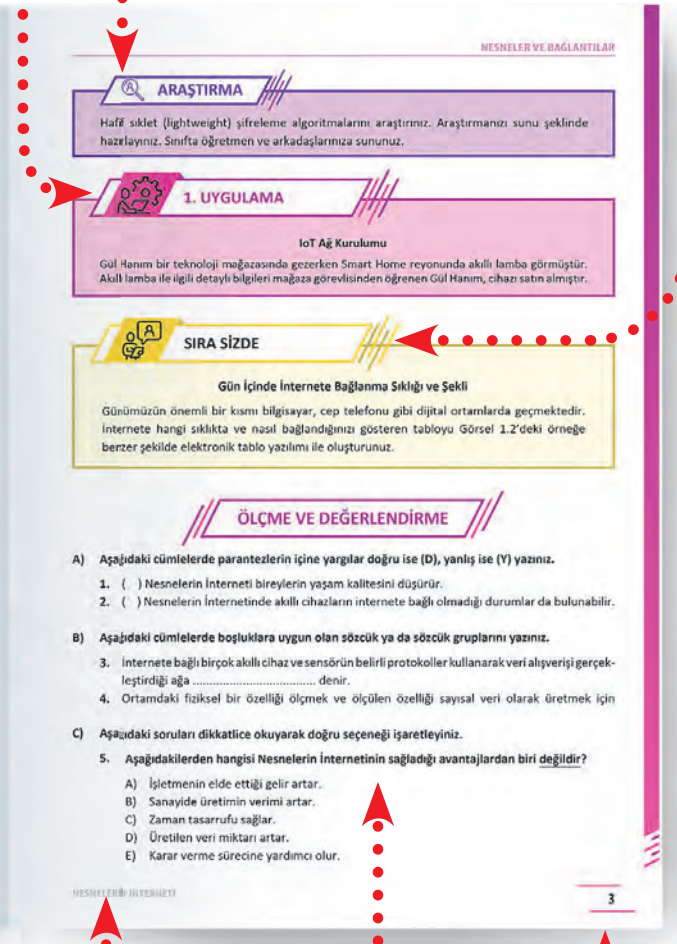
- Tracert aracı **tracert** *<hedef uç cihaz adresi>*,
- Ping aracı **ping** *<hedef uç cihaz adresi>* şeklinde kullanılır.

Tablo 1.3: ESP32 Mikrodenetleyici Kartının Özellikleri

İşlemci Çekirdeği	2
İşlemci Hızı	240MHz
İşlemci Mimarisi	32 bits
Wi-Fi	IEEE802.11 b/g/n
Bluetooth	Mevcut

2

NESNELERİN İNTERNETİ


NESNELER VE BAĞLANTILAR

ARAŞTIRMA

Hafif siklet (lightweight) şifreleme algoritmalarını araştırınız. Araştırmanızı sunu şeklinde hazırlayınız. Sınıfta öğretmen ve arkadaşlarınıza sununuz.

1. UYGULAMA

IoT Ağ Kurulumu

Gül Hanım bir teknoloji mağazasında gezerken Smart Home reyonunda akıllı lamba görmüştür. Akıllı lamba ile ilgili detaylı bilgileri mağaza görevlisinden öğrenen Gül Hanım, cihazı satın almıştır.

SIRA SİZDE

Gün İçinde İnternete Bağlanma Sıklığı ve Şekli

Günümüzün önemli bir kısmı bilgisayar, cep telefonu gibi dijital ortamlarda geçmektedir. İnternete hangi sıklıkta ve nasıl bağlandığınızı gösteren tabloyu Görsel 1.2'deki örneğe benzer şekilde elektronik tablo yazılımı ile oluşturunuz.

ÖLÇME VE DEĞERLENDİRME

A) **Aşağıdaki cümlelerde parantezlerin içine yargılar doğru ise (D), yanlış ise (Y) yazınız.**

1. ( ) Nesnelerin İnterneti bireylerin yaşam kalitesini düşürür.
2. ( ) Nesnelerin İnternetinde akıllı cihazların internete bağlı olmadığı durumlar da bulunabilir.

B) **Aşağıdaki cümlelerde boşluklara uygun olan sözcük ya da sözcük gruplarını yazınız.**

3. İnternete bağlı birçok akıllı cihaz ve sensörün belirli protokoller kullanarak veri alışverişi gerçekleştirdiği ağa ..... denir.
4. Ortamdaki fiziksel bir özelliği ölçmek ve ölçülen özelliği sayısal veri olarak üretmek için

C) **Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.**

5. **Aşağıdakilerden hangisi Nesnelerin İnternetinin sağladığı avantajlardan biri değildir?**

A) İşletmenin elde ettiği gelir artar.
B) Sanayide üretimin verimi artar.
C) Zaman tasarrufu sağlar.
D) Üretilen veri miktarı artar.
E) Karar verme sürecine yardımcı olur.

NESNELERİN İNTERNETİ

3

Konuyla ilgili notları gösterir.

Tabloları gösterir.

Kitap adını gösterir.

Sayfa numaralarını gösterir.

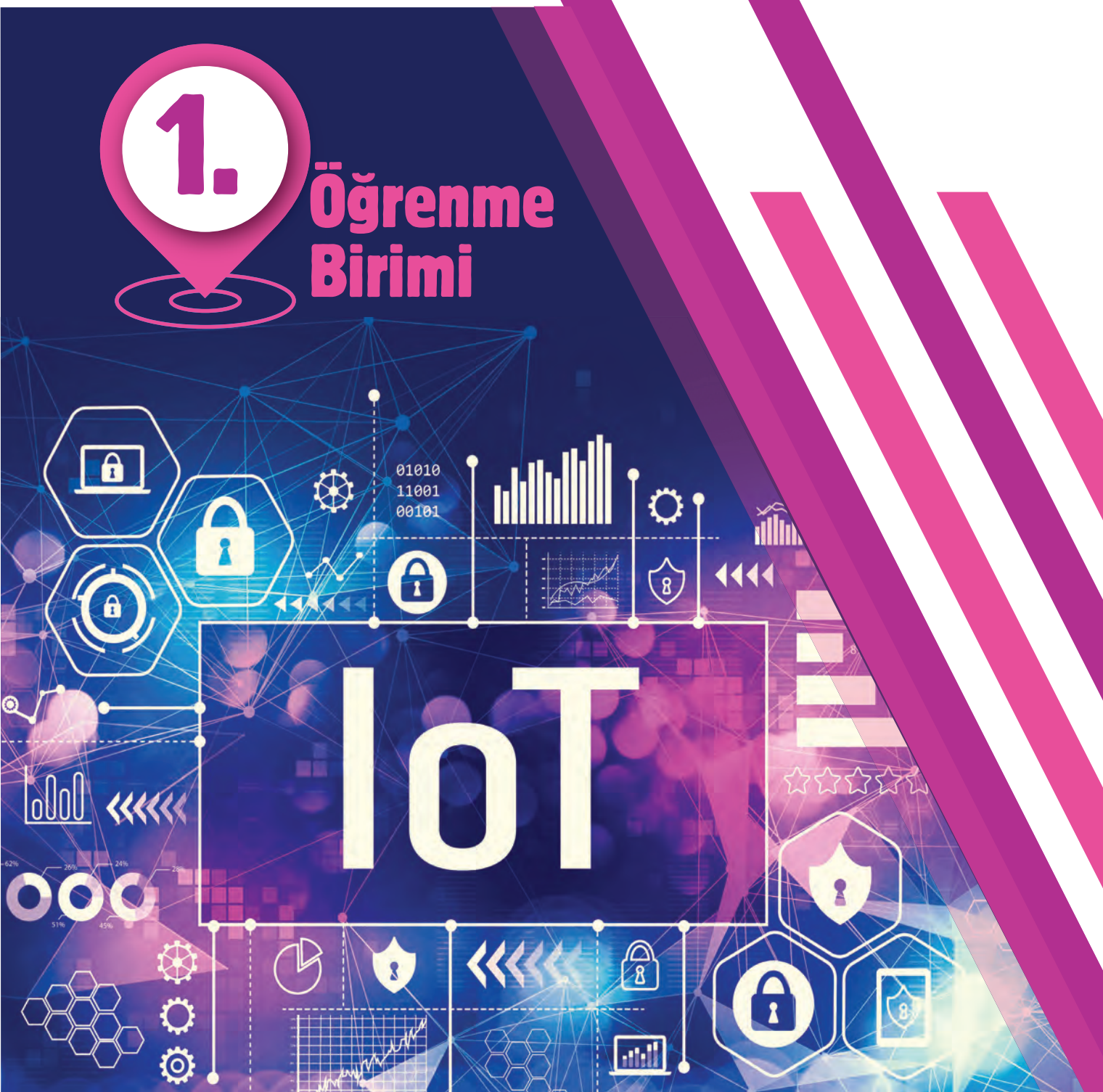
Konu ile ilgili örnek olayları gösterir.

Ölçme ve değerlendirme sorularını gösterir.

*** Bu ders materyalinde ölçü birimlerinin uluslararası kısaltmaları kullanılmıştır.**

NESNELER VE BAĞLANTILAR

1. Öğrenme Birimi



KONULAR

1.1. NESNELERİN İNTERNETİ (INTERNET OF THINGS)
1.2. IoT BİLEŞENLERİ
1.3. İLETİŞİM MODELLERİ
1.4. VERİ GİZLİLİĞİ

NELER ÖĞRENECEKSİNİZ?

- Nesnelerin İnterneti kavramı
- Nesnelerin İnternetinin kullanıldığı alanlar
- Nesnelerin İnternetinin avantaj ve dezavantajları
- Nesnelerin İnterneti bileşenleri
- Nesnelerin İnterneti için kullanılabilecek cihazlar
- Nesnelerin İnternetinde iletişim modelleri
- Nesnelerin İnternetinde veri gizliliğinin önemi

TEMEL KAVRAMLAR

akıllı şehir, aktüatör, bağlantı, bulut bilişim, denetleyici, gizlilik, güvenlik, iletişim modeli, metadata, nesne, nesnelerin interneti (IoT), sensör, veri toplama

HAZIRLIK ÇALIŞMALARI

1. Hangi nesneler internete bağlanır?
2. Akıllı cihazlara örnekler veriniz.
3. Akıllı cihazların insan hayatına olan etkileri nelerdir?
4. Akıllı cihazlar sizce verileri nasıl toplar?

http://kitap.eba.gov.tr/KodSor.php?KOD=14396

1.1. NESNELERİN İNTERNETİ (INTERNET OF THINGS)

Günümüzde akıllı telefon, akıllı TV, akıllı saat gibi birçok cihaz internete bağlı şekilde bulunur. Mobil ağların ve internetin gelişimiyle her geçen gün bu akıllı cihazların sayısı ve çeşidi artmaktadır. Bu artış birbirine bağlı bilgisayarların, birbirlerine bağlı cihaz ve nesnelerle veri alışverişi yapmasını kaçınılmaz hâle getirmiştir. Arabalardan kıyafetlere, dijital saatlerden elektrik ve su sayaçlarına, kahve makinesinden futbol topuna, akıllı evlerden akıllı şehirlere kadar akla gelebilecek her şeyin birbirine bağlanması Nesnelerin İnternetine dönüşmüştür (Görsel 1.1).

Nesnelerin İnterneti (IoT), internete bağlı milyonlarca akıllı cihaz ve sensörün belirli protokollerle iletişime geçtiği bir ağdır. Nesnelerin İnterneti (IoT) kavramı ilk kez 1999 yılında Kevin Ashton tarafından RFID teknolojilerine yönelik bir sunumda ortaya atılmıştır. Nesnelerin İnternetinde akıllı cihazların internete bağlı olmadığı durumlar da bulunabilir. Bu duruma örnek olarak RFID, RTLS ve Beacon vb. teknolojilerin bazı cihazlar ile bilgi üretmeleri verilebilir.

Görsel 1.1: Nesnelerin İnterneti (IoT)

ÖRNEK OLAY

Tanınmış bir şirketin yöneticisi, arabası ile yakındaki bir şehre toplantıya gider. Yolculuk esnasında gidilecek mesafe göz önüne alındığında kalan benzin miktarının yeterli olmayacağını bildiren bir mesaj akıllı telefon ekranında belirir. Bu mesajda yakındaki bir benzin istasyonunun detayları, bir sonraki benzin istasyonunun uzaklığı verilir ve buna göre benzini doldurması tavsiye edilir.

Yukarıdaki örnek olayda verilen bilginin doğru zamanda alınması Nesnelerin İnterneti kavramıyla mümkündür.

SIRA SİZDE

Gün İçinde İnternete Bağlanma Sıklığı ve Şekli

Günümüzün önemli bir kısmı bilgisayar, cep telefonu gibi dijital ortamlarda geçmektedir. Dijital ortamlarda çoğunlukla internet, sosyal ağ, sohbet, görüntülü konuşma, çevrimiçi toplantı, bankacılık, e-ticaret vb. uygulamalar kullanılmaktadır. İnternete hangi sıklıkta ve nasıl bağlandığınızı gösteren tabloyu Görsel 1.2'deki örneğe benzer şekilde elektronik tablo yazılımı ile oluşturunuz.

	A	B	C
1	GÜN İÇİNDE İNTERNETE BAĞLANMA SIKLIĞI VE ŞEKLİ		
2			
3	CİHAZ	YAPILAN İŞ VEYA KULLANILAN UYGULAMA	SÜRE (SAAT)
4	Akıllı telefon	Instagram	1
5		Twitter	1,5
6		Spotify	2,5
7		Telegram / Whatsapp	3,5
8		Görüntülü konuşma	1
9		Bankacılık işlemleri	0,5
10	Dizüstü bilgisayar	Çevrimiçi toplantı	2
11		Youtube	2,5
12	Akıllı TV	İnternetten film izleme	2
13			
14		TOPLAM SÜRE	16,5

Görsel 1.2: İnternete bağlanma sıklığı ve şekli

1.1.1. Nesnelerin İnternetinin Kullanıldığı Alanlar

Her şeyin akıllı hâle geldiği bu dönemde Nesnelerin İnternetinin girmediği bir alan kalmamıştır (Görsel 1.3). Nesnelerin İnternetinin en çok kullanıldığı alanlar aşağıda belirtilmiştir:

- Akıllı ev bina otomasyonu
- Akıllı şehir
- Endüstri
- Sağlık
- Enerji sistemleri
- Ulaşım
- Meteoroloji
- Tarım ve seracılık
- Hayvancılık
- Afet yönetimi
- Çevre
- Medya
- Pazarlama
- Reklamcılık
- Lojistik
- Askeriye ve güvenlik
- Eğitim
- Giyilebilir teknolojiler
- Kamu hizmetleri

Görsel 1.3: IoT kullanım alanları

1.1.2. Nesnelerin İnternetinin Kullanıldığı Ürünler

Nesnelerin İnternetinin kullanıldığı birçok ürün bulunur. Bu ürünler çeşitli endüstri alanlarında, evlerde, hastanelerde, taşıtlarda, sera vb. yerlerde kullanılır.

- **Smart Things:** Akıllı ev sistemleri için kullanılan bir üründür. Ev içindeki elektronik aletlerin, ısıtma ve aydınlatma sistemlerinin tek bir uygulamayla otomatikleşmesini ve yönetilmesini sağlar.
- **OttoLock Akıllı Kilit:** Ev ve iş yerlerine anahtarsız otomatik giriş-çıkış yapmak amacıyla kullanılan bir üründür. Cep telefonu üzerinden ev ve iş yeri kapılarını kilitleme, ev ve iş yerinden uzaktayken gelen misafirlere eve ve iş yerine giriş izni tanımlama, eve ve iş yerine kimin giriş-çıkış yaptığını cep telefonundan takip etme imkânı sunar.
- **HAPIfork:** Sağlığını iyileştirmek, kilo vermek veya yeme hızını yavaşlatmak isteyen kişiler için tasarlanmış bir akıllı çataldır. Yemek yeme davranışını değiştirmeye yardımcı olur.
- **Solar Curtain:** Güneş enerjisi ile elektrik üretilmesini sağlayan bir perdedir. Mobil uygulama ile perdeye uzaktan erişim sağlanabilir. Hatta üretilen elektriğin anlık, haftalık ve yıllık olarak hazırlanan raporu takip edilebilir.
- **MiCoach Futbol Topu:** Vurulan topun hızı, ivmesi, vuruş şiddeti, gidiş-dönüş yörüngesi ve topun aldığı falso verisi mobil uygulama ile anlık takip edilir.
- **iBeat Heart Watch:** Kalp krizini fark ederek hayat kurtarmayı hedefleyen bir akıllı saattir. Kişinin kalp ritmindeki düzensizliklerini, kan dolaşımını sürekli kontrol eder. Anormal bir durum oluştuğunda ise acil servisler ve tanıdıklar dâhil herkese otomatik olarak haber verir.
- **Dropcam:** Ev ve ofislere kurulan kameraların mobil cihaz veya bilgisayar üzerinden izlenmesine imkân sunar.
- **Family Hub Buzdolabı:** Alışveriş listesi hazırlama ve sipariş verme özelliği bulunur. Ayrıca markette iken mobil uygulama üzerinden buzdolabı içindeki eksik olan ürünler kontrol edilerek ihtiyaca göre alışveriş yapılabilir.
- **Edyn:** Bahçe ve tarlalarda toprağa hangi ürünün nasıl ekilmesi ve toprağın hangi aralıklarla sulanması gerektiği konusunda önerilerde bulunur.
- **Akıllı Tişört:** Kişinin biyolojik ve fizyolojik bilgilerini okuyarak performansı hakkında bilgi sahibi olmasını sağlar. Maç anında oyuncunun kalp atışı, stres düzeyi, nefes alışı ve verişi gibi verilerini toplar.

1.1.3. Nesnelerin İnternetinin Avantaj ve Dezavantajları

Nesnelerin İnternetinin avantajları şunlardır:

- Bireylerin yaşam kalitesini artırır.
- Zaman ve para tasarrufu sağlar.
- Daha fazla bilginin kolay bir şekilde toplanmasını sağlar.
- Daha iyi kararlar almaya yardımcı olur.
- Endüstride üretimin daha verimli olmasını sağlar.
- Daha fazla gelir elde edilmesini sağlar.
- İş süreçlerini izlemeyi kolaylaştırır.
- Daha fazla müşteri memnuniyeti sağlar.

Nesnelerin İnternetinin dezavantajları şunlardır:

- Veriler iletilirken gizlilik riski artar.
- Güvenlik zafiyeti riski artar.
- Çok karmaşık sistemlerde başarısızlık olasılıkları artar.
- Kişisel mahremiyeti riske atar.
- Depolanması gereken veri miktarı artar.

1.2. IoT BİLEŞENLERİ

Farklı türlerde birçok IoT cihazı bulunur. Bu cihazların çoğu işlevlerini gerçekleştirmek için kontrolör (denetleyici), sensör ve aktüatör (eyleyici) bileşenlerini kullanır. IoT cihazında bulunan sensörler aracılığı ile ortamdaki olaylar, değişiklikler ve fiziksel büyüklük verileri algılanıp gerçek zamanlı bir şekilde denetleyiciye gönderilir. Denetleyici bu verileri analiz eder ve işler. Veriler denetleyicinin işleyemeyeceği büyüklükte ise işlenmesi için buluta gönderilebilir. Denetleyici, veriler işlendikten sonra eylemleri gerçekleştirmek için aktüatörleri kullanır. Bu süreç sürekli tekrarlanır (Görsel 1.4).

Görsel 1.4: IoT Bileşenleri

1.2.1. Sensörler

Sensör, ortamdaki bir tür bilgiyi tespit ederek çevresel bir özelliği ölçmek için kullanılan ve nicel veriler üreten bir cihazdır. Tespit edilen bilgi; ışık, nem, mesafe, hareket, basınç, hız, sıcaklık, eğim veya başka bir çevresel özellik olabilir. Sensörler sistemin duyu organları gibidir. Algılanması gereken farklı çevresel özellikler, farklı tiplerde sensörler gerektirir. Sensörler, analog ve dijital olmak üzere iki tipte bulunur. Sensörler, denetleyiciye kablolu veya kablosuz şekilde bağlanır. Sensörler, verileri analiz yapılıp karar verilmesi için denetleyiciye gönderir.

1.2.2. Aktüatörler (Eyleyiciler)

Aktüatör bir sistemi veya mekanizmayı kontrol etme, hareket ettirme amacıyla kullanılan bir tür motordur. Bir enerji kaynağı tarafından çalıştırılır. Bu enerji kaynağı; elektrik, hidrolik (sıvı basıncı), pnömatik (hava basıncı) veya termal (sıcaklık) olabilir. Diğer bir deyişle aktüatörler elektrik sinyalinin fiziksel çıkışa dönüştürülmesinden sorumludur. Bu fiziksel çıktı sayesinde bir kullanıcıya LED'ler ile bilgi verme, buzzer ile sesli uyarı verme veya kullanıcının bulunduğu ortamın sıcaklığını değiştirme işlemleri yapılabilir.

1.2.3. Kontrolörler (Denetleyiciler)

Kontrolörler, sensörlerden gelen verileri toplamaktan ve ağ bağlantısı sağlamaktan sorumludur. Kontrolörler, sensörlerden topladıkları verilerle anında karar verebilir veya verileri analiz için daha güçlü bir bilgisayara gönderebilir. Bu güçlü bilgisayar, kontrolör ile aynı yerel alan ağında veya uzak bir veri merkezinde olabilir.

IoT projelerinde genellikle Arduino, NodeMCU, ESP32, Raspberry Pi gibi denetleyiciler birlikte kullanılır. Örneğin sensörlerden gelen veriler Arduino ile toplanabilir, ardından bu veriler Raspberry Pi ile işlenip görselleştirilebilir.

1.2.4. Bağlantı

Nesnelerin İnternetinde bağlantı genel olarak üç türde incelenir. Bağlantı türleri şunlardır:

- **Güç Bağlantıları:** IoT cihazları bir güç kaynağına bağlanmalıdır. Bu güç bağlantısı pil, AC kaynak, haricî DC kaynak, PoE (Ethernet üzerinden) veya yenilenebilir enerji kaynaklarından sağlanır.

- **Devre Bağlantıları:** Cihaz içindeki sensör ve aktüatörler denetleyicilere devre kartları ve teller ile bağlanır.
- **Ağ Bağlantıları:** IoT cihazları birbirlerine bağlanarak küçük ağlar oluşturur. Bu ağlar bakır, fiber optik kablo ve kablosuz medya ortamları ile kurulur. Medya türüne göre kullanılan protokol ile iletişim gerçekleşir.

1. UYGULAMA

Gül Hanım bir teknoloji mağazasında gezerken Smart Home reyonunda akıllı lamba görmüştür. Bu akıllı lamba uzaktan kontrol edilebilen bir IoT nesnesidir. Akıllı lamba ile ilgili detaylı bilgileri mağaza görevlisinden öğrenen Gül Hanım, cihazı satın almıştır. Akıllı lamba IoT nesnesinin kullanım kılavuzunu inceleyen Gül Hanım'ın gerekli adımları uygulayarak Görsel 1.5'teki IoT ağ kurulumunu gerçekleştirmesine yardımcı olunuz.

Kurulumu tamamladıktan sonra ping ve tracert araçlarını kullanarak IoT ağı test edilir. Dizüstü bilgisayar ile IoT nesnesinin kontrolü gerçekleştirilir.

Görsel 1.5: Basit bir IoT ağ örneği

NOT

- Tracert aracı **tracert** ***<hedef uç cihaz adresi>***,
- Ping aracı **ping** ***<hedef uç cihaz adresi>*** şeklinde kullanılır.

1. Adım : Network Devices (1) bölümünden Wireless Devices (2) türünü seçiniz. Home Gateway (3) cihazını **A** bölgesine yerleştiriniz (Görsel 1.6).

Görsel 1.6: Home Gateway cihazının yerleşimi

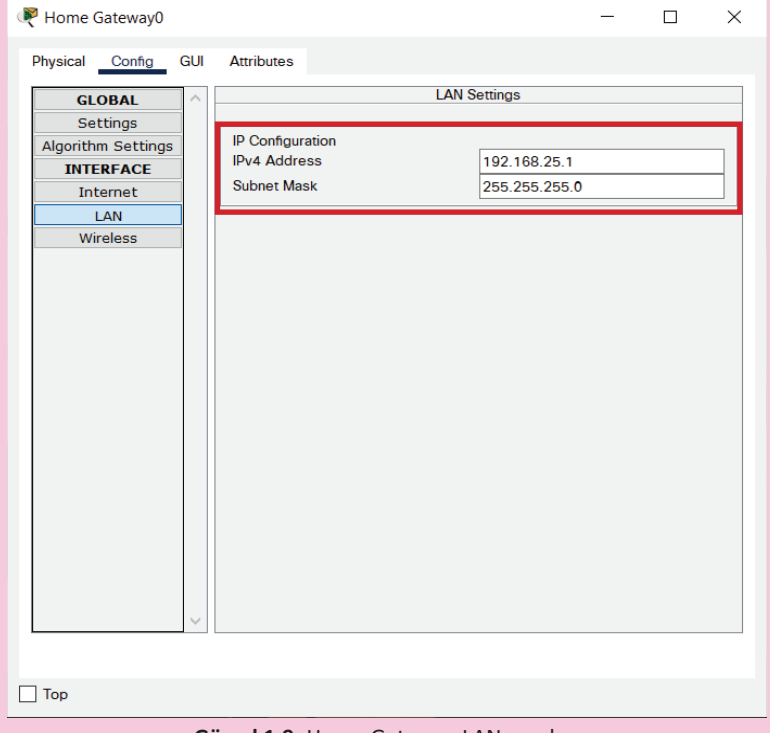
2. Adım : End Devices (1) bölümünden End Devices (2) türünü seçiniz. Dizüstü (Laptop) (3) uç nokta cihazını **B** bölgesine yerleştiriniz (Görsel 1.7).

Görsel 1.7: Dizüstü (Laptop) uç nokta cihazının yerleşimi

3. Adım : End Devices (1) bölümünden Home (2) türünü seçiniz. Light (3) IoT cihazını **C** bölgesine yerleştiriniz (Görsel 1.8).

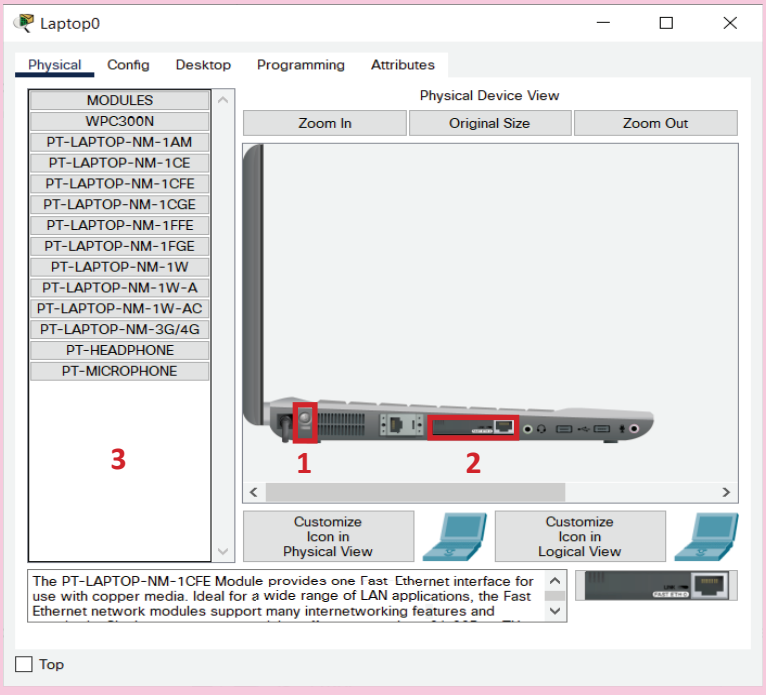
Görsel 1.8: Light IoT cihazının yerleşimi

4. Adım : Home Gateway cihazına tıklayınız. Gelen ekranın Config sekmesinden **INTERFACE** LAN ayarlarını yapınız (Görsel 1.9). Ayarları yaptıktan sonra formu kapatınız.



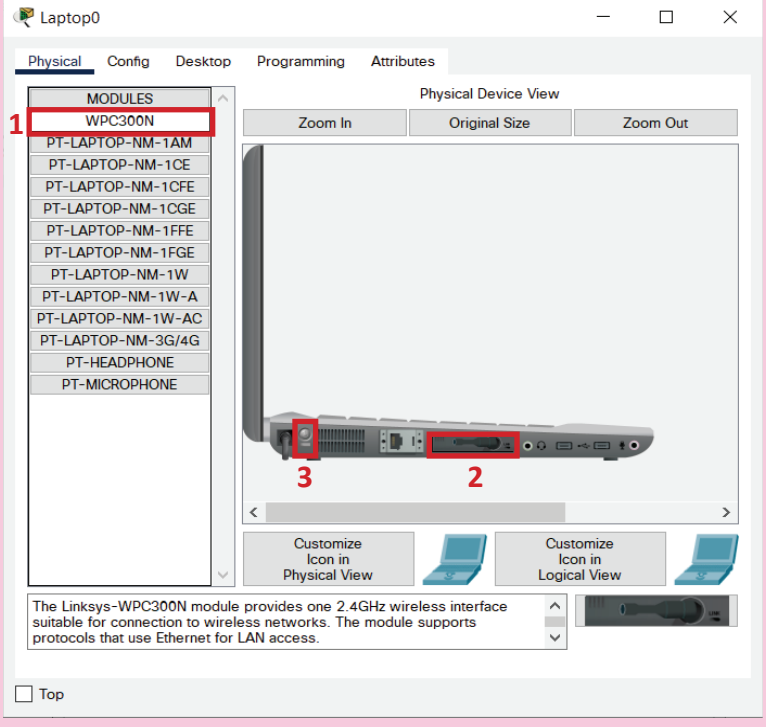
Görsel 1.9: Home Gateway LAN ayarları

5. Adım : Dizüstü (Laptop) cihazına tıklayınız. Gelen ekranın Physical sekmesinde Dizüstü (Laptop) gücünü kesmek için açma kapama (**1**) düğmesine tıklayınız. Ethernet modülünü (**2**) MODULES kısmına (**3**) sürükleyip bırakınız (Görsel 1.10).



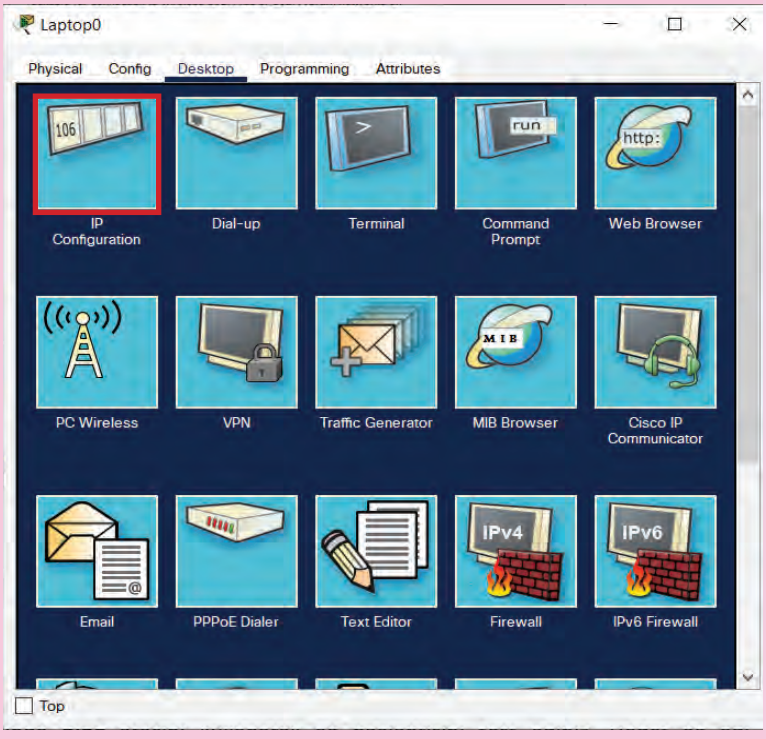
Görsel 1.10: Dizüstü (Laptop) uç nokta cihazının Ethernet modülünü çıkarma

6. Adım : Wireless modülünü (**1**) ethernet modülünden boşalan yuvaya (**2**) sürükleyip bırakınız. Dizüstü (Laptop) gücünü vermek için açma kapama (**3**) düğmesine tıklayınız (Görsel 1.11).



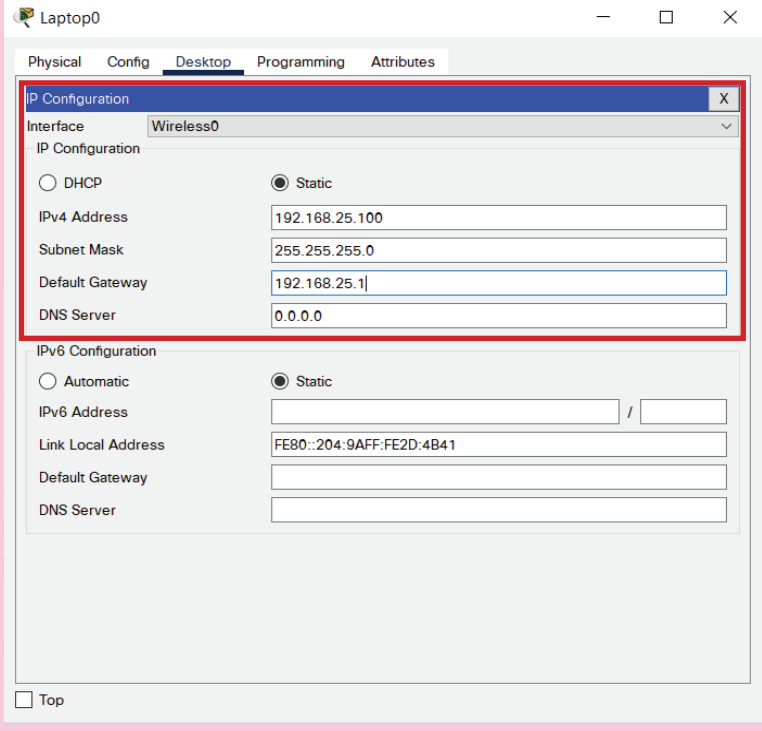
Görsel 1.11: Dizüstü (Laptop) uç nokta cihazına Wireless modülünü yerleştirme

7. Adım : Desktop sekmesinde **IP Configuration** kutucuğuna tıklayınız (Görsel 1.12).



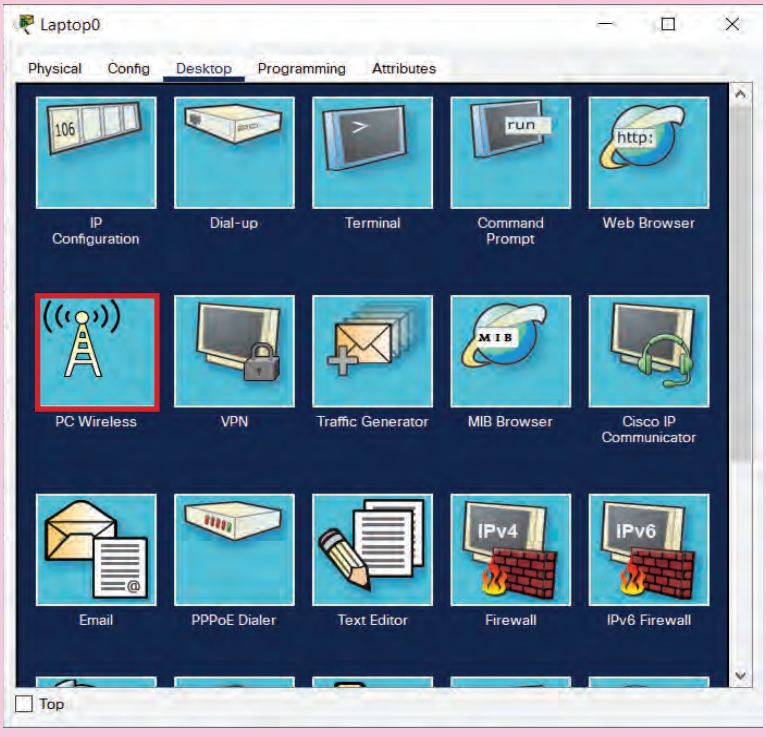
Görsel 1.12: Dizüstü (Laptop) uç nokta cihazının IP konfigürasyon kutucuğu

8. Adım : Açılan **IP Configuration** penceresinde ilgili ayarları yapınız. Ayarları yaptıktan sonra pencereyi kapatınız (Görsel 1.13).



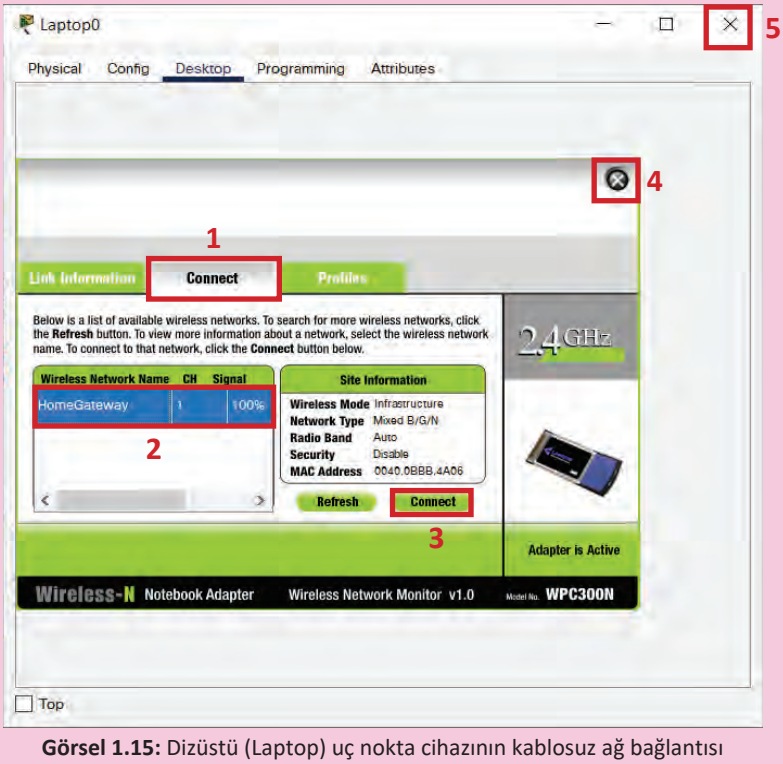
Görsel 1.13: Dizüstü (Laptop) uç nokta cihazının IP konfigürasyon ayarları

9. Adım : Desktop sekmesinde PC Wireless kutucuğuna tıklayınız (Görsel 1.14).



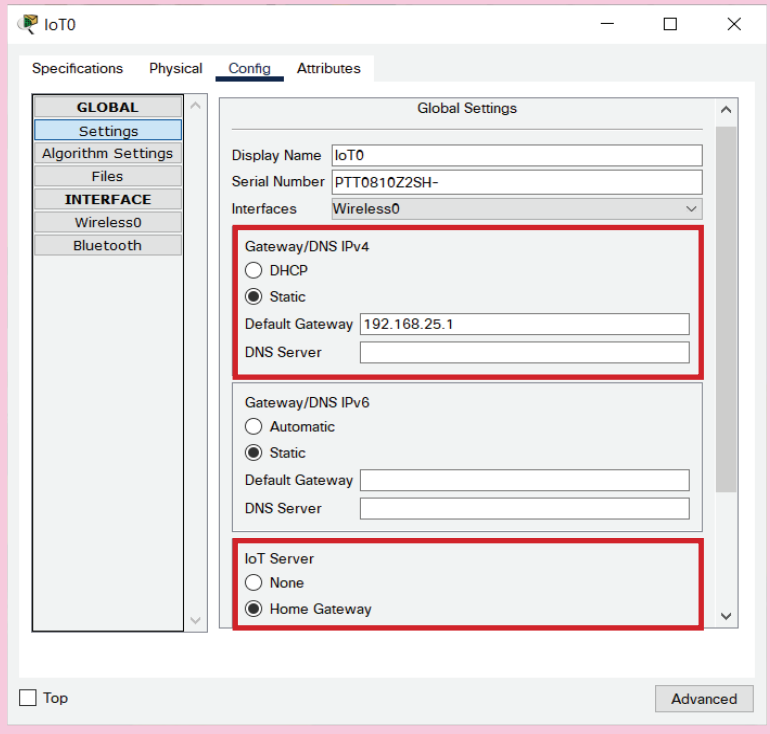
Görsel 1.14: Dizüstü (Laptop) uç nokta cihazının PC Wireless kutucuğu

10. Adım : Açılan pencerede Connect (1) sekmesine tıklayınız. HomeGateway (2) kablosuz ağ adını seçiniz. Connect (3) butonuna tıklayınız. Ayarları yaptıktan sonra pencereyi (4) kapatınız. Dizüstü (Laptop) formunu (5) kapatınız (Görsel 1.15).



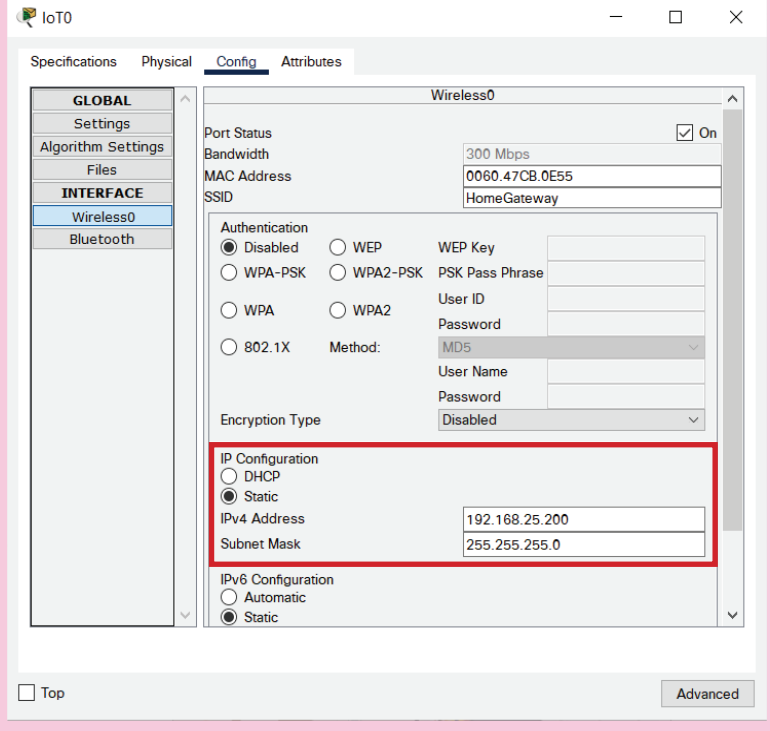
Görsel 1.15: Dizüstü (Laptop) uç nokta cihazının kablosuz ağ bağlantısı

11. Adım : Light IoT cihazına tıklayınız. Gelen ekranın Config sekmesinden **Global Settings** ayarlarını yapınız (Görsel 1.16).



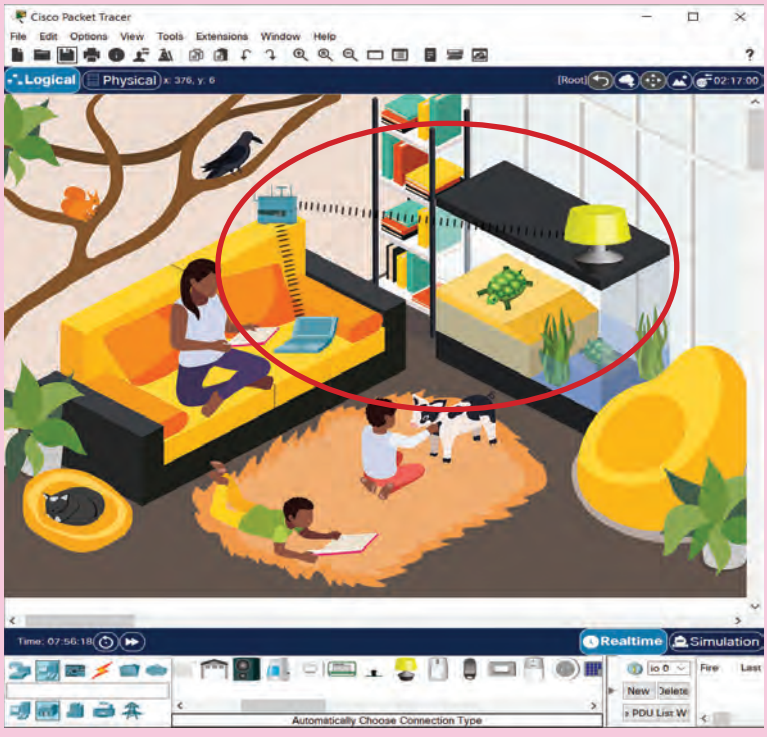
Görsel 1.16: Light IoT cihazının Global Settings ayarları

12. Adım : Config sekmesinden **INTERFACE** Wireless0 ayarlarını yapınız (Görsel 1.17). Ayarları yaptıktan sonra formu kapatınız.



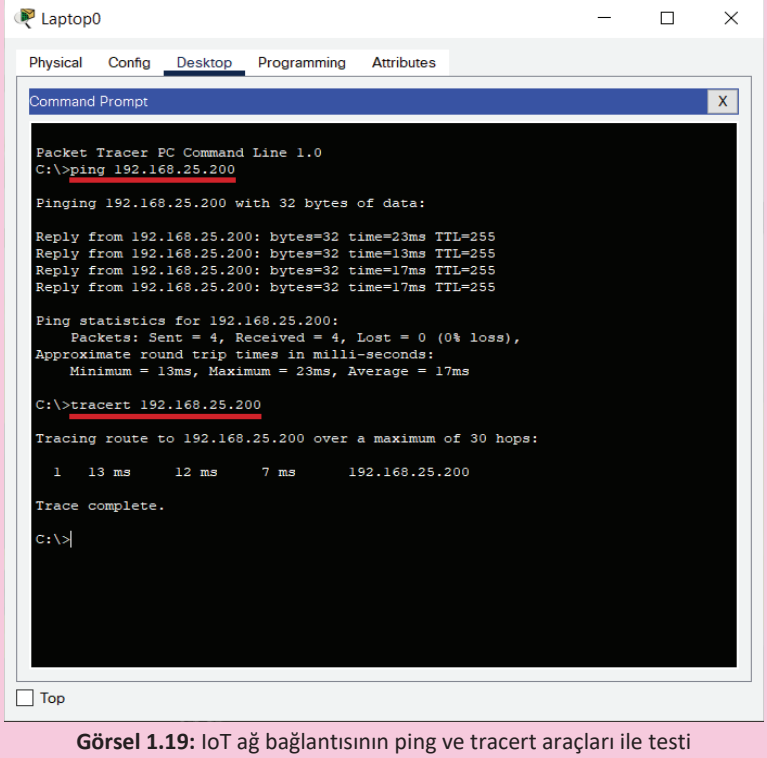
Görsel 1.17: Light IoT cihazının Wireless IP konfigürasyon ayarları

13. Adım : IoT ağındaki kablosuz bağlantıları kontrol ediniz (Görsel 1.18).



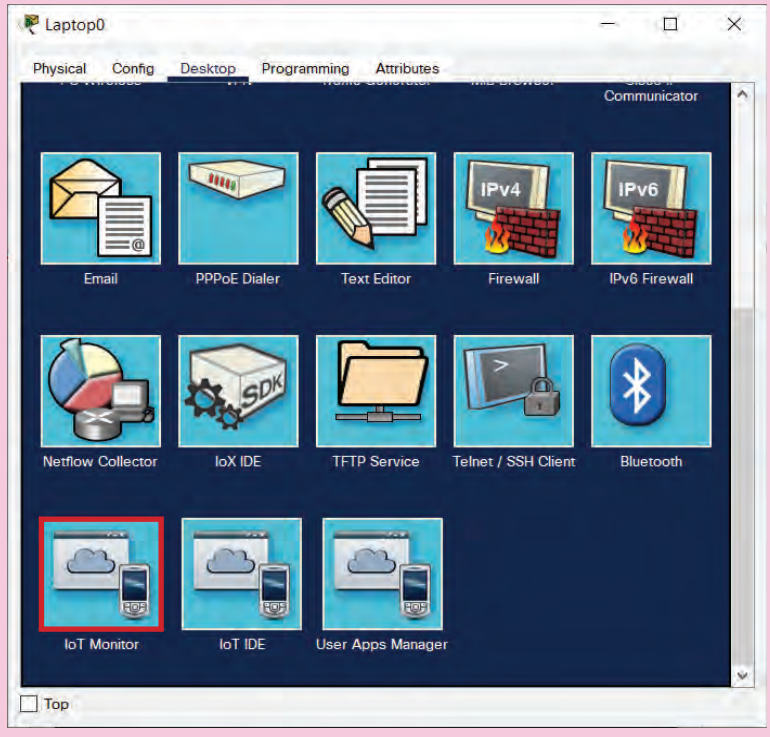
Görsel 1.18: IoT kablosuz ağ bağlantısı

14. Adım : Dizüstü (Laptop) cihazına tıklayınız. **Desktop** sekmesinde Command Prompt kutucuğuna tıklayınız. Komut isteminde **ping** ve **tracert** aracını kullanarak ağ bağlantısını test ediniz. Test işleminden sonra pencereyi kapatınız (Görsel 1.19).



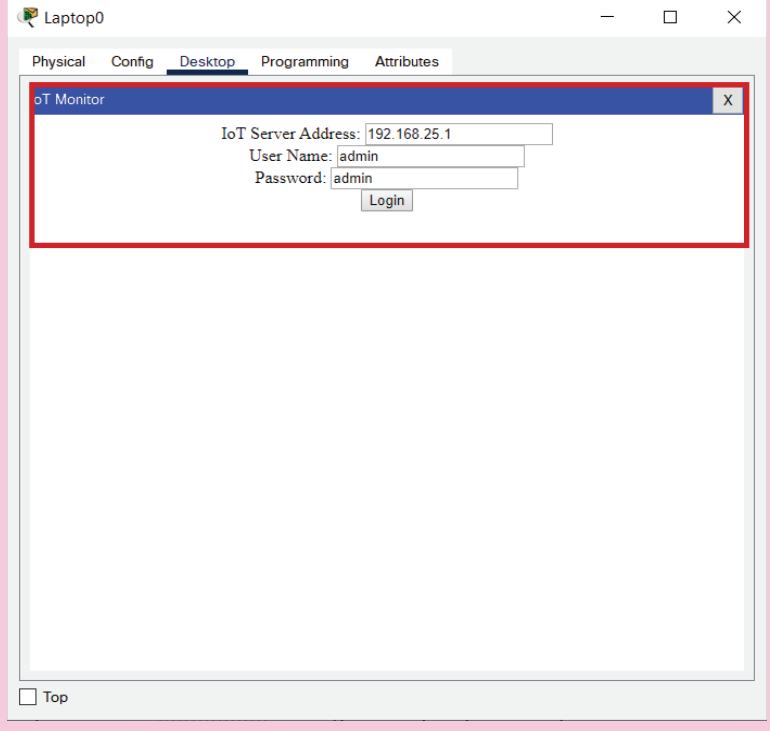
Görsel 1.19: IoT ağ bağlantısının ping ve tracert araçları ile testi

15. Adım : Desktop sekmesinde **IoT Monitor** kutucuğuna tıklayınız (Görsel 1.20).



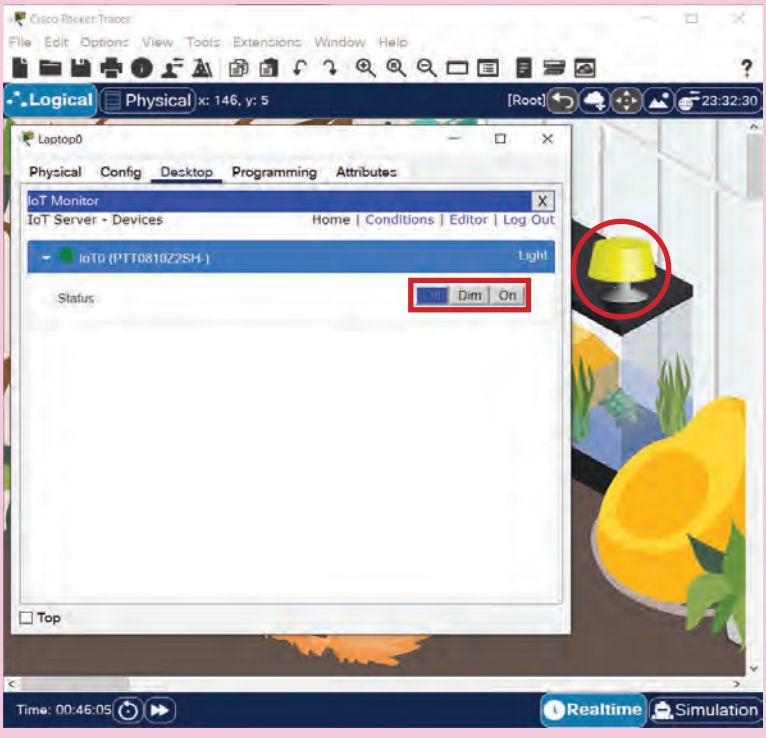
Görsel 1.20: Dizüstü (Laptop) uç nokta cihazının IoT Monitor kutucuğu

16. Adım : Açılan pencerede IoT Sunucu (Home Gateway) IP adresini, kullanıcı adını ve parolasını giriniz. Login butonuna tıklayınız (Görsel 1.21).

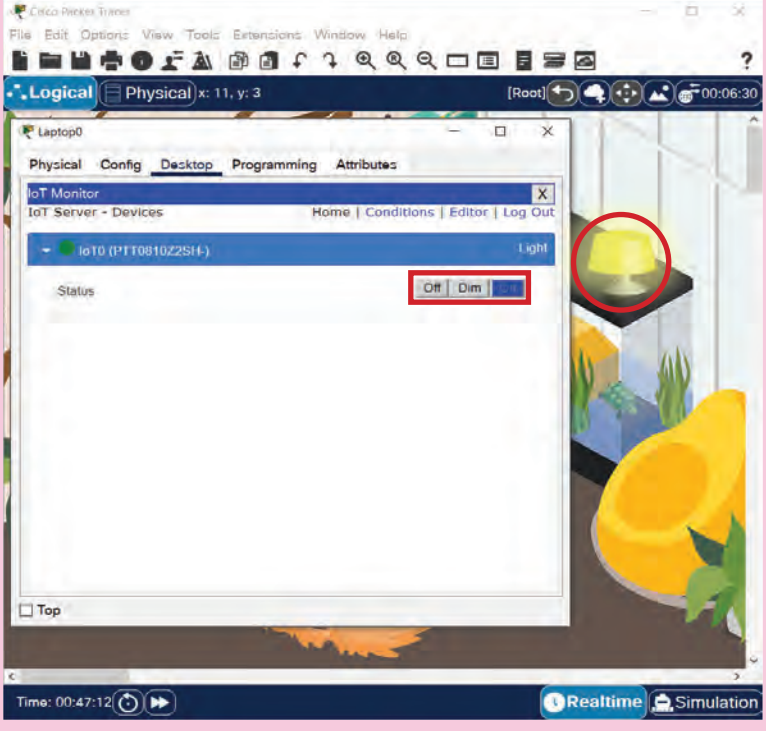


Görsel 1.21: IoT Monitor login işlemi

17. Adım : IoT Monitor penceresinden Light IoT nesnesini On-Off butonlarıyla kontrol ediniz (Görsel 1.22, Görsel 1.23).



Görsel 1.22: Light IoT nesnesinin kontrolü ve Off durumu



Görsel 1.23: Light IoT nesnesinin kontrolü ve On durumu

SIRA SİZDE

IoT cihazlarına hayranlık duymaya başlayan Gül Hanım teknoloji mağazasından akıllı kahve makinesi sipariş vermiştir. Gül Hanım'ın akıllı kahve makinesini IoT ağına dâhil etmesi konusunda yardıma ihtiyacı vardır. Gül Hanım'a gerekli işlem adımlarını göstererek kahve makinesini uzaktan çalıştırmasını sağlayacak işlem basamaklarını defterinize yazınız ve daha sonra öğretmeniniz ile cevaplarınızı kontrol ediniz (Görsel 1.24).

Görsel 1.24: Akıllı kahve makinesinin uzaktan kontrolü

1.2.5. Analiz (Veri İşleme)

Veri işleme, sensörler tarafından toplanan ham verilerin anlamlı ve kullanılabilir bilgiye dönüşmesidir. Veri işleme yerelde veya bulutta gerçekleşebilir. Analiz, IoT ekosistemi içindeki en önemli bileşendir.

1.2.6. Kullanıcı Arayüzü

Kullanıcı arayüzü, IoT sisteminin ve sistemde bulunan nesnelerin kontrol edilebileceği ve yönetilebileceği bir platformdur. Arayüz olarak mobil uygulama, web sitesi, masaüstü uygulama veya buton, LCD ekran devresi gibi donanımsal bir tasarım kullanılabilir.

1.2.7. Bulut Bilişim

Bulut bilişim; verileri depolamak, verileri analiz etmek, çevrimiçi uygulamalara erişim sağlamak, kişisel ve kurumsal kullanım için yedekleme hizmetleri sağlamak amacıyla kullanılır. IoT cihaz sayısının ve çeşidinin artmasıyla birlikte işlenmesi gereken veri miktarı da artmıştır. IoT sistemlerinin büyüyüp daha karmaşık hâle gelmesiyle birlikte verilerin denetleyicilerle ve kişisel bilgisayarlarla işlenmesi daha da zorlaşmıştır. Bu nedenle her geçen gün artan verinin bulut bilişim ile işlenmesi kaçınılmaz hâle gelmiştir.

SIRA SİZDE

Görsel 1.25: Akıllı tarımda IoT bileşenleri

Tarım sektöründe çalışan Ayşe Hanım üretim, yönetim ve kontrolü otomatik hâle getirerek iş yükünü azaltmak istemektedir. Bu nedenle internette çeşitli araştırmalar yapar. Araştırmalardan edindiği bilgilere göre tarla ve serasında Nesnelerin İnternetini uygulamaya karar verir. Ardından tarla ve serasına IoT için gerekli bileşenleri kurar (Görsel 1.25).

Ayşe Hanım'ın kurduğu akıllı tarım sisteminde yer alan IoT bileşenlerini tespit ediniz. Tespit ettiğiniz bileşenlerin akıllı tarımda hangi amaçlarla kullanıldığını aşağıda nokta ile gösterilen satırlara yazınız.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

1.3. İLETİŞİM MODELLERİ

IoT ekosisteminde çok fazla sayıda cihaz çeşidi bulunur. Heterojen bir yapısı olan bu ekosistem için en önemli zorluk, IoT cihazlarının internete ve birbirlerine güvenli bir şekilde bağlanabilmesini sağlamaktır. Bu amaçla birçok teknoloji ve protokol geliştirilmiştir.

IoT sistemleri arasında etkili iletişim sağlamak için tutarlı, güvenli ve yaygın olarak tanınan teknolojilerin ve standartların kullanılması gerekir. Bu nedenle çeşitli kuruluşlar tarafından referans modeller geliştirilmiştir. Bu referans modeller şunlardır:

- Katmanlı ağ modelleri
- Bağlantı seviyelerine dayalı model
- İletişim türlerine dayalı modeller
- Üç katmanlı IoT mimari modeli

1.3.1. Katmanlı Ağ Modelleri

Katmanlı ağ modelleri bir ağın nasıl çalıştığını göstermek için kullanılır. Katmanlı ağ modeli kullanmanın faydaları şunlardır:

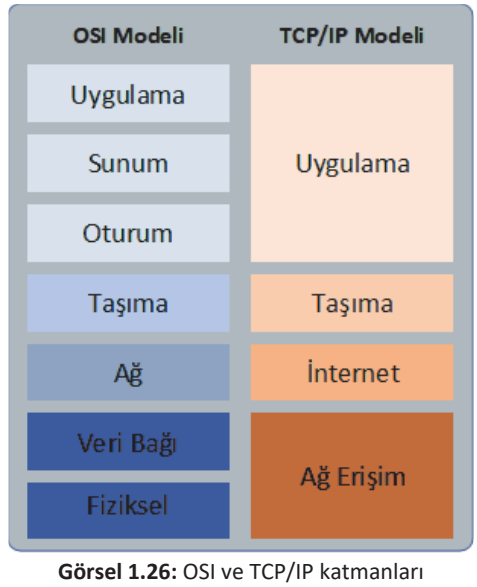
- Protokol tasarımına yardımcı olur.
- Farklı satıcıların ürünleri birlikte uyumlu çalışabildiği için rekabeti teşvik eder.
- Bir katmandaki teknoloji değişikliklerinin diğer katmanları etkilemesini önler.
- Ağ işlevlerini ve yeteneklerini tanımlamak için ortak bir dil sağlar.
- Ağ işlemlerini yönetilebilir küçük parçalara bölerek daha az karmaşıklık sağlar.
- Kullanıcıya yazılımsal veya donanımsal yenilik ve güncellemelerde esneklik kazandırır.
- Daha hızlı ürün geliştirme sağlar.
- Modüler çalışmaya izin verir.

1.3.1.1. OSI ve TCP/IP Modelleri

OSI ve TCP/IP modelleri ağ bağlantılarını tanımlamak için kullanılır. Her iki model de katmanlardan oluşur. Her katmanın bir işlevi vardır. OSI ve TCP/IP modelleri sıklıkla birbirlerinin yerine kullanılır (Görsel 1.26).

OSI modeli, her katmanda oluşabilecek kapsamlı bir işlev ve hizmet listesi sağlar. İki bilgisayar arasındaki iletişimin nasıl gerçekleşeceğini tanımlar.

TCP/IP, bilgisayarlar arası veri iletişiminin kurallarını koyan bir iletişim protokolleri bütünüdür. TCP/IP modeli genellikle internet modeli olarak adlandırılır. TCP/IP modeli daha çok uygulamaya yöneliktir.



Görsel 1.26: OSI ve TCP/IP katmanları

1.3.1.2. IoT Referans Modeli

IoT Referans Modeli yedi seviyeden oluşur. Bu model; ortak terminoloji sağlamak, IoT uygulamalarında bilginin nasıl aktığını ve işlendiğini tanımlamak, IoT çalışmalarına rehberlik etmek ve hız kazandırmak amacıyla geliştirilmiştir (Görsel 1.27).

Seviye	Açıklama
1. Fiziksel Cihazlar ve Denetleyiciler	• IoT mimarisi tarafından yönetilen, bilgi gönderip alan çok çeşitli uç nokta cihazları ve sensörleri içerir.
2. Bağlanabilirlik	• Cihazlar ve ağ arasında, ağlar arasında ve 3. seviyedeki ağ ile veri işleme arasında zamanında güvenilir veri iletimi yapar.
3. Sis Bilişim	• Veriler, depolama ve daha üst düzey işlemler için uygun bilgilere dönüştürülür.
4. Veri Toplama	• Hareket halindeki veri, üst seviyelerde kullanılmak amacıyla bekleyen durumdaki veriye dönüştürülür.
5. Veri Soyutlama	• Verileri ve depolamayı uygulamayı geliştirecek şekilde oluşturur.
6. Uygulama	• Cihaz verilerinin niteliğine ve iş gereksinimlerine dayalı bilgileri yorumlama ve raporlama imkânı verir.
7. İş Birliği ve Süreçler	• İnsanlar ve iş süreci arasında gereken iletişimi ve iş birliğini sağlamak için birçok uygulama sunar.

Görsel 1.27: IoT Referans Modeli

1.3.2. Bağlantı Seviyelerine Dayalı Model

Bu modelde akıllı nesneler arasındaki bağlantıların seviyesine dayalı basit bir yaklaşım benimsenmiştir.

Akıllı nesneler arasındaki bağlantı seviyeleri şunlardır:

- Cihazdan-Cihaza
- Cihazdan-Buluta
- Cihazdan-Ağ Geçidine-Buluta
- Cihazdan-Ağ Geçidine-Buluta-Uygulamaya

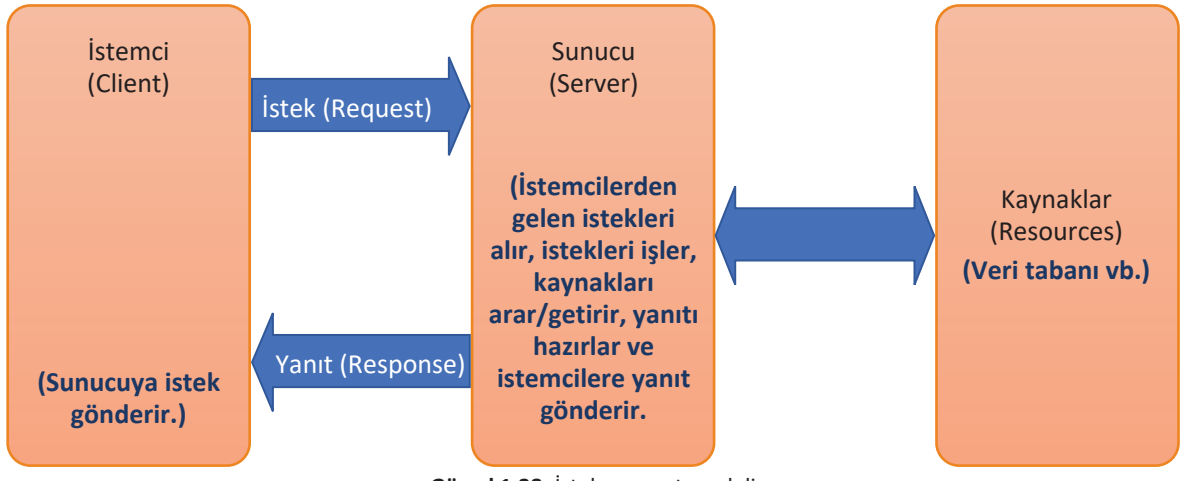
1.3.3. İletişim Türlerine Dayalı Modeller

İletişim türlerine dayalı modeller dörde ayrılır.

- İstek ve yanıt modeli
- Yayın ve abone modeli
- İtme ve çekme modeli
- Özel çift modeli

1.3.3.1. İstek ve Yanıt Modeli (Request-Response Model)

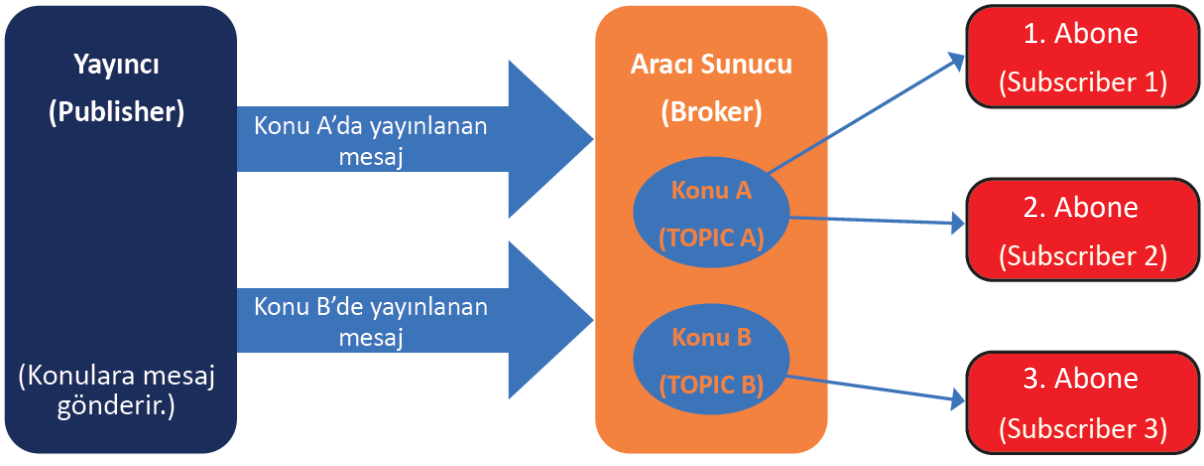
İstek ve yanıt modeli, istemci ve sunucu mimarisinden oluşur. İstek ve yanıt modeli durum bilgisizdir. Bu nedenle her istek bağımsız olarak işlenir. Bu modelde IoT cihazı istemci rolündedir. İstemci, sunucuya bir istek gönderir. Gönderilen bu istek veri aktarımı veya veri yükleme talebi olabilir. Sunucu, uzak veya yerel olabilir ve birden çok istemcinin isteklerini karşılayabilir. Sunucu isteği aldığında nasıl yanıt vereceğine karar verir, yanıtı hazırlar ve istemciye gönderir (Görsel 1.28).



Görsel 1.28: İstek ve yanıt modeli

1.3.3.2. Yayın ve Abone Modeli (Publish-Subscribe Model)

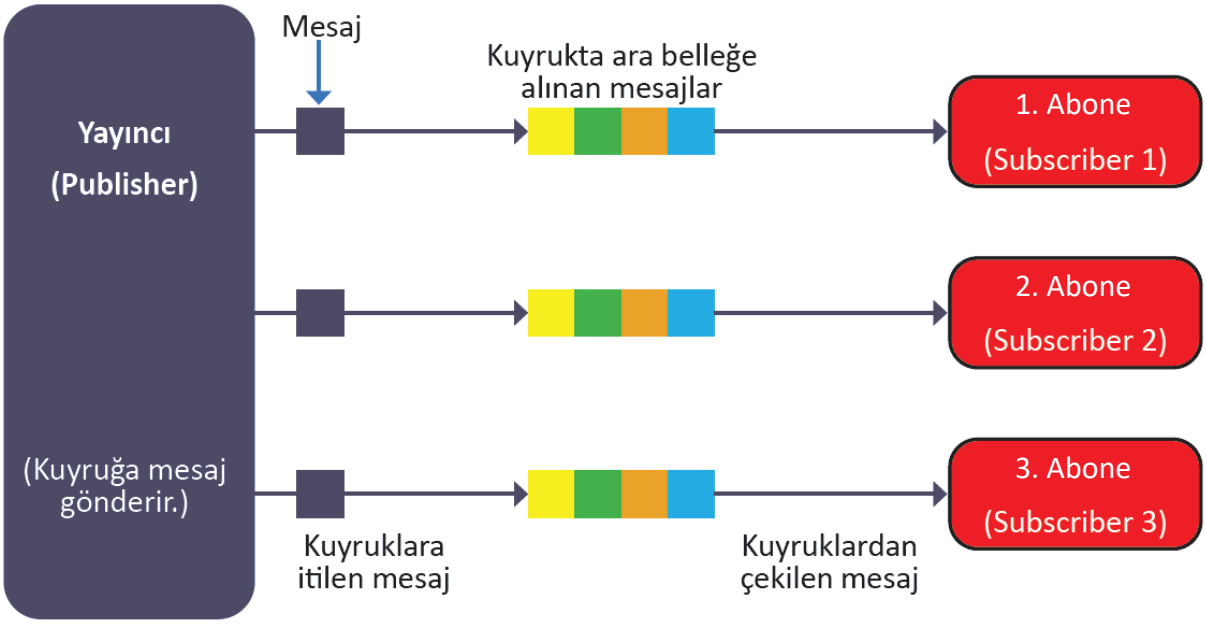
Yayın ve abone modeli; yayıncı (publisher), aracı sunucu (broker) ve abone (subscriber) olmak üzere üç yapıdan oluşur. Yayıncılar, verileri aracı sunucu tarafından yönetilen konuya gönderir. Aracı sunucu da ilgili konuya gelen verileri abonelere iletir. Aracı sunucuların görevi, yayıncılardan gelen verileri kabul etmek ve bunları uygun abonelere göndermektir (Görsel 1.29).



Görsel 1.29: Yayın ve abone modeli

1.3.3.3. İtme ve Çekme Modeli (Push-Pull Model)

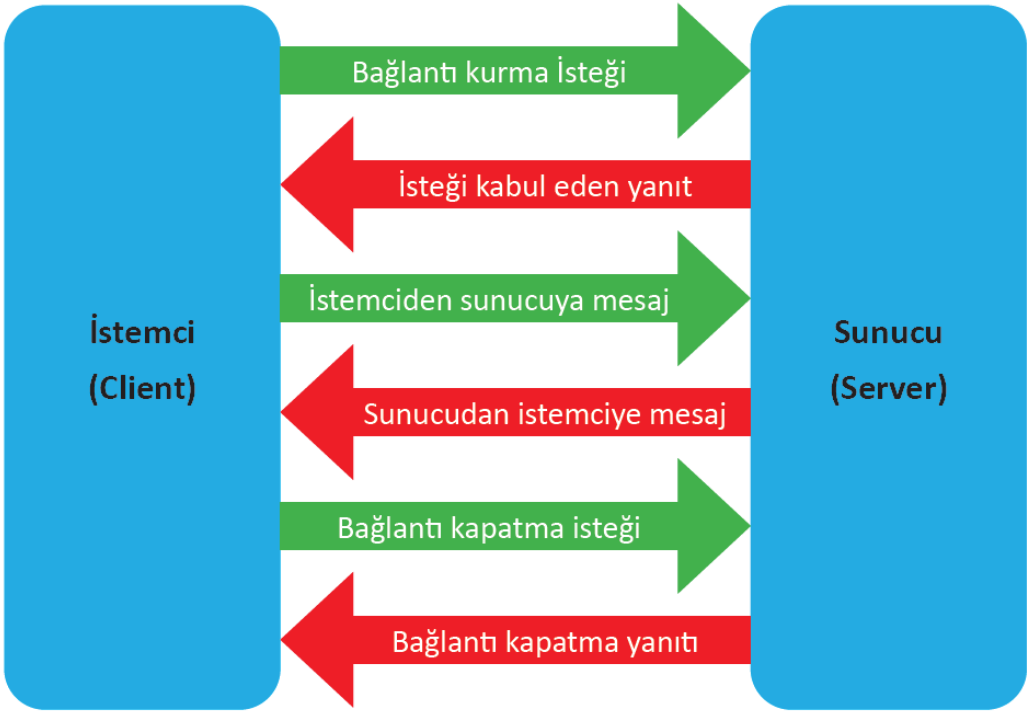
Bu model; veri yayıncıları, veri aboneleri ve veri kuyruklarından oluşur. Yayıncılar mesajı / verileri yayınlar ve sıraya iter. Diğer tarafta bulunan aboneler, verileri kuyruktan çeker. Kuyruk, veri itme veya çekme hızında fark oluştuğunda mesaj için ara bellek görevi görür. Yayıncılar, abonelerin indirebileceğinden daha hızlı bir oranda veri ürettiğinde kullanışlıdır (Görsel 1.30).



Görsel 1.30: İtme ve çekme modeli

1.3.3.4. Özel Çift Modeli (Exclusive Pair Model)

Bu model, istemci ve sunucu arasında çift yönlü bir iletişim sağlar. İstemci ve sunucu, bağlantı kurulduktan sonra birbirlerine mesaj gönderebilir. İstemci, bağlantıyı kapatmak için bir istek gönderene kadar bağlantı açık kalır (Görsel 1.31).



Görsel 1.31: Özel çift modeli

1.3.4. Üç Katmanlı IoT Mimari Modeli

Üç katmanlı IoT mimarisi temel model kabul edilir. Bu mimari modelde sırasıyla donanım katmanı, iletişim

katmanı ve uygulama katmanı bulunur (Görsel 1.32).

Görsel 1.32: Üç katmanlı IoT mimari modeli

Birinci katmanda sensörler ve aktüatörler gibi fiziksel donanımlar bulunur. Veri toplamanın gerçekleştiği ve nesnelerin algılandığı katmandır. RFID, BLE, NFC, LoRaWAN ve NB-IoT protokolleri bu katmanda kullanılır.

İkinci katman; sensör verisi toplayan, ağ ve internet bağlantısı sağlayan sistemlerden oluşur. Veri iletimi ve veri işlemenin gerçekleştiği katmandır. İkinci katman, donanım katmanından gelen verileri işler ve üçüncü katmana iletir. IPv6, TCP, UDP, ICMP, 6LoWPAN gibi protokoller bu katmanda kullanılır.

Üçüncü katmanda işlenen veriler bir uygulamaya gönderilir veya başka bir uygulamayı geliştirmek için kullanılır. Kullanılabilir sonuçların gözlemlenebildiği katmandır. HTTP, MQTT, REST ve CoAP protokolleri bu katmanda kullanılır.

1.4. VERİ GİZLİLİĞİ

IoT cihazlarının internete bağlanmasıyla birlikte verilerin gizliliği daha önemli hâle gelmiştir. IoT cihazlar ile kullanıcılardan toplanan kişisel veriler ve iş verileri gizlilik ve güvenlik sorunlarının ortaya çıkmasına neden olur. Buzdolabından en çok neleri tükettiğiniz, kahve makinesinden kahveyi ne kadar sevdiğiniz, kombinizden ev sıcaklığınız, akıllı ev sisteminden evden ne sıklıkla çıktığınız, akıllı tişörtünüzden sağlık durumunuz ile ilgili veriler IoT cihaz üreten firmalara, bulut sistemlerine iletilir. IoT cihaz üreticileri, iletilen bu veriler ile ürünlerini geliştirerek kullanıcıların hayatını kolaylaştırabilir. Aksi durumda bu veriler etik olmayan amaçlar için de kullanılabilir. Bu nedenle verileri toplayan firmalar, verilerin gizlilik ve güvenliğinin sağlanmasından sorumludur. Verilerin iletilmesi ve depolanması sırasında şifrelenmesi gerekir. Verilerin üçüncü şahısların eline geçmesiyle birlikte üretici firmalar itibar kaybı yaşayabilir, kullanıcılar da maddi ve manevi zarar görebilir.

1.4.1. Metadata

Metadata, metaveri veya üst veri olarak da ifade edilir. Metadata, bir cihazın ya da verinin ögelerini tanımlayan bilgilerdir. Metadata kısaca veri hakkındaki bilgiler olarak tanımlanır. Metadata dijital bir nesneye gömülebilir veya ayrı olarak saklanabilir. Metadata genellikle kullanıcı tarafından görülmez. Örnek olarak web sayfalarında kullanılan meta etiketleri, dijital fotoğraf makineleri ile çekilen fotoğraflara işlenen EXIF (fotoğrafın çekildiği tarih, fotoğraf makinesinin markası, modeli ve ayarları, fotoğrafın çekildiği yerin GPS koordinatları vs.) bilgileri verilebilir. Gönderilen bir e-posta; erişim kaynağı, zaman, IP adresi, e-posta erişiminde kullanılan cihaz, alıcı bilgileri gibi pek çok metadatayı da içerir. Metadata bilgileri kişinin gizliliğini istila etmek, hareketlerini izlemek veya muhtemelen parasını veya dijital kimliğini ele geçirmek için de kullanılabilir.

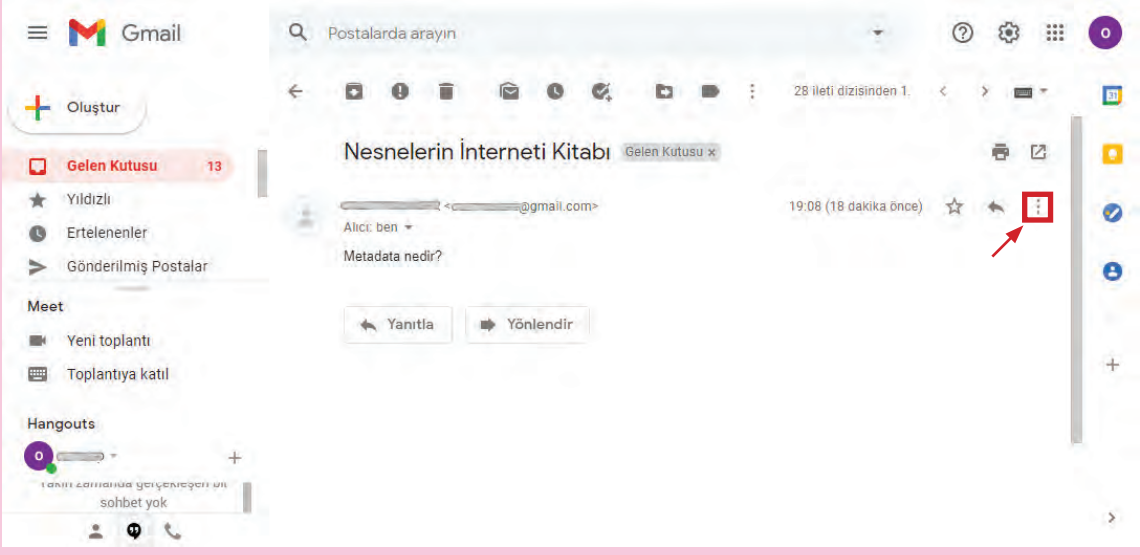
2. UYGULAMA

Metadatanın İncelenmesi

Gmail servisinden gelen bir e-postanın metadatasını görüntüleyiniz. Gmail servisinin ileti üst bilgi aracı **(https://toolbox.googleapps.com/apps/messageheader)** ile metadatayı analiz ediniz.

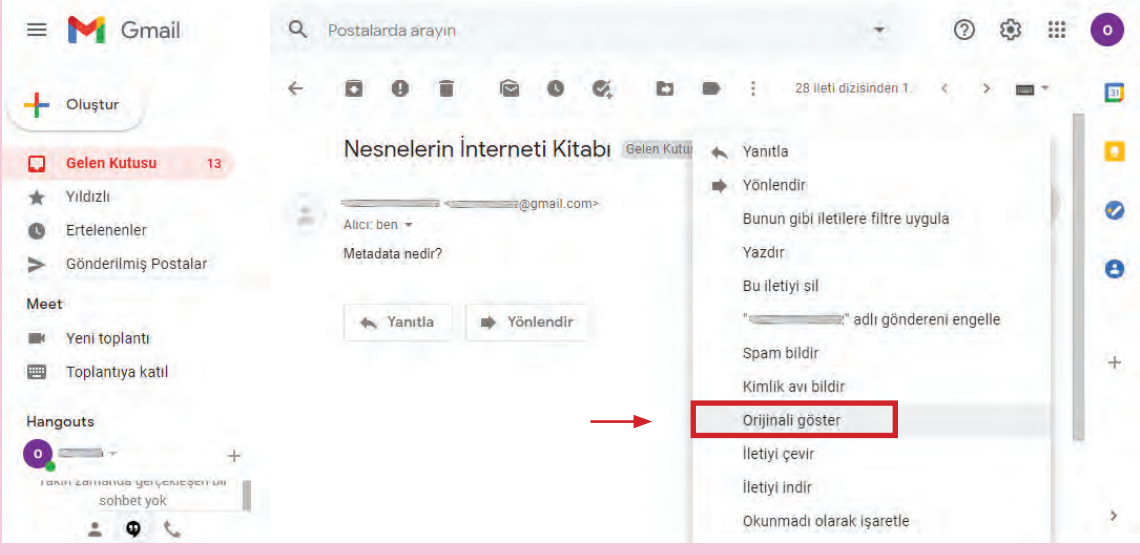
1. Adım : Gmail hesabınıza giriniz. Gmail hesabınız yoksa oluşturunuz ve bir e-posta gönderilmesini sağlayınız. Gmail hesabınıza gelen bir e-postayı açınız.

Görsel 1.33'te kırmızı ok ile işaret edilen Diğer simgesine tıklayınız.



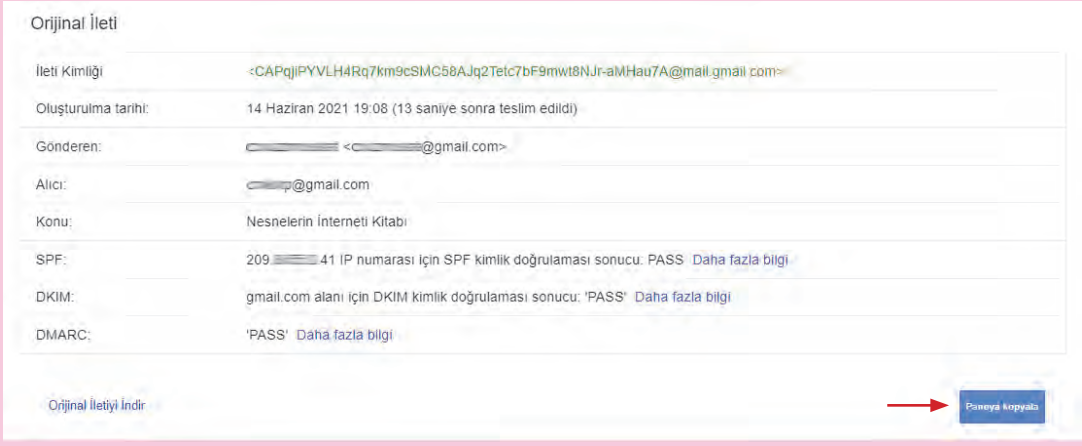
Görsel 1.33: E-postada Diğer simgesi

2. Adım : Görsel 1.34'te kırmızı ok ile işaret edilen Orijinali göster seçeneğini tıklayınız.



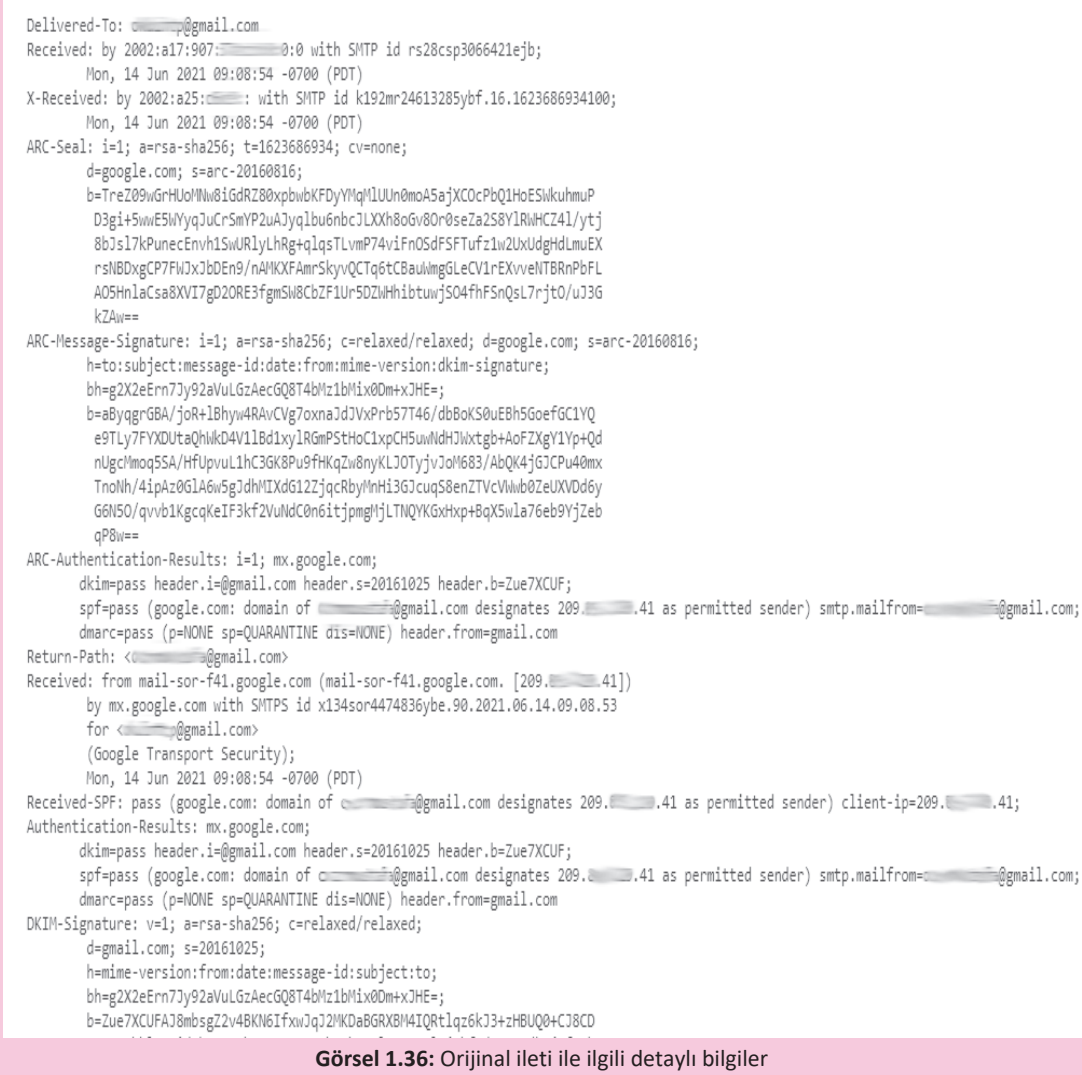
Görsel 1.34: E-postada Orijinali göster seçeneği

3. Adım : Görsel 1.35'te kırmızı ok ile işaret edilen **Panoya kopyala** butonuna tıklayarak orijinal iletiyi kopyalayınız. Görsel 1.35'te orijinal ileti ile ilgili bazı temel bilgiler yer alır.



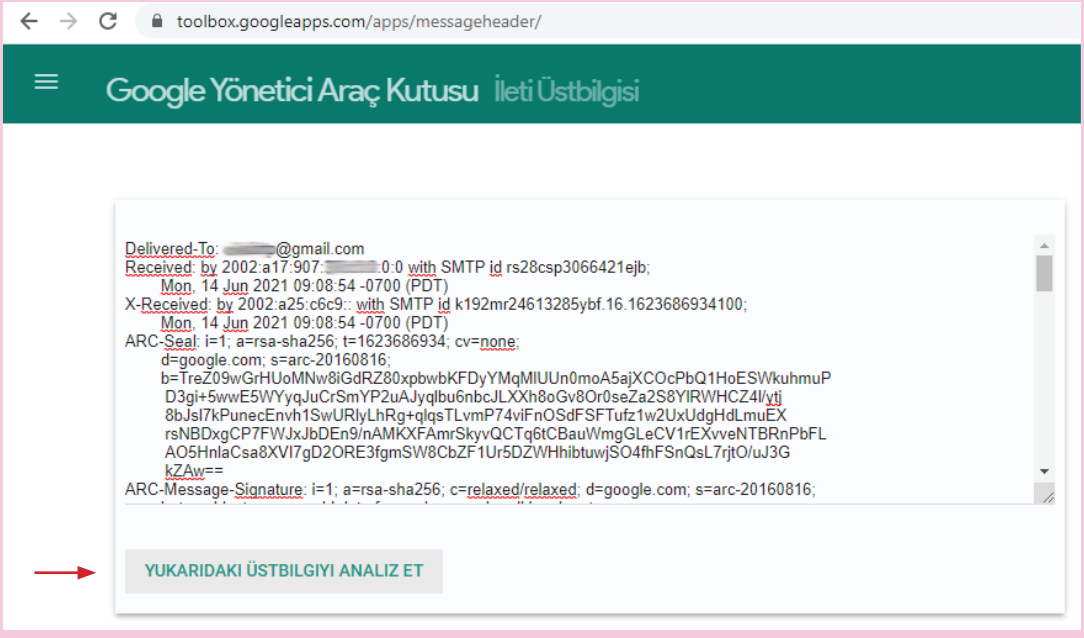
Görsel 1.35: Orijinal iletiyi kopyalama

Sayfanın altında yer alan anlaşılması zor detaylı bilgiler ise Görsel 1.36'da görülmektedir.



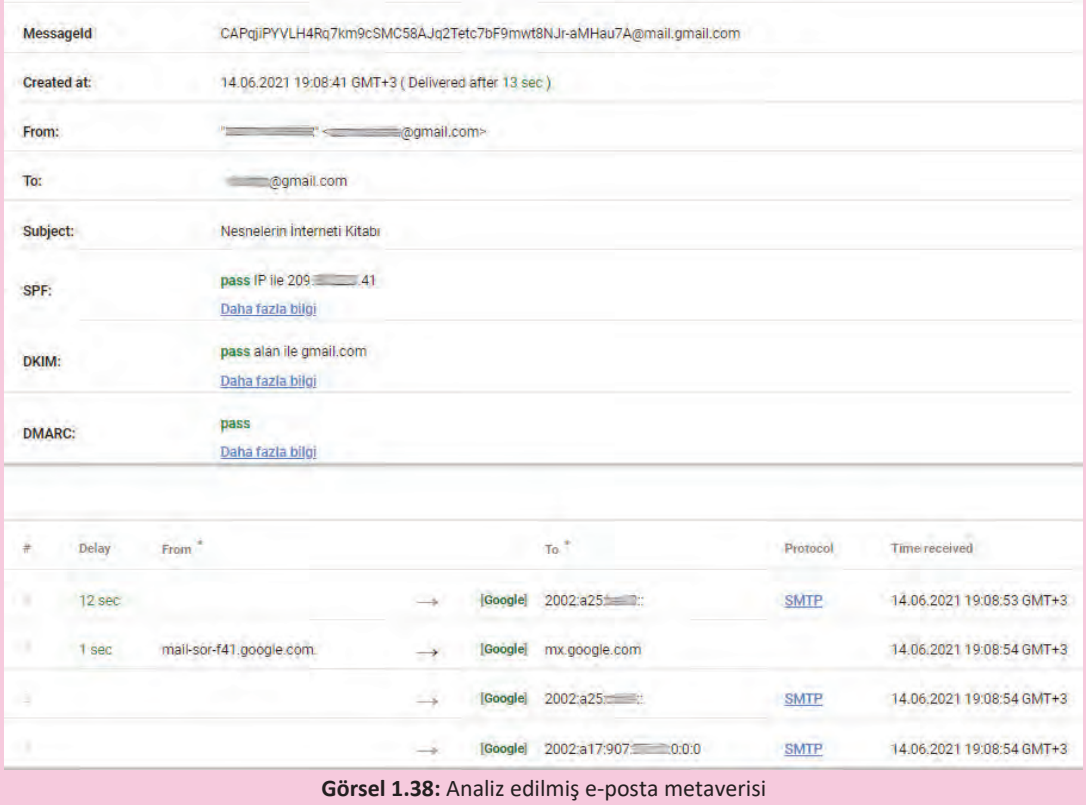
Görsel 1.36: Orijinal ileti ile ilgili detaylı bilgiler

4. Adım : Görsel 1.37'deki detaylı bilgileri daha düzenli, anlamlı şekilde görmek için kopyaladığınız orijinal iletiyi **https://toolbox.googleapps.com/apps/messageheader** web sayfasında bulunan e-posta üst bilgi metin kutusuna yapıştırınız. Görsel 1.37'de kırmızı ok ile işaret edilen **YUKARIDAKİ ÜSTBİLGİYİ ANALİZ ET** butonuna tıklayınız.



Görsel 1.37: İleti üstbilgisini analiz etme

5. Adım: Görsel 1.38'de görülen analiz edilmiş e-posta metaverisini inceleyiniz.



Görsel 1.38: Analiz edilmiş e-posta metaverisi

1.4.2. IoT Cihazlarının Gizliliğe Etkisi

IoT sistemleri, fiziksel ortamları izlemek ve karar verip değişiklikleri uygulamak için tasarlanmıştır. Bu IoT sistemleri, büyük miktarlarda metadataların oluşturulmasını sağlayabilir. IoT sistemleri, kullanıcıları korumak için güvenlik ve gizlilik göz önünde bulundurularak tasarlanmalıdır.

Gizlilikle ilgili öneriler ve tasarım hususları şunlardır:

- **Şeffaflık:** Kullanıcılar ne tür kişisel verilerin toplandığını, verilerin neden toplandığını ve nerede saklanacağını bilmelidir.
- **Veri Toplama ve Kullanma:** Akıllı cihazlar yalnızca amaçları ile ilgili yeterli miktarda veriyi saklamalıdır. Kişinin kimliğini gizleyen veriler kullanılmalıdır.
- **Veri Erişimi:** Akıllı cihazlar tarafından toplanan kişisel verilere kimin, hangi koşullar altında erişebileceği belirlenmelidir.

Bilgi transferi sırasında yetkisiz kullanıcıların verileri ele geçirememeleri gizlilik unsuru ile ilgilidir. Bu unsurun korunması için yetkisiz kullanıcının iletilen veriye erişim sağlaması, verinin içeriği hakkında bilgi sahibi olması engellenmelidir. Bunun için şifreleme ve kimlik doğrulama yöntemleri önerilir.

Verilerin şifrelenmesi, şifreli aktarılması ve saklanması ile ilgili işlem adımları verilerin gizliliğini sağlamak için yaygın biçimde kullanılır. Düşük işlemci hızı, güç tüketimi ve kısıtlı bellek miktarına sahip IoT sistemlerde hafif sıklet (lightweight) kriptografi algoritmalarının kullanılması tercih edilir. Hafif sıklet kriptografi ile güvenlik zafiyetine ve gizlilik ihlaline sebep olmadan, düşük maliyetli ve optimum performanslı şifreleme hedeflenir.

Bilgi transferi öncesinde IoT ağındaki cihazların kimlik doğrulamaları yapılmalıdır. Kimlik doğrulama ile yetkisiz cihazların bilgi transferi sürecinde yer almaları engellenir. Ağdaki her bir IoT cihazının gerçek olduğundan emin olmak, yetkisiz IoT cihazların ağa katılmasını önlemek için kimlik doğrulama yönteminin kullanılması gerekir.

ARAŞTIRMA

Hafif sıklet (lightweight) şifreleme algoritmalarını araştırınız. Araştırmanızı sunu şeklinde hazırlayınız. Sınıfta arkadaşlarınızla paylaşınız.

1.4.3. IoT Cihazlarında Güvenlik

IoT güvenliği hem fiziksel cihaz güvenliğini hem de ağ güvenliğini kapsar. IoT cihazları teknolojiye ayak uydurmak, rekabeti sürdürmek için gerekli ağ bağlantı yetenekleriyle geliştirilir. Üretilen IoT cihazlarının çoğu ağ güvenliğini sağlama, gelişmiş güvenlik özellikleri sunma yönünden yetersiz kalır. Örneğin bir odadaki nem ve sıcaklığı takip eden sensörler gelişmiş şifreleme veya diğer güvenlik önlemlerini yerine getiremez. Ayrıca birçok IoT cihaz "ayarla ve unut" düşüncesiyle geliştirildiği için güvenlik güncellemesi almaz. Güvenlik önlemlerinin mümkün olduğunca çok sayıda güvenlik açığını kapsayacak şekilde tasarlanması gerekir. IoT cihaz verilerinin yasa dışı erişimini, değiştirilmesini veya kaybını önlemek için minimum düzeyde standartlaştırılmış veri güvenliği önlemleri alınmalıdır. Güvenlik açıkları riskini en aza indirmek için güçlü şifreleme ve kimlik doğrulama kullanılmalıdır. IoT cihazların varsayılan kullanıcı adı ve parola bilgileri değiştirilmelidir.

IoT'ta ağ güvenliğini etkileyen bazı faktörler şunlardır:

- Cihaz sayısının artması
- Cihaz çeşidinin fazla olması
- Cihazların konumlarının dağıtık olması
- Toplanan verilerin türünün birbirinden farklı olması
- Toplanan verilerin miktarının artması
- Cihazların güvenlik güncellemesinin yapılmaması

IoT cihazlarının varsayılan oturum bilgilerinin değiştirilmeden kullanılması önemli bir güvenlik zafiyetine neden olmuştur. 2016 yılında Mirai zararlı yazılımı, çoğunluğu IP kamera olan IoT cihazlarına kaba kuvvet

sözlük saldırısı gerçekleştirmiştir. Mirai, bu saldırı sonucunda varsayılan kullanıcı adı ve parolaları değiştirilmeyen IoT cihazlarına Telnet üzerinden erişim sağlamıştır. Mirai, erişim sağladığı IoT cihazları uzaktan kontrol edilebilen botlara dönüştürmüştür. Mirai Botnet, IoT cihazları yıkıcı DDoS saldırıları başlatmak için kullanmıştır.

Teknolojik gelişmelerin etkilediği oyuncakların da akıllı hâle gelmesi beraberinde birçok zafiyeti getirmiştir. Mikrofon ve kamera barındıran bir akıllı oyuncağa saldırgan tarafından erişim sağlanmıştır. Saldırganın bu oyuncaklar üzerinden çocuklarla iletişime geçmesi, çocukların günlük aktivitelerini izlemesi ve kaydetmesi büyük risk olarak görülmüştür.

Giyilebilir akıllı cihazların da yaygın şekilde kullanılmaya başlanması saldırganların dikkatini çekmiştir. Bu cihazların, hem ev hem şirket ortamında kullanılması güvenlik endişelerinin duyulmasına neden olmuştur. Giyilebilir akıllı cihaza ev ortamında bulaşan bir zararlı yazılım, BYOD (Kendi Cihazını Getir) politikasının zayıf olduğu bir şirket ağında hızla yayılarak büyük bir tehdit hâline gelmiştir.

Sağlık sektöründe de IoT teknolojilerinin kullanılması beraberinde birçok güvenlik açığını getirmiştir. Bu güvenlik açıklarından bazıları Bluetooth ve radyo frekansı (RF) özellikli tıbbi cihazlarda ortaya çıkmıştır. Sağlık cihazı üreticileri 2016 yılında akıllı kalp pillerinin saldırganların hedefi olabileceğine dair bilgilendirilmiştir. Kalp pilleri, hastanın kalp atışını kontrol etmeye yardımcı olmak için hastanın göğsüne cerrahi olarak implante edilir. Bu cihazlar, RF üzerinden uzaktan saldırılara karşı savunmasız olan yerleşik bir mikroişlemci ve ürün yazılımı içerir. Kalp piline yapılacak bir siber saldırı hastanın ölümüne neden olabilir. 2017 yılında bu güvenlik açığını yamalayacak yazılım güncellemesi yayınlanmıştır. Bu güncelleme kalp pilinin çıkartılmasına gerek kalmadan RF üzerinden yapılmıştır.

ARAŞTIRMA

Güvenlik açığı tespit edilmiş IoT cihazları hakkında bir araştırma yapınız. Seçtiğiniz bir IoT cihazın zafiyetlerini gösteren bir poster hazırlayınız.

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde parantezlerin içine yargılar doğru ise "D", yanlış ise "Y" yazınız.

1. ( ) Nesnelerin İnterneti bireylerin yaşam kalitesini düşürür.
2. ( ) Nesnelerin İnternetinin endüstride kullanılmasıyla üretim daha verimli hâle gelir.
3. ( ) IoT Referans Modeli üç seviyeden oluşur.
4. ( ) Metadata, veri hakkındaki bilgiler olarak tanımlanır.

B) Aşağıdaki cümlelerde boşluklara uygun olan sözcük ya da sözcük gruplarını yazınız.

5. İnternete bağlı birçok akıllı cihaz ve sensörün belirli protokoller kullanarak veri alışverişi gerçekleştirdiği ağa ..... denir.
6. Ortamdaki fiziksel bir özelliği ölçmek ve ölçülen özelliği sayısal veri olarak üretmek için ..... kullanılır.
7. Yayın ve abone modeli yayıncı, aracı sunucu ve ..... olmak üzere üç önemli bileşen içerir.

C) Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

8. Aşağıdakilerden hangisi Nesnelerin İnternetinin sağladığı avantajlardan biri değildir?

A) Zaman tasarrufu sağlar.
B) Üretilen veri miktarı artar.
C) Sanayide üretimin verimi artar.
D) İşletmenin elde ettiği gelir artar.
E) Karar verme sürecine yardımcı olur.

9. Aşağıdakilerden hangisi bir sistemi veya mekanizmayı kontrol etme, hareket ettirme amacıyla kullanılan bir tür motordur?

A) Ağ geçidi
B) Aktüatör
C) Arduino
D) Kontrolör
E) Sensör

10. Aşağıdakilerden hangisi IoT sistemde bulunan nesnelerin kontrol edilebileceği ve yönetilebileceği platformdur?

A) Bulut
B) Denetleyici
C) Eyleyici
D) Kullanıcı arayüzü
E) Sensör

11. Aşağıdakilerden hangisi üç katmanlı IoT mimari modelinin donanım katmanında kullanılan protokollerden biri değildir?

A) BLE
B) LoRaWAN
C) MQTT
D) NFC
E) RFID

12. Bir akıllı ev sisteminde bulunan güvenlik kamerası kapıya gelen kişinin fotoğrafını çekip depolama birimine kaydeder. Bu sistemde çekilen fotoğraflara EXIF bilgileri de eklenmektedir. Bu bilgileri ifade eden en doğru kavram aşağıdakilerden hangisidir?

A) Analiz
B) Arayüz
C) Gizli bilgi
D) Ham veri
E) Metadata

13. Aşağıdakilerden hangisi IoT cihazlarının gizliliği ile ilgili öneri ve tasarım hususlarından biri değildir?

A) Akıllı cihazların kullandığı verilerde kişinin kimliği gizli tutulmalıdır.
B) Akıllı cihazlar amaçları doğrultusunda yeterli miktarda veriyi depolamalıdır.
C) Akıllı cihaz kullanıcıları kişisel verilerin hangi ortamda depolanacağını bilmelidir.
D) Akıllı cihaz kullanıcıları toplanan kişisel verilerin türü hakkında bilgi sahibi olmalıdır.
E) Akıllı cihazlar tarafından toplanan kişisel verilere üretici firmanın tüm kullanıcıları erişebilmelidir.

14. Aşağıdakilerden hangisi IoT ağ güvenliğini etkileyen faktörlerden biri değildir?

A) Akıllı cihaz sayısının artması
B) Akıllı cihaz çeşitliliğinin artması
C) Akıllı cihazın Ethernet (PoE) üzerinden beslenmesi
D) Akıllı cihazlar tarafından toplanan veri boyutunun artması
E) Akıllı cihazlar tarafından farklı türlerde verilerin toplanması

DEVRE ELEMANLARI, MİKRODENETLEYİCİLER VE SENSÖRLER

2. Öğrenme Birimi

KONULAR

2.1. DEVRE ELEMANLARI
2.2. BREADBOARD KULLANIMI
2.3. MİKRODENETLEYİCİLER
2.4. SENSÖRLER

NELER ÖĞRENECEKSİNİZ?

- Devrede yer alan temel bileşenlerden direncin önemi
- LED kullanırken ön direncin önemi
- LED için gerekli miktardaki ön direnci hesaplama
- Buzzer, transistör, röle ve motor sürücü entegrelerini kullanma ve çalışma mantıklarını kavrama
- Buton ve anahtarların mikrodenetleyici sistemlere bağlantısı
- Temel devre bileşenlerini kullanarak blok şeması verilmiş devreleri breadboard üzerine kurma
- Mikroişlemci ve mikrodenetleyici arasındaki farklar
- Mikrodenetleyicinin temel yapısı ve ek özellikleri
- İhtiyaca göre mikrodenetleyici seçme
- IoT uygulamalarında en çok tercih edilen sensörlerin çalışma mantığı
- Sensörlerin devrelerde kullanımı

TEMEL KAVRAMLAR

anahtar, breadboard, buton, buzzer, direnç, IoT, LED, mikrodenetleyici, mikroişlemci, motor sürücü entegresi, röle, sensör, tinkercad, transistör

HAZIRLIK ÇALIŞMALARI

1. Analog bilgi ve dijital bilgi kavramları size ne ifade ediyor? Açıklayınız.
2. Mikroişlemci ve mikrodenetleyicinin benzer ve farklı yönleri hakkında bildiklerinizi arkadaşlarınızla paylaşınız.
3. İnternet üzerinden kontrol edilebilen donanımların nasıl çalıştığını arkadaşlarınızla değerlendiriniz.
4. IoT cihazları sizce bulunduğu ortamı nasıl dinler ve ortamda nasıl değişiklikler yapar?

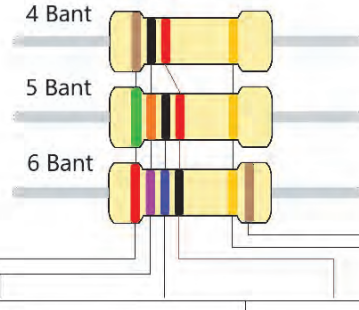
http://kitap.eba.gov.tr/KodSor.php?KOD=14397

2.1. DEVRE ELEMANLARI

İnternete bağlanabilmek ve çevredeki birçok nesneyi internet üzerinden kontrol edebilmek için temel devre elemanları kullanılır.

2.1.1. Direnç

Elektrik akımına karşı gösterilen zorluğa direnç denir. Direnç sembolü "R" harfidir. Direnç birimi Ω'dur (Ohm). Görsel 2.1'deki direnç renk kodları kullanılarak dirençlerin değerleri okunur.



Renkler	Katsayı Değeri			Çarpan	Tolerans	Sıcaklık Katsayısı
	1.Bant	2.Bant	3.Bant			
Siyah	0	0	0	1		
Kahve	1	1	1	10	%1	100 ppm
Kırmızı	2	2	2	100	%2	50 ppm
Turuncu	3	3	3	1 K		15 ppm
Sarı	4	4	4	10 K		25 ppm
Yeşil	5	5	5	100 K	%0,5	
Mavi	6	6	6	1 M	%0,25	
Mor	7	7	7	10 M	%0,10	
Gri	8	8	8			
Beyaz	9	9	9			
Altın				0,1	%5	
Gümüş				0,01	%10	

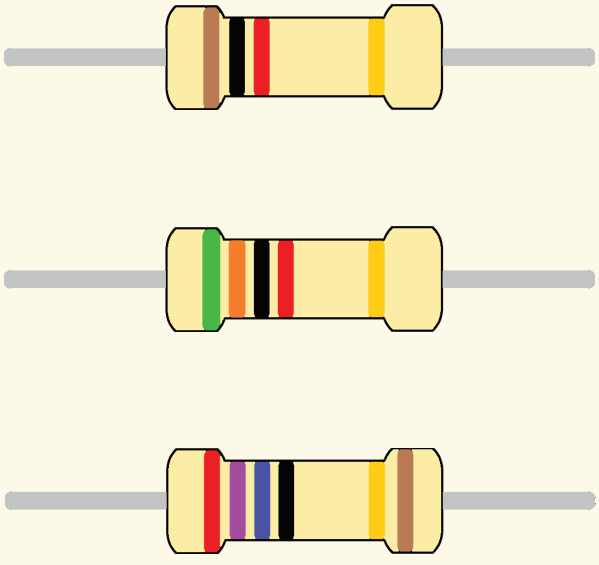
Görsel 2.1: Direnç renk kodları ve direnç değeri hesaplama

Dirençler incelendiğinde toleransın sonda ayrı şekilde konumlandığı görülür. Bu sayede dirençlerde ters okuma durumu oluşmaz. Direnç değerleri hesaplandığında şu sonuçlara ulaşılır:

- 4 bantlı dirençte ilk iki bant katsayı iken üçüncü bant çarpan ve son bant ise toleranstır.
- 5 bantlı dirençte ilk üç bant katsayı iken dördüncü bant çarpan ve son bant ise toleranstır.
- 6 bantlı direnç de 5 bantlı ile aynı olup toleransın yanına sıcaklık katsayısı gelmektedir.

SIRA SİZDE

Görsel 2.1'deki bilgiler ışığında aşağıdaki direnç değerlerini hesaplayınız.



Direnç bir devre içindeki akımı sınırlandırmak ve gerilimi bölmek için kullanılır ($V = I * R_T$). Gerilimi bölmek için [$V_2 = V_s * (R_2 / R_1 + R_2)$] formülü kullanılabilir. Dirençler ile 5 volt seviyesinde çalışan mikrodenetleyici kartının veri gönderme pinindeki gerilim seviyesini 3,3 volta düşürmek için kullanılacaktır.

NOT

Wi-Fi modülleri genellikle 3,3 volt seviyesinde çalışır. Ancak daha ayrıntılı bilgi için kullanılan Wi-Fi modülüne ait döküman incelenebilir.

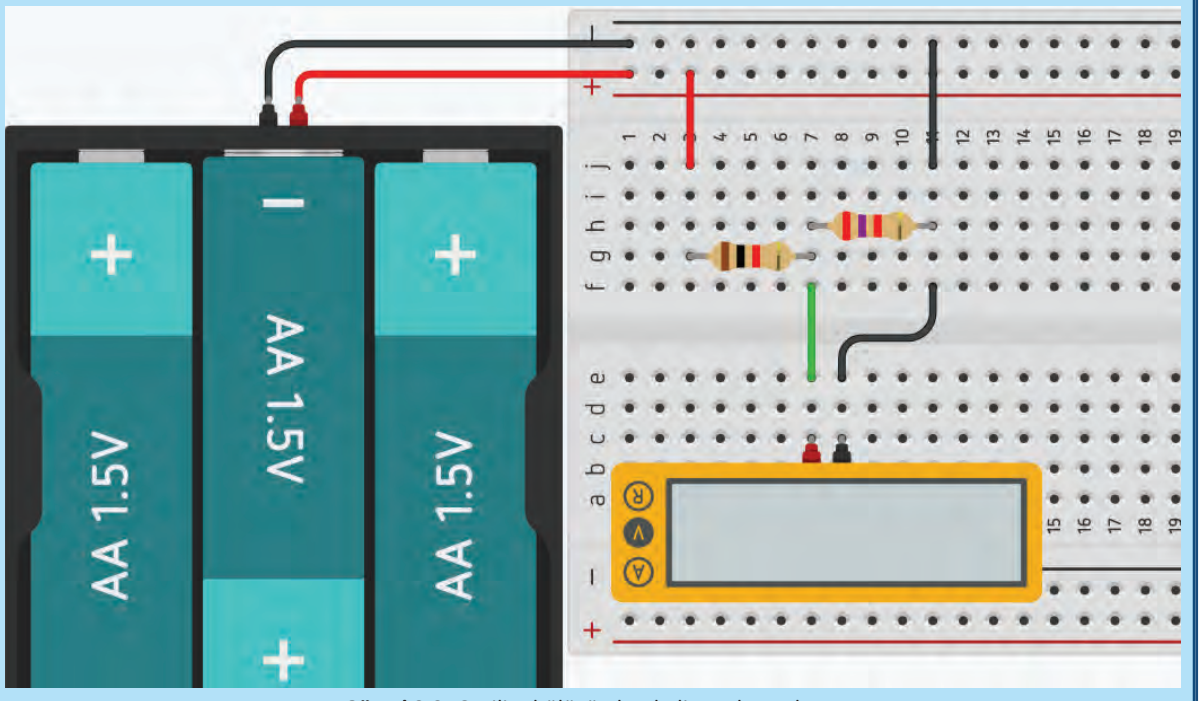
1. UYGULAMA

4,5 volt kaynakla 3,3 voltluk Wi-Fi modülü sürülecektir. R1 direnci 1 KΩ tercih edildiğine göre R2 direnci kaç KΩ **olmalıdır?**

1. Adım : Bu işlem için hesaplamaları [$V_2 = V_s * (R_2 / R_1 + R_2)$] formülünü uygulayarak yapınız.

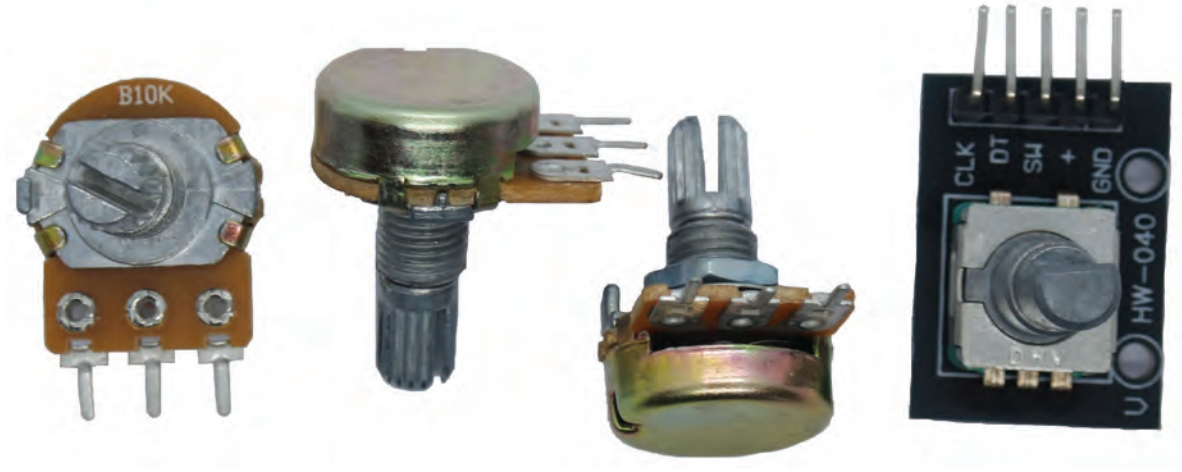
3,3 = 4,5 * (R2 / 1 + R2)
3,3 = 4,5 * R2 / (1 + R2)
3,3 + 3,3 R2 = 4,5 R2
R2 = 3,3 / 1,2 = 2,75 KΩ olarak bulunur.
Bu değerde direnç bulunmadığı için 2,7 KΩ direnç tercih edilir.

2. Adım : Görsel 2.2'deki gibi devreyi Tinkercad sitesi üzerinde kurunuz. Direnç değeri olarak 2,7 KΩ ile 2,75 KΩ arasında direnç üzerine düşen gerilim farkını gözlemleyiniz.



Görsel 2.2: Gerilim bölücü olarak direnç hesaplama

Gerilim bölücü olarak iki direnç yerine potansiyometre kullanılabilir. Potansiyometre sıfır ile maksimum değer arasında istenilen aralıkta ayarlanabilir. Görsel 2.3'te görüldüğü gibi potansiyometrede üç pin bulunmaktadır. Bunlardan orta pin ayarlanabilen, ilk ve son pinler ise güç bağlantısı yapılabilen pinlerdir.



Görsel 2.3: Potansiyometre (Ayarlanabilir direnç)

2.1.2. LED

LED'ler, üzerinden akım geçtiğinde ışık yayan diyotlardır. Ayrıca üzerlerine sabit gerilim düşürerek zener diyot gibi davranırlar. LED'ler renklerine göre farklı gerilim düşümlerine sahiptir fakat genellikle 2 V kabul edilir ve hesaplamalar 2 V 20 mA'e (miliAmper) göre yapılır.

LED renklerine göre gerilim düşümleri Görsel 2.4'te verilmiştir.

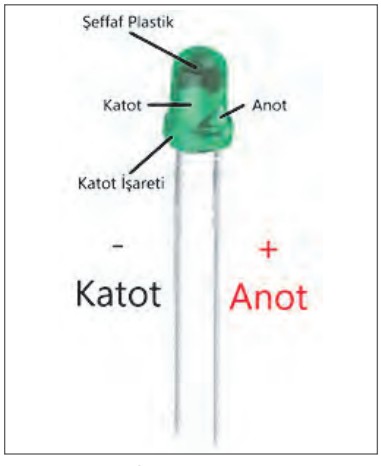
LED Renkleri	LED'e Gerekli Gerilim	LED'e Gerekli Akım
	1,5-2,5 volt	10 – 20 mA
	2,2-3,3 volt	10 – 20 mA
	3,3-4,5 volt	15 – 30 mA
	1,8-2,8 volt	10 – 20 mA
	3,3-4,5 volt	15 – 30 mA

Görsel 2.4: LED renklerine göre ihtiyaç duyulan gerilim ve akım değerleri

Buradaki değerler, ortalama değerlerdir. Değerler, satın alınan LED'in dokümanına bakılarak hesaplanmalıdır. LED öncesi takılacak direnç hesaplaması için;

Kaynak Gerilimi = Direnç Gerilimi + LED Gerilimi formülü kullanılır.

LED'in içyapısı Görsel 2.5'te verilmiştir.



Görsel 2.5: LED içyapısı

2. UYGULAMA

5 volt kaynakla kırmızı LED sürülecektir. LED'in gerilimi 2 V, akımı ise 20 mA olarak belirlenmiştir. Buna göre;

1. Adım : Bu işlem için hesaplamaları **Kaynak Gerilimi = Direnç Gerilimi + LED Gerilimi** formülünü uygulayarak yapınız (**Direnç Gerilimi = Direnç üzerinden geçen akım * Direnç**).

$5\ V = R.20\ mA + 2\ V$

$3\ V = R.20.10^{-3}A$

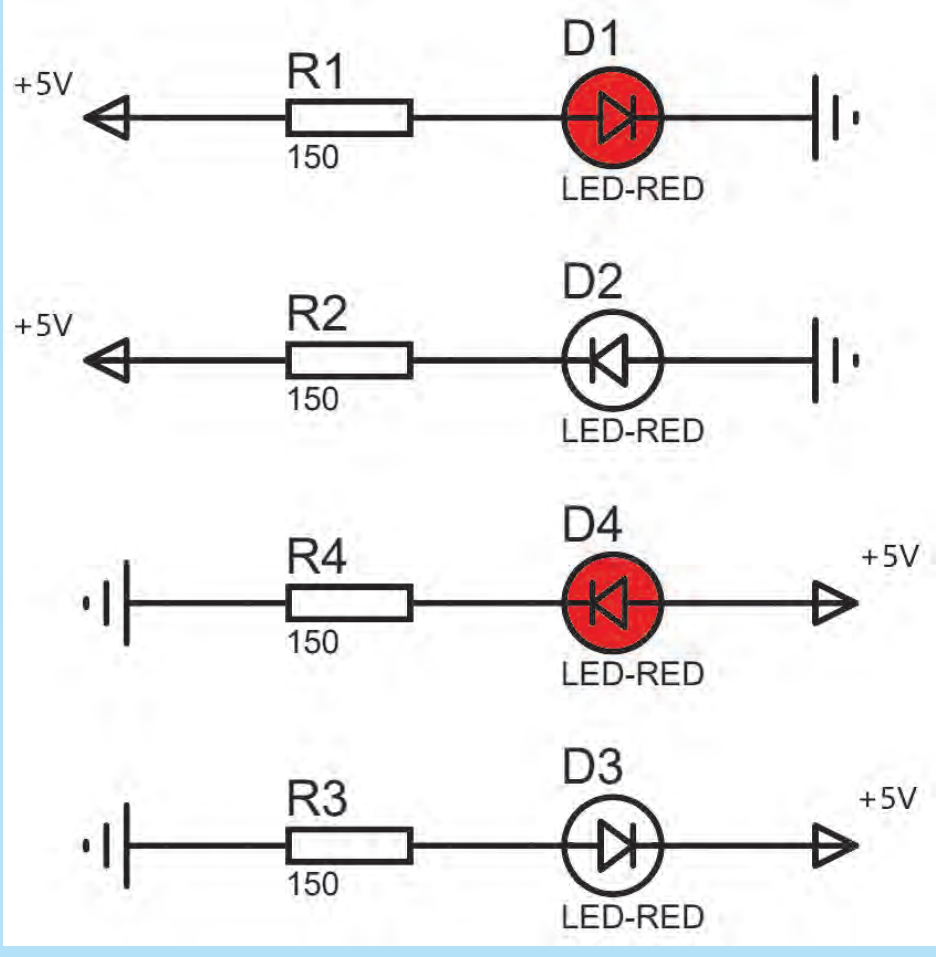
$3000\ V = R.20$

$R = 150\ \Omega$ olacaktır.

NOT

LED'ler diyot gibi davrandığı için üzerinden tek bir yönde akımın akmasına izin verir. LED'ler ters bağlandığında üzerinden akım geçmesine izin vermediği için LED yanmaz (Görsel 2.6).

2. Adım : LED'lerle kurulabilecek olası devreler Görsel 2.6'da verilmiştir. 5 voltluk bir kaynak ile çalışmayı gerçekleştiriniz.



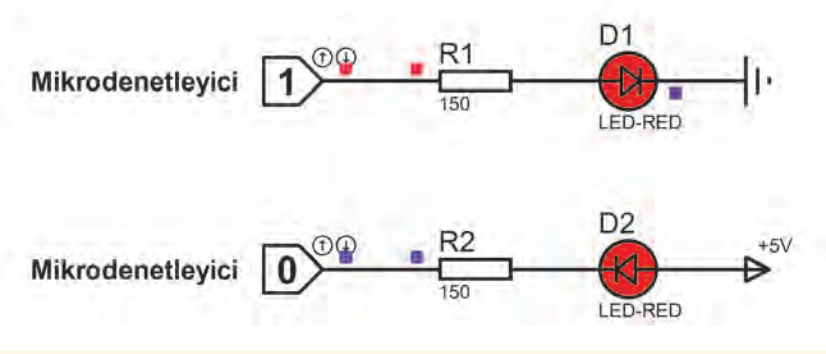
Görsel 2.6: LED'in yanabilmesi için gerekli durumlar

Devreler incelendiğinde LED'in çalışabilmesi için 0'da aktif ve 1'de aktif olmak üzere iki durum bulunmaktadır.

- **0'da Aktif:** Bu bağlantı şeklinde LED'in anot bacağı sistemin gerilim kaynağına bağlı olmalıdır ve LED'in yanabilmesi için mikrodenetleyicili sistemin bağlı olduğu pinden LED'in katot bacağına "-" (GND) uygulanmalıdır.
- **1'de Aktif:** Bu bağlantı şeklinde LED'in katot bacağı sistemin "-" (GND) bağlı olmalıdır ve LED'in yanabilmesi için mikrodenetleyicili sistemin bağlı olduğu pinden LED'in anot bacağına 5 V uygulanmalıdır.

SIRA SİZDE

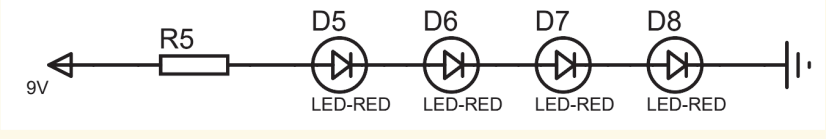
Görsel 2.7'deki devre şemasında mikrodenetleyicili uygulama kartı kullanıp 1 saniye her iki LED'i yakarak, 2 saniye her iki LED'i söndürerek çalıştırınız.

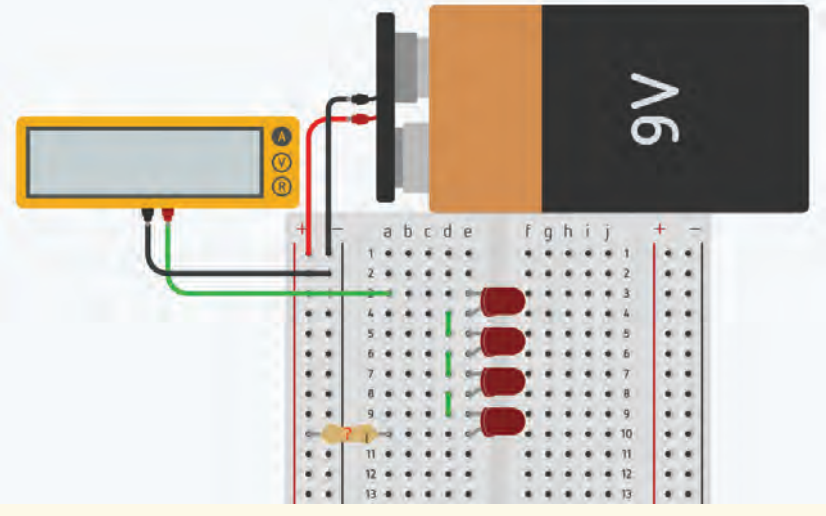


Görsel 2.7: Mikrodenetleyiciyle LED'in lojik 1 ve lojik 0'da yanması

SIRA SİZDE

Birbirine seri bağlanmış 4 adet kırmızı LED, 9 V pil ile çalıştırılmak istenmektedir (Görsel 2.8). Kullanılması gereken ön direnç kaç Ω olmalıdır? Bir adet kırmızı LED'in gerilim değerini 2 V, akım değerini 20 mA olarak hesaplayınız. Hesapladığınız değere göre şekildeki devreyi kurunuz ve akım değerini gözlemleyiniz.

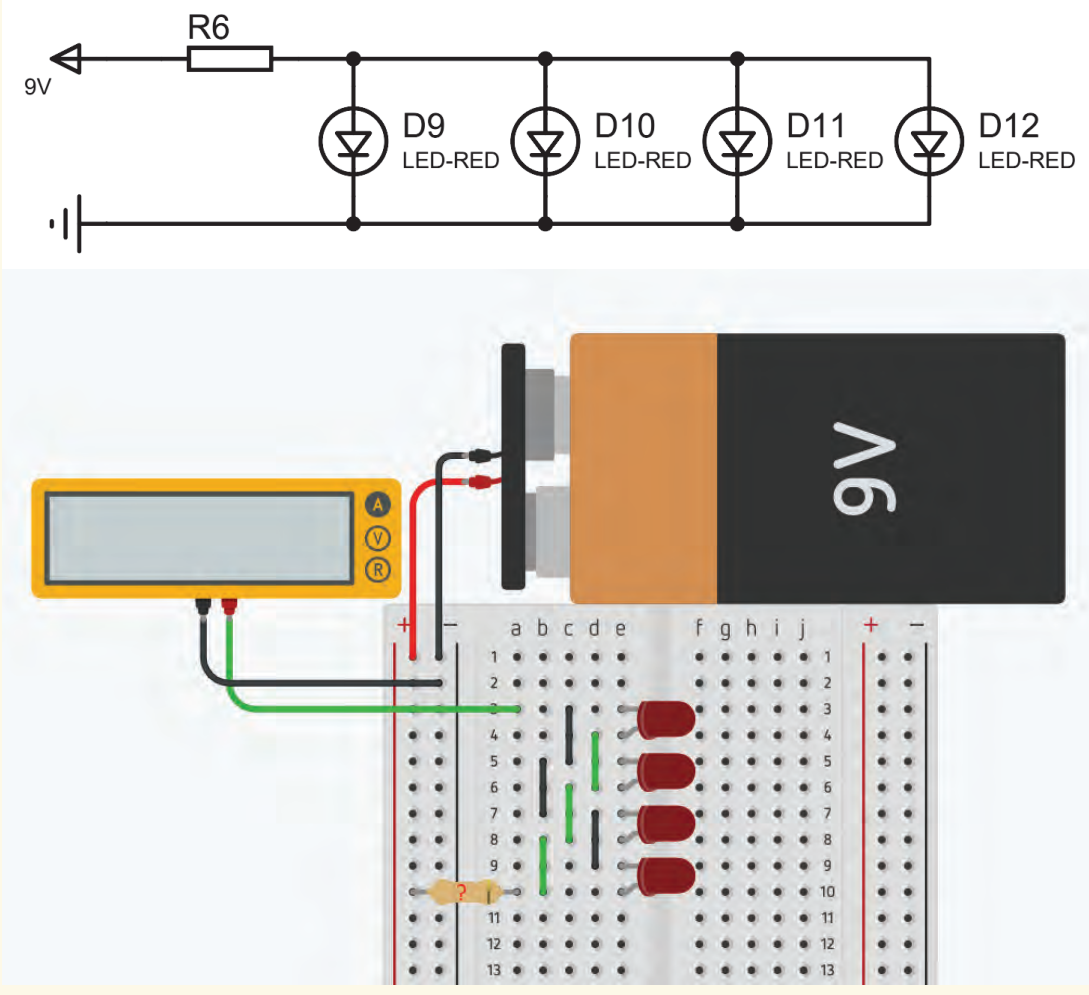




Görsel 2.8: Seri bağlantı

SIRA SİZDE

Birbirine paralel bağlanmış 4 adet kırmızı LED, 9 V pil ile çalıştırılmak istenmektedir (Görsel 2.9). Kullanılması gereken ön direnç kaç Ω olmalıdır? Bir adet kırmızı LED'in gerilim değerini 2 V, akım değerini 20 mA olarak hesaplayınız. Hesapladığınız değere göre şekildeki devreyi kurunuz. Her bir LED'in üzerine düşen akımı ve toplam akım değerini gözlemleyiniz.



Görsel 2.9: Paralel bağlantı

NOT

Bu bağlantılarda LED ve mikrodenetleyicili sistem arasında akımı sınırlamak ve gerilimi bölmek için uygun değerlerde direnç hesaplanarak LED ve mikrodenetleyicili sistem arasına hesaplanan direnç değeri takılmalıdır.

ARAŞTIRMA

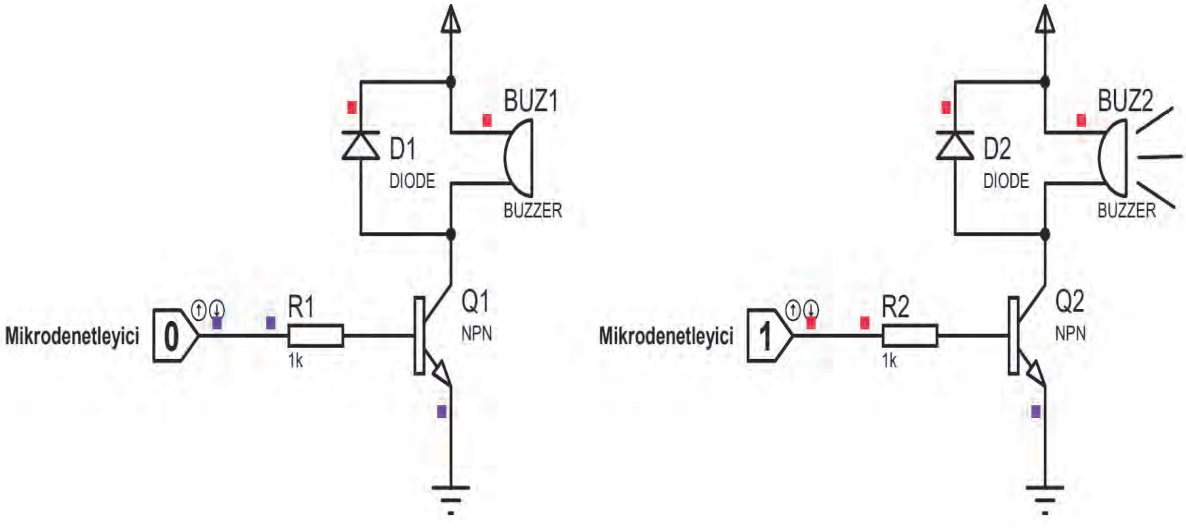
Kirchhoff'un (Kirşof) Akımlar Kanunu ve Gerilimler Kanunu'nu araştırınız. Araştırmanızı bir rapor hâline getiriniz ve sınıfta sunum yapınız.

Birçok devrede LED'ler transistörü, röleyi, motoru veya bir valfi sembol etmek için kullanılır.

2.1.2.1. Buzzer

Aktif (devreli) ve pasif (devresiz) olmak üzere iki buzzer türü bulunur.

- **Aktif (Devreli) Buzzer:** Buzzerin tek ses frekansında temiz bir ses için ayarlanmış bir devre bulunur. Bu tip buzzerlar (+) bacağına mikrodenetleyiciden sinyal gönderildiği anda sabit ses çıkarır. Bu buzzerların sesi daha çok asansörlerin aşırı yük uyarısında ya da arabaların park yaparken mesafe geri bildirimlerinde duyulur.
- **Pasif (Devresiz) Buzzer:** Buzzerin sesini duyabilmek için belirli frekans aralığında sinyal gönderildiğinde ses çıkarır. Gönderilen frekans değiştirildikçe çıkan ses değişir, nota bilgisine göre çaldırma işlemi yapılır.



Görsel 2.10: Buzzerin mikrodenetleyiciyle kullanımı

Aktif ve pasif buzzer türleri aynı şekilde mikrodenetleyiciye bağlanır (Görsel 2.10). Aktif buzzer herhangi bir pin ile çalıştırılabilirken, pasif buzzer PWM yöntemi kullanılarak çalıştırılır.

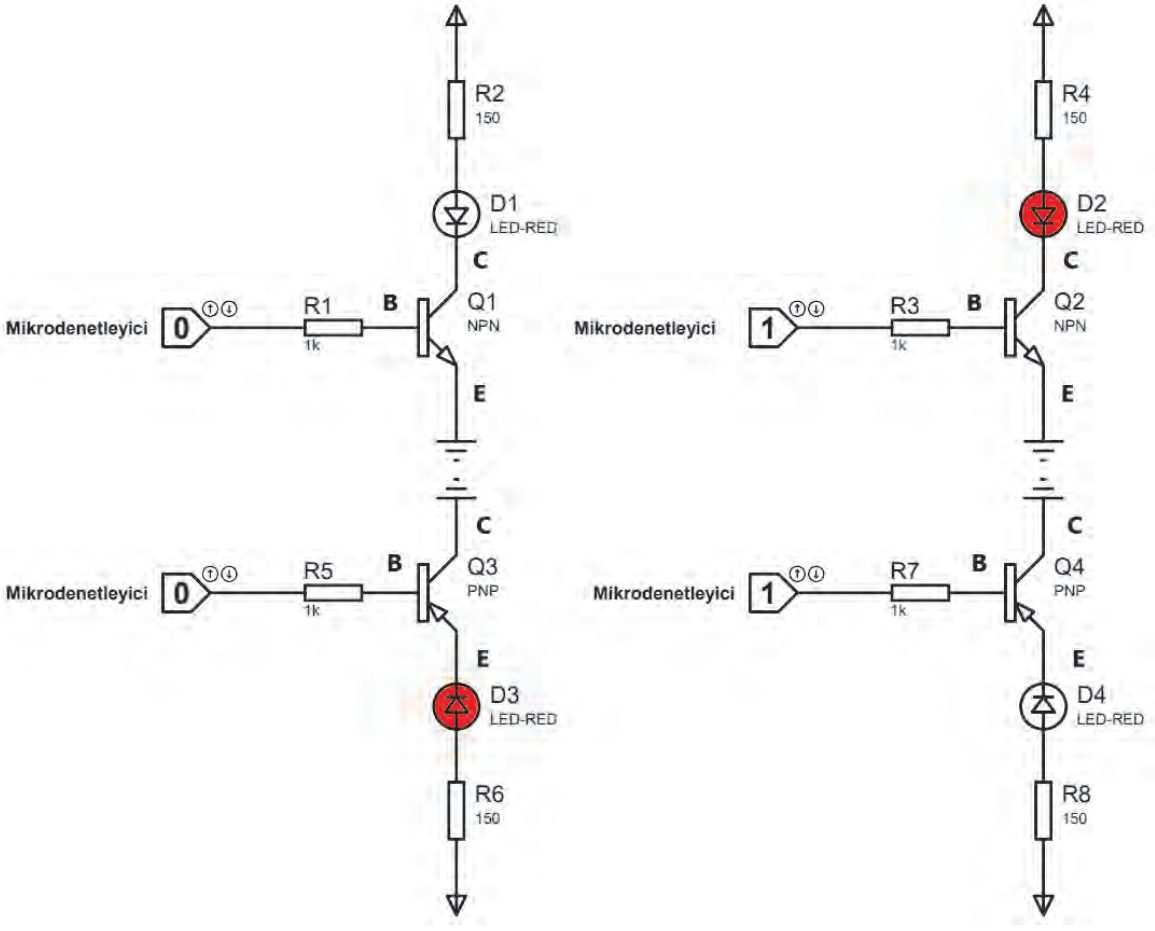
SIRA SİZDE

Görsel 2.10'daki devre şemasında mikrodenetleyicili uygulama kartı kullanıp 1 saniye dijital olarak 1 ve 2 saniye dijital olarak 0 uygulayınız. Bu durumun buzzer üzerindeki etkisini gözlemleyiniz.

2.1.2.2. Transistör

Transistör, küçük elektrik sinyalleri ile yüksek elektrik yüklerinin kontrolünü sağlayan anahtarlama elemanıdır. Transistör genellikle üç bacaktan oluşur. Bu bacaklar; base (bayz-B), emiter (emitter-E) ve kolektör (collector-C) olarak adlandırılır. Transistörlerin NPN ve PNP olmak üzere iki tipi vardır.

- **NPN:** Bayz bacağına uygulanan küçük gerilim ile kolektör ve emiter arasından yüksek gerilimlerin akmasına izin verir. Bayz bacağına uygulanan gerilim kesildiğinde kolektör ve emiter arasındaki bağlantı kesilir (Görsel 2.11).
- **PNP:** Bayz bacağında sinyal uygulanmadığında emiter ve kolektör arasından yüksek gerilimin akmasına izin verirken bayz bacağına küçük gerilim uygulandığında emiter ve kolektör arasındaki bağlantı kesilir (Görsel 2.11).



Görsel 2.11: NPN ve PNP transistörlerin mikrodenetleyiciyle kullanımı

SIRA SİZDE

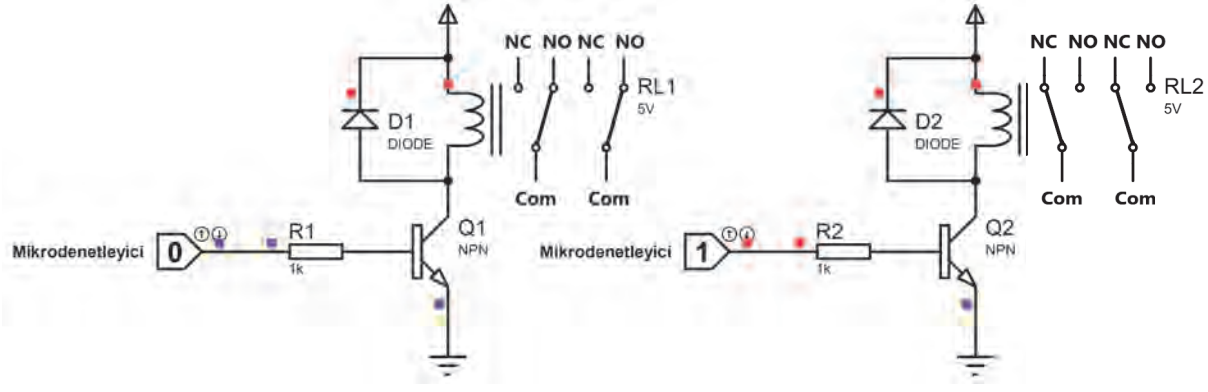
Görsel 2.11'deki NPN ve PNP transistörlerin devre şemalarında mikrodenetleyicili uygulama kartı kullanıp 1 saniye dijital olarak 1 ve 1 saniye dijital olarak 0 uygulayınız. Bu durumun LED'ler üzerindeki etkisini gözlemleyiniz.

2.1.2.3. Röle

Röle, üzerinde manyetikleştirme için bir adet bobin ve COM, normalde kapalı olan anahtar ile normalde açık olan anahtar barındıran bir yapıdır (Görsel 2.12). Bobin pinlerine akım uygulanmadığında NC (Normal Close-Normalde Kapalı) olan pin, COM (ortak) uç ile temas hâlindedir. NO (Normal Open-Normalde Açık) olan pin, COM ucu ile ayrı hâldedir. Rölenin bobin bacaklarına uygulanan gerilim sonucunda bobin üzerin-

Görsel 2.12: Röle

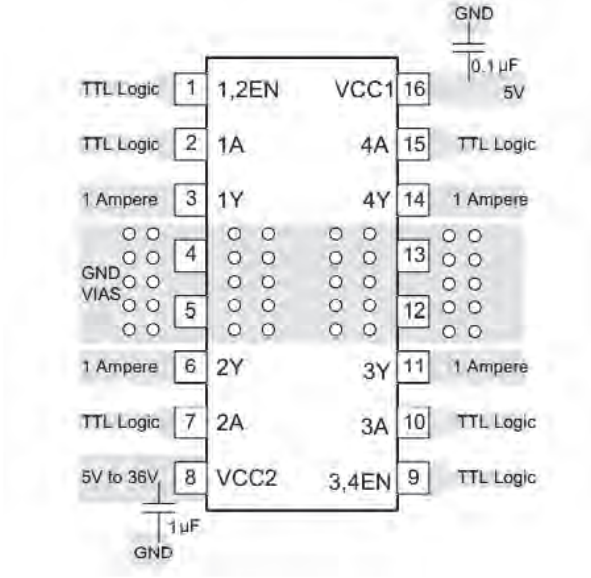
den geçen akım, bobini manyetikleştirir ve COM ucunu kendine doğru çeker. Bu işlem sonrasında COM pini NC'den ayrılıp NO pinine temas eder. Bu yapısı sayesinde mikrodenetleyicili sistemlerin yüksek gerilimli sistemlerden yalıtılması sağlanarak küçük gerilimler ile çok büyük gerilimler kontrol edilebilir hâle gelir. Röleler ile evlerde elektrikle çalışan bütün donanımlar kontrol edilebilir. Evlerdeki yüksek gerilimli bağlantılar COM, NC ve NO pinleri kullanılarak yapılır (Görsel 2.13).



Görsel 2.13: Rölenin mikrodenetleyiciyle kullanımı

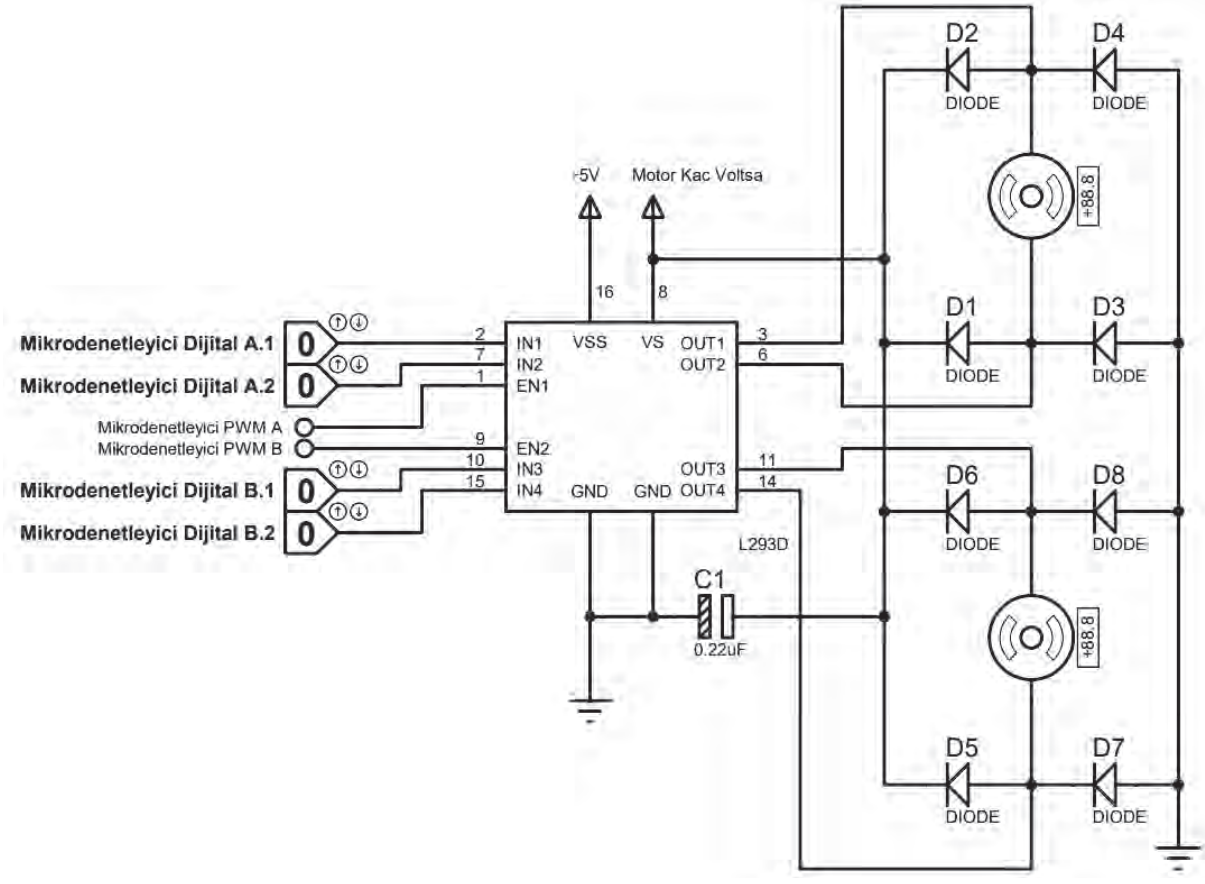
2.1.2.4. Motor Sürücü Devreleri

Motor sürücü devreleri içyapısında 4 adet buffer bulundurur. Bufferlar motor sürücüsünün güç bağlantı noktasına bağlanan kaynak kadar gerilimi ve akımı yükselterek gönderilen 5 V 20 mA seviyesindeki giriş bilgisini motor üzerine yönlendirir. Böylelikle yüksek güçlere sahip motorlar sürülebilir. Görsel 2.14'te L293D motor sürücü entegresinin içyapısı ve bağlantı şeması görülmektedir.



Görsel 2.14: L293(D-B) motor sürücü entegresinin pin yapısı

1 ve 9 numaralı bacaklar Enable (Aktif) bacağıdır. Bu pinlere PWM (Pulse With Modulation) uygulanarak motorların dönüş hızı ayarlanır. 2, 7, 10 ve 15 numaralı bacaklar dijital bilgi uygulanan pinlerdir. Bu pinlere uygulanan düşük seviyeli sinyaller bufferlar yardımıyla güçlendirilerek sırası ile 3, 6, 11 ve 14 numaralı pinlerden çıkış verir. Motorun dönebilmesi için motora bağlı iki pinden biri çıkışa sinyal gönderirken diğerinin bilgi göndermemesi gerekir. Böylelikle çıkış olan pinden çıkış olmayan pine doğru akım akar ve motor o yönde hareket eder. Motoru ters yönde döndürmek için ise giriş pinlerindeki bilgiler terslenir. Motorun iki yöne dönmesi sağlanır. Böylece iki ayrı motor, iki yönde hareket ettirilir. Motor sürücü entegresi kullanılırken Görsel 2.14'teki devre uygulanmalıdır.



Görsel 2.15: L293(D-B) motor sürücü entegresinin mikrodenetleyiciyle çift yönlü ve dönüş hızı ayarlanabilen devre şeması

Görsel 2.15'teki devre şeması Görsel 2.14'teki pin yapısı dikkate alınarak iki ayrı motor, iki yönlü ve dönüş hızı ayarlanabilecek şekilde hazırlanmıştır ancak her bir çıkışa tek motor bağlanarak da dört motor, tek yönlü ve dönüş hızı ayarlanacak şekilde kontrol edilebilir. Ayrıca bu entegreye DC motor haricinde bir adet step motor da sürülebilir.

SIRA SİZDE

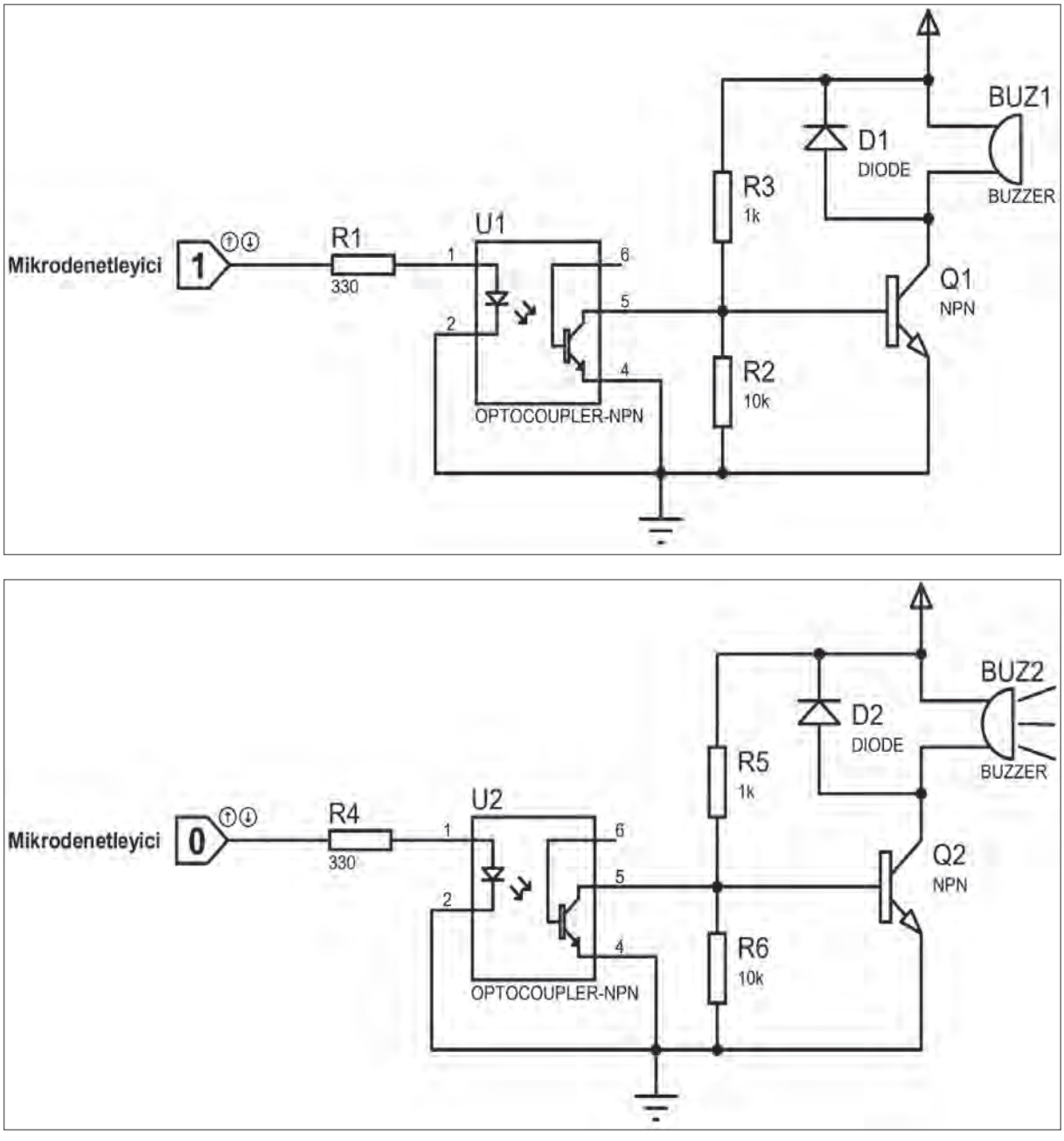
Görsel 2.15'teki devre şemasında mikrodenetleyicili uygulama kartı ile iki motorda aşağıdaki işlemleri gerçekleştiriniz.

- Motorları saat yönünde, yarı hızda, 4 saniye döndürünüz.
- Motorları saat yönünde, maksimum hızda, 4 saniye döndürünüz.
- Saat yönünde döndürdüğünüz her iki motoru durdurunuz.
- Motorları saat yönünün tersinde, yarı hızda, 4 saniye döndürünüz.
- Motorları saat yönünün tersinde, maksimum hızda, 4 saniye döndürünüz.
- Saat yönünün tersine döndürdüğünüz her iki motoru da durdurunuz.

2.1.2.5. Optokuplör

Optokuplör, bir adet LED ve bu LED'e duyarlı foto transistörden (Normal bir transistörün base bacağı ışığa duyarlı bir malzeme ile kaplanıp bu bölüme ışık uygulandığında tetiklenmeyi sağlar.) oluşan ve yalıtım amaçlı kullanılan devre elemanıdır. Optokuplörün transistörlü, opamplı ve triaclı türleri de vardır. Uygulamalarda 4N25 transistörlü optokuplör türü kullanılacaktır.

Optokuplör için örnek devre şeması Görsel 2.16'da verilmiştir. Devre dikkatli incelendiğinde mikrodenetleyicili sistem ile yüksek güçlerde çalışan yapıların birbirinden ışık ile ayrıldığı görülür. Bu sayede yüksek güçlerde çalışan sistemlerdeki herhangi bir arıza, mikrodenetleyicili sisteme zarar vermez. Buzzerın bağlı olduğu bölüme buzzer çıkartılarak yüksek güç tüketen herhangi bir donanım bağlanabilir.



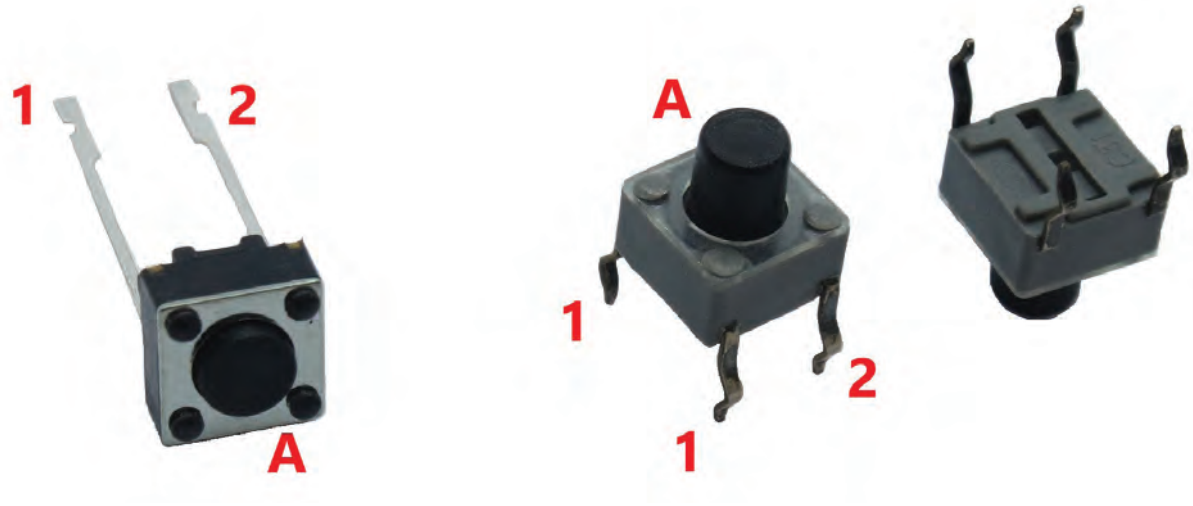
Görsel 2.16: Optokuplörün mikrodenetleyiciyle kullanımı

SIRA SİZDE

Görsel 2.16'daki devre şemasında mikrodenetleyicili uygulama kartı kullanıp 1 saniye dijital olarak, 1 ve 2 saniye dijital olarak 0 uygulayınız. Bu durumun buzzer üzerindeki etkisini gözlemleyiniz. Optokuplörün hangi davranışı gösterdiğini inceleyiniz.

2.1.3. Buton

Buton, üzerindeki (A) düğmesine baskı uygulandığında 1 ve 2 numaralı iki pin arasında bağlantı sağlayan ve (A) düğmesinden baskı çekildiğinde iki pin arasındaki bağlantıyı kesip eski konumuna dönen devre elemanıdır (Görsel 2.17).



Görsel 2.17: İki ve dört pinli buton

Butonun iki pinli ve dört pinli olan çeşitleri vardır. Dört pinli butonlarda birbirine yakın olan iki pin buton özelliği gösterirken birbirine uzak olan iki pin kendi arasında bağlantılıdır (Kısa Devre). Bu sayede 1 numaralı pinler ile 2 numaralı pinler arasında buton özelliği görülür.

2.1.4. Anahtar

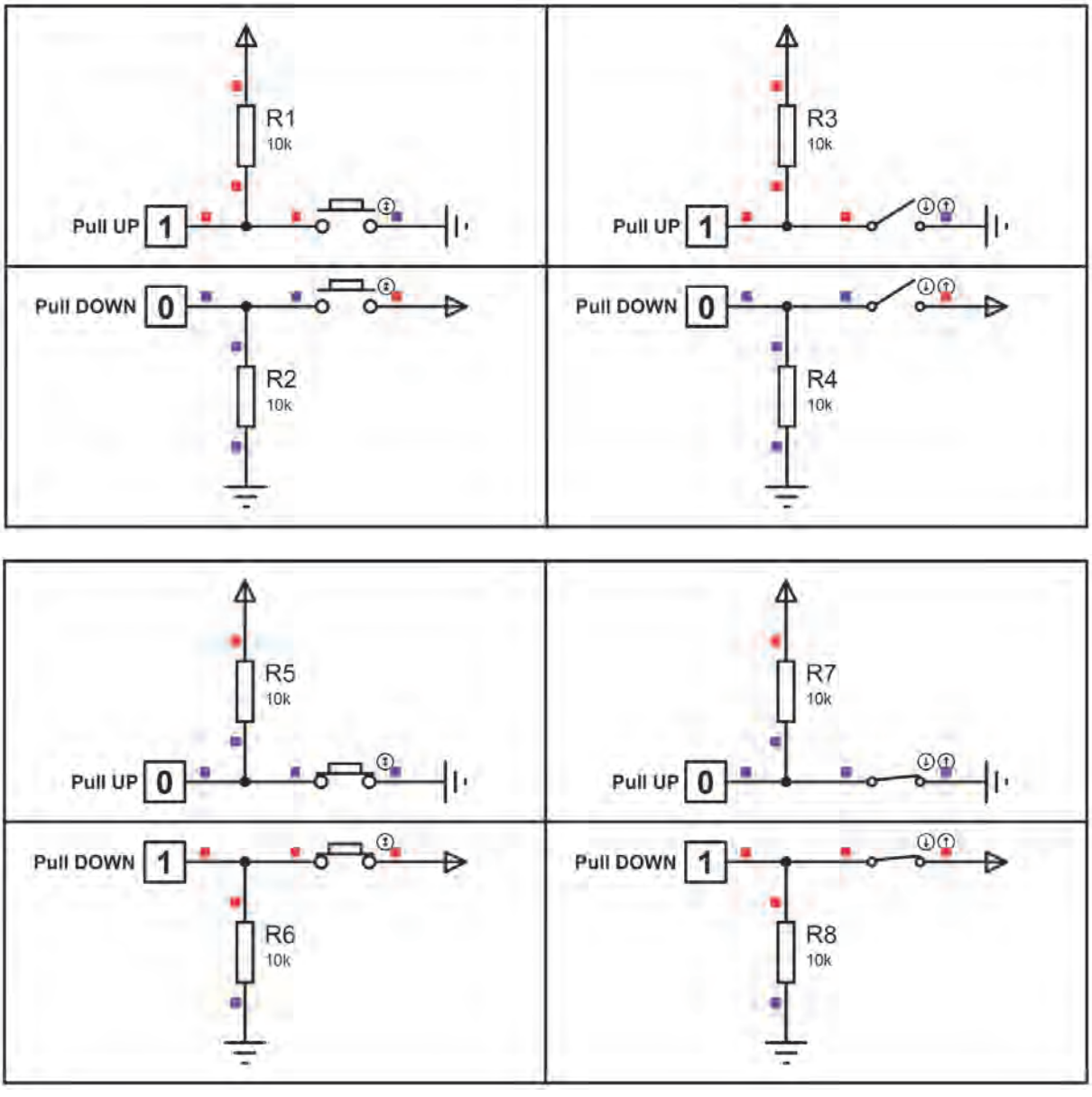
Anahtarların sınır anahtarı ve aç kapa anahtarı olmak üzere iki türü vardır (Görsel 2.18).

- **Sınır Anahtarı:** Temas yüzeyine baskı uygulanmadığında NC ve COM uçları kısa devre hâlinde, NO açıktadır. Temas yüzeyine baskı uygulandığında içindeki anahtar, COM ile NO'yu kısa devre hâline getirip NC'yi açar. Temas yüzeyine baskı bırakıldığında anahtar eski konumuna dönerek çalışır. Çalışma yapısı incelendiğinde anahtarların rölelere benzediği görülür.

Görsel 2.18: Sınır ve aç kapa anahtarları

- **Aç Kapa Anahtar:** Bu anahtar, baskı uygulandıktan sonra konum değiştirip baskı uygulaması sona erse bile konumunu koruyan anahtar tipidir. Eski konumuna getirmek için anahtara tekrar baskı uygulanması gerekir. Bu anahtar, "1" konumunda iki pin arasında kısa devre oluşturup akımın akmasına izin verirken, "0" konumunda iki pin arasındaki bağlantıyı keser ve hattı koparır.

Mikrodenetleyicilerle ve uygulama kartlarıyla buton veya anahtar kullanabilmek için PullUP ve PullDOWN olmak üzere iki farklı yöntem vardır.



Görsel 2.19: Buton ve anahtarın pullUP, pullDOWN yöntemleri ile kullanılması ve aldığı değerler

Görsel 2.19 dikkatli incelendiğinde şu bilgilere ulaşılır:

- PullUP yönteminde butona basılı olmadığı sürece mantıksal olarak 1 bilgisi gönderilir. Butona basıldığı anda mantıksal olarak 0 bilgisi gönderilir.
- PullDOWN yönteminde butona basılı olmadığı sürece mantıksal olarak 0 bilgisi gönderilir. Butona basıldığı anda mantıksal olarak 1 bilgisi gönderilir.

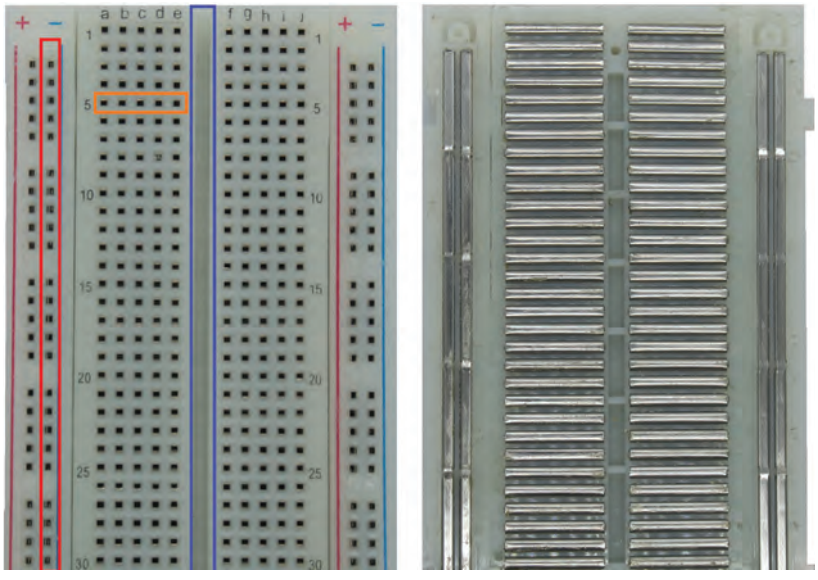
SIRA SİZDE

Görsel 2.19'daki buton ve anahtar bağlama yöntemlerinde mikrodenetleyicili uygulama kartı kullanarak anahtar açık iken butona basıldığında butona basıldığı sürece LED'i yakan ve buton bırakıldığında LED'i söndüren, anahtar kapalı iken butona bir defa basıldığında LED'i yakan ve butona tekrar basıldığında LED'i söndüren sistemi tasarlayınız.

2.2. BREADBOARD KULLANIMI

Breadboard, devrede kullanılacak bileşenleri kolaylıkla monte etmek için ihtiyaç duyulan bir elemandır. Breadboard sayesinde devre elemanlarını lehimlemeye gerek kalmaz. Bu da elemanları yeniden kullanma ve istenilen yerde devre tasarımını rahatlıkla değiştirebilme olanağı sağlar. Breadboardlar piyasada farklı ebatlarda bulunur. Devre tasarımına göre en uygun breadboard seçimini yapmak önemlidir.

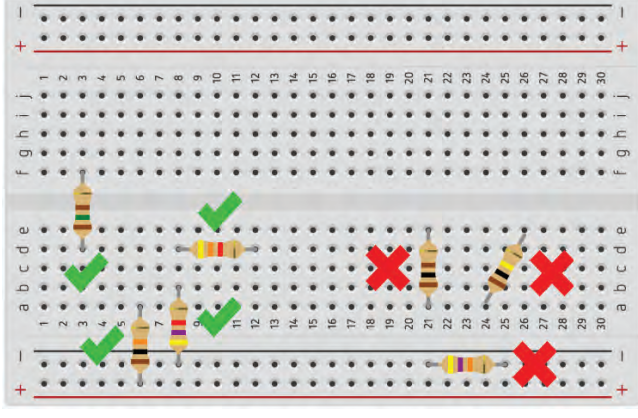
Breadboard kullanılırken dikkat edilmesi gereken nokta, dikey ve yatay şeritlerdir.



Görsel 2.20: Breadboard üzerinde yatay ve dikey şeritler

Görsel 2.20'deki breadboard incelendiğinde kırmızı ile gösterilen alan **bus şerit** olarak adlandırılır. Bus şerit genellikle güç beslemeleri için kullanılır. Bus şerit dikey olarak iletim hâlindedir. Burada bulunan "+" ve "-" etiketleri bağlayıcı değildir, ancak yapılan devrelerde en çok ihtiyaç duyulan bağlantılar "+" ve "-" olduğu için bu şekilde etiketlendirilmiştir. Turuncu renk ile gösterilen alan **soket şerit** olarak adlandırılır ve devredeki bileşenleri takmak için kullanılır. Mavi renkte gösterilen orta kısım **entegre bölgesi** olarak adlandırılır ve breadboardu ortadan ikiye bölerek breadbordun sol ve sağ kısımları arasındaki iletimi keser. Bir başka deyişle soket şeritler beşer yatay pimden oluşur.

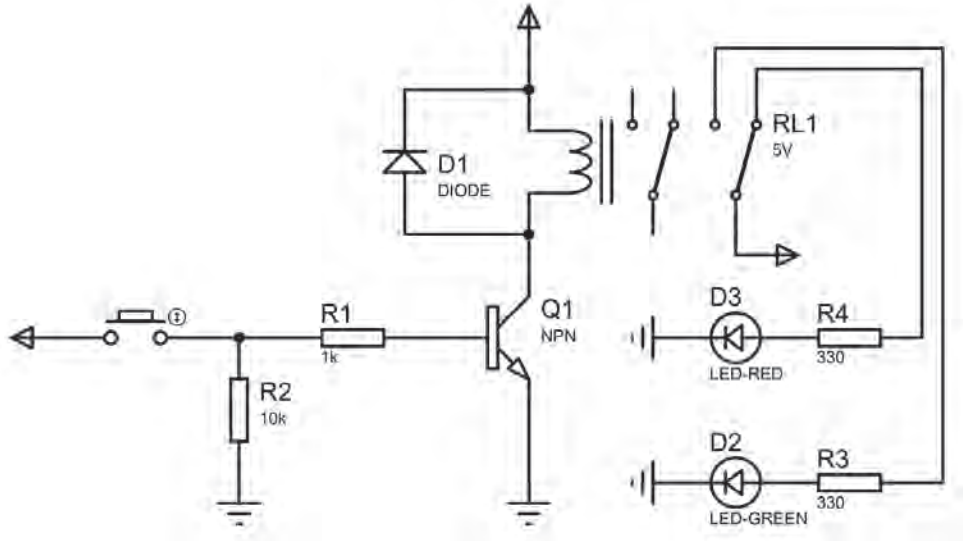
Elemanlar breadboarda yerleştirilirken bus şerit ve soket şeride dikkat edilmelidir. Örnek yerleşimler Görsel 2.21'deki gibi olmalıdır. Hatalı olan yerleşimlerde bus şerit ve soket şerit bağlantılarına dikkat edilmemiştir. Uygulanacak akım kendine en kolay yolu tercih edeceği için direnç bağlantılarının devre üzerinde hiçbir hükmü ve etkisi olmayacaktır. Bir diğer hatalı bağlantıda ise çapraz bağlantılar devre okumayı güçleştirdiği için tercih edilmemektedir.



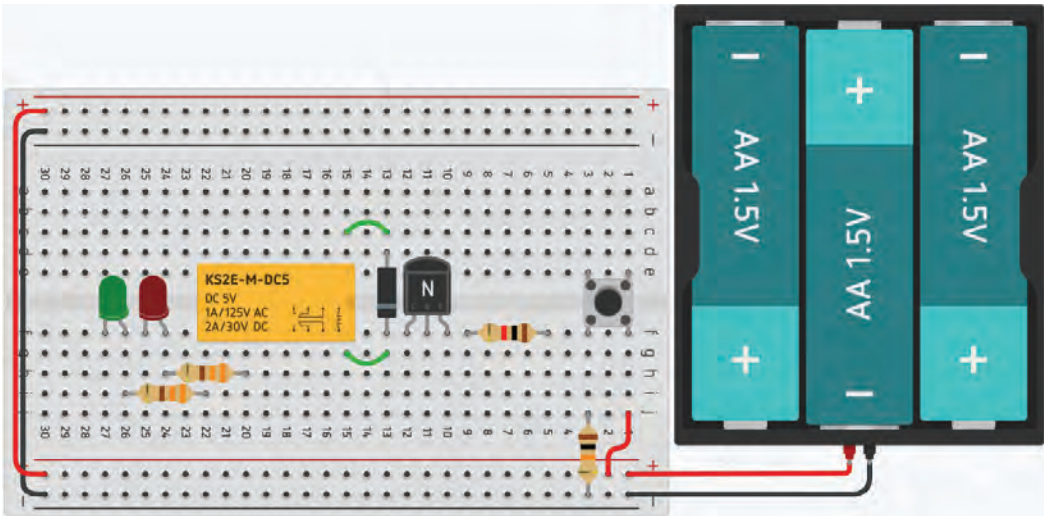
Görsel 2.21: Breadboard kullanımı

SIRA SİZDE

Görsel 2.22'de blok şeması verilen devreyi Görsel 2.23'te verilen breadbord üzerine eksik kalan yerleri tamamlayacak şekilde uygulayınız. İşlemi tamamladıktan sonra devreyi Tinkercad sitesi üzerinde kurup devrenin çalışma şeklini test ediniz.



Görsel 2.22: Buton ile röle sürme devresi



Görsel 2.23: Devredeki eksik bağlantıların tamamlanması

SIRA SİZDE

Görsel 2.22'deki röle devre şemasında mikrodenetleyicili uygulama kartı kullanıp 1 saniye dijital olarak 1 ve 1 saniye dijital olarak 0 uygulayınız. Bu durumun LED'ler üzerindeki etkisini gözlemleyiniz.

2.3. MİKRODENETLEYİCİLER

Mikrodenetleyicilerin merkezinde mikroişlemci yer alır. Mikrodenetleyici, Harvard veya Von Nuemann mimarilerinden biri kullanılarak bellek (RAM&ROM) yönetimi oluşturulup giriş-çıkış portlarının bağlanması ile elde edilen temel yapıdır. Mikroişlemci, bellek ve giriş-çıkış portları bir araya gelerek mikrodenetleyiciyi oluşturur. Mikrodenetleyici sistemine ek özellikler (ADC-Analog Digital Converter-Analog Dijital Çevirici, DAC-Digital Analog Converter-Dijital Analog Çevirici, seri haberleşme portu, kesmeler, zamanlayıcı ve sayıcı) katılabilir. Mikrodenetleyiciler, yazılan programı ROM hafızasına alarak mikroişlemci üzerinde derleyen ve elektronik makinenin amacına göre çıkışlar veren küçük bilgisayarlardır. Mikrodenetleyiciler, çalıştırılması istenen programı hazırlayıp programın kontrolünü yapabilme yetisine sahiptir ve gerçek zamanlı uygulamaları çalıştırmak için tasarlanmıştır.

Genel olarak mikrodenetleyicinin yapısında Görsel 2.24'teki birimler bulunur. Mikrodenetleyicinin tasarlanma amacına göre bu birimlere ek özellikler getirilebilir.

Görsel 2.24: Mikrodenetleyici içyapısı

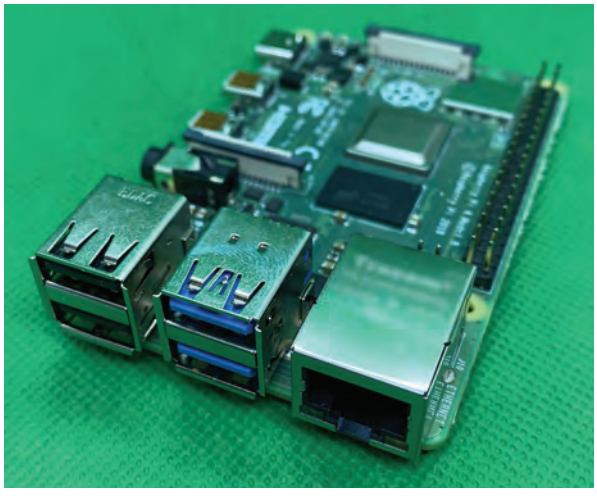
- **MİB (Merkezî İşlem Birimi):** ROM bellekte yer alan programın çalıştırılmasını ve diğer birimlerle iletişimini sağlar.
- **RAM (Random Access Memory / Rastgele Erişimli Bellek) Birimi:** Kullanıcının tanımlamış olduğu değerlerin ve yapılan işlemlerin geçici olarak tutulduğu bellek türüdür.
- **ROM (Read Only Memory / Sadece Okunabilir Bellek) Birimi:** Mikrodenetleyicilerde sistemin programlandığı şekilde çalışması için kaydedilmiş programları tutar.
- **G / Ç Portları:** Bu portlar dış ortama gerekli sinyallerin gönderilmesinde veya dış ortamdan istenen sinyallerin alınmasında kullanılır.
- **Seri / Paralel İletişim Birimleri:** Seri haberleşmede MCU sistemi içinde senkron (SPI, I2C) ve asenkron (USART) olmak üzere iki iletişim yöntemi kullanılır. Asenkron olan haberleşme türünde sadece 2 pin (Tx, Rx) yeterlidir. Senkron haberleşmede ise I2C için veri gönderme ve ona eşlik eden saat darbesi pini, SPI için ise bağlantı yapılacak donanım sayısı kadar pin ve data gönderme pini gereklidir. Tasarlanan sistemde yeterli pin sayısı varsa ve veri aktarılacak mesafe kısa ise paralel bağlantı tipi tercih edilebilir. Paralel bağlantıda veri hattının bit sayısı kadar pin kullanılmalıdır.
- **A / D ve D / A Çeviriciler:** A / D çeviriciler çevresel ortamdan alınan analog sinyalleri dijital, bir başka deyişle ikilik sayı sistemine çevirirken, D / A çeviriciler bu işlemin tam tersini yapmaktadır.
- **Zamanlayıcı ve Sayıcı Birimi:** Mikrodenetleyici içinde sayma ve zamanlama görevlerini gerçekleştirerek program akışını bozmadan belirlenen zaman veya sayıya ulaşıldığında kesme işlemini gerçekleştirir. Kesme ile ilgili işlem tamamlandıktan sonra mikrodenetleyicinin içinde çalışan ana programda kaldığı yerden devam eder.
- **Kesmeler:** Mikrodenetleyicilerin sahip olduğu ek donanımlar kadar kesmeleri vardır. Kesmeler, mikroişlemciye yük olmadan istenen işlemi bitirerek ilgili kesmenin mikroişlemciyi haberdar etmesine dayanır. Örneğin seri haberleşme kanalından USART yöntemi ile veri gelmeye başladığı anda mikrodenetleyicinin içinde seri haberleşme birimi mikroişlemciye kesme gönderip kendisine veri geldiğini bildirerek veriyi okur. Kesme bilgisini alan mikroişlemci, seri haberleşme biriminden gelen veriyi okur ve işlemine devam eder.

Mikrodenetleyiciler yapılarındaki donanım parçalarını (MİB, RAM, ROM, G/Ç birimleri vb.) tuttukları için gömülü sistemler olarak da adlandırılır. Mikrodenetleyiciler genellikle sensörlerden aldıkları çevresel verileri, oluşturacakları sistemin amaçlarına uygun şekilde işleyerek çalıştırır.

Bir robotik projesi için mikrodenetleyici seçerken şunlara dikkat edilmelidir:

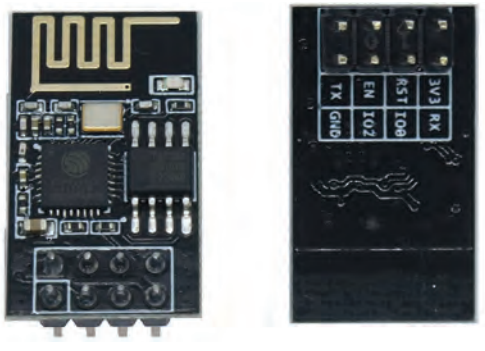
- Birçok ek özelliğe sahip olması (seri bağlantı tipleri, zamanlayıcı ve sayıcı vb.)
- Ucuz ve kolay elde edilmesi
- Programlama kolaylığı
- Kaynak ve kütüphanelerinin çok olması
- Mikrodenetleyici ek donanımlarının (shield) olması

IoT cihazlarda merkezde bir mikrodenetleyici ve buna ek olarak Wi-Fi ağına bağlanabilen bir modül yer alır. Bu sistemlerde Atmel, Microchip Pic veya STM mikrodenetleyicileri kullanılır. Bu mikrodenetleyiciler ile birlikte ESP ailesinden Wi-Fi modülü ile ev ağına bağlantı sağlanarak internet üzerinden mikrodenetleyiciye bağlı olan herhangi bir cihaz kontrol edilebilir (Görsel 2.25).



Görsel 2.25: Raspberry Pi 4 kontrolcüsü

Görsel 2.26'daki ESP Wi-Fi modüllerinin üzerinde yer alan mikrodenetleyici ile yukarıda adı geçen mikrodenetleyiciler olmadan da internet üzerinden nesneler kontrol edilebilir. Görsel 2.25'teki Raspberry Pi gibi üzerinde kendi Wi-Fi modülünü barındıran sistemler de bu amaç için kullanılabilir.



Görsel 2.26: ESP8266-1 Wi-Fi modülü ve pin yapısı

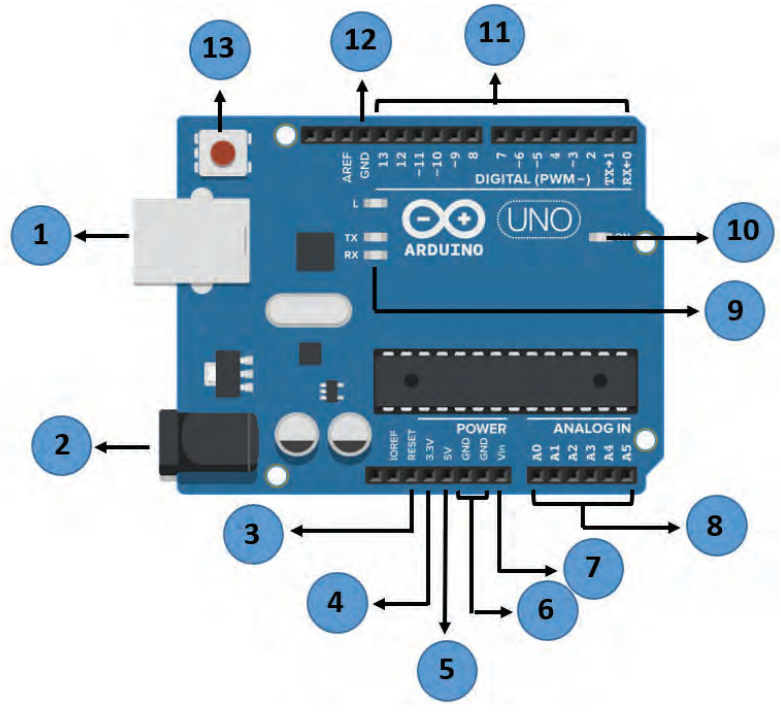
2.3.1. Nesnelerin İnterneti Uygulamalarında Kullanılan Mikrodenetleyici Devre Kartları

Nesnelerin interneti ve robotik uygulamalarında kullanılan kartlar benzerlik gösterir. Bu uygulamalar için geliştirilen özel mikrodenetleyici kartlar bulunur. Bunların başında Arduino UNO, Arduino Wi-Fi, ESP8266, ESP32 ve Rspberry Pi gibi kartlar bulunur.

2.3.1.1. Arduino UNO Mikrodenetleyici Kartı

Arduino UNO, Arduino kartları arasında en fazla tercih edilen modellerinden biridir. Robotik uygulamaları ve nesnelerin interneti uygulamaları geliştirmek için uygun bir mikrodenetleyici kartıdır. Mikro kontroller olarak ATmega328P işlemcisi kullanmaktadır. Altı analog girişe ve altı tanesi Sinyal Genişlik Modülasyonu (PWM) kullanan 14 giriş / çıkış pinine sahiptir.

PWM: Arduino gibi kartlarda kullanılan sinyal modülasyon tekniğidir. PWM tekniğinin altında anahtarlama prensibi yatar. Örneğin kullanılan PWM pini (iğnesi, ayağı, bacağı) "ON" konumunda ise bu pine 5 volt uygulanırken pin "OFF" konumuna alındığında ise 0 volt uygulanır. Bu sayede güç kaybı önlenir ve dijital kontrol üniteleri daha kolay bir şekilde kontrol edilir. Görsel 2.27'de Arduino UNO mikrodenetleyici devre kartının yapısı ve Tablo 2.1'de bağlantı uçlarının açıklamaları yer almaktadır.



Görsel 2.27: Raspberry Pi 4 kontrolcüsü

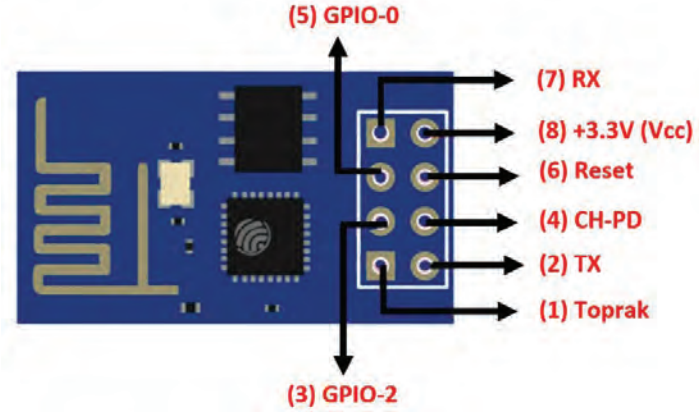
Tablo 2.1: Arduino UNO Mikrodenetleyici Devre Kartının Bağlantı Uçları ve Açıklaması

Gösterim Numarası	Açıklaması
1	USB kablosu bağlantı girişidir. Bilgisayar ile Arduino kartı arasında veri iletişimi yapmak için kullanılır.
2	Arduino kartı için güç girişidir.
3	Arduino kartının resetlenmesi işlemi için kullanılan pin girişidir. Bu pine bağlanacak bir buton ile dışarıdan Arduino kartına yüklenmiş olan program yeniden başlatılabilir.
4	3.3 volt çıkış veren besleme pinidir. Arduino kartı ile kullanılan bileşenler 3.3 volt ve 5 volt ile çalışmaktadır. 3.3 volt ile çalışan bileşenler için bu pin kullanılır.
5	5 volt çıkış veren besleme pinidir. Arduino kartı ile kullanılan bileşenler 3.3 volt ve 5 volt ile çalışmaktadır. 5 volt ile çalışan bileşenler için bu pin kullanılır.
6	GND (Ground-Toprak) - Arduino'da devreyi topraklamak için kullanılan pindir.
7	Vin olarak adlandırılan bu pin Arduino kartına AC şebeke güç kaynağı gibi haricî bir güç kaynağından güç sağlamak için kullanılabilir.
8	Arduino'da 5 analog pin bulunur (A0, A1, A2, A3, A4 ve A5). Bu analog pinler, nem ve sıcaklık sensörü gibi analog sensörden gelen sinyali okur ve bunları dijital bir değere dönüştürür.
9	TX, RX LED'leri; veri gönderme (TX) ve alma (RX) sırasında veri alışverişini gözlemlemek için kullanılan LED'lerdir.
10	Arduino kartının güç girişi doğru bir şekilde bağlandığında yanması gereken LED'dir. Eğer LED yanmıyorsa güç bağlantısında problem olabileceği düşünülebilir.
11	PWM pinleridir. Arduino UNO mikrodenetleyici kartı 14 adet dijital giriş / çıkış (I / O) pinine sahiptir. Bu pinler 0 ve 1 değerlerini okumak için kullanılır. ON ve OFF yapılabilecek bileşenler (LED, Röle gibi) " ~ " olarak gösterilen PWM pinlerine bağlanır.
12	GND (Ground-Toprak) - Arduino'da devreyi topraklamak için kullanılan pindir. Kartın sağ ve sol tarafında konumlandırılarak bağlanacak bileşenlere eşirimde kolaylık sağlanır.
13	Bu butona basarak Arduino kartının resetleme işlemi yapılır. Resetleme işlemi ile kart içinde yüklü olan program yeniden başlatılır.

Arduino UNO mikrodenetleyici kartının genel özellikleri şunlardır:

- USB ile bilgisayara kolay bir şekilde bağlanabilme
- PWM gibi modları destekleme
- 16MHz'lik bir mikroişlemciye sahip olma
- Doğrudan USB porttaki enerji ile çalışabilme
- 32 KB USB belleğe sahip olma
- Arayüzünün açık kaynak kodlu olması

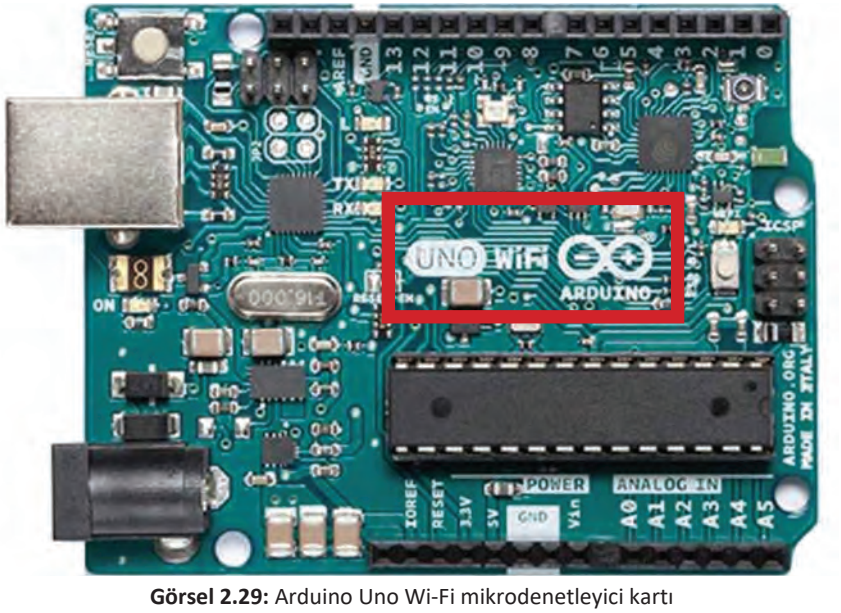
Arduino UNO mikrodenetleyici kartının üzerinde tümleşik Wi-Fi entegresi yoksa bu kartın internete erişimi bulunmaz. Arduino UNO, nesnelerin interneti gibi uygulamalarda kullanılacaksa internete bağlanması için mutlaka Wi-Fi modülüne ihtiyaç vardır. Görsel 2.28'de Seri Wi-Fi Modülü ve bağlantı uçları gösterilmiştir. Arduino UNO ile Seri Wi-Fi Modülü birbirine bağlanarak sensörlerden toplanan veriler seri Wi-Fi modülü aracılığı ile internete aktarılır.



Görsel 2.28: Seri Wi-Fi modülü ve bağlantı pinleri

2.3.1.2. Arduino UNO Wi-Fi Mikrodenetleyici Kartı

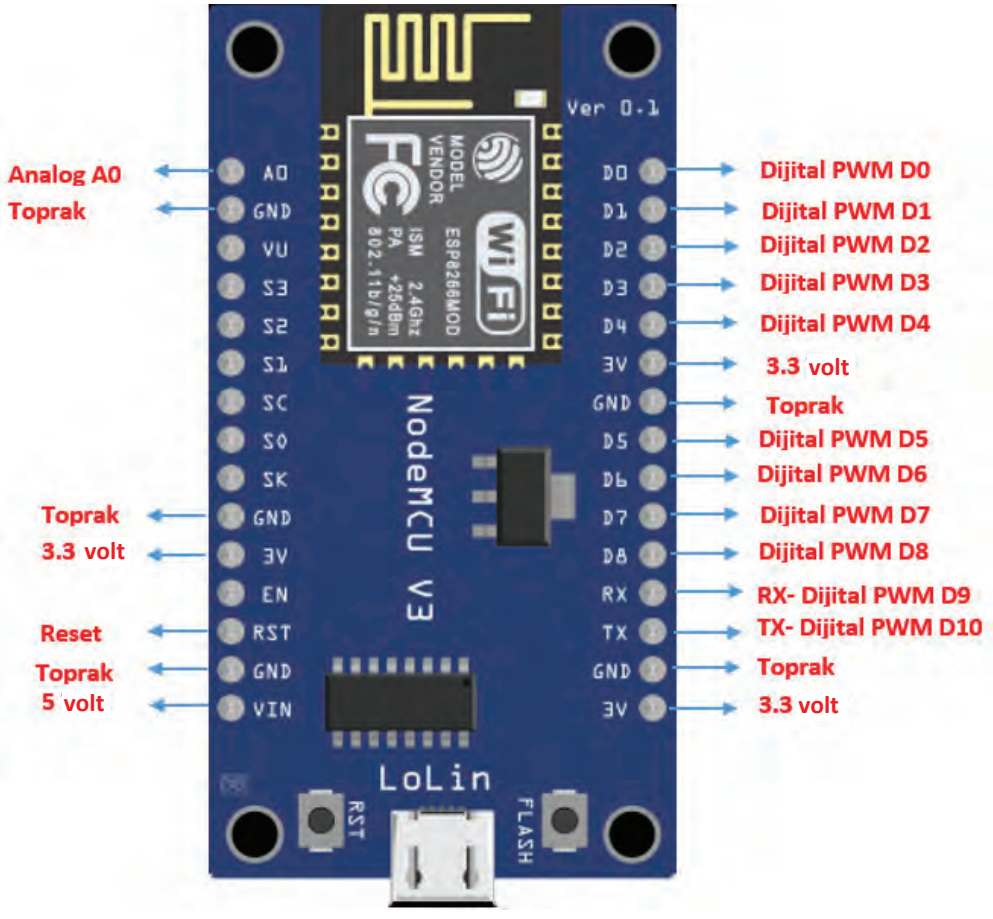
Genel olarak Arduino UNO ile aynı yapıdadır. Sadece farklı olarak üzerinde tümleşik bir Wi-Fi modülü bulunur. Entegre Wi-Fi modülü, Wi-Fi ağına erişim sağlayabilen entegre TCP/IP protokolünü destekleyen bağımsız bir erişim noktası görevi görür. Görsel 2.29'da Arduino UNO Wi-Fi Modülü gösterilmiştir. Bu mikrodenetleyici kartın Wi-Fi modülü sayesinde internete erişimi bulunur.



Görsel 2.29: Arduino Uno Wi-Fi mikrodenetleyici kartı

2.3.1.3. NodeMCU Mikrodenetleyici Kart

NodeMCU (Node MicroController Unit), nesnelerin interneti tabanlı uygulamalarda nesneleri birbiri ile bağlayan ve haberleştiren, Wi-Fi protokolünü kullanarak veri aktarımına izin verebilen ESP8266 tabanlı mikrodenetleyici devre kartıdır. NodeMCU, üzerinde GPIO, PWM, ADC gibi önemli özelliklerinden bazıları bulunmasından dolayı, nesnelerin interneti gibi uygulamalarda ve geliştirilen projelerde oldukça sık kullanılır. Görsel 2.30'da NodeMCU mikrodenetleyici devre kartı yapısı ve bağlantı pinleri gösterilmiştir.



Görsel 2.30: ESP8266 tabanlı NodeMCU mikrodenetleyici devre kartı yapısı ve pinleri

NodeMCU mikrodenetleyici kartın kullanımı kolay ve Arduino IDE yazılım dili ile programlanabilme özelliği mevcuttur. USB kablosu ile bilgisayara kolaylıkla bağlanabildiği için nesnelerin interneti uygulamalarında sıklıkla tercih edilir. NodeMCU mikrodenetleyici kartın özellikleri Tablo 2.2'de gösterilmiştir.

Tablo 2.2: NodeMCU Mikrodenetleyici Kartının Özellikleri

İşlemci Hızı	80 Mhz
USB Bağlantı Tipi	Micro USB
Çalışma Gerilimi	3.3 volt
Çıkış Gerilimi	4.5 volt – 10 volt
Flaş Bellek / SRAM	4 MB / 64 KB
Dijital Giriş Çıkış Pinleri	10 Adet
Analog Pin	1 Adet
Dahili Wi-Fi	802.11 b/g/n

2.3.1.4. ESP32 Mikrodenetleyici Kart

ESP32, üzerinde entegre Wi-Fi ve Bluetooth entegrelere sahip düşük enerji tüketen bir mikrodenetleyicidir. ESP32 esasında bir çip olup üretici tarafından bu çipi kullanan geliştirme kartlarına "ESP32"adı verilmiştir.

Geliştirme stratejisine bağlı olarak daha hızlı bir mikroişlemci, bellek, Wi-Fi ve bluetooth özellikleri ile ESP8266'nın yerine geliştirilmiştir. Standart modellerinde sıcaklık, dokunma sensörü ve hall etkisi (manyetik alanın ölçülmesi) sensörü gibi dâhilî sensörler bulunur. Standart bir ESP32 kartının sahip olduğu özellikler Tablo 2.3'te belirtilmiştir. Görsel 2.31'de ESP32 mikrodenetleyici kartının yapısı ve pinleri gösterilmiştir.

Tablo 2.3: ESP32 Mikrodenetleyici Kartının Özellikleri

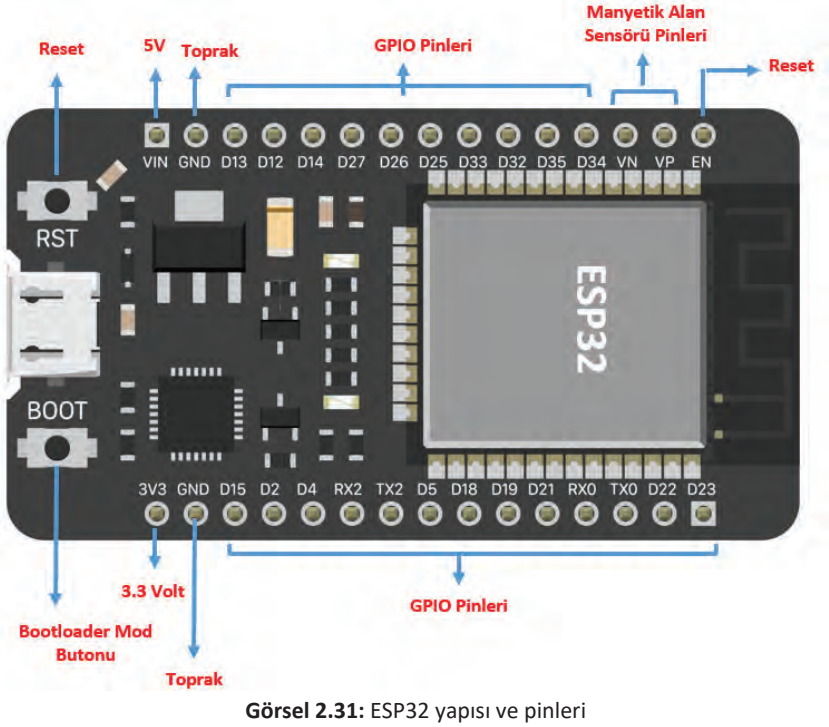
İşlemci Çekirdeği	2
İşlemci Hızı	240MHz
İşlemci Mimarisi	32 bits
Wi-Fi	IEEE802.11 b/g/n
Bluetooth	Mevcut
Flaş / RAM	16MB / 520KB
GPIO Pin	22

Farklı üreticiler tarafından üretilen birçok marka ve model ESP32 mikrodenetleyici geliştirme kartları bulunur. Geliştirilecek projeye göre ESP32 kartı seçilmelidir. Kart seçilirken dikkat edilmesi gereken özellikler aşağıdaki gibidir:

- GPIO sayıları (Her bir GPIO pini üç türlü giriş ve çıkışı destekler. Bunlar analog, dijital ve dâhilî sensör verileridir.)
- Wi-Fi anteni
- Üzerindeki dâhilî sensörler
- USB bellek miktarı

ESP32 sahip olduğu özellikler sayesinde nesnelerin interneti uygulamalarında sıklıkla kullanılır. ESP32'nin nesnelerin interneti alanındaki kullanım alanları şunlardır:

1. **Ağ İletişimi:** Cihazların yönlendiricilere bağlanması ve veri iletişimi yapmaları
2. **Noktandan Noktaya Bağlantı:** ESP32'ler ve farklı cihazlar arasında doğrudan iletişim kurma
3. **Web Sunucusu:** Sınırlı erişim imkânı ile olsa da web sunucusu olarak hizmet verme
4. **Veri İşleme:** Analog veya dijital sensörlerden gelen verileri işleme ve hesaplama



Görsel 2.31: ESP32 yapısı ve pinleri

2.4. SENSÖRLER

Sensörler; robotlarda mikrodenetleyicinin ısı, ışık ve ses gibi dışarıdan gelen fiziksel olayları algılayıp yorumlamasına yardımcı olan bileşenlerdir.

2.4.1. Çıkış Türüne Göre Sensörler

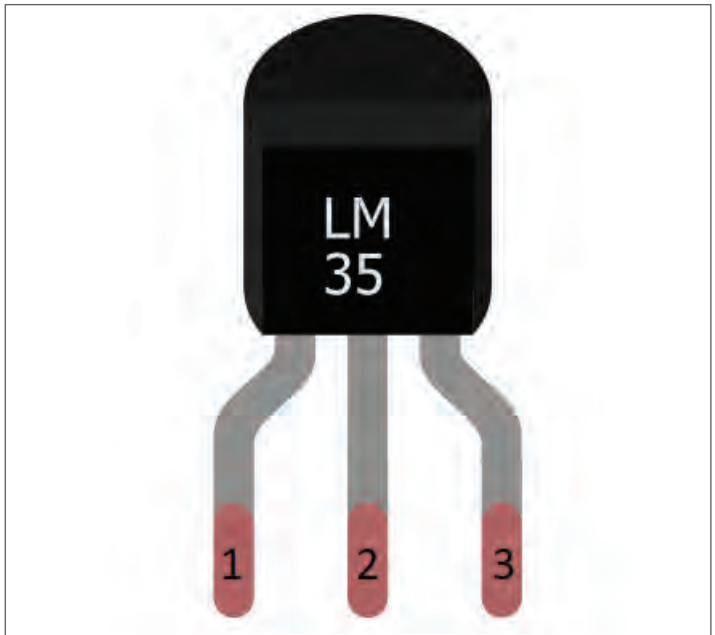
Çıkış türüne göre sensörler üç grupta incelenir.

- **Analog Sensörler:** Dış ortamdaki ışık, sıcaklık, nem, ses, rüzgâr şiddeti gibi değerleri sürekli okuyup bu bilgileri belirli değerler arasında gerilim değeri olarak geri döndüren yapılardır.
- **Dijital Sensörler:** Dış ortamdaki ışık, sıcaklık, nem, ses, rüzgâr şiddeti gibi değerleri sürekli okuyup bu bilgileri ikilik sayı sisteminde geri döndüren yapılardır.
- **Hem Analog Hem Dijital Sensörler:** Dış ortamdan okuduğu değerleri gerilim olarak 1 pin üzerinden gösterirken aynı zamanda belirlenen bir eşik değerine göre yapılan ölçüm, eşik değerinin yukarısında yer alıyorsa lojik 0; yapılan ölçüm, eşik değerinin altında yer alıyorsa lojik 1 olarak geri döndüren yapılardır.

2.4.1.1. Sıcaklık Sensörü

Dış ortamdaki sıcaklık değerini ölçüp analog veya dijital olarak çıkış veren elemanlardır.

Analog çıkış veren sensörlere örnek olabilecek en belirgin sensör LM35'tir. LM35 sensörünün dokümanı incelendiğinde bu sensörün 4-30 volt arasında gerilim değerlerinde çalışabildiği ve 0 ile 150 ^{o}C arasında sıcaklık değerini ölçebildiği gözlemlenir. LM35, ölçtüğü sıcaklık değerini her 1 ^{o}C başına 10 milivoltluk artış olarak sensör çıkışında gösterir. Sensörün görünüşü ve pin yapısı Görsel 2.32'de verilmiştir.



1. 4-30 volt besleme gerilimi
2. Analog çıkış
3. GND toprak hattı

Görsel 2.32: LM35 sıcaklık sensörü ve pin yapısı

3. UYGULAMA

A hastanesine yeni çıkan grip virüsü için aşı getirtilmiştir. Bu aşının saklanabilmesi için kritik sıcaklık değerleri bulunmaktadır. Buna göre aşı için aşağıdaki şu özellikler sağlanmalıdır:

- Aşı 10 ^{o}C ve altında saklanmalıdır.
- Aşı 11 ^{o}C ve 25 ^{o}C arasında uygulanabilir.
- 26 ^{o}C ve üzerinde aşı bozulmaktadır.

Aşı ile ilgili yukarıdaki koşulları kontrol edebilen bir sistem tasarlanmak istenmektedir. Bu çalışma için LM35 sensörünü kullanınız.

Gerekli Malzemeler

- 1 adet mikrodenetleyicili uygulama kartı
- 1 adet LM35
- 1 adet breadbord
- Bağlantı kabloları
- 1 adet RGB LED
- 3 adet 220 Ω direnç (kırmızı, kırmızı, kahverengi)

1. Adım : Görsel 2.33'teki programı yazıp mikrodenetleyicili uygulama kartına yükleyiniz.

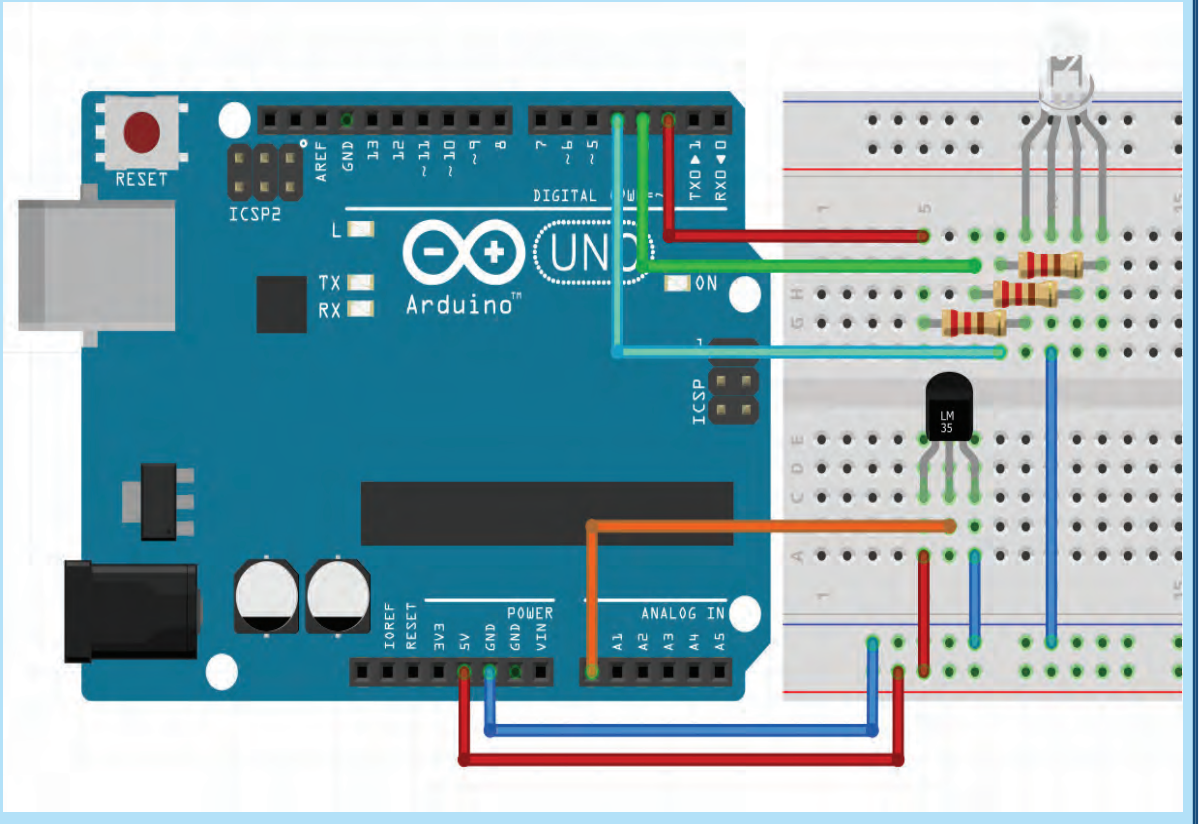
```
const int kled = 2;
const int yled = 3;
const int mled = 4;

float ogdeger;
float osensdeger;
float osdeger;

void setup()
{
  pinMode(kled, OUTPUT);
  pinMode(yled, OUTPUT);
  pinMode(mled, OUTPUT);
  digitalWrite(kled, 0);
  digitalWrite(yled, 0);
  digitalWrite(mled, 0);
}

void loop()
{
  osensdeger = analogRead(A0);
  ogdeger = (osensdeger / 1023)*5000;
  osdeger = ogdeger / 10.0;
  if (osdeger <= 10)
  {
    digitalWrite(kled, 0);
    digitalWrite(yled, 0);
    digitalWrite(mled, 1);
  }
  else if (osdeger <= 25)
  {
    digitalWrite(kled, 1);
    digitalWrite(yled, 1);
    digitalWrite(mled, 0);
  }
  else
  {
    digitalWrite(kled, 1);
    digitalWrite(yled, 0);
    digitalWrite(mled, 0);
  }
}
```

Görsel 2.33: LM35 sıcaklık sensörü için hazırlanmış program

NOT

Program incelendiğinde analog dijital çevirme işleminde 0 volt karşılığı, 0 değeridir. 5 voltun karşılığı ise 1023 değeridir. Bu nedenle analogRead ile okunan değerin gerilime çevrilmesi gerekir. Bu gerilim değeri için sensörden okunan değer 1023'e bölünüp 5000 ile çarpılır. Bu işlem sonucunda sensörden okunan değerin milivolt cinsinden karşılığı bulunur. LM35 sensörünün her 1 ^{o}C'de 10 milivolt değer artışı verdiği bilinmektedir. Bu nedenle okunan gerilim değeri 10'a bölünerek okunan sıcaklık değeri elde edilir.

2. Adım : Görsel 2.34'teki devre şemasını hazırlayıp LM35 sensörü üzerine ısıtma ve soğutma işlemi uygulayarak çalışmanın sonuçlarını gözlemleyiniz.

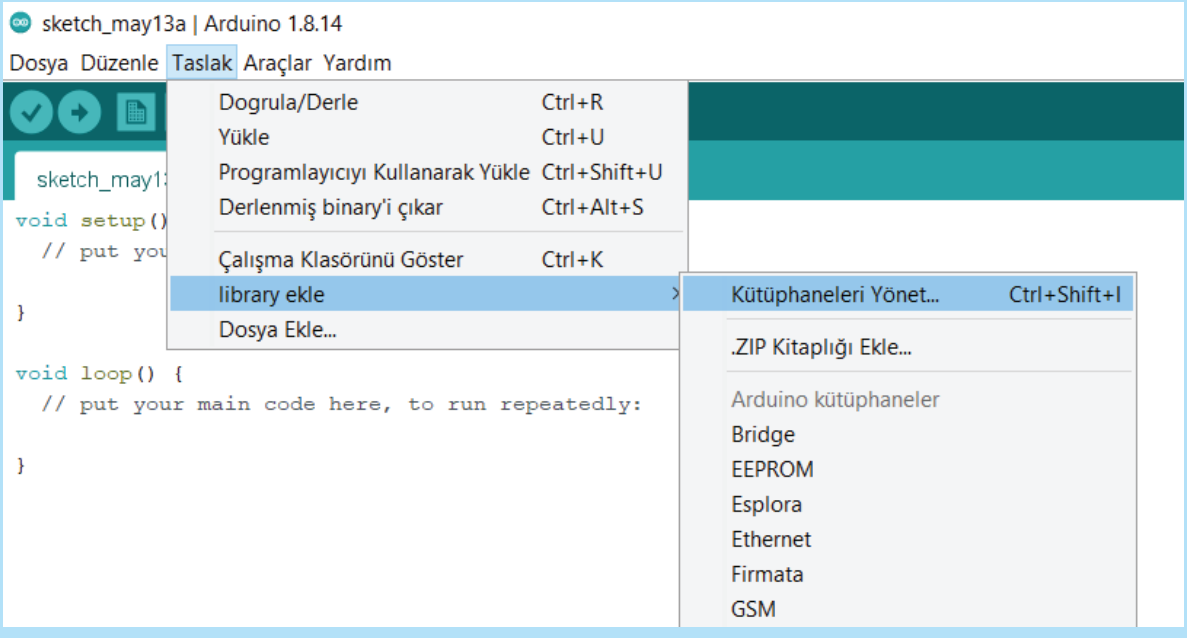


Görsel 2.34: LM35 sıcaklık sensörünün mikrodenetleyicili uygulama kartı ile kullanımı

Dijital çıkış veren sensörlere DHT-11 ve DS18B20 gibi sensörler örnek olarak gösterilebilir. Bu tip sensörler ölçme işlemini gerçekleştirip kendi içinde ölçülen değeri dijital sayıya çevirir. Bu işlemin ardından oluşan dijital sayıyı okunması için kendi içindeki kaydedicilerde saklar. Kullanıcı, 1-Wire seri haberleşme protokolü ile sensörlerin kaydedicisindeki dijital veriyi okuyup değerlendirebilir.

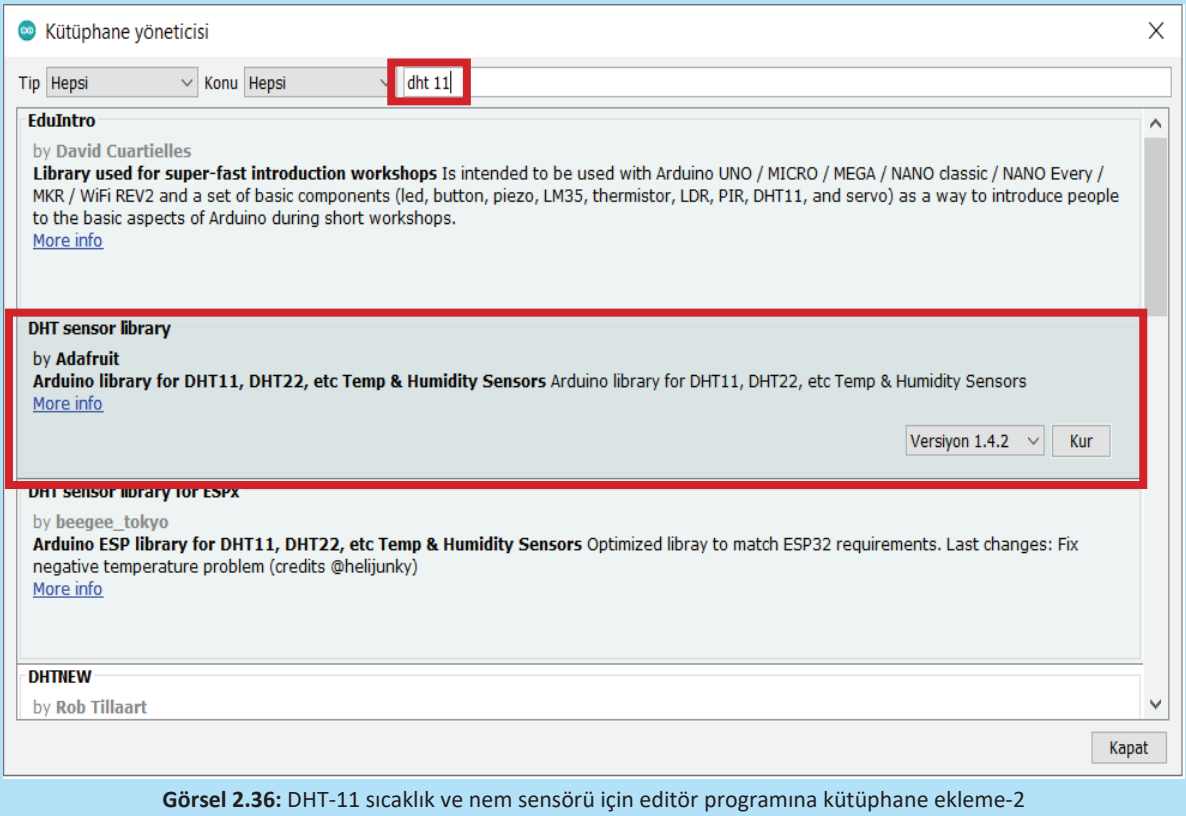
Bu çalışmada hem 3,3 voltta hem de 5 voltta çalışabilmesinden ve nem bilgisini aktarabilmesinden DHT-11 sensörü incelenecektir. Görsel 2.33'teki devre şemasına göre DHT-11 sensörünün sinyal bacağı 4,7 KΩ'luk direnç ile 5 volt besleme hattına bağlanmalıdır. Tek bir kablo üzerinden hem veri gönderilir hem de veri alınır. Tek hat üzerinden veriler bit bit iletildiği için verilerin birleştirilmesi ve veri transfer hızı önemlidir. DHT-11 için kütüphaneye ekleme yapılmalıdır.

Bunun için program editöründe **Taslak>library ekle>Kütüphaneleri Yönet** yolu izlenir (Görsel 2.35).



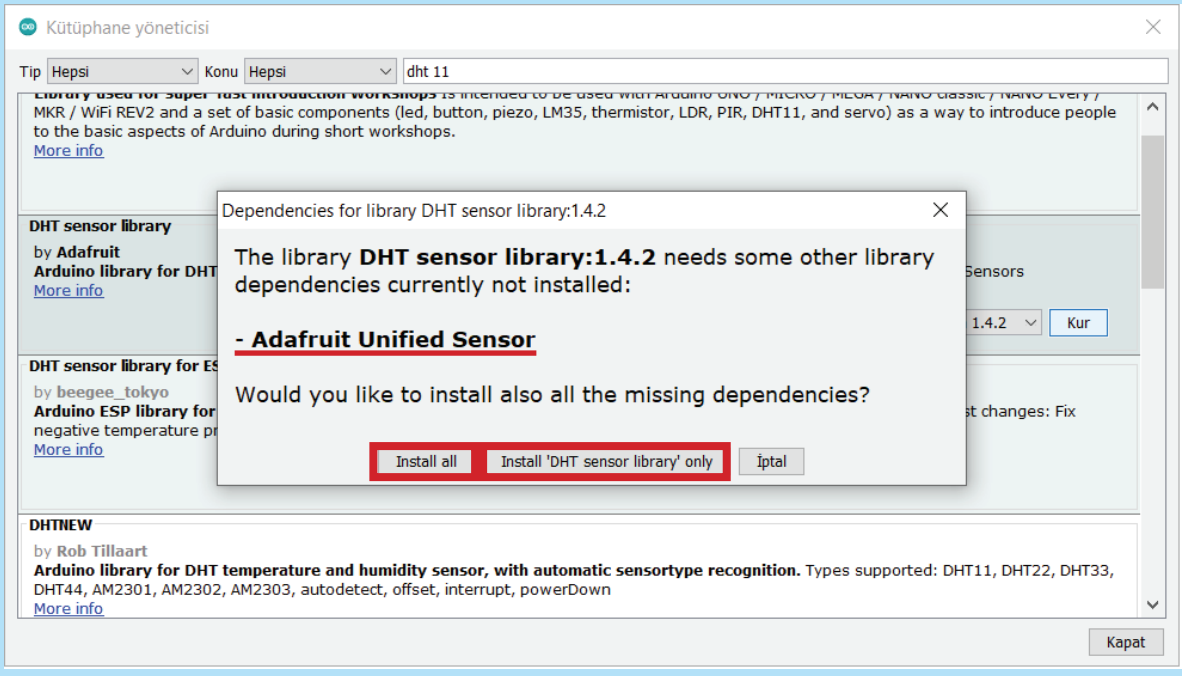
Görsel 2.35: DHT-11 sıcaklık ve nem sensörü için editör programına kütüphane ekleme-1

Açılan ekranda arama bölümüne dht 11 yazıp güncel olan kütüphaneyi Kur butonuna basarak indirebilirsiniz (Görsel 2.36).



Görsel 2.36: DHT-11 sıcaklık ve nem sensörü için editör programına kütüphane ekleme-2

Program, kurulum işlemine geçmeden önce kullanıcıya ek kütüphane önerebilir. **Install all**'a tıklanırsa kırmızı çizgi ile çizilmiş kütüphaneyi de diğer kütüphanenin yanında kurar. **Install 'DHT sensor library' only** seçeneğinde ise sadece DHT-11'in kütüphanesini kurar (Görsel 2.37).



Görsel 2.37: DHT-11 sıcaklık ve nem sensörü için editör programına kütüphane ekleme-3

Kütüphane eklendikten sonra Taslak>library ekle kısmına gelindiğinde en altta 'DHT sensor library' bölümü görülür. Bu bölüme tıklandığında program editöründe ilk iki satıra kütüphane uzantıları eklenir. Artık DHT-11 kullanıma hazırdır.

4. UYGULAMA

A hastanesine yeni çıkan grip virüsü için aşı getirtilmiştir. Bu aşının saklanabilmesi için kritik sıcaklık değerleri bulunmaktadır. Buna göre aşı için aşağıdaki şu özellikler sağlanmalıdır:

- Aşı 10 ^{o}C ve altında saklanmalıdır.
- Aşı 11 ^{o}C ve 25 ^{o}C arasında uygulanabilir.
- 26 ^{o}C ve üzerinde aşı bozulmaktadır.

Aşı ile ilgili yukarıdaki koşulları kontrol edebilen bir sistem tasarlanmak istenmektedir. Bu çalışma için DHT-11 sensörünü kullanınız.

Gerekli Malzemeler

- 1 adet mikrodenetleyicili uygulama kartı
- 1 adet DHT-11
- 1 adet breadbord
- Bağlantı kabloları
- 1 adet RGB LED
- 3 adet 220 Ω direnç (kırmızı, kırmızı, kahverengi)

1. Adım : Görsel 2.38'deki programı yazıp mikrodenetleyicili uygulama kartına yükleyiniz.

```
#include <DHT.h>
#include <DHT_U.h>

const int DHTPin = 5;
const int kled = 4;
const int yled = 3;
const int mled = 2;
float nem;
float sicaklik;

DHT dht(DHTPin, DHT11);

void setup()
{
  dht.begin();
}

void loop()
{
  delay(2000);
  nem = dht.readHumidity();
  sicaklik = dht.readTemperature();
  if (int(sicaklik) <= 10)
  {
    digitalWrite(kled, 0);
    digitalWrite(yled, 0);
    digitalWrite(mled, 1);
  }
  else if (int(sicaklik) <= 25)
  {
    digitalWrite(kled, 1);
    digitalWrite(yled, 1);
    digitalWrite(mled, 0);
  }
  else
  {
    digitalWrite(kled, 1);
    digitalWrite(yled, 0);
    digitalWrite(mled, 0);
  }
}
```

Görsel 2.38: DHT-11 sıcaklık ve nem sensörü için hazırlanmış program

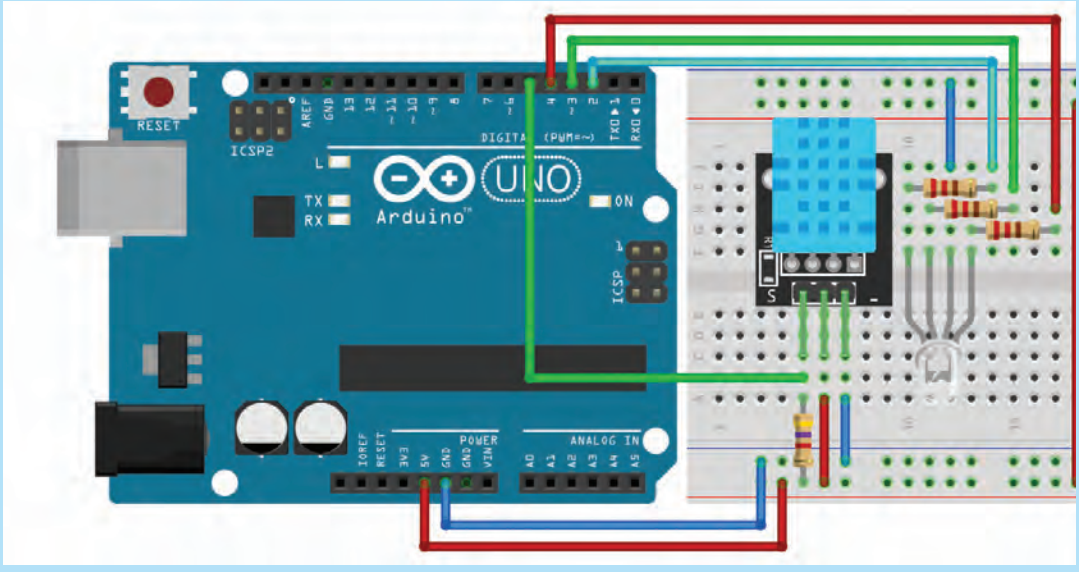
DHT-11 sensörü 0-50 ^{o}C arasında ölçme işlemi yapabilir. Aşağıdaki kodlar incelendiğinde şu bilgilere ulaşılır:

DHT dht(DHTPin, DHT11);
// Sensörün hangi pinde bağlı olduğu ve hangi modelinin kullanıldığı bilgisini aktarır.

nem = dht.readHumidity();
// Ölçme işleminin ardından sensörden nem değeri okunur.

sicaklik = dht.readTemperature();
// Ölçme işleminin ardından sensörden sıcaklık değeri okunur.

2. Adım : Görsel 2.39'daki devre şemasını hazırlayıp DHT-11 sensörü üzerine ısıtma ve soğutma işlemi uygulayarak çalışma sonuçlarını gözlemleyiniz.



Görsel 2.39: DHT-11 sıcaklık ve nem sensörünün mikrodenetleyicili uygulama kartı ile kullanımı

SIRA SİZDE

Antalya'da seracılık yapan Ferhat, kapya biber yetiştirmektedir. Kapya biber yetiştirilmesine uygunluk gösteren bir sistem tasarlanmasını istemektedir. Kapya biberin yetiştirilmesinde şu hususlara dikkat edilmelidir:

- Kapya biber, 20 ^{o}C altındaki sıcaklıklarda nem miktarı ne olursa olsun bozulur.
- 20 ^{o}C ile 30 ^{o}C arasındaki sıcaklıklarda, nem oranı %50 seviyesinde kapya biber için ideal yetiştirme ortamıdır. Ayrıca bu sıcaklıklar arasında nem %80'nin üzerine çıktığında çiftçi uyarılmalıdır.
- 30 ^{o}C ile 35 ^{o}C arasındaki sıcaklıklar kapya biber için kritiktir. Çiftçi bu konuda uyarılmalıdır.

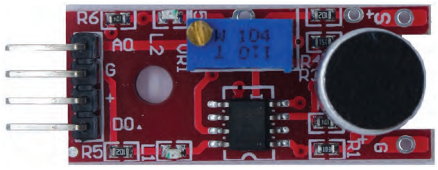
Bu sistem için DHT-11 sensörü kullanılarak çiftçiye hem sesli hem de kırmızı (sıcaklık yüksek), sarı (nem yüksek) ve mavi (biber yetiştirimine uygun) ışıklarla uyarı verilecektir. Sesli uyarılarda sürekli ses (sıcaklık yüksek) ve kesik kesik ses (nem yüksek) uygulanacaktır. Uyarılara dikkat ederek istenen sistemi tasarlayınız.

2.4.1.2. Ses Sensörü (Mikrofon)

Mikrofonda kapasitör özelliği gösteren birbirine yakın iki plaka vardır. Dış ortama yakın olan plaka, esnek yapıdadır ve ses uygulandıkça titreşir. Bu plaka, titreşim sonucu arkadaki sabit plakaya yakınlaşıp uzaklaşır. Bu titreşime göre sensörün çıkışında bir gerilim oluşur ve bu gerilim sese göre dalgalanır. Bu sensör modülü hem analog hem de dijital çıkış verebilir. Modülün dijital çıkışı modül üzerinde yer alan potansiyometre ile belirlenen bir eşik değerinin üzerindeyse lojik 0, belirlenen eşik değerinin altındaysa lojik 1 olur.

Görsel 2.40 incelendiğinde şu bilgilere ulaşılır:

- D0 yazan pin, dijital çıkıştır.
- + yazan pin, 5 volt beslemedir.
- G yazan pin, GND (toprak) beslemedir.
- A0 yazan pin, analog çıkıştır.



Görsel 2.40: Ses sensörü ve pin yapısı

Ses sensörü, istenen ya da mikrodenetleyicili kartta boş kalan pin durumuna göre D0 (dijital) veya A0 (analog) olarak kullanılabilir.

5. UYGULAMA

Görsel 2.40'taki ses sensörü ile alkış sesi uygulandığında mikrodenetleyicili uygulama kartı üzerindeki 13 numaralı pine bağlı L LED'ini 1 saniye yakan çalışmayı hem sensörün dijital çıkışını hem de sensörün analog çıkışını kullanarak uygulayınız.

Gerekli Malzemeler

- 1 adet mikrodenetleyicili uygulama kartı
- 1 adet ses sensörü modülü
- 1 adet breadbord
- Bağlantı kabloları

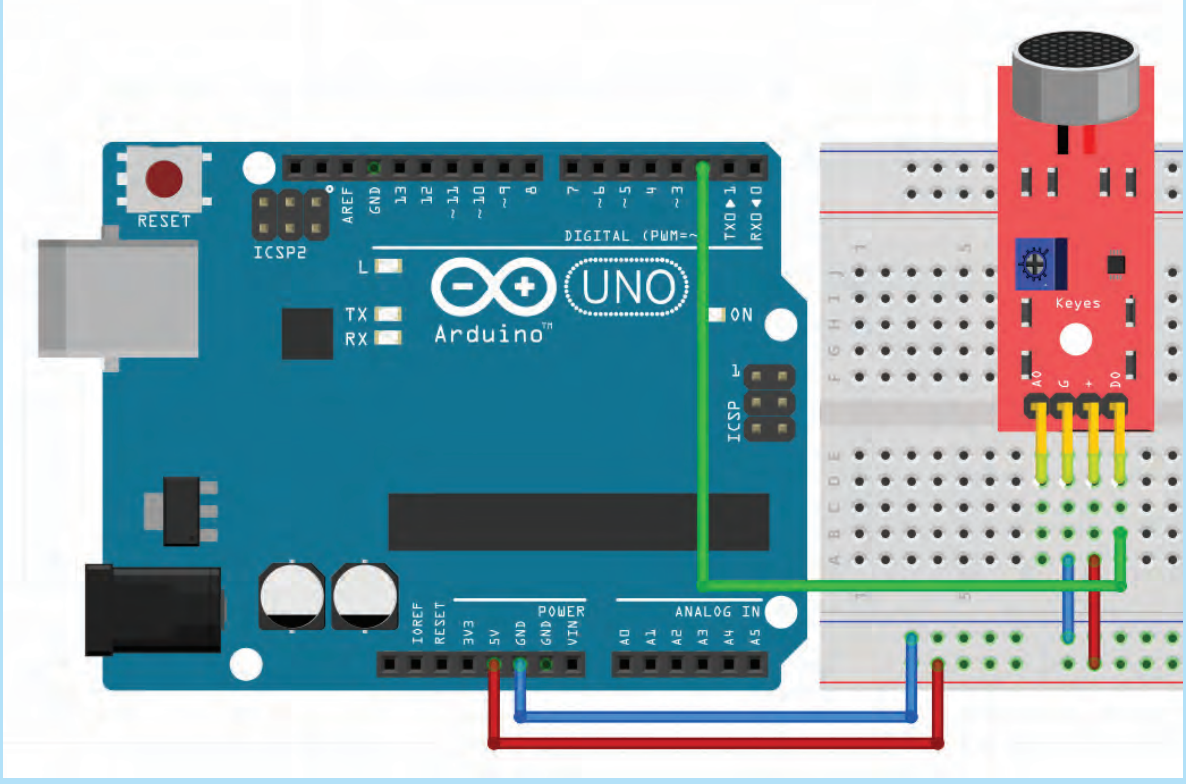
» **Digital Yöntem**

1. Adım : Sensörün eşik değerini ayarlayınız. Bu ayarlama için sensörünüzün (+) pinini +5 volt besleme hattına, G pinini GND besleme hattına bağlayınız. Bir tornavida yardımı ile potansiyometreyi saat yönünün tersine (sola) L2 LED'i sönünceye kadar çeviriniz. L2 LED'i söndükten sonra alkış sesinin şiddetine göre saat yönünün tersine ya da saat yönünde çevirerek eşik değerini tanımlayınız. Bu tanımlama sonrasında sensör kullanıma hazırdır.

2. Adım : Görsel 2.41.a'daki programı yazıp mikrodenetleyicili karta yükleyiniz. Devre şeması hazırlanıp alkış sesi uygulandığında mikrodenetleyicili uygulama kartı üzerindeki L LED'i yanacak ve 1 saniye sonra sönecektir.

```
1   const int sesDigi = 2;
2   const int sesLed = 13;
3
4   void setup()
5   {
6     pinMode(sesDigi, INPUT);
7     pinMode(sesLed, OUTPUT);
8     digitalWrite(sesLed,0);
9   }
10
11  void loop()
12  {
13    if(digitalRead(sesDigi)==1)
14    {
15      digitalWrite(sesLed, 1);
16      delay(1000);
17    }
18    digitalWrite(sesLed, 0);
19  }
```

Görsel 2.41.a: Ses sensörü için hazırlanmış program



Görsel 2.41.b: Ses sensörünün dijital olarak mikrodenetleyicili uygulama kartı ile kullanımı

» **Analog Yöntem**

1. Adım : Modül üzerinde yer alan A0 bağlantısı, karşılaştırma devresini devre dışı bırakıp direkt sensörün ilgili bacağından analog bilgi okumayı sağlar. Bu yöntemde öncelikle sensörü mikrodenetleyicili uygulama kartı ile görseldeki gibi bağlayıp sensörden seri bilgi okunmalıdır (Görsel 2.41.b). Alkış sesi uygulandığında seri monitörden okunan veri seviyesi eşik değeri olarak belirlenirse programda bu eşik seviyesi ve altı seste LED yanar. Bu yöntemle ölçme işlemi yapıldığında eşik değeri 470 ile 570 arasındadır. Bu değer, ölçümlerinizde ortam gürültüsüne göre farklı çıkabilir. Sensörün ölçümünün ortam gürültüsünden etkilenmemesi için alt değere yakın tercih yapınız.

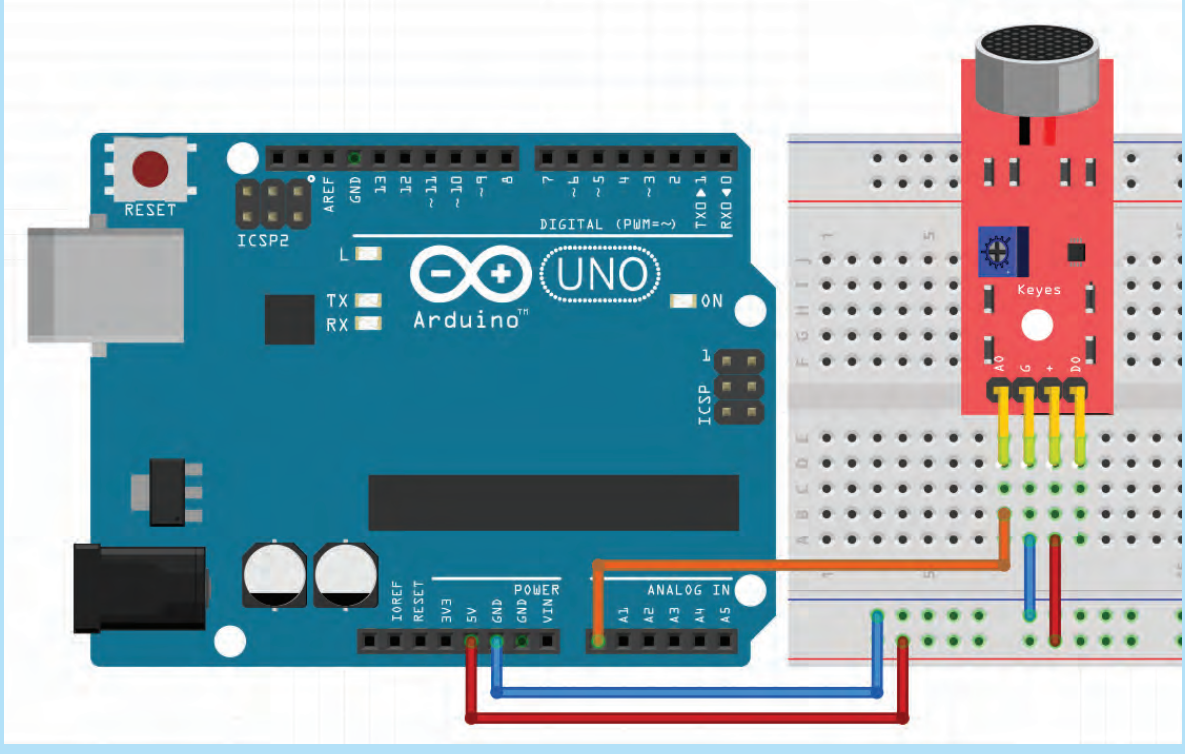
NOT

Eşik değerini belirleyebilmek için seri monitörden sürekli bilgi okuma programı mikrodenetleyicili karta yüklenirse kart bilgisayara her bağlandığında içindeki program gereği kesintisiz olarak bilgisayara veri gönderir. Bu nedenle sürekli seri porta veri olduğu için tekrardan farklı bir program yüklenemez. Bu durumun önüne geçebilmek, eşik değerini belirleyebilmek için bir şarta bağlı seri bilgi okuma işlemi yapılması önerilir. Örneğin bir pine buton bağlanırsa bu butona basılı olduğu sürece seri bilgi gönderilebilir. Bir diğer yöntemde ise veri gönderme işlemi arasına 1 saniyelik gecikme eklenebilir.

2. Adım : Görsel 2.42.a'daki programı yazıp mikrodenetleyicili karta yükleyiniz. Devre şeması hazırlanıp (Görsel 2.42.b) alkış sesi uygulandığında mikrodenetleyicili uygulama kartı üzerindeki L LED'i yanacak ve 1 saniye sonra sönecektir.

```
1  const int sesLed = 13;
2  const int sesAna = A0;
3
4  void setup()
5  {
6    pinMode(sesAna, INPUT);
7    pinMode(sesLed, OUTPUT);
8    digitalWrite(sesLed,0);
9  }
10 void loop()
11 {
12   if(analogRead(sesAna)<=490)
13   {
14     digitalWrite(sesLed, 1);
15     delay(1000);
16   }
17   digitalWrite(sesLed, 0);
18 }
```

Görsel 2.42.a: Ses sensörü için hazırlanmış program



Görsel 2.42.b: Ses sensörünün analog olarak mikrodenetleyicili uygulama kartı ile kullanımı

SIRA SİZDE

Uyumadan önce düzenli şekilde kitap okuyan Mustafa, LED aydınlatmasını alkış sesiyle kontrol etmek istemektedir.

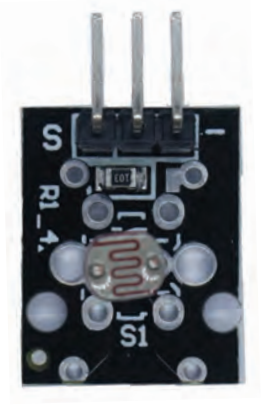
Bu sistemde ses sensörü ile alkış sesi uygulandığında LED aydınlatmayı yakan, tekrar alkış sesi uygulandığında LED aydınlatmayı söndüren çalışmayı hem dijital çıkış hem de analog çıkış için hazırlayınız.

NOT

LED aydınlatma sistemi yüksek gerilim ve akımla çalıştığı için ya transistörlü sistemle ya da röleli sistemle kullanılmalıdır.

2.4.1.3. Işık Seviye Sensörü

Işık seviye sensörü, üzerine düşen ışık şiddetine göre direnç değeri değişen sensördür. Genellikle ışık şiddetini ölçmek için LDR (Light Dependent Resistor) sensörü kullanılır. Bu sensörün üzerine düşen ışık şiddeti arttıkça direnç değeri azalır ancak LDR bu işlemi doğrusal gerçekleştirmediği için çok sağlıklı sonuçlar elde edilemez. Bu nedenle LDR daha çok ışık var-yok veya daha basit seviyelerde kullanılır. Hassas çalışmalar için bu alanda daha gelişmiş sensörler tercih edilebilir. LDR sensörü kullanılırken sensörün mikrodenetleyicili sisteme bağlantıları Görsel 2.43'teki gibi yapılmalıdır.



Görsel 2.43: LDR sensör modülü

SIRA SİZDE

Doktor Melike'nin evinde Hınzır adında doymak bilmeyen bir kedisi vardır. Melike nöbet günlerinde kedisinin yeterince beslenemediğini düşünmektedir. Melike, Hınzır'ın bu gibi durumlarda beslenebilmesi için bir sistem tasarlanmasını istemektedir.

Bu sistem için LED ile LDR üzerine ışık uygulanacak ve kedi LED ile LDR arasına kafasını uzattığında LDR üzerindeki ışık kesildiğinden ötürü servo motor yardımıyla bir miktar mama yemek kabına aktarılacaktır. Servo motor 0 derecelik konumundan 90 derecelik konuma dönüp haznesindeki mamanın kaba aktarılmasını sağlayacak ve 2 saniye bu konumda bekleyecektir. Servo motor 2 saniyenin sonunda tekrar 0 derecelik konumuna dönüp kapalı hâlini alacaktır. Bu sistemi tasarlayarak çalışmayı oluşturunuz.

2.4.1.4. Mesafe Sensörü

Mesafe sensörü, sensörün belirli bir nesneye olan uzaklığını ölçmeyi sağlayan yapıdır. Mesafe sensöründe genellikle ışık ve ses bilgilerinden faydalanılarak ölçme işlemi gerçekleştirilir. Işıklı olan sistemlerde lazer diyot veya infrared LED kullanılır. Sesli olan sistemlerde ise ultrasonic sesler kullanılır. Her iki yöntemde de bilgi kısa süreliğine gönderilir ve hemen kesilir. Daha sonra gönderilen bilginin karşıdan yansıması beklenir ve süre tutulur. Veri geri döndüğünde yol problemindeki **<< Yol = Hız * Zaman >>** formülü uygulanır. Veri hem giderken hem de dönerken aynı mesafeyi aldığı için yol bilgisi ikiye bölünerek sensör ile nesne arasındaki mesafe hesaplanır.

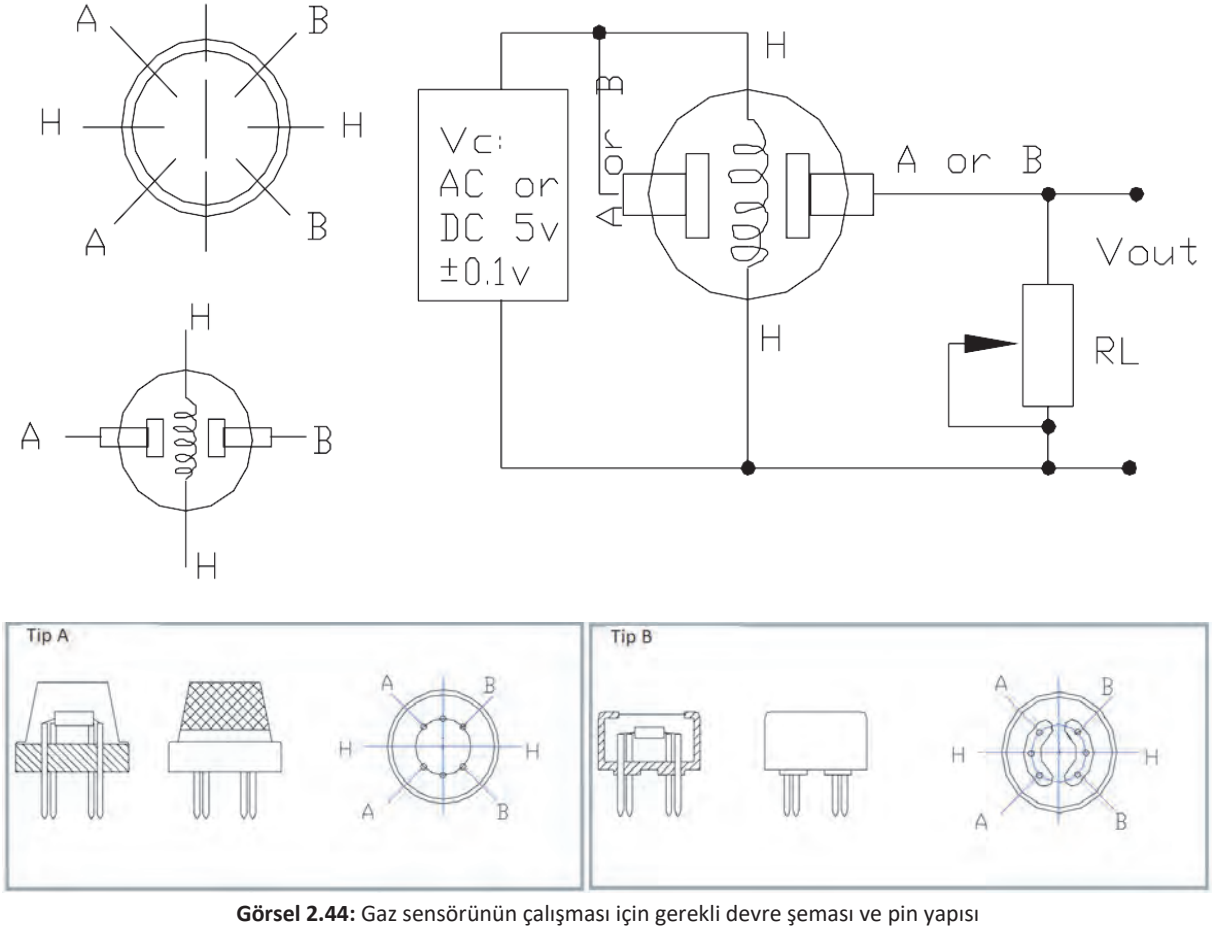
SIRA SİZDE

Bir ortaokulda görev yapan Beden Eğitimi Öğretmeni Ekrem, okuldaki bütün öğrencilerin boylarını düzenli olarak takip etmek ve uyguladığı eğitimin öğrencilerin boyu üzerindeki etkisini araştırmak istemektedir. Her öğrencinin boyunu ölçme işlemi çok zaman almaktadır. Bunun için bir sistem tasarlanmasını istemektedir.

Bu sistem için 2 metre yüksekliğinde bir sopanın ucuna HC SR04 sensörü yerleştirilerek boyu ölçülmek istenen öğrencinin yanına bu sopa dikilecektir. Sopa üzerinde yer alan ölçme işleminin başlatılması için butona basılacak ve LCD ekran üzerinde öğrencinin boyu yazacaktır. Bu sistemi tasarlayarak çalışmayı oluşturunuz.

2.4.1.5. Gaz Sensörü

Gaz sensöründe genellikle MQ serisi yaygın olarak kullanılır. Bu seride sensör bağlantısı yapılır ve sensörün ısınması beklenir. Sensör ısındıktan sonra sensörün içindeki tel hangi gaza duyarlı ise o gaz ile temas hâline geçtiği zaman direnç değerinde değişmeye yol açarak analog bilgiyi geri döndürür. Bu sensörler tek olarak satılabildiği gibi modül olarak da satılır. Sensörün modül olan versiyonunda hem analog hem de dijital çıkış bulunur ve sensör için gerekli değerlere sahip devre ile kullanıma hazırdır. Modülün dijital çıkışı modül üzerinde yer alan potansiyometre ile belirlenen bir eşik değerinin üzerindeyse lojik 0, belirlenen eşik değerinin altındaysa lojik 1 olur. Modül olmayan versiyonda ise sensörün dokümanında yer alan devre uygulanmalı ve uygun değerlerde dirençler takılmalıdır (Görsel 2.44).



Görsel 2.44: Gaz sensörünün çalışması için gerekli devre şeması ve pin yapısı

MQ-2: Sensörün üretici dokümanı incelendiğinde 300 ile 10000 ppm aralığında sigara dumanı, LPG, propan ve hidrojen gazlarının yoğunluğuna göre RL direnci olarak 2 KΩ ile 20 KΩ arasında değerleri aldığı görülür.

MQ-3: Sensörün üretici dokümanı incelendiğinde 0,04 ile 4 mg/L aralığında alkol yoğunluğuna göre 1 MΩ ile 8 MΩ arasındaki değerleri aldığı görülür. RL direnci olarak 200 KΩ tercih edilmelidir.

MQ-4: Sensörün üretici dokümanı incelendiğinde 200 ile 10000 ppm aralığında doğal gaz yoğunluğuna göre 10 KΩ ile 60 KΩ arasındaki değerleri aldığı görülür. RL direnci olarak 20 KΩ tercih edilmelidir.

MQ-5: Sensörün üretici dokümanı incelendiğinde 200 ile 10000 ppm aralığında LPG, LNG ve doğal gaz yoğunluğuna göre 10 KΩ ile 60 KΩ arasındaki değerleri aldığı görülür. RL direnci olarak 20 KΩ tercih edilmelidir.

MQ-6: Sensörün üretici dokümanı incelendiğinde 200 ile 10000 ppm aralığında LPG, LGN, izobütan ve propan gazlarının yoğunluğuna göre 10 KΩ ile 60 KΩ arasındaki değerleri aldığı görülür. RL direnci olarak 20 KΩ tercih edilmelidir.

MQ-7: Sensörün üretici dokümanı incelendiğinde 20 ile 2000 ppm aralığında karbonmonoksit gazının yoğunluğuna göre 2 KΩ ile 20 KΩ arasındaki değerleri aldığı görülür. RL direnci olarak 5 KΩ, 10 KΩ ve 47 KΩ değerlerinden biri tercih edilmelidir.

MQ-9: Sensörün üretici dokümanı incelendiğinde 100 ile 10000 ppm aralığında yanıcı gazların yoğunluğuna göre 2 KΩ ile 20 KΩ arasındaki değerleri aldığı görülür.

6. UYGULAMA

Gaz sensörü modülü ile çakmak gazı uygulandığında mikrodenetleyicili uygulama kartı üzerindeki 13 numaralı pine bağlı LED'i yakan ve gaz dağıldığında 13 numaralı pine bağlı LED'i söndüren yapıyı hem sensörün dijital çıkışını hem de sensörün analog çıkışını kullanarak uygulayınız.

Gerekli Malzemeler

- 1 adet mikrodenetleyicili uygulama kartı
- 1 adet gaz sensörü MQ-5 modülü
- 1 adet breadbord
- Bağlantı kabloları

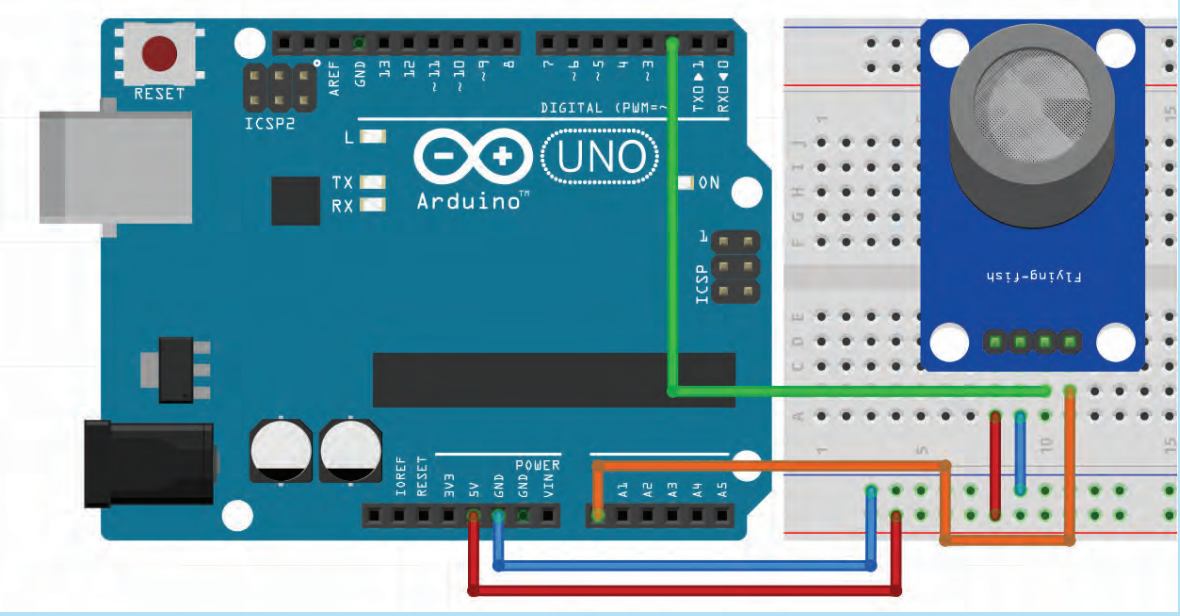
1. Adım : Dijital ayarlamaların yapılması için sensörün eşik değerini belirleyiniz. Bu ayarlama için sensörünüzün (+) pinini +5 volt besleme hattına, G pinini GND besleme hattına bağlayınız. Bir tornavida yardımı ile potansiyometreyi saat yönünün tersine (sola) L2 LED'i sönünceye kadar çeviriniz. L2 LED'i söndükten sonra çakmak gazı uygulayınız. Sensörün hassasiyetine göre saat yönünün tersine ya da saat yönünde çevirerek eşik değerini tanımlayınız. Bu tanımlama sonrası sensör kullanıma hazırdır.

2. Adım : Görsel 2.45.a'daki programı yazıp mikrodenetleyicili karta yükleyiniz. Görsel 2.45.b'deki devre şemasını hazırlayıp çakmak gazını sensöre uygulayınız. Mikrodenetleyicili uygulama kartı üzerindeki L LED'i gazı algıladığında yanacak, gaz kesildiğinde sönecektir.

```
1   const int led = 13;
2   const int gPin = 2;
3
4   void setup()
5   {
6     pinMode(gPin, INPUT);
7     pinMode(led, OUTPUT);
8     digitalWrite(led, 0);
```

```
  Serial.begin(9600);
  Serial.println("Sensörün hazırlanması bekleniyor.");
  delay(20000);
  Serial.println("Sensörün hazır.");
}

void loop()
{
  if (digitalRead(gPin) == 1)
  {
    Serial.print("Okunan Gaz: ");
    Serial.println(analogRead(A0));
    digitalWrite(led, 1);
  }
  else
  {
    digitalWrite(led, 0);
  }
}
```

Görsel 2.45.a: Gaz sensörü için hazırlanmış program



Görsel 2.45.b: Gaz sensörü modülünün hem analog hem de dijital olarak mikrodenetleyicili uygulama kartı ile kullanımı

Program ilk çalıştığında sensörün ısınabilmesi için 20 saniye bekletilir. Sensör hazır hâle gelince LPG gazı uygulanır, belirlenmiş eşik değeri aşılınca sensörün analog çıkışından mikrodenetleyicili uygulama kartına gaz değeri aktarılır. Mikrodenetleyicili uygulama kartı ise bu bilgiyi USB portu üzerinden bilgisayara aktarıp seri monitör üzerinden değeri kullanıcıya yansıtır.

SIRA SİZDE

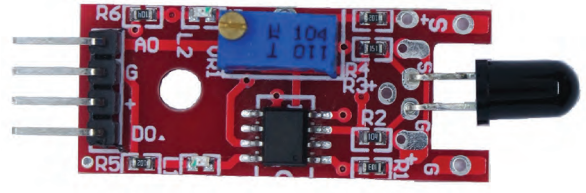
Ocakta pişen yemeğin taşmasıyla ateş sönüp gaz kesilmeden sanki ateş yanıyormuş gibi ocak gaz vermeye devam etmektedir. Bu olayların sonucu çok kötü olabilmektedir. Bunların önüne geçebilmek için doğal gaz ve LPG gazına duyarlı MQ-5 sensör kullanarak böyle durumlarda evdeki kullanıcıya hem sesli uyarı veren hem de hava tahliyesi yapan bir sistem tasarlayınız.

2.4.1.6. Alev Sensörü

Alev sensörü modülü üzerindeki infrared alıcı, 760 nm ile 1100 nm arasındaki dalga boyuna sahip ışıklara duyarlıdır. Bu sensör modülü hem analog hem de dijital çıkış verebilir. Modülün dijital çıkışı modül üzerinde yer alan potansiyometre ile belirlenen bir eşik değerinin üzerindeyse lojik 0, belirlenen eşik değerinin altındaysa lojik 1 olur. Sensör alev görmediğinde analog pininden 1024 seviyesinde, dijital pininden lojik olarak 1 bilgisi gönderir. Alev gördüğünde ise analog pininden 500 altında bir değer, dijital pininden ise lojik olarak 0 bilgisi gönderir.

Görsel 2.46 incelendiğinde şu bilgilere ulaşılır:

- D0 yazan pin, dijital çıkıştır.
- + yazan pin, 5 volt beslemedir.
- G yazan pin, GND (toprak) beslemedir.
- A0 yazan pin, analog çıkıştır.



Görsel 2.46: Alev sensör modülü

Alev sensörü, istenen ya da mikrodenetleyicili kartta boş kalan pin durumuna göre D0 (dijital) veya A0 (analog) olarak kullanılabilir.

7. UYGULAMA

Görsel 2.46'daki alev sensörü kullanılarak çakmak ile ateş yakıldığında mikrodenetleyicili uygulama kartı üzerindeki 13 numaralı pine bağlı LED'i yakan, ateş sonlandırıldığında 13 numaralı pine bağlı LED'i söndüren yapıyı hem sensörün dijital çıkışını hem de sensörün analog çıkışını kullanarak uygulayınız.

Gerekli Malzemeler

- 1 adet mikrodenetleyicili uygulama kartı
- 1 adet alev sensörü modülü
- 1 adet breadbord
- Bağlantı kabloları

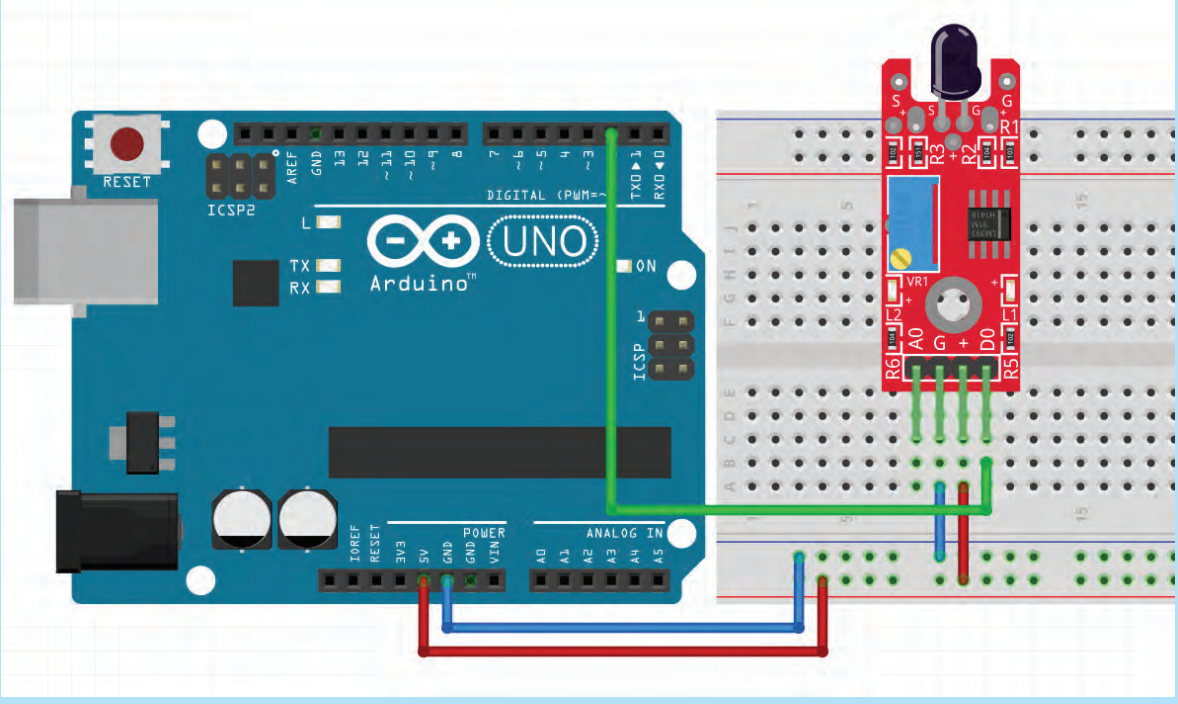
» **Digital Yöntem**

1. Adım : Sensörün eşik değerini ayarlayınız. Bu ayarlama için sensörünüzün (+) pinini +5 volt besleme hattına, G pinini GND besleme hattına bağlayınız. Bir tornavida yardımı ile potansiyometreyi saat yönünün tersine (sola) L2 LED'i sönünceye kadar çeviriniz. L2 LED'i söndükten sonra çakmakla yakılan ateşe tepkisini ölçünüz. Sensörün hassasiyetine göre saat yönünün tersine ya da saat yönünde çevirerek eşik değerini tanımlayınız. Bu tanımlama sonrası sensör kullanıma hazırdır.

2. Adım : Görsel 2.47.a'daki programı yazıp mikrodenetleyicili karta yükleyiniz. Görsel 2.47.b'deki devre şeması hazırlanıp ateşe doğru çevrildiğinde mikrodenetleyicili uygulama kartı üzerindeki L LED'i yanacak, farklı bir yöne çevrildiğinde sönecektir.

```
1   const int alevled = 13;
2   const int alevdigi = 2;
3
4   void setup()
5   {
6     pinMode(alevdigi, INPUT);
7     pinMode(alevled, OUTPUT);
8     digitalWrite(alevdigi, 0);
9   }
10
11  void loop()
12  {
13    if(digitalRead(alevdigi) == 1)
14      digitalWrite(alevled,1);
15    else
16      digitalWrite(alevled,0);
17    delay(100);
18  }
```

Görsel 2.47.a: Alev sensörü için hazırlanmış program



Görsel 2.47.b: Alev sensörü modülünün dijital olarak mikrodenetleyicili uygulama kartı ile kullanımı

» **Analog Yöntem**

1. Adım : Modül üzerinde yer alan A0 bağlantısı, karşılaştırma devresini devre dışı bırakıp direkt sensörün ilgili bacağından analog bilgi okumayı sağlar. Bu yöntemde öncelikle sensörü mikrodenetleyicili uygulama kartı ile görseldeki gibi bağlayıp sensörden seri bilgi okunmalıdır. Sensör aleve doğru çevrildiğinde seri monitörden okunan veri seviyesi eşik değeri olarak belirlenirse programda bu eşik seviyesi ve altındaki değerlerde LED çıkışı ayarlanır. Bu yöntemle ölçme işlemi yapıldığında eşik değerinin 500 ve aşağısı olduğu görülür. Bu değer, ölçümlerinizde ortam ışık seviyesine göre farklı çıkabilir. Ortam ışık seviyesi yüksekse sensörün ölçüm sonucunun ortam ışığından etkilenmemesi için 500 değerinin altında bir tercihte bulunulabilir.

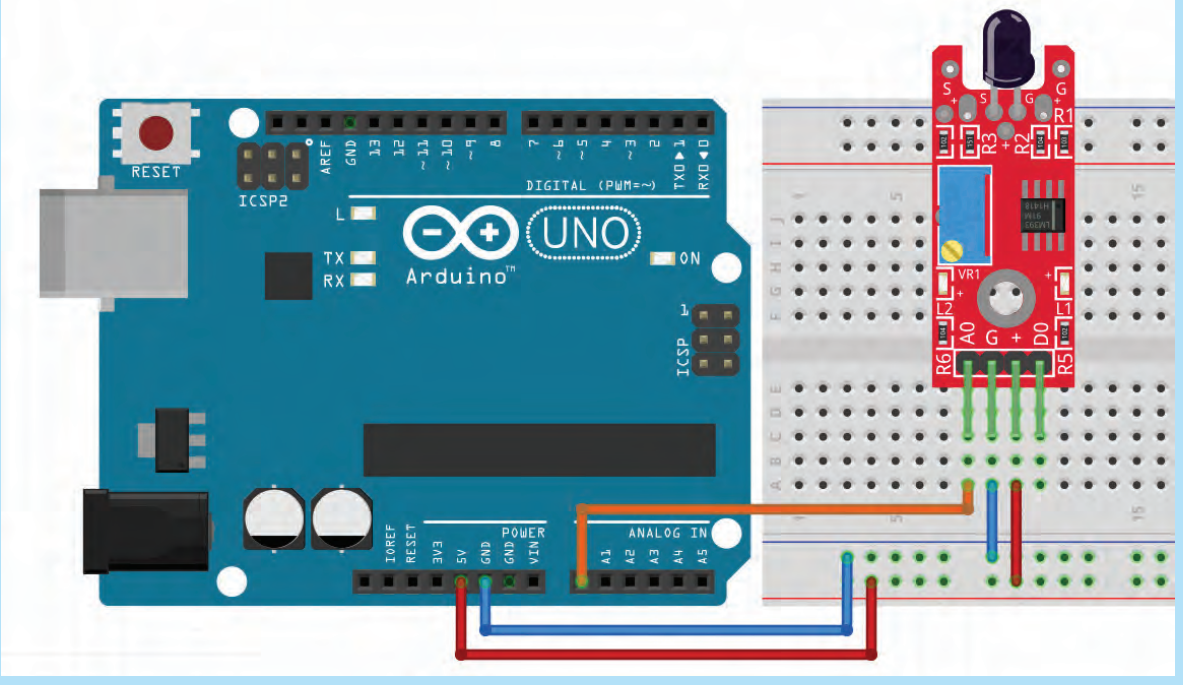
NOT

Eşik değerini belirleyebilmek için seri monitörden sürekli bilgi okuma programı mikrodenetleyicili karta yüklenirse kart bilgisayara her bağlandığında içindeki program gereği kesintisiz olarak bilgisayara veri gönderir. Bu nedenle sürekli seri portta veri olduğu için tekrardan farklı bir program yüklenemez. Bu durumun önüne geçebilmek, eşik değerini belirleyebilmek için bir şarta bağlı seri bilgi okuma işlemi yapılması önerilir. Örneğin bir pine buton bağlanırsa bu butona basılı olduğu sürece seri bilgi gönderilebilir. Bir diğer yöntemde ise veri gönderme işlemi arasına 1 saniyelik gecikme eklenebilir.

2. Adım : Görsel 2.48.a'daki programı yazıp mikrodenetleyicili karta yükleyiniz. Görsel 2.48.b'deki devre şeması hazırlanıp ateşe doğru çevrildiğinde mikrodenetleyicili uygulama kartı üzerindeki L LED'i yanacak, farklı bir yöne çevrildiğinde sönecektir.

```
1   const int alevled = 13;
2
3   void setup()
4   {
5     pinMode(alevled, OUTPUT);
6     digitalWrite(alevled, 0);
7   }
8
9   void loop()
10  {
11    if (analogRead(A0) <= 500)
12      digitalWrite(alevled, 1);
13    else
14      digitalWrite(alevled, 0);
15    delay(100);
16  }
```

Görsel 2.48.a: Alev sensörü için hazırlanmış program



Görsel 2.48.b: Alev sensörü modülünün analog olarak mikrodenetleyicili uygulama kartı ile kullanımı

SIRA SİZDE

Bir sinema salonu için yangın durumunda acil çıkış levhalarını aydınlatacak ve sesli uyarı yapacak bir sistem tasarlanmak istenmektedir.

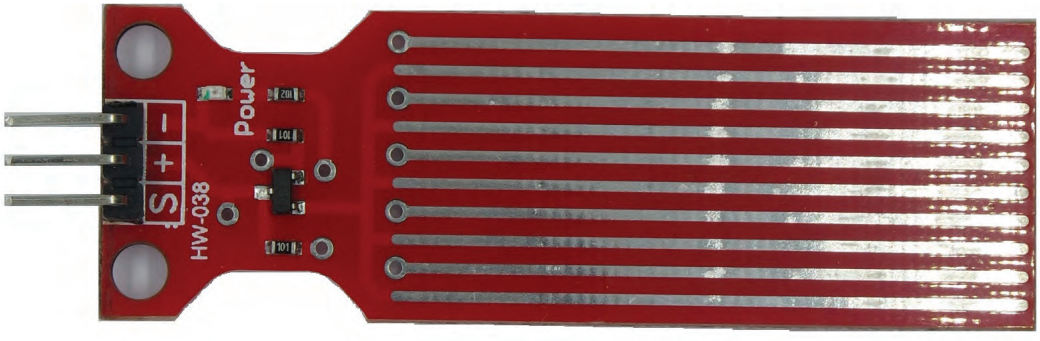
Alev sensörü kullanarak yangın tespit edildiği anda acil çıkışların LED'lerini yakacak ve buzzer ile sesli uyarı verecek bir sistem tasarlayınız. Yangın bittiğinde bu sistemi durdurmak için sisteme bir buton dâhil ediniz. Yangın bittiği zaman bu butona basıldığında uyarıları durduracak çalışmayı hem sensörün dijital çıkışı hem de sensörün analog çıkışı için hazırlayınız.

2.4.1.7. Su Taşkını Sensörü

Görsel 2.49'daki sensörün yapısı incelendiğinde su, gri renkli iki hattı kısa devre ederek transistörün sürülmesini sağlar. Sürülen transistör 0-5 volt arasında analog çıkış vererek mikrodenetleyicili sisteme sensörün okuduğu bilgiyi aktarır.

Görsel 2.49 incelendiğinde şu bilgilere ulaşılır:

- - yazan pin, GND (toprak) beslemedir.
- + yazan pin, 5 volt beslemedir.
- S yazan pin, analog çıkıştır.



Görsel 2.49: Su seviye sensörü modülü

8. UYGULAMA

Görsel 2.49'daki su seviye sensörünü kullanarak bir bardakta iletken sıvı seviyesi %10'u geçtiğinde mikrodenetleyicili uygulama kartı üzerindeki 13 numaralı pine bağlı LED'i 100 milisaniye yakan ve söndüren çalışmayı uygulayınız.

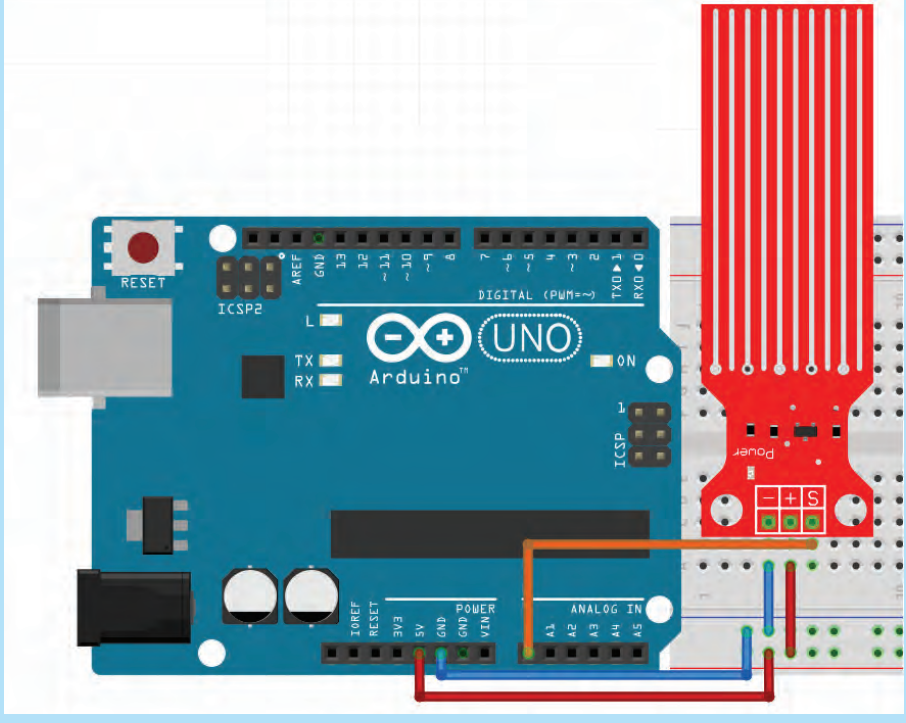
Gerekli Malzemeler

- 1 adet mikrodenetleyicili uygulama kartı
- 1 adet su seviye sensörü modülü (kırmızı olan)
- 1 adet breadbord
- Bağlantı kabloları

1. Adım : Görsel 2.50.a'daki programı mikrodenetleyicili uygulama kartına yükleyiniz.
2. Adım : Görsel 2.50.b'deki devre şemasını uygulayınız.

```
const int suSPin = A0;
const int suSLed = 13;
void setup()
{
  pinMode(suSLed, OUTPUT);
  digitalWrite(suSLed, 0);
}
void loop()
{
  if (analogRead(suSPin) > 100)
  {
    digitalWrite(suSLed, HIGH);
    delay(100);
    digitalWrite(suSLed, LOW);
    delay(100);
  }
  digitalWrite(suSLed, LOW);
}
```

Görsel 2.50.a: Su seviye sensörü için hazırlanmış program



Görsel 2.50.b: Su seviye sensörü modülünün mikrodenetleyicili uygulama kartı ile kullanımı

3. Adım : Bir bardak suyun içine sensörü daldırınız ve suyun derinliğine göre LED tepkisini gözlemleyiniz.

SIRA SİZDE

Ufak bir su kabı içindeki suyun seviyesine göre;

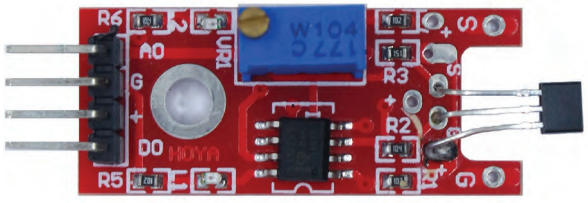
- Su seviyesi %10'un altındaysa sürekli,
- Su seviyesi %50'nin altındaysa 500 milisaniyelik kesik,
- Su seviyesi %70'in altındaysa 1 saniyelik kesik,
- Su seviyesi %70 ve üzeriyse buzzeri susturan sistemi tasarlayınız.

2.4.1.8. Manyetik Alan Sensörü

Sensörün yapısı incelenirse manyetik alan sensörünün hem analog hem de dijital çıkış verebildiği görülür. Modülün dijital çıkışı modül üzerinde yer alan potansiyometre ile belirlenen bir eşik değerinin üzerindeyse lojik 0, belirlenen eşik değerinin altındaysa lojik 1 olur.

Görsel 2.51 incelendiğinde şu bilgilere ulaşılır:

- D0 yazan pin, dijital çıkıştır.
- + yazan pin, 5 volt beslemedir.
- G yazan pin, GND (toprak) beslemedir.
- A0 yazan pin, analog çıkıştır.



Görsel 2.51: Manyetik alan sensörü modülü

9. UYGULAMA

Görsel 2.51'deki manyetik alan sensörünü kullanarak belirli bir alanda oluşan manyetik alanı tespit ediniz. Buradaki manyetik alan gücünü seri bağlantı üzerinden bilgisayara aktaran çalışmayı uygulayınız.

Gerekli Malzemeler

- 1 adet mikrodenetleyicili uygulama kartı
- 1 adet manyetik alan sensörü modülü
- 1 adet breadbord
- Bağlantı kabloları

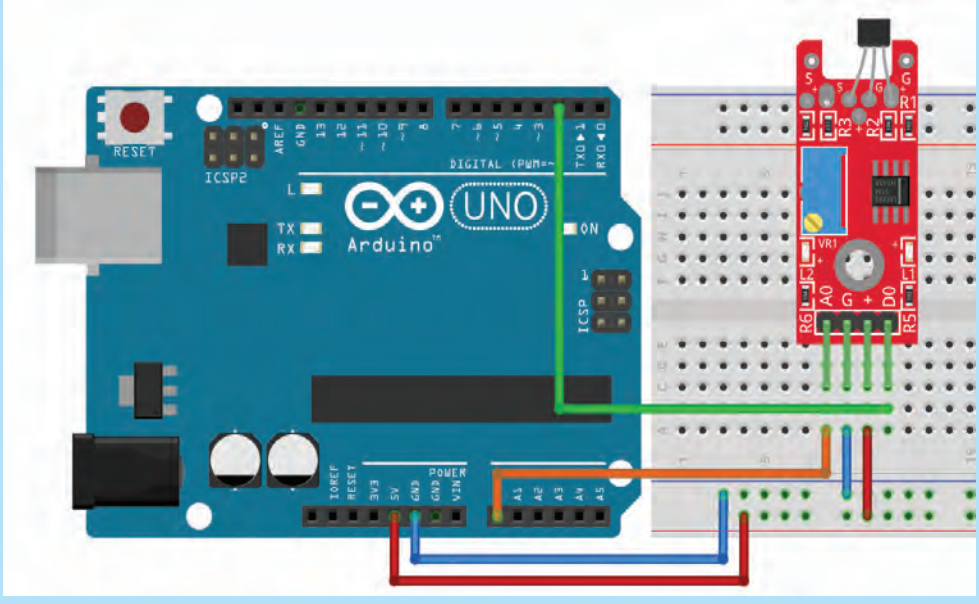
1. Adım : Sensörün eşik değerini ayarlayınız. Bu ayarlama için sensörünüzün (+) pinini +5 volt besleme hattına, G pinini GND besleme hattına bağlayınız. Bir tornavida yardımı ile potansiyometreyi saat yönünün tersine (sola) L2 LED'i sönünceye kadar çeviriniz. L2 LED'i söndükten sonra mıknatısla tepkisini ölçünüz. Sensörün hassasiyetine göre saat yönünün tersine ya da saat yönünde çevirerek eşik değerini tanımlayınız. Bu tanımlama sonrası sensör kullanıma hazırdır.

2. Adım : Görsel 2.52.a'daki programı uygulayınız.

```
const int led = 13;
const int dPin = 3;

void setup ()
{
  pinMode (led, OUTPUT);
  pinMode (dPin, INPUT);
  Serial.begin(9600);
}

void loop ()
{
  if (digitalRead(dPin)  == HIGH)
  {
    Serial.println(analogRead(A0));
    digitalWrite (led, HIGH);
  }
  else
  {
    digitalWrite (led, LOW);
  }
  delay(100);
}
```

Görsel 2.52.a: Manyetik alan sensörü için hazırlanmış program

Görsel 2.52.b'deki devre şemasını uygulayınız.



Görsel 2.52.b: Manyetik alan sensörü modülünün mikrodenetleyicili uygulama kartı ile kullanımı

NOT

Analog pininden işlem yapabilmek için öncelikle Görsel 2.52.a'daki programın ve Görsel 2.52.b'deki devrenin uygulanması son derece önemlidir. Bu sayede istediğiniz eşik değerini seri monitör üzerinden rahatlıkla gözlemleyip buna uygun çalışmalar yapabilirsiniz.

SIRA SİZDE

Manyetik sensör kullanılarak bisiklet için hızölçer yapılacaktır. Bu sistem için bisikletin jantına bir adet mıknatıs ve bisikletin çatalına mıknatısa hizalı manyetik okuyucu yerleştirilecektir. Bisiklet kullanıcısının gittiği kilometreyi ve yaptığı hızı gösterecek 2 satır 8 sütunluk LCD ekran ile bilgi aktarılacaktır. İstenenler doğrultusunda sistemi tasarlayınız.

NOT

Bisikletle alınan yolu bulabilmek için **Bisikletin Tekerlek Çevresi * Attığı Tur Sayısı** formülü kullanılır. Hız için **Tekerleğin Çevresi / Bir Turda Geçen Süre** formülü göz önünde bulundurulmalıdır.

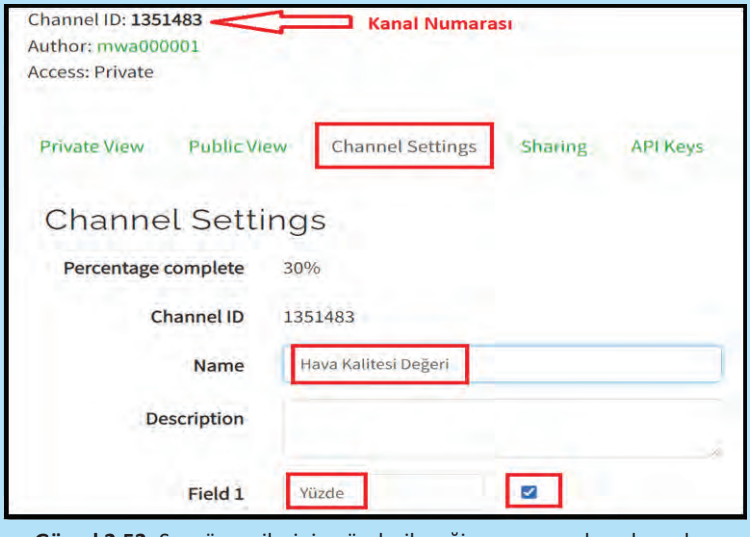
10. UYGULAMA

Bu örnekte NodeMCU kullanılarak IoT tabanlı hava kirliliği / hava kalitesi izleme sistemi yapılır. Toplanan hava kalitesi verileri, o çevredeki hava kalitesi hakkında kayıt yapabilecek ve sonuçlar çıkarabilecek Thingspeak'e gönderilir.

Gerekli Malzemeler

- 1 adet mikrodenetleyicili uygulama kartı
- 1 adet MQ135 sensör
- 1 adet I2C LCD panel
- 1 adet breadbord
- Bağlantı kabloları

1. Adım : Hava kalitesi verilerini almak için Thingspeak hesabı tanımlayınız. Thingspeak hesabını henüz yapmadıysanız, öncelikle bir Thingspeak hesabı oluşturmalısınız.
2. Adım : Görsel 2.53'te gösterilen Kanal ismi (Name) ve Alan (Field) kısımlarını doldurunuz. Daha sonra sayfanın en altına gidiniz ve kanalı **kaydet**'i tıklayınız.

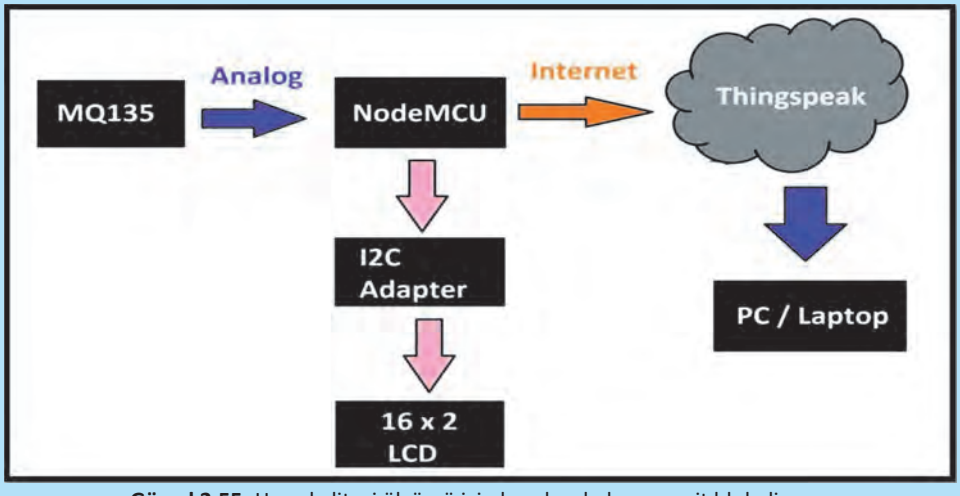


Görsel 2.53: Sensör verilerinin gönderileceği sunucunun kanal ayarları

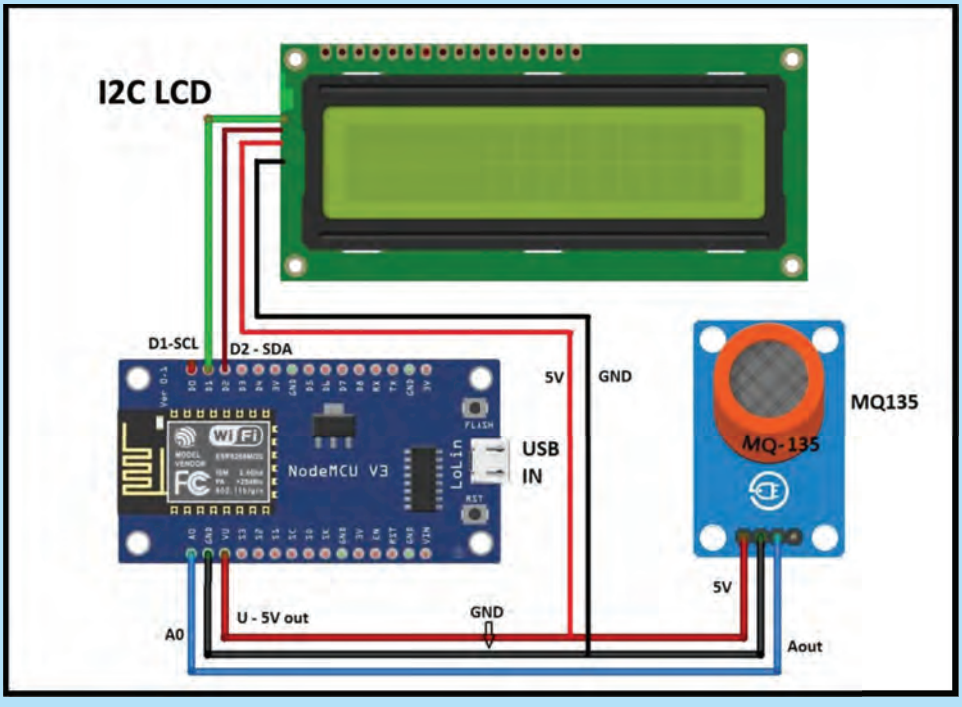
3. Adım : Go to **API Keys** sekmesine gidiniz ve size özel üretilen **Write API Key** alanında verilen **Key** kodunu not alınız (Görsel 2.54).

Görsel 2.54: Sensör verilerinin gönderileceği sunucuya veri yazmak için gerekli anahtar kodu

NodeMCU, bir Arduino kartının birçok işlevini yerine getirebilir ve yerleşik ESP8266 tabanlı Wi-Fi modülüne de sahiptir. MQ135 sensör verilerinin analogdan dijitale dönüştürülmesi, toplanan verilerin Thingspeak'e gönderilmesi ve bu verilerin kullanıcılar tarafından gözlenmesi için kurulacak devrenin blok şeması (Görsel 2.55) ile devre bağlantı şeması (Görsel 2.56) verilmiştir. Devrenizi Görsel 2.56'daki gibi bord üzerinde kurunuz.



Görsel 2.55: Hava kalitesi ölçümü için kurulacak devreye ait blok diyagram



Görsel 2.56: Hava kalitesi ölçümü için devre şeması

Bu devrede ayrıca kablo bağlantılarının sayısını azaltmak için LCD ekran için I2C adaptör modülü kullanılacaktır.

NodeMCU'nun I2C pinleri SCL olan D1 ve SDA olan D2, I2C adaptör modülüne bağlıdır.

MQ135 sensörü 5V beslemeye ihtiyaç duyar. Vcc pinini 5V çıkış veren NodeMCU'nun "VU" pinine bağlayabilirsiniz. Farklı bir üreticiden bir NodeMCU'nuz varsa MQ135'in Vcc'sini NodeMCU'nun 5V sağlayabilen "Vin" pinine bağlayabilirsiniz.

5. Adım : NodeMCU'ya yüklemek ve hava kalitesi ölçümünü yapabilmek için aşağıdaki kodu kullandığınız editöre yazınız.

```
#include <LiquidCrystal_I2C.h>
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
//----------- Wifi Bilgilerinizi Giriniz---------//
char ssid[] = "xxxxxx";   // SSID ismi
char pass[] = "yyyyyy";   // Ağ şifresi
//----------------------------------------//
//------- Sensör verilerinin gönderileceği API Key bilgileri--------//
unsigned long Channel_ID = 00000; // Kanal Numaranız (Channel ID)
const char * WriteAPIKey = "ABCDEF"; // API Key Kodunuz (write API Key)
// ----------------------------------------//

const int Field_number = 1;
float value;
int raw;
WiFiClient  client;

void setup()
{
  WiFi.mode(WIFI_STA);
  ThingSpeak.begin(client);
  lcd.init();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("IoT Hava Kalitesi");
  lcd.setCursor(0, 1);
  lcd.print("İzleme Sistemi");
  delay(2000);
  internet();
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("MQ135 sensor");
  lcd.setCursor(0, 1);
  lcd.print("Veri Topluyor...");
  for (int i = 0; i < 3; i++)
  {
    delay(20000);
    delay(20000);
    delay(20000);
  }
}

void loop()
{
  get_value();
  upload();
}

```

```
50 void get_value()
51 {
52   lcd.clear();
53   lcd.setCursor(0, 0);
54   lcd.print("Hava Kriliği(%)");
55   lcd.setCursor(0, 1);
56   raw = analogRead(A0);
57   value = map(raw, 0, 1024, 1, 100);
58   lcd.print(value);
59   lcd.print(" %");
60   delay(5000);
61 }
62
63 void internet()
64 {
65   if (WiFi.status() == WL_CONNECTED)
66   {
67     lcd.clear();
68     lcd.setCursor(0, 0);
69     lcd.print("Bağlanıyor");
70     lcd.setCursor(0, 1);
71     lcd.print("internet!");
72     delay(2000);
73   }
74   if (WiFi.status() != WL_CONNECTED)
75   {
76     lcd.clear();
77     lcd.setCursor(0, 0);
78     lcd.print("Bağlanıyor");
79     lcd.setCursor(0, 1);
80     lcd.print(ssid);
81     lcd.print(" SSID...");
82     for (int i = 0; i < 5; i++)
83     {
84       WiFi.begin(ssid, pass);
85       delay(5000);
86     }
87     if (WiFi.status() != WL_CONNECTED)
88     {
89       lcd.clear();
90       lcd.setCursor(0, 0);
91       lcd.print("Internet Yok");
92       lcd.setCursor(0, 1);
93       lcd.print("Bağlantı !");
94       delay(3000);
95     }
96     else if (WiFi.status() == WL_CONNECTED)
97     {
98       lcd.clear();
99       lcd.setCursor(0, 0);
100      lcd.print("Bağlanıyor");
101      lcd.setCursor(0, 1);
```

```
      lcd.print("internet!");
      delay(2000);
    }
  }
}

void upload()
{
  ThingSpeak.writeField(Channel_ID, Field_number, value, WriteAPIKey);
}

```

6. Adım : NodeMCU'ya kod yükleme işlemi için aşağıdaki işlemleri sırası ile takip ediniz.

- NodeMCU'yu USB kablosunu kullanarak bilgisayarınıza bağlayınız.
- Menüden **Tools > Board > NodeMCU 1.0** seçeneğini seçiniz.
- Menüden **Tools > Upload speed > 115200** seçeneğini seçiniz.
- Yerleşik flaş düğmesini basılı tutunuz.
- Sıfırlama düğmesine bir kez basıp bırakınız.
- Sıfırlama düğmesini bıraktıktan sonra flaş düğmesini bırakınız.
- **Yükle**'yi tıklayınız.

7. Adım : *Thingspeak* web sayfasını açarak "Private View" sekmesine basınız ve grafiğinizin oluşmasını bekleyiniz. Sensörden aldığınız veriler ekranda görünecektir (Görsel 2.57).

Görsel 2.57: Sensörden gelen verilerin grafikleştirilmesi

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

1. Dirençler devredeki ……sınırlandırmak ve ……bölmek için kullanılır.
2. Mikrodenetleyicilerde yazılan programın saklandığı hafıza birime ……. denir.

B) Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

3. 5 bantlı bir dirençteki renkler kırmızı, kırmızı, siyah, kırmızı ve altındır. Bu direncin değeri aşağıdakilerden hangisidir?

A) 220 Ω B) 2,2 KΩ C) 22 KΩ D) 220 KΩ E) 2,2 MΩ

4. Mikrodenetleyici ile melodi çalmak için aşağıdakilerden hangisi kullanılmalıdır?

A) Aktif (Devreli) Buzzer
B) LED
C) Optokuplör
D) Pasif (Devresiz) Buzzer
E) Transistör

5. **I. Optokuplör**
II. Röle
III. Transistör

Mikrodenetleyici ile yüksek güç çeken donanım sürmek için aşağıdakilerden hangileri birlikte kullanılmalıdır?

A) I, II ve III B) Yalnız I C) Yalnız II D) Yalnız III E) II ve III

6. İçinde LED ve bu LED'e duyarlı transistör barındıran, LED yandığında yapısındaki transistörü süren devre elemanı aşağıdakilerden hangisidir?

A) Optokuplör B) Transistör C) LDR D) Röle E) Buzzer

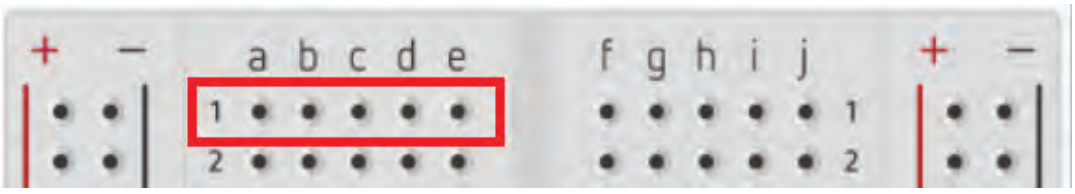
7. Base bacağına gelen küçük bir tetikleme sinyali ile collector ve emitter bacakları arasında yüksek miktarda gerilim ve akım geçmesine izin veren sürme elemanı aşağıdakilerden hangisidir?

A) Direnç
B) L293D motor sürücü
C) Optokuplör
D) Röle
E) Transistör

8. Motor sürücü entegrelerindeki Enable pini hakkında aşağıda verilen bilgilerden hangisi doğrudur?

A) Boş bırakılmalıdır.
B) + bağlantısı için kullanılır.
C) GND bağlantısı için kullanılır.
D) Motorun dönüş hızını ayarlar.
E) Motorun dönüş yönünü ayarlar.

9.



Şekildeki breadbord üzerinde kırmızı çerçeve ile işaretlenmiş bölümün adı aşağıdakilerden hangisidir?

A) Entegre bölgesi
B) Soket şerit
C) Bus şerit
D) 1 numaralı bölge
E) abcde bölgesi

10. **I. Mikroişlemci**
II. RAM / ROM
III. ADC / DAC
IV. Zamanlayıcı ve sayıcı

Mikrodenetleyici içindeki temel bileşenler aşağıdakilerden hangisidir?

A) Yalnız I B) Yalnız II C) I ve II D) I, II ve IV E) I, II ve III

11. **Mikrodenetleyici içinde kurulduktan sonra ana program akışını bozmadan arka planda kendi kendine çalışan yapı aşağıdakilerden hangisidir?**

A) Giriş-çıkış birimi
B) Mikroişlemci
C) RAM
D) ROM
E) Zamanlayıcı ve sayıcı

12. **Dış ortamdaki bilgileri mikrodenetleyicinin anladığı ikilik sayı sistemine çeviren birim aşağıdakilerden hangisidir?**

A) Mikroişlemci B) Zamanlayıcı ve sayıcı C) DAC D) ADC E) Seri port

13. **Dış ortamdaki sıcaklığı ölçmek için aşağıdakilerden hangisi kullanılır?**

A) LDR B) MQ-4 C) LCD D) LM35 E) Röle

14. I. Sensörün güç bağlantıları yapılır.
II. Sensörün ısınması beklenir.
III. Gaz uygulanır.
IV. Gaza duyarlı tel üzerinde direnç değişir.
V. Bilgi mikrodenetleyiciden okunur.
Yukarıdaki işlem basamaklarını doğru bir şekilde sıralayınız.

15. **12 voltluk bir kaynaktan 3,3 voltluk Wi-Fi modülü sürülecektir. R1 direnci 1 KΩ olduğuna göre R2 direnci kaç KΩ olmalıdır?**

16. **Birbirine seri bağlı 5 adet beyaz LED 30 volt kaynakla beslenecektir. Beyaz LED'lerin parlak yanması istenmektedir. Bu nedenle 1 adet beyaz LED için 4,5 volt ve 30 mA değerleri kullanılacaktır. Ön direnç değeri kaç Ω olmalıdır?**

NESNELERİN İNTERNETİNDE PROGRAMLAMA

3. Öğrenme Birimi

KONULAR

3.1. BLOK TEMELLİ PROGRAMLAMA
3.2. PYTHON İLE PROGRAMLAMA
3.3. VERİ İŞLEME SÜREÇLERİ
3.4. API'LER
3.5. KOD GÜVENLİĞİ
3.6. RASPBERRY Pİ KULLANIMI
3.7. SİMÜLASYON ARACI

NELER ÖĞRENECEKSİNİZ?

- Blok temelli uygulama aracını kullanma
- Blok programlama
- Python ile programlama
- Veri ve bilgi tanımlarını yapma
- IoT sistem bileşenlerini tanıma
- Veri işleme döngüsünü yapma
- API'lerin çalışma mantığı

TEMEL KAVRAMLAR

API, bilgi, blok kodlama, IoT, Python, Rasperry Pi, Python, veri, veri işleme döngüsü

HAZIRLIK ÇALIŞMALARI

1. Blok temelli kodlama ile yapılan program uygulamalarından bildiklerinizi arkadaşlarınızla paylaşınız.
2. IoT cihazlarında veri ve bilgiye örnekler veriniz.
3. Veri işlemenin önemini açıklayınız.

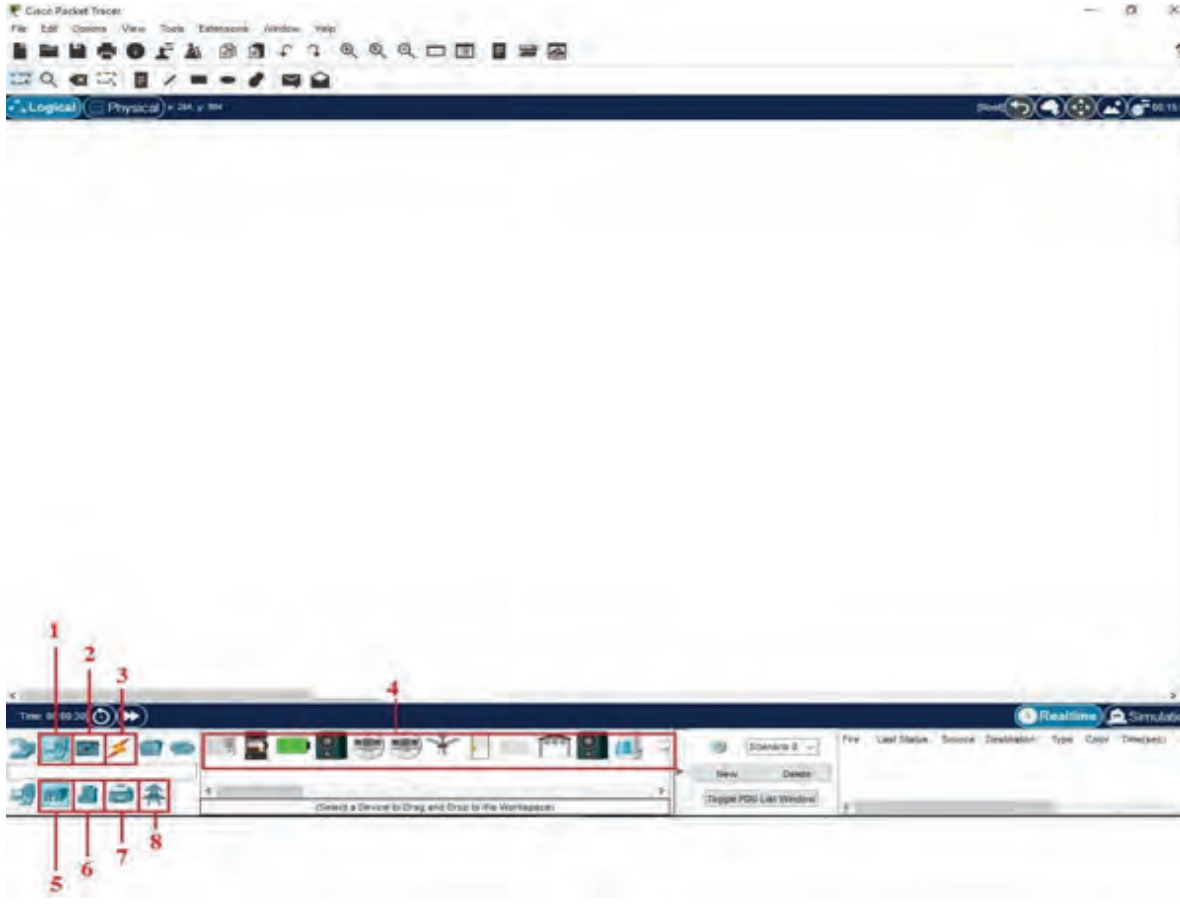
http://kitap.eba.gov.tr/KodSor.php?KOD=14398

3.1. BLOK TEMELLİ PROGRAMLAMA

Blok temelli programlama günümüz popüler programlama teknikleri arasında önemli bir yere sahiptir. Bloklar sayesinde programlar daha kolay yazılabilmektedir.

3.1.1. Blok Temelli Uygulama Aracı

Nesnelerin İnterneti (IoT) birden fazla programlama dili ile gerçekleştirilebilir. Programlama dili ile uygulama geliştirmek için kullanılan programlama dilinin öğrenilmesinin yanı sıra yazılım dili söz dizimi (syntax) iyi bilinmelidir ancak blok programlama ile söz dizimi kurallarının bilinmesine gerek yoktur. Programlamada kullanılan blokların nasıl sıralanacağı ile görevleri, kullanıcılara kolaylık sağlanması amacıyla belirlenmiş ve gruplandırılmıştır. Blok programlama ile oluşturulmuş bir program JavaScript, PHP veya Python diline çevrilerek çalıştırılır. Bu öğrenme biriminde simülatör uygulaması kullanılacaktır. Simülatör uygulamasına kullanıcı adı ve şifre ile giriş yapıldıktan sonra programlamada kullanılacak nesnelerin bulunduğu menüler Görsel 3.1'de verilmiştir.



Görsel 3.1: IoT cihaz grupları

Görsel 3.1'de yer alan rakamların işaret ettiği nesneler aşağıda belirtilmiştir:

1: Kullanıcı cihazları
2: Bileşenler (elektronik devre kartları)
3: Kablo türleri
4: Seçili grup içinde bulunan nesneler
5: Ev otomasyonunda kullanılan nesneler
6: Akıllı şehir otomasyonlarında kullanılabilecek nesneler
7: Endüstriyel otomasyonlarda kullanılabilecek nesneler
8: Yenilenebilir enerji otomasyonlarında kullanılabilecek nesneler

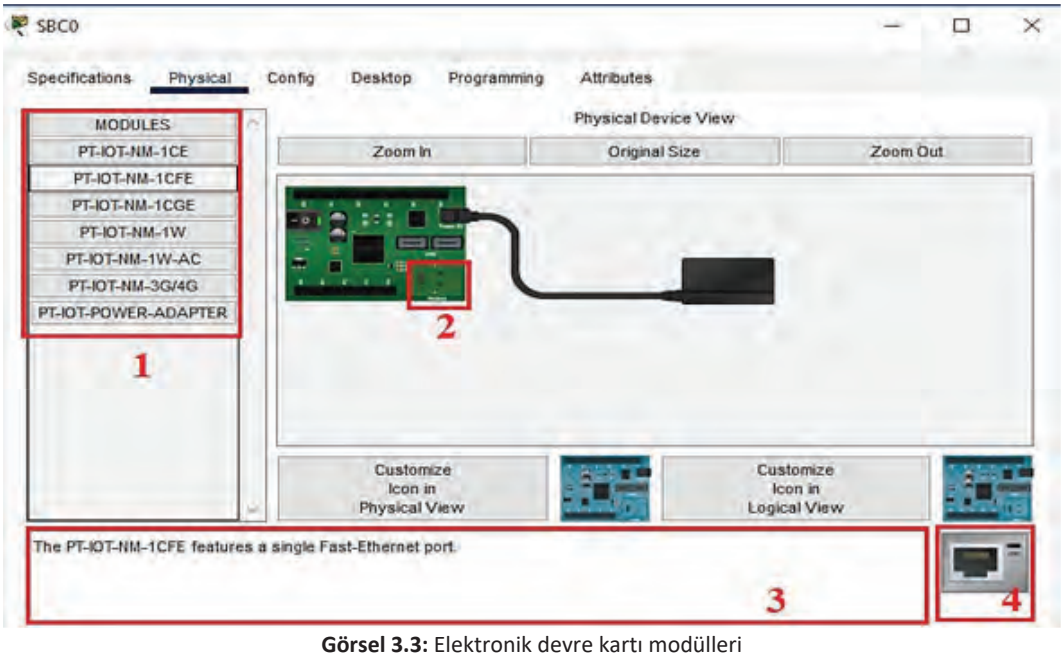
Uygulamada kullanılacak grubun seçilmesi ile gruba ait nesneler orta sekmede görünür. Uygulamada kullanılacak nesne seçilir ve uygulama alanına yerleştirilir. Gerçekleştirilmek istenen uygulama daha sonra blok programlama ile yazılır.

3.1.2. Blok Programlama

Simülatör uygulamasında blok programlamaya başlamak için öncelikle Bileşenler grubundan otomasyonda kullanılacak elektronik devre kartı seçilir (Görsel 3.2).

Görsel 3.2: IoT elektronik devre kartı

Seçilen elektronik devre kartı uygulama alanına yerleştirildikten sonra üzerine farenin (mouse) sol tuşu ile tıklandığında açılan pencerenin Physical sekmesinde kartın üzerine yerleştirilebilecek modüller görülür. Seçilen modülün bilgileri açıklama alanında bulunur.

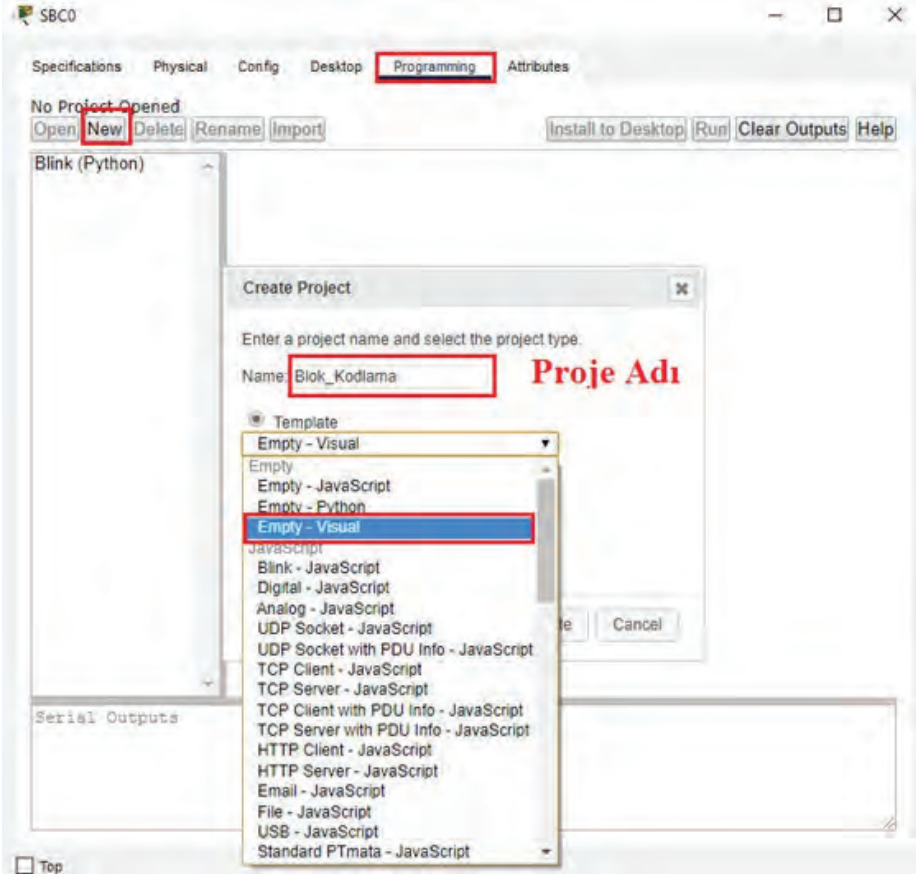


Görsel 3.3: Elektronik devre kartı modülleri

Görsel 3.3'te yer alan rakamların işaret ettiği nesneler şunlardır:

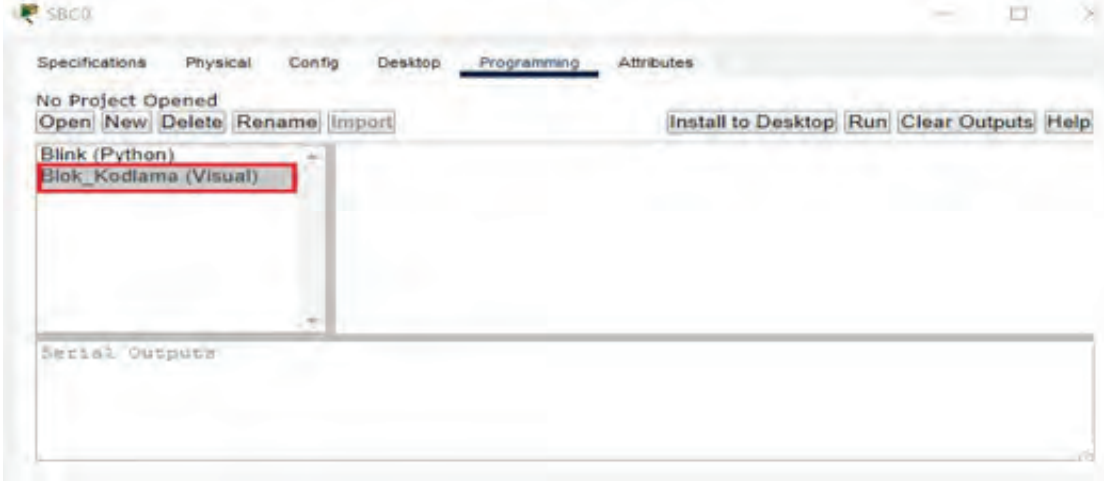
1: Seçilen elektronik devre kartına eklenebilecek modüller
2: Modülün ekleneceği alan
3: Modül bilgilendirme alanı
4: Seçilen modülün görüntüsü

Seçilen modül, sürükle bırak yöntemiyle modülün ekleneceği alana bırakıldıktan sonra kullanıma hazır olarak elektronik devre kartına eklenir.



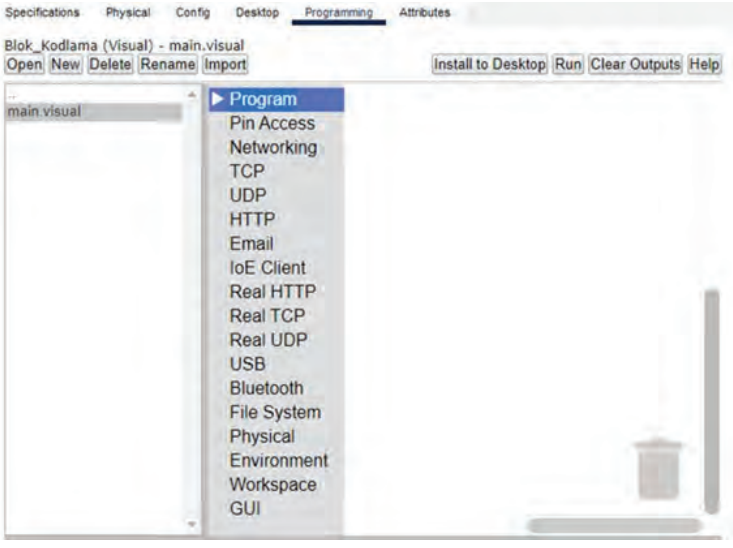
Görsel 3.4: Blok kodlama için yapılandırma

Modül eklendikten sonra Programming sekmesine tıklanarak işleme devam edilir ve sırası ile No Project Opened kısmından New seçeneği ile yeni oluşturulacak proje için gerekli tercihlerin yapılacağı menüye geçilir. Açılan bu menüde proje adı girildikten sonra blok kodlama yapabilmek için Empty_Visual seçeneği seçilip Create butonuna basılır (Görsel 3.4).



Görsel 3.5: Blok kodlama yapılandırması

Oluşturulan yeni proje, girilen ismi ile listeye eklenir. Bu projede çalışma yapmak için farenin sol tuşu ile çift tıklanır (Görsel 3.5).



Görsel 3.6: Blok kodlama grupları

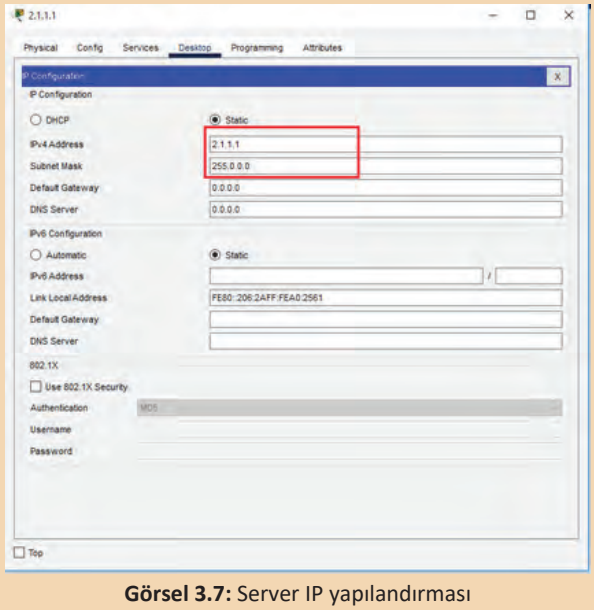
Açılan listeden main.visual seçeneği seçilip fare ile çift tıklandığında Görsel 3.6'da görülen blokların gruplandığı ve blok kodlamanın yapılacağı alanla karşılaşılır. Burada her grup içinde kendi amacına uygun kod blokları yer alır ve seçilen blok, sürükle bırak yöntemi ile kullanılır.

1. UYGULAMA

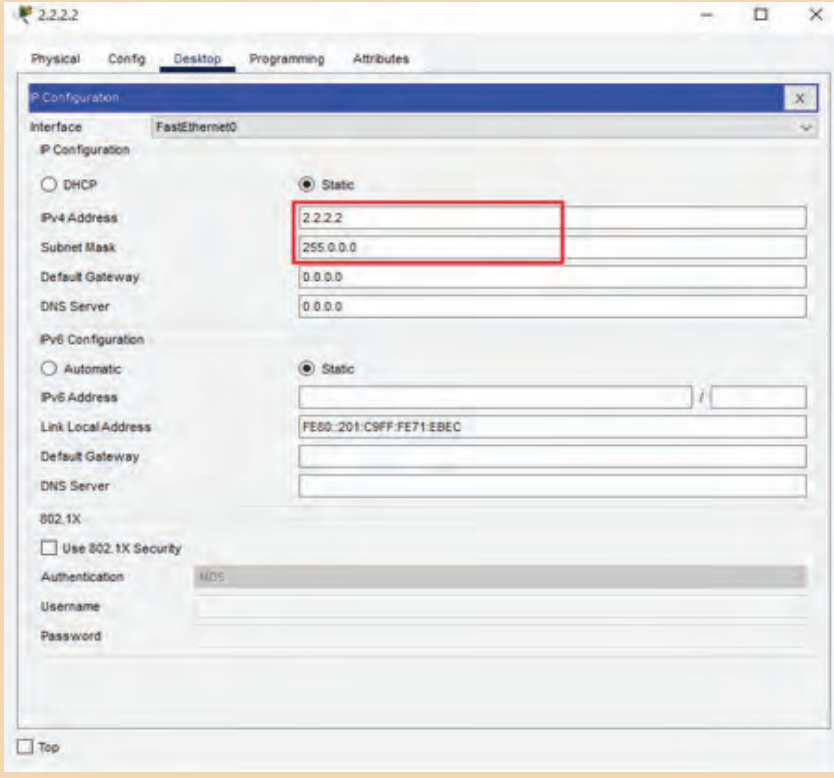
IoT Cihazlarda Elektronik Posta Gönderme İşlemi

Evin dışındaki hareketleri 5 saniye aralıklarla kontrol eden, hareket algılandığında evin içindeki lambayı açan ve ev sahibine e-posta gönderen uygulamayı yapınız. Kullanılacak cihaz bilgileri aşağıda verilmiştir.

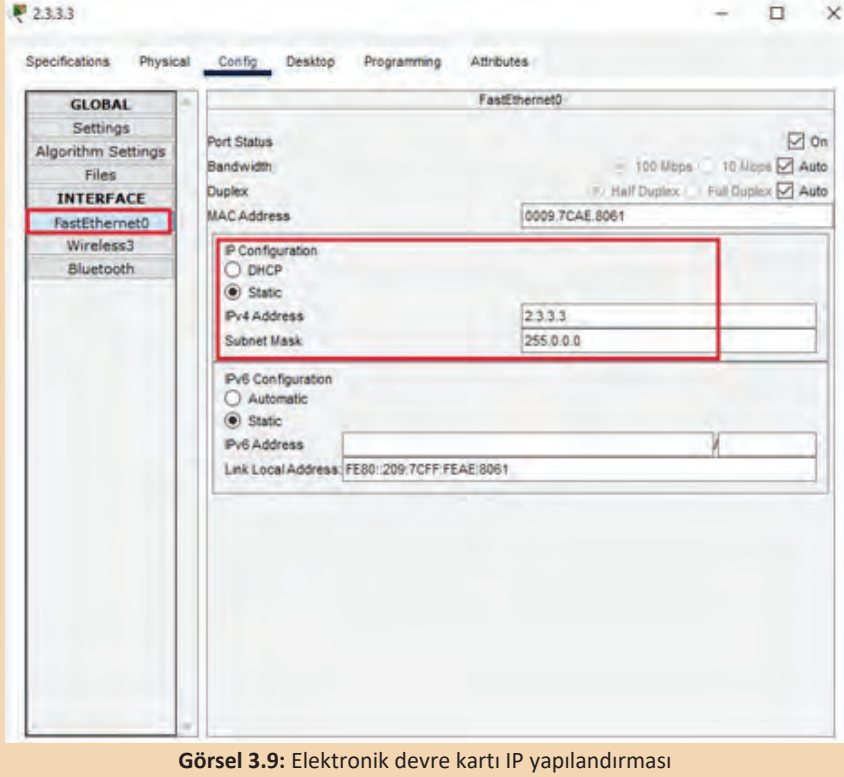
Server	IP 2.1.1.1	Subnet Mask 255.0.0.0 (Görsel 3.7)
Dizüstü (Laptop)	IP 2.2.2.2	Subnet Mask 255.0.0.0 (Görsel 3.8)
Elektronik Devre Kartı	IP 2.3.3.3	Subnet Mask 255.0.0.0 (Görsel 3.9)



Görsel 3.7: Server IP yapılandırması

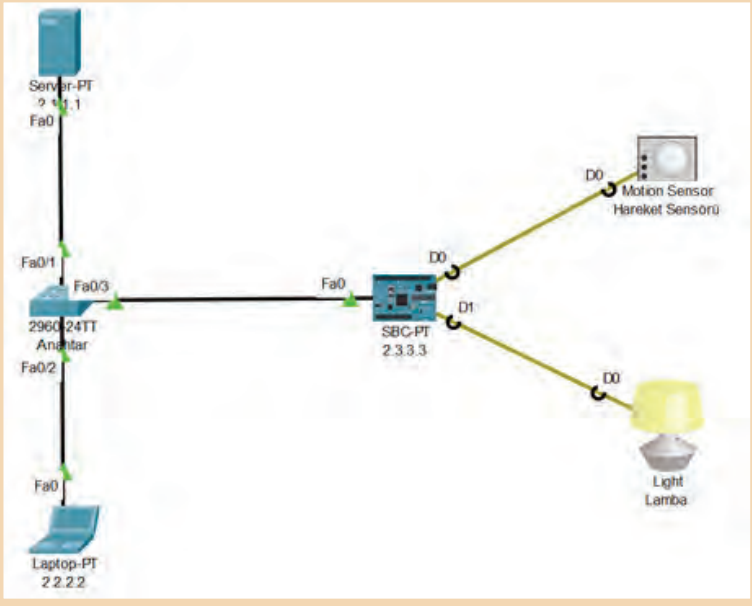


Görsel 3.8: Dizüstü (Laptop) bilgisayar IP yapılandırması



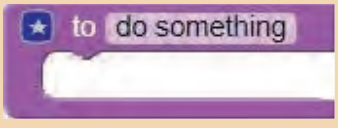
Görsel 3.9: Elektronik devre kartı IP yapılandırması

Kullanılacak cihazların IP yapılandırma işlemleri tamamlandıktan sonra cihazların birbiriyle bağlantısı gerçekleştirilir.

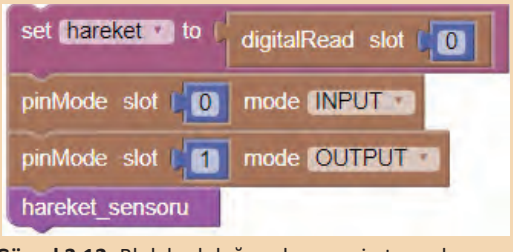


Görsel 3.10: Tüm cihazların bağlantı yapılmış hâli

Sistemde kullanılacak cihazların kablolamaları yapıldıktan sonra blok kodlama sayfasına dönülerek blok kodlar yazılır (Görsel 3.10). Kullanım kolaylığı açısından her işlem için ayrı fonksiyon kodlaması yapılır. Bunun için blok kodlama sayfasında Program > Functions sekmesinden Görsel 3.11'deki fonksiyon tanımlama bloku seçilip programlamada kullanılacak blok tanımlamaları ve fonksiyonların görevine göre gerekli blok programları yazılır.



Görsel 3.11: Fonksiyon tanımlama bloku



Görsel 3.12: Blok kod değer okuma, pin tanımlama ve fonksiyon çağırma

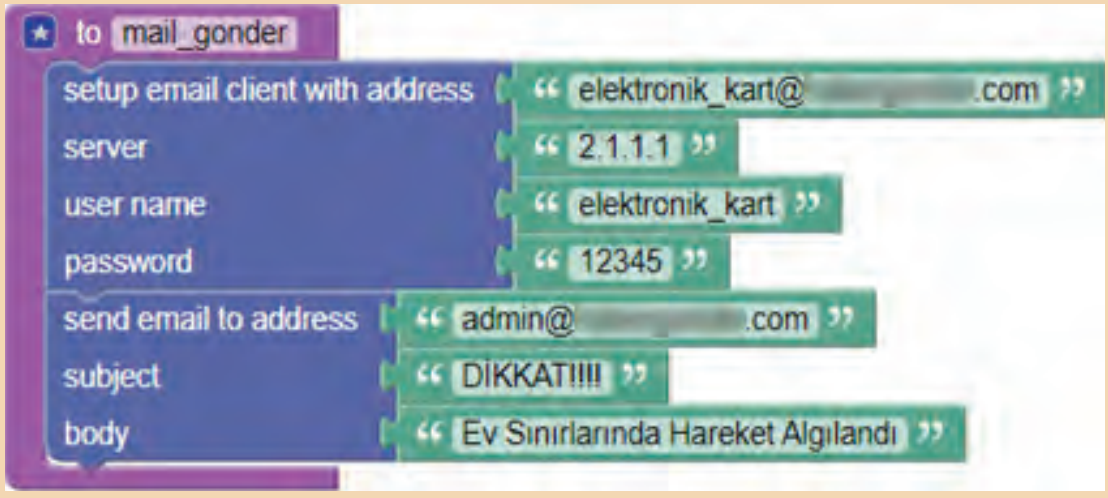
Bu blok ile elektronik devre kartının D0 pininden hareket adlı değişkene okuma değeri aktarılır. 0 numaralı pin giriş pini, 1 numaralı pin çıkış pini olarak tanımlandıktan sonra hareket_sensoru isimli fonksiyon çağrılır (Görsel 3.12).

Bu blokta hareket_sensoru fonksiyonu ile yapılacak işlemler görülmektedir. Fonksiyon çağrıldıktan sonra while döngüsü ile 5 saniye aralıklarla mail_gonder, sensor_oku ve hareket_durumu fonksiyonlarının çalıştırılması sağlanır (Görsel 3.13).

Görsel 3.13: Hareket sensörü fonksiyon içeriği

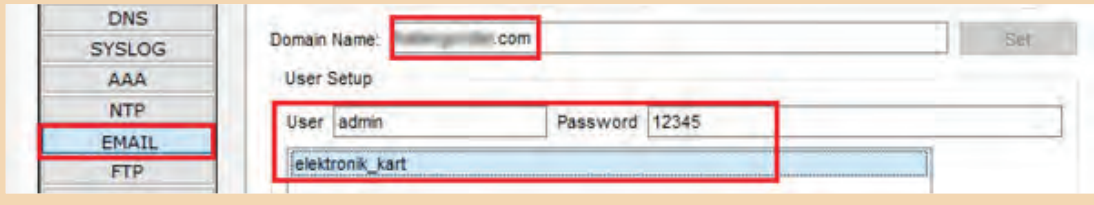
Bu blokta mail_gonder fonksiyonu ile yapılacak işlemler görülmektedir.

İlk blokta setup email client with address kısmında e-posta istemcisinin adresi girilir. Server kısmında kullanılan Server IP adresi girilir. User ve password kısmında ise Server'de tanımlanan kullanıcı adı ve şifre girilir. İkinci blokta ise send email to address bölümüne e-posta gönderilecek adres girilir. Subject kısmına e-posta konu başlığı, body kısmına da e-posta içeriği girilir. Buradaki mail adresleri ve kullanıcı isimleri Server üzerinde tanımlanan değerlerle aynı olmak zorundadır. Aksi takdirde e-posta gönderilemez (Görsel 3.14).

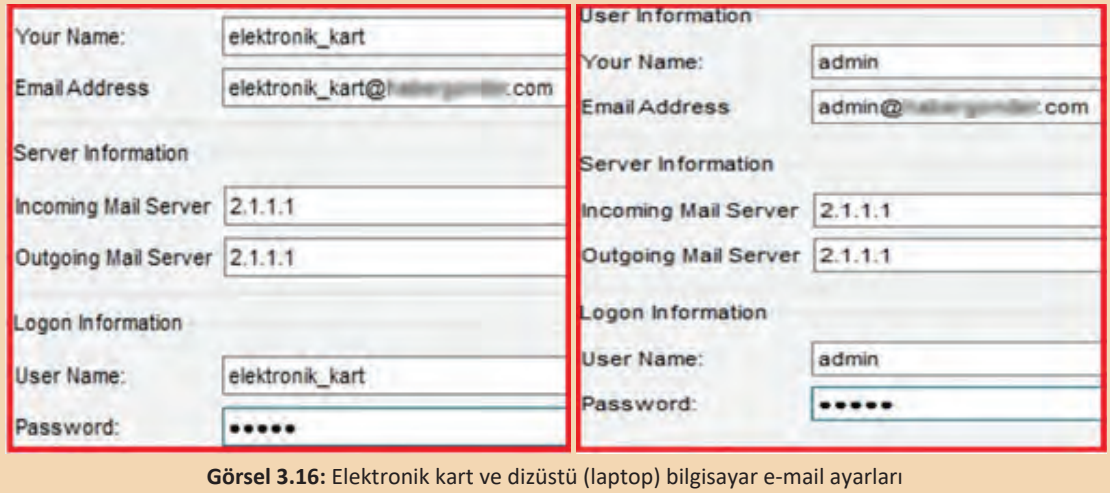


Görsel 3.14: mail_gonder kod bloku

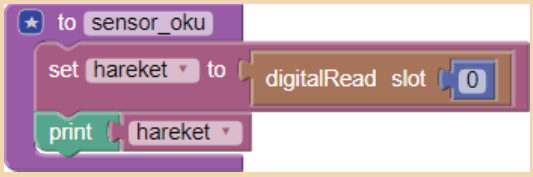
Görsel 3.15'te Server e-mail ayarları, Görsel 3.16'da elektronik kart ve dizüstü bilgisayar (laptop) e-mail ayarları görülmektedir.



Görsel 3.15: Server e-mail ayarları



Görsel 3.16: Elektronik kart ve dizüstü (laptop) bilgisayar e-mail ayarları



Görsel 3.17: sensor_oku kod bloku

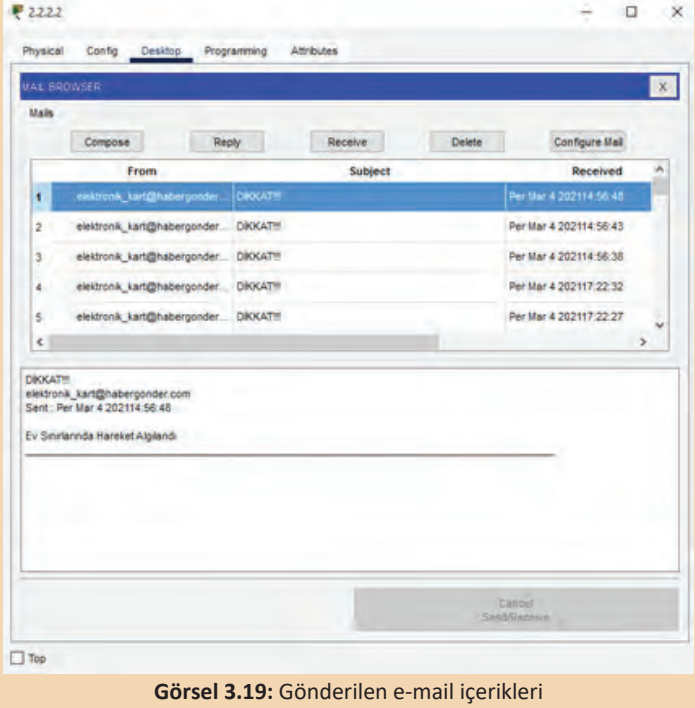
Bu blokta sensor_oku fonksiyonu ile yapılacak işlemler görülür. Fonksiyon çağrıldıktan sonra D0 (dijital 0) pininden okunan değer, hareket isimli değişkene aktarılacak ve seri port ekranında hareket değişkeninin içeriği gösterilecektir. Hareket tespit edilirse değişken içeriği 1023 olacak, hareket tespit edilmezse değişken değeri 0 olacaktır (Görsel 3.17).

Bu blokta hareket_durumu fonksiyonu ile yapılacak işlemler görülür. Fonksiyon hareket değişkeninin değeri ile girilen 1023 değeri karşılaştırılır. Hareket değişkenine hareket algılanıp 1023 değeri atandı ise seri port ekranına Hareket Algılandı uyarısı verilir ve D1 (dijital 1) çıkışının değeri de 2 yapılarak lambanın yanması sağlanır (Görsel 3.18).

Görsel 3.18: Hareket durumu kod bloku

Çıkış değerinin 2 olarak verilmesi lambanın çalışma şartına göredir. 0- Kapalı, 1- Az parlak, 2- Çok parlak değerlerini ifade eder.

Görsel 3.19'da kodun çalıştırılması ile hareketin tespit edilmesi sonucu gönderilen e-maillerin listesi ve içeriği görülmektedir.



Görsel 3.19: Gönderilen e-mail içerikleri

SIRA SİZDE

Kapının ne zaman açık veya kapalı olduğunu gösteren ışıklara sahip, hareketle çalışan bir kapı tasarlanarak kapı açıldığında bir iç aydınlatmanın yanacağı ve yöneticilere bir e-postanın gönderileceği programı aşağıdaki şartlara göre blok kodlama ile yapınız.

a) Hareket sensöründe hareket algılandığında kapının açılmasını sağlayınız.
b) Kapı açık iken RGB LED'in yeşil, kapalı iken RGB LED'in kırmızı yanmasını sağlayınız.
c) Hareket algılandığında ve kapı açıldığında iç aydınlatmanın yanmasını sağlayınız.
ç) Kapıdan içeri biri girdiğinde otomatik olarak yöneticiye e-mail atılmasını sağlayınız.

3.2. PYTHON İLE PROGRAMLAMA

Python, kolay öğrenimi ve basit söz dizimi (syntax) ile popüler bir üst seviye programlama dilidir. Yapay zekâ, görüntü işleme gibi birçok alanda kullanılır.

SIRA SİZDE

Klavyeden girilen iki sayının toplamını bulan Python kodunu yazınız.

```
sayi1 = input("1. sayıyı giriniz : ")
sayi2 = input("2. sayıyı giriniz : ")
toplam = int(sayi1)+int(sayi2)
print("Toplam :{0} ".format(toplam))
```

SIRA SİZDE

for döngüsü ile 1 ile 50 arasındaki çift sayıları bulan Python kodunu yazınız.

```
for i in range(1,51):
    if i%2==0:
        print(i)
```

2. UYGULAMA

while döngüsü ile 1 ile 50 arasındaki tek sayıların toplamını bulan Python kodunu yazınız.

```
sayac = 0
toplam = 0
while sayac < 50:
    if sayac%2 == 1:
        toplam += sayac
sayac += 1
print("Toplam : {0}" .format(toplam))
```

3. UYGULAMA

Satışı yapılan bir takım elbisenin fiyatı 500,00 TL, ayakkabının fiyatı ise 200,00 TL olarak belirlenmiştir. Satış esnasında peşin ödemelerde ürünün fiyatında %20 indirim yaparak, kredi kartı ödemelerinde ise ürünün fiyatına %5 komisyon ekleyerek ödeme bilgisini veren Python kodunu yazınız.

```
secim = input("Takım elbise için (T/t), Ayakkabı için (A/a) tuşlayınız : ")
pesin = input("Peşin mi(E/H) : ")
fiyat = 0

if secim == 'T' or 't':
    fiyat = 500
elif secim == 'A' or 'a':
    fiyat = 200

if pesin == 'E' or pesin == 'e':
    fiyat=fiyat*0.8
else:
    fiyat = fiyat*1.05

print("Ödemeniz gereken ücret :{}".format(fiyat))
```

4. UYGULAMA

Klavyeden yarıçapı girilen dairenin alanını fonksiyon kullanarak hesaplayan Python kodunu yazınız.

```
def daire_alan(r):
    alan = 3.14* float(r)*float(r)
    print("Dairenin alanı: {0}" .format(alan))
    return alan
yaricap = input("Alanı hesaplanacak dairenin yarıçapı : ")
daire_alan(yaricap)
```

5. UYGULAMA

Kısa ve uzun kenar bilgileri klavyeden girilen dikdörtgenin alanını ve çevresini farklı fonksiyonlar kullanarak ayrı ayrı hesaplayan Python kodunu yazınız.

```
def d_alan(u_kenar,k_kenar):
    alan = float(u_kenar)*float(k_kenar)
    print("Dikdörtgenin alanı: {0}" .format(alan))
    return alan

def d_cevre(uu_kenar,kk_kenar):
    cevre = (float(uu_kenar)+float(kk_kenar))*2
    print("Dikdörtgenin çevresi: {0}" .format(cevre))
    return cevre

uzun_kenar = input("Dikdörtgenin uzun kenarının değeri : ")
kisa_kenar = input("Dikdörtgenin kısa kenarının değeri : ")

secim=input("Alan hesaplamak için (1) , Çevre hesaplamak için (2)")

if secim == '1':
    d_alan(uzun_kenar,kisa_kenar)

elif secim == '2':
    d_cevre(uzun_kenar,kisa_kenar)

else:
     print("Hatalı seçim yaptınız.")
```

6. UYGULAMA

1 ile 500 arasında rastgele seçilen bir sayının kaç tahminde bulunduğunu tespit eden Python kodunu yazınız (Random kütüphanesindeki ranint metodu kullanılacaktır. 0 (sıfır) değeri girildiğinde programdan çıkılacaktır.).

```
from random import randint
    random_sayi = randint(1, 500)
sayac = 0

while True:
    sayac += 1
    sayi = int(input("1 ile 500 arasında değer giriniz. (0 Çıkış):"))
    if(sayi == 0):
        print("Oyundan çıktınız.")
        break
    elif sayi < random_sayi:
        print("Yüksek bir sayı giriniz.")
        continue
    elif sayi > random_sayi:
        print("Düşük bir sayı giriniz.")
        continue
    else:
        print("Rastgele seçilen sayı {0}!".format(random_sayi))
        print("Tahmin sayınız {0}".format(sayac))
```

3.3. VERİ İŞLEME SÜREÇLERİ

Nesnelerin İnterneti (IoT) ile akıllanan cihazlar, hayatın her alanına entegre olmuştur. Bunlar, kullanıcılara çeşitli bilgiler gönderen vazgeçilmez cihazlardır. IoT ile akıllanan cihazlar; ulaşım, sağlık, tarım, hayvancılık, otomasyon, engellilere yönelik projeler gibi birçok alanda kullanılmaktadır. IoT cihazlarının gönderdiği veriler ile olumlu veya olumsuz durumların tespit edilmesi kolaylaşmıştır. IoT cihazların verileri toplayıp anlaşılır bir şekilde kullanıcılara sunması ile birçok olaya müdahale süresi azalmıştır.

Verimli çalışan bir IoT sistemi dört bileşenden meydana gelir. Bu bileşenler; sensörler / cihazlar, bağlantı, veri işleme ve kullanıcı arayüzüdür.

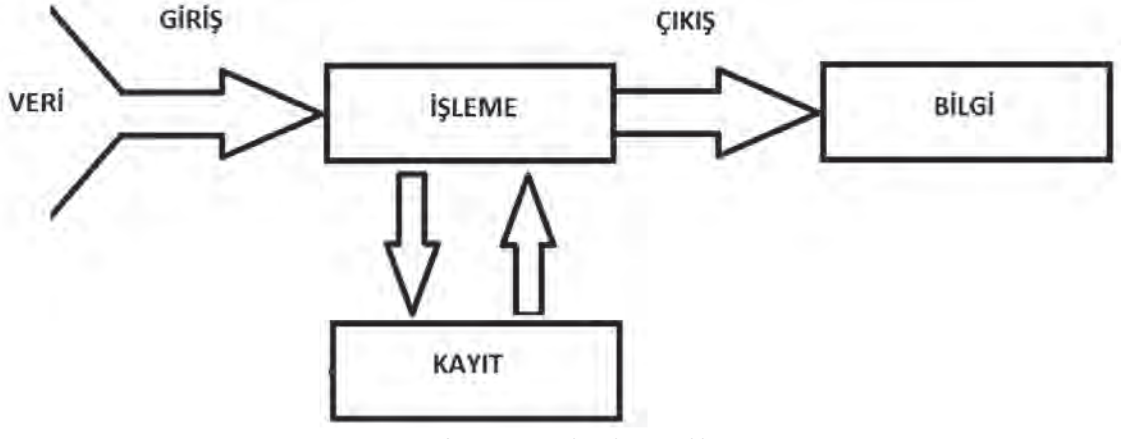
- **Sensörler / Cihazlar** : Bulundukları ortamdan, oluşturulan sistem için veri toplama görevini yerine getirir.
- **Bağlantı** : Toplanan verilerin kablolu veya kablosuz ağlarla sağlanan internet bağlantısıdır.
- **Veri İşleme** : Sensör ve cihazlar tarafından toplanan ham verilerin kullanılabilir hâle gelmesidir.
- **Kullanıcı Arayüzü** : Kullanılabilir hâle gelen verileri kullanıcıların anlayıp yorumlayacağı uygulamalardır.

IoT sistemlerinde dört bileşenin tümü önemlidir ancak veri işleme en zorlayıcı ve önemli bileşendir. Gönderilen ham verinin işlenerek anlamlandırılması ve kullanılması bu sistemlerin görevlerini sağlıklı, hızlı ve güvenilir yapmalarını sağlar. IoT cihazların fazlalığı, verilerin hızlı ve fazla miktarlarda üretilmesine sebep olmaktadır ancak sensörlerin ürettiği ve topladığı her veri kullanılabilir değildir. Verilerin kullanılabilir olması için bilgiye dönüşmesi gerekir.

Veri, işlenmemiş ve gereksiz içeriğe sahip ham yapıdadır. Bilgi ise işlenmiş ve kullanılabilir hâle dönüşmüş veridir. Veri işlemenin girdisi veri, veri işlemenin çıktısı bilgidir.

Veri işleme döngüsü; girdi, işleme ve çıktı olmak üzere üç temel aşamadan oluşur (Görsel 3.20).

- **Girdi** : Veri işlemenin ilk aşamasıdır. Toplanan verilerin bilgisayar tarafından işlenebilmesi, makine tarafından okunabilir forma dönüşümü ile mümkündür. Girdi aşamasında bu dönüşüm gerçekleşir. Verilerin kullanılabilir bilgiye dönüşmesi buradaki girdi verilerine bağlı olduğu için önemli bir aşamadır.
- **İşleme** : Bilgisayar ham verilerden bilgiyi bu aşamada elde eder. Bunu yaparken sınıflandırma, sıralama ve hesaplama gibi farklı veri işleme teknikleri kullanılabilir.
    - **Sınıflandırma** : Veriler farklı gruplara ayrılarak yapılır.
    - **Sıralama** : Veriler belirlenen sıralama yöntemine göre düzenlenir.
    - **Hesaplama** : Sayısal veriler üzerinde aritmetiksel ve mantıksal işlemlerin yapılmasıyla gerçekleşir.
- **Çıktı** : İşlenen verilerin okunabilir forma dönüştürüldüğü ve kullanıcılara servis edildiği aşamadır. Çıktılar saklanarak girdi şeklinde kullanılabilir.



Görsel 3.20: Veri işlem basamakları

IoT cihazlarında veri işlemede dikkat edilmesi gereken hususlar şunlardır:

- **İstenilen Çıktı** : IoT sensör ve cihazlarda üretilen ham verilerin işlenmesi sonucu elde edilen bilginin uygulamada kullanılabilir olmasıdır.
- **Verilerin Saklanması** : Sensörlerin topladığı verilerin kapasitesinin fazlalığı nedeniyle depolama bulut sistemde gerçekleşmelidir ancak veri kapasitesi çok fazla olduğu için bir veri saklama politikası oluşturulmalıdır.
- **Güncelleme Sıklığı** : Veri işlemeye başlamadan önce güncelleme sıklığı belirlenmelidir.
- **Veri İşleme Araçları** : IoT cihazların kullanım durumlarına göre farklı veri işleme araçları mevcuttur.

3.4. API'LER

API, Application Programming Interface'in (Uygulama Programlama Arayüzü) kısaltmasıdır. API, türleri farklı olan uygulamaların aynı yapı içinde iletişime geçerek çalışmasını sağlayan bir yazılımdır. Farklı yazılımların haberleşmesini sağlayan arayüzdür. API'ler farklı türden yazılımları ortak bir noktada buluşturarak oluşabilecek karmaşıklıkları önler.

3.4.1. API'lerin Çalışması

Kullanıcılara birçok kolaylık sağlayan API'ler ile internet ortamında birçok alanda karşılaşılır. Bu kadar sık kullanılan API'lerin çalışma mantığı şu şekilde açıklanabilir:

- Veriler, kullanılan uygulama tarafından bir sunucuya gönderilir.
- Verileri alan sunucu, bu verileri yorumlar ve gerekli işlemleri yaptıktan sonra tekrar uygulamaya gönderir.
- Gelen veri kümesi uygulama tarafından yorumlanarak kullanıcının anlayacağı şekilde gösterilir. Böylece API'nin çalışması tamamlanır.

Kullanıcıların işlemleri ayrı ayrı yapması gereken durumlarda API'ler kullanılarak işlemler tek seferde gerçekleşir. Uygulamanın kendisine giriş yapma ihtiyacı duyulmadan API tarafından sunulan özellikler, fonksiyonlarla veri toplama ve gönderme işlemi yapılabilir ancak genellikle API'lerle sağlanan erişimler belirli kısıtlamalarla gerçekleşir ve loglanır.

API'ler sağladıkları kolaylıklar ile birçok alanda kullanılır. Anlık mesaj gönderilmesi, sosyal medya platformları bunların başında gelir. Bunun yanında yine web tarayıcıları, işletim sisteminin API'leriyle depolama alanlarını, ses, kamera ve daha birçok donanımını kullanabilir. İnternet ortamında API'lerle çok sık karşılaşılır. Yazılım şirketlerinin sınırlı veya sınırsız kullanımlı API hizmeti vermesi buna bir etkendir. Birçok program geliştirici, bedeli karşılığında bu API'leri kullanır.

API'leri kullanabilmek için yapılması gerekenler şunlardır:

- API hizmeti veren şirketin web sitesinde mevcut bulunan API doküman sayfasından API'yi kullanmak için "API key" edinilir. API key, uygulamanın verdiği hizmete erişim için kullanılır.
- Geliştirici bölümünde yer alan bilgilerin kullanımı ile API ve uygulama arasında bağlantı kurulduktan sonra API kullanıma hazır hâle gelir. Böylece ana uygulamadaki fonksiyonlardan API'nin kullanıldığı uygulamalarda yararlanılır.

3.4.2. SOAP API

SOAP (Simple Object Access Protocol); merkezi olmayan, dağıtılmış bir ortamda bilgi alışverişi için kullanılan hafif XML tabanlı bir protokoldür. Dil ve platformdan bağımsız iletişim sağlar. SOAP bilgi alışverişi için kullanıldığından SOAP mesajları bir istek / yanıt düzeninde iletilir. SOAP kapsayıcısı, gelen SOAP isteklerini kabul eden ve bunları yayınlanmış bileşenlere gönderen bir SOAP uygulamasıdır. SOAP kapsayıcısı, istekleri SOAP ile bileşenlerin ana dili arasında otomatik olarak çevirir. SOAP kapsayıcıları birçok programlama diliyle uyumludur (Görsel 3.21).

Bir SOAP mesajının yapısı aşağıdaki şekildedir.

- **Envelope** : Tüm SOAP servis istek ve cevaplarının bilgilerini içerir. SOAP servisinin XML root elemanı olmak zorundadır. İçeriğinde header ve body olma durumundadır.
- **Header** : Her zaman olmasa da meta-data gibi bilgileri iletmeye yarar.
- **Body** : Her zaman SOAP mesajlaşmada bulunur. Yapılan istekte mesaj adı ve parametreleri XML formatında iletilir ve sunucudan gelen cevap da Body kısmında XML olarak geri döndürülür.
- **Fault** : Yapılan istek sonucu dönülen hata ve durumunu içerir. Fault tagi ile WSDL içeriğinde bilgisi verilir.

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2003/05/soap-envelope/"
soap:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
<soap:Header>
...
</soap:Header>
<soap:Body>
...
  <soap:Fault>
  ...
  </soap:Fault>
</soap:Body>
</soap:Envelope>
```

Görsel 3.21: SOAP API

3.4.3. REST API

REST (REpresentational State Transfer), Client-Server iletişimi yapabilecek bir mimaridir. REST, 2000 yılında Roy Fielding tarafından doktora tezi olarak hazırlanmıştır. Günümüzde SOAP ve WSDL tabanlı web servislerine alternatif olarak kullanılır. SOAP'taki sabit URL ve metotlar ile kurulan iletişim yerine REST'te değişken URL ile metotlar üzerinden iletişim kurulur. Bu sayede REST API'lerin kendilerine özel URI (Uniform Resourde Identifier) bilgileri vardır. Bu bilgiler sayesinde kaynak tam olarak işaret edilir. REST, HTTP üzerinden platform bağımsız çalışır ve veri akışını en az yükle sağlar. REST API'leri JSON, XML gibi çeşitli türde çıktı verebilir. Az yer kaplaması ve birçok platformda kullanılması dolayısıyla JSON en çok tercih edilen çıktı türüdür.

REST mimarisinde HTTP metotları GET, POST, PUT ve DELETE ile desteklenir.

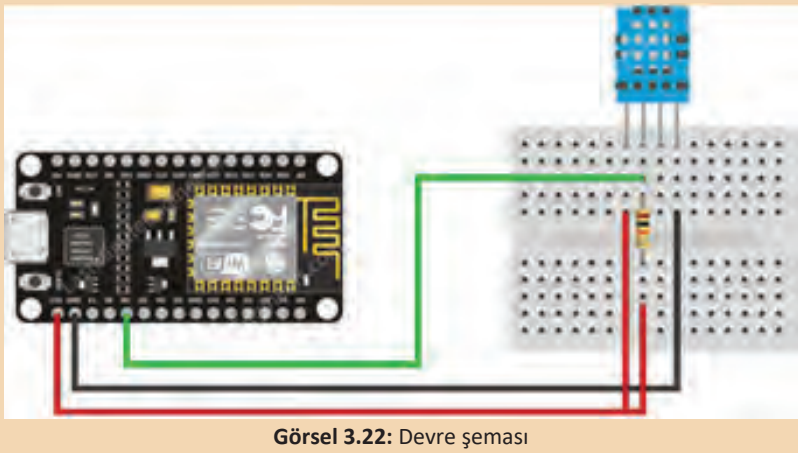
- **GET**: Veri listelemede, görüntülemede kullanılır. Listelenmek istenen veri ile ilgili bilgiler URL'de açık bir şekilde görülür.
- **POST**: Yeni bir veri oluşturmak için kullanılır ve yeni oluşturan verinin URL'si döndürülür.
- **PUT**: Bütün bir verinin bir başka veri ile güncellenmesi için kullanılır.
- **DELETE**: Belirtilen veriyi silmek için kullanılır.

7. UYGULAMA

NodeMCU ve DHT 22 sensörü ile sıcaklık ve nem değerlerinin ölçümü yapılarak sıcaklık ve nem değerlerinin web ortamında yayınlanmasını sağlayan API uygulamasını yapınız.

Gerekli Malzemeler

- NodeMCU
- DHT 22 sensör
- Breadboard
- 1 adet 1 k ohm direnç
- Erkek-dişi atlama kablosu

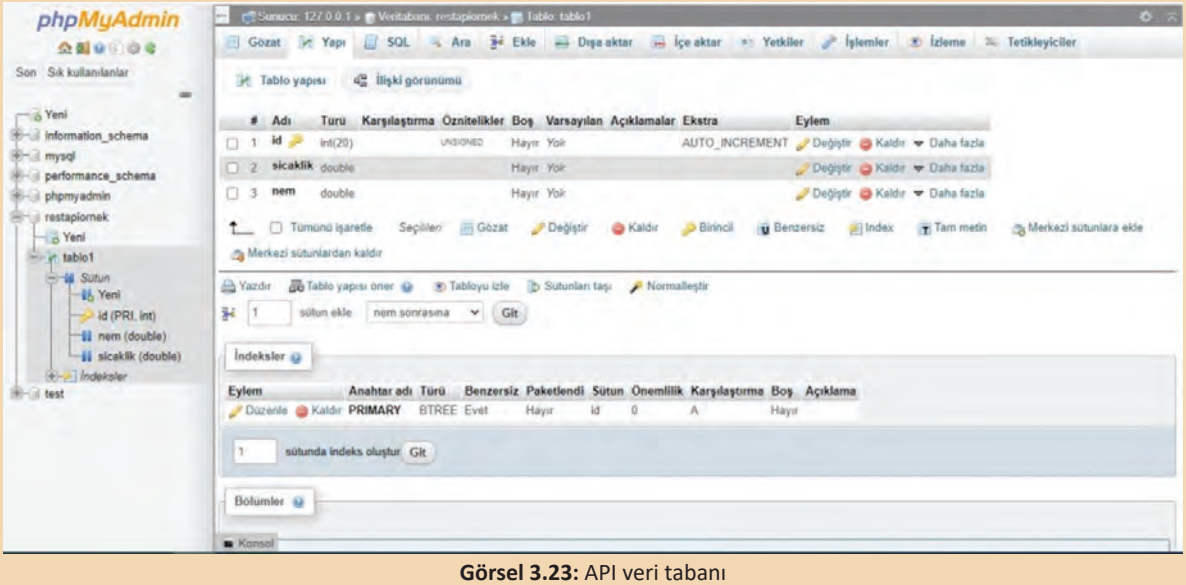


Görsel 3.22: Devre şeması

Görsel 3.22'de gösterilen devre kurulduktan sonra Arduino IDE editörü, yerel makinede XAMPP sunucu ve bir metin düzenleyici kullanılacaktır.

Sunucu Tarafı Programı

1. Adım : Verilerin saklanması için "restapiornek" adında bir veri tabanı oluşturunuz. Bu uygulamada kullanılacak olan veri tabanı Görsel 3.23'te gösterilmektedir. Uygulamada sıcaklık ve nem değerleri veri tabanında saklanacaktır.



Görsel 3.23: API veri tabanı

2. Adım : Yerel makine üzerinde çalışan sunucuda veri tabanı bağlantısının sağlandığı Görsel 3.24'te gösterilen kodlar ile veritabani.php dosyasını oluşturunuz.

C:\xampp\htdocs**webapi\config**

```php
<?php
    class Database {
        private $host = "localhost";
        private $database_name = "restapiornek";
        private $username = "root";
        private $password = "";
        public $conn;
        public function getConnection(){
            $this->conn = null;
            try{
                $this->conn = new PDO("mysql:host=" . $this->host . ";dbname=" . $this->database_name,
                $this->username, $this->password);
                $this->conn->exec("set names utf8");
            }catch(PDOException $exception){
                echo "Database could not be connected: " . $exception->getMessage();
            }
            return $this->conn;
        }
    }
?>
```

Görsel 3.24: Veri tabanı bağlantısı

3. Adım : Veri tabanına kayıt oluşturmak için bir sınıf oluşturunuz. Görsel 3.25'te bu sınıfa ait kodlar gösterilmektedir.

```php
<?php
    class Veritabani_kayit{
        private $conn;
        private $db_table = "tablo1";
        public $id;
        public $sicaklik;
        public $nem;
        public function __construct($db){
            $this->conn = $db;
        }
        public function createLogData(){
            $sqlQuery = "INSERT INTO
                        ". $this->db_table ."
                    SET
                        sicaklik = :sicaklik,
                        nem = :nem";
            $stmt = $this->conn->prepare($sqlQuery);
            $this->sicaklik=htmlspecialchars(strip_tags($this-> sicaklik));
            $this->nem=htmlspecialchars(strip_tags($this-> nem));
            $stmt->bindParam(":sicaklik ", $this->sicaklik);
            $stmt->bindParam(":nem ", $this-> nem);
            if($stmt->execute()){
               return true;
            }
            return false;
        }
    }
?>
```

Görsel 3.25: Veri tabanı kayıt oluşturma

GET metodunun uygulanacağı REST API uygulaması Görsel 3.26'da gösterilmektedir. Bu API'de GET metodu ile mysql veri tabanına NodeMCU ile okunan sıcaklık ve nem değerleri, internet üzerinden yerel makine üzerinde bulunan sunucu üzerindeki veri tabanına kaydedilir.

GET - http://localhost/webapi/api/index.php?sicaklik=22.5&nem=50

```php
<?php
    header("Access-Control-Allow-Origin: *");
    header("Content-Type: application/json; charset=UTF-8");
    include_once '../config/veritabani.php';
    include_once '../class/Veritabani_kayit.php';
    $database = new Database();
    $db = $database->getConnection();
    $item = new Veritabani_kayit($db);
    if ($_SERVER['REQUEST_METHOD'] === 'GET'){
        // The request is using the GET method
        $item->sicaklik = isset($_GET['sicaklik']) ? $_GET['sicaklik'] : die('Hatali yapi!');
        $item->nem = isset($_GET['nem']) ? $_GET['nem'] : die('Hatali yapi!');
    }else {
        die('Hatali istek metodu');
    }

    if($item->createLogData()){
        echo 'Veri basarili bir sekilde olusturuldu.';
    } else{
        echo 'Veri olusutulamadi.';
    }
?>
```

Görsel 3.26: Restapi uygulaması

GET yöntemi, veri istekleri için yaygın olarak kullanılır. Aynı zamanda veri depolamak için de kullanılabilir. NodeMCU üzerinden gelen sensör verileri sunucuda işlenmek üzere URL'de saklanır.

NodeMCU içine yazılan programda ilk olarak NodeMCU kartı Wi-Fi ağına bağlanır. Wi-Fi ağına bağlandıktan sonra sensörlerden veri okunur. Okunan sensör değerleri GET metodu ile yerel makinede bulunan sunucu üzerinde çalışan REST API'ye gönderilir. Gelen değerleri REST API ayrıştırır ve mysql veri tabanına kaydeder.

NodeMCU üzerine yazılan kodlar üç kısımda açıklanabilir. İlk olarak tanımlama kısmı, ikinci olarak ayarlama(setup) ve son olarak GET metodu ile veri gönderilen loop() açıklanacaktır.

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include "DHT.h"
#define DHTPIN 4
#define DHTTYPE DHT22
const char* ssid = "Senin Wifi SSID Adı";
const char* password = "Senin Wifi Şifre";
DHT dht(DHTPIN, DHTTYPE);
```

Görsel 3.27: Tanımlama kodları

Görsel 3.27'de tanımlama kodları gösterilmektedir. Bu kısımda kullanılacak kütüphaneler ve global değişkenler tanımlanmıştır.

```
void setup() {
  Serial.begin(9600);
  //Serial.println(F("DHTxx test!"));
  dht.begin();
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
    int i=0;
    while(WiFi.status() != WL_CONNECTED){
      Serial.print(".");
      delay(1000);
    }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.println();
  delay(2000);
}
```

Görsel 3.28'de ayarlamalara ait kodlar gösterilmektedir. Bu kısımda Wi-Fi bağlantısı ve sensör bağlantısı oluşturulup seri port ekranında da kontrol işlemi gerçekleştirilmektedir.

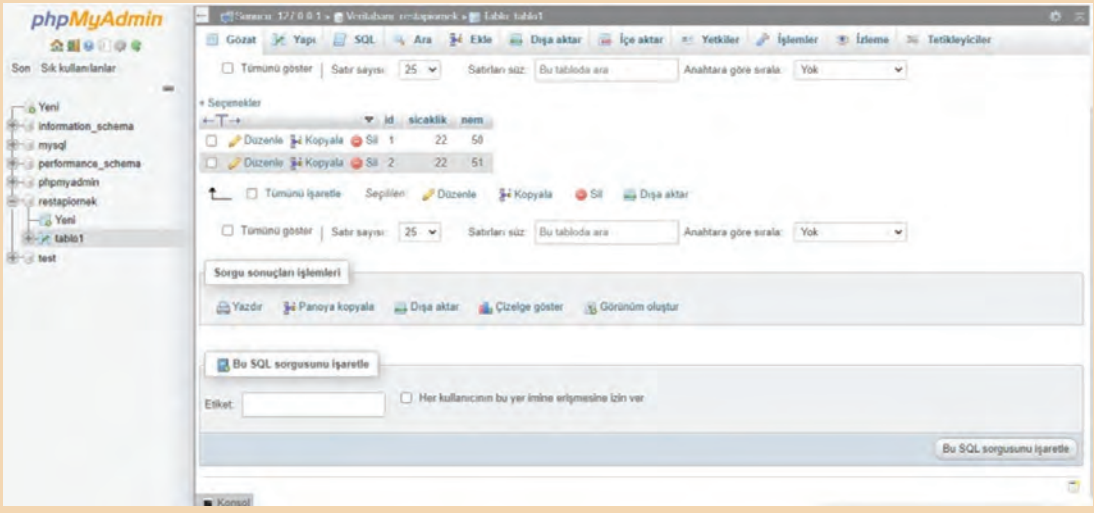
Görsel 3.28: Ayarlama kodları

Görsel 3.29'daki sıcaklık ve nem değerleri okunarak URL ile sunucuda bulunan API'ye gönderilerek veri tabanına kayıt işlemi gerçekleşir.

```
void loop() {
  double nem = dht.readHumidity();
  double sicaklik = dht.readTemperature();
    if (isnan(sicaklik) || isnan(nem)) {
    Serial.println(F("DHT sensörden okuma başarısız!"));
    return;
  }
  Serial.print(F("Sicaklik:"));
  Serial.print(sicaklik);
  Serial.print(F("°C   Nem:"));
  Serial.print(nem);
  Serial.println(F("%"));
  String address;
  address ="http://192.168.0.8/webapi/api/index.php?sicaklik=";
  address += String(sicaklik);
  address += "&nem=";
  address += String(nem) ;
  HTTPClient http;
  http.begin(address);
  int httpCode = http.GET();
  String payload;
  if (httpCode > 0) {
      payload = http.getString();
      payload.trim();
      if( payload.length() > 0 ){
         Serial.println(payload + "\n");
      }
  }
    http.end();
  delay(60000);
}
```

Görsel 3.29: Sıcaklık ve nem okuma kodları

Uygulama ile çalıştırılan kodların sonuçları Görsel 3.30'da gösterilmektedir. Veri tabanına 60 saniye aralıklarla veriler kaydedilir.



Görsel 3.30: Veri tabanı kayıtları

3.4.4. HTTP Durum Kodları

API'lerden dönen çeşitli HTTP durum kodları ile yapılan işlemlerin başarılı olup olmadığı, meydana gelen hataların nedenleri öğrenilebilir.

HTTP durum kodları kategorileri şunlardır:

- **1xx:** Bilgilendiricidir (Örnek: 101 Anahtarlama protokolleri).
- **2xx:** Başarılı işlemlerin habercisidir (Örnek: 202 Kabul).
- **3xx:** Yönlendirme amaçlı kullanılır (Örnek: 305 Proxy kullanımı).
- **4xx:** İstemci kaynaklı hatalardır (Örnek: 400 Hatalı istek).
- **5xx**: Sunucu kaynaklı hatalardır (Örnek: 505 HTTP sürümü desteklenmiyor.).

3.4.5. RESTful API

RESTful, REST mimarisini temel alır. Client-Server arası platform bağımsızdır. Herhangi bir Clinet sunucudaki RESTful servisine adres çubuğuna yazılarak bir web isteği gibi bağlanılabilir. Örneğin **www.alanadi.com/malzeme/12345** gibi bir URL verilip bu adres çağrıldığında ID değeri 12345 olan malzemelerin detayları JSON formatında geri döner. Bu veri uygun formata dönüştürülüp kullanıcıya sunularak istenilen detay görüntülenir.

RESTful servislerinin taşıması gereken karakteristik özellikler şunlardır:

- **Client-Server** : İstemci ve sunucu birbirleri hakkında bilgi sahibi değildir. İstemci veri talep eder, sunucu da bu bilgiyi sağlar.
- **Stateless** : Sunucu tarafında istemci ile ilgili hiçbir bilgi tutulmaz.
- **Cacheable** : Sunucu tarafından gönderilen yanıtta önbelleklenebilirlik durumu header (başlık) kısmında gönderilir ve istemci, gönderilen veriyi önbellekleyebilir. Başlıkta gelen bilgiye göre istemci veriyi belli bir süre önbellekte tutabilir veya tutamaz.
- **Layered System** : İstemcinin sunucu tarafında hangi katmana bağlandığının bilinmemesidir.
- **Uniform Interface** : İstemci ile sunucu arasında ortak bir URI formatında arayüz bulunmasıdır.
- **Code on Demand** : Sunucular, istemcinin fonksiyonelliğini geçici olarak icra ederek kod yollayabilir, genişletebilir ya da özelleştirebilir. Opsiyonel bir özelliktir. Örneğin web tarayıcılarından bir site üzerinde işlem yapıldığında bilgisayardaki bir uygulamayı açma isteğini göndermesi olarak ifade edilebilir.

3.5. KOD GÜVENLİĞİ

Nesnelerin İnterneti ile günlük yaşamda kullanılan birçok cihazdan (akıllı saat, buzdolabı, kahve makinesi, sağlık, ulaşım vb.) veriler paylaşılır. Farklı türdeki IoT cihazların paylaştığı veriler, kullanıcı gizliliği ve mahremiyeti konusunda endişeler oluşturabilir. IoT cihazlarının sayısı günden güne artmaktadır. Herhangi bir IoT cihazındaki bir güvenlik açığı, aynı türdeki bir çok IoT cihazını etkilenmesine sebep olmaktadır. Örneğin Ekim 2016'da sunucu hizmeti veren bir şirketin sunucuları üzerinden güvenli olmayan IoT cihazların zararlı yazılım gibi kullanılması ile yapılan büyük boyutlu DDoS atağı, ABD'deki birçok popüler on-line sitenin devre dışı kalmasına neden olmuştur. Bu atakta 150.000 güvenli olmayan IoT cihazın kolayca kullanılabildiği görülmüştür. 2015 yılında "Ortadaki Adam Saldırısı (Man In The Middle)" ile bir aracın klima, radyo, cam silecekleri gibi ekipmanlarının kontrolü sağlanmış ve bir milyonun üzerinde araç güvenlik açığı nedeniyle geri çağrılmıştır.

Bu saldırılar IoT cihazların güvenliğinin ne kadar önemli olduğunu bir kez daha göstermiştir. IoT cihazlarda güvenlik yazılımı eksikliği, daha az deneyimli cihaz üreticileri, aynı güvenlik özelliklerine sahip fazla miktarda cihaz olması bilgisayar korsanlarını bu cihazlara yönlendirmiştir. 2019 yılı verilerine göre IoT cihaz saldırıları bir önceki yıla göre %900 artış göstermiştir. Bilgisayar korsanları tarafından ele geçirilmiş bir IoT cihazın tespit edilmesi çok uzun zaman alır ve cihaz, sahibi tarafından fiziksel olarak kapatılmadan çalışmaya devam eder.

IoT cihazlarda güvenliğin sağlanması için dikkat edilmesi gereken hususlar şunlardır:

- Cihazlar yetkili kişiler hariç dışarıdan müdahalelere karşı korumalı olmalıdır.
- Cihazlar kullanıldıkça var olan güvenlik açıkları tespit edilmeli, kullanılan IoT cihazla ilgili güncellemeler takip edilmeli ve cihazlarda güncelleme yapılmalıdır. Bu cihazlarda güncelleme dışarıdan müdahale anlamına geldiğinden zararlı yazılımlardan korunmak için dijital imza, sertifikasyon gibi güvenlik önlemleri uygulanmalıdır.
- Cihazların dinamik testlerden geçirilmesi önemlidir. Dinamik test hem kod hem de donanım açıklarının tespit edilmesini sağlar.
- Kullanım ömrünü tamamlayan IoT cihazlar, içindeki verilere ulaşılamayacak şekilde imha edilmelidir.
- IoT cihazlara erişim için kullanılan kimlik doğrulama (kullanıcı adı ve şifresi) güçlü olmalıdır. Güçlü şifreleme ve güvenlik protokolleri kullanılmalıdır. Cihazların parolaları güvenli olsa bile cihazlar arasındaki iletişim hacklenebilir. Haberleşmede kullanılan protokollere uygun olarak şifreleme yapılmalıdır.

Nesnelerin İnterneti ortamlarında bulunan bazı zafiyetler şunlardır:

İnjeksiyon Zafiyetleri : İnjeksiyon saldırılarının temel amacı sisteme zararlı kodlar sızdırmak veya sistemden veriler elde etmektir. IoT nesnelerinde ise asıl amaç kullanıcının özel anahtarını (API KEY) ele geçirip bilgileri istediği gibi değiştirip silerek sisteme zarar vermektir. En riskli açıklar injeksiyon açıklarıdır.

Bu açıklar web uygulamalarının kodlama sürecindeyken yapılan hatalardan ve dikkatsizlikten meydana gelir. İnjeksiyon açıkları ile hedef sistemde veri okunabilir, silinebilir veya veriye zarar verilebilir. Örneğin akıllı ev otomasyonu sistemlerinde evin içinde IoT nesneleri olduğu düşünülürse kullanıcı giriş bilgileri paneli web üzerinde tutulduğunda eğer kodları yazan kişi gerekli filtreleme önlemlerini almazsa saldırgan sisteme giriş yapar ve akıllı ev otomasyonunda bulunan verileri kullanarak sisteme zarar verebilir.

İnjeksiyon çeşitleri aşağıdaki türlere ayrılır:

a) **Sql İnjeksiyon** : SQL veri tabanlarına erişim ve yönetim için kullanılan standart bir yapıdır. Veri tabanına SQL ile veriler işlenirken araya birtakım karakterlerin eklenmesiyle Sql injeksiyon meydana gelir. Sql injeksiyon ile saldırı yapılan sitede veri tabanında veriler okunabilir, okunan veriler ile panellere erişim sağlanabilir, uzaktan kod vs. çalıştırılabilir. Sql injeksiyonun da kendi arasında çeşitleri vardır. Bunlar kısaca Classic Sql, Error-based Sql, Union-based Sql, Blind Sql, Boolean-based Sql, Time-based Sql, Out-of-band Sql'dir.

b) **Command İnjeksiyon** : Genellikle sunucu bazlı çalışan uygulamaların gerekli filtrelemelerden geçirilmeden direkt olarak kodla çalıştırılmasına dayanır. Çalıştırılan bu komutlar serverda hâkimiyet sağlar. Bu zafiyetle beraber sistemde sistem yöneticisi olunabilir ve sistem uzaktan yönetilebilir. Uzaktan yönetilen IoT sistemlerindeki veriler, bu yöntem ile manipüle edilebilir.

c) **XPATH İnjeksiyon** : XML, kullanıcıların verilerini kendi kendine yapılandırmasına olanak sağlayan bir dildir. XPATH, xml verileri üzerinde herhangi bir değişiklik yapılmaksızın veri almak / okumak için kullanılan bir dildir. Kullanıcı girdilerinden XPATH sorguları oluşturan web sitelerine sql injeksiyondaki gibi bazı karakterlerin araya sıkıştırılmasıyla oluşur. Sql injeksiyona benzer bir yapıdadır.

ç) **LDAP İnjeksiyon** : LDAP, dizin hizmetlerinin yönetiminde kullanılan bir protokoldür. LDAP ile dizinlerde kayıt ekleme, silme, düzenleme, arama gibi işlemler yapılabilir. LDAP cümleciklerinin belirli bir denetlemeye tabi tutulmadığı uygulamalarda LDAP cümleciklerinin değiştirilmesi ile saldırı gerçekleşir. LDAP ile sql injeksiyonda olduğu gibi veriler okunabilir, değiştirilebilir, yetkisiz kullanıcılara yetkiler verilebilir. IoT'ta kullanıcı yetkilendirmeleri olan bir sistem mevcutsa bu yöntem ile yetkiler değiştirilebilir. Örneğin savunma sanayine yönelik uzaktan yönetimli bir güvenlik cihazı olduğu düşünülürse bu cihaza web panelinden erişen kullanıcılar olabilir. Bazı kullanıcılar cihazın kullanım hakkına sahipken, bazıları ise cihazın sadece görüntüleme hakkına sahiptir. LDAP açıklarına sahip bir kodlama ile hazırlanan sistemde saldırgan yetkisiz alanlara ulaşarak o yetkilere erişebilir ve geri dönülemez sorunlar ortaya çıkarabilir.

d) **PHP Object İnjeksiyon** : PHP'de kullanıcıdan alınan verinin "unserialize()" fonksiyonundan geçirilmesi sonucu ile oluşur.

e) **SSI İnjeksiyon** : SSI, web uygulamalarında statik yapılı sayfalara dinamik içerik eklemeyi sağlayan yapıdır. Server tabanlı çalışmakta olup servera gönderilen zararlı kodların çalışmasıyla beraber SSI injeksiyon oluşur.

Hatalı Kimlik Doğrulama ve Oturum Yönetimi Zafiyeti : Oturum saldırılarına yönelik gerçekleştirilen saldırı türleridir. Buna en basitinden oltalama (phishing) saldırıları örnek olarak gösterilebilir. Bu yöntem, hedef kitleye yönelik sahte sistemler hazırlanıp kişilerin yemlenmesi esasına dayanır. Buradaki temel amaç kullanıcı hesap bilgilerini veya kredi kartı bilgileri gibi bilgileri elde etmektir. IoT uygulamalarında kredi kartı bilgilerine ulaşabilen bir IoT sisteminde akıllı buzdolabı, akıllı araç sistemleri gibi bu sistemlerde kullanıcı hesap bilgileri ve kredi kartı bilgileri bulunur. Saldırgan bu açıkları kullanarak oturum bilgilerine ulaşabilir ve maddi zarar verebilir.

Cross-Site Scripting (XSS) Zafiyeti : XSS zafiyeti html, css, javascript ile hazırlanmış zararlı kodların kullanıcıların tarayıcısında izinsiz olarak çalıştırılmasına olanak sağlayan bir açıktır. XSS ile hedef kullanıcının oturum bilgileri, tuş girişleri, tarayıcı yönetimi gibi işlevler gerçekleştirilebilir. Dışarıdan girilen değerlerin filtrelenmemesi sonucunda ortaya çıkan bir zafiyettir. XSS'te kendi içinde üç bölüme ayrılır. Bunlar; Reflected XSS, Dom Based XSS ve Stored XSS'tir.

Güvensiz Doğrudan Nesne Referansları Zafiyeti : Güvensiz bir şekilde direkt doğrudan nesnelere erişim sağlanabilir. Genellikle her nesnenin bir id değeri veya buna karşılık gelen bir yapısı vardır. Bu yapılar örnek olarak gösterilirse php_id=1 olduğu varsayıldığında normal kullanıcılar id değeri 1 olan nesneye erişim sağlarken, id değeri 2 olan nesneye erişim sağlamaması gerektiği hâlde erişim sağlamasından ortaya çıkan zafiyettir. Tamamen sistemdeki yetkilendirilmelerden kaynaklanan bir durumdur.

Hatalı Güvenlik Yapılandırması Zafiyeti : Sistemlerde bulunan güvenliklerin yanlış yapılandırılmasından ortaya çıkan zafiyettir. Genellikle sunucu yapılandırılmalarında varsayılan ayarlar kullanılır. Varsayılan ayarlar güvenli değildir. Bunu önlemek için güvenlikler doğru bir şekilde yapılandırılmalı ve gereksiz servislere yer verilmemelidir. IoT sunucularında yapılan hatalı bir güvenlik duvarı yapılandırması bu zafiyeti kullanarak kullanıcıların zarar görmesine sebebiyet verebilir.

Hassas Veri Zafiyeti : Hassas verilere yönelik olan saldırılardır. Hassas veriler genellikle yedekler, kimlik bilgileri, kredi kartı gibi verilerdir. Bu teknikler genellikle loglama, bruteforce, tarayıcı zafiyetleri, arp spoofing gibi saldırılara dayanır.

Cross-Site Request Forgery (CSRF) Zafiyeti : CSRF saldırıları güven saldırıları olarak da bilinir. Buradaki temel hedef sitede yetkili bir kullanıcıya işlev yaptırmaktır. Bu zafiyet türü bir örnek üzerinden şöyle açıklanabilir. Hedef site http://hedef-IOTsite.com ve sitede yüzlerce kullanıcı aynı zamanda birkaç admin olduğu varsayılsın. Saldırganın temel hedefi site üzerinde yetkili yani admin olmak istemesidir. Daha önceden hazırlamış olduğu html ile panel yönetiminden gerekli ayarlamaları yapılmış bir script ile hedef admine yedirmesiyle işlevlerin gerçekleştirilmesi sonucu ortaya çıkan zafiyettir.

URL Yönlendirme Zafiyetleri : Günümüzde hemen hemen herkesin denk geldiği sahte link yönlendirmeleri ve fake siteler bu başlık altında incelenir. Özellikle sosyal medyada bu yöntemler çok kullanılır. IoT sitelerinin

fake siteleri oluşturularak bu zafiyet ile saldırı düzenlenebilir.

! DİKKAT

Bu tarz saldırılarda çözüm için bilgi girişi yapılan nesnelerden alınan bilgiler filtrelenmeli, daha sonra SQL ifadeleri ile birleştirilmelidir. Örneğin kullanıcı adı ve şifre girişi yapılan bölümlerden alınan veriler, SQL ifadesi ile birleştirilmeden önce filtrelenmelidir. Filtrelenme en önemli çözüm tekniğidir.

3.5.1 HTML İnjeksiyon Zafiyeti

HTML injeksiyon olarak adlandırılan bu zafiyet siteye dışarıdan herhangi bir kişinin html kodu enjekte etmesine olanak tanıyan bir güvenlik açığıdır. HTML injeksiyon, kullanıcıların dışarıdan veri girdisi yaptığı bütün form işlemlerinde oluşabilen bir zafiyettir. Web sitesi içinde bulunan bir yazı kutusu, bir liste kutusu, arama kutusu, yorum yapma alanları gibi veri girdisinin yapılacağı alanlarda kullanılan bir açıktır. XSS ile benzerlik gösteren özellikleri olsa da HTML injeksiyon sadece HTML etiketleri ile kullanılır. Html komutları kullanılarak hazırlanmış olan bir IoT sisteminde bu açıklar kullanılarak saldırı düzenlenebilir.

HTML İnjeksiyon Zafiyeti kullanılarak aşağıdaki işlemler yapılabilir.

- IoT Web sayfasının içerik bilgilerini değiştirme
- IoT kullanıcı oturum verilerini elde etme
- CSRF karşıtı işlemlerin keşfi
- Tarayıcıda kaydedilen parolaları elde etme

IoT Web Sayfasının İçerik Bilgilerini Değiştirme : En basit saldırı tekniklerinden biridir. Saldırganın zafiyetli site üzerinde sitenin görünürlüğünü ya da içinde ekli olan dosyaları, görselleri, yazıları değiştirmesi ile meydana gelen bir saldırı türüdür. Örneğin, saldırgan satmak istediği bir ürünün görsel reklamını eklemek için depolanan bir HTML eklemesi kullanabilir.

IoT Hassas Oturum Verilerini Elde Etme : Saldırgan bu saldırı tekniğinde sitenin içinde hazır olarak verilen form elementlerini kullanarak ya da kendi eklemiş olduğu html form kodları ile bir form sayfası oluşturabilir ve bu sayfaya girilen değerleri kendi local ağına yönlendirip kullanıcı verilerini çalmaya yönelik bir saldırı gerçekleştirebilir.

CSRF Karşıtı İşlemlerin Keşfi : Bu saldırı türünde saldırgan CSRF (Siteler Arası İstek Sahteciliği) olarak bilinen saldırı türünü kullanabilmek için CSRF karşıtı belirteçleri sızdırmaya çalışır. Bu konuda yararlanacağı kodlar HTML kodlarıdır ve bu saldırıyı gerçekleştirmek için HTML injeksiyon zafiyetinden yararlanır.

Tarayıcıda Depolanan Parolaları Elde Etme : HTML eklemeleri, saldırganlar tarafından tarayıcı parola yöneticileri tarafından otomatik olarak doldurulan formları yerleştirmek için de kullanılabilir. Saldırgan uygun bir form eklemeyi başarırsa parola yöneticisi kullanıcı kimlik bilgilerini otomatik olarak ekler.

HTML injeksiyon zafiyeti; kodları yazan kişinin özensiz, plansız ve güvenlik risklerini hiçe sayarak web sitesini geliştirmesi sonucu ortaya çıkar. PHP dilinin eski ve güvensiz sürümlerinin kullanılması, tarayıcı eklentileri ve eklentilerin sürümlerinin eski olması ayrıca HTML veri girişi alanlarının gerekli kontrollerden geçmeden kodlanması bu zafiyeti saldırganların kullanmasına olanak verir. Bu zafiyetin bir IoT sisteminde olması gerek maddi gerekse manevi kayıpların ortaya çıkmasına sebebiyet verebilir.

8. UYGULAMA

Hedefte olan bir akıllı ev sistemleri web sitesinin HTML açıklarını kullanarak kullanıcı verilerinin elde edilmesini sağlayınız.

1. Adım : Hedef sitede veri girişi yapılan alanlara HTML kodları girerek açık olup olmadığını deneyiniz (Görsel 3.31).
2. Adım : Hedef sitede veri girişi yapılan alanlara girilen kodların çalışıp çalışmadığını test ediniz. Kodlar çalışıyorsa açık var demektir ve işleme devam edilir (Görsel 3.32).

Görsel 3.31: HTML açık keşfi

Görsel 3.32: Girilen Html kodlarının çalışması ve açığın tespit edilmesi

3. Adım : Akıllı ev için hazırlanmış web sitesinde html injeksiyon zafiyetini tespit ediniz ve bu açığı kullanarak siteye girecek kullanıcıların bilgilerini ele geçirme amaçlı bir metin gönderip (Yeni sürüm hizmete girdi) o metinde bulunan adrese giriş yapmalarını sağlayınız (Görsel 3.33).

Örnek Html Kodu : <h1><mark>Dikkat!!!</mark>Yeni Sürüm Hizmetinize Girdi Yeni Sürüme Geçmek İçin TIKLAYINIZ</h1>TIKLAYINIZ

Görsel 3.33: Üyelerin giriş yaptığı sitede zafiyeti kullanma

Görsel 3.34: Sahte giriş paneli

4. Adım : Zafiyet kullanarak giriş yapan kişileri hazırlanan sahte siteye yönlendiriniz. Kişilerin bilgilerini girmesini isteyiniz ve verileri ele geçiriniz. Bu noktada saldırganın uygulayacağı sosyal mühendislik yöntemleri, inanılırlığını arttırarak verilerin ele geçirilmesini sağlayabilir (Görsel 3.34).

3.6. RASPBERRY PI KULLANIMI

Raspberry Pi, gelişmekte olan ülkelerin okullarında basit bilgisayar bilimini öğretmek amacıyla geliştirilen tek kartlı bilgisayardır (SBC- Single Board Computer). Raspberry Pi, küçük boyutu ve ucuz maliyetiyle çok fazla talep görmektedir.

3.6.1. Raspberry Pi Donanım Özellikleri

Görsel 3.35: Raspberry Pi 4

Raspberry Pi (Görsel 3.35), basit bilgisayar olarak geliştirilen bir kart olması sebebiyle klavye, mouse, kasa gibi dış bileşenleri bulundurmaz. Bu bileşenler haricen takılır. VideoCore IV GPU grafik işlem birimine sahiptir. Boot işlemleri ve veri depolama için Raspberry Pi'de SD kart kullanılır. Kart üzerinde USB 2.0-USB 3.0 portları, HDMI video çıkışı, ses çıkışı, MIPI kamera girişi, GPIO arayüzü ve 5V MicroUSB güç girişi mevcuttur. Raspberry Pi; Raspian, Piora, Snappy Ubuntu Core, Pardus ARM, Arch Linux ARm ve Windows 10 IoT Core işletim sistemlerini destekler. Programlama dili olarak Python, BBC Basic, C ve Perl kullanılır.

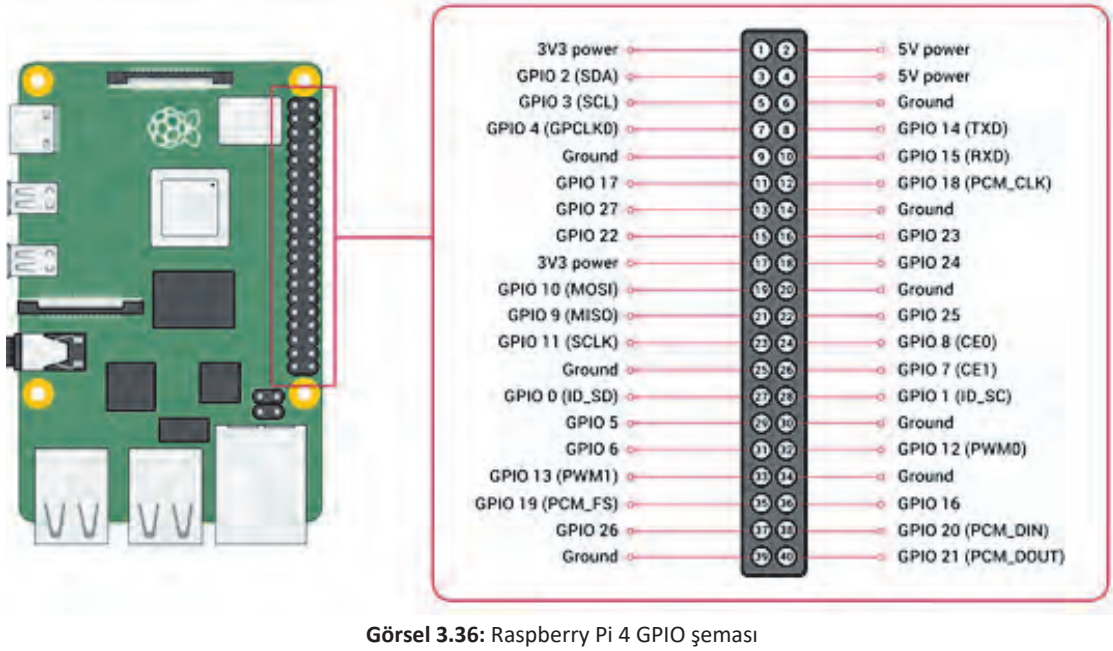
Raspberry Pi kartının Raspberry Pi 1A+ / 1B+, Raspberry Pi 2B, Raspberry Pi 3A+ / 3B / 3B+, Raspberry Pi 4B, Raspberry Pi Pico, Raspberry Pi Zero, Raspberry Pi Zero W modelleri mevcuttur. Bu modeller arasında boyut, bellek, CPU, USB portları, video çıkışı, depolama birimi, GPIO, ethernet ve sarf edilen güç cinsinden farklılıklar mevcuttur. Kullanım alanına ve projeye göre uygun kart seçimi yapılmalıdır.

Tablo 3.1'de Raspberry Pi elektronik kartının versiyon donanım değişimlerine yer verilmiştir.

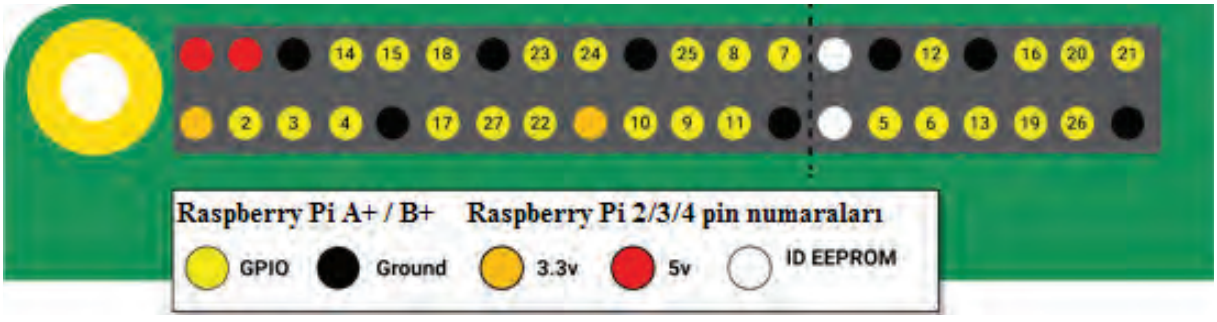
Tablo 3.1: Raspberry Pi Gelişimi

	İşlemci	RAM	USB Portu	Video Çıkışı	Ethernet	GPIO Pin Sayısı	Hafıza Birimi
Model A	700 MHz Tek çekirdek	256 MB	1 adet	Kompozit, HDMI	-	26 pin	SD kart
Model A+	700 MHz Tek çekirdek	256 MB	1 adet	HDMI	-	40 pin	mikroSD kart
Model B	700 MHz Tek çekirdek	512 MB	4 adet	Kompozit, HDMI	Ethernet	26 pin	SD kart
Model B+	700 MHz Tek çekirdek	512 MB	4 adet	HDMI	Ethernet	40 pin	mikroSD kart
Raspberry Pi 2	900 MHz Dört çekirdek	1GB	4 adet	HDMI	Ethernet	40 pin	mikroSD kart
Raspberry Pi 3	1.2 GHz Dört çekirdek	1GB	4 adet	HDMI	Ethernet, Wi-fi 802.11n, Bluetooth 4.1	40 pin	mikroSD kart
Raspberry Pi 3 B+	1.2 GHz Dört çekirdek	1GB	4 adet	HDMI	2.4 GHz ve 5 GHz IEEE 802.11b/g/n/ AC kablosuz LAN, Bluetooth 4.2	40 pin	mikroSD kart
Raspberry Pi Zero	1 GHz Tek çekirdek	512 MB	1 adet mikro USB	Mini HDMI	-	40 pin	mikroSD kart
Raspberry Pi 4 B	1.5 GHz Dört çekirdek	2GB, 4GB, 8GB	2 adet USB 2.0 2 adet USB 3.0	Mikro HDMI	Gigabit Ethernet	40 pin	mikroSD kart

Görsel 3.36'da ve Görsel 3.37'de pin numaraları ve çıkışları görülmektedir.



Görsel 3.36: Raspberry Pi 4 GPIO şeması



Görsel 3.37: GPIO pin numaraları

NOT

AnRaspberry Pi 400 : Klavyede tasarlanmış bir Raspberry Pi'dir. Raspberry Pi 400 için "kendi başına bir bilgisayar" denilebilir. Raspberry Pi 400, Raspberry Pi 4'ü temel alan amaca yönelik bir kart içerir.

Raspberry Pi Pico : Mikrodenetleyici kartı olarak tasarlanmıştır. C/C++ veya Micropython ile programlanabilir.

Raspberry Pi modelleri, mikroUSB tipinde bir güç beslemesi girişine sahiptir. Güncel Raspberry Pi versiyonlarında ise Type-C bulunur.

3.6.2. Raspberry Pi Donanımına İşletim Sistemi Kurulumu

Raspberry Pi kartına masaüstü işletim sistemi yüklemek için **https://www.raspberrypi.com/software/operating-systems/** sitesine giriş yapıldığında dört kurulum şekli ile karşılaşılır (Görsel 3.38).

Görsel 3.38: Kurulum tipi seçim sayfası

1. Raspberry Pi OS kurulum tipinde üç seçenek bulunur.
    - **Raspberry Pi OS with desktop:** En çok tercih edilen işletim sistemi tipidir. Arayüzü ve kullanım kolaylığı tercih edilme sebebidir.
    - **Raspberry Pi OS with desktop and recommended software:** Arayüz yanında bazı uygulamalar da mevcut olarak kurulur.
    - **Raspberry Pi OS Lite:** Arayüz mevcut değildir, tüm işlemler terminal aracılığıyla yapılır.
2. Raspberry Pi OS (Legacy) kurulum tipinde iki seçenek bulunur.
    - **Raspberry Pi OS (Legacy) with desktop:** Arayüze sahip eski kararlı Debian sürümünü içerir.
    - **Raspberry Pi OS Lite (Legacy):** Eski kararlı Debian sürümü terminaline sahiptir. Arayüz mevcut değildir.
3. **Raspberry Pi Desktop:** Masaüstü ve dizüstü (laptop) bilgisayarlara kurulum yapılabilecek Debian sürümünü içerir.
4. **Third-party software:** Raspberry Pi'de kullanılabilecek diğer işletim sistemlerini içerir.

Görsel 3.39: Raspberry Pi Imager

Raspberry Pi'de kullanıma hazır bir işletim sistemi yüklü microSD karta hızlı ve kolay yoldan ulaşmak için https://www.raspberrypi.org/software/ adresindeki Raspberry Pi Imager programı indirilip bilgisayara kurulur. İşletim sistemine uygun kapasitedeki mikroSD kart (İşletim sistemleri için minimum 8 GB, Lite versiyon işletim sistemleri için 4 GB mikroSD kullanılmalıdır.) okuyucuya yerleştirilir ve program çalıştırılır. Otomatik kurulum için kullanılan işletim sistemine göre SD kart okuyucuya yerleştirilir ve program çalıştırılır (Görsel 3.39).

Raspberry Pi Imager programı sayesinde seçili olan SD karta Raspberry Pi'de çalışacak kurulum tipleri listelenir. Kurulmak istenen kurulum tipi ve kurulum yapılacak SD kart seçildikten sonra YAZ butonuna tıklanarak işletim sisteminin kurulumuna başlanır (Görsel 3.40).

İşletim sistemini manuel kurmak için kurulmak istenen işletim sisteminin imajı indirilir ve Boot işlemini yapacak programlar yardımı ile SD karta kurulum sağlanır.

Görsel 3.40: Raspberry Pi işletim sistemi seçim ekranı

Görsel 3.41: İşletim sistemi ana ekran görüntüsü

SD kart Raspberry Pi'ye takılıp çalıştırıldığında Raspberry Pi kullanıma hazırdır (Görsel 3.41).

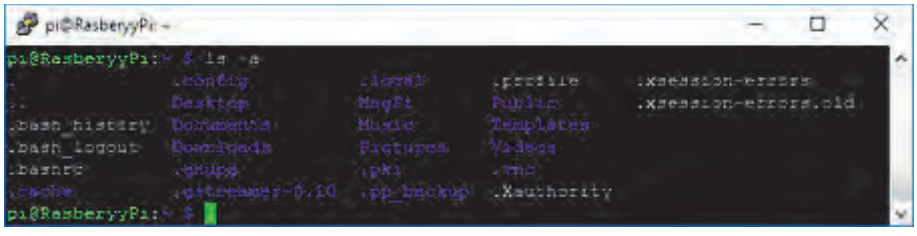
3.6.3. Raspberry Pi Donanımında Temel Linux Komutlar

Raspberry Pi'nin terminal ekranında temel Linux komutları kullanılarak işlemler gerçekleştirilir.

ls Komutu: Dizin içeriğini görmek için kullanılır (Görsel 3.42). Varsayılan olarak bulunulan dizinin içeriğini gösterir. Başka bir dizin içeriği gösterilmek istenirse dizin yolu yazılmalıdır.

ls komutu ile kullanılacak parametreler şunlardır:

- **-a** parametresi, gizli dosyalar dâhil bütün dosyaları gösterir.
- **-l** parametresi, liste biçiminde listeleme yapar.
- **-h** parametresi, dosya boyut bilgisini gösterir (KB, MB).
- **-R** parametresi, mevcut dizinden alt dizinlere kadar listeleme yapar.

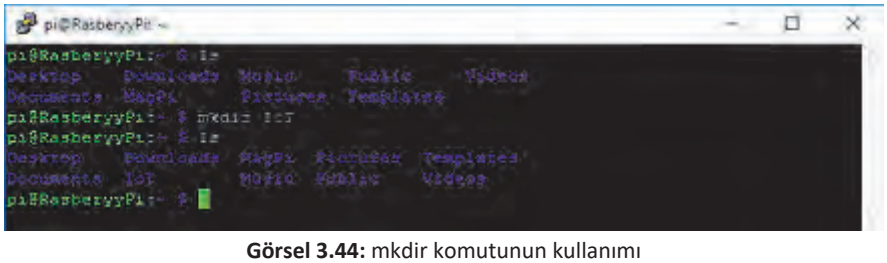


Görsel 3.42: ls komutunun kullanımı

pwd (print working directory) Komutu: Bulunulan dizini öğrenmek için kullanılır (Görsel 3.43).

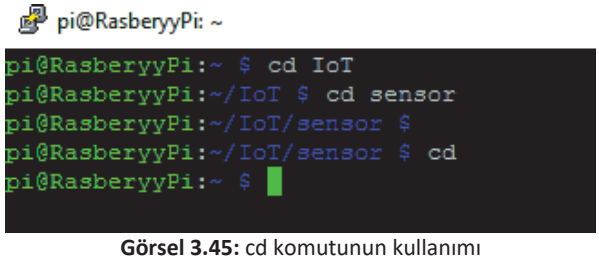
Görsel 3.43: pwd komutunun kullanımı

mkdir (make directory) Komutu: Dizin oluşturmak için kullanılır (Görsel 3.44). Ayrıca -p parametresi ile iç içe dizin oluşturulur (ör. mkdir -p IoT/sensor).



Görsel 3.44: mkdir komutunun kullanımı

cd (change directory) Komutu: Bulunulan dizinden başka dizine geçmek için kullanılır (Görsel 3.45). Örneğin cd IoT komutu ile mevcut bulunan IoT dizinine geçilir. Ana dizine dönmek için cd komutunun kullanımı yeterlidir.



Görsel 3.45: cd komutunun kullanımı

cp (copy) Komutu: Dosya ve dizinleri kopyalamak için kullanılır. Bu komutun kullanımı cp [kaynak] [hedef] şeklindedir. cp komutu ile -R parametresi kullanılır. -R parametresi ile kaynak içindeki dosya ve dizinlerle kopyalama işlemi gerçekleşir.

mv (move) Komutu: Dosya ve dizinleri taşımak için kullanılabileceği gibi dosya ve dizin isimlerini değiştirmek için de kullanılabilir. Bu komutun kullanımı mv [kaynak] [hedef] şeklindedir.

ln (link) Komutu: Dosya ve dizinler arasında link vermek amacıyla kullanılır.

Bu komutun kullanımı ln [kaynak link ismi] [hedef link ismi] şeklindedir.

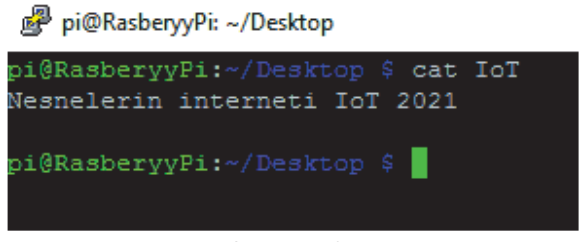
clear Komutu: Komut ve satırları temizler, imleci ilk satıra taşır.

rm (remove) Komutu: Dosya ve dizinleri silmek için kullanılır.

rm komutu ile kullanılacak parametreler şunlardır:

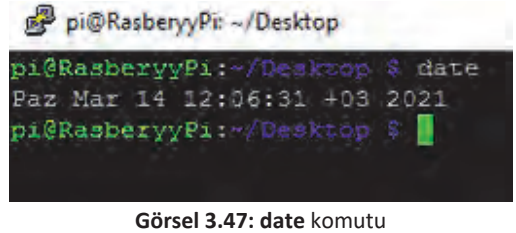
- **-r** parametresi, dizin silme işlemi esnasında silinen dizin içinde dosya ve dizinler varsa kullanıcıdan onay alarak silme işlemini yapar.
- **-f** parametresi, dosya işlemi silme esnasında kullanıcıdan onay almadan silme işlemini yapar.

cat (concatenate files) Komutu: Dosya içeriğini liste hâlinde gösterir (Görsel 3.46).



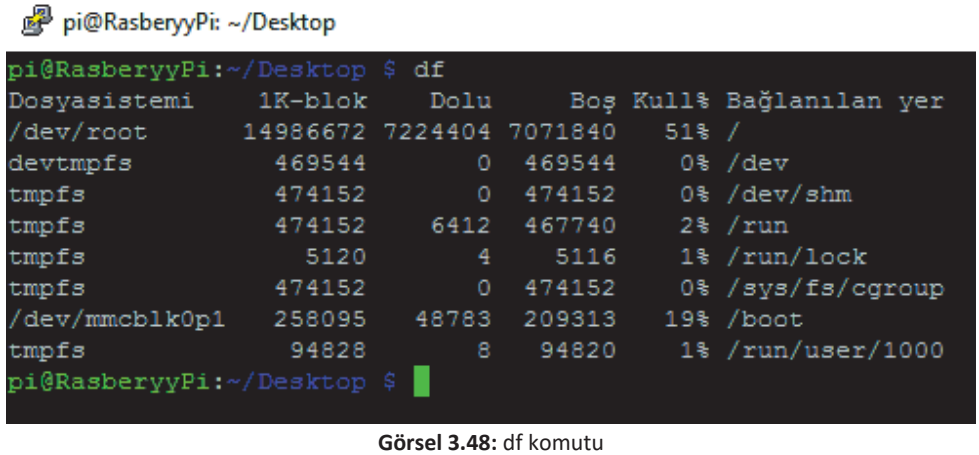
Görsel 3.46: cat komutu

date Komutu: Sistem tarihi ve saatini görüntülemek için kullanılır (Görsel 3.47).



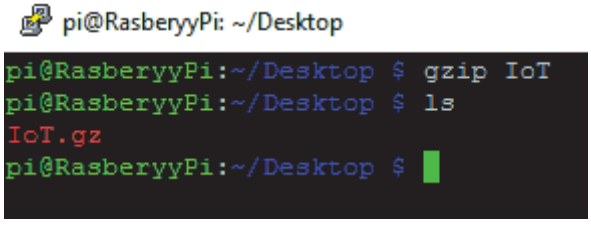
Görsel 3.47: date komutu

df (display file system) Komutu: Kullanılan disk (HDD, SD kart) kapasitesi ve boş alan boyutunu gösterir (Görsel 3.48).



Görsel 3.48: df komutu

gzip Komutu: Dizin ve dosyaları sıkıştırmak ve arşivlemek için kullanılır (Görsel 3.49).

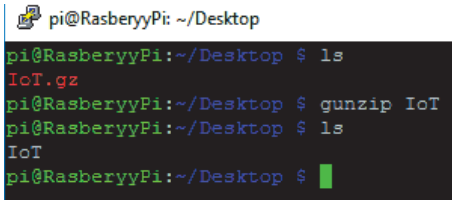


Görsel 3.49: gzip komutu

gzip komutu ile kullanılacak parametreler şunlardır:

-d parametresi, sıkıştırılmış dosya ve dizinleri açar.
-r parametresi, dizin adı ile kullanıldığında dizinin sahip olduğu tüm alt dizinlere bakar ve varsa tüm dosyaları ayrı ayrı sıkıştırır.

gunzip Komutu: Sıkıştırma işlemi uygulanan dosya ve dizinleri açmak için kullanılır (Görsel 3.50).



Görsel 3.50: gunzip komutu

tar Komutu: Sistemdeki dosya veya dizinleri beraber paketleyip arşiv oluşturur. Önceden oluşturulmuş arşivden dosyaları geri çıkartır. Oluşturulan arşivlerin uzantısı ".tar"dır.

Bu komutun kullanımı tar [parametre] arşiv_ismi arşiv_yapilacak_dosya/lar şeklindedir.

tar komutu ile kullanılacak parametreler şunlardır:

-c (create): Arşiv dosyası oluşturulacağını, uzantısının ".tar" olacağını belirtir.
-x (extract): Mevcut bulunan arşiv dosyasının açılacağını belirtir.
-t (tabel of contents): Tar komutu ile oluşturulmuş arşiv dosyasının içeriğini görüntülemek için kullanılır.
-v (verbose): Tar komutu ile dosya oluşturulurken veya açılırken arşiv içindeki dosyaların isimlerini listeler.
-z: Arşivdeki dosyaların gzip ve gunzip komutları ile kullanılacağını belirtir.

nano Komutu: Yeni bir metin dosyası oluşturmak, metin dosyasını düzenlemek veya görüntülemek için kullanılır. Bu komutun kullanımı nano [dosya_adı] şeklindedir.

grep Komutu: Bir dosyadaki metinde arama yapmak için kullanılır. Bu komutun kullanımı grep [aranacak_metin] [dosya_adı] şeklindedir.

su (switch user) Komutu: İşletim sisteminde tanımlı olan kullanıcılar arasında değişiklik yapmak için kullanılır. Bu komutun kullanımı su [kullanıcı_adı] şeklindedir.

sudo (super user do) Komutu: Komut yöneticisi ya da kök izinleri gerektiren görevlerin yapılmasını sağlar. Bu komut dikkatli kullanılmalıdır. Komutun yanlış kullanılması, sistemsel hataların oluşmasına yol açabilir.

apt (advanced packet tool) Komutu: Programları kurmak için kullanılır.

- **sudo apt install paket_adı:** Paket adı girilen paketin kurulmasını sağlar.
- **sudo apt update:** Paket veri tabanını güncellemek için kullanılır.
- **sudo apt upgrade:** Yüklü olan tüm paketleri güncellemek için kullanılır.
- **sudo apt remove paket_adı:** Paket adı girilen paketin silinmesini sağlar.

raspi-config Komutu: Raspberry Pi konfigürasyon menüsünü açar.

ifconfig Komutu: Kullanılan kablosuz ağın durumunu gösterir.

iwconfig Komutu: Hangi ağa bağlanıldığını gösterir.

hostname Komutu: Raspberry Pi'nin ağda kullandığı ismi gösterir. **- I** parametresi kullanılarak Rasberry Pi cihazın IP adresi öğrenilir.

wget Komutu: Dosya ve dizinlerin indirilmesini sağlar.

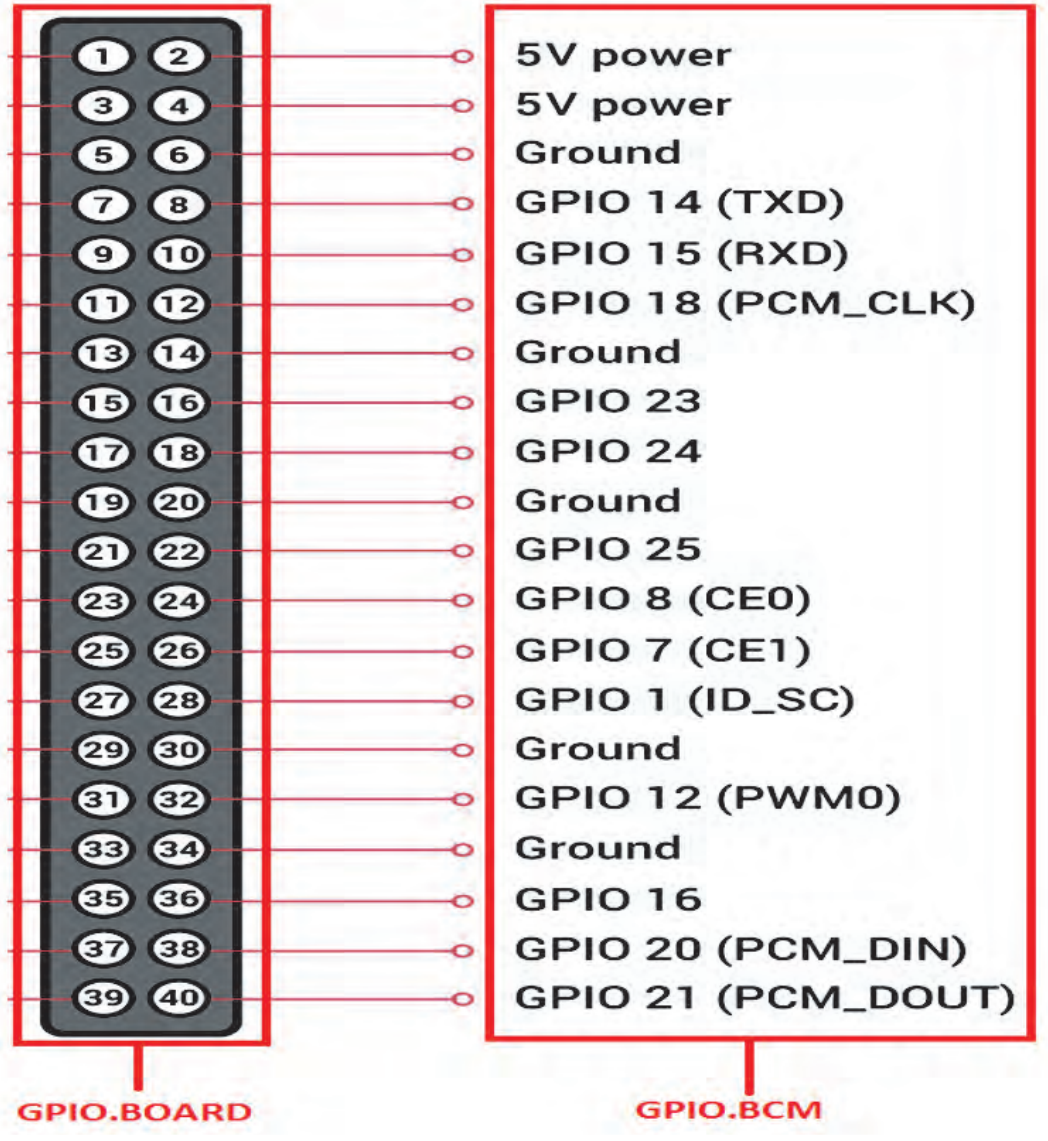
3.6.4. Raspberry Pi Donanımında Python Diliyle Uygulamalar

Raspberry Pi'de GPIO pinleri diğer elektronik kartlardaki gibi izole bir yapıya sahip olmadığı için kısa devre olma ihtimali yüksektir. Bu nedenle dikkat edilmelidir.

GPIO.BOARD: Bu kod, kart üzerindeki pinlerin numaralarına göre kodlanmasını sağlar (Görsel 3.51).

GPIO.BCM: Bu kod, kart üzerindeki GPIO pinlerinin numaralarına göre kodlanmasını sağlar (Görsel 3.51).

Sensörlerin kullanılabilmesi için kütüphanelerin Raspberry Pi cihazında yüklü olması gerektiği unutulmamalıdır.



Görsel 3.51: BOARD ve BCM

9. UYGULAMA

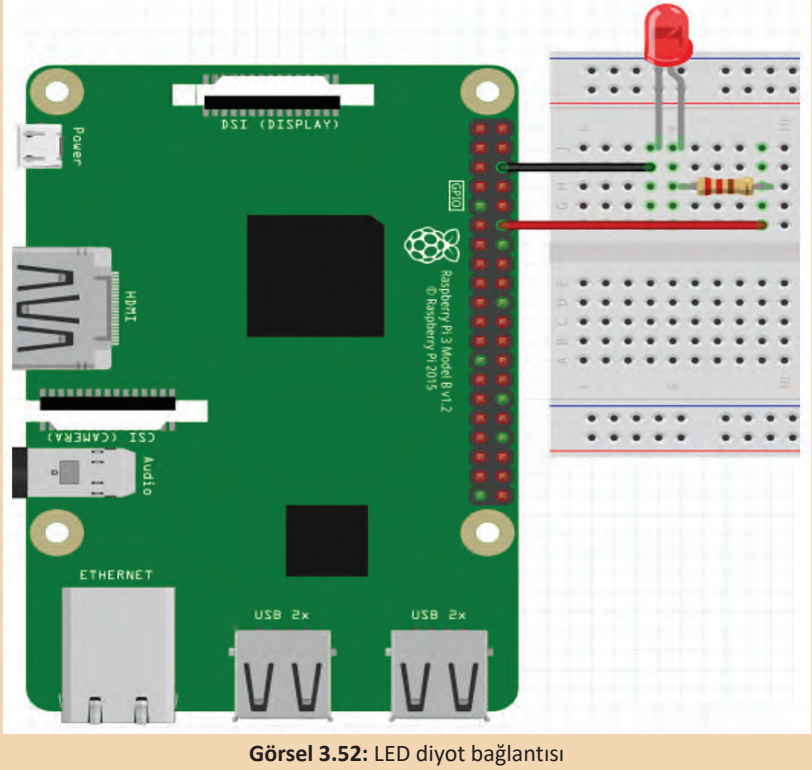
Raspberry Pi kullanarak LED'i 5 defa yanıp söndürecek uygulamayı yapınız.

Gerekli Malzemeler

- Raspberry Pi
- Breadboard
- LED
- 220 Ω veya 330 Ω direnç
- 2 adet erkek-dişi atlama kablosu (jumper)

Görsel 3.52'deki devre kurulduktan sonra aşağıdaki kod Raspberry Pi'de bulunan Mu editör ekranında yazılır ve devre çalıştırılır.

```
import time
 import RPi.GPIO as GPIO

 GPIO.setmode(GPIO.BOARD)
 GPIO.setup(12,GPIO.OUT)
     for i in range(0,5):
         GPIO.output(12,GPIO.HIGH)
         time.sleep(1)
         GPIO.output(12,GPIO.LOW)
         time.sleep(1)
 GPIO.cleanup()
```



Görsel 3.52: LED diyot bağlantısı

10. UYGULAMA

Raspberry Pi HC-SR04 sensörü ile mesafe ölçümü uygulamasını yapınız.

Gerekli Malzemeler

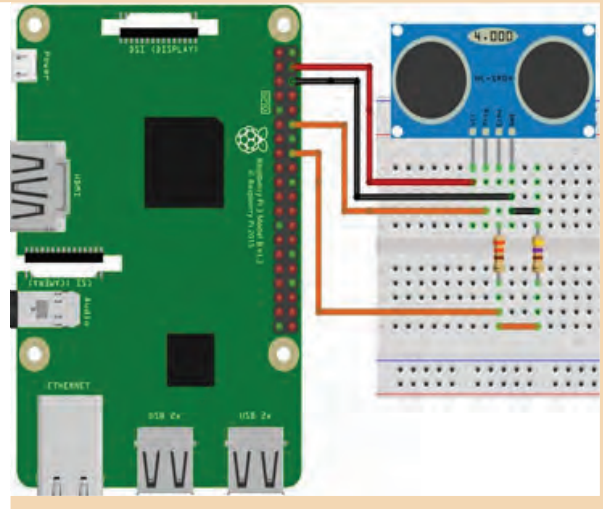
- Raspberry Pi
- Breadboard
- HC-SR04 sensörü
- 2 adet 1 kΩ direnç
- Erkek-dişi atlama kablosu

Görsel 3.53'teki devre kurulduktan sonra aşağıdaki kod Raspberry Pi'de bulunan Mu editör ekranında yazılır ve devre çalıştırılır.

NOT

HC-SR04 sensörü 3-400 cm aralığında ölçüm yapabilmektedir.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)  veya (GPIO.BCM)
GPIO.setwarnings(False)

TRIG = 12    (BCM için TRIG 18)
ECHO = 16    (BCM için ECHO 23)

print ("HC-SR04 ile Mesafe Ölçümü")

GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)

while True:

    GPIO.output(TRIG, True)
    time.sleep(0.00001)
    GPIO.output(TRIG, False)

    while GPIO.input(ECHO)==0:
        pulse_start = time.time()

    while GPIO.input(ECHO)==1:
        pulse_end = time.time()

    pulse_duration = pulse_end - pulse_start

    distance=round(pulse_duration * 17150,2)
    print ("Mesafe:%.2fcm"%(distance))
```



Görsel 3.53: HC-SR04 Ultrasonik mesafe sensörü bağlantısı

11. UYGULAMA

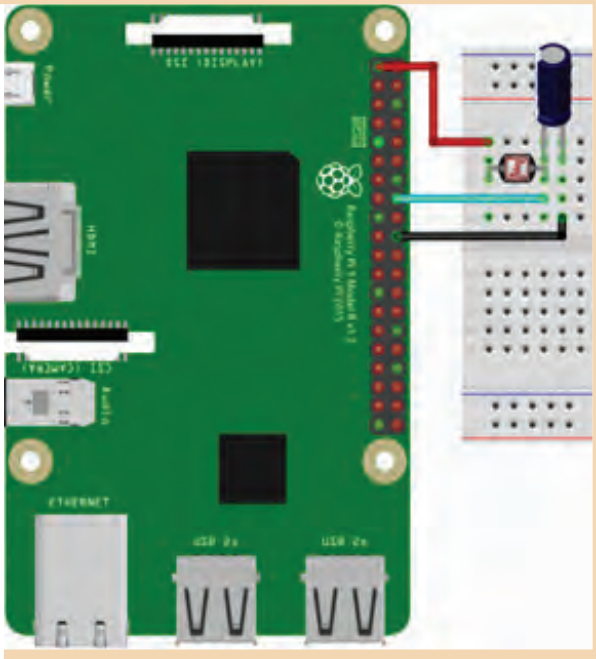
Raspberry Pi LDR ile ışık şiddeti ölçme uygulamasını yapınız.

Gerekli Malzemeler

- Raspberry Pi
- Breadboard
- LDR
- 1 µF kondansatör
- Erkek-dişi atlama kablosu

Görsel 3.54'teki devre kurulduktan sonra aşağıdaki kod Raspberry Pi'de bulunan Mu editör ekranında yazılır ve devre çalıştırılır.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

ldr = 16

def ldr_oku (ldr):
    oku = 0
    GPIO.setup(ldr, GPIO.OUT)
    GPIO.output(ldr, False)
    #GPIO.output(ldr, GPIO.LOW)

    time.sleep(1)

   GPIO.setup(ldr, GPIO.IN)
    while (GPIO.input(ldr) == 0):
        oku += 1
     return oku

    while True:
        print("LDR Değeri:")
        print(ldr_oku(ldr)
        time.sleep(1)
    GPIO.cleanup()
```



Görsel 3.54: LDR bağlantısı

12. UYGULAMA

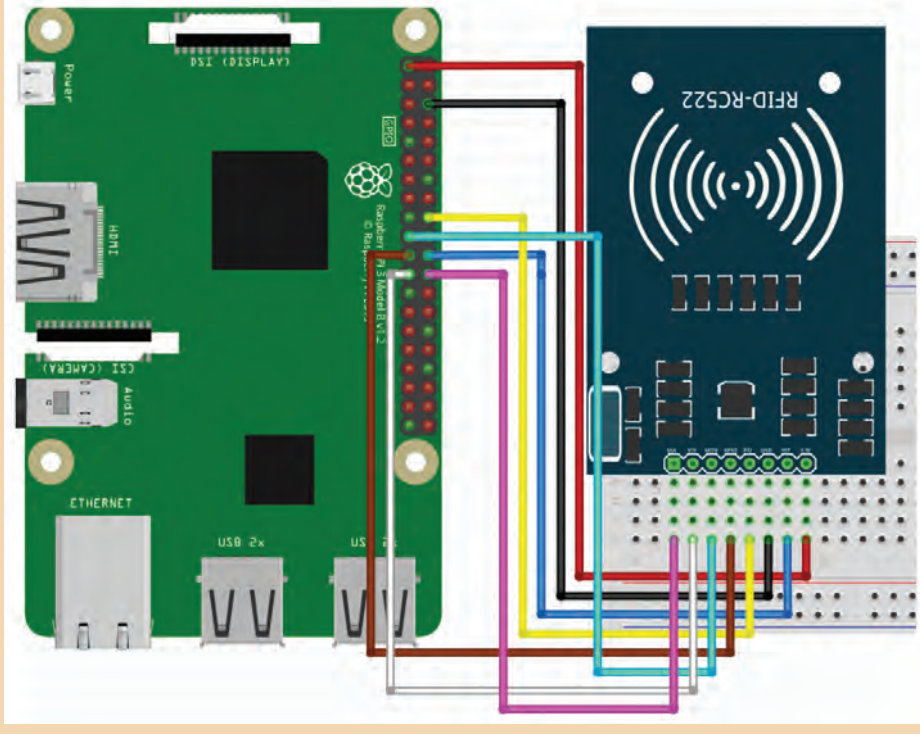
Raspberry Pi RFID kart numarası okuma uygulamasını yapınız.

Gerekli Malzemeler

- Raspberry Pi
- Breadboard
- RC522 RFID seti
- Erkek-dişi atlama kablosu

Görsel 3.55'teki devre kurulduktan sonra RFID kartını çalıştırabilmek için gerekli olan kütüphane yüklenmelidir. Kütüphane, terminal ekranında sudo pip install pi-rc522 kodu yazılarak yüklenir. Aşağıdaki kod Raspberry Pi'de bulunan Mu editör ekranında yazılır ve devre çalıştırılır.

```
from pirc522 import RFID
import signal
import time

rdr = RFID()
util = rdr.util()
util.debug = True
print("Kartı okutunuz...")
rdr.wait_for_tag()
(error, data) = rdr.request()

if not error:
    print("Kart okunuyor!")
    (error, uid) = rdr.anticoll()
    if not error:
    kart_uid = str(uid[0])+" "+str(uid[1])+" "+str(uid[2])+" "+str(uid[3])+" "+str(uid[4])
    print(kart_uid)
```



Görsel 3.55: RC522 RFID bağlantısı

13. UYGULAMA

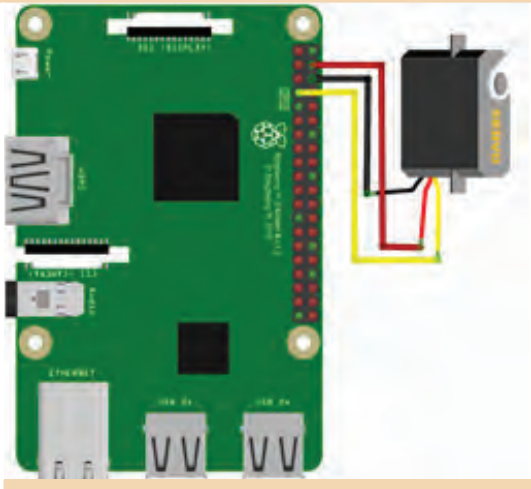
Raspberry Pi servo motor kullanımı uygulamasını yapınız.

Gerekli Malzemeler

- Raspberry Pi
- Breadboard
- Micro servo motor 5 V
- Erkek-dişi atlama kablosu

Görsel 3.56'daki bağlantılar yapıldıktan sonra program kodları Raspberry Pi'de bulunan Mu editör ekranında yazılır ve devre çalıştırılır.

```
import RPi.GPIO as GPIO
import time

servoPIN = 7
GPIO.setmode(GPIO.BOARD)
GPIO.setup(servoPIN, GPIO.OUT)

p = GPIO.PWM(servoPIN, 50)
# GPIO 7 için PWM 50Hz

p.start(2.5)
p.ChangeDutyCycle(5)
time.sleep(0.5)
p.ChangeDutyCycle(7.5)
time.sleep(0.5)
p.ChangeDutyCycle(10)
time.sleep(0.5)
p.ChangeDutyCycle(12.5)
time.sleep(0.5)
p.ChangeDutyCycle(10)
time.sleep(0.5)
p.ChangeDutyCycle(7.5)
time.sleep(0.5)
p.ChangeDutyCycle(5)
time.sleep(0.5)
p.ChangeDutyCycle(2.5)
time.sleep(0.5)
p.stop()
GPIO.cleanup()
```



Görsel 3.56: Servo motor bağlantısı

14. UYGULAMA

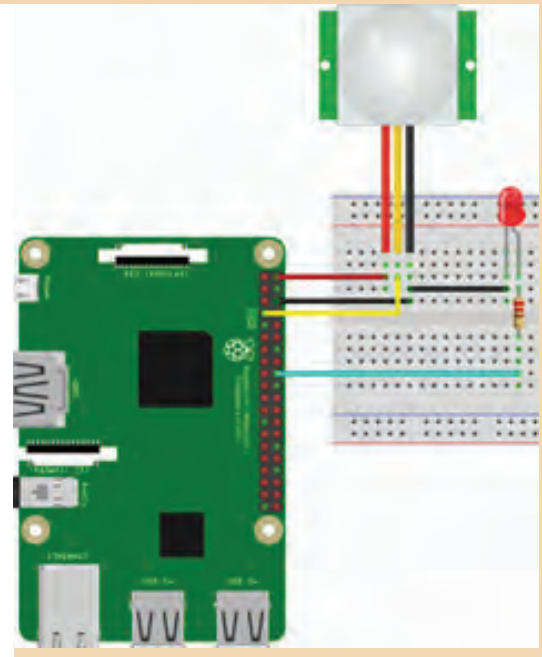
Raspberry Pi PIR sensörü ile hareket algılama uygulamasını yapınız.

Gerekli Malzemeler

- Raspberry Pi
- Breadboard
- Erkek-dişi atlama kablosu
- PIR sensör
- LED

Görsel 3.57'deki bağlantılar yapıldıktan sonra program kodları Raspberry Pi'de bulunan Mu editör ekranında yazılır ve devre çalıştırılır.

```
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)

pir_pin = 7
led_pin = 18

GPIO.setup(pir_pin, GPIO.IN)
GPIO.setup(led_pin, GPIO.OUT)
GPIO.output(led_pin, True)
while True:
    if GPIO.input(pir_pin):
        print("LED yandı.")
        GPIO.output(led_pin, 1)
        time.sleep(10);
        print("LED söndü.")
        GPIO.output(led_pin, 0)
        time.sleep(3)
    time.sleep(1)
```



Görsel 3.57: PIR hareket sensörü ve LED bağlantısı

15. UYGULAMA

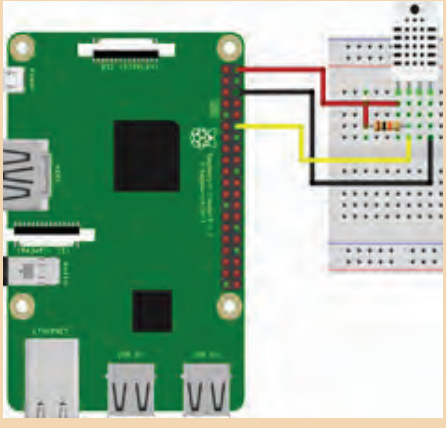
Raspberry Pi DHT22 kullanımı ile sıcaklık ve nem algılama uygulamasını yapınız.

Gerekli Malzemeler

- Raspberry Pi
- Breadboard
- Erkek-dişi atlama kablosu
- DHT22
- 10 kΩ direnç

Görsel 3.58'deki devre kurulduktan sonra DHT22 sensörünü çalıştırabilmek için gerekli olan kütüphane yüklenmelidir. Kütüphane yüklendikten sonra aşağıdaki kod Raspberry Pi'de bulunan Mu editör ekranında yazılır ve devre çalıştırılır.

```
import RPi.GPIO as GPIO
import dht22
import time
import datetime

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

deger = dht22.DHT22(pin=14)
while True:
    sicaklik = deger.read()
    if sicaklik.is_valid():
        print("Sıcaklık %-2.1f C" % sicaklik.temperature)
        print("Nem: %-2.1f %%" % sicaklik.humidity)
    time.sleep(6)
```

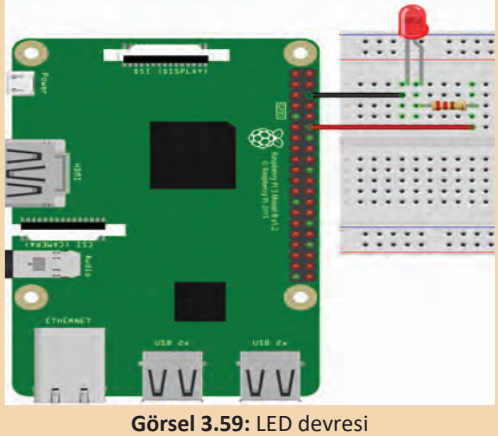


Görsel 3.58: Servo motor bağlantısı

16. UYGULAMA

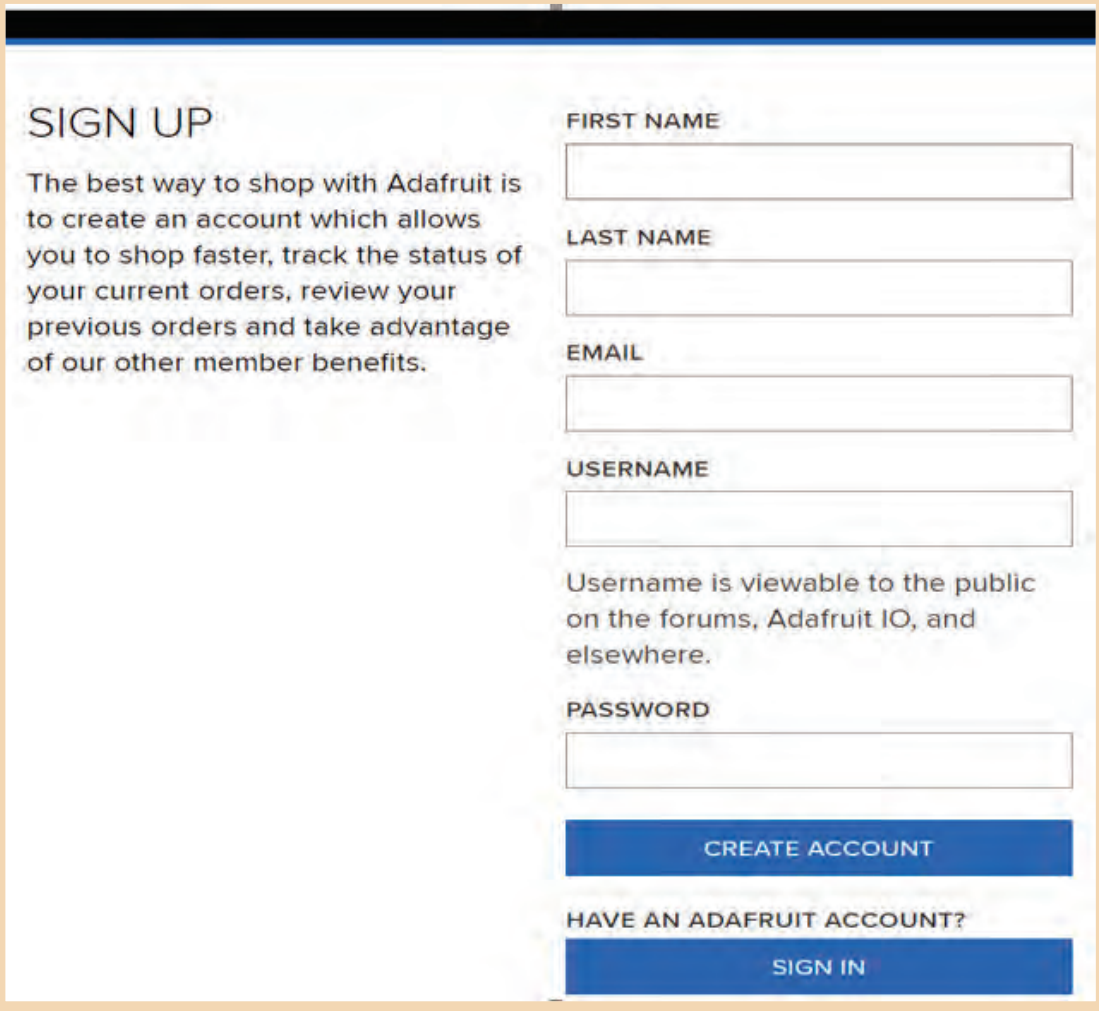
Nesnelerin İnterneti ile LED yakma uygulamasında farklı lokasyonda olan cihazların MQTT (makineler arası mesaj taşıma) yapısını kullanarak mesajlaşmasını, mesaj içeriğine göre LED'in yanıp sönmesini sağlayınız.

1. Adım : Raspberry Pi 3B kartının üzerinde yandaki devreyi kurunuz (Görsel 3.59).



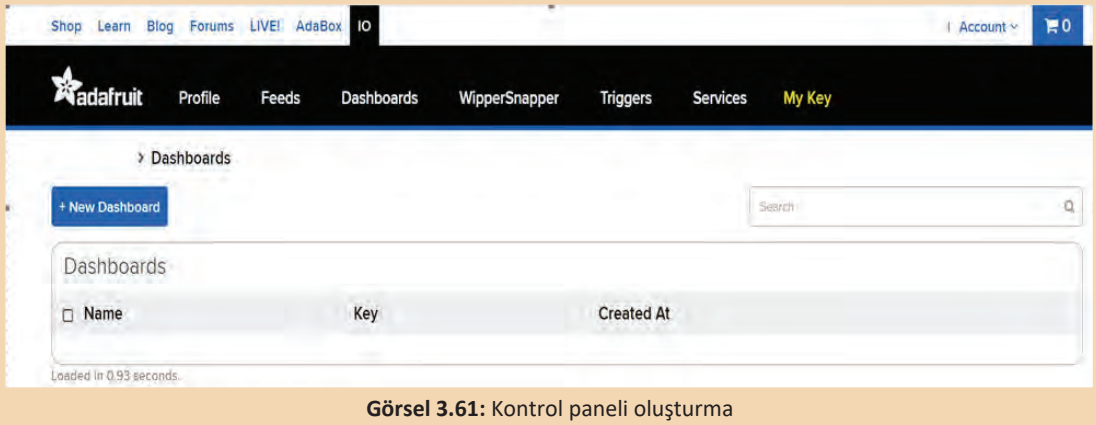
Görsel 3.59: LED devresi

2. Adım : Makineler arasında iletişimin gerçekleşmesini sağlayacak olan MQTT desteğinin sağlanması için **https://www.adafruit.com** adresinden ücretsiz üyelik oluşturup giriş yapınız (Görsel 3.60).



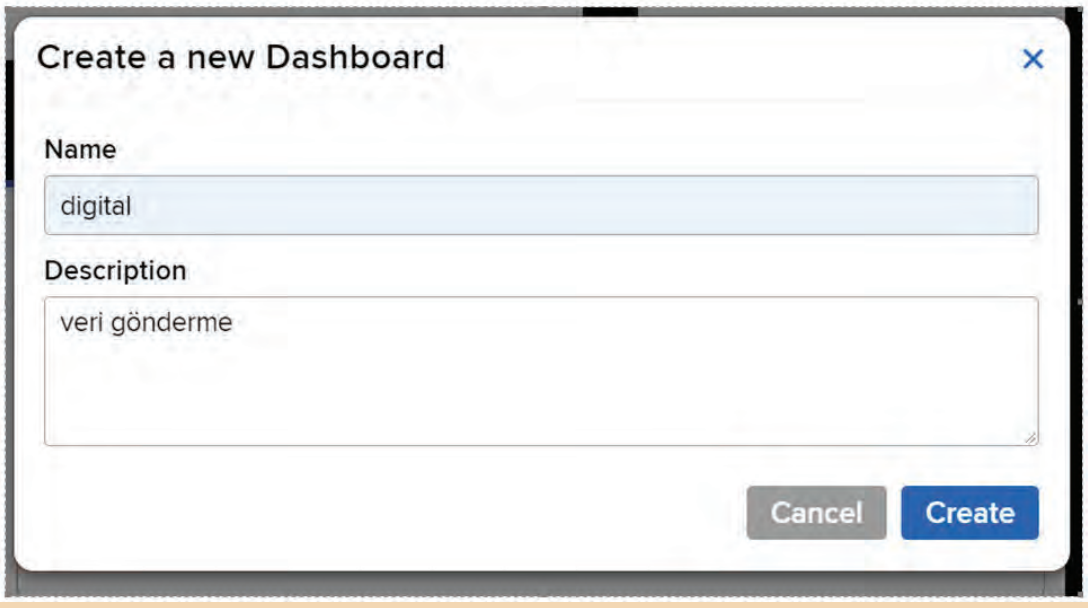
Görsel 3.60: Adafurit yeni kullanıcı oluşturma

3. Adım : Daha sonra IO sekmesine basıp New Dashboard butonuna tıklayınız (Görsel 3.61).



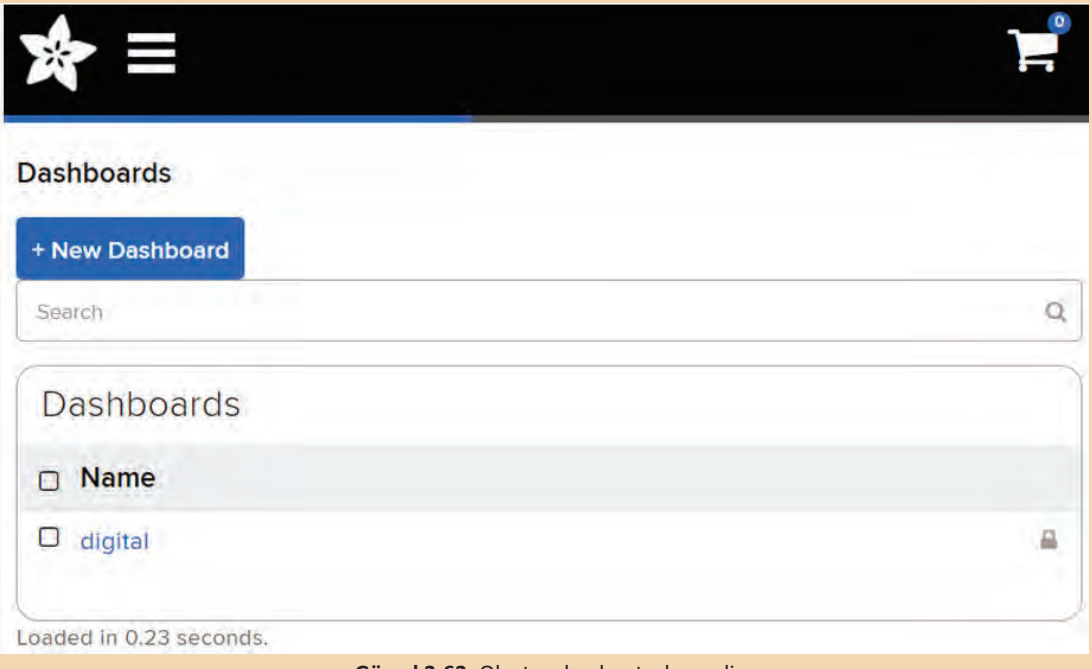
Görsel 3.61: Kontrol paneli oluşturma

4. Adım : Yeni tanımlanacak olan kontrol panelinin ismi ile açıklamasını giriniz ve Create butonuna tıklayınız (Görsel 3.62).



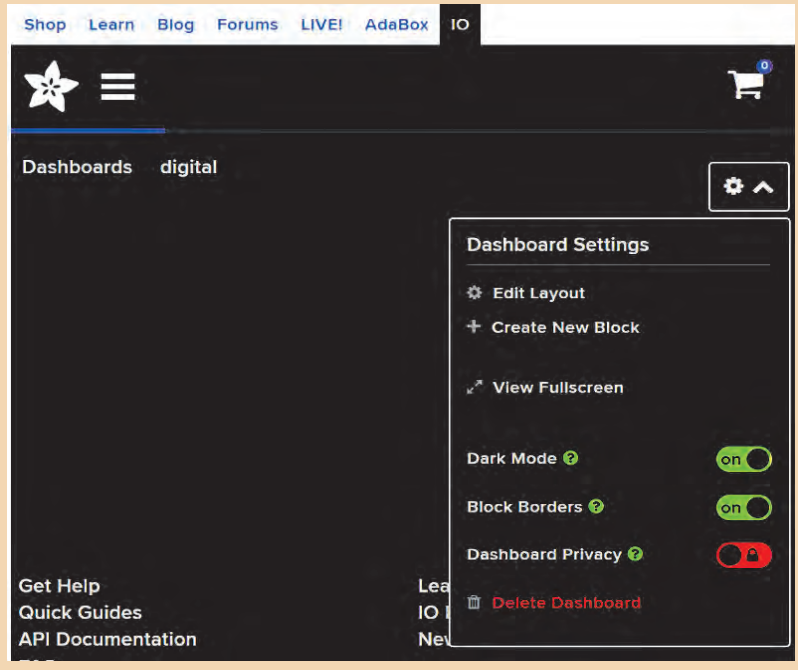
Görsel 3.62: Kontrol paneli isimlendirme

5. Adım : Yeni oluşturulan digital isimli kontrol paneli sekmesine (Görsel 3.63) tıklayın.



Görsel 3.63: Oluşturulan kontrol paneli

Açılan yeni sayfadaki ögesini tıklayarak gerekli ayarları yapınız (Görsel 3.64).



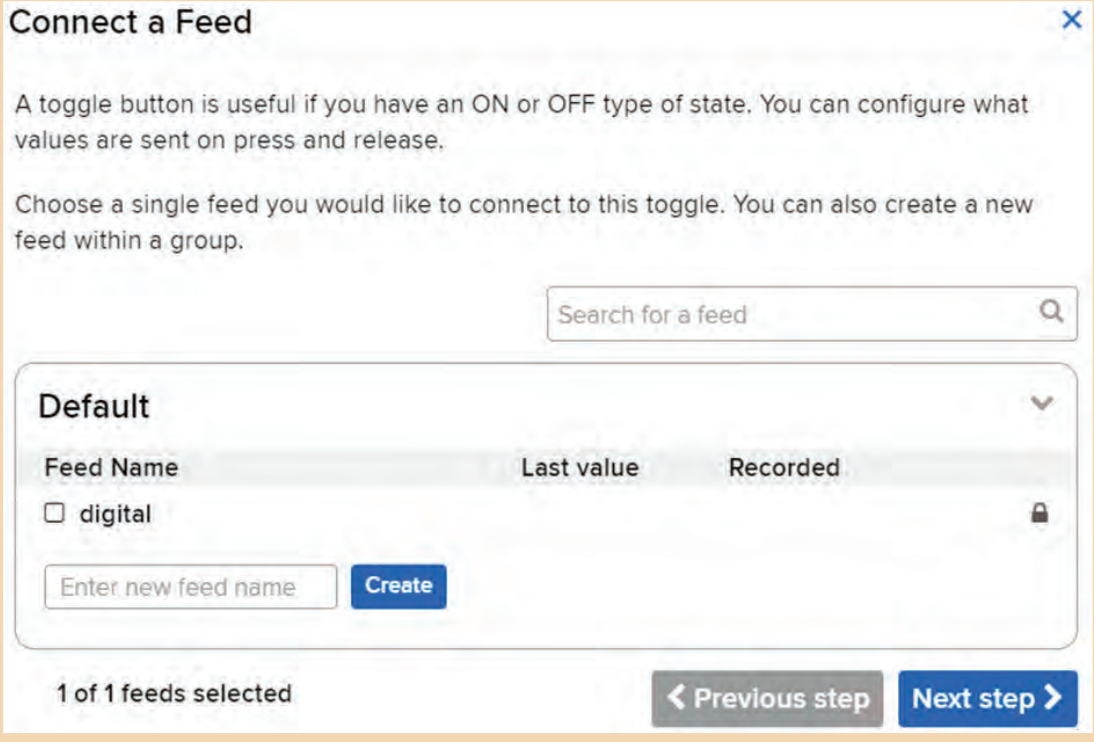
Görsel 3.64: Kontrol paneli ayar sayfası

6. Adım : Açılan yeni pencerede Create New Block seçeneğini tıklayarak Adafruit tarafından kullanıcılar için oluşturulmuş hazır bloklardan uygun olanı seçiniz (Görsel 3.65).

Görsel 3.65: Kontrol blokları

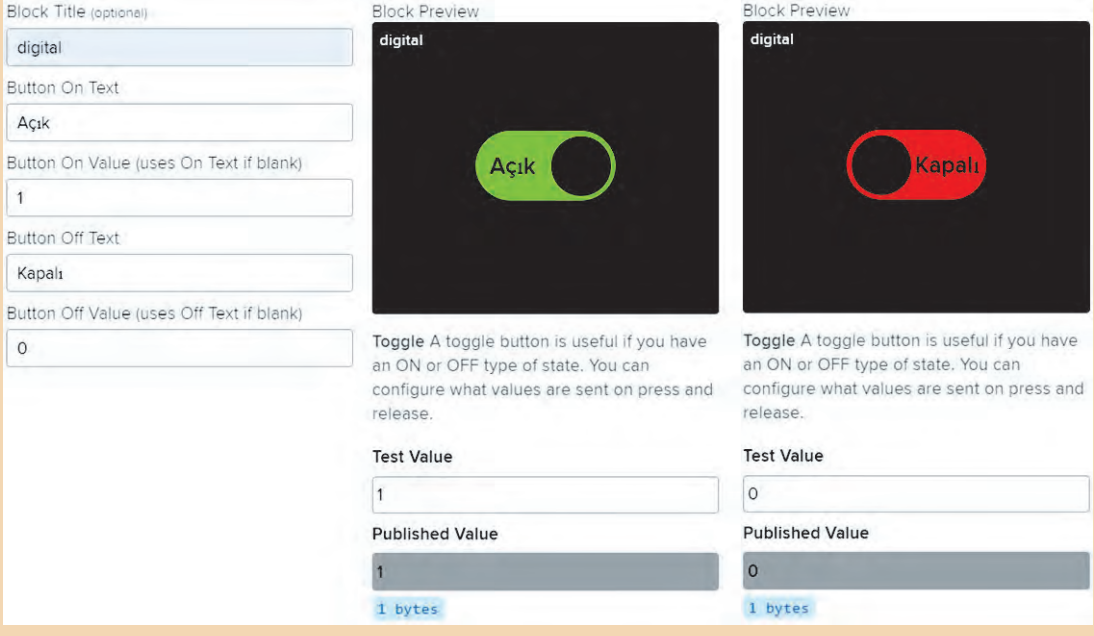
7. Adım : Yapılan uygulamada IoT ile LED yakma işlemi gerçekleştirileceği için

blokunu seçiniz. Açılan pencerede seçilen blok ile digital isimli bağlantı oluşturunuz (Görsel 3.66).



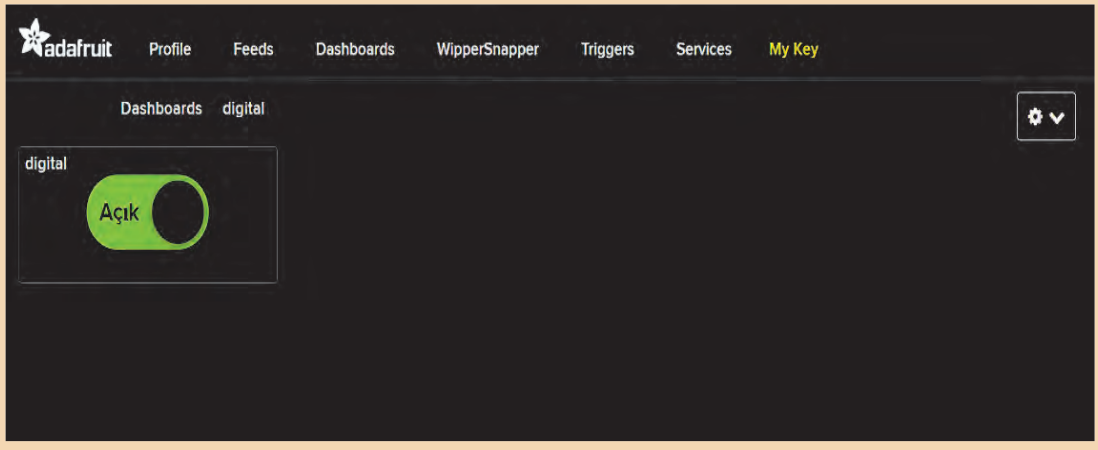
Görsel 3.66: Bağlantı oluşturma

8. Adım : Next Step butonu ile oluşturulan blok ayarlarının yapıldığı pencere ile blok başlığını, butonun açık ve kapalı hâlindeki ismini ve buton dönüş değerleri değişikliğini görseldeki gibi yapınız (Görsel 3.67).



Görsel 3.67: Blok oluşturma ve deneme

9. Adım : Create block butonuna basarak blok panoya ekleyiniz.



Görsel 3.68: Seçilmiş blok görünümü

Oluşturulan bu blokun Raspberry Pi de kullanılabilmesi için API keylerinin belirtilmesi gerekir. Görsel 3.68'de görüldüğü gibi bu API keylere **My Key** sekmesinden ulaşılabilir. Burada API keye ulaşım için gerekli olan kullanıcı adı ve key bilgisi görülmektedir (Görsel 3.69).

Görsel 3.69: API Key

10. Adım : Raspberry Pi üzerinde Terminal ekranına geçiş yaparak aşağıdaki komutları sırası ile çalıştırınız.

sudo apt-get update

sudo apt-get upgrade

pip3 install RPI.GPIO

pip3 install adafruit-io

Raspberry Pi üzerinde bir text dosyası açılarak diğer sayfadaki kodlar yazılıp **led.py** uzantılı olarak kaydedilir.

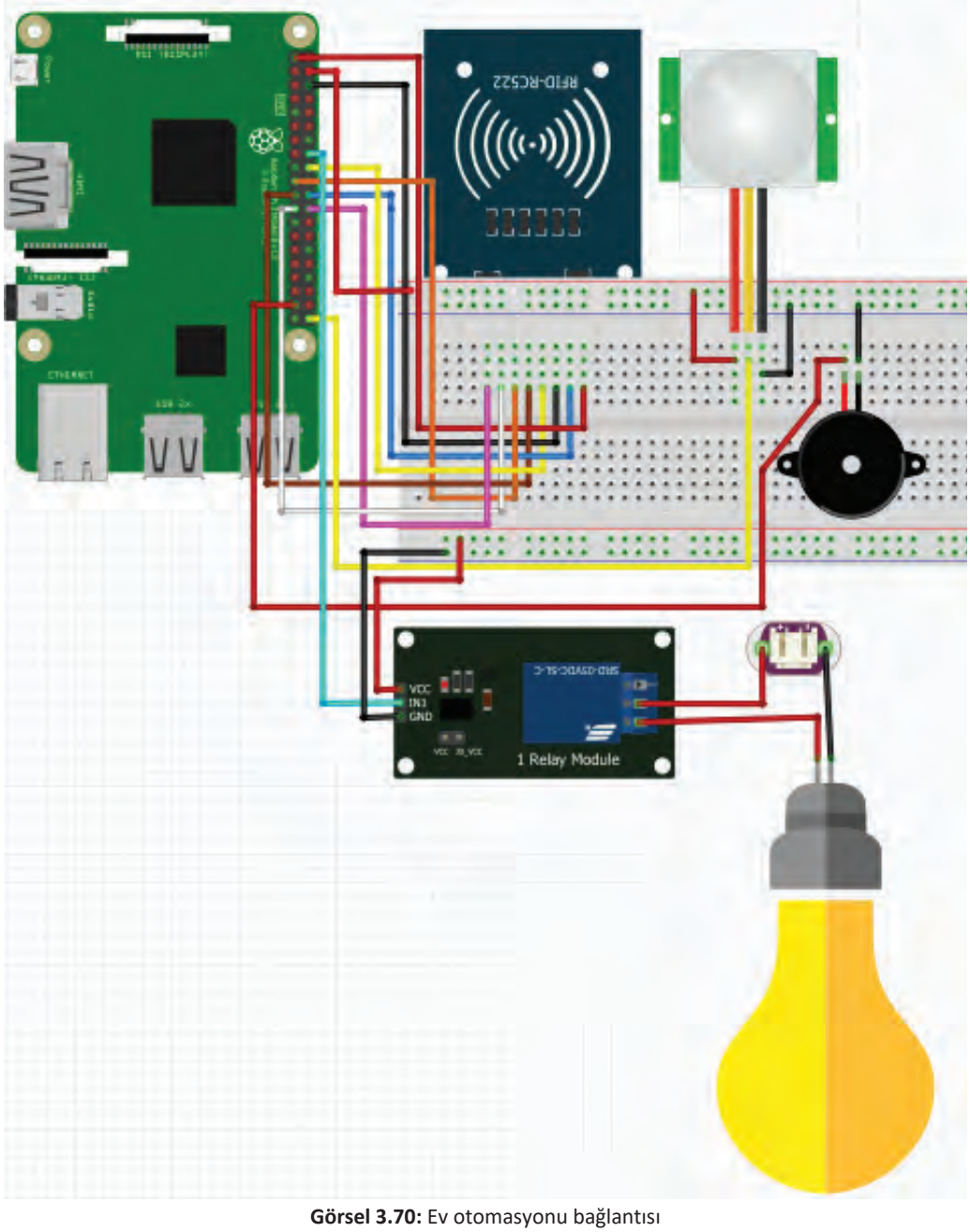
```
import RPi.GPIO as GPIO
import time

# Adafruit Rest Client yüklemesi.
from Adafruit_IO import Client, Feed, RequestError
GPIO.setmode(GPIO.board)

#11 nolu pin çıkış pini olarak ayarlandı.
GPIO.setup(11,GPIO.OUTPUT)

# Oluşturduğunuz Adafruit IO key.
ADAFRUIT_IO_KEY = 'AIO_Key'

# Adafruit IO'daki kullanıcı adınız.
ADAFRUIT_IO_USERNAME = 'AIO_Kullanıcı_Adı'

# REST Clinet bağlantı ayarları tanımı.
aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)

try: # Eğer 'digital' isimli yayıncımız var ise
digital = aio.feeds('digital')
except RequestError: # yoksa digital isimli yayıncı oluştur
feed = Feed(name="digital")
        digital = aio.create_feed(feed)

while True:
        data = aio.receive(digital.key)
        if int(data.value) == 1:
                print('Led YANDI\n')
        elif int(data.value) == 0:
                print('Led SÖNDÜ \n')
    # Led durumunu
        GPIO.OUTPUT(11,int(data.value))
    # Adafruit'e gönderilen istekler için bekleme süresi
        time.sleep(0.5)
```

Kayıt işlemi tamamlandıktan sonra Terminal ekranına gelinip **python3 led.py** komutu ile devre çalıştırılarak Adafruit üzerinde oluşturulan LED yakıp söndürülür.

3.6.5. Raspberry Pi Donanımıyla Ev Otomasyonu Projesi

Raspberry Pi kartı ile basit bir ev otomasyonu projesi Görsel 3.70'teki gibi oluşturulabilir. Ev otomasyonu projesinde kullanılacak malzemeler şunlardır:

- 1 adet PIR (Hareket) sensörü
- 1 adet RC522 RFID kart modülü
- 1 adet buzzer
- 1 adet röle
- 1 adet lamba



Görsel 3.70: Ev otomasyonu bağlantısı

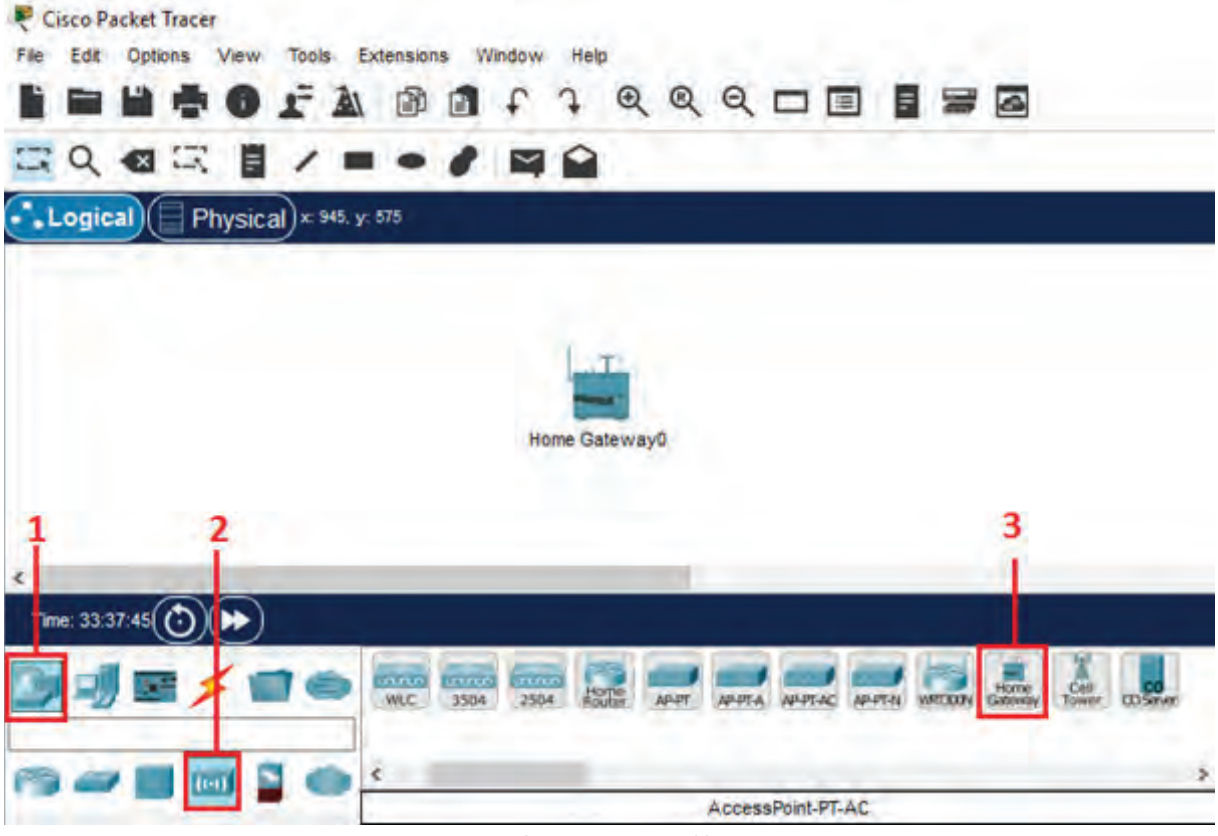
Basit ev otomasyonunda aşağıdaki uygulamalar gerçekleştirilecektir.

- Hareket algılandıktan sonra 20 saniye lamba yanacaktır.
- Okutulan RFID kart sisteme tanımlı olan RFID kart değilse buzzer aktif olacaktır.

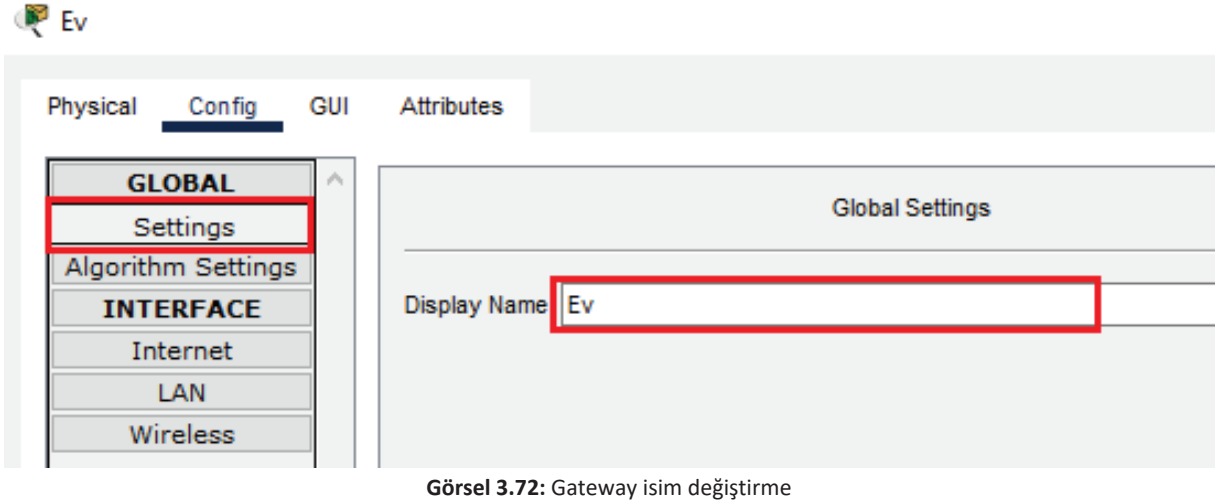
```
from pirc522 import RFID
import signal
import time
import RPi.GPIO as GPIO

pir_pin = 40
role_pin =8
buzzer_pin=37

GPIO.setmode(GPIO.BOARD)
GPIO.setup(pir_pin, GPIO.IN)
GPIO.setup(role_pin, GPIO.OUT)
GPIO.setup(buzzer_pin, GPIO.OUT)
GPIO.output(role_pin, True)

def kart_oku:
    rdr = RFID()
    util = rdr.util()
    util.debug = True

    while True:
        rdr.wait_for_tag()
        (error, data) = rdr.request()
        if not error:
        print("\nKartı okutunuz!")
        (error, uid) = rdr.anticoll()

        if not error:
        kart_uid = str(uid[0])+" "+str(uid[1])+" "+str(uid[2])+" "+str(uid[3])+" "+str(uid[4])

        if (kart_uid == ("64 150 49 131 100") :
            GPIO.output(role_pin, False)
            print("Lamba yandı.")
            time.sleep(20)
            GPIO.output(buzzer_pin,False)
        else:
            GPIO.output(role_pin, False)
            print("Lamba yandı.")
            time.sleep(20)
            GPIO.output(buzzer_pin,True)
            print("Alarm devrede!")
            time.sleep(50)


while True:
    if GPIO.input(pir_pin):
        kart_oku()
        time.sleep(10)
        GPIO.output(buzzer_pin,True)
```

3.7. SİMÜLASYON ARACI

Bu bölümde simülatör programı ile akıllı ev otomasyonu yapılacaktır. Akıllı ev için kullanılacak elektronik devre kartı, Görsel 3.71'deki gibi simülatör çalışma alanına eklenir ve Görsel 3.72'deki gibi yeniden isimlendirilir.



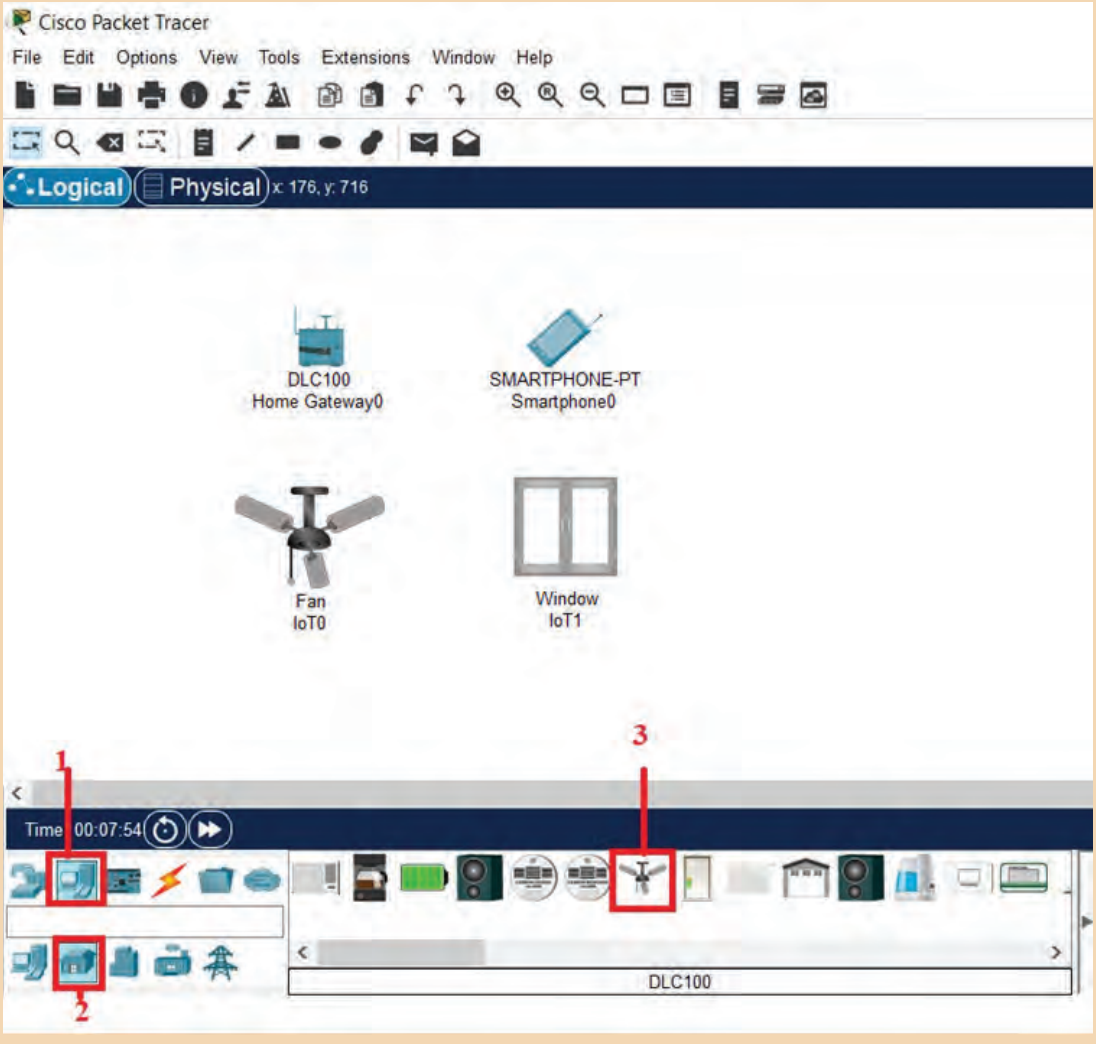
Görsel 3.71: Gateway ekleme



Görsel 3.72: Gateway isim değiştirme

17. UYGULAMA

Bir cep telefonu uygulaması ile belirlenen sıcaklık eşik değerinde fan çalıştıran ve pencerelerin kapanmasını sağlayan uygulamayı simülasyon programını kullanarak yapınız.

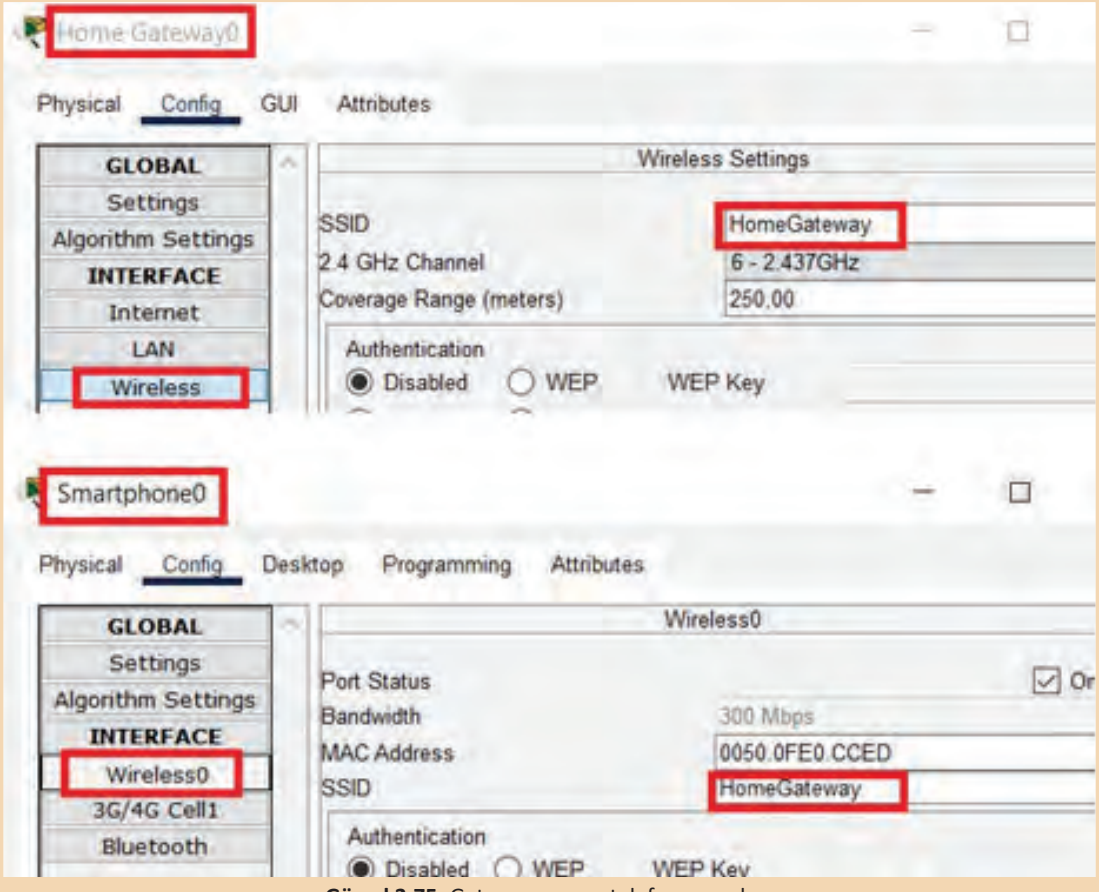


Görsel 3.73: Uygulamada kullanılacak IoT cihazlar

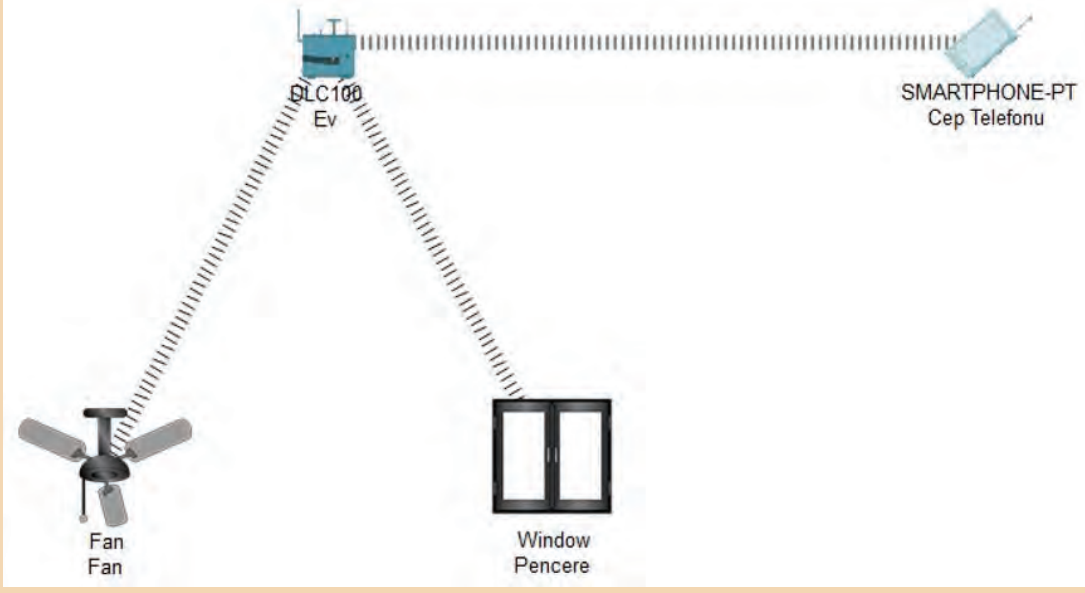
Görsel 3.74: Uygulamada kullanılacak IoT cihaz grupları

Simülatör uygulama alanına Görsel 3.73'teki sıralama adımları takip edilerek fan ve pencere, Görsel 3.74'teki sıralama adımları takip edilerek cep telefonu eklenir. Fan ve pencere Gateway'e

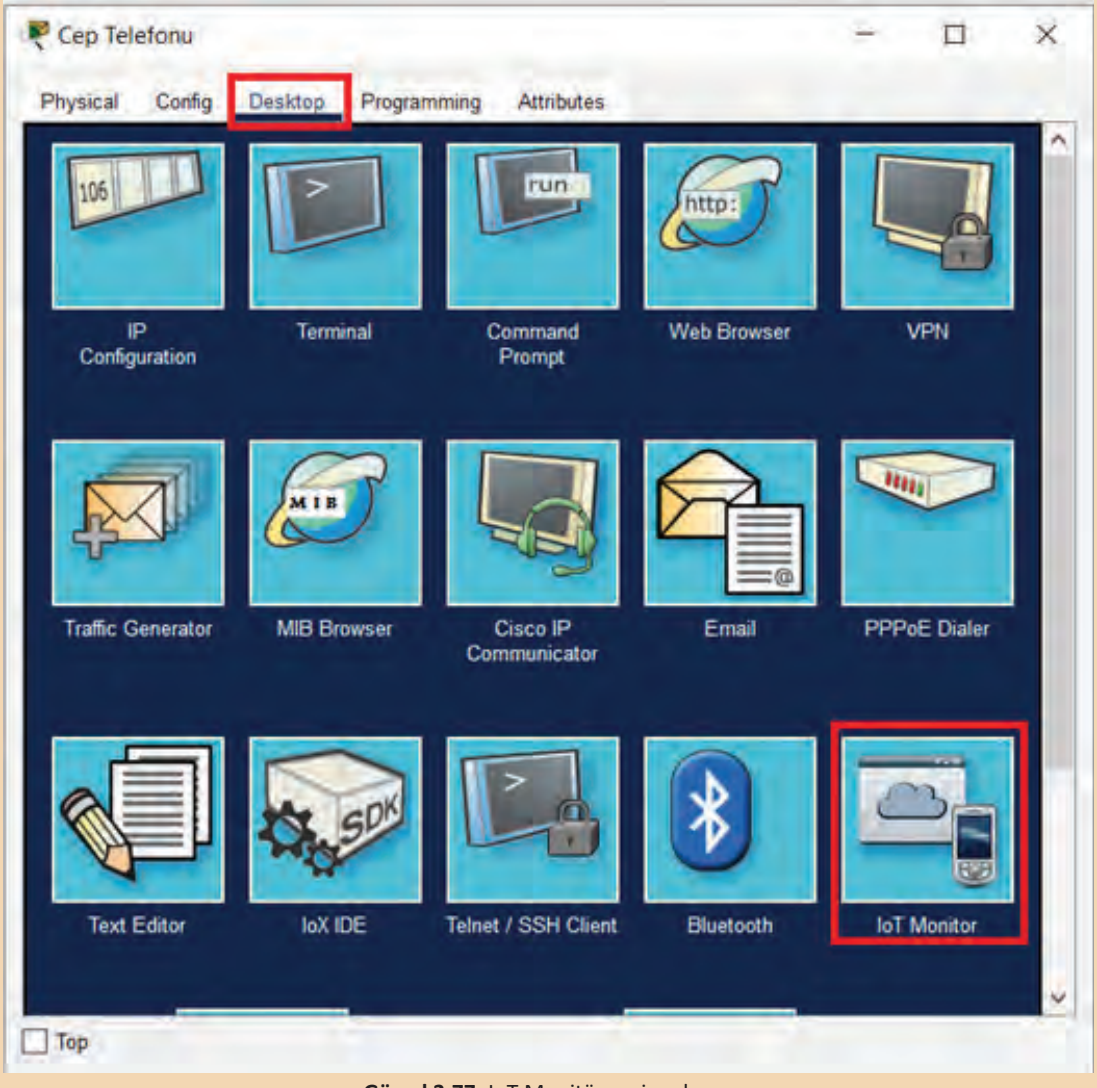
herhangi bir ayar yapmadan bağlanır ancak cep telefonunun bağlanabilmesi için Görsel 3.75'teki gibi cep telefonu SSID ile Gateway SSID aynı olmalıdır. Bu ayar da yapıldıktan sonra eklenen tüm cihazlar Gateway üzerinden iletişime hazır hâle gelir ve Görsel 3.76'daki gibi bağlantı sağlanır.



Görsel 3.75: Gateway ve cep telefonu ayarları



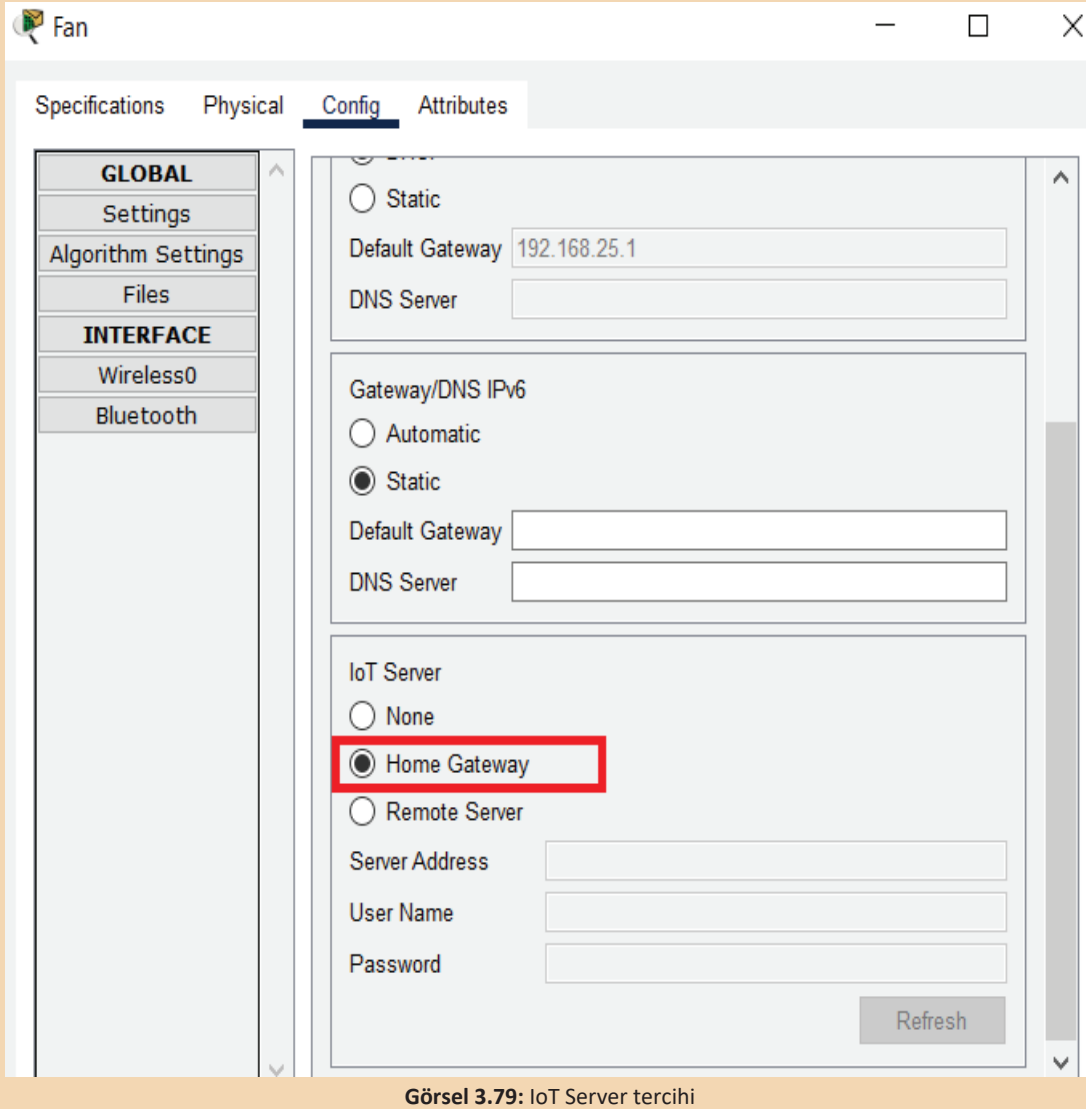
Görsel 3.76: Bağlantı kurulmuş görünüm



Görsel 3.77: IoT Monitör seçim ekranı

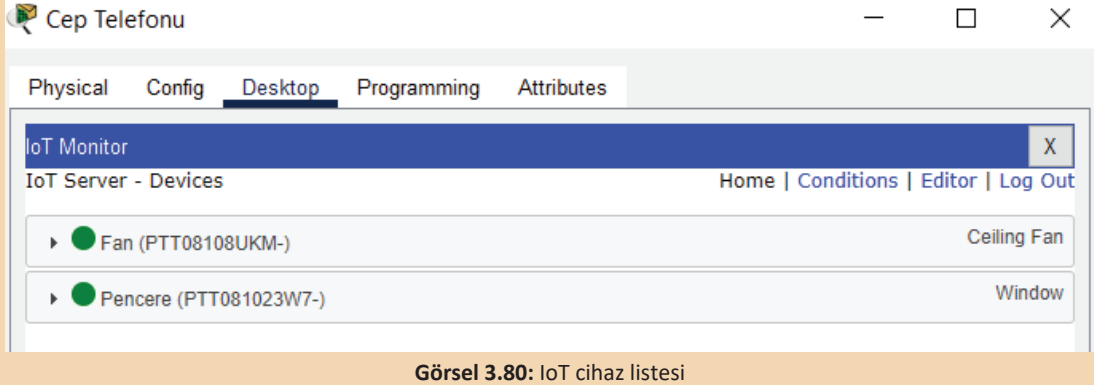
Cep telefonu nesnesine çift tıklandığında Görsel 3.77'de açılan pencereden Desktop sekmesi seçilir ve IoT Monitör tıklanır. IoT Monitör penceresinde gerekli girişler yapıldıktan sonra Görsel 3.78'deki gibi açılan pencerede herhangi bir nesne listelenmiyorsa kullanılan tüm IoT cihazların Config > IoT Server panelinde Home Gateway seçeneği seçilir (Görsel 3.79).

Görsel 3.78: IoT cihaz listesi



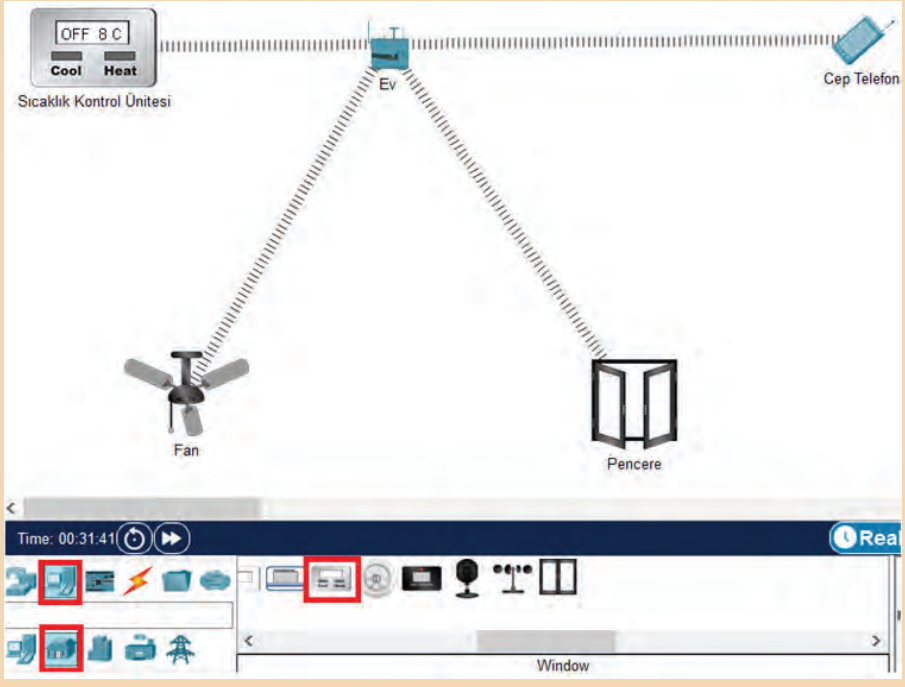
Görsel 3.79: IoT Server tercihi

Cihazların ayarları yapıldıktan sonra IoT Monitör seçeneği seçildiğinde Görsel 3.80'deki gibi IoT cihazlar listelenir.



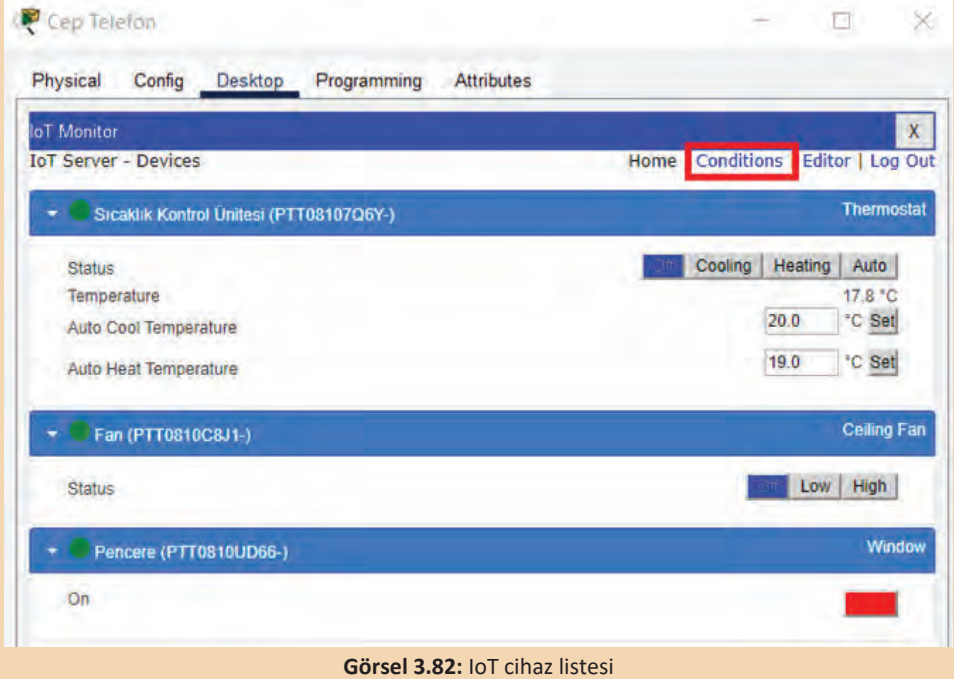
Görsel 3.80: IoT cihaz listesi

Bu cihazların seçilmesi ile simülasyon programı üzerinden açma / kapama işlemleri uzaktan cep telefonu ile gerçekleştirilir. Uygulamanın belirli sıcaklık eşiğine göre otomatik olarak çalışması için mevcut sisteme Görsel 3.81'deki termostat eklenir. IoT Server bölümünde Home Gateway ayarı yapıldıktan sonra Görsel 3.81'deki gibi termostat bağlanır.



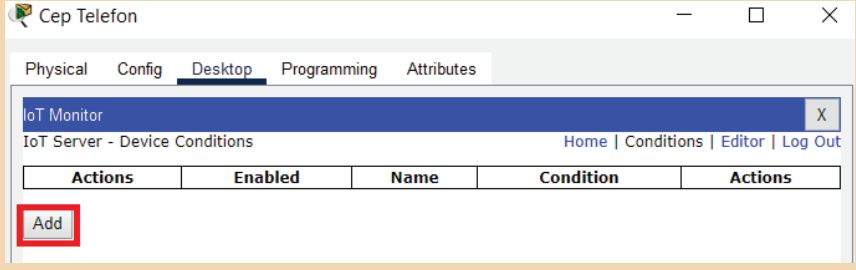
Görsel 3.81: Termostat ekleme

Cep telefonundan IoT Monitör >Desktop sekmesi seçildiğinde Görsel 3.78'de IoT cihaz listesi görülür. Şartlar belirlenip IoT cihazların bu şartlara göre tepki vermesi istenirse Görsel 3.82'deki **Conditions** sekmesi seçilir.



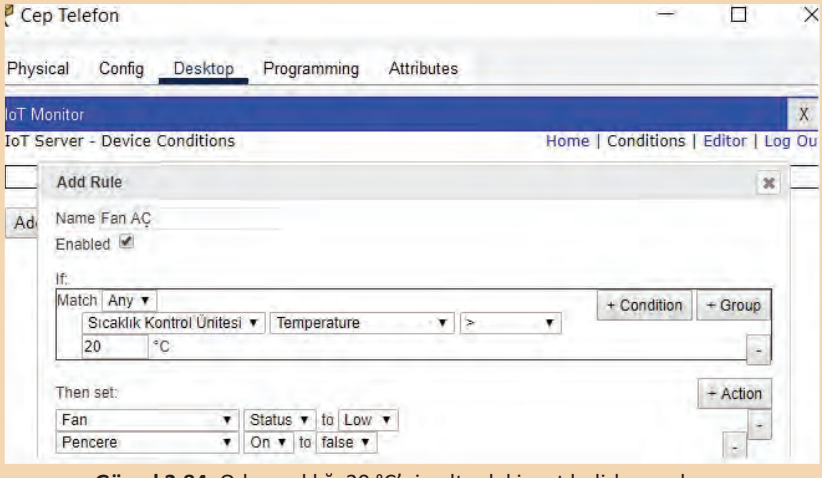
Görsel 3.82: IoT cihaz listesi

Bu uygulamada oda sıcaklığı 20 °C'nin altında ise pencere açık, fan kapalı konumda; oda sıcaklığı 20 °C'nin üstünde ise pencere kapalı, fan açık konumda olacak şartlarını eklemek için Görsel 3.83'teki **Add** butonu tıklanır.

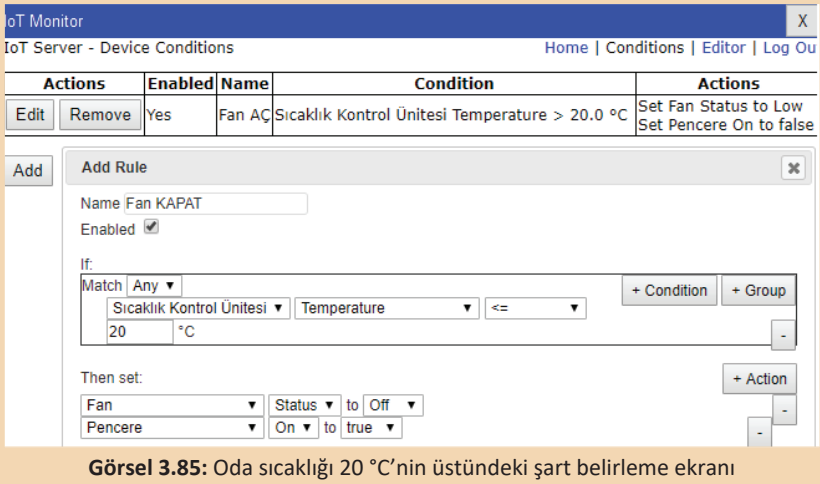


Görsel 3.83: Şart ekleme sekmesi

Belirlenen sıcaklık eşik değeri aşıldıktan sonra yapılacak işlemleri belirlemek için **Add** sekmesi seçilir. Görsel 3.84'teki gibi şart oluşturulur ve **OK** seçeneğine basılır. Görsel 3.84'te eklenen şart görülmektedir. Yeni şart için yine **Add** seçeneği tıklanır ve yeni şartlar girilir.



Görsel 3.84: Oda sıcaklığı 20 °C'nin altındaki şart belirleme ekranı



Görsel 3.85: Oda sıcaklığı 20 °C'nin üstündeki şart belirleme ekranı

Görsel 3.85'teki şartlara göre pencere ilk başta açık olacak, sıcaklık kontrol ünitesi değeri 20 °C'nin üzerine çıktığında pencere kapanıp fan düşük seviyede çalışacaktır. Böylelikle IoT cihazları bir evdeki sıcaklık kontrol işlemini gerçekleştirir.

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde parantez içine yargılar doğru ise "D", yanlış ise "Y" yazınız.

1. ( ) Get komutu yeni bir veri oluşturmak için kullanılır.
2. ( ) API'lerden dönen çeşitli HTTP durum kodları ile yapılan işlemlerin başarılı olup olmadığı tespit edilir.
3. ( ) Kullanılabilir veri bilgiye dönüşmez.
4. ( ) REST API, Server-Client iletişimi yapabilecek mimari değildir.
5. ( ) 101 Anahtarlama protokolü bilgilendirme amaçlıdır.

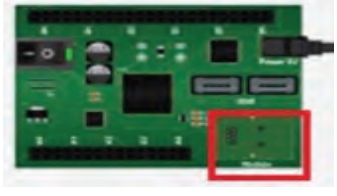
B) Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

6. Veri işleme döngüsü girdi, işleme ve ..... aşamalarından oluşur.
7. Veri işleme sırasında sınıflandırma, sıralama ve ..... teknikleri kullanılır.
8. Türleri farklı olan uygulamaların aynı yapı içinde iletişime geçerek çalışmasını sağlayan yazılımlara ..... denir.
9. REST mimarisinde HTTP metotları olan GET, POST, PUT ve ..... desteklenmektedir.
10. Kullanılabilir hâle gelen verilerin kullanıcıların anlayıp yorumlayacağı uygulamalara ..... denir.
11. Raspberry Pi; küçük boyutlu, düşük maliyetli, tek ..... bilgisayar olarak tasarlanmıştır.
12. Raspberry Pi'de ..... komutu ile GPIO pinleri kart sıralamasına göre programlanır.

C) Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

13. Görselde gösterilen alana gelebilecek nesne aşağıdakilerden hangisidir?

A) Elektronik devre kartı girişleri
B) Elektronik devre kartı çıkışları
C) Elektronik devre kartı kabloları
D) Elektronik devre kartı modülleri
E) Elektronik devre kartı bölümleri



14. Proje oluştururken blok kodlama yapmak için aşağıdakilerden hangisi seçilmelidir?

A) Blink–Visual
B) Blink–Python
C) Empty–Visual
D) Empty–Pyhton
E) Empty–JavaScript

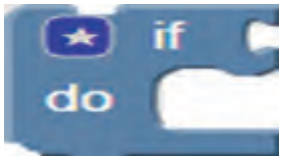
15. Görseldeki bloğun görevini aşağıdakilerden hangisi en iyi tanımlar?

A) Döngü
B) Değişken
C) Fonksiyon
D) Mantıksal karar
E) Giriş-Çıkış birimi

16. Görseldeki bloğun görevini aşağıdakilerden hangisi en iyi tanımlar?

A) Döngü
B) TCP fonksiyonu
C) Text fonksiyonu
D) HTTP fonksiyonu
E) Matematik fonksiyonu

17. Görseldeki bloğun görevini aşağıdakilerden hangisi en iyi tanımlar?



A) Dosya bilgileri
B) E-mail bilgileri
C) Mantıksal karar
D) Bluetooth bilgileri
E) Koordinat kullanımı

18. Aşağıdakilerden hangisi IoT sistemlerinde verimli iletişimin sağlanması için olması gereken bileşenlerden değildir?

A) Bağlantı
B) Kablolama
C) Kullanıcı arayüzü
D) Sensörler
E) Veri işleme

19. Aşağıdakilerden hangisi işlenmemiş yapıdadır?

A) Bilgi
B) Çıktı
C) Data
D) Girdi
E) Veri

20. Aşağıdaki Linux komutlarından hangisi dizin içeriğini listelemek için kullanılır?

A) cd
B) ls
C) mkdir
D) mv
E) pwd

21. Aşağıdakilerden hangisi su komutunun yaptığı görevi açıklar?

A) Arşiv dosyası oluşturur.
B) Yeni metin belgesi oluşturur.
C) Dosyadaki metinlerde arama yapılmasını sağlar.
D) Komut yöneticisi görevlerinin yapılmasını sağlar.
E) İşletim sisteminde tanımlı olan kullanıcılar arasında değişiklik yapar.

22. IoT cihazlarda iletişim kurulumunun sağlanması için aynı olması gereken değer bilgisi aşağıdakilerden hangisidir?

A) IoT cihaz
B) Nesne
C) Setting
D) SSID
E) Wireless

23. Aşağıdakilerden hangisi IoT Monitör ekranında cihaz listesi boş ise kontrol edilmelidir?

A) Desktop ayarları
B) Gateway ayarları
C) IoT Server ayarları
D) SSID ayarları
E) Wireless ayarları

24. Aşağıdakilerden hangisi IoT cihazlarda en sık kullanılan atak türüdür?

A) DDos
B) FreeDos
C) Malware
D) Oltalama
E) Şifre saldırısı

25. Aşağıdakilerden hangisi html, CSS, javascript ile hazırlanmış zararlı kodların kullanıcıların tarayıcısında izinsiz olarak çalıştırılmasına fırsat veren bir açıktır?

A) Cross Site Scripting
B) Güvensiz doğrudan nesne referansları
C) Hatalı güvenlik yapılandırması
D) SQL injeksiyon
E) URL yönlendirme

26. IoT cihazlarda güvenliği sağlamak için aşağıdakilerden hangisine dikkat edilmesi gerekir?

A) Cihazları dışarıdan müdahaleye açık şekilde bulundurma
B) Cihaz güncelleştirmelerine gerek duyulmadan kullanma
C) Kullanım ömrü dolan cihazları kullanılmaya devam etme
D) Kimlik doğrulamaya gerek duymama
E) Cihazlara ait dinamik testleri yapma

BİLGİSAYAR AĞLARI, SİS VE BULUT BİLİŞİM

4. Öğrenme Birimi

KONULAR

4.1. YEREL VE GENEL ALAN AĞLARI
4.2. KABLOLU VE KABLOSUZ ORTAMLAR
4.3. AĞ PROTOKOLLERİ
4.4. IoT KABLOSUZ İLETİŞİM TEKNOLOJİLERİ
4.5. IoT PROTOKOLLERİ
4.6. SİS VE BULUT BİLİŞİM
4.7. BÜYÜK VERİ
4.8. BULUT BİLİŞİMDE GÜVENLİK

NELER ÖĞRENECEKSİNİZ?

- Nesnelerin İnternetinde ağ kavramlarını tanımlama
- Ağları çeşitli yönlerden sınıflandırma
- Kablolu ve kablosuz ortamda kullanılan ortamlar
- Nesnelerin İnterneti kapsamında ağ protokolleri
- Nesnelerin İnternetine özgü protokolleri tanımlama
- Kablosuz IoT iletişim teknolojileri
- Sis ve Bulut Bilişim kavramları
- Büyük verinin IoT içindeki yeri
- Bulut Bilişimde güvenliğin önemi
- MQTT protokolünü kullanarak uygulama yapma
- Simülasyon programı aracılığı ile küçük bir IoT ağı geliştirme

TEMEL KAVRAMLAR

AMQP, big data, bluetooth, cloud, CoAP, HDFS, LAN, LoRAWAN, MAN, MapReduce, MQTT, protokol, RFID, sis Bilişim, WAN, XMPP, ZigBee, 6LoWPAN

HAZIRLIK ÇALIŞMALARI

1. Kablosuz iletişimin hayatımıza sağladığı kolaylıklar nelerdir?
2. Büyük verilerin depolanmasının zorlukları neler olabilir?
3. Otobüs ve akıllı durakların iletişimi nasıl gerçekleşir?
4. Herkese açık olan bulut bilişim uygulamaları nelerdir? Araştırınız.

http://kitap.eba.gov.tr/KodSor.php?KOD=14399

4.1. YEREL VE GENEL ALAN AĞLARI

Nesnelerin İnterneti'ndeki nesne kavramı aslında teknik olarak her şeyi ifade eder. Bu nesneler, çeşitli mikrodenetleyiciler ve sensörler ile birlikte kullanılarak çeşitli uygulamalar geliştirilir. Bu uygulamalar endüstride ve robotik alanda sıkça kullanılır. Nesnelerin İnterneti uygulamalarını bu uygulamalardan ayıran şey, ağa bağlanabilme yeteneğidir. Bu sayede herhangi bir endüstri uygulaması, bir robotik uygulaması ya da giyilebilir teknoloji uygulaması Nesnelerin İnterneti'nin konusu olabilmektedir. Bu sayede ağ iletişiminin sunduğu çeşitli avantajlar, IoT alanına taşınmıştır.

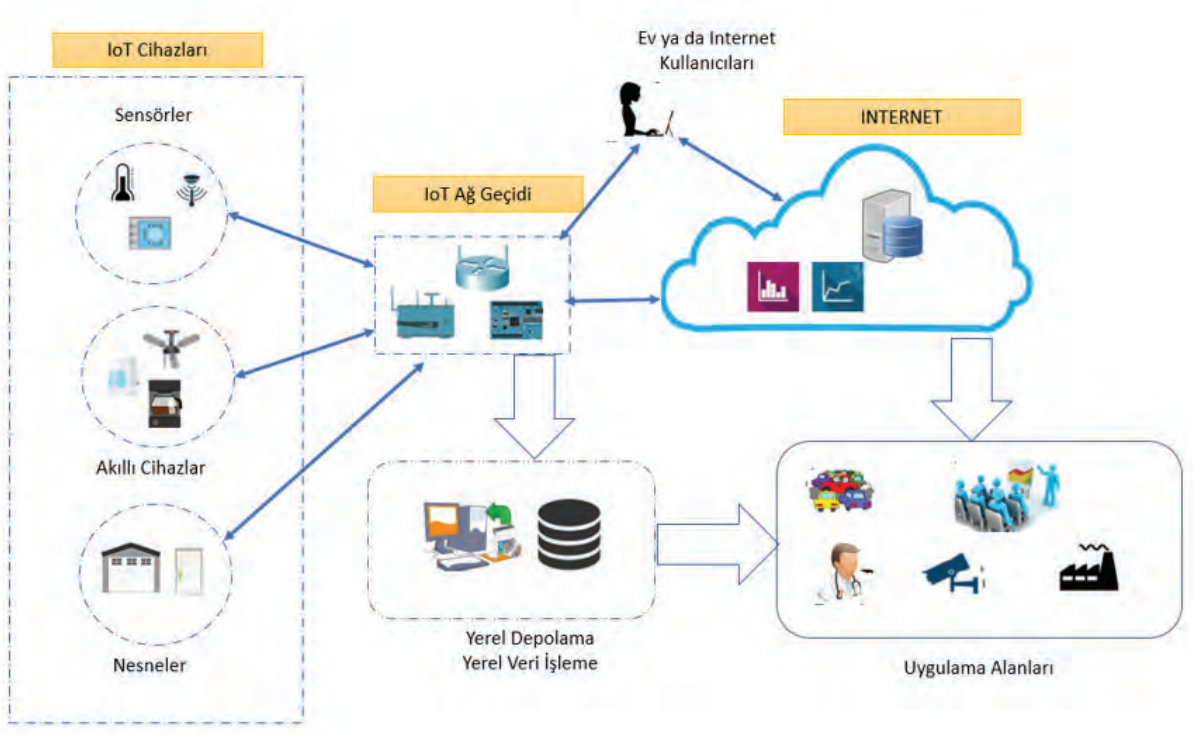
IoT alanında yapılan uygulamalar, ev ağı gibi küçük ölçekli bir ağ ortamından, internet tabanlı bulut uygulamalarına kadar geniş yelpazede birçok kullanım alanı sunmuştur. Mevcut ağ üzerine rahatlıkla ya da çok az bir çaba ile eklenebilmesi IoT'yi internetin bir uzantısı yapmıştır.

Küçük ev ağında ya da internet tabanlı bulut yapılarında IoT uygulamalarını geliştirebilmek için öncelikle temel ağ kavramları bilinmelidir.

4.1.1. IoT Ekosistemi

Nesnelerin İnterneti, son yıllarda oldukça popüler hâle gelen, hemen her gün yeni uygulamalarının çevrede görüldüğü ve gittikçe de yaygınlaşan bir alandır. Bu hızlı gelişimin arkasında yatan sebepler; iletişim teknolojilerinde yaşanan gelişmeler, yeni iletişim standartlarının oluşması ve buna paralel olarak daha ucuz ağ teknolojilerinin gelişmesidir.

Nesnelerin İnterneti sadece yeni teknolojiler ile değil, mevcut olan diğer ağ teknolojileri ile de uyumlu çalışması, hâlihazırda var olan ağ sistemlerine de kolaylıkla entegre edilebilen uygulamaların geliştirilmesini de sağlamıştır.



Görsel 4.1: IoT ekosistemi

Görsel 4.1'de örnek bir IoT topolojisi görülmektedir. Burada IoT yeni bir internet değil, mevcut internetin bir uzantısı ve genişlemesi olarak ele alınır. Çoğu zaman böyle bir ağ, **IoT ekosistemi** olarak adlandırılır.

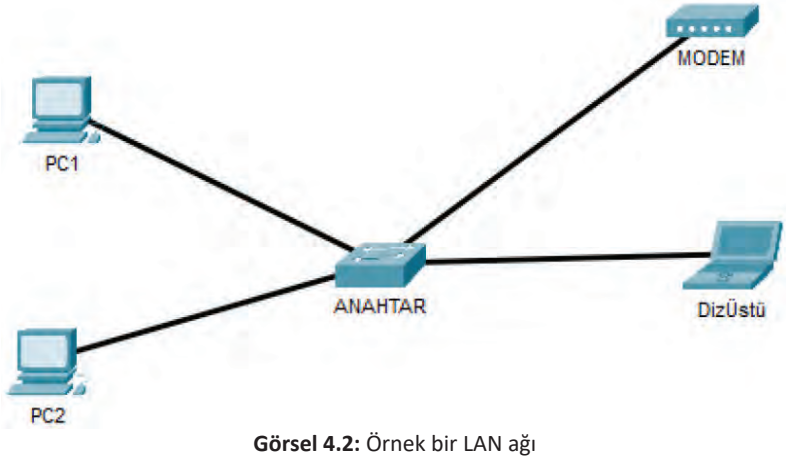
4.1.2. LAN

Yerel Alan Ağları (Local Area Network) kelimesinin kısaltması olan LAN, genel itibariyle coğrafi olarak küçük bir alanda bulunan ve bir otorite tarafından yönetilen ağları tanımlamak için kullanılan bir kavramdır.

Bir ev ağı göz önüne alındığında, bir cihaza kablolu ya da kablosuz olarak bağlanan sınırlı sayıda cihazdan oluştuğu ve bu cihazların kontrolünün evdeki kullanıcılara ait olduğu görülür. O hâlde bir ev ağı, temel olarak bir LAN ağıdır. Aynı mantıkla bir şirkette bilgisayarların, ağ yazıcılarının, sunucuların ve çeşitli ağ donanımlarının bulunduğu yapılar da yine bir LAN oluşturur.

Görsel 4.2'de görülen Yerel Alan Ağlarının özelliklerine bakıldığında genel olarak aşağıdaki hususlardan bahsedilebilir:

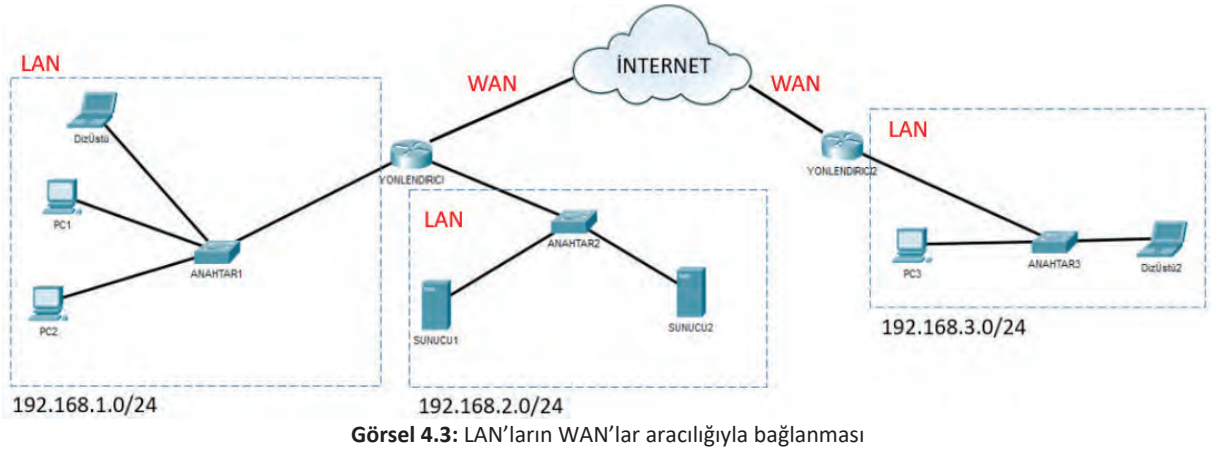
- Coğrafi olarak küçük bir alanda yer alır.
- Bir kullanıcı ya da kullanıcı grubunun kontrolündedir.
- Yüksek hızlı ağ kapasitesi vardır.
- Cihazlar Switch ve / veya Access Point gibi cihazlar kullanılarak birbirine bağlanır.



Görsel 4.2: Örnek bir LAN ağı

4.1.3. WAN

Temel olarak her bir ev ya da şirket ağı LAN'a örnektir. O hâlde dünya üzerinde milyonlarca hatta milyarlarca LAN'ın varlığından bahsedilebilir. Ancak bu LAN ağlarının sadece kendi küçük sınırları içinde sunduğu avantaj yeterli değildir. LAN'ların da birbiri ile iletişime geçmesi gerekir. Örneğin bir şirkette yer alan bir sunucudan bir verinin alınması ihtiyacı doğabilir ya da daha basit bir ifadeyle farklı bir konumda (LAN'da) bulunan bir kullanıcı ile anlık olarak iletişime geçmek gerekebilir. Bu durum, her biri farklı bir otoritenin kontrolünde olan LAN'ların da birbiri ile bağlantılı olması gerektiği gerçeğini doğurur. Görsel 4.3'te LAN ve WAN arasındaki ilişki gösterilmektedir.



Görsel 4.3: LAN'ların WAN'lar aracılığıyla bağlanması

Yerel Alan Ağlarının birbiri ile iletişim kurmasını sağlayan yapılara **Geniş Alan Ağları (WAN - Wide Area Network)** denir. WAN'lar, LAN'lar arası iletişimi sağlamak üzere telekomünikasyon kurumları tarafından belirli ücret karşılığında kullanıcılara verilen bir hizmettir.

4.1.4. Diğer Ağ Kavramları

Boyutları ve kullanım alanı itibariyle ağ sınıflandırması yaparken LAN ve WAN kavramlarının dışında WBAN ve PAN gibi iki sınıftan daha bahsedilebilir.

Kişisel Alan Ağları (PAN-Personal Area Network): Genellikle bir cihazın çevresinde yer alan cihazlar ile kurduğu küçük çaplı ağlardır. Bir bilgisayar ile yazıcı veya kamera gibi donanımlara bağlantı kurarak oluşturdukları ağlardır. Bu ağ, bluetooth gibi kablosuz bir teknoloji ile sağlanıyorsa WPAN adını alır.

WBAN (Wireless Body Area Network): Sensörler ve çeşitli medikal teknolojiler ile insan vücuduna bağlanan cihazların oluşturduğu bazen BAN olarak da adlandırılan ağdır. BAN cihazları, implantlar olarak vücuda gömülebildiği gibi giysilerle ya da taşınabilir cihazlarla da sağlanabilir. WBAN teknolojileri özellikle sağlık alanında geniş uygulamalar sunduğu için Nesnelerin İnterneti'nde önemli role sahiptir.

Ağ sınıflandırmasında kullanılan bu kavramlar, iletişim teknolojisi kablosuz olduğunda da farklı isimlerle anılır. Örneğin LAN ağının kablosuz olarak sunulması WLAN, MAN ağının kablosuz olarak sunulmasına WMAN ve aynı şekilde WAN'ın kablosuz olarak sunulmasına da WWAN denilir.

MAN (Metropolitan Area Network), birçok LAN sistemini barındıran, coğrafi ölçekte LAN ve WAN arasında büyüklüğe sahip olan bir ağ sistemidir. Metropol alan ağı genellikle bir şehir veya kampüste kullanılır.

4.2. KABLOLU VE KABLOSUZ ORTAMLAR

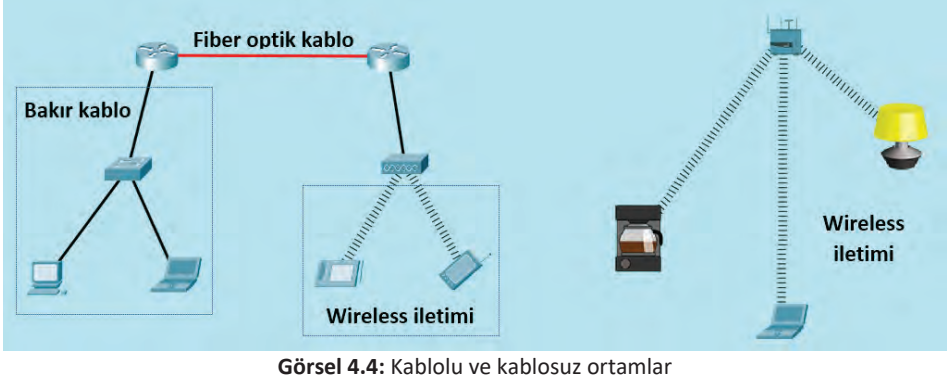
Ağda, mesaj iletiminin kaynaktan hedefe doğru yapıldığı fiziksel kanala ortam (medya) adı verilir. Bu ortamlar kablolu veya kablosuz şekilde iletişim sağlar (Görsel 4.4).

Kablolu ortamlarda bakır ve fiber optik kablolar kullanılır.

- Bakır kablolarda veriler elektriksel darbelerle kodlanır.
- Fiber optik kablolarda ise veriler ışık darbeleri ile kodlanır.

Kablosuz ortamlarda wireless iletimi kullanılır.

- Wireless iletiminde veriler, radyo dalgaları kullanılarak kodlanır.



Görsel 4.4: Kablolu ve kablosuz ortamlar

ARAŞTIRMA

Ağlarda kullanılan bakır ve fiber optik kablolar hakkında araştırma yapınız. Araştırmanızı görsellerle destekleyiniz. Araştırma sonuçlarını, bir sunum hâline getirerek sınıfta arkadaşlarınızla paylaşınız.

4.3. AĞ PROTOKOLLERİ

Gerçek hayatta iletişim kurarken uyulması gereken kurallar olduğu gibi bilgisayar ağlarında da cihazların kendi aralarında konuşurken uymak zorunda olduğu kurallar vardır. Bilgisayar ağları (network) dünyasındaki bu kuralları protokoller belirler. Bir iletişim söz konusu olduğunda aynı dili konuşmak, anlamak ve anlaşılmak insanlar için ne kadar önemliyse cihazların haberleşmesinde de aynı dili kullanmak o derece önemlidir. İnsanlar arasındaki iletişim esnasında dil, konuşma hızı, cümle yapısı, konuşma sırasını bekleme, karşıdakinin sözünü kesmeme, anlaşılmadığı durumda tekrar etme, anlaşıldığına dair karşıdan bir onay bekleme gibi kuralların bilgisayar ağları dünyasına uyarlanması ve bu iletişimin alıcı ile gönderici açısından anlaşılır bir hâle getirilmesinden sorumlu kurallar bütünü **protokol** olarak tanımlanabilir.

Protokoller; farklı üreticilerin, farklı ürünlerinin aynı platformda iletişimini (standardizasyonu) sağlar. Protokoller, ağ iletişiminin sağlıklı bir şekilde gerçekleştirilmesini sağlayan kurallardır. Protokoller; cihazların birbirleri ile nasıl iletişim kuracağı, veri alışverişinin nasıl olacağı, verilerin doğru hedefe doğru şekilde transfer edilmesi için gerekli olan ek bilgileri ve bu bilgilerin iletişimde nasıl kullanılacağını belirtir. Ağ kapsamında yer alan protokoller, birbiriyle ilişkili birden çok protokolden oluşur. Bu nedenle genellikle **protokol kümesi** olarak adlandırılır.

Bilgisayar ağları ilk ortaya çıktığında birçok farklı protokol kümesi bulunmasına rağmen günümüzde en yaygın kullanılan ve diğerlerinin yerini alan **TCP/IP** protokol kümesidir. Bu protokol kümesinde onlarca farklı protokol yer alır. Ancak en önemlileri **TCP** ve **IP** olan iki protokoldür. Bu nedenle protokol kümesi bu ikisinin adıyla anılır.

4.3.1. İnternet Protokolü (IP)

TCP/IP protokol kümesinde yer alan ve kümeye ismini veren esas protokoldür. Genel olarak bu protokolün amacı, adresleme sağlamak ve veriyi taşımaktır. IP'nin sunduğu adreslemenin anlaşılabilmesi için **bir noktadan diğerine kargo gönderme** analojisi yararlı olacaktır. Kargo gönderirken gönderilen paketin doğru hedefe ulaşması için paket üzerine, alıcı adı ve adresi olmak üzere iki bilgi yazılır. Kargo üzerine yazılan adres; kargonun alıcısını tanımlayan, alıcısını bulmayı sağlayan bir bilgidir. Kargo gönderiminde kullanılan **alıcı adresi** kavramına karşılık ağ cihazları arasındaki bu amacı internet protokolü (IP) karşılar. Yalnız adres bilgisindeki cadde, sokak gibi kavramlar yerine ağ cihazları arasındaki iletişimde ikili sistemdeki sayılar kullanılır. Günümüzde 32 bitlik IPv4 ve 128 bitlik IPv6 olmak üzere iki farklı IP sürümü bulunmaktadır.

Örneğin bir IPv4 adresi aşağıdaki gibi 32 bitten oluşabilir:

11000000101010000000000100000111

Bu adres bilgisi, günlük hayatında onluk sayı sistemini kullanan insanlar için kolay okunabilir değildir. Ayrıca yapılandırma esnasında bu ifadenin ikilik olarak yazılması, büyük olasılıkla hata yapılmasına neden olacaktır. Bu nedenle makinelerin gördüğü bu adres yapısı, insan kullanımı için onluk sayı sistemine bölünerek dönüştürülür. 32 bitlik bu IPv4 adresi her biri 8 bit (1 byte) olan 4 bölüme (oktet) ayrılır ve aralarına birer nokta konur.

11000000.10101000.00000001.00000111

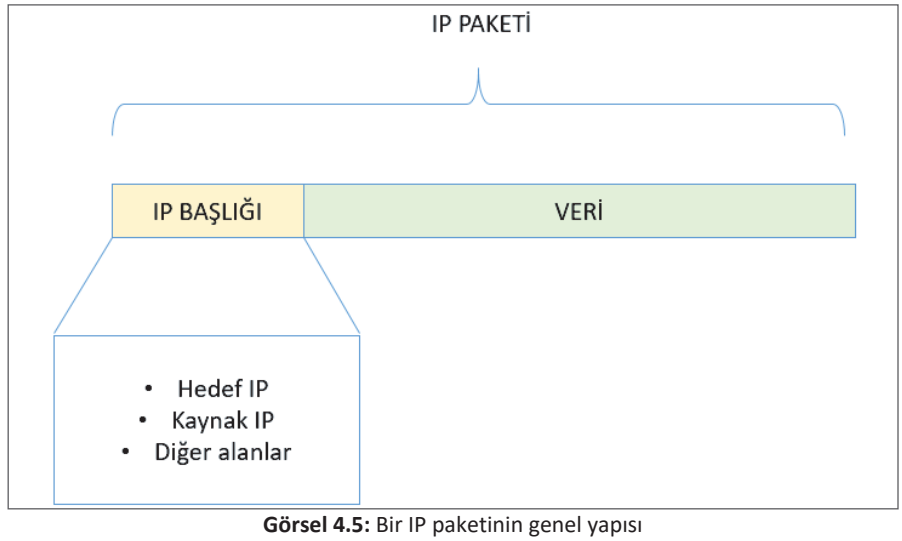
Bu işlemin ardından her sekizli bölüm onluk sayı sistemine dönüştürülür.

192.168.1.7

IP adresleri, cihazları tanımlayan tekil ve benzersiz sayılardır. Ağ üzerinde yer alan her cihazın iletişim kurmak için bir IP adresinin olması gerekir. Bir kullanıcının taşıdığı bir akıllı telefon, web sayfasını barındıran bir web

sunucusu ya da evdeki sıcaklık bilgisini internet ortamında bir konuma gönderen bir IoT mikrodenetleyicisinin bir IP adresi vardır.

IP protokolünün diğer bir amacı da veriyi taşımaktır. Bilgisayar ağlarında kaynak ile hedef arasındaki her iletişim bir veri taşır. Taşınan büyük ya da küçük her veri, IP protokolünün sorumluluğunda taşınır. Diğer bir ifadeyle verinin hangi adresten geldiği ve hangi adrese gideceği gibi bilgiler, taşınan veriye eklenir. Veriden ve verinin doğru adrese ulaşmasını sağlayan ek bilgilerden oluşan bu yapıya **IP paketi** denir. Bir IP paketi, Görsel 4.5'te gösterildiği gibi veriden ve adresleme bilgisi gibi bilgileri taşıyan başlık bilgilerinden oluşur.



Görsel 4.5: Bir IP paketinin genel yapısı

Bilgisayar ağlarında kaynak ile hedef arasında sadece sıcaklık verisini taşıyan küçük veriler gönderilebildiği gibi video gibi büyük veriler de gönderilebilir. Ancak IP'nin tek bir pakette taşıyabileceği veri sınırlıdır. Aynı şekilde, iletişimin kurulduğu kablolu ya da kablosuz ortamın da taşıyabileceği bir sınırı vardır. Bu nedenle IP paketleri duruma göre parçalanarak gönderilebilir. Parçalanma durumunda, parçaya ilişkin bilgiler, IP başlık içinde çeşitli alanlar ile gösterilir.

IP'nin en önemli unsurlarından biri de verinin taşınmasını sağlamaktır. Ancak verinin taşınmasına ilişkin nasıl taşınacağı, paketin hedefe ulaşamaması durumunda neler olacağı, veri alıcı cihaza ulaştığında hangi uygulamayla ilişkilendirileceği gibi bilgiler IP içinde bulunmaz. Bu ihtiyacı karşılamak amacıyla TCP ve UDP gibi taşıma protokolleri kullanılır.

4.3.2. TCP ve UDP Protokolleri

TCP: Uygulamalar ve hizmetler arası bağlantıyı kuran güvenilir bir protokoldür. Verinin kaynaktaki hangi uygulamadan çıkıp hedefteki hangi uygulamaya gittiği, TCP başlık bilgisinde yer alan ve her biri 16 bit olan kaynak ve hedef port numaraları tarafından belirlenir.

TCP'nin güvenilir olması, veri transferinde alıcı cihazın veriyi aldığını doğrulaması (Onay-Ack) ve olası paket kayıplarında paketi tekrar göndermesi ile ilişkilidir. Bir başka ifadeyle verinin karşı uygulama tarafından alınıp alınmadığı bilinir, bu da TCP'yi güvenilir yapar. TCP, verilerin parçalar hâlinde gönderildiği durumlarda parçalara sıra numarası (Sıra-sequence) ekleyerek alıcı tarafta doğru bir şekilde sıralanmasını sağlar. TCP veri iletişimini başlatmadan önce 3-yollu el sıkışma işlemini yapar. Böylece gönderici ile alıcı arasında bir anlaşma niteliğinde olan oturum kurulmuş olur.

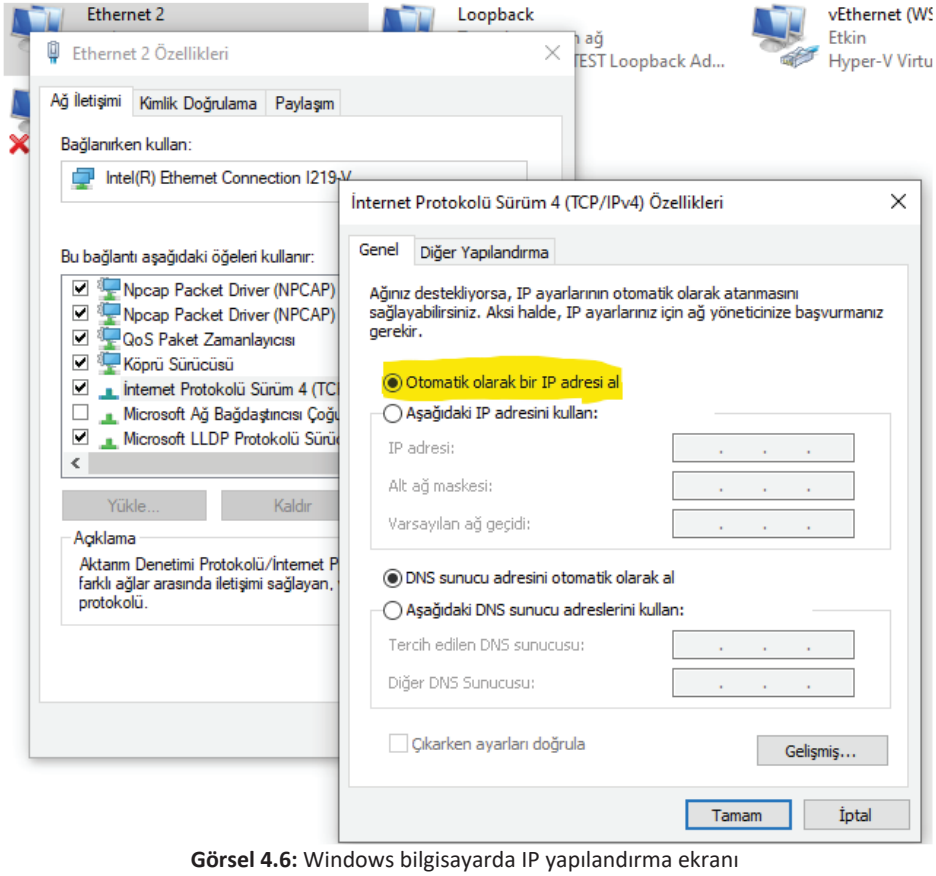
IoT uygulamalarında kullanılan cihazların genellikle kısıtlı cihazlar olduğu göz önüne alınırsa TCP protokolünün bu cihazlara ek yük getireceği görülür. Bu nedenle başlangıçta TCP ile geliştirilen bazı uygulamalar zamanla yerini UDP gibi daha basit protokollere bırakmıştır.

UDP: Uygulamalar ya da hizmetler arasında veri iletimini sağlayan diğer bir protokoldür. UDP, port numaralarını kullanarak uygulama verilerini iletir ancak TCP'ye göre oldukça basit bir yapıdadır. Verilerin ulaşıp ulaşmadığını kontrol etmez, gönderilmeyen paketleri tekrar iletmez, sıralı değildir. Bu nedenle güvensizdir, başlık yapısı basittir.

UDP, hız gerektiren ve paket kayıplarının tolere edilebildiği ses akışı, video akışı gibi uygulamalarda tercih edilir. Basit yapısı nedeniyle IoT uygulamalarında oldukça yaygın kullanılır.

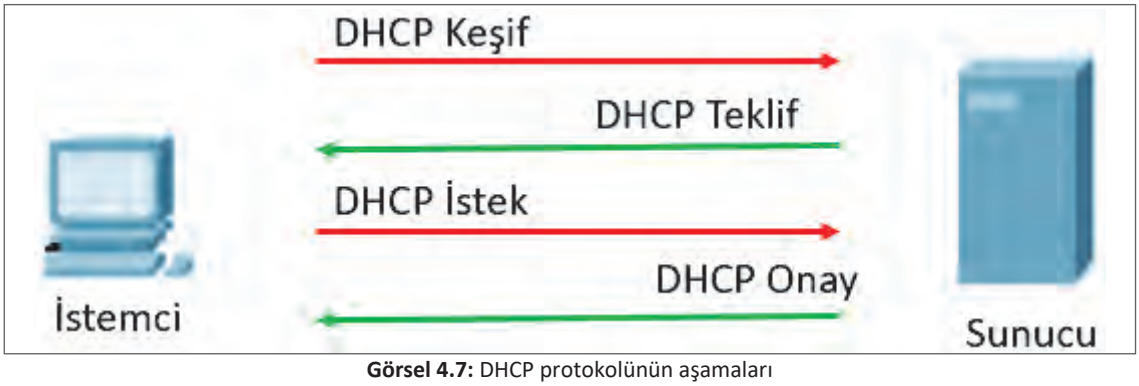
4.3.3. Diğer Protokoller ve Teknolojiler

Dinamik Host Yapılandırma Protokolü (DHCP): Cihazların ağ iletişimi kurabilmesi için benzersiz IP adreslerinin olması gerekir. Bu adresler cihazlara statik olarak atanabildiği gibi adreslerin otomatik olarak alınması da sağlanabilir. DHCP, bu amaç doğrultusunda geliştirilen bir protokoldür. Cihazların ihtiyaç duyduğu IP, Alt Ağ Maskesi, Ağ Geçidi ve DNS adresi gibi bilgileri cihazlara kiralar. Windows işletim sistemlerinde ihtiyaç duyulan bu bilgiler, Görsel 4.6'daki IP yapılandırma ekranında gösterilmiştir.



Görsel 4.6: Windows bilgisayarda IP yapılandırma ekranı

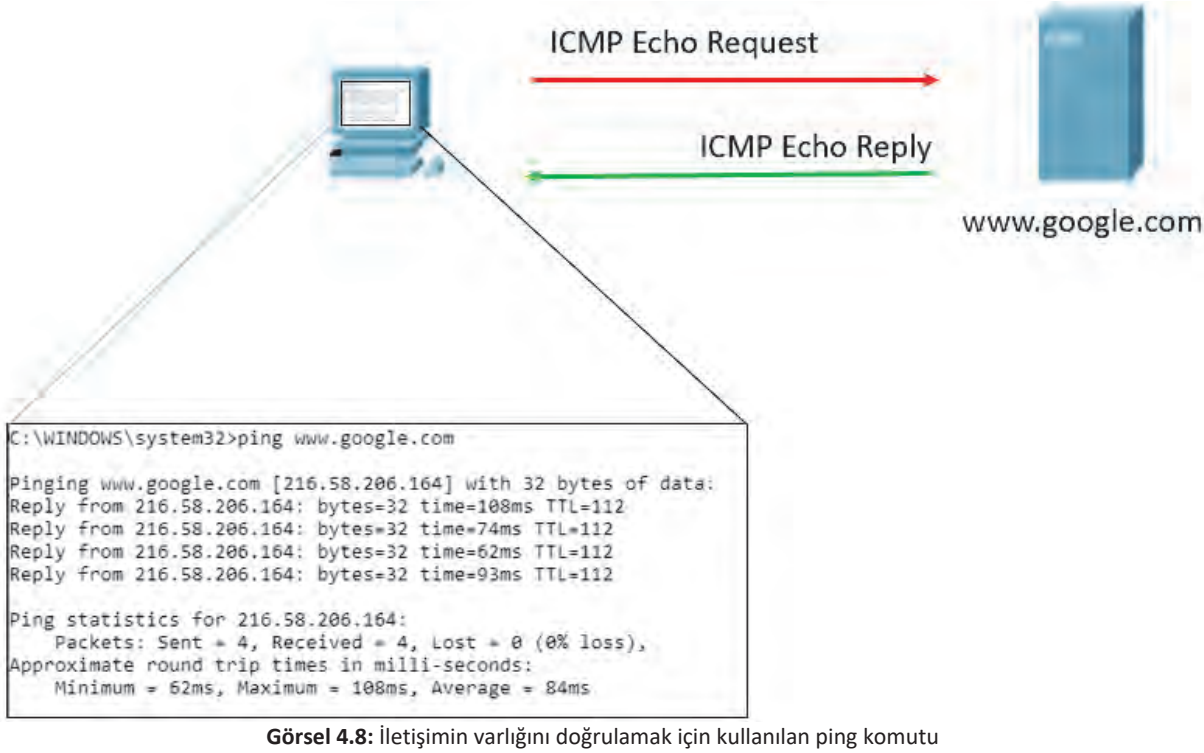
Otomatik olarak bir IP adresi al seçeneği işaretlendiğinde DHCP devreye girer ve ihtiyaç duyulan bilgileri bulmak üzere DHCP işlemini başlatır. İstemci cihazlara verilecek olan IP yapılandırma bilgileri DHCP sunucusu olarak adlandırılan bir cihazda önceden tanımlı olmalıdır. İstemci ile sunucu arasında toplamda dört aşamadan oluşan DHCP'nin işlem akışları ve isimleri Görsel 4.7'de verilmiştir.



Görsel 4.7: DHCP protokolünün aşamaları

Küçük ağlarda (örneğin ev ağlarında) modem-yönlendiriciler, IP dağıtmak üzere hazır olarak tasarlanmış DHCP sunucuları olarak çalışır. Bu nedenle ev ağlarında modeme bağlanan cihaz, ek bir yapılandırmaya gerek kalmadan IP adresi alabilir.

İnternet Kontrol Mesaj Protokolü (ICMP): Bir ağ ortamında gerek sorun gidermek gerek iletişimin varlığının doğrulanması için kontrol edilme ihtiyacı vardır. ICMP, bu amaç doğrultusunda geliştirilmiş ve belirli bir IP'nin erişilebilir olup olmadığını doğrulamak için kullanılan bir protokoldür. ICMP; erişimin olup olmadığının yanı sıra erişimin olmamasının olası nedenini, erişilen cihaza ve uzaklığına dair çeşitli bilgileri de sunan yararlı bir protokoldür. Görsel 4.8'deki gibi **ping** ve **traceroute** (Windows'ta **tracert**) gibi komutlar bu protokolü kullanır.



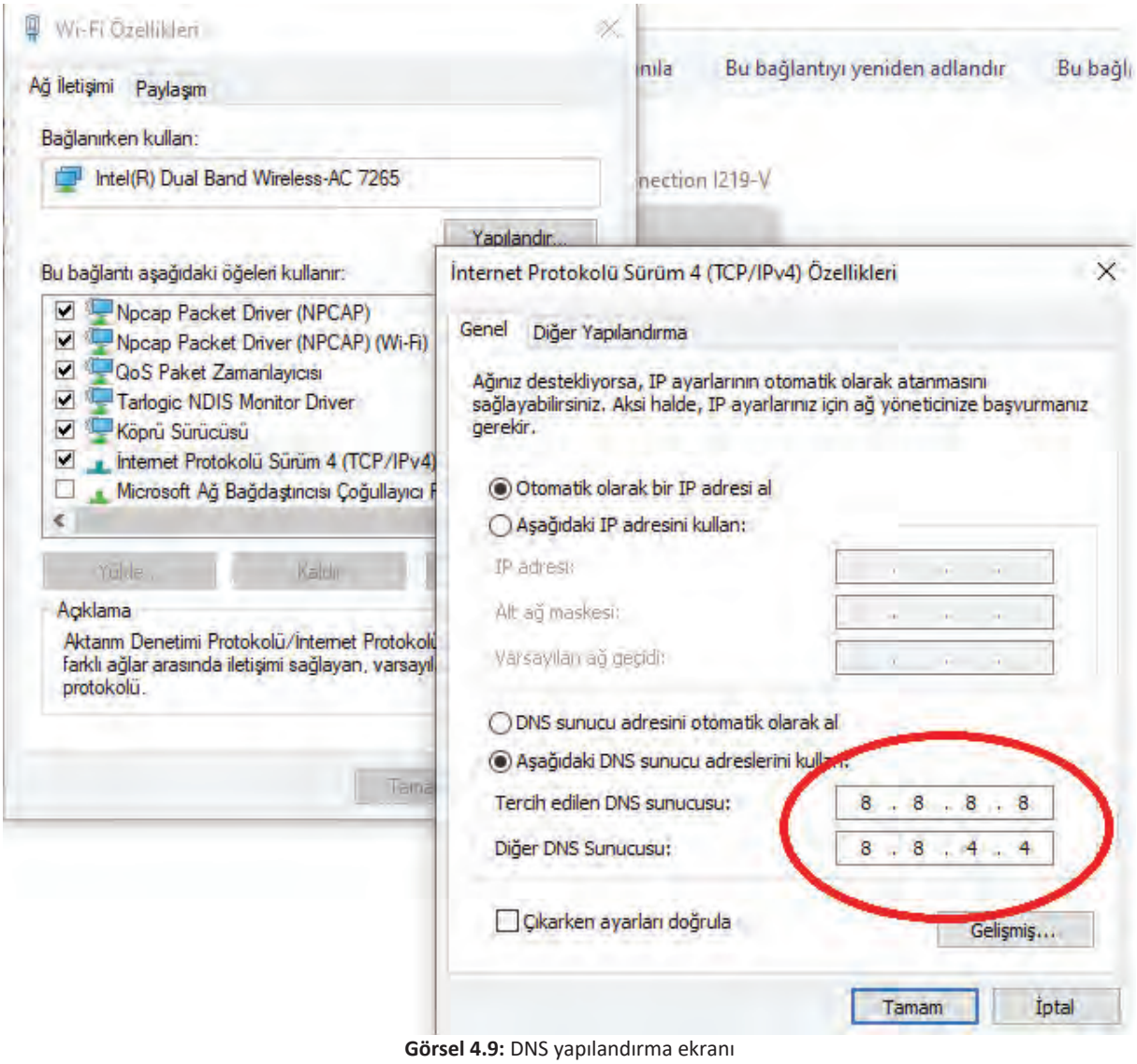
Görsel 4.8: İletişimin varlığını doğrulamak için kullanılan ping komutu

İletişimin varlığını doğrulamak isteyen kaynak, ICMP yankı isteği (**Echo Request**) gönderir. Bu isteği alan hedef cihaz, (cevap vermemek üzere yapılandırılmadığı sürece) ICMP yankı cevabı (**Echo Reply**) ile cevap verir. Cevaptaki TTL değeri cihazın işletim sistemi hakkında ipucu ve uzaklığı (**hop count**) hakkında yaklaşık bir bilgi verir.

HiperText Transfer Protokolü (HTTP): İnternet ortamında en yaygın kullanılan uygulama protokolüdür. World Wide Web'de (WWW) metin, grafik, görüntü, ses, video ve diğer multimedya dosyalarını değiş tokuş etmek için bir dizi kuralın tanımlandığı protokoldür. İnternet uygulamalarında en çok kullanılan uygulama protokolü olmasına rağmen IoT uygulamaları için gereksiz ve ek yük oluşturabilecek alanlar içerir.

Etki Alanı Sistemi (DNS): İnternet ve ağ ortamında cihazları adresleyen benzersiz tanımlayıcı IP adresidir. Web aracılığıyla iletişime geçilmek istenen cihazın IP adresi yazılarak iletişim kurulur. Ancak özellikle internet ortamında IP adresi yerine akılda daha kalıcı olduğu için isim kullanmak daha kullanışlıdır. Bu isimlere daha teknik bir ifadeyle **domain adı** denir. Ancak IP paketlerinde hedef adres kısmına domain adları değil, IP adresleri yazılabilir. Bu nedenle domain adı ile IP adreslerinin dönüştürülmesi gerekir. DNS, bu amaç doğrultusunda geliştirilmiş UDP kullanan bir uygulama protokolüdür. Erişilmek istenen domain adının IP adresine dönüştürülmesinin ardından IP-domain eşleşmesi kaynak cihazın belleğinde tutulur.

İnternet ortamında bir web adresine ismi ile erişilmek istendiğinde işletim sistemi öncelikle bu önbelleği araştırır. Önceden bir çözümleme yapıldıysa bellekte yer alır. Doğrudan bellekte bulunan adres kullanılır. Aksi durumda host cihazın Görsel 4.9'daki IP yapılandırma penceresinde belirtilen DNS sunucusuna isim çözümlemesi için hiyerarşik bir sorgu süreci başlatılır.



Görsel 4.9: DNS yapılandırma ekranı

Adres Çözümleme Protokolü (ARP): Bu protokol, bir IPv4 adresi ile fiziksel donanım adresi (MAC adresi) arasında dinamik adres eşlemesi sağlar. Ethernet iletişiminde farklı bir ağ ile iletişime geçiliyorsa ağ geçidinin, aynı ağdaki cihaz ile iletişime geçiliyorsa doğrudan hedef cihazın MAC adresinin bilinmesi gerekir. Bu işlem, kaynak cihaz tarafından başlatılan bir ARP İsteği (**ARP Request**) ile başlar. Bu istek broadcast bir mesajdır ve bu nedenle ağdaki tüm cihazlara gider. MAC adresi öğrenilmek istenen cihaz (aslında IP) ister doğrudan aynı ağdaki cihaz olsun ister ağ geçidi olsun bu ağdaki tüm cihazların aldığı gibi bu mesajı alır. Olması gereken ARP işleyişine göre sorgulanmak istenen IP adresine sahip cihaz dışındakilerin bu mesajı dikkate almaması ve cevap vermemesidir. Doğru cihaz, kendi MAC adresini içeren tekli yayın (unicast) bir cevap verir (**ARP Reply**). Böylece trafiği ilk başlatan cihaz, istediği MAC adresi öğrenir. Bu IP-MAC eşleşmesini belirli bir süreliğine belleğinde (arp cache) tutar.

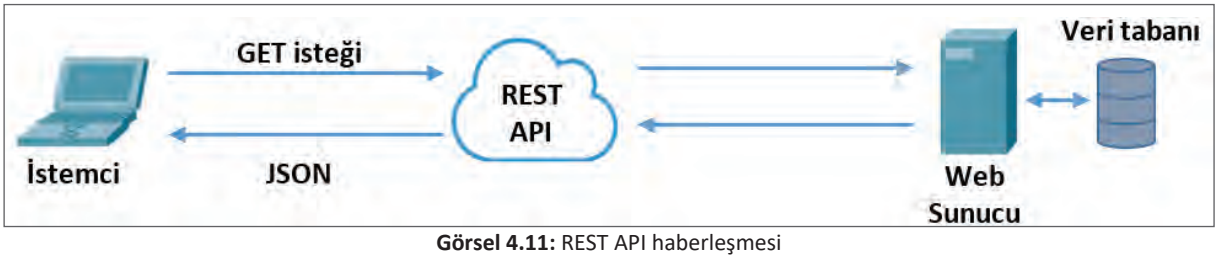
Windows işletim sisteminde arp önbelleğini görüntülemek için Görsel 4.10'daki ekran görüntüsünde olduğu gibi **arp -a** komutu kullanılabilir.

```
C:\>arp -a

Interface: 192.168.56.1 --- 0x5
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static

Interface: 192.168.1.103 --- 0xe
  Internet Address      Physical Address      Type
  192.168.1.1           5c-63-bf-06-d0-13     dynamic
  192.168.1.100         48-44-f7-51-46-df     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
```

Görsel 4.10: Windows işletim sisteminde Arp önbelleği

Temsili Durum Transferi (REST): İşletim sistemi ve programlama dilinden bağımsız şekilde cihazlar arasında bağlantı kurmak için kullanılan bir protokoldür. Bu protokol, istemci ve sunucu arasında HTTP üzerinden hızlı iletişim kurulmasını sağlar. Bu iletişim, istek / yanıt modeline dayanır. REST, HTTP metotlarını ve durum kodlarını kullanarak gelen isteklere genellikle JSON formatında yanıt verir.

REST haberleşmesi Görsel 4.11'de gösterilmiştir. Görsel 4.11'de istemci REST API'ye GET isteğinde bulunur. REST API gelen isteği işler. Web sunucusu üzerinden veri tabanı ile iletişime geçilir. Web sunucu REST API'ye yanıtı gönderir. REST API gelen yanıtı JSON formatında istemciye gönderir.



Görsel 4.11: REST API haberleşmesi

4.4. IoT KABLOSUZ İLETİŞİM TEKNOLOJİLERİ

İnternetin gelişim sürecinde açıkça görünen kablosuz teknolojiler, Nesnelerin İnterneti'nde de geniş kullanım alanı bulmuştur. Hatta bu kablosuz teknolojilerde yaşanan geliştirmeler ve kablosuz teknolojilerin ucuzlaması sonucu yaygın kullanımı, Nesnelerin İnterneti'nin bu denli yaygınlaşmasını sağlamıştır.

LAN, WAN ve PAN ağları çerçevesinde Nesnelerin İnterneti'ne konu olan kablosuz teknolojilerden bazıları şunlardır:

- 802.11 Wi-Fi ailesi
- RFID
- ZigBee
- 4G / 5G
- Lora
- Bluetooth

IoT ekosisteminde ağ ortamına bağlanan akıllı cihazların takibi ve kontrolü mümkündür. Bu cihazları ağa dâhil etmenin en kolay yolu Wi-Fi kullanımıdır. Örneğin evdeki kablosuz ağa Wi-Fi ile bağlanan akıllı lamba, yine aynı ağa bağlı cep telefonu üzerinden kontrol edilebilir. Çocuk odasındaki IP kamera görüntüleri Wi-Fi üzerinden takip edilebilir.

ARAŞTIRMA

IEEE 802.11, ZigBee standartları ve kullanım alanları hakkında araştırma yapınız. Araştırma sonuçlarını sınıfta arkadaşlarınızla paylaşınız.

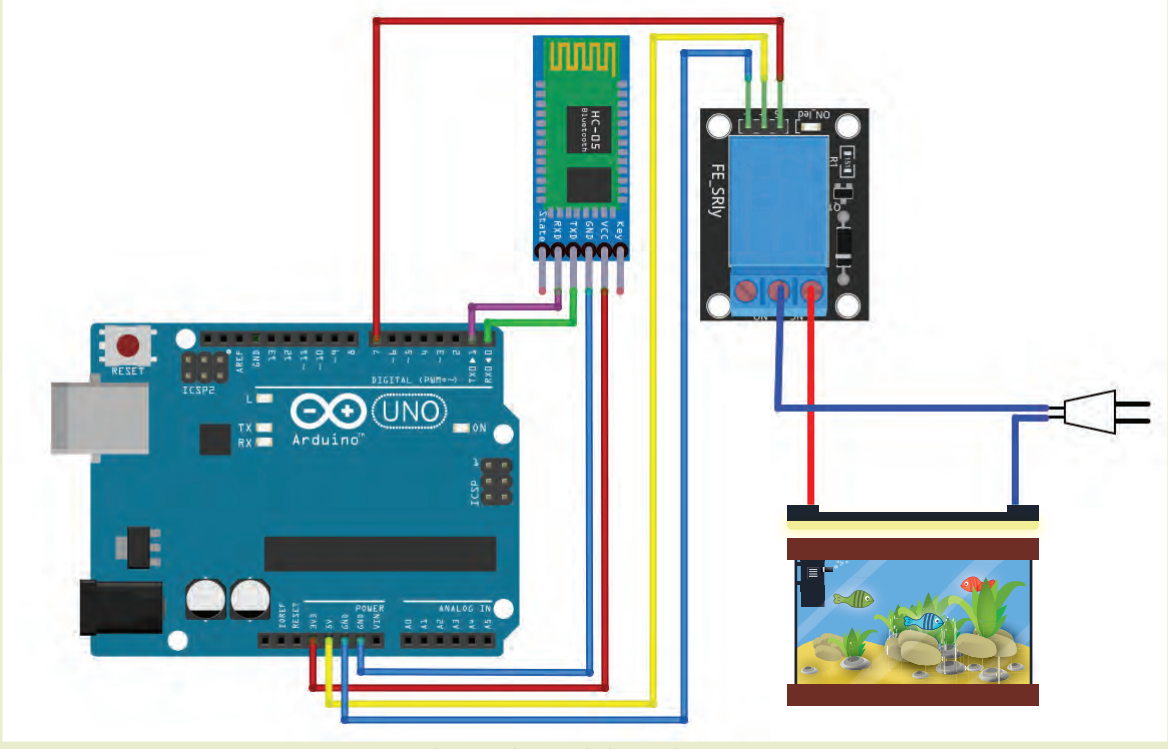
ZigBee, endüstriyel ortamlarda kullanılan düşük maliyetli, düşük veri hızlı ve düşük güçlü kablosuz mesh ağ standardıdır. Cihazdan cihaza ağlarının ihtiyaçlarını karşılamak için geliştirilmiştir.

4G, çoklu ortam uygulamalarını az enerji tüketimiyle izleme ve yüksek kaliteli bağlantı sağlama özelliklerine sahiptir. 4G teknolojisi ile bir tarladaki kameradan HD video akışı kesintisiz olarak takip edilebilir.

5G, hızlı veri aktarımı ve düşük güç tüketimi özellikleriyle IoT ekosisteminde kablosuz veri iletimi altyapısını güçlendirir. 5G teknolojisi ile kargo teslimatı yapan bir drone'un coğrafi konumu yüksek veri akışıyla takip edilebilir.

1. UYGULAMA

Arda evinde bulunan akvaryuma LED sistemi kurmuştur. Arda, LED sistemini mobil uygulama üzerinden Bluetooth yardımıyla kontrol etmek ister. Bunun için araştırmalar yapar. Araştırma sonucunda Arduino UNO, Bluetooth modülü ve röle satın alması gerektiğine karar verir. Gerekli malzemeleri satın alır. Daha sonra Görsel 4.12'deki devre şemasını kurmaya çalışır. Arda'ya devre şemasını kurması, Arduino ve mobil uygulama programlarını hazırlaması adımlarında yardımcı olunuz.

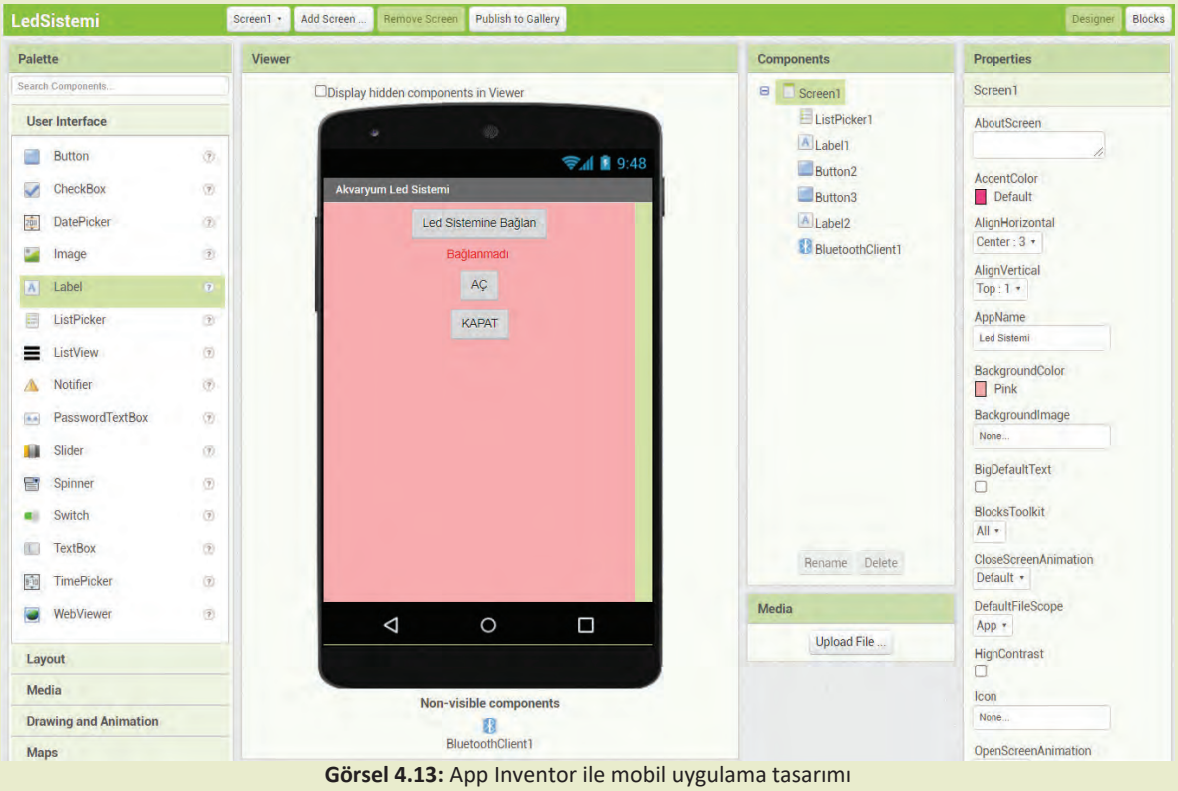


Görsel 4.12: Bluetooth ile IoT devre şeması

1. Adım : Devre şemasını kurunuz.
2. Adım : Arduino gömülü yazılım kodunu yazınız. Kodu Arduino kartınıza yükleyiniz. Yükleme esnasında Arduino 0 ve 1 numaralı pinlerden kabloları çıkarınız. Yükleme tamamlanınca tekrar takınız.

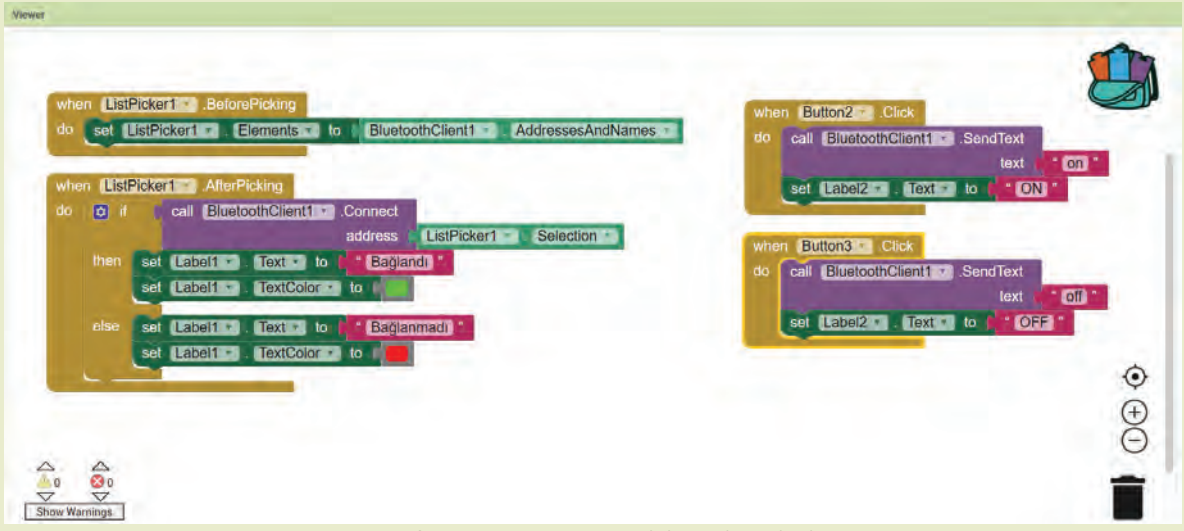
```
int role=7;
String yanit="";
void setup()
{
 Serial.begin(9600);
 pinMode(role, OUTPUT);
}
void loop()
{
  if(Serial.available())
   {
    yanit=Serial.readString();
    if(yanit=="on")
       digitalWrite(role, 1);
    else if(yanit=="off")
       digitalWrite(role, 0);
   }
}
```

3. Adım : Mobil uygulama tasarımını App Inventor ile hazırlayınız (Görsel 4.13).



Görsel 4.13: App Inventor ile mobil uygulama tasarımı

4. Adım : Mobil uygulama yazılımının kodunu App Inventor ile hazırlayınız (Görsel 4.14).



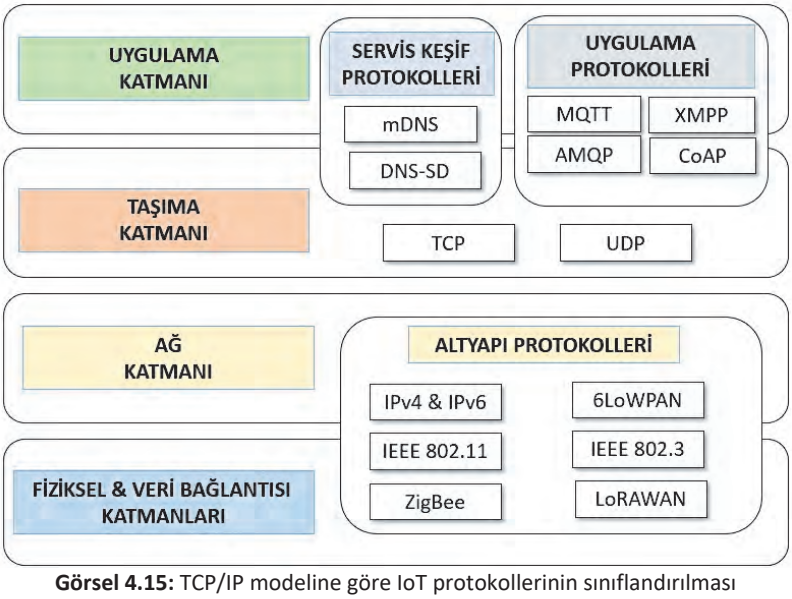
Görsel 4.14: App Inventor mobil uygulama kodu

5. Adım : App Inventor Build menüsünden Android App (.apk) seçeneği ile LedSistemi.apk dosyasını indiriniz. İndirilen dosyayı telefonunuza veya tabletinize kurunuz. LED sistemini kontrol ediniz.

4.5. IoT PROTOKOLLERİ

Nesnelerin İnterneti'nde iletişim teknolojileri kullanılarak çeşitli nesnelerin ağa bağlanabilirliği sağlanır. Bu sayede akıllı evler, giyilebilir akıllı cihazlar ve otomasyon sistemleri geliştirilebilir. Nesnelerin İnterneti'nde amaç genellikle insan hayatını kolaylaştıran uygulamalar geliştirmektir. Son kullanıcı açısından bakıldığında çoğunlukla nesneler arasındaki iletişimi sağlayan protokollerin varlığı görünür. Ancak Nesnelerin İnterneti'nin çalışmasında kullanıcıya görünmeyen altyapı protokolleri ve keşif protokolleri gibi farklı amaçlar doğrultusunda geliştirilen protokoller de vardır.

IoT protokolleri; Uygulama Protokolleri, Servis Keşif Protokolleri ve Altyapı Protokolleri olarak üç kategoride incelenir. Görsel 4.15'te IoT protokollerinin TCP/IP modelindeki konumları gösterilmiştir.



Görsel 4.15: TCP/IP modeline göre IoT protokollerinin sınıflandırılması

4.5.1. Altyapı Protokolleri

Nesnelerin İnterneti mevcut internet altyapısı üzerine kurulduğu için ağ iletişiminde kullanılan klasik protokoller de yine IoT protokolleri kapsamında incelenir. IoT uygulamaları mevcut IP ağı üzerinde çalışabilir. Bu nedenle standart IP ağlarında kullanılan Ethernet gibi teknolojileri ya da altyapı protokollerini kullanabilir. Kablolu Ethernet (IEEE 802.3) ya da Wi-Fi (IEEE 802.11 ailesi) hâlâ yerel alan ağlarında çok sık kullanılan bir standarttır. Bu nedenle Nesnelerin İnterneti'nin LAN uygulamalarında dolayısıyla IPv4 ya da IPv6 ağlarında kullanılabilir. Arduino ve Raspberry Pi gibi IoT denetleyicilerinde Ethernet desteği sunması sayesinde birçok IoT uygulama prototiplerinde ve küçük çaplı uygulamalarda kullanılır.

IPv4 ve IPv6: Nesnelerin İnterneti'nde yer alan cihazların adreslenmesinde, cihazların benzersiz olarak tanımlanmasında IPv4 ve IPv6 protokolleri kullanılır. Cihazların ürettiği veriler IP paketlerinde taşındığı için bu iki protokol IoT ekosisteminde yaygın olarak kullanılan protokollerdir.

6LoWPAN: Kısıtlı kaynaklara sahip cihazların IPv6 paketleri, IEEE 802.15.4 standardı ile tanımlanan düşük güç tüketimli WPAN üzerinden iletmesini sağlamak için kullanılır.

LoRAWAN: Düşük güçle (pille) çalışan kısıtlı cihazları, uygun maliyetle kablosuz geniş alan ağlarına bağlamak için tasarlanmış ağ protokolüdür. Endüstriyel IoT uygulamalarında kullanımı yaygındır.

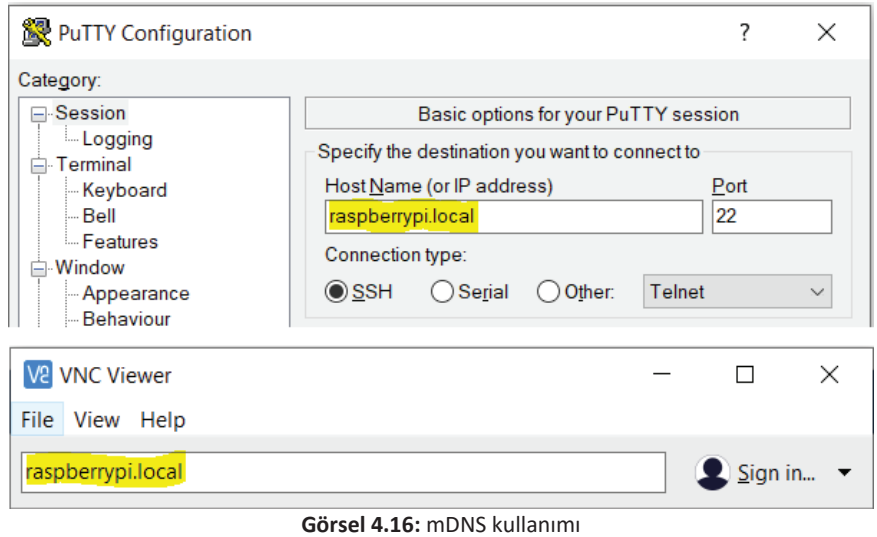
IEEE 802.3: Kablolu LAN yapısında kullanılan Ethernet teknolojisidir. LAN ağlarında geliştirilen IoT uygulamaları, Ethernet mimarisine dayanır. Bu nedenle fiziksel adreslemede MAC adresi olarak kullanılan tekil tanımlayıcıların kullanıldığı çerçeveler (frame) kullanılır. Ethernet Shield kartı desteği ile kullanılan Arduino cihazlarında bu teknoloji kullanılmaktadır.

Aşağıdaki örnek yapılandırmada Arduino Ethernet geliştirme kartında MAC adresinin nasıl kullanılacağı gösterilmiştir.

```
#include <SPI.h>
#include <Ethernet.h>

byte mac[] = {0xB4, 0x21, 0x8A, 0xF0, 0x17, 0xE8};
IPAddress ip(192, 168, 1, 177);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);
```

4.5.2. Servis Keşif Protokolleri

mDNS: DNS sunucusu olmayan yerel bir ağda bilgisayar adlarının, IP adreslerine çözümlenmesini sağlayan bir protokoldür. Bu protokol, yerel alan ağında çalışan bir Raspberry Pi'ye bilgisayar adını kullanarak SSH veya VNC ile bağlantı kurmayı sağlar (Görsel 4.16).



Görsel 4.16: mDNS kullanımı

NOT

Yerel mDNS çözümlemesi için Bonjour Print Services yazılımı kurulmalıdır.

DNS-SD: Hizmetleri etki alanı içindeki bilgisayar adlarıyla eşleştiren bir protokoldür.

4.5.3. Uygulama Protokolleri

Bu protokoller, OSI modelinin yedinci katmanı olan uygulama katmanında çalıştıkları için uygulama protokolleri olarak adlandırılır. Bunlar, kullanıcıların ya da yazılımların ihtiyaç duyduğu hizmetleri sağlayan, veri akışını gerçekleştiren ve kullanıcının doğrudan etkileşim kurduğu mesajlaşma protokolleridir.

Günümüz internetinde en yaygın kullanılan uygulama protokolü http'dir. HTTP, her ne kadar IoT uygulamalarında da çok yaygın olarak kullanılsa da IoT açısından hantal bir protokoldür çünkü IoT cihazları genellikle sınırlı kapasiteye sahip donanımlardır. Bu sınırlı cihazların IoT içindeki kullanımı, veri göndermek ve almak gibi genelde basit işlemlerdir. Oysa http, daha gelişmiş ve karmaşık web işlemleri için kullanılır. Bu nedenle veri iletişimi için daha basit ve cihazlarda çok az ek yük getiren protokollerin geliştirilmesine ihtiyaç duyulur.

IoT uygulamalarında kullanılan, IoT için geliştirilen ya da IoT'ye uyarlanan protokollerden bazıları şunlardır:

- MQTT
- CoAP
- AMQP
- XMPP

4.5.3.1. Message Queuing Telemetry Transport (MQTT)

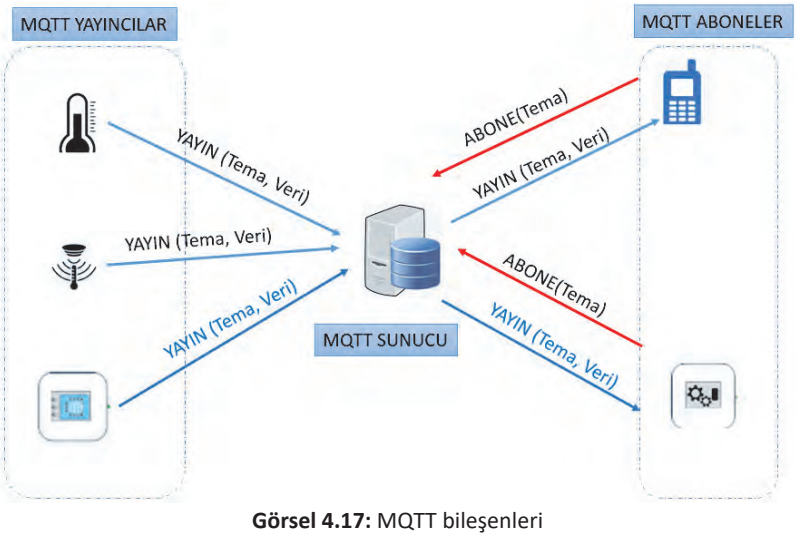
IoT uygulamalarında en yaygın kullanılan IoT'ye özgü uygulama protokolüdür. IBM tarafından açık standart olarak geliştirilmiştir. Özellikle uzak noktalarda bulunan cihazlar arasında makine-makine iletişiminde (M2M), cihazlara çok az ek yük getiren hafif bir yapıda tasarlanmıştır. HTTP protokolünün aksine istek-yanıt yapısında değil, yayın-abone mantığında geliştirilmiştir.

Görsel 4.17'de görüldüğü gibi MQTT protokolünde sunucu, abone ve yayıncı olmak üzere üç bileşen bulunur.

Yayıncı: Verinin kaynağıdır. IoT uygulamalarında bilginin elde edilmesi görevini görür. Elde ettiği bilgiyi abonelerine göndermekle yükümlüdür.

Abone: Belirli bir kaynaktaki veriyi almak isteyen istemcidir.

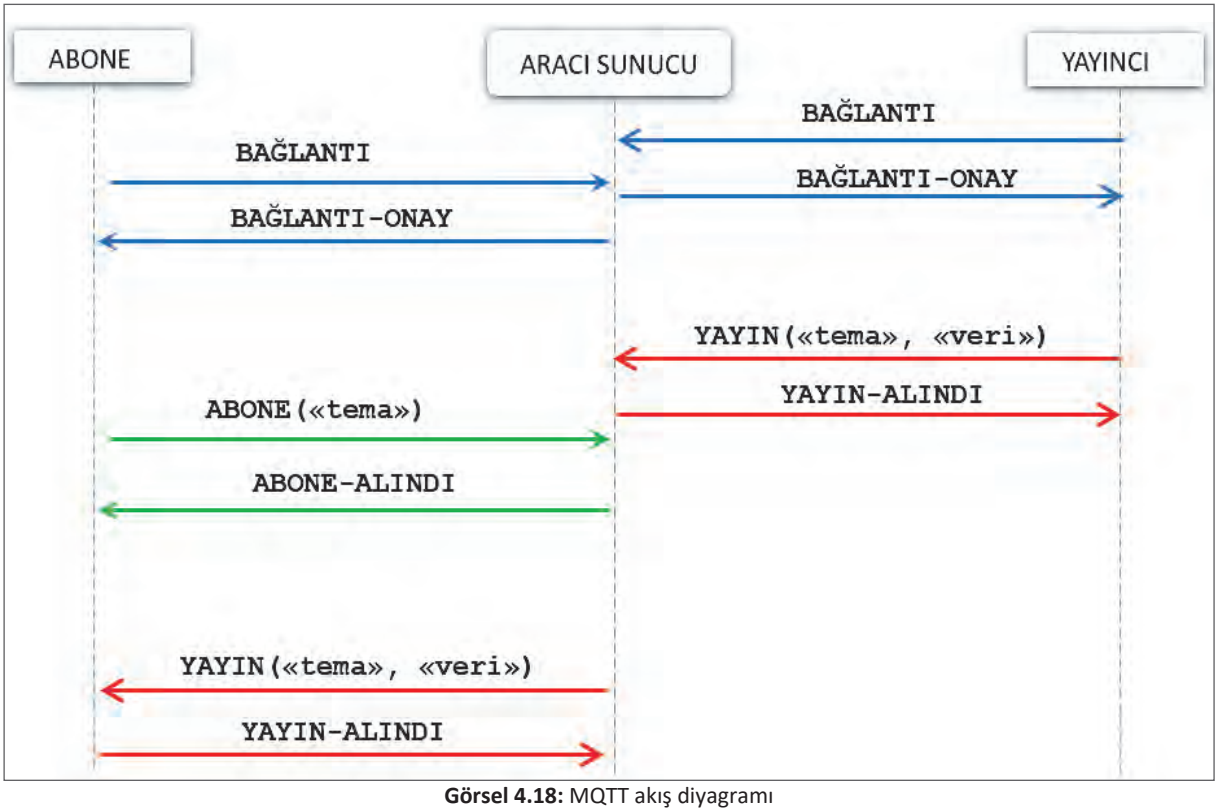
Aracı Sunucu: Yayıncı ile abone arasındaki iletişimin sağlanmasından sorumlu sunucudur. Genellikle **broker** olarak adlandırılır.



Görsel 4.17: MQTT bileşenleri

MQTT yapısında diğer bir kavram da **tema (topic)** olarak isimlendirilen hiyerarşik etiketleme sistemidir. Bu etiketleme sistemi genellikle konum ve hizmetin birlikte kullanılmasıdır. Tema bilgisinde düzeyler "/" ile birbirinden ayrılır. Örneğin akıllı ev örneğinin uygulandığı bir ortamda mutfaktaki sıcaklık bilgisi için tema **"mutfak/sıcaklık"** olarak, nem miktarı için ise **"mutfak/nem"** olarak ifade edilebilir. Tema yapısında **"+"** ve **"#"** gibi birden çok düzeyi ifade eden joker karakterler kullanılabilir. Örneğin bir evde bütün konumlardaki sıcaklığı ifade eden bir tema **"ev/#/sıcaklık"** olarak gösterilebilir.

Tema, MQTT'nin işleyişinde abonelik ve kayıt işlemlerinde kullanıldığı için önemli bir özelliktir. Bileşenlerin tema kullanarak gerçekleştirdiği iletişimler Görsel 4.18'de gösterilmiştir.



Görsel 4.18: MQTT akış diyagramı

MQTT yapısında abone, yayıncı ve sunucu arasında farklı amaçlar doğrultusunda kullanılan birçok paket türü bulunur. Abone ve yayıncılar öncelikle MQTT sunucu ile bağlantı kurmalıdır. Bunun için hem abone hem de yayıncı tarafından sunucuya **Bağlantı (Connect)** paketi gönderilir ve sunucu tarafından da **Bağlantı Onay (Connect-Ack)** ile bağlantı isteği onaylanır. Bağlantıda isteğe bağlı olarak kimlik doğrulaması da yapılabilir.

MQTT yapısında veri iletimi ve alındı bildirimi için **Yayın (Publish)** ve **Yayın-Alındı (PubRec)** paketleri kullanılır. Yayın paketi, herhangi bir tema ile etiketlenmiş ve içinde veri barındıran bir iletidir.

Abonelik işlemi için kullanılan diğer bir paket çifti de **Abone (Subscribe)** ve **Abone-Alındı (SubRec)** paketleridir. Abone paketi herhangi bir temaya abone olmak isteyen istemciler tarafından sunucuya gönderilen paket türüdür. Sunucu tarafında verilen cevap ise abonelik işleminin gerçekleştiğini **Abone-Alındı** mesajı ile doğrulayacaktır.

MQTT mesaj yapısında verilerin alıcıya ulaşıp ulaşmadığının garantisi üç farklı hizmet kalitesi **(QoS - Quality of Service)** seviyesi ile sağlanır. QoS seviyesi, aracı sunucu ile istemci arasındaki iletişimin güvenilirliği olarak tanımlanır.

MQTT mesaj yapısında desteklenen hizmet kalitesi seviyeleri şunlardır:

- QoS 0 (En fazla bir defa)
- QoS 1 (En azından bir defa)
- QoS 2 (Kesinlikle bir defa)

QoS 0: Yayıncı, mesajı aracı sunucuya en fazla bir kez yayınlar. Yayınlanan mesaj bağlantı kopması sonucu kaybolabilir ve aboneye iletilmeyebilir. Bu hizmet kalitesi seviyesinde mesajın aboneye ulaşıp ulaşmadığı kontrol edilmez. Bu nedenle en güvensiz hizmet kalitesi seviyesi olarak nitelendirilir. Mesaj, yayıncı ve aracı sunucuda depolanmaz. Mesaj gönderildikten sonra silinir. QoS 0, en düşük trafiğin olduğu hizmet kalitesi seviyesidir.

QoS 1: Yayıncı, mesajı aracı sunucuya en az bir kez yayınlar. Mesaj, birden fazla iletilebilir. Yayıncı, yayınladığı mesajın bir kopyasını alındı onayı alana kadar saklar. Yayıncı, bu onayı aldığında mesajı siler. Yayıncının, alındı onayını belli bir süre almaması hâlinde mesaj tekrar gönderilir. Bu döngü, yayıncıya alındı onayı iletilinceye kadar devam eder. Bu durumda kopya mesajlar oluşabilir. Mesajlar yayıncı ve aracı sunucuda depolanır. Bundan dolayı mesaj kaybı yaşanmaz.

QoS 2: Yayıncı, mesajı aracı sunucuya kesinlikle bir defa yayınlar. Mesaj, tam olarak bir kez iletilir. Kısaca kopya mesajlar oluşmaz. Mesajlar yayıncı ve aracı sunucuda depolanır. Bundan dolayı mesaj kaybı yaşanmaz. Bu hizmet kalitesi seviyesinde mesajın gönderildiğini ve onayın alındığını doğrulamak için bir tür el sıkışma gerçekleşir. Bu el sıkışmada, belli bir sırada iletilen dört paket kullanılır. El sıkışma tamamlandığında yayıncı ve aracı sunucu, mesajın tam olarak bir kez gönderildiğinden emin olur. Bu nedenle en güvenli hizmet kalitesi seviyesi olarak nitelendirilir. QoS 2, en fazla trafiğin olduğu hizmet kalitesi seviyesidir.

QoS 0, QoS 1 ve QoS 2 hizmet kalitesi seviyeleri hakkında yayıncı ve aracı sunucu iletişimi üzerinden bilgi verilmiştir. Aynı bilgiler aracı sunucu ve abone arasında da geçerlidir.

MQTT protokolünün genel özellikleri şunlardır:

- Asenkron çalışır.
- Temaya dayalı adresleme kullanır.
- SSL / TLS destekler.
- Hızlı haberleşme sağlar.
- TCP/IP protokol kümesine sahip her sistemde çalışır.
- En düşük seviyede kaynak kullanır.
- Aracı sunucu temelli haberleşme sağlar.
- Şifresiz bağlantılarda 1883 port numarasını kullanır.

2. UYGULAMA

Simülasyon Aracı ile MQTT Uygulaması

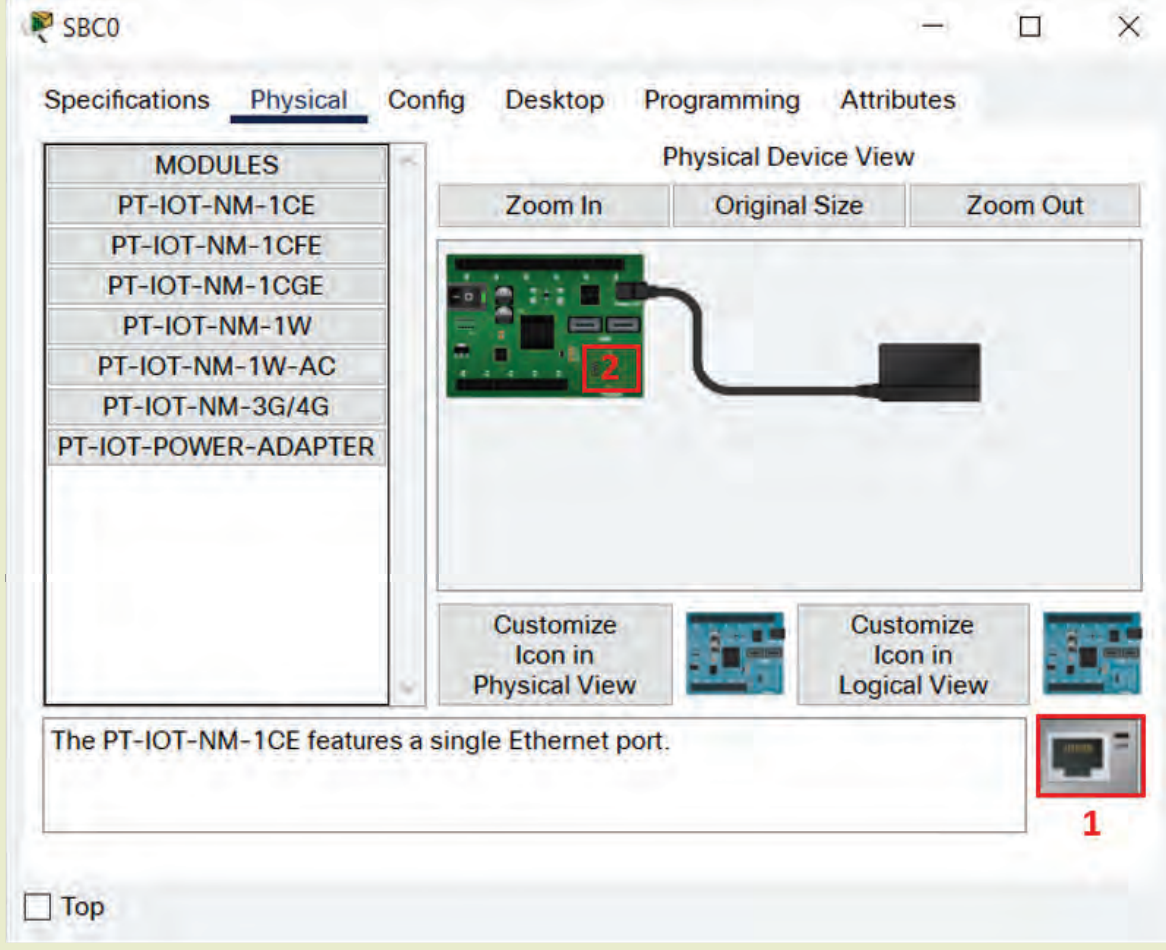
MQTT kullanarak sıcaklık bilgilerini periyodik şekilde sunucuya gönderen uygulamayı aşağıdaki adımları takip ederek hazırlayınız. Dizüstü (Laptop) bilgisayar ile sıcaklık verilerini takip ediniz.

1. Adım : Simülasyon aracını açınız.

2. Adım : Simülasyon aracına birer adet Laptop, Server, Switch, SBC Board ve Temperature Sensor ekleyiniz (Görsel 4.19).

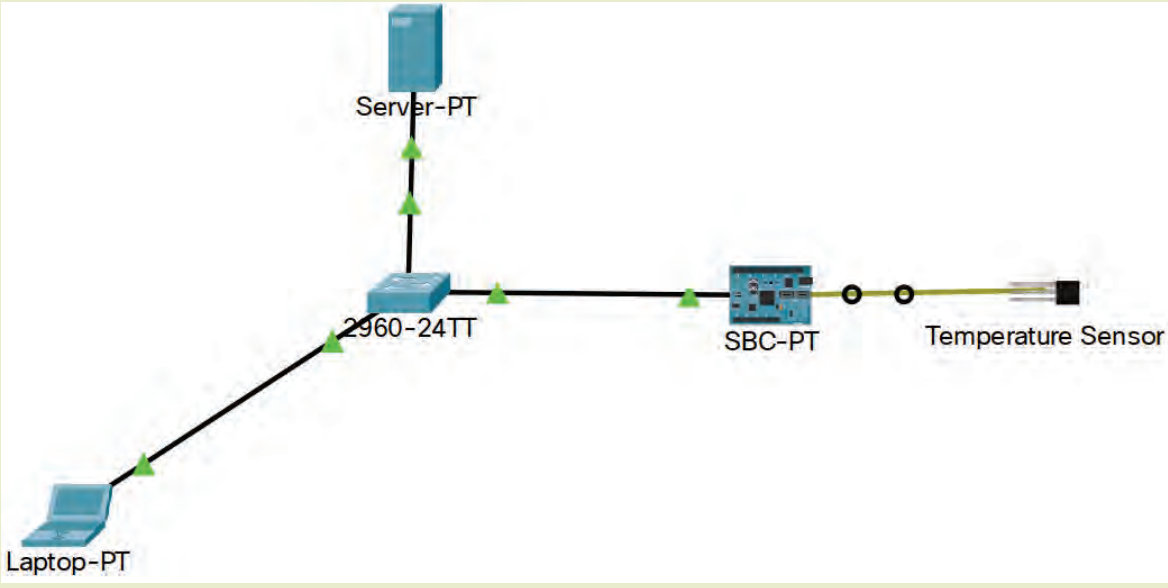
Görsel 4.19: Uygulama bileşenleri

3. Adım : SBC Board denetleyicisine tıklayınız. Gelen ekranda sağ alt köşede yer alan Ethernet portunu **(1)** denetleyicinin **(2)** numaralı yerine sürükleyip bırakınız (Görsel 4.20).



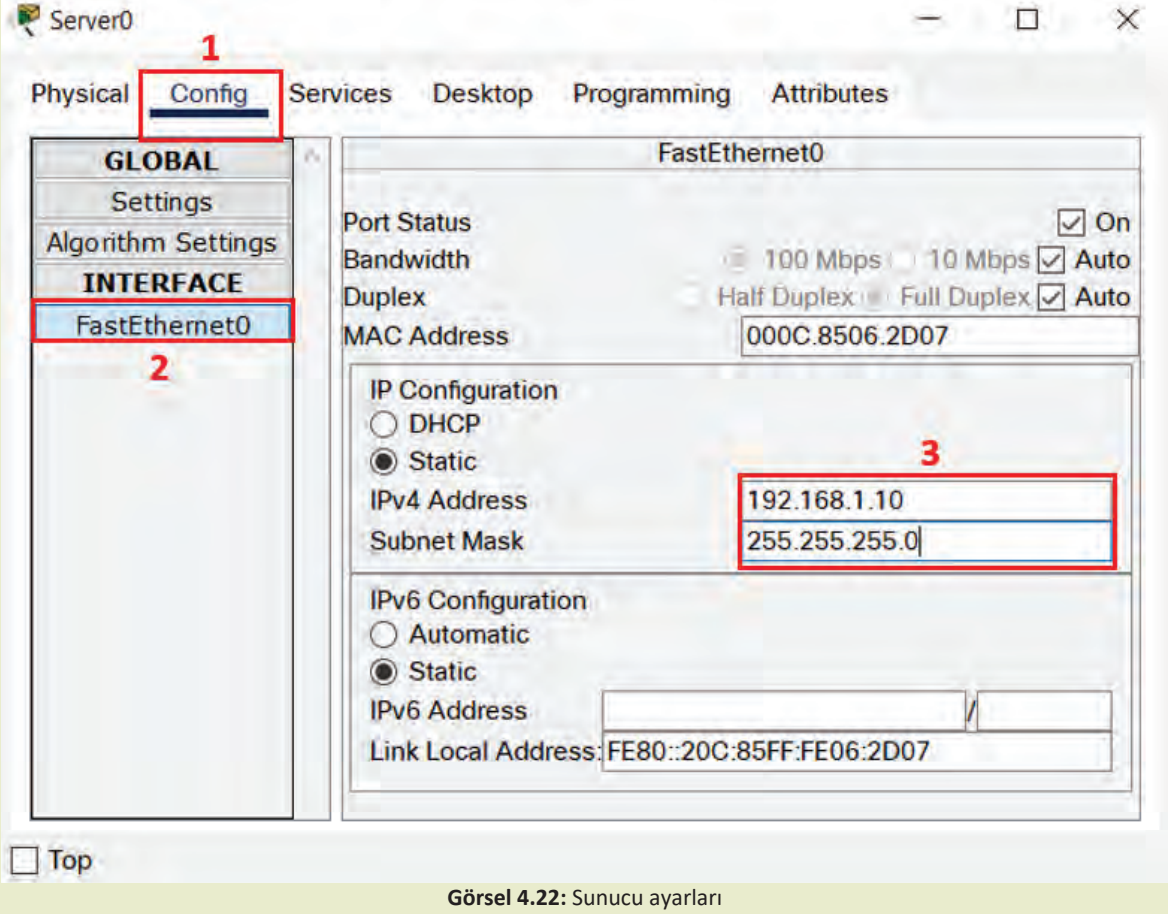
Görsel 4.20: SBC Board denetleyicisi Ethernet ayarı

4. Adım : Bağlantıları yapınız. Sıcaklık sensörünü SBC Board'a bağlarken IoT Custom Cable kullanınız. Port seçimlerini D0 olacak şekilde ayarlayınız (Görsel 4.21).



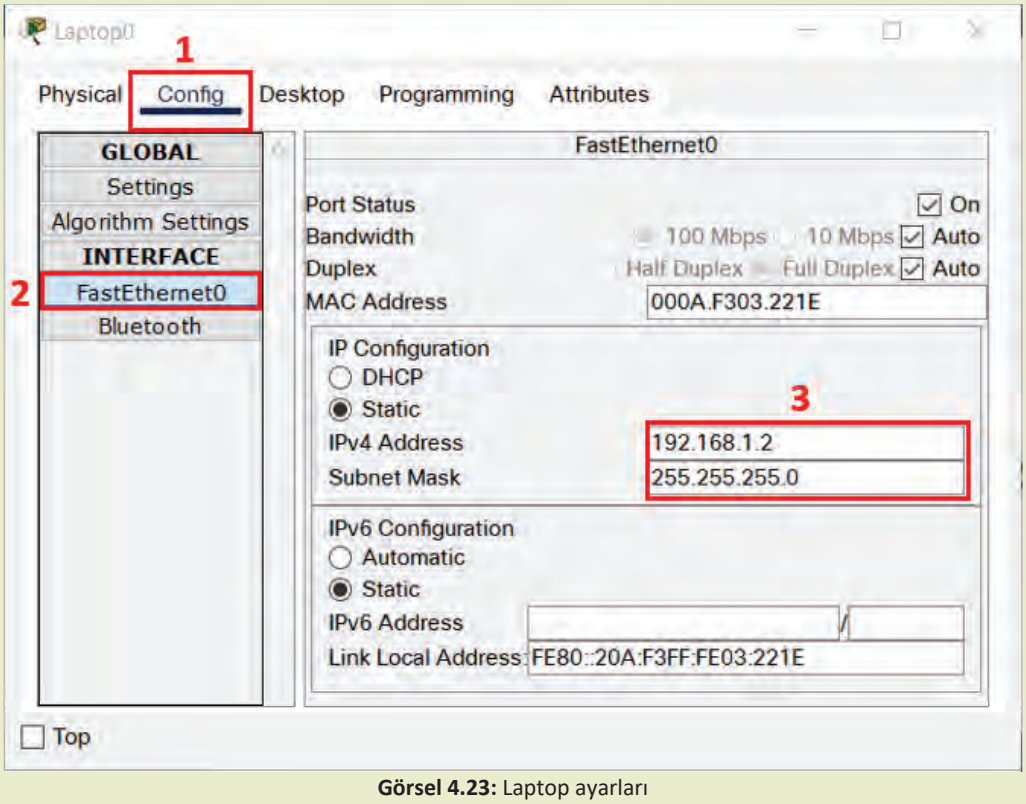
Görsel 4.21: Bağlantılar

5. Adım : Sunucuya IP adresi ve Subnet Mask atamasını yapınız (Görsel 4.22).



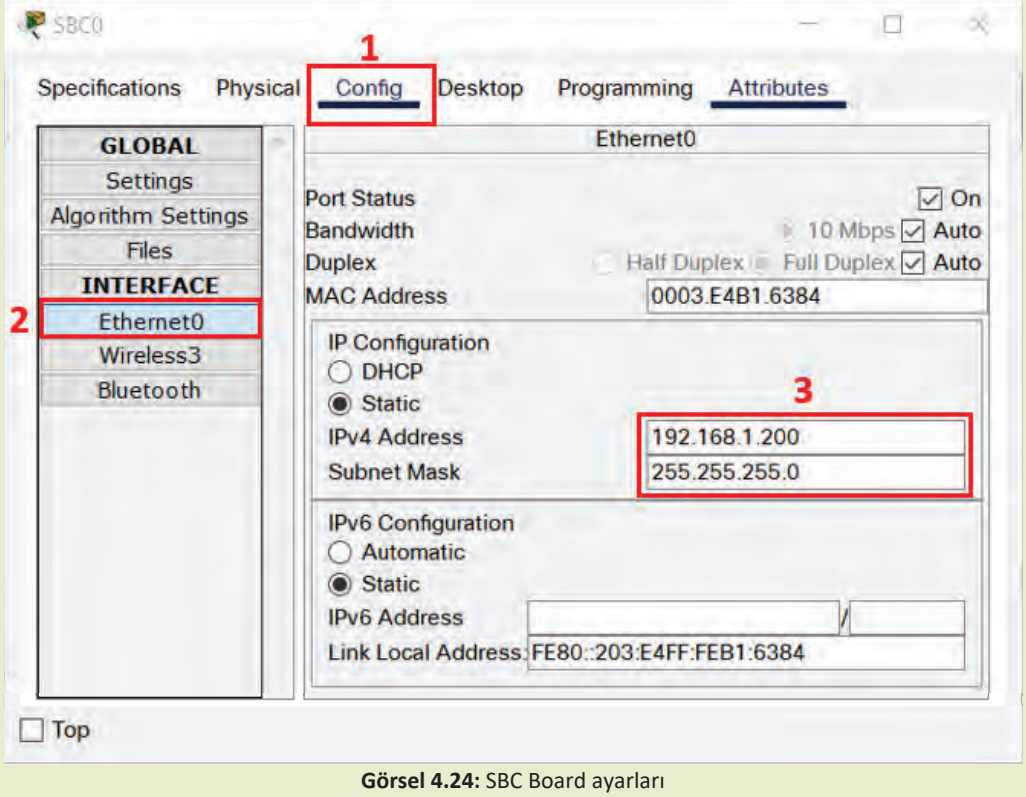
Görsel 4.22: Sunucu ayarları

6. Adım : Laptop cihazına IP adresi ve Subnet Mask atamasını yapınız (Görsel 4.23).



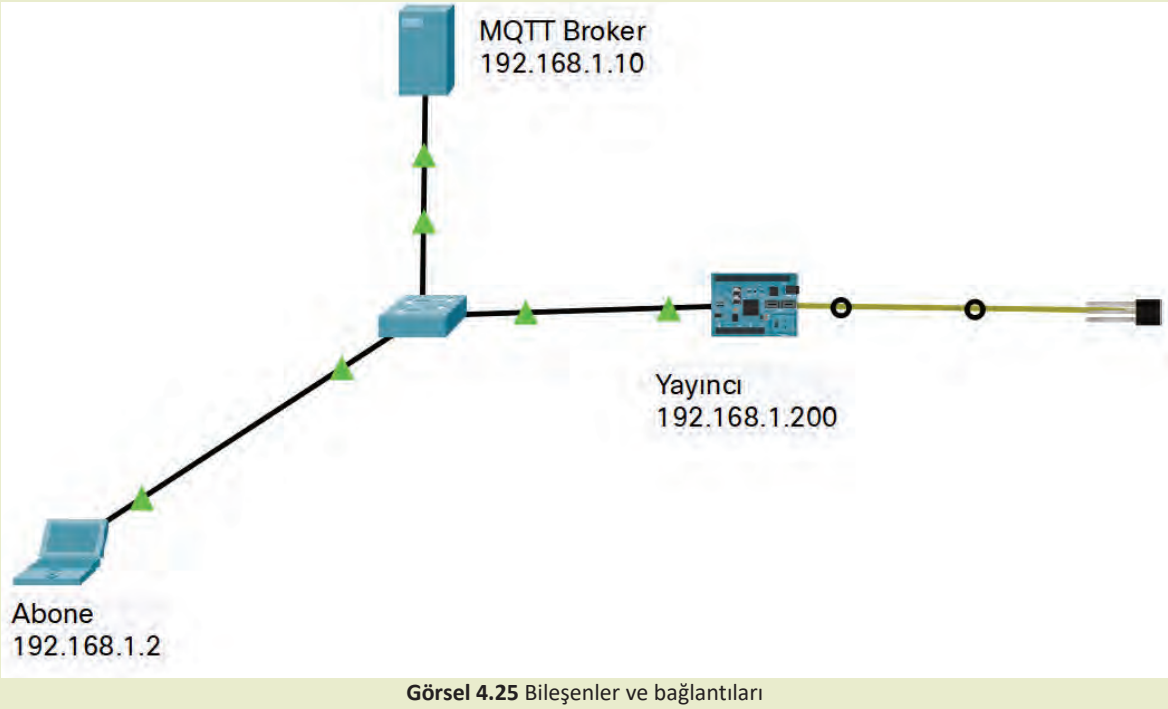
Görsel 4.23: Laptop ayarları

7. Adım : SBC Board denetleyicisine IP adresi ve Subnet Mask atamasını yapınız (Görsel 4.24).



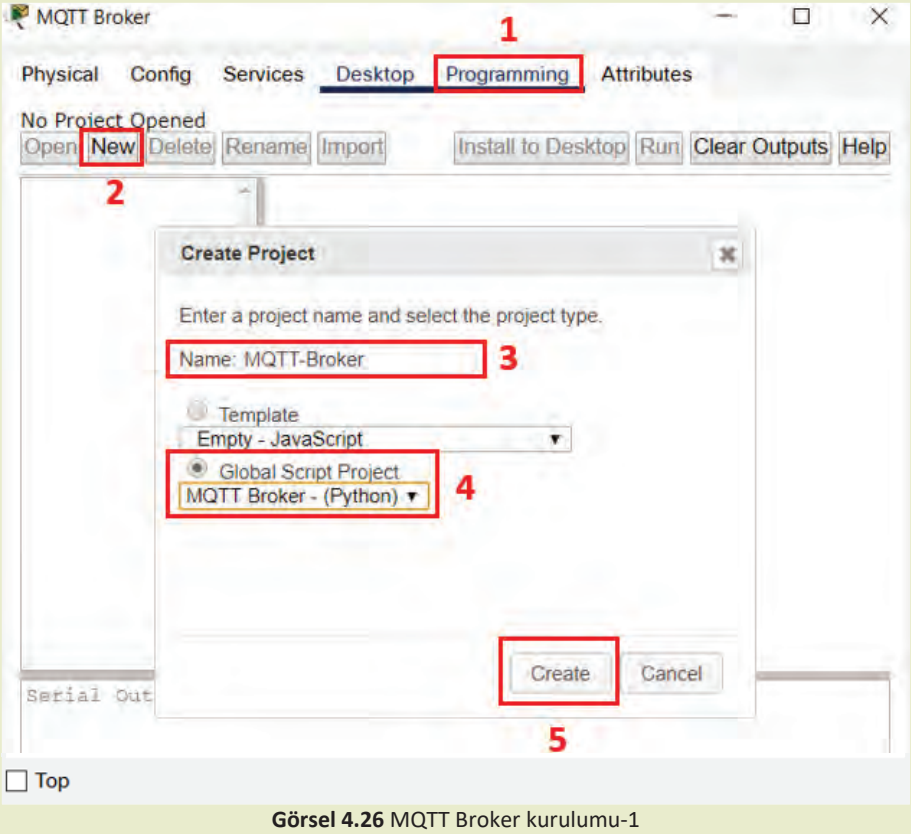
Görsel 4.24: SBC Board ayarları

8. Adım : Uygulamada bağlantıların, IP adresi ayarlarının doğru olduğundan emin olunuz (Görsel 4.25).

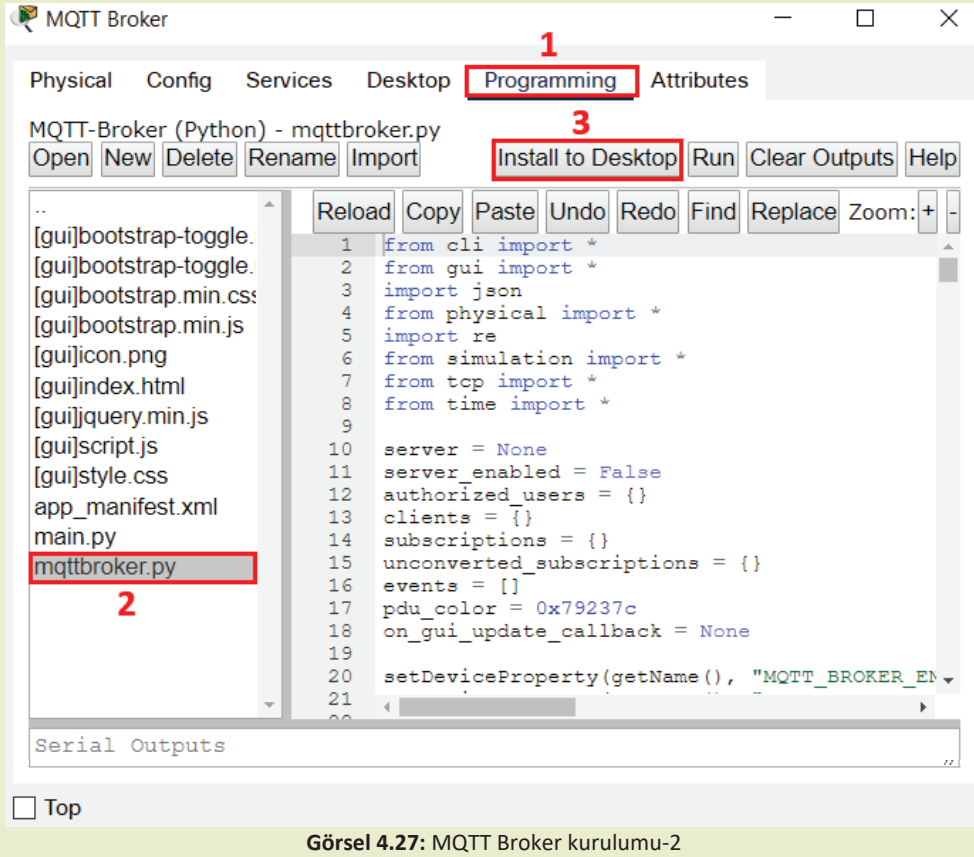


Görsel 4.25 Bileşenler ve bağlantıları

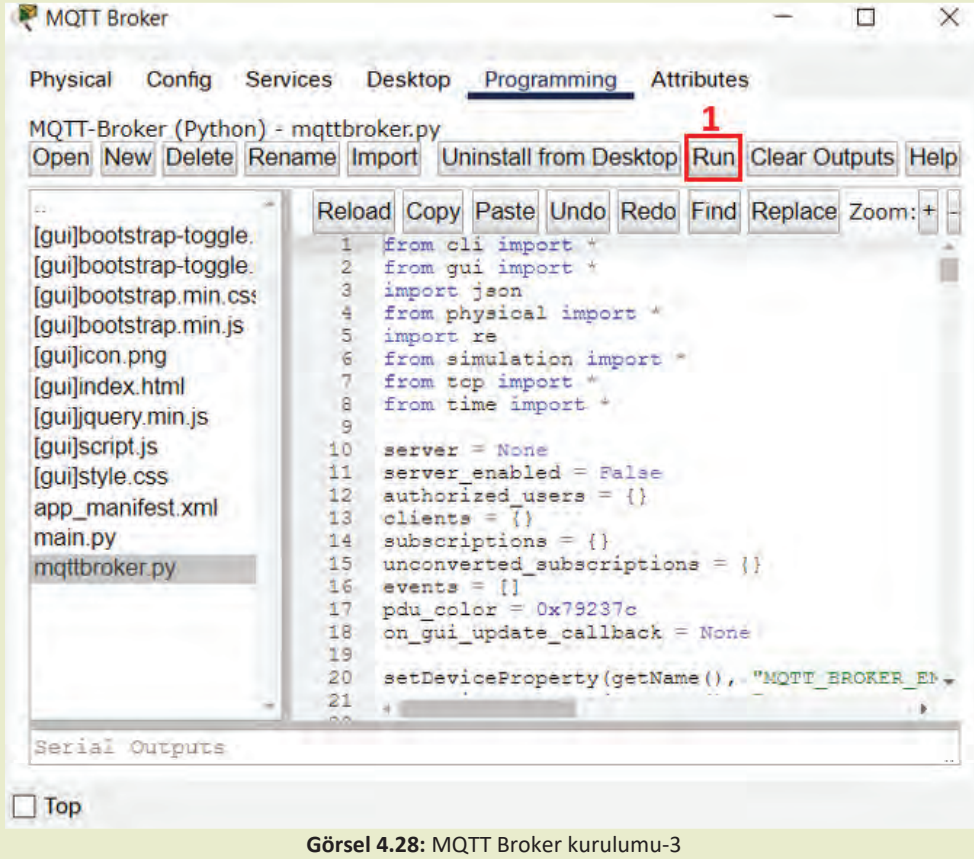
9. Adım : Sunucuya tıklayınız. MQTT Broker kurulumunu yapınız (Görsel 4.26, Görsel 4.27, Görsel 4.28, Görsel 4.29).



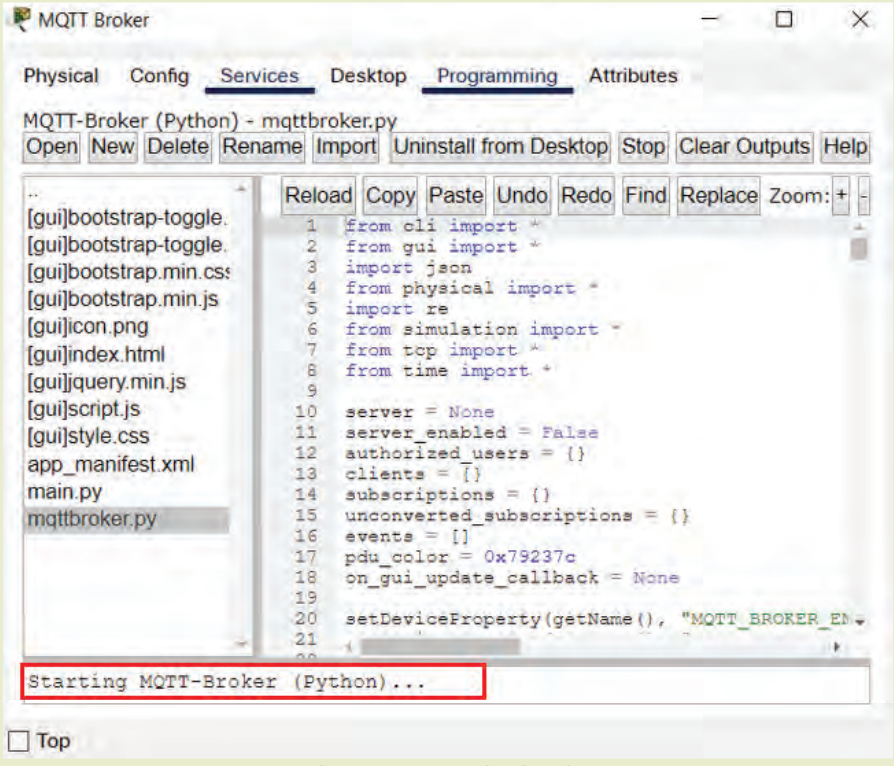
Görsel 4.26 MQTT Broker kurulumu-1



Görsel 4.27: MQTT Broker kurulumu-2

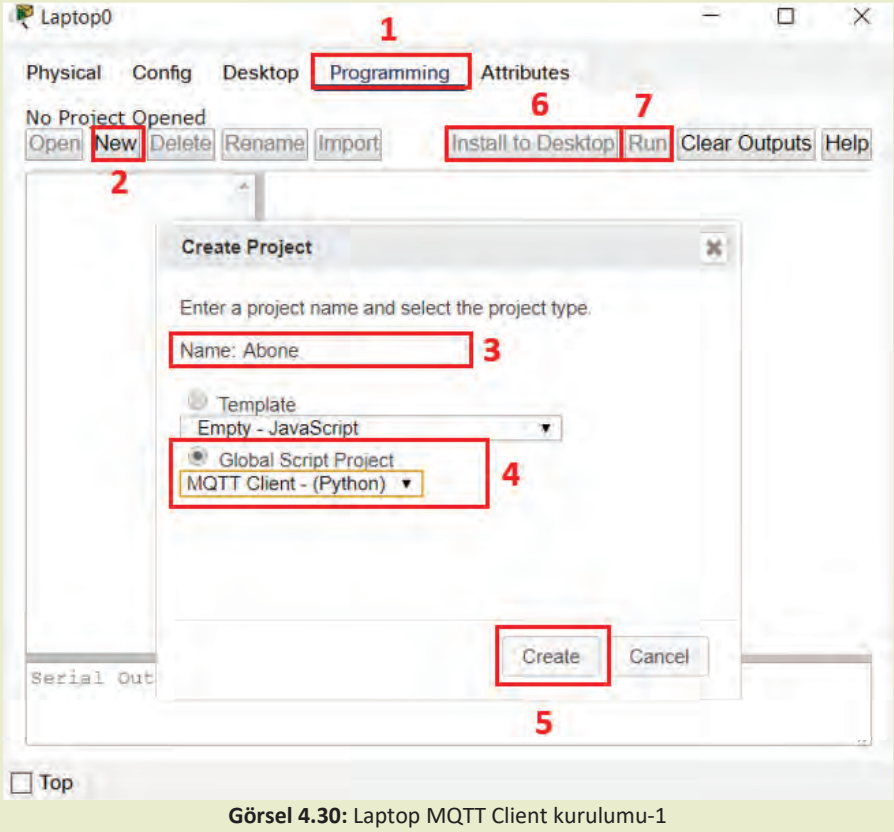


Görsel 4.28: MQTT Broker kurulumu-3

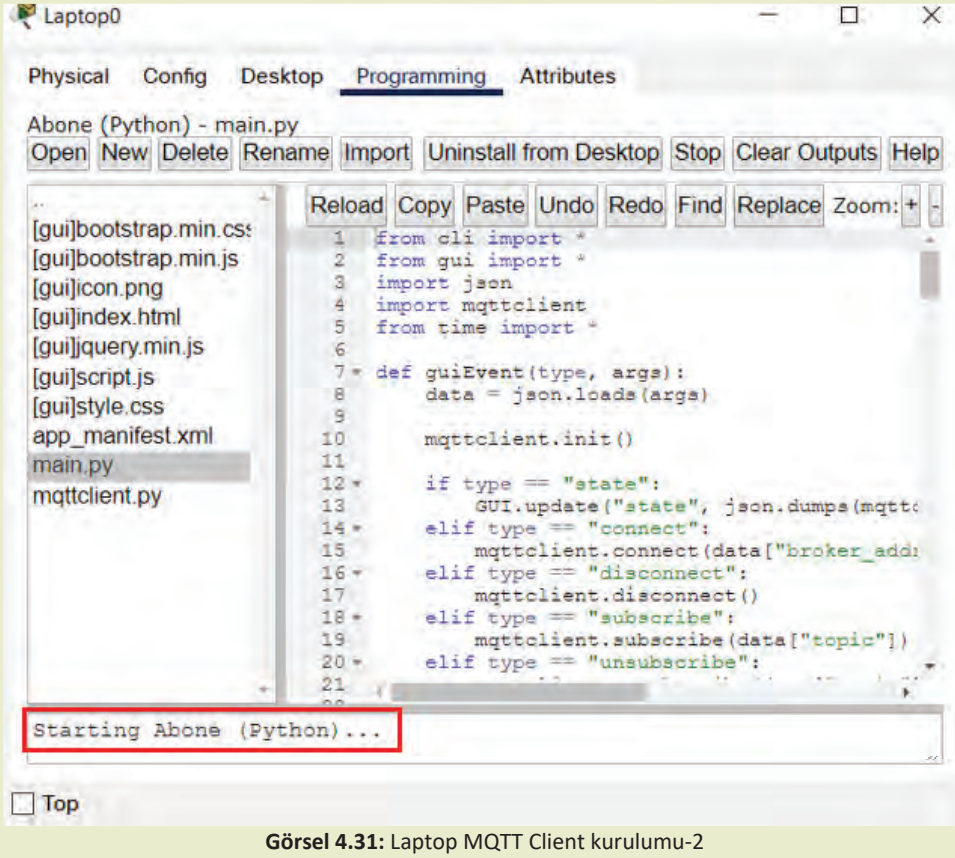


Görsel 4.29: MQTT Broker kurulumu-4

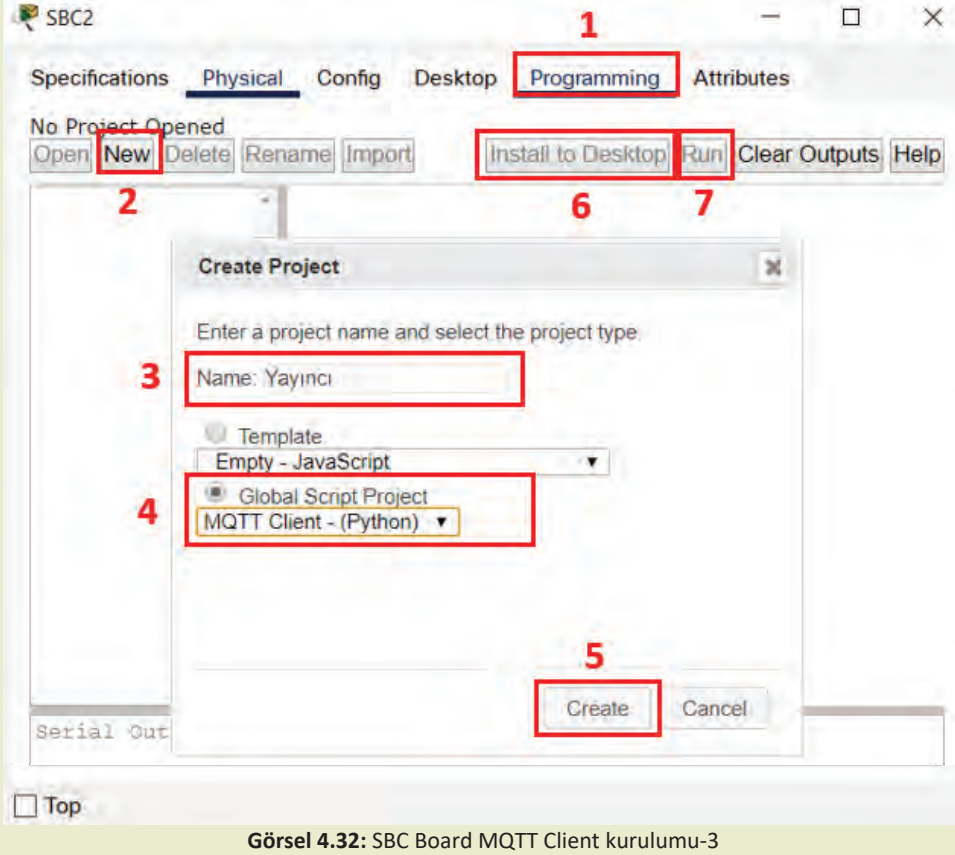
10. Adım : Laptop cihazına ve SBC Board denetleyicisine MQTT Client kurulumunu yapınız (Görsel 4.30, Görsel 4.31, Görsel 4.32, Görsel 4.33).



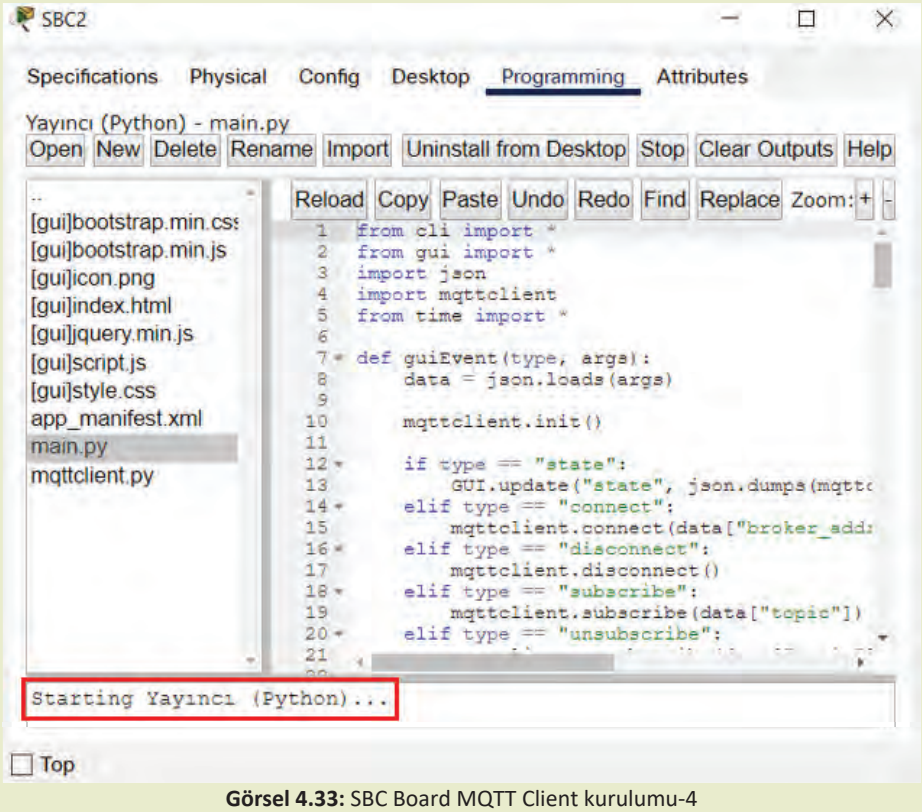
Görsel 4.30: Laptop MQTT Client kurulumu-1



Görsel 4.31: Laptop MQTT Client kurulumu-2

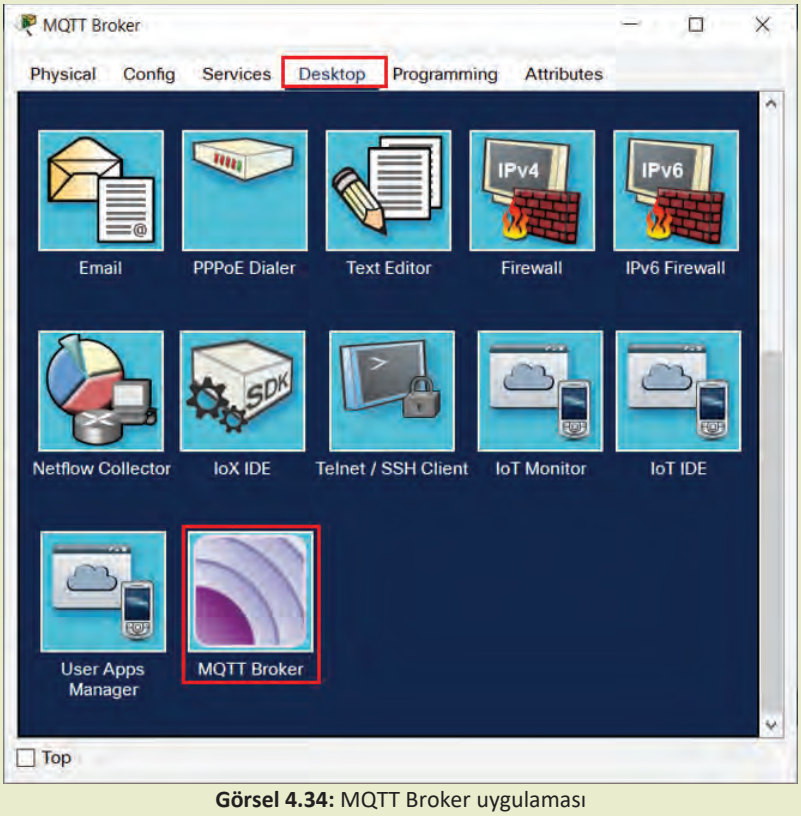


Görsel 4.32: SBC Board MQTT Client kurulumu-3



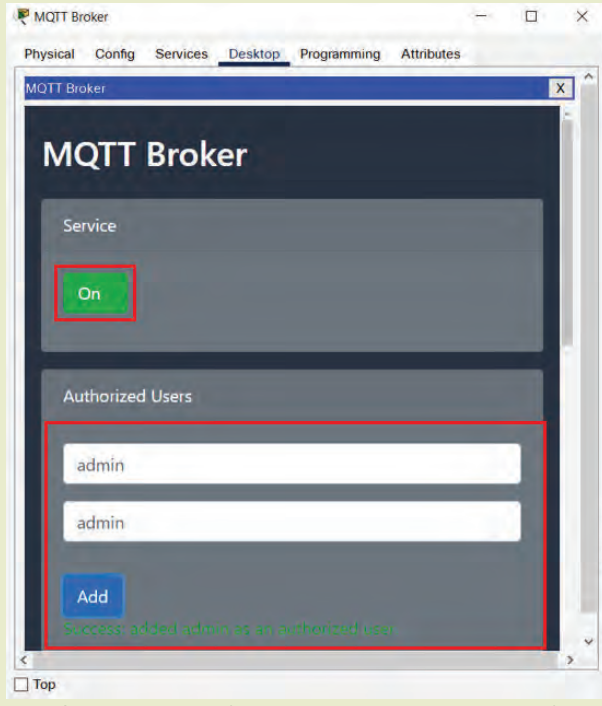
Görsel 4.33: SBC Board MQTT Client kurulumu-4

11. Adım : Sunucuya tıklayınız. Desktop sekmesinden MQTT Broker uygulamasına tıklayınız (Görsel 4.34).



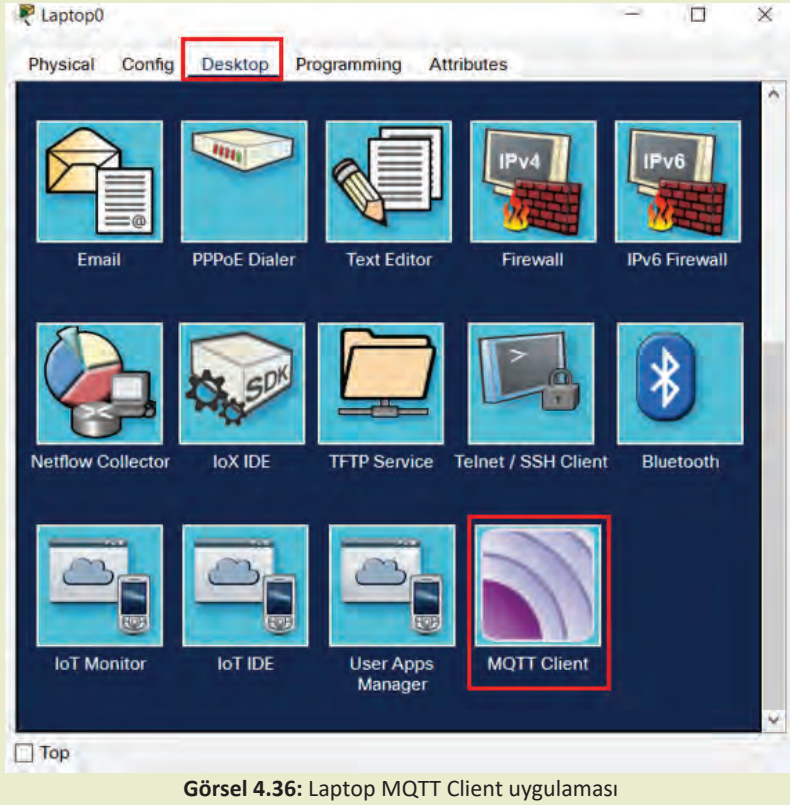
Görsel 4.34: MQTT Broker uygulaması

12. Adım : Gelen ekranda Service ayarının On olduğundan emin olunuz. Authorized Users bölümünde username ve password ekleyiniz (Görsel 4.35).



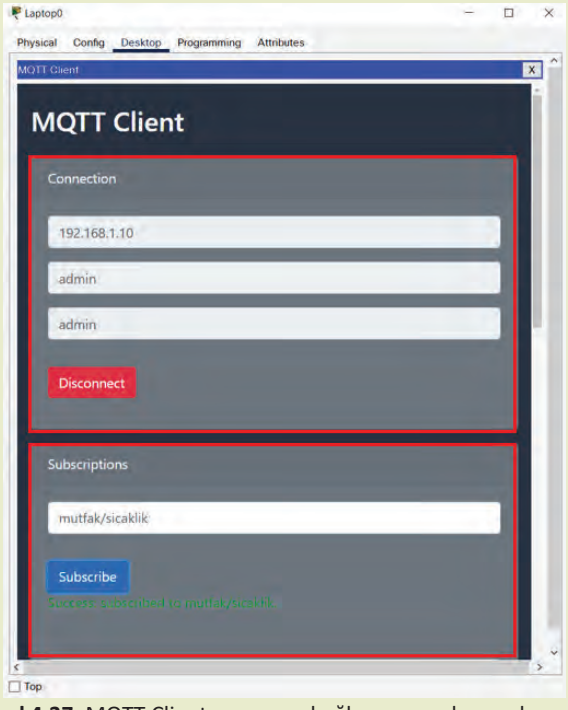
Görsel 4.35: MQTT Broker servis, username ve password ayarı

13. Adım : Laptop cihazına tıklayınız. Desktop sekmesinden MQTT Client uygulamasına tıklayınız (Görsel 4.36).



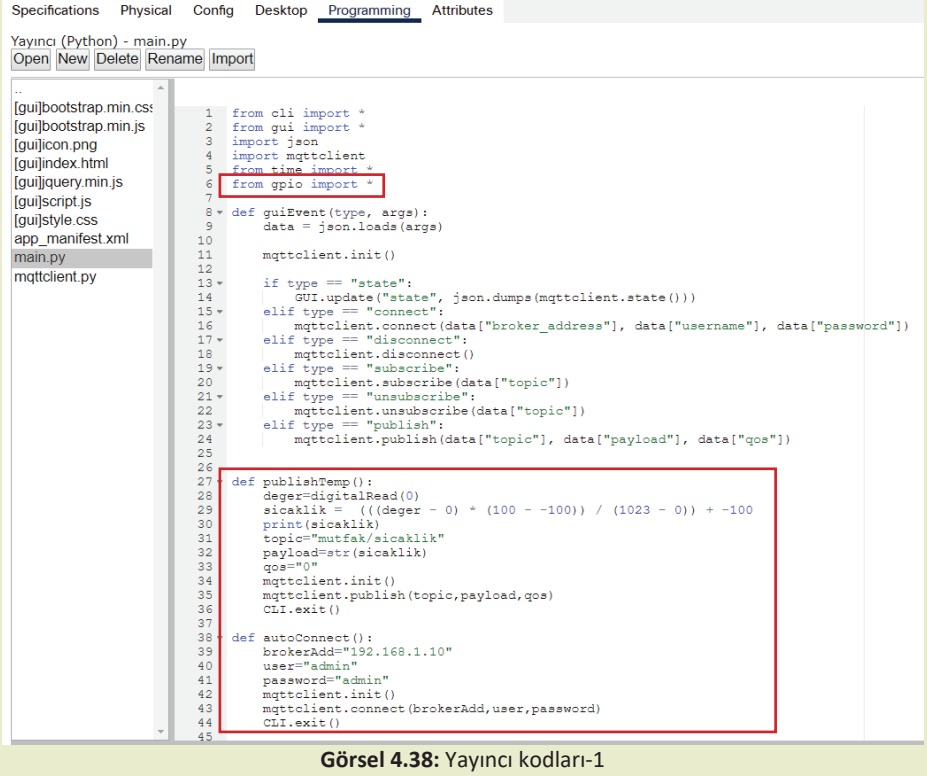
Görsel 4.36: Laptop MQTT Client uygulaması

14. Adım : Gelen ekranın Connection bölümünde MQTT Sunucunun IP adresini, username ve password ayarlarını giriniz. Connect butonuna basınız (Görsel 4.37).

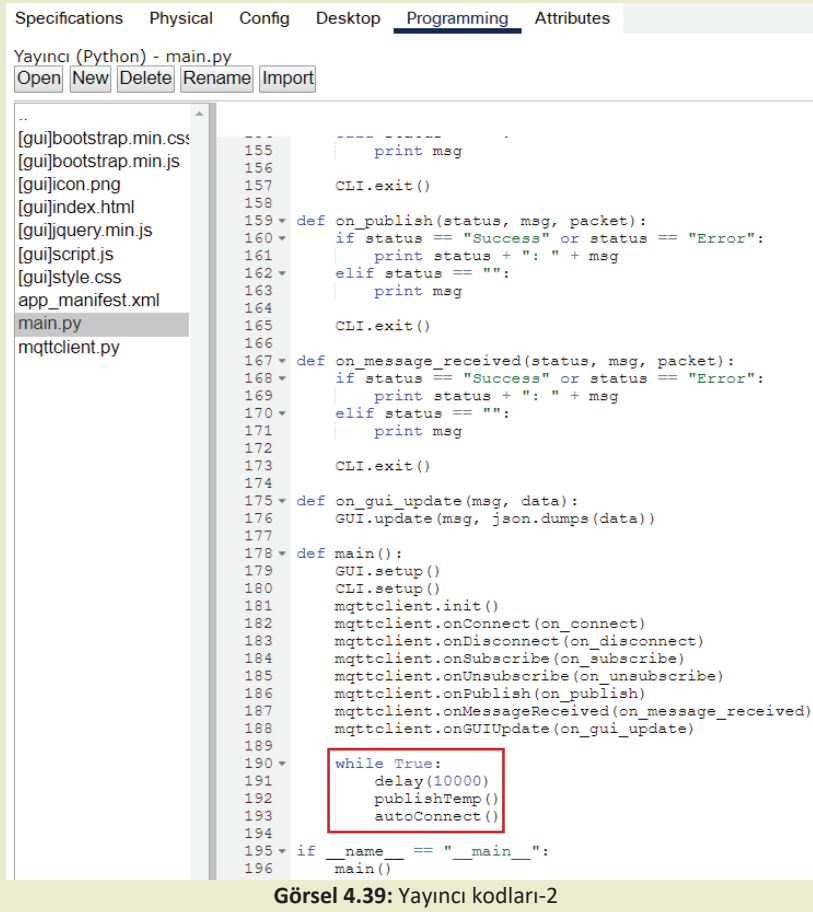


Görsel 4.37: MQTT Client sunucuya bağlanma ve abone olma ayarı

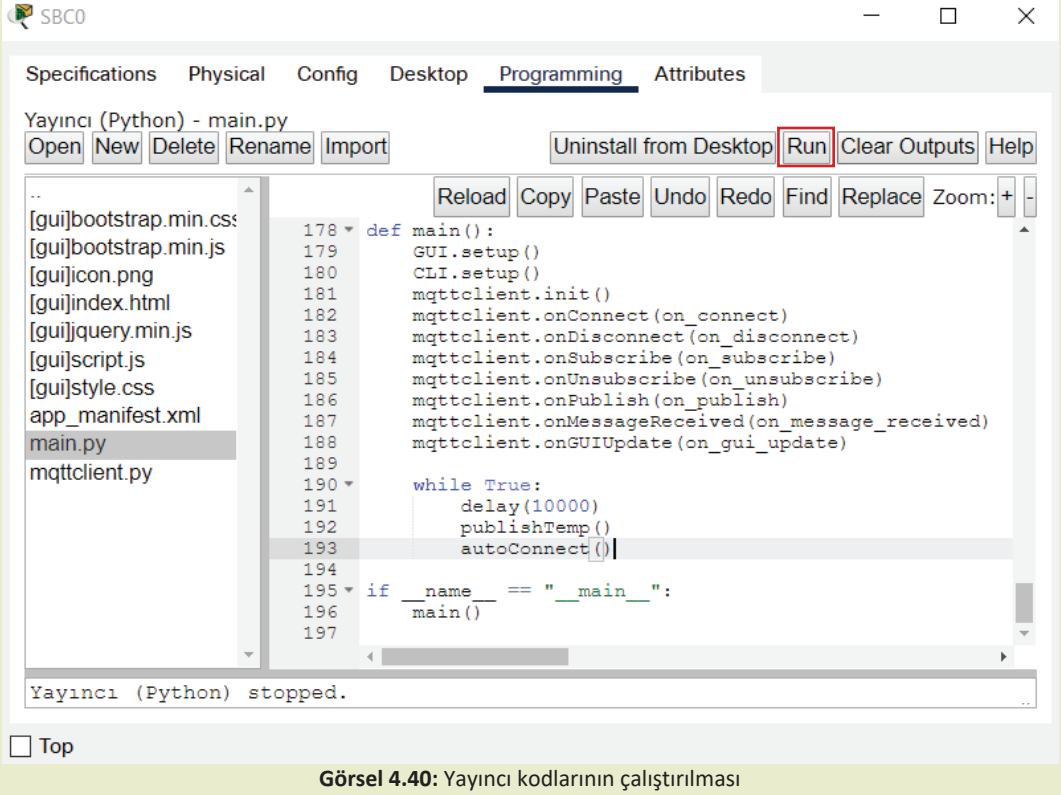
15. Adım : SBC Board denetleyicisine tıklayınız. Programming sekmesinden main.py dosyasını açınız. Gerekli kod eklemelerini yapınız. Run butonuna basarak programı çalıştırınız (Görsel 4.38, Görsel 4.39, Görsel 4.40).



Görsel 4.38: Yayıncı kodları-1

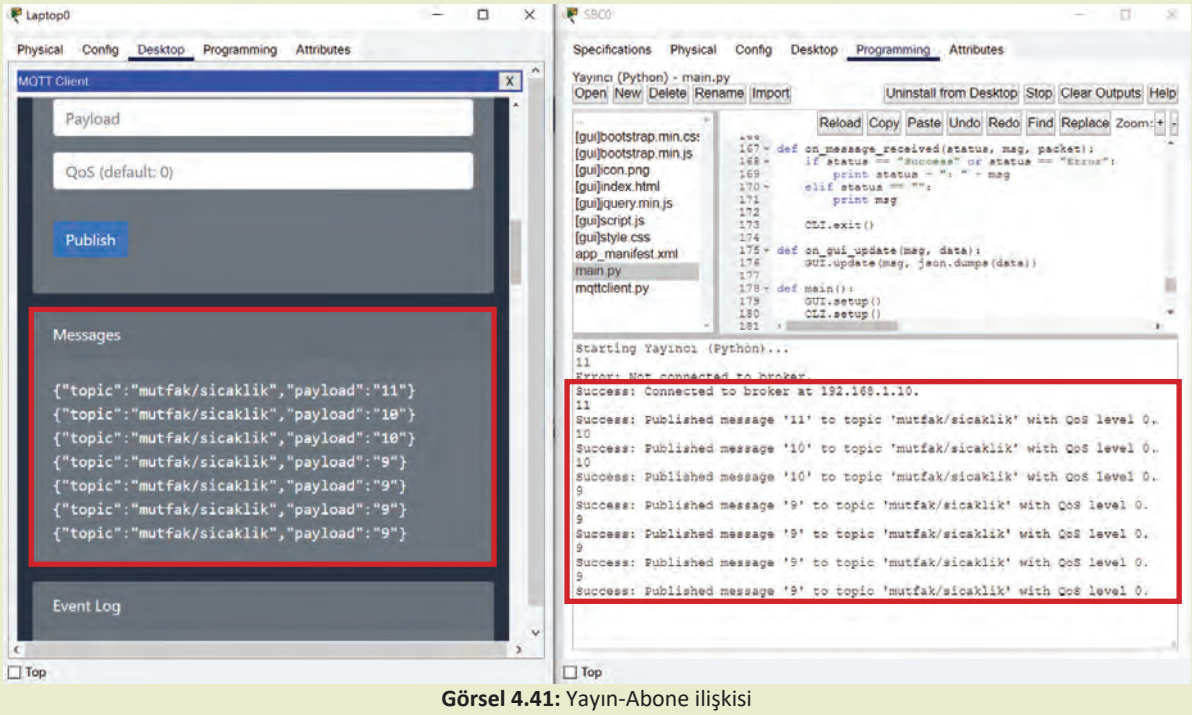


Görsel 4.39: Yayıncı kodları-2



Görsel 4.40: Yayıncı kodlarının çalıştırılması

16. Adım : Program çalıştıktan sonra sıcaklık değerlerini Laptop cihazındaki MQTT Client ekranından takip ediniz (Görsel 4.41).



Görsel 4.41: Yayın-Abone ilişkisi

4.5.3.2. Constrained Application Protocol (CoAP)

CoAP, HTTP protokolü üzerinde çalışan REST temelli mesaj transferi sağlayan bir protokoldür. CoAP, kısıtlı kaynaklara (bellek miktarı, işlem kapasitesi, güç vb.) sahip cihazlar arasında ve kısıtlı bant genişliğine sahip ağlarda çalışmak üzere tasarlanan, istek / yanıt iletişim modelini kullanan bir web haberleşme protokolüdür. Bu protokol, UDP portları üzerinde asenkron olarak çalışır.

Makine-makine iletişiminde (M2M) ve IoT uygulamalarında kullanılmak üzere tasarlanan bu protokolde istemci ve sunucu rolleri bulunur. CoAP çalıştıran her bir IoT nesne, CoAP sunucusu olarak görev yapar. CoAP istemciler, sunucudaki hizmete erişmek için GET, POST komutlarıyla istek paketleri gönderir. Sunucu, gelen mesajlara yanıt paketleri göndererek karşılık verir (Görsel 4.42).

Görsel 4.42: CoAP haberleşmesi

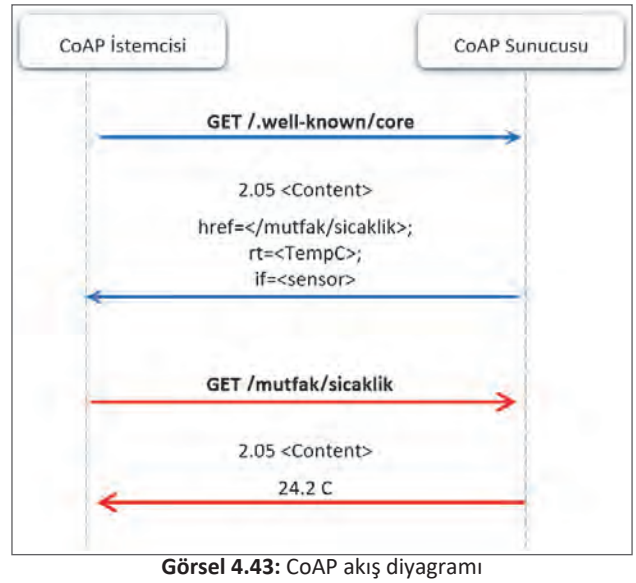
CoAP çalıştırılan bir yapıda URI adresleri kullanılır. Örnek bir URI adresi aşağıda tanımlanmıştır.

GET coap://192.168.1.100:5683/mutfak/sicaklik

CoAP çalışan istemcinin web tarayıcısında örnek URI adresi ile CoAP sunucudan GET isteğinde bulunulur. CoAP yanıtı ile alınan mutfak sıcaklık verisi web tarayıcısında görüntülenir.

CoAP protokolünün gerçekleştirdiği iletişimler Görsel 4.43'te detaylı şekilde gösterilmiştir.

Görsel 4.43'te istemci tarafından servislerin keşfi amacıyla sunucuya "/.well-known/core" isteği gönderilir. Sunucu da sıcaklık verisini belirten "/mutfak/sicaklik" parametresi ile yanıt verir. Bu yanıt sayesinde istemci, sunucudaki servislerin neler olduğunu öğrenir. İstemci, GET komutu ile istediği servise ait verilere yani sensör değerlerine erişebilir.



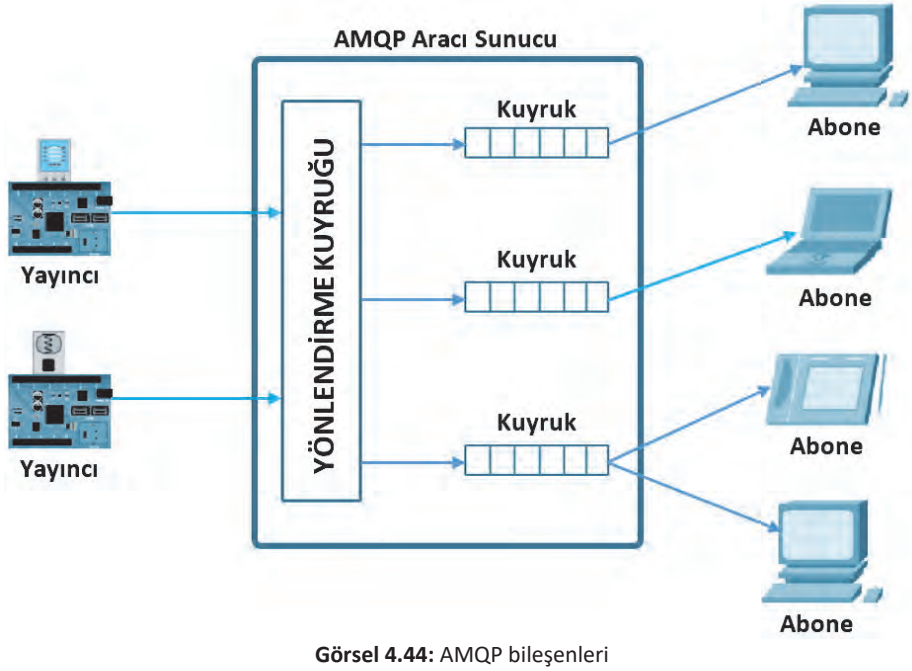
Görsel 4.43: CoAP akış diyagramı

4.5.3.3. Advanced Message Queuing Protocol (AMQP)

AMQP, farklı platformlardan gelen uygulama verilerinin birlikte çalışabilmesini sağlamak amacıyla geliştirilmiş, mesajlaşmaya dayalı bir protokoldür. TCP kullanan bu protokol, mesajları kaybetmemeye odaklı geliştirildiğinden yavaştır.

AMQP çalışan bir sistemde, farklı programlama dilleri ile farklı platformlara yönelik yazılmış olan uygulamalar birbirleri ile haberleşebilir. Bu haberleşme, kullanılan teknolojik altyapılardan tamamen bağımsızdır.

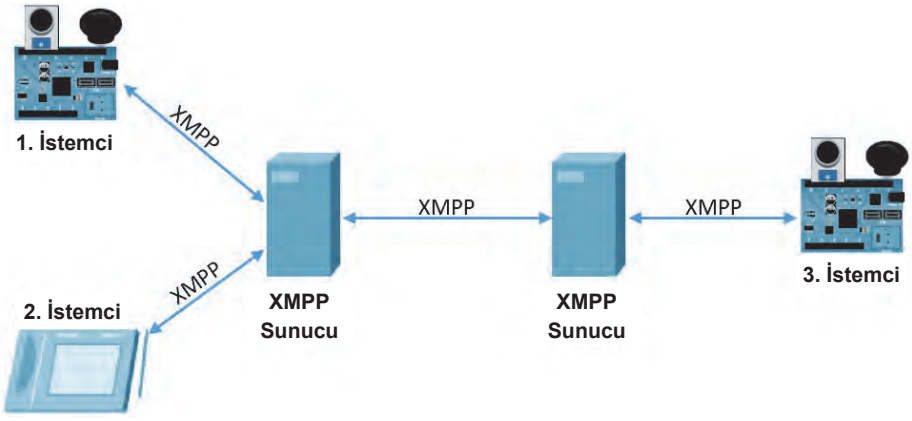
AMQP protokolünde kuyruk yapısına sahip aracı sunucu, abone ve yayıncı olmak üzere üç bileşen bulunur (Görsel 4.44).



Görsel 4.44: AMQP bileşenleri

4.5.3.4. Extensible Messaging and Presence Protocol (XMPP)

XMPP, basit ve esnek yapısıyla XML biçimini kullanan anlık mesajlaşma protokolüdür. Bu protokol, TCP tabanlı olup sunucu / istemci mantığı ile çalışır. IoT nesnelerinin haberleşmesinde kullanılır (Görsel 4.45).



Görsel 4.45: XMPP haberleşmesi

Görsel 4.45'teki sistemde XMPP haberleşmesi gerçekleşir. Bu haberleşmede 1.İstemci'de algılanan ses, 3. İstemci'de anlık olarak dinlenebilir. 2. İstemci'den gönderilen bir mesaj, 3. İstemci'den sesli mesaj olarak alınabilir. 3.İstemci'nin bulunduğu evde herhangi bir ses algılandığında ev sahibinin tableti olan 2. İstemci'ye "Evde birileri var" yazılı mesaj gönderilebilir.

SIRA SİZDE

MQTT, CoAP ve AMQP Uygulama protokollerini Mesaj Boyutu, Güvenilirlik ve Mesajlaşma Modu yönünden karşılaştırmasını ait bir sunum hazırlayınız. Hazırladığınız sunuyu sınıfta anlatınız.

4.6. SİS VE BULUT BİLİŞİM

Her geçen gün farklı türden IoT cihazının artışı, toplanan verilerin de hızla çoğalmasına yol açmıştır. Toplanan veriler, sensör verileri yanında IoT cihazla ilişkili zaman ve konum verilerini de içerir. Tüm bu verilerin IoT ekosistemindeki uygulamalar arasında paylaşılması kaçınılmazdır. Bu durum beraberinde daha fazla birlikte çalışabilirlik gerektirir. Bu gereklilik, yeni veri yönetimi sorunlarını ortaya çıkarır. Bulut ve sis bilişim, IoT ekosistemindeki veri yönetimi sorunlarıyla başa çıkmak için kullanılan en önemli model yaklaşımlardır.

4.6.1. Bulut Bilişim (Cloud Computing) Modeli

Bulut bilişim, hızlı ölçeklenebilir donanım ve yazılım altyapısı olan, sahip olduğu bilgi işlem kaynaklarını ağ veya internet üzerinden kullanıcılar arasında paylaştıran bilişim hizmetlerinin genel adıdır (Görsel 4.46). Bulut bilişimin en önemli amacı, donanım ve yazılım kaynaklarını minimum yönetim çabasıyla hızlı bir şekilde kullanılabilir hâle getirmektir.

Görsel 4.46: Bulut bilişim

Bulut bilişim, birbirine bağlı güçlü sunucular ağıdır. Birçok kurum, depolama ve veri işleme için ek yük getiren sunucu sistemleri kurmak yerine bulut hizmetleri kiralamayı tercih eder. Bulut bilişimde bazı hizmetler kısıtlı şekilde ücretsiz kullanıma sunulabilir. Bu hizmetlere daha fazla ihtiyaç duyulduğunda **kullandıkça öde** modeli kullanılır. Bu model ile hizmet kiralama yapan bir kullanıcı, hizmete

kullandığı kadarıyla ödeme yapar. Kullandıkça öde modeli; kullanıcıya esnek, güvenli ve düşük maliyetli bir kullanım sunar.

Bulut sağlayıcıları; donanım ve yazılım kaynaklarını barındırmak, bulut hizmetleri sunmak için veri merkezlerini (data center) kullanır. Bu sağlayıcılar, bulut hizmetlerinin ve bilgi işlem kaynaklarının erişilebilirliğini sağlamak için birkaç uzak veri merkezinde yer tutar.

Bulut bilişim modelinin avantajlarından bazıları şunlardır:

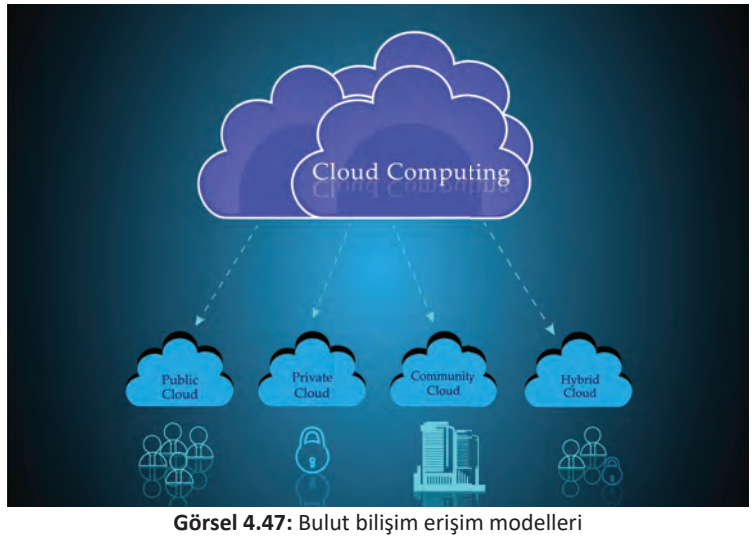
- Kullanıcılar, verilerine konum ve zamandan bağımsız şekilde erişir.
- Ekipman, enerji ve fiziksel tesis gereksinimleri azalır.
- Personel eğitimi ihtiyacı azalır.
- Ekipman bakım, onarım ve yönetim ihtiyacı azalır.
- Gerekli donanım ve yazılım maliyetini düşürür.
- Altyapı karmaşasını ortadan kaldırır.
- API mimarisi üzerinden kullanım kolaylığı sunar.
- Daha fazla depolama alanı ve hızlı veri transferi sağlar.
- Veri işleme için esnek platformlar sunar.
- Platform bağımlılığı azalır.
- Veri yedekliliği ve paylaşım izni ayarları ile güvenliği sağlar.
- BT ekiplerinin performans ve verimliliği artar.
- Kurumun BT operasyonlarının yürütülmesinde kolaylık sağlar.

Bulut bilişim, IoT ekosisteminin önemli bir bileşenidir. Cihazların haberleşmesi, sensör verilerinin depolanması ve işlenmesi için bulut tabanlı hizmetlerin kullanılması yaklaşımı IoT sistemlerin yaygınlaşmasıyla standart hâline gelmiştir.

4.6.1.1. Bulut Bilişim Erişim Modelleri

Bulut bilişim erişim modelleri dörde ayrılır (Görsel 4.47). Bu erişim modelleri şunlardır:

- Genel Bulut (Public Cloud)
- Özel Bulut (Private Cloud)
- Hibrit Bulut (Hybrid Cloud)
- Topluluk Bulut (Community Cloud)



Görsel 4.47: Bulut bilişim erişim modelleri

Genel Bulut (Public Cloud): Hizmet sağlayıcının depolama, uygulama gibi kaynakları internet üzerinden son kullanıcılara sunduğu bulut hizmetidir. Bu bulut hizmeti, kullandıkça öde modeliyle ya da ücretsiz olarak sunulur. Küçük ve orta ölçekli kurumlar tarafından tercih edilir. Google Drive, Yandex Disk bulut ortamları örnek olarak verilebilir.

Özel Bulut (Private Cloud): Kurum içindeki verilerin ve uygulamaların gizliliğinin sağlandığı, bilgi işlem kay-

naklarının sadece kurum kullanıcılarına özel olarak sunulduğu bulut hizmetidir. Sunulan kaynaklara kurum dışından erişim sağlanamaz. Büyük ölçekli kurumlar tarafından tercih edilir.

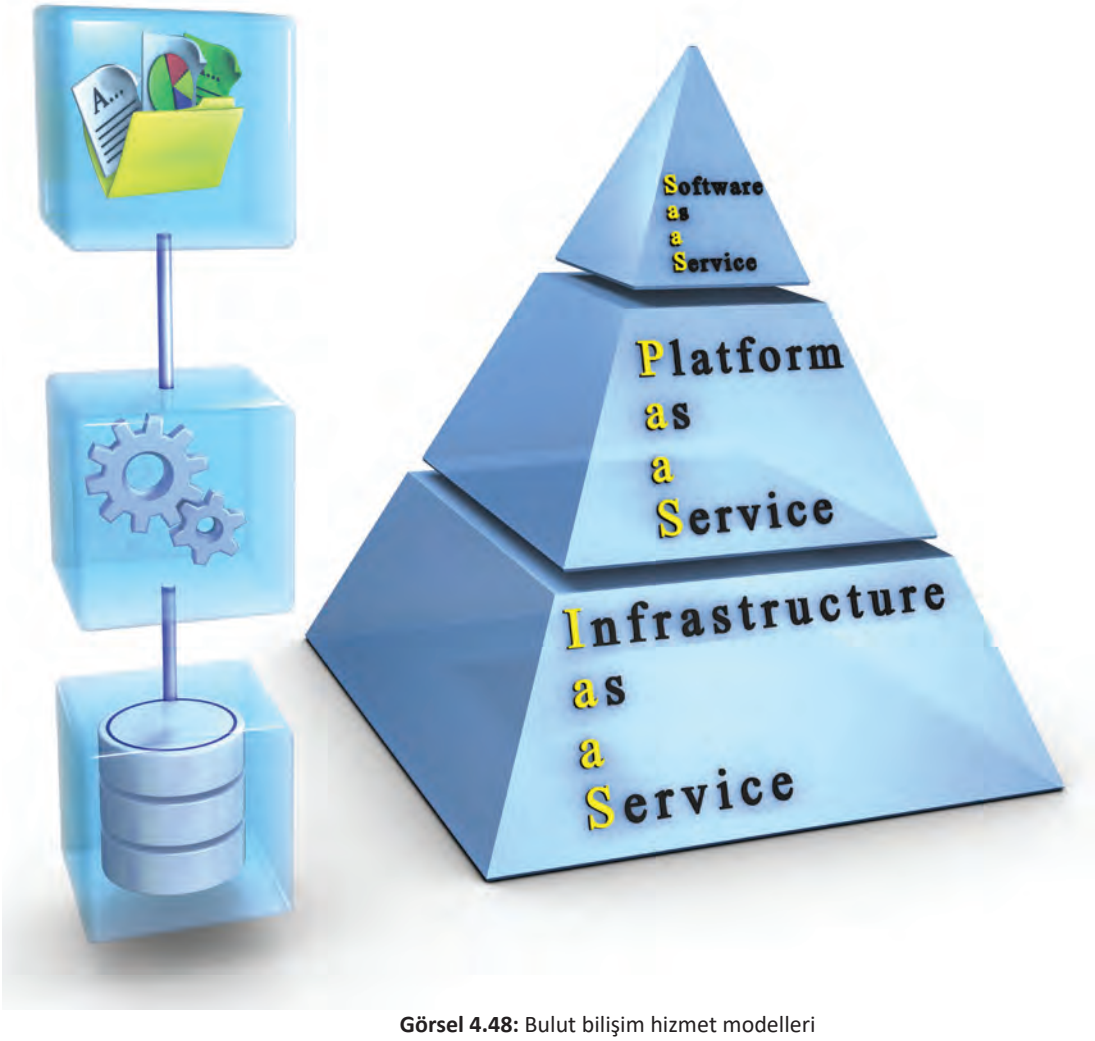
Hibrit Bulut (Hybrid Cloud): Özel ve genel bulut kaynaklarının bütünleştirildiği bulut hizmetidir. Bu bulut hizmetinde, kurumun kritik ve hassas bilgileri özel bulutta saklanırken, son kullanıcıya açık olan hizmetlere genel bulut üzerinden ulaşılır.

Topluluk Bulut (Community Cloud): Hizmetin birkaç kurumla ortak kullanıldığı ve bilgi işlem kaynaklarının belirli topluluğa sunulduğu bulut hizmetidir.

4.6.1.2. Bulut Bilişim Hizmet Modelleri

Bulut bilişim hizmet modelleri üçe ayrılır (Görsel 4.48). Bu hizmet modelleri şunlardır:

- Hizmet Olarak Altyapı (Infrastructure as a Service)
- Hizmet Olarak Platform (Platform as a Service)
- Hizmet Olarak Yazılım (Software as a Service)



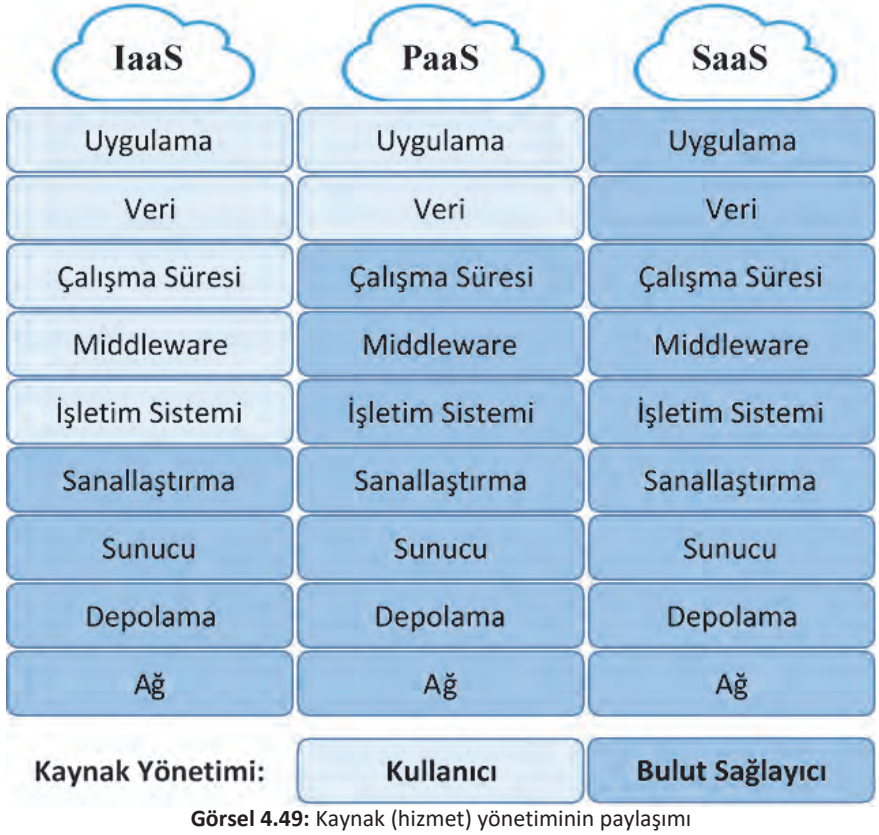
Görsel 4.48: Bulut bilişim hizmet modelleri

Hizmet Olarak Altyapı (IaaS): Bulut sağlayıcının sunucu, ağ altyapısı, depolama alanı, işlemci, bellek gibi kaynakları internet üzerinden sunduğu bulut hizmetidir. Bu hizmetin kullanıcıları, ihtiyaç duyulan donanımları satın almak yerine kullandıkça öde modeli ile kiralar. Kullanıcı kiraladığı donanımın kapasitesini istediği zaman değiştirebilir. Bu yönüyle IaaS, ölçeklenebilir bir yapıya sahiptir. Kullanıcı kiraladığı donanımlara ihtiyaç duyduğu işletim sistemini ve yazılımları kurabilir. Kiralanan altyapı hizmetinin yedeklenmesi, yönetimi, bakımı ve onarımı bulut sağlayıcıya aittir.

Hizmet Olarak Platform (PaaS): Bulut sağlayıcının sunucu, depolama alanı, ağ altyapısı, işletim sistemi, veri tabanı yönetim sistemi ve yazılım platformu sunduğu bulut hizmetidir. Bu bulut hizmetiyle kullanıcı, veri tabanı altyapısı kurmakla uğraşmadan hızlı bir şekilde uygulama geliştirip çalıştırabilir. Kullanıcılar, uygulama geliştirmede kullandıkları araçları ve yazılım lisanslarını bulut sağlayıcıdan kiralar. Kiralanan platform hizmetinin yedeklenmesi, yönetimi, bakımı ve onarımı bulut sağlayıcıya aittir. Kullanıcı yalnızca uygulama geliştirmeye odaklanır. PaaS, uygulama geliştirme ve dağıtım süreçlerinin tek elden yönetilmesini, uygulamaların uygun maliyetle geliştirilmesini ve test edilmesini sağlar.

Hizmet Olarak Yazılım (SaaS): Bulut sağlayıcının, belirli bir amaç için geliştirilmiş yazılımları internet üzerinden sunduğu bulut hizmetidir. Bu bulut hizmetiyle mesajlaşma, süreç planlama, IoT veri işleme, muhasebe, finans, e-posta gibi yazılımlar kurulum gerektirmeden web arayüz üzerinden son kullanıcıya sunulur. Kullanıcılar, bulut sağlayıcının sunduğu bu yazılımları kullandıkça öde modeliyle kiralar. Kiralanan yazılım hizmetin yedeklenmesi, bakımı ve onarımı bulut sağlayıcıya aittir. SaaS ile sunulan uygulama, çalışma zamanı, işletim sistemi, sanallaştırma, depolama alanı gibi bilgi işlem kaynaklarının yönetimi bulut sağlayıcı tarafından gerçekleşir.

Bulut bilişim hizmet modellerinin sunduğu kaynakların yönetimi kullanıcı ve bulut sağlayıcı arasında paylaşılır. Kaynak yönetiminin paylaşımı Görsel 4.49'da gösterilmiştir.



Görsel 4.49: Kaynak (hizmet) yönetiminin paylaşımı

4.6.1.3. IoT Bulut Bilişim Hizmetleri

Nesnelerin İnterneti'nde kullanılan birçok bulut bilişim hizmeti bulunur. Bu bulut bilişim hizmetlerinden bazıları şunlardır:

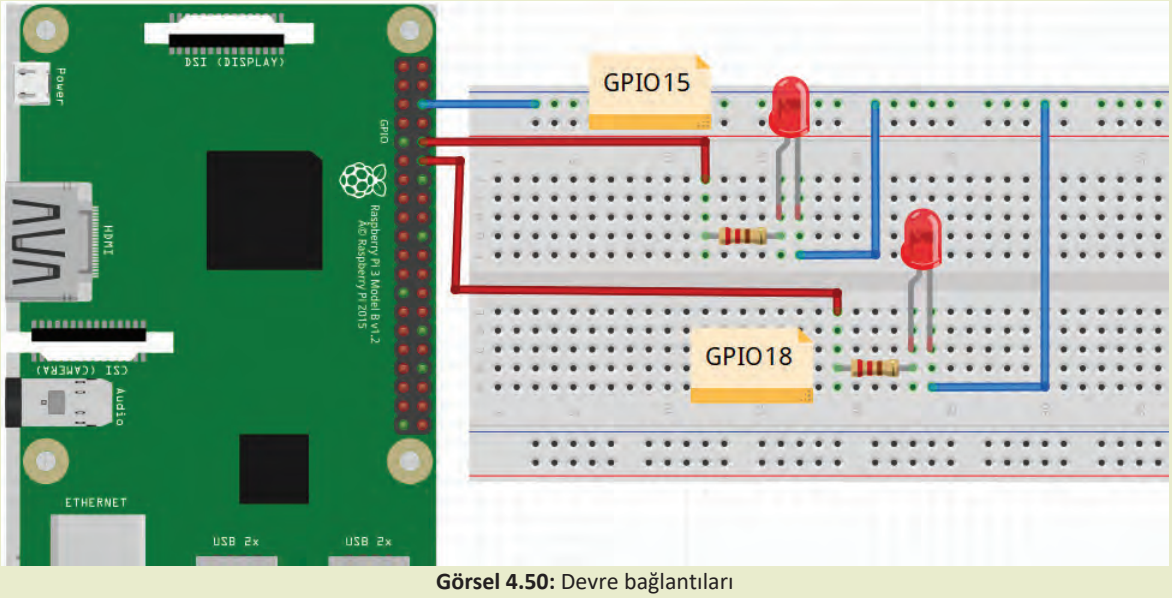
- Adafruit
- Arduino Cloud IoT
- AWS IoT
- Favoriot
- Google Cloud IoT
- Grafana Cloud
- IBM Watson IoT
- IFTTT
- Microsoft Azure IoT
- Temboo
- ThingsBoard
- ThingSpeak

3. UYGULAMA

Adafruit ile Bulut Hizmeti Uygulaması

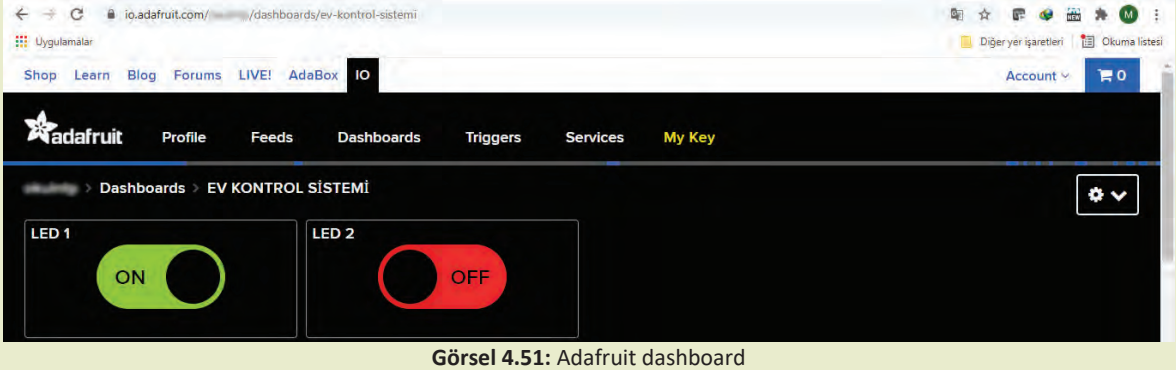
Adafruit dashboard üzerinden Raspberry Pi üzerindeki iki LED'i kontrol eden uygulamayı aşağıdaki adımları takip ederek hazırlayınız.

1. Adım : Devre bağlantılarını yapınız (Görsel 4.50).



Görsel 4.50: Devre bağlantıları

2. Adım : Adafruit bulut hizmetinde dashboardı hazırlayınız (Görsel 4.51).



Görsel 4.51: Adafruit dashboard

3. Adım : Raspberry Pi'de terminal ekranından MQTT kütüphanelerini yükleyiniz.

sudo apt-get install mosquitto
sudo apt-get install mosquitto-clients
sudo pip install paho-mqtt

4. Adım : Raspberry Pi'de Python kodunu çalıştırınız.

```
from gpiozero import LED
import paho.mqtt.client as mqtt

led1 = LED(15)
led2 = LED(18)
```

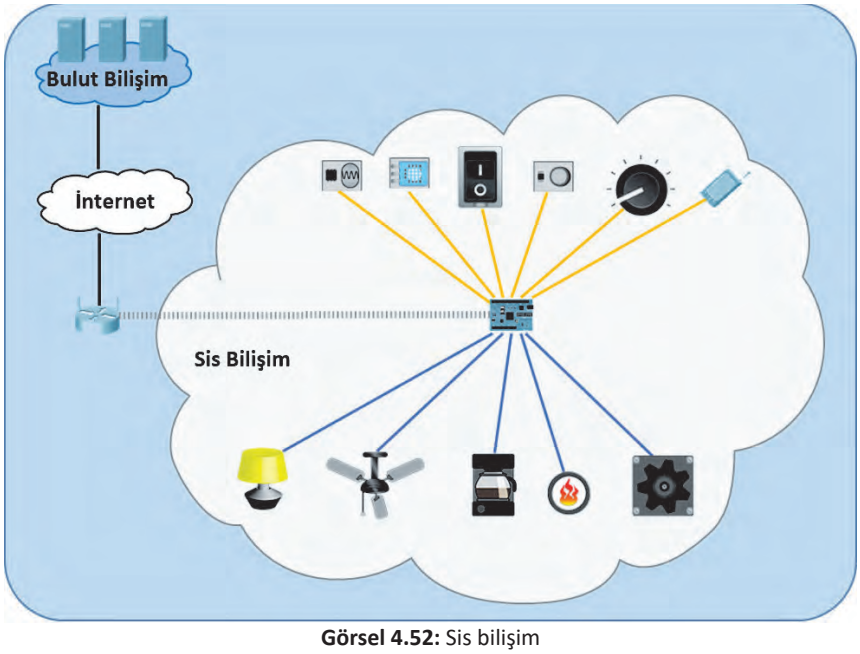
```
def baglanti_saglandiginda(client, userdata, flags, rc):
  if rc==0:
    print('Baglanti gerceklesti')
  client.subscribe("okulntp/feeds/led1", 0)
  client.subscribe("okulntp/feeds/led2", 0)

def mesaj_geldiginde(client, userdata, msg):
  mesaj = msg.payload
  konu=msg.topic
  print(str(msg)+" "+msg.topic+" "+str(msg.payload))
  print(str(konu.split('/')[2]))
  lamba=str(konu.split('/')[2])
  if mesaj == "ON":
    if lamba=='led1':
      led1.on()
    elif lamba=='led2':
      led2.on()
  elif mesaj == "OFF":
    if lamba=='led1':
      led1.off()
    elif lamba=='led2':
      led2.off()

istemci = mqtt.Client()

istemci.on_connect = baglanti_saglandiginda
istemci.on_message = mesaj_geldiginde

istemci.username_pw_set("KULLANICI ADI","API KEY")

istemci.connect("io.adafruit.com", port = 1883)

istemci.loop_forever()
```

5. Adım : Adafruit dashboard üzerinden LED'leri açıp kapatınız.

4.6.2. Sis Bilişim (Fog Computing) Modeli

Bir IoT sistemde, sensörlerle ortamdaki ani değişikliklerin algılanmasıyla aktüatörlerin hızlı bir şekilde tepki vermesi gereken durumlar olabilir. Bu durumlarda veri işlemenin ve karar vermenin daha hızlı gerçekleşmesi gerekir. Merkezi veri işlemenin gerçekleştiği bulut bilişim modelinin uzak ağlardan hizmet vermesi gecikmelere neden olabilir. Hatta internet bağlantısının kopması durumunda veri işleme tamamen aksayabilir. Bu sorunları gidermek için sis bilişim modeli önerilir.

Sis bilişim, uç cihazlardan ve sensörlerden gelen verilerin buluta gönderilmeden yerel ağda işlenmesini sağlayan bir modeldir. Sis bilişim ile ön işlemesi yapılan veriler gerektiğinde daha ayrıntılı analiz edilmesi için buluta gönderilebilir (Görsel 4.52).



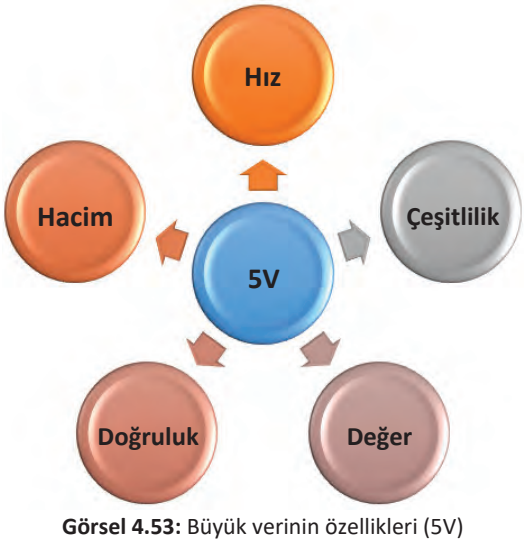
Görsel 4.52: Sis bilişim

Sis bilişim, ağ sınırına yakın cihazlara dağıtılmış hesaplama altyapısı sunar. Bu altyapı ile kritik verilerin ağın ötesine taşınması önlenerek veri gizliliği sağlanır. Bu sayede güvenlik artar. Sis bilişimin düşük bant genişliği ile çalışabilmesi ve merkezî sunucuya bağımlılığı azaltmasıyla ağın trafik yükü azalır.

4.7. BÜYÜK VERİ

Sosyal medya, fotoğraf ve video paylaşımlarının, blogların, elektronik postaların, log kayıtlarının, uydu görüntülerinin, finansal bilgilerin internetteki hacmi günden güne artmaktadır. Biriken bu veri yığınlarına mobil, IoT ve giyilebilir cihazlardan hatta sensörlerden gelen veriler de eklenir. Bu durum veri üretiminde üstel büyümeye neden olur.

Büyük veri; geleneksel yöntem ve araçlar ile işlenmesi zor olan, çeşitlilik içeren ve hacmi katlanarak artan veriyi ifade eder. Büyük verinin beş ana özelliği vardır. Bu özellikler 5V ile ifade edilir (Görsel 4.53).



Görsel 4.53: Büyük verinin özellikleri (5V)

Hacim (Volume): Katlanarak artan verinin büyüklüğünü, kapladığı alanı ifade eder.

Hız (Velocity): Verinin üretilme hızını ifade eder. Verinin hacmi hızla artar. Artan verinin hızlı analiz edilmesi gerekir.

Çeşitlilik (Variety): Verilerin farklı kaynaklardan geldiğini ve farklı türlerde olduğunu ifade eder. Veriler akıllı telefon, sosyal ağ, IoT cihaz, sensör, uydu gibi kaynaklardan gelebilir. Gelen veriler ses, fotoğraf, log kaydı, sensör verisi, e-posta, uydu görüntüsü, coğrafi koordinat vb. türlerden oluşabilir.

Değer (Value): Verilerden yararlı, anlamlı ve değerli olanların elde edilmesini ifade eder. Anlamlı veriler elde etmek için veri madenciliği yöntemleri kullanılır.

Doğruluk (Veracity): Verilerin doğruluğunu ve güvenilirliğini ifade eder.

ARAŞTIRMA

Yapılandırılmış, yapılandırılmamış ve yarı yapılandırılmış veri türleri hakkında araştırma yapınız. Araştırma sonuçlarını sınıfta arkadaşlarınızla paylaşınız.

ARAŞTIRMA

Büyük verinin 5V özelliklerine değişkenlik (variability) ve görselleştirme (visualization) eklenir. Yeni eklenen özelliklerle birlikte 7V olarak adlandırılır. Değişkenlik ve görselleştirme özelliklerinin ne ifade ettiğini araştırınız. Araştırma sonuçlarını yazınız.

Değişkenlik (Variability): .....

.....

.....

Görselleştirme (Visualization): .....

.....

.....

Analiz edilmeye hazır, ücretsiz ve kullanıma açık olan veri setleri bulunur. Bu veri setlerine ulaşmak için kaggle web sitesi yaygın olarak kullanılır (Görsel 4.54).

Görsel 4.54: Kaggle veri seti örnekleri

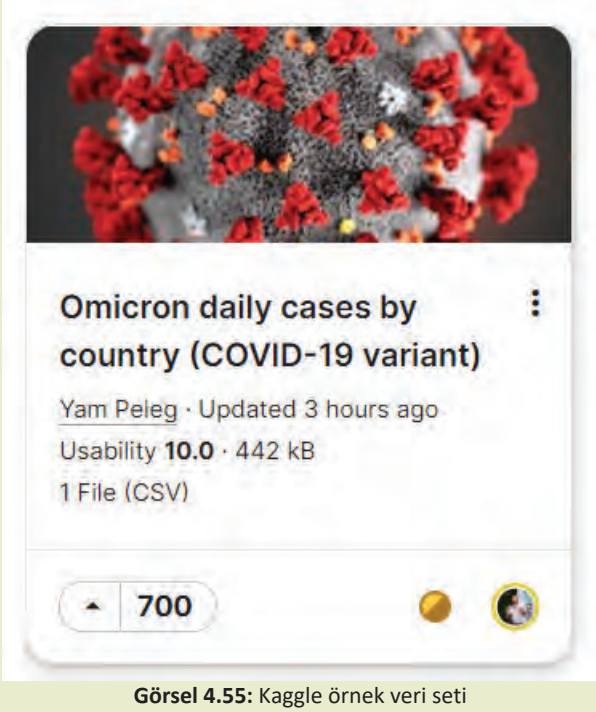
Veri madenciliği, veri setlerindeki gizli kalmış kalıpları ve örüntüleri keşfederek ham verileri anlamlı bilgilere dönüştürme işidir.

4. UYGULAMA

Örnek Veri Seti Uygulaması

Kaggle web sitesinden COVID-19 ile ilgili örnek veri setini (covid-variants.csv) indiriniz. Örnek veri setinden iki ülkenin alpha varyantı verilerini aşağıdaki adımları takip ederek kıyaslayınız.

1. Adım : Örnek veri setini indiriniz (Görsel 4.55).



Görsel 4.55: Kaggle örnek veri seti

2. Adım : Veriyi incelemek için kodları çalıştırınız.

```
#grafik modulu
import plotly.graph_objs as go
import plotly.express as px

#veri setini yuklemek icin pandas modulu
import pandas as pd

#veri setini okuyoruz
df = pd.read_csv('covid-variants.csv')

#veriyi inceleyelim
df.head()
```

3. Adım : Sonuçları inceleyiniz (Görsel 4.56).

	location	date	variant	num_sequences	perc_sequences	num_sequences_total
0	Angola	2020-07-06	Alpha	0	0.0	3
1	Angola	2020-07-06	B.1.1.277	0	0.0	3
2	Angola	2020-07-06	B.1.1.302	0	0.0	3
3	Angola	2020-07-06	B.1.1.519	0	0.0	3
4	Angola	2020-07-06	B.1.160	0	0.0	3

Görsel 4.56: İlk beş veri

4. Adım : İki ülkenin verilerini kıyaslamak için kodları çalıştırınız.

```
#Fransa ve Almanya'nin alfa varyanti vaka
#sayisini kiyaslamak icin verileri okuyalım

al = df[(df.location == 'Germany') & (df.variant == 'Alpha')]
al = al.sort_values(by="date")
al.head()

fr = df[(df.location == 'France') & (df.variant == 'Alpha')]
fr = fr.sort_values(by="date")
fr.head()
```

5. Adım : Sonuçları kıyaslayınız (Görsel 4.57).

	location	date	variant	num_sequences	perc_sequences	num_sequences_total
30168	Germany	2020-05-11	Alpha	0	0.0	105
30192	Germany	2020-05-25	Alpha	0	0.0	94
30216	Germany	2020-06-08	Alpha	0	0.0	82
30240	Germany	2020-06-22	Alpha	0	0.0	1528
30264	Germany	2020-07-06	Alpha	0	0.0	390

	location	date	variant	num_sequences	perc_sequences	num_sequences_total
27480	France	2020-05-11	Alpha	0	0.0	201
27504	France	2020-05-25	Alpha	0	0.0	64
27528	France	2020-06-08	Alpha	0	0.0	33
27552	France	2020-06-22	Alpha	0	0.0	45
27576	France	2020-07-06	Alpha	0	0.0	30

Görsel 4.57: İki ülkenin ilk beş verisi

6. Adım : İki ülkenin verilerini grafikle görselleştiriniz. Kodları yazınız.

```
#grafigi olusturalim

fig = go.Figure()
fig.add_trace(go.Scatter(x = fr['date'],
                         y = fr['num_sequences_total'],
                         mode = 'lines',
                         name = 'Fransa',
                         marker_color = 'DarkBlue'))

fig.add_trace(go.Scatter(x = al['date'],
                         y = al['num_sequences_total'],
                         mode = 'lines',
                         name = 'Almanya',
                         marker_color = 'DarkOrchid',
                         line = dict(dash = 'dot')))

fig.update_layout(title = '<b>Grafigimiz<b>',
                  title_x = 0.5,
                  title_font= dict(size = 18, color = 'Darkred'),
                  template = 'plotly_white')
fig.show()
```

7. Adım : Grafiği inceleyiniz (Görsel 4.58).

Görsel 4.58: İki ülkenin alpha varyantının kıyaslanması

4.7.1. Büyük Veri Kullanım Örnekleri

Sosyal ağ kullanıcılarının paylaşım, yorum, beğeni, arkadaşlık vb. verileri internet ortamında kayıt altında tutulur. Tutulan bu veriler analiz edilerek kullanıcının; ilgisini çekebilecek paylaşımlar, takip edebileceği sayfalar, izlemek isteyebileceği videolar hatta ihtiyaç duyduğu ürüne ait reklamlar kullanıcıya önerilir.

Elektronik ticaretin yapıldığı alışveriş sitelerinde kullanıcıların incelediği, yorum yaptığı ve satın aldığı ürünler kayıt altında tutulur. Kayıt altında tutulan verilerin analiz edilmesi sonucunda kullanıcıya ürün önerileri yapılır. Hatta kullanıcının ilgisini çeken ürün indirime girdiğinde kullanıcıya bilgi mesajı gönderilir.

Hastaların tahlil, tıbbi görüntü, ilaç, randevu vb. medikal verileri kayıt altında tutulur. Bu veriler analiz edilerek halk sağlığının gözlemlenmesi, salgın hastalıkların araştırılması, ihtiyacı artan ilaçların tespit edilmesi ve hastalara verilecek hizmetin iyileştirilmesi ile ilgili çalışmalar yapılır.

Banka müşterilerinin kredi kartı kullanımı, para transferi, nakit hareketi, fatura ve kredi ödemesi gibi verileri

kayıt altında tutulur. Banka, bu verileri analiz ederek müşteri davranışı hakkında çıkarımlarda bulunur. Banka temsilcisi, çıkarım sonucuna göre müşteriye özel kredi teklifi sunabilir, kredi kartı limit artırımı önerisinde bulunabilir.

Sürekli glikoz ölçümü yapan bir IoT cihaz, kan şekeri verilerini kısa süreli aralıklarla buluta gönderir. Bulutta bulunan kan şekeri verileri analiz edilerek kullanıcı için anlamlı bilgiler üretilir. Üretilen anlamlı bilgiler grafiklerle görselleştirilebilir. Kullanıcı; gün içindeki kan şekeri eğilimi hakkında bilgi sahibi olur.

Büyük verinin kullanıldığı birçok sektör bulunur. Örnek olarak medya, oyun, eğitim, pazarlama, lojistik, üretim, sigortacılık, enerji, ulaşım, siber güvenlik ve meteoroloji sektörleri verilebilir.

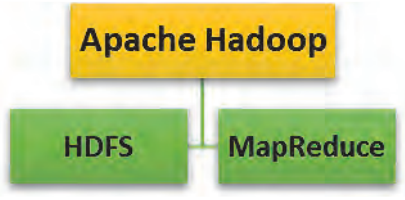
4.7.2. Büyük Veri Depolama Ortamları

Büyük veri, genellikle veri merkezlerindeki sunucularda depolanır. Büyük veri, birden fazla veri merkezinin sunucularına dağıtılır. Bu sunucularda dağıtık veri depolama sistemleri kullanılır. Böylece büyük verinin güvenliği, erişilebilirliği ve yedekliği sağlanır.

Hadoop Dağıtık Dosya Sistemi (HDFS): Analizi yapılacak olan büyük veri setlerinin dağıtık olarak depolandığı, ölçeklenebilir bir dosya sistemidir. Bu dosya sistemi Java programlama dili ile yazılmıştır. HDFS, büyük dosyaları birden fazla sunucuya parçalı şekilde dağıtır. Sunucularda depolanan dosya parçaları sanal olarak birbirine bağlıdır.

MapReduce: HDFS'de tutulan dosya parçalarını paralel şekilde analiz eden bir uygulamadır.

Apache Hadoop, HDFS ve MapReduce yazılımlarını barındıran bir koleksiyondur (Görsel 4.59).



Görsel 4.59: Apache Hadoop yapısı

4.7.3. Veri Görselleştirme Araçları

Analizi yapılan veriler, yöneticilere ve karar vericilere sunulmalıdır. Sunumda yaygın olarak çizgi, sütun, çubuk, pasta ve dağılım grafikleri kullanılır. Kullanılan grafiklerle görselleştirme sağlanır (Görsel 4.60).

Görsel 4.60: Veri görselleştirme araçları

4.8. BULUT BİLİŞİMDE GÜVENLİK

Uygulamaların ve hassas verilerin bulut ortamına taşınmasıyla birlikte güvenlik sorunları ortaya çıkmıştır. Genel bulut (public cloud) yapısında önemli güvenlik sorunları ile karşılaşılır. Bu sorunların en başında veri güvenliği gelir. Bulut bilişim, genellikle verilerin güvenli bir ortamda depolanması ve depolanan verilere her an her yerden erişilebilmesi için kullanılır.

Bulut bilişim güvenliği; bulut ortamını, bulutta depolanan verileri, bulutta çalışan sistemleri ve uygulamaları korumaya yönelik çabaların bütünüdür. Bu çaba; altyapı, uygulamalar ve platformlarda verileri gizli ve güvende tutmayı sağlar. Bulut güvenliği sağlama yöntemleri arasında yeni nesil güvenlik duvarları, sızma testi ve sanal özel ağlar (VPN) yer alır.

Bulut sağlayıcıları, hizmet sağlamak için veri merkezlerini ve sunucularını kullanır. Bu sunucularda depolanan verilerin çalınmasını önlemek için şifreleme yöntemleri kullanılır. Beklenmedik bir felaket anında oluşabilecek kayıpları en aza indirmek için verilerin düzenli yedeği alınır.

Bulut sağlayıcıları, bulut ortamına yalnızca yetkili kullanıcıların erişmesini sağlamak için güvenilir kimlik yönetimi ve kimlik doğrulama sistemleri sunar.

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde parantez içine yargılar doğru ise "D", yanlış ise "Y" yazınız.

1. ( ) WAN, coğrafi olarak küçük bir alanı kapsar.
2. ( ) HTTP, IoT uygulamaları için ek yük oluşturan ağır bir protokoldür.
3. ( ) Kablosuz ağ ortamlarında veriler radyo dalgalarıyla iletilir.
4. ( ) MQTT, senkron çalışan bir IoT protokolüdür.
5. ( ) REST API, json formatında cevap döndürür.
6. ( ) Düşük güçle çalışan cihazların IPv6 paketi iletimini 6LoWPAN sağlar.

B) Aşağıdaki cümlelerde boş bırakılan yerlere doğru sözcükleri yazınız.

7. Nesnelerin İnterneti'nin endüstriyel alanında sıkça kullanılan ve pille çalışan cihazları kablosuz geniş alan ağlarına bağlayan altyapı protokolü ..... olarak adlandırılır.
8. Yayıncı ve abone arasındaki haberleşmenin yönetiminden sorumlu aracı sunucuya .... denir.
9. XMPP, sunucu ve istemci mantığı ile çalışan ..... biçimini kullanarak anlık mesajlaşma sağlayan bir protokoldür.
10. Sensörlerden gelen verilerin bulut sunucusuna gönderilmeden ağ sınırına yakın cihazlarda analiz edilmesini sağlayan modele ..... denir.

C) Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

11. Aşağıdakilerden hangisi bulut bilişim erişim modellerinden biri değildir?

A) Genel
B) Hibrit
C) Özel
D) Platform
E) Topluluk

12. Aşağıdakilerden hangisi verilerin birçok farklı kaynaktan geldiğini ifade eden büyük veri özelliğidir?

A) Çeşitlilik
B) Değer
C) Doğruluk
D) Hacim
E) Hız

13. Aşağıdakilerden hangisi UDP protokolünün bir özelliği değildir?

A) Başlık yapısı sadedir.
B) TCP protokolüne göre daha basittir.
C) Hedefe ulaşmayan paketleri tekrar gönderir.
D) Video akışı uygulamalarında yaygın kullanılır.
E) Paket kaybının önemli olmadığı uygulamalarda kullanılır.

14. Aşağıdakilerden hangisi IoT uygulama katmanı protokolüdür?

A) CoAP
B) mDNS
C) TCP
D) ZigBee
E) 6LoWPAN

15. Aşağıdakilerden hangisi bir kablosuz iletişim protokolü değildir?

A) Bluetooth
B) Ethernet
C) RFID
D) Wi-Fi
E) 4G

16. Aşağıdakilerden hangisi Apache Hadoop koleksiyonunda büyük verinin dağıtık depolandığı kısımdır?

A) DNS-SD
B) HDFS
C) Kaggle
D) MapReduce
E) QoS

17. Aşağıdakilerden hangisi odadaki nemi ifade eden "oda/nem" etiketine MQTT protokolünde verilen isimdir?

A) Abone
B) Aracı sunucu
C) Hizmet kalitesi
D) Tema
E) Yayıncı

18. Aşağıdakilerden hangisi işletim sistemi kaynak yönetiminin kullanıcıda olduğu bulut bilişim hizmet modelidir?

A) CoAP
B) IaaS
C) mDNS
D) PaaS
E) SaaS

19. Aşağıdakilerden hangisi QoS 0 hizmet kalitesi seviyesi hakkında yanlış bilgi içerir?

A) Mesaj sunucuda depolanır.
B) Mesaj en fazla bir defa yayınlanır.
C) En düşük trafik yoğunluğuna sahiptir.
D) En güvensiz hizmet kalitesi seviyesidir.
E) Mesajın aboneye iletilmediği durumlar olabilir.

20. Aşağıdakilerden hangisi kuyruk yapısına sahip sunucu, yayıncı ve abone bileşenlerinden oluşan IoT uygulama katmanı protokolüdür?

A) AMQP
B) CoAP
C) HTTP
D) MQTT
E) XMPP

NESNELERİN İNTERNETİNDE GÜVENLİK

5. Öğrenme Birimi

KONULAR

5.1. IoT'TA GÜVENLİK RİSKLERİ
5.2. IoT SİSTEM MİMARİLERİ
5.3. IoT DONANIM KATMANI GÜVENLİĞİ
5.4. IoT İLETİŞİM KATMANI GÜVENLİĞİ
5.5. UYGULAMA KATMANI GÜVENLİĞİ

NELER ÖĞRENECEKSİNİZ?

- Nesnelerin internetinin mimarisi ve katmanları
- Nesnelerin internetinde güvenlik riskleri
- Katmanlara göre saldırı türleri ve önlemleri
- Uykudan yoksun bırakma saldırısı
- Karadelik saldırısı
- Solucan deliği saldırısı

TEMEL KAVRAMLAR

donanım katmanı, düğüme ele geçirme, gider deliği, güvenlik riskleri, karadelik, sistem mimarisi, solucan deliği, uygulama katmanı

HAZIRLIK ÇALIŞMALARI

1. Nesnelerin interneti ağları ile kablolu internet ağları arasında ne gibi benzerlikler ve farklılıklar vardır?
2. Nesnelerin interneti ağlarının mimari yapısı ile OSI modeli katmanını karşılaştırdığınızda ne gibi benzerlikler ve farklılıklar görüyorsunuz? Arkadaşlarınızla paylaşınız.

http://kitap.eba.gov.tr/KodSor.php?KOD=14400

5.1. IoT'TA GÜVENLİK RİSKLERİ

IoT teknolojisinin gelişimi ile hayatın kolaylaşması, verimliliğin artması sağlanmıştır. Bununla beraber IoT teknolojisinin kullanımının da bazı riskleri bulunur. En önemli riskler ise bilgi güvenliği, sistem güvenliği ve erişim sorunlarıdır. Ağ saldırıları teknikleri, bulunulan ortama göre değişiklik gösterir. Kablolu ortamlarda yapılan saldırı teknikleri kablosuz ortamlarda yapılamayabilir. Bu bölümde kablosuz ortamlardan biri olan IoT ortamlarında güvenlik konusu üzerinde durulmuştur.

IoT cihazları internet üzerinden veri gönderme ve kontrol edilebilme özelliklerine sahip olduklarından iletilen veri miktarı oldukça fazladır. Verinin artması riskleri de artırır. Kablosuz algılayıcı (Düğüm de denir.) ağları içinde yer alan IoT ağlarında bazı güvenlik gereksinimleri bulunur. Bunlar aşağıda listelenmiş ve Görsel 5.1'de gösterilmiştir.

Görsel 5.1: IoT ağları güvenlik gereksinimleri

1. **Veri Gizliliği:** Bir algılayıcının topladığı verilerin başka kişi veya komşu algılayıcılara sızdırılmasının engellenmesidir.
2. **Veri Bütünlüğü:** Kötü amaçlı bir algılayıcının diğer algılayıcılara erişerek veriyi bozması ve ağın çalışmasını engellemesidir.
3. **Tutarlılık:** Sistemdeki düğümlerin ortama göre çalışmasının ve veri gönderim şeklinin değişmemesidir.
4. **İzleme ve Görüntüleme:** IoT ağına bağlı düğümlerin ve yönlendiricilerin gerçek zamanlı izlenmesi ve denetlenmesidir.
5. **Kaynak Doğrulama:** Bir algılayıcının başka bir algılayıcının rolünü yaparak verileri ele geçirmemesi için kaynak doğrulaması işlemidir.
6. **Kullanırlılık:** Algılayıcıların servis devamlılığını sağlamasıdır (DoS saldırısı sırasında bile). Saldırılar, bir tür batarya bitirmeye yönelik bir saldırı ise devre dışı kalan algılayıcının olduğu yerde oluşan güvenlik zafiyeti nedeniyle sızmalar veya saldırılar yapılabilir.
7. **Ölçeklenebilirlik:** Ağın büyüyebilmesi ve gerektiğinde küçülebilmesi sırasında ihtiyaçların karşılanmasıdır.
8. **Lokalizasyon:** Algılayıcı konumlarının doğru bir şekilde belirlenmesidir. Yangın tespiti yapan bir algılayıcıdan gelen verinin konumu bilinmelidir.
9. **Şifreleme:** Düğümler ve yönlendiriciler arası iletilen verinin açık hâlde olmaması, şifrelenmesi işlemidir.
10. **Erişim:** Verilerin sadece yetkili kullanıcılarca okunabilmesidir.

Bu özelliklere sahip bir IoT ağı ile verilerin güvenli şekilde iletimi sağlanır, sadece erişim izni verilen kullanıcılar tarafından verilere erişim sağlanır. Sistem sürekli izlenir ve analizleri yapılır. Yukarıda bahsedilen gereksinimlerin yerine getirilmesi için aşağıdaki işlemlerin yapılması güvenlik risklerini azaltacaktır (Görsel 5.2).

Görsel 5.2: IoT sistemleri için güvenlik risklerinin azaltılması için yapılması gereken işlemler

1. **Madde: Web Arayüzü Yapılandırması:** IoT cihazlarını yönetim için web arayüzünün yapılandırılması gerekir. Cihazların ayarlarının yapıldığı arayüze erişim için kullanılan varsayılan şifreler değiştirilmelidir.
2. **Madde: Yetkili Kişilere Erişim:** IoT cihazlarına sadece yetkili kişilerce erişim sağlanmalıdır. Bu kişilere erişim için şifre tanımlanmalı ve bu şifrelerin güvenli şifre özelliklerine sahip olması gerekir. Şifrelerin güvenlik için belirli aralıklarla değiştirilmesi de güvenlik riskini azaltmaya yardımcı olur. Yönetici ve kullanıcı yetkileri ayrıca tanımlanmalı ve erişim hakları sınırlandırılmalıdır.
3. **Madde: İletilen Verilerin Şifrelenmesi:** IoT cihazları arasında yapılan veri iletişimi sırasında verinin açık ve okunabilir bir şekilde gönderilmemesi gerekir. IoT ağına sızmış kötü niyetli kullanıcılar verileri okuyabilir, değiştirebilir veya silebilir. Bu sorunların engellenebilmesi için veriler şifrelenmeli ve ağ trafiği içinde şifreli olarak yer almalıdır.
4. **Madde: Gerekli Hizmetlere İzin Verilmesi:** IoT ağı içinde kullanılan tüm cihazlara ait sadece ihtiyaç duyulan hizmetler (servisler) açılmalıdır. Açık olan hizmetler ne kadar fazla ise meydana gelebilecek zafiyet de o kadar fazla olur. Bu sebeple ihtiyaç duyulan servisler açılmalı, diğer servisler kapalı hâle getirilmelidir.
5. **Madde: Zafiyetler İçin Güncelleştirmeler:** IoT ağında kullanılan cihazların üzerinde çalışan bir yazılım mevcuttur. Bu yazılım üretici firmalarca yüklenir. Zaman içinde oluşabilecek risklere ve zafiyetlere karşı üretici firma tarafından yayınlanan güncellemeler takip edilmeli ve cihazlara yüklenmelidir.
6. **Madde: Fiziksel Güvenlik:** IoT cihazlarına yapılan saldırılar arasında IoT cihazını ele geçirme saldırıları olduğu da unutulmamalıdır. Cihazların fiziksel olarak ortada bir yerde bulunmaması, korunaklı bir yapı içinde yer alması, kolaylıkla sökülüp alınmaması veya değiştirilmemesi gerekir. Hatta cihazın erişime açık portlarının (USB, firewire gibi) erişime engellenmesi veya kapatılması da güvenlik risklerini azaltacak uygulamalardır.

SIRA SİZDE

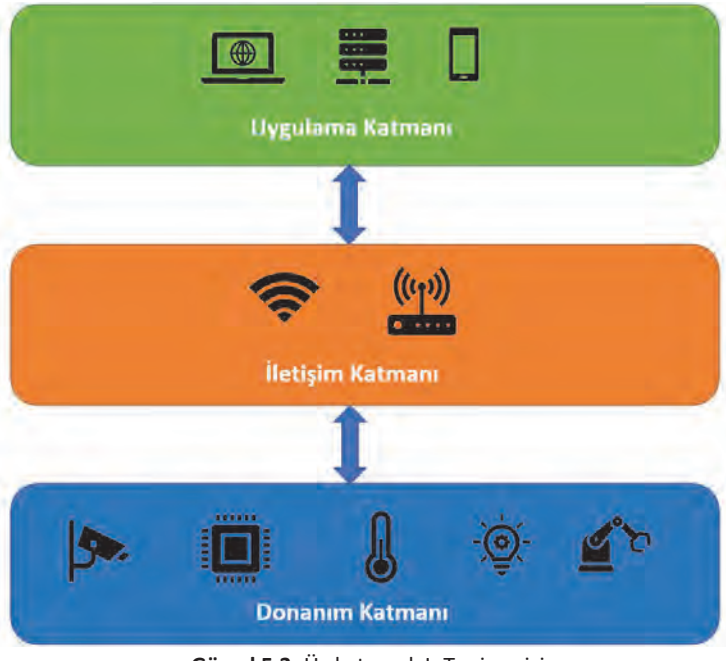
IoT uygulamanızda kullandığınız veya kullanmayı düşündüğünüz bir cihazın (sensör veya diğer cihazlar) yönetim paneline erişim için kullanılan varsayılan şifrenin değişim işlemini yapınız.

ARAŞTIRMA

IoT sistemlerine yapılan saldırı / atak tipleri gruplandırıldığında bunlar iç ve dış ataklar olmak üzere ikiye ayrılır. İç ve dış atak tiplerini araştırarak bulgularınızı bir sunu hâline getiriniz ve sınıfınıza sunumunu yapınız.

5.2. IoT SİSTEM MİMARİLERİ

IoT teknolojileri, insanlar tarafından sağladığı fayda ve hizmetler için kullanılır. IoT teknolojilerinin sağladığı birçok fayda olmasına rağmen yönetim, enerji verimliliği, güvenlik, gizlilik gibi bazı zorlukları da bulunur. IoT cihazları kullanıldığı zaman boyunca büyük miktarda bir veri meydana getirir. Oluşan büyük miktardaki verinin yönetilmesi ve IoT teknolojisinin kullanımındaki zorluklar; katmanlara ayrılmış modelleri zorunlu kılar. Bu modellere mimari adı verilir. Nesnelerin interneti özelinde mimari; fiziksel bileşenleri, ağ organizasyonu ve ayarları, yönetimsel işlemleri ve kullanılacak veri tipini ifade eden bir yapıdır. Ancak IoT sistemleri birçok farklı teknolojiyi kapsayan bir teknoloji olduğundan IoT için standartlaşmış tek bir mimari bulunmaz. Temel ve yaygın olarak IoT sistemlerinde 3 katmanlı bir mimari kullanılırken daha özelleştirilmiş IoT sistemlerinde 4, 5 veya 7 katmanlı mimariler de kullanılabilir. Mimari içinde yer alan her bir katman genellikle yöneticilerinin IoT sisteminin tutarlılığını değerlendirmesi, izlemesini sürdürmesi ve yönetimini kolaylaştırması için kullanılır. Görsel 5.3'te üç katmanlı IoT mimarisi görülmektedir.



Görsel 5.3: Üç katmanlı IoT mimarisi

5.2.1. Uygulama Katmanı

Bu katman IoT sisteminin fiziksel katmanıdır. Bu katmanda sensörler, cihazlar ve bağlı portlar yer alır. Bu katmanda sistemin ihtiyacına göre sensörler aracılığı ile veriler toplanır.

5.2.2. İletişim Katmanı

Bu katmanda, donanım katmanında toplanan verilerin iletilmesi gerçekleştirilir. Cihazların diğer sensör (düğüm), sunucu ve ağ cihazlarına bağlanma ve verilerin iletilmesi işleri bu katmanda yapılır.

5.2.3. Donanım Katmanı

Uygulama katmanı kullanıcıların IoT sistemleri ile iletişimde bulundukları katmandır. Kullanıcıların IoT sistemini kullanma, izleme ve yönetme gibi işlemleri için özelleştirilmiş yazılımlar bu katmanda çalıştırılır. Yazılımlar, IoT sensörlerine veya cihazlarına erişim sağlayarak hem kullanıcının isteklerini uygulama katmanına iletir hem de uygulama katmanından gelen verileri anlamlandırarak görüntülenmesini sağlar.

5.3. IoT DONANIM KATMANI GÜVENLİĞİ

Genel amacı yaşam kalitesini artırmak olan IoT teknolojisinin akıllı binalardan iş yerlerine ve şehirlere kadar kullanım alanı bulunur. IoT cihazları farklı türde birçok veriyi toplayarak insana yardımcı olurken aynı zamanda güvenlik ve gizlilik gibi protokolleri de önemli hâle getirmektedir. Tüm ağ teknolojilerinde olduğu gibi IoT sistemlerinde de güvenlik ve gizlilik protokollerinde sürekli olarak iyileştirme ve güncelleme yapılmasına rağmen güvenlik riskleri hâlâ devam etmektedir. Bu sebeple IoT sistemleri tasarlanırken güvenlik tasarımına önem verilmelidir. Güvenlik risklerinin azaltılması için alınması gereken önlemler, donanım katmanından başlayarak üst katmanlara doğru devam etmelidir.

5.3.1. Fiziksel Katmanda Karşılaşılabilecek Güvenlik Riskleri ve Alınacak Önlemler

IoT fiziksel katmana yönelik yapılan saldırılar yalnızca bir düğümün fiziksel olarak ele geçirilmesini veya iletişimin ele geçirilmesini amaçlamaz. Aynı zamanda iletişim sisteminin ayrıntılarını öğrenerek tüm IoT sistem güvenliğini de tehdit eder. Bu nedenle IoT fiziksel katmanına yönelik saldırı tipleri bilinmelidir.

5.3.1.1. Düğümü Ele Geçirme (Tempering) Saldırısı

Bu saldırı türü kurcalama veya Tempering olarak da bilinir ve tamamen fiziksel bir saldırı türüdür. Amaç, algılayıcıların (Sensör veya düğüm de denilir.) ele geçirilmesidir. Ele geçirilen algılayıcılara, saldırgan tarafından kendisi için özelleştirilmiş yeni yazılım yükleme veya yüklü yazılımı güncelleyerek değiştirme gibi teknikler uygulanarak saldırı gerçekleştirilir.

Düğümü ele geçirme saldırı türüne karşı alınacak önlemler şunlardır:

- Hafıza içeriğine bir müdahale yapıldığında hafıza sıfırlanır ve tekrar kullanılamaz hâle getirilir.
- Yeni bir yazılım yüklenememesi için hafıza içeriği şifrelenebilir. Ancak bu işlemin performans düşmesi gibi olumsuz tarafı mevcuttur.
- Yazılımın değiştirilememesi için yazılım koruma teknikleri kullanılabilir (obfuscation) veya kullanıcı adı ve şifre kombinasyonu ile kodlara erişim sağlanabilir.

5.3.1.2. Yayın Bozma (Jamming) Saldırısı

Bu saldırı türünde saldırı yapacak kişi, güçlü bir anten kullanarak yeni sinyaller üretir. Bu sinyaller IoT cihazlarına (algılayıcılara veya düğümlere) parazit oluşturur. Bu parazit iletişim hâlindeki cihazların yayınını bozar. Yayın bozma saldırısının bazı türleri aşağıda gösterilmiştir.

- **Sürekli yayın bozma:** Yayını sürekli olarak bozmak için sürekli radyo sinyalleri gönderilir.
- **Aldatıcı yayın bozma (Deseptive):** Bu saldırı türünde, sürekli radyo sinyali gönderilmez. Bunun yerine düzenli olarak gönderilen paketlerin arasına algılayıcıları aldatmaya yönelik paketler eklenir. Bu şekilde gönderilen paketlerin algılanması çok daha zordur.
- **Rastgele yayın bozma:** Sinyaller rastgele bir zaman aralığında ve rastgele bir sürede gönderilir. Sinyal gönderilmediği zamanlarda saldırgan düğüm uyku moduna geçer. Yeniden sinyal göndereceği zaman uyku modundan çıkar ve rastgele yeniden sinyal gönderir.
- **Tepkili yayın bozma (Reactive):** Ağdaki trafik durumuna göre pozisyonunu belirler. Ağda bir trafik varsa sinyal gönderme işlemi yaparak iletişimde bozulmalara neden olur. Ağda trafik yoksa beklemede kalır. Bu nedenle tespit edilmesi oldukça zordur.

Yayın bozma saldırılarına karşı alınabilecek önlemler şunlardır:

- Sinyal gücü, taşıyıcı algılama süresi ve paket teslim oranı gibi bilgiler kullanılarak hesaplama yapılır. Hesaplama sonucu, belirli bir değer üzerindeyse sinyalde yayın bozulması olduğu söylenebilir. FHSS (Frekans Atlamalı Dağınık Yayılma) veya DSSS (Düz Sıralı Dağınık Yayılma) gibi teknikler kullanılarak engellenebilir.
- Bölgesel Haritalama (regional mapping) tekniği kullanılarak yayın bozma saldırısı yapılan bölgelerin tahmin edilmesi sağlanabilir. Saldırı tespit edilirse iletişim kanalı değiştirilerek saldırı önlenebilir.

5.3.1.3. Zararlı Kod Aşılama (Malicious Code Injection) Saldırısı

Algılayıcı düğüme fiziksel veya uzaktan erişerek zararlı kodların yerleştirildiği bir saldırı türüdür.

Zararlı kod aşılama saldırısına karşı alınabilecek önlemler şunlardır:

- Rastgeleleştirme Algoritmaları (IRS) anahtarı kullanarak istenmeyen zararlı kod bloku, işlemci tarafından derlenmesi engellenerek önlenebilir.
- AES şifreleme algoritması kullanılarak rastgeleleştirme işlemi gerçekleştirilir ve işlemcinin kodları çalıştırması engellenebilir.

5.3.1.4. Uykudan Yoksun Bırakma Saldırısı

IoT cihazları bir batarya ile beslenen aygıtlar oldukları için enerjileri sonsuz değildir. IoT sistemlerinde cihazlar, enerji tüketiminin en aza indirilmesi için işlem yapmadıkları zamanlarda uyku moduna geçirilir. Ancak ağda bir saldırgan düğüm varsa algılayıcıları sürekli meşgul ederek onların uyku moduna geçmesini engeller. Bu nedenle batarya ömürleri beklenenden daha kısa sürede dolar.

Uykudan yoksun bırakma saldırısına karşı alınabilecek önlemler şunlardır:

- Cihazların enerji tüketimi izlenir ve beklenenden fazla enerji tüketen cihazların iletişim bilgileri kontrol edilerek buna sebep veren IoT cihazı tespit edilir.

5.4. IoT İLETİŞİM KATMANI GÜVENLİĞİ

IoT iletişim katmanı; sensörler, cihazlar ve kullanılan diğer servislerin birbirleriyle haberleşmelerini sağlayan katmandır ve IoT sistemlerinin omurgası olarak kabul edilir. IoT sistemindeki bir üst katman olan uygulama katmanı ile farklı işletim faaliyetleri yürütülür. Tüm fiziksel sistem, diğer düğümlerle paylaşılması gereken miktarda veri ve bilgi ile yüklenir. IoT sisteminde bir iletişim protokolü aracılığıyla düğümler arasında uygun bir bağlantı ağının kurulması ve paylaşılması gereken verilerin iletilmesi gerekir. İletişim, IoT sistem tasarımcısı tarafından tanımlanan protokole göre kablolu veya kablosuz olabilir. Bunlar arasında Ethernet, Wi-Fi, NFC, ZigBee, WSN (Wireless Sensor Network), LTE (Long Term Evolution), 5G, LoRaWAN, IPV6/6LowWPAN ya da Bluetooth tercih edilebilir.

5.4.1. İletişim Katmanında Karşılaşılabilecek Güvenlik Riskleri ve Alınacak Önlemler

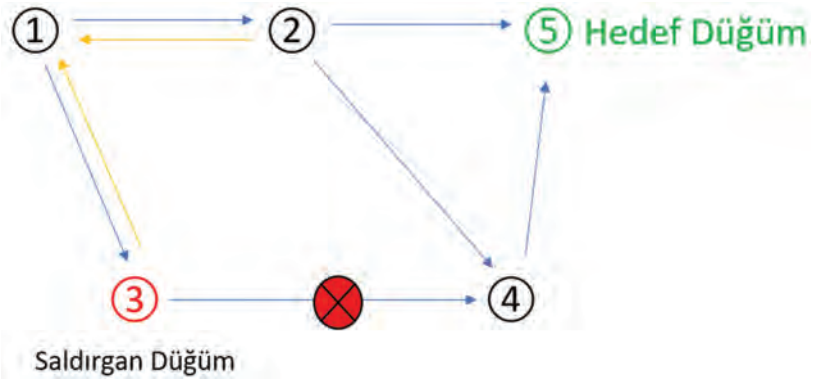
IoT sensör ve cihazları arasında veri iletişimi sürerken verilerin güvenliği ve gizliliğine yönelik olarak gerçekleştirilen saldırılar oldukça çeşitlidir. Bu saldırılar ve alınacak tedbirler başlıklar hâlinde aşağıdaki şekildedir.

5.4.1.1. Gider Deliği (Sinkhole) Saldırısı

Saldırgan, ağdaki tüm trafiği belirli bir bölgeye çekmesi dolayısı ile diğer algılayıcıların sağlıklı veri göndermesinin ve almasının engellenmesi sonucunda oluşan saldırılardır. Daha çok en kısa yol bilgisini komşu algılayıcılar ile paylaşarak tüm trafiğin kendi üzerinden aksamasını sağlayabilir.

5.4.1.2. Karadelik (Blackhole) Saldırısı

Saldırgan düğüm, yönlendirme tablosu üzerinde değişiklik yapar ve kendine komşu olan algılayıcı düğümlerin kendisine veri göndermesini sağlar ancak bu paketleri başkasına göndermez ve veri akışı durur. Bu saldırı türü ile ağdaki cihazlardan baz istasyonuna (Sink) veri gönderimi olmayacağı için sink devre dışı kalabilir. Görsel 5.4'te karadelik saldırısına ait bir örnek verilmiştir.



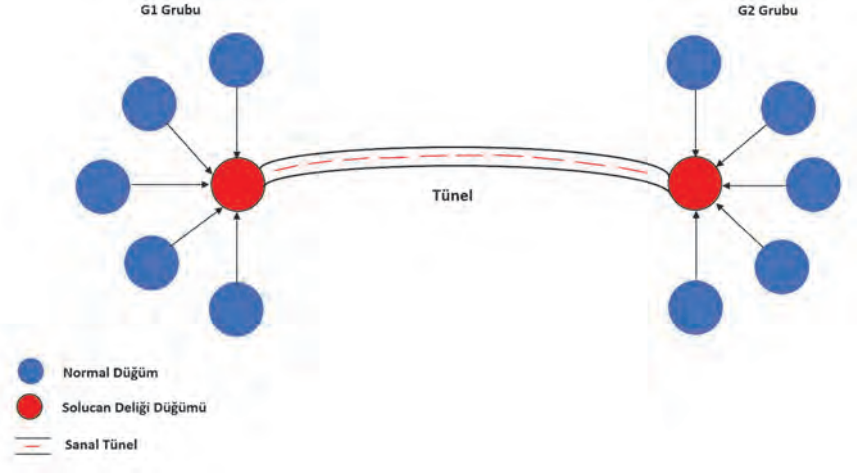
Görsel 5.4: Karadelik saldırısı gösterimi

Bu örnekte 1 numaralı düğümden 5 numaralı düğüme bir veri gönderimi yapılacaktır. Bunun için;

- 1 numaralı düğüm kendi komşuları olan 2 ve 3 numaralı düğümlere bakarak 5 numaralı düğüme giden en kısa yolun hesaplamasını yapar.
- 2 numaralı düğümün 5 numaralı düğüme daha yakın olduğu varsayılsa bile 3 numaralı saldırgan düğüm, en kısa yolun kendisinden geçtiğini 1 numaralı düğüme bildirir.
- 3 numaralı düğüm bu bildirimi 2 numaralı düğümden süre olarak daha önce yaparak 1 numaralı düğümü ikna eder. Bunun sebebi en erken gelen bildirimin en yakın komşudan gelecek olmasıdır.
- Bu durumda 1 numaralı düğüm veri paketini 3 numaralı düğüme gönderir.
- 3 numaralı düğüm veriyi hiçbir zaman 4 numaralı düğüme aktarmaz. 5 numaralı sink düğümü, 1 numaralı düğümden gelen veri paketlerini alamaz ve iletişim başarısız olur.

5.4.1.3. Solucan Deliği (Wormhole) Saldırısı

Bölgesel olarak iki kısma ayrılmış bir IoT ağında iki kısımda da saldırgan düğümler arasında oldukça düşük gecikme süreli veri iletişim bağlantısı (tünelleme) yapılan saldırı türüdür. Saldırgan düğümler aldıkları paketleri tünelin karşısında bulunan diğer saldırgan düğüme gönderir. Diğer düğümler verileri komşu düğümlere gönderdiklerini sanırlar ama veriler başka bir yere aktarılır. Görsel 5.5'te solucan deliği saldırısına örnek verilmiştir.



Görsel 5.5: Solucan deliği saldırısı gösterimi

Solucan deliği saldırısını tespit etmek için kullanılan bazı teknikler aşağıda gösterilmiştir.

- **Packet Leash Tekniği:** İki düğüm arasındaki veri iletim zamanından mesafeyi hesaplama tekniğidir. Düğümler arası mesafe normal bir zamanda (saldırının olmadığı gözetimli bir zaman) hesaplanır. İletişim süresi hesaplanmış olan bu mesafede daha öncekilerden farklı bir değer hesaplanırsa solucan deliği saldırısının olduğu kanaatine varılır.
- **Delphi Tekniği:** Basit bir gecikme analizi yaklaşımı kullanılır. Veriyi gönderen düğüm her atlamadaki gecikmenin ortalamasını hesaplar. Delphi tekniğinde, solucan saldırısı varsa atlama sayısının artmış olacağı varsayılır.

5.4.1.4. Sybil Saldırısı

Sybil saldırısı bir düğümden sahte olarak başka bir düğümün üretilmesi (kopya düğüm) ile oluşan atak türüdür. Bu saldırıyı engellemek için önerilen çözüm yöntemleri ise kimlik doğrulama ve şifreleme tekniklerinin beraberce kullanılmasıdır.

5.4.1.5. Merhaba Seli (Hello Flood) Saldırısı

Bir düğüm kendine komşu olan düğümleri tanımak ve keşfetmek için "Hello" paketleri gönderir. Paketi alan düğüm verinin kendine yakın bir komşusundan geldiğini düşünür. Oysaki güçlü sinyal üreten bir saldırgan düğüm ***hello*** paketleri göndererek düğümlerin yanıt vermesini sağlar. Bu sayede ağ sürekli olarak meşgul olur ve bu paketleri alan düğümlerin güç tüketimi artar.

SIRA SİZDE

Packet Leash tekniğini araştırarak bir sunum hazırlayınız ve sınıfta sunumunu yapınız.

5.5. UYGULAMA KATMANI GÜVENLİĞİ

IoT uygulama katmanı, uygulamaya özel hizmetlerin kullanıcıya sunulmasından sorumlu katmandır. Akıllı evler, akıllı şehirler ve diğer IoT uygulamalarının yer alabileceği çeşitli uygulamalar için kullanılan katmandır.

5.5.1. Uygulama Katmanında Karşılaşılabilecek Güvenlik Riskleri ve Alınacak Önlemler

Uygulama katmanı genel olarak en fazla saldırı yapılan katmanlardan biridir. Uygulama katmanı IoT cihazları ile kullanıcılar arasındaki iletişimi sağladığından, farklı türde oluşabilecek birçok güvenlik riskleri mevcuttur. IoT cihazlarının saldırılardan korunması için bazı güvenlik kısıtlamaları uygulanmalıdır. Saldırılardan bazıları aradaki adam saldırısı (man in the middle), SQL kod aşılama saldırısı (SQL injection), kopyalama saldırısı, yeniden programlama saldırısı ve DoS saldırısıdır.

5.5.1.1. Aradaki Adam (Man in the Middle) Saldırıları

MQTT protokolü, istemci ve üye olmak üzere iki gruba ayrılır. Üyeler, istemci tarafından yayınlanan mesajları alırlar. Saldırgan MQTT aracısını kontrol eder ve araya girerek giden ve gelen mesajları okur. Dsniff, Cain, Ettercap, Wsniff, Airjack gibi saldırılar bu gruba girer.

Bu saldırı riskini en aza indirmek için anahtarlama yönetim protokolleri (ağ yapısına özel) ile veriler şifrelenmelidir.

5.5.1.2. SQL Injection Saldırısı

Saldırgan metin girişi yapılan herhangi bir yerde özel karakterler veya sorgular kullanarak SQL kodlarının istediği gibi çıktılar vermesini ve bu şekilde veri tabanına erişmesini sağlar. IoT cihazlarında kullanıcının özel anahtarlarını dahi ele geçirebilir.

Bu saldırı riskini en aza indirmek için veri girişi yapılan yerlerde girilen veriler filtrelenmelidir. Sorgu yapılan veri tabanına erişmeden önce dijital imza kontrolü ve erişimde kısıtlama doğrulaması yapılmalıdır. Erişim kontrolünde uyuşmazlık veya şüpheli bir durum varsa erişime engel konulmalıdır.

5.5.1.3. Kopyalama Saldırısı

Saldırgan, algılayıcı düğümleri ele geçirip onların kopyalarını oluşturur. Burada ele geçirme fiziksel olarak ele geçirilip düğümün kodlarını başka bir cihaza aktarır. Cihazın iletişim yönetimi kendi elinde olduğundan ağ içindeki veri trafiğini izler ve verilerin gizliliğini, erişilebilirliğini ihlal eder.

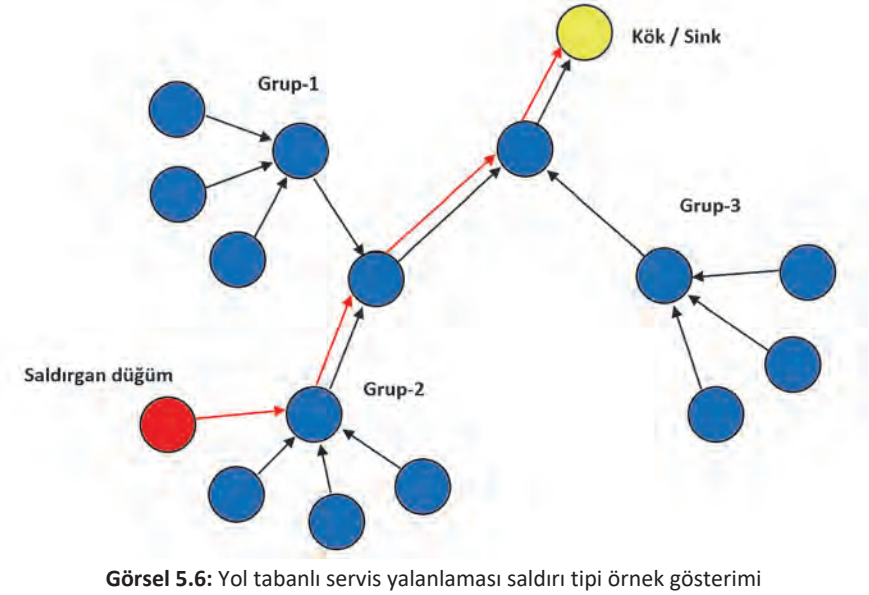
Bu saldırı riskini en aza indirmek için bir düğümün her bir komşusu onun imzasını doğrular. Ağda aynı anda aynı imzadan veya kimlik numarasından bir başka düğüm varsa en son eklenen düğüm saldırgan düğümdür denilir ve onunla iletişim kurulmaz. Eğer mevcut cihaz başka kötümcül (saldırgan) düğümle yer değiştirirse farklı bir yöntem önerilebilir. Her düğüm üzerinde (ön belleğinde) en kısa yol ve bunlara ait cihaz ID bilgileri yer alır. İletişim sırasında cihazlardan biri, farklı bir cihaz ID'si ile karşılaşırsa bir sorun olduğu tespit edilebilir. Bu işlemlerin yapılabilmesi için cihazların ağa eklenmeden önce yönetici tarafından doğru bir şekilde mimari tasarımının yapılması şarttır.

5.5.1.4. Yeniden Programlama Saldırısı

Genelde fiziksel olarak cihaza yönelik yapılan saldırılardır. Ayrıca ağ güvenliği zayıf olan IoT sistemlerinde cihazlara erişebilen saldırganlar cihazı yeniden programlayabilmektedir. Bu yönü ile bu saldırı türü, cihazın ele geçirilmesi yöntemine benzer. Eğer cihaz gereken güvenlik tedbirleri alınmadan programlanmışsa saldırgan, cihazlara erişmekte zorlanmaz ve yeniden programlama yapabilir.
Bu güvenlik riskini en aza indirmek için cihazlara erişim için şifre konulmalıdır. Programlama yapabilmek için erişim izni protokolleri uygulanmalıdır.

5.5.1.5. Yol Tabanlı Servis Yalanlaması (Path Based DoS) Saldırısı

Hiyerarşik ağaç yapısındaki bir algılayıcı ağında en alttaki yaprak düğümün ağa sürekli mesaj göndermesi nedeniyle kök ağaca kadar (Sink) giden bir yol sürekli meşgul edilmiş olabilir. Bu süreçte ağacın o bölümü sürekli işgal edilir. Sürekli meşgul olan düğümlerin şarjları tükenir ve yaşam süresi azalır (Görsel 5.6).



Görsel 5.6: Yol tabanlı servis yalanlaması saldırı tipi örnek gösterimi

Bu saldırı riskini en aza indirmek için paket kimlik doğrulaması, zaman damgası kullanılması, yeniden tekrarlamaya karşı koruma (anti-replay) teknikleri kullanılabilir.

5.5.1.6. DoS (Denial od Services) Saldırısı

Ağın hizmet veremeyecek kadar meşgul edilmesi için sürekli mesaj gönderimi (SYN paketleri) yapılarak gerçekleştirilen bir saldırı türüdür. DoS atağının genel amacı, IoT ağında kullanılan kaynakları tüketmek için gereksiz paketler gönderme ve IoT cihazlarının mevcut kaynaklardan veya ağ servislerinden yararlanmasını engellemektir. DoS atağı, bir IoT cihazının kendisinden beklenen görevi yapmasının engellenmesi veya performansının büyük bir oranda düşürülmesi olarak da tanımlanabilir.

IoT sistemlerinde kullanılan düğümler ve diğer tüm cihazlar fiziksel veya uzaktan ele geçirilebilir olduğu için ağ içinden DoS atağı geliştirilebilir. DoS saldırısı, IoT ağının tüm katmanlarında meydana gelebilir. Donanım katmanında yapılan DoS atağı gürültü yaratarak iletişimin engellenmesi olarak yapılabilirken, iletişim katmanında çarpışma, paket düşürme, hatalı yönlendirme veya karadelik oluşturma türünde yapılabilir.

Bu saldırı riskini en aza indirmek için kaynak kullanımını sınırlandırma, güçlü kimlik doğrulama ve trafik tanımlama gibi metotlar kullanılır. Bu metotların kullanılmasında genellikle yapay zekâ teknikleri ile istatiksel yaklaşımlar önerilmektedir. Bu yaklaşımlardan en yaygın olarak kullanılanlar aşağıda listelenmiştir:

- Bayes ağları
- KNN (En yakın komşu) algoritması
- SVM (Destek vektör makinesi) algoritması
- Derin yapay sinir ağları

5.5.1.7. Algılayıcı Düğümün Boğulması (Overhelming Sensor Node) Saldırısı

DoS ve DDoS saldırısının bir türevidir. Amaç, bir veya daha fazla düğümün sürekli meşgul edilmesidir. Bunun için hedef düğüme yoğun ve sürekli bir şekilde mesajlar gönderilir. Sürekli ve yoğun şekilde gönderilen mesajlar, IoT sisteminde kullanılan bant genişliğini ve bataryayı oldukça fazla şekilde tüketir. Bant genişliği azalan IoT ağında iletişim yapılamaz hâle gelir. Ayrıca hedef düğümlerin bataryası bittiğinden iletişimde sorunlar oluşur ve veri iletimi yapılamaz. Bu sebeple de ağ çökme noktasına gelir.

Bu saldırı riskini en aza indirmek için yönlendirme algoritmalarında enerji tüketimine önem verilmelidir. Saldırının tespitinde DoS saldırı tipinde olduğu gibi yapay zekâ tabanlı algoritmalar kullanılabilir.

ARAŞTIRMA

- SYN paketlerini, paket kimlik doğrulaması tekniğini, yeniden tekrarlamaya karşı koruma (anti-replay) tekniğini araştırarak ayrıntılı bir çalışma hazırlayınız. Hazırladığınız çalışmayı raporlaştırınız ve sınıfta sunum yapınız.
- Ettercap, Wsniff saldırılarını araştırınız. Nasıl yapıldıklarını ve ne gibi önlemler alınması gerektiğini sınıfınızla tartışınız.

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde parantez içine yargılar doğru ise "D", yanlış ise "Y" yazınız.

1. ( ) IoT mimarisi genel olarak üç katmandan oluşur. Ancak 4 veya 5 katmanlı IoT mimarileri de bulunur.

2. ( ) IoT cihazları arasında veri iletimi yapılırken veriler açık ve okunabilir şekilde yapılandırılmalıdır.

3. ( ) DoS saldırısı uygulama katmanına özgü bir saldırı türüdür.

B) Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

4. Aşağıdakilerden hangisi IoT ağlarındaki güvenlik gereksinimlerinden biri değildir?

A) Kaynak Doğrulama
B) Kopyalama
C) Şifreleme
D) Tutarlılık
E) Veri Gizliliği

5. Aşağıdakilerden hangisi IoT sistemlerinde güvenlik risklerinin azaltılması için yapılması gereken işlemlerden biri değildir?

A) Verinin şifrelenmesi
B) Web arayüzü yapılandırması
C) Güncelleştirmelerin yapılması
D) Sadece gerekli servislere izin verilmesi
E) Düğümlerin bataryalarının belirli aralıklarla değiştirilmesi

6. Aşağıdakilerden hangisi iletişim katmanı saldırılarından biridir?

A) Karadelik saldırısı
B) Kopyalama saldırısı
C) Yayın bozma saldırısı
D) Yeniden programlama saldırısı
E) Uykudan yoksun bırakma saldırısı

IOT UYGULAMALARI

6. Öğrenme Birimi

KONULAR

6.1. ENDÜSTRİYEL IoT UYGULAMALARI
6.2. IoT GÜVENLİĞİ
6.3. IoT SİSTEM UYGULAMALARI
6.4. IoT'TA MAKİNE ÖĞRENMESİ VE YAPAY ZEKÂ

NELER ÖĞRENECEKSİNİZ?

- Endüstriyel IoT uygulamaları geliştirme
- IoT'ta güvenlik ilkeleri
- IoT'ta sistem uygulamaları geliştirme
- IoT sistemlerde makine öğrenmesi ve yapay zekâ mantığı

TEMEL KAVRAMLAR

akıllı şebeke, endüstri 4.0, güvenlik, makine öğrenimi, Nesnelerin İnterneti, security, smart grid, thingspeak, yapay zekâ

HAZIRLIK ÇALIŞMALARI

1. Yeni nesil fabrikalar hiç insan gücü olmadan üretim yapabilir mi? Fabrika farklı bir ülkede, makinelerin yönetimi farklı bir ülkede olabilir mi? Düşüncelerinizi paylaşınız.
2. Hayati önem taşıyan ameliyatların uzaktan yapılması nasıl mümkün olmaktadır?
3. Çeşitli iş kollarında IoT nesneleri kullanılırken ortaya çıkabilecek güvenlik problemleri neler olabilir? Sınıfta tartışınız.

http://kitap.eba.gov.tr/KodSor.php?KOD=14401

6.1. ENDÜSTRİYEL IoT UYGULAMALARI

Nesnelerin İnterneti, çeşitli sensörler ve mikrodenetleyiciler yardımıyla cihazların birbirine bağlanarak verilerin uzaktan kontrol edilip analiz edilmesi mantığına dayanır. Çeşitli sensörler ile birbirini ve dış ortamı algılayarak iletişime geçen cihazlar; fabrikalarda, şehirlerde, okullarda, evlerde ve çevrede sürekli olarak verileri toplayıp büyük bir bilgi deposu oluşturabilir. IoT cihazları, bu bilgi deposunda bulunan verileri analiz ederek internet üzerinden kontrol edilebilen cihazlar hâline gelir.

Nesnelerin internetinin uzaktan algılama ve analiz etme özelliğiyle birçok alanda çeşitli uygulamalarla karşılaşılır. Sağlık, akıllı şehirler, akıllı enerji sistemleri, lojistik, tarım ve daha birçok sektörde nesnelerin interneti uygulamaları kullanılır.

IoT cihazları sensörler aracılığı ile çevredeki verileri algılar, ağ kurarak diğer nesnelerle haberleşir, topladıkları verileri depolayarak büyük bir veri alanı oluşturur. IoT cihazları bu bilgilerin analiz edilmesini sağlar ve bu veriler ile bulut sistemlerdeki servisleri kullanır. Bütün bu özellikleri sayesinde IoT kavramı endüstriyel uygulamalarda kullanılmaya başlanmıştır.

Nesnelerin interneti cihazlarının endüstride ilk kullanım örnekleri RFID cihazları ile olmuştur. RFID okuyucularla fabrikalardaki üretim ağı sistematiğe bağlanmış olup hammadde girişinden son ürün olarak çıkışına kadar RFID ile veriler takip edilmiştir. Bu sayede üretim aşamaları kolayca takip edilerek hem ürün güvenliği hem de süreç kolaylığı sağlanmıştır.

SCADA: SCADA, nesnelerin internetinin önemli uygulama alanlarından biridir. SCADA, merkezî izleme ve uzakta bulunan üretim ve iletim sistemlerinin kontrolünü sağlar. Bu izleme ve kontrol aktüatörler, kontrolörler, haberleşme cihazları ve merkezî ünite ile yapılır. Sensörlerden gelen veriler merkezî istasyona iletilir ve insan makine arayüzü olan HMI arabirimiyle kontrol edilir. Ayrıca daha sonra işlemleri analiz etmek amacıyla bu veriler zaman damgalı olarak saklanır.

Akıllı Ölçüm: Geleneksel enerji ölçüm sistemlerini nesnelerin interneti teknolojisi ile akıllı hâle getirmek akıllı şebeke uygulamalarının en önemli unsurlarındandır. IoT ile akıllı sayaçlar, uzaktan ölçüm işlemlerini yöneterek işletme maliyetlerini azaltmaya yardımcı olur. Ayrıca periyodik enerji maliyeti tahminlerinin yapılmasını, enerji hırsızlığı ve kaybının önüne geçilmesini sağlar. IoT tabanlı cihazlar ile mobil ve web uygulamaları üzerinden bu ölçümler kontrol edilebilir.

Akıllı Şebekeler: Akıllı ölçümler bölümünde belirtildiği gibi akıllı şebekeler tedarikçi ve tüketici arasında iletişim kurulmasını sağlar (Görsel 6.1). Bu şebekelerin en önemli bileşenleri akıllı sayaçlardır. Akıllı şebekeler, elektrik üretimine optimize ve dağıtım yük talebine göre mevcut enerji arzının daha iyi kullanılmasını sağlar. Özellikle tüketimin yoğun olduğu saatlerde iyi bir güç dağıtımı için sistemin etkili şekilde koordine edilerek trafo otomasyonu arza göre sağlanır. Üretim istasyonları arasında eksik veya fazla olan üretimde çevrimiçi iletişim ile planlama yapılır.

Görsel 6.1: Akıllı şebekelerin iletişimi

IOT sistemlerin endüstride kullanımının ve otomasyon sistemlerine entegre edilmesinin büyük faydaları bulunur. Bu faydalar aşağıda belirtilmiştir.

- Anlık olarak üretim aşamasının izlenmesi
- Hata kontrolünün IoT cihazları tarafından otomatize olarak yapılması ve insan hatasının minimuma indirgenmesi
- İnsan sağlığı ve güvenliğinin maksimuma çıkması
- Akıllı sensörler, akıllı denetleyiciler ve makineler arası ağ kurma özellikleri sayesinde üretimden maksimum verim elde etme
- Üretim verilerinin dünyanın her yerinden erişilebilir olması
- Hammadde eksikliğinin gerçek zamanlı tespiti ve önlem mekanizmalarının hızlı çalışmasının sağlanması
- Arızaların erken tespiti ve fabrika üretim hattının durma süresinin minimuma indirgenmesi

6.1.1. Enerji Sektöründe IoT

Nesnelerin internetinde insan gücüne ihtiyaç duymaksızın çeşitli cihazlar ve algılayıcılar birbirleri ile iletişim kurarak bilgi toplar ve bu bilgileri kullanarak çeşitli işler gerçekleştirir. Enerji sektöründe nesnelerin internetinin bu özelliği kullanılarak birçok uygulama ile karşılaşılır. Çevreye zararlı olan gazların tespiti ve takibi, akıllı sayaçlar ve akıllı şebeke sistemleri ile enerji kullanımının gerçek zamanlı izlenmesi bu uygulamalardan bazılarıdır. İzlenen verilerin analiz edilmesi ile enerjinin verimli kullanımının sağlanması da bu cihazlar sayesinde olur. Çevrede bulunan nesnelerin internete bağlanması ve birbirleri ile iletişimde olması zamandan ve kullanılan enerjiden tasarruf sağlar. Sensörlerle çalışan sokak lambaları, akıllı ev sistemleri, akıllı su sistemleri ve kombiler enerjinin daha verimli ve tasarruflu kullanımında önemli bir adım olarak karşılaşılan örneklerdir.

IoT, elektrik enerjisi sektöründe güç tüketimini ve maliyetini düşürmek amaçlı kablosuz teknolojili bir altyapı oluşturur. Enerji sektöründe IoT kullanımının örneklerinden bazıları; SCADA, akıllı çözüm, bina otomasyonu, akıllı şebekeler ve kamu aydınlatma olarak sayılabilir.

Bir fabrikanın enerji noktaları uzaktan izlenebilir. Elektrik tüketimi verileri anlık olarak şebekeye bağlanmış IOT mikrodenetleyicisi ve sensörler tarafından otomatik takip edilir. Bu kapsamda saat saat harcanan enerji tüketimi kayıt altına alınır. Gerekli noktalarda sistem otomatik analiz yaparak enerji tüketiminde verimlilik sağlamaya çalışır. Bu kapsamda enerji tüketiminin akşam saatlerinde azaldığı ve sabahtan öğlene kadar en yüksek noktaya ulaştığı izlenir. Bu izleme esnasında çeşitli noktalarda bulunan cihazların fazla enerji kullandığı ve bu cihazların fazla olmasının gereksiz olduğu tespit edilmiştir. Ayrıca elektrik tedariki yapan firma ile de anlaşılarak sabah saatlerinde daha uygun fiyattan elektrik tedarik edilirken akşam saatlerinde daha pahalı tutardan alınması sağlanmıştır. Bu analizler sayesinde fabrikanın elektrik fatura tutarı azaltılır ve enerji verimliliği de artırılır. Ayrıca atık kontrol mekanizması da nesnelerin interneti ile kontrol altına alınıp takip edilerek çevrenin sürdürülebilirlik özelliğine katkıda bulunulur.

6.1.2. Sağlık Sektöründe IoT

Bilgi, iletişim ve internet teknolojilerinde gelişmeler, sağlık hizmetlerinde önemli bir rol oynar. Bu gelişmeler medikal enformasyon sistemlerinin geliştirilmesine katkıda bulunur. Tıp ve bilgi teknolojilerinin yakınlaşması ile sağlık hizmetleri dönüşüm sürecine giriş durumundadır. Bu yakınlaşma neticesinde sağlık hizmetlerinde maliyetlerin düşmesi, verimsizliklerin azalması ve en önemlisi ise yaşam süresinin uzaması gibi gelişmeler sağlanmıştır.

Nesnelerin internetinin sağlık sektörü içinde dikkat çekici birçok etkisi bulunur. Nesnelerin interneti uzaktan sağlık izleme, fitness programları, kronik hastalıklar ve yaşlı bakımı gibi önemli uygulamalar sağlar. Ayrıca evde tedavi ve ilaçla uyumluluk, sağlık profesyonelleri açısından önemli bir diğer uygulamadır. Bu durumda

çeşitli tıbbi cihazlar, sensörler, tanılama ve görüntüleme cihazları, IOT'nin ana parçasını oluşturan akıllı cihazlar veya nesneler olarak sayılabilir.

NOT

Geliştirilen bir giyilebilir kalp ritmi analizi cihazı ile algılayıcılar tarafından hastanın kalp atış ve diğer bilgileri otomatik kayıt altına alınır. Kalp atış bilgileri hastanın ve doktorun cep telefonunda bulunan mobil uygulamaya sürekli olarak bilgi akışı sağlar. Kalp atışlarında bir anormallik olduğu takdirde doktor ve hasta bildirimler sayesinde haberdar olur. Hayati tehlike boyutuna varabilecek anormalliklerde ise sistem otomatik olarak sağlık kuruluşlarını arayarak bilgilendirir. Hastanın kalp verileri analiz edilerek hastanın yaşantısı organize edilir.

6.1.3. Akıllı Şehirlerde IoT

Şehirlerin kentsel sürdürülebilirlik planlarında teknolojinin önemli bir rolü bulunur. Bunun sebebi, yeni teknolojilerin vatandaşlara fayda sağlayan sağlam çözümler sunmasıdır. Şehirler, akıllı sistemleri yürütmüş oldukları faaliyetlere (altyapı, eğitim, sosyal faaliyet vb.) dâhil etmeyi amaçlar. Akıllı şehir uygulamaları vatandaşlara sunulan hizmetlerin kalitesini artıran ve tüm süreçleri daha verimli hâle getiren akıllı teknolojilerle yönetilir.

Akıllı şehirler IoT çözümlerinin yanı sıra yapay zekâ, makine öğrenmesi, bulut bilişim hizmetleri ve uygulama programlama arayüzleri vb. teknolojileri de kullanır (Görsel 6.2).

Dünyanın dört bir yanındaki şehirler akıllı teknoloji geliştirme ve uygulamada farklı aşamalardadır. Tamamen akıllı şehirler yapma yolunda öncülük eden şehirler bulunmaktadır.

Görsel 6.2: Akıllı şehirlere temsili gösterim

NOT

Singapur şehir devleti kamusal alanların temizliğini, kalabalık yoğunluğunu ve kayıtlı araçların hareketini izleyen IoT kameralarıyla tamamen akıllı şehirler oluşturma yarışında önde gelen ülkelerden biri olarak kabul ediliyor. Singapur ayrıca enerji kullanımını, atık yönetimini ve su kullanımını gerçek zamanlı olarak izlemek için sistemlere sahiptir. Ek olarak yaşlıların sağlık ve esenliğini sağlamak için otonom araç testi ve izleme sistemi bulunmaktadır.

6.1.4. Tarım Uygulamalarında IoT

Çiftçiliğin olağanüstü hava koşulları, artan iklim değişikliği ve çevresel etkileri gibi zorluklarla mücadele etmesine rağmen daha fazla gıda talebi karşılanmalıdır. Artan bu ihtiyaçları karşılamak için tarımın yeni teknolojilere yönelmesi gerekir. IoT teknolojilerine dayanan yeni akıllı tarım uygulamaları, tarım endüstrisinin gübre kullanımını optimize etmekten çiftlik araçları rotalarının verimliliğini artırmaya kadar atıkları azaltmasını ve verimliliği artırmasını sağlayacaktır.

IoT tabanlı akıllı tarımda, mahsul alanını sensörler (ışık, nem, sıcaklık, toprak nemi vb.) yardımıyla izlemek ve sulama sistemini otomatikleştirmek için bir sistem inşa edilmiştir. Çiftçiler tarla koşullarını her yerden izleyebilirler. IoT tabanlı akıllı tarım, geleneksel yaklaşımla karşılaştırıldığında oldukça verimlidir.

IoT tabanlı akıllı tarım uygulamaları; sadece geleneksel, büyük tarım operasyonlarını hedeflemekle kalmaz, aynı zamanda organik tarım, aile çiftçiliği (karmaşık veya küçük alanlar, belirli sığırlar ve/veya kültürler gibi) tarımdaki diğer büyüyen veya yaygın eğilimleri yükseltmek için yeni çözümler olabilir.

Kurulan akıllı bir sera ile üretilecek tarımsal ürünlerin hangi sıcaklıkta olacağı kayıt altına alınarak izlenmektedir. Sıcaklığın artması gerekiyorsa sistem otomatik olarak ısı vericilerin derecesini artırmaktadır. Sıcaklık ortamda çok fazlaysa havalandırma sistemini devreye sokmaktadır. Toprak ve üretilen ürünün nem miktarı sensörler ile algılanarak otomatik sulanmaktadır. Ayrıca tarımsal drone cihazları ile ilaçlama yapılarak sistem tam olarak otomatik hâle getirilmiştir. Böylelikle ürünler daha sağlıklı ve daha performanslı üretilmektedir.

ARAŞTIRMA

Siz de akıllı tarıma örnek olarak gösterebileceğiniz bir çalışmayı araştırınız ve arkadaşlarınıza sunumunu yapınız.

6.2. IoT GÜVENLİĞİ

IOT cihazlar internet ağına ya da kurumlardaki yerel ağlara bağlanıp çevre ve birbirleriyle etkileşime giren cihazların oluşturduğu sistemlerdir. IoT cihazları birçok bilgiyi topladığı gibi bu bilgilerin birleşiminden oluşan büyük veri alanları da oluşturur. Akıllı kilitler, akıllı ev cihazları, internet üzerinden kontrol edilen araçlar, kameralar gibi birçok nesne internete bağlanır ve bilgiler toplanıp analiz edilir. Bu noktada IOT cihazlarının güvenliğinin çok önemli olduğu gerçeği ile karşılaşılır.

Cihaz güvenliklerinin sağlanmaması ya da güvenli cihaz üretilmemesi sonucunda nesneler siber saldırılara açık bir hâle gelir. Günümüzde birçok cihaz internete bağlı olarak işlev gördüğünden bilgisayar korsanları cihazları ele geçirmek için daha fazla fırsata sahiptir.

Bu noktada IoT cihazlarının üretim aşamasından fiziksel olarak bulunduğu ağ ortamına kadar güvenlik ilkeleri uygulanmalıdır. Bu noktada alınabilecek çeşitli güvenlik uygulama ilkeleri şunlardır:

- Güvenli cihaz geliştirme ilkesi
- Güvenli bağlantı oluşturma ilkesi
- Ağ güvenliğini sağlama ilkesi
- Veri depolama güvenliği ilkesi

6.2.1. Güvenli Cihaz Geliştirme İlkesi

IoT uygulamaları geliştiricileri yazılımlarını oluştururken her işlemde güvenliğe önem vermelidir. SQL kodlarında bulunan açıklar, HTML açıkları veya veri tabanı kod zafiyetleri saldırganlar için birçok fırsat sunar ve yazılımdan kaynaklı cihazlarda güvenlik zafiyetleri doğurur.

Güvenlik Geliştirme Yaşam Döngüsü, bunun yapılmasına yardımcı olan bir yazılım geliştirme sürecidir.

Bu süreç aşağıdaki yedi aşamadan oluşur:

- **Eğitim Aşaması:** Güvenli tasarım, tehdit modelleme, güvenli kodlama, güvenlik testi ve gizlilik ile ilgili en iyi uygulamalar da dâhil olmak üzere güvenli yazılımlar geliştirmek için temel ilkelerin bulunduğu aşamadır.
- **Gereksinimler Aşaması:** Proje başlangıcı adımı, yasal gereksinimler de dâhil olmak üzere projenin temel güvenlik ve gizlilik sorunlarını değerlendirmek ve listelemek için gerekli görünen aşamadır.
- **Tasarım Aşaması:** Tüm yazılım özellikleri güvenli değildir. Bu nedenle tasarımda ek güvenlik katmanlarının eklenebileceği aşamadır.
- **Uygulama Aşaması:** Onaylanmış araçların kullanıldığı, güvenli olmayan tüm işlevlerin kaldırıldığı ve bu aşamalar sırasında doğru analizlerin gerçekleştirildiği aşamadır.
- **Doğrulama Aşaması:** Bu aşama, kodun gereksinimler ve tasarım aşamalarında belirlenen güvenlik ve gizlilik kurallarını karşıladığından emin olmak için kullanılan aşamadır.
- **Piyasaya Sürme Aşaması:** Güvenlik olaylarını izlemek ve bunları hızla yanıtlamak için bir plan oluşturulan aşamadır.
- **İzleme Aşaması:** Yazılımın kullanıcı dönütleri sonrasında geliştirildiği ve güvenliğin artırıldığı aşamadır.

IoT yazılımları, bu aşamalar ve bu aşamaların ilkeleri doğrultusunda tasarlanıp yazılımı hazırlanmalıdır.

ARAŞTIRMA

Akıllı ev sistemine sahip bir kişi, evinin bütün noktalarını internet üzerinden bir uygulama aracılığı ile kontrol etmekte ve izlemektedir. Bu kişi akıllı kilit yöntemiyle ofisinin kapısını açmakta ve çeşitli dijital asistanlar yardımıyla da günlük rutin işlerini takip etmektedir. Kişinin cihaz güvenliği yazılımı noktasında yaşayacağı bir aksaklığın ne tür sorunlar doğurabileceğini araştırınız. Araştırma sonuçlarını sınıfta tartışınız.

6.2.2. Güvenli Bağlantı Oluşturma İlkesi

Yerel ağa ya da internet ağına bağlanarak işlem gören IOT cihazlarında gerek nesnenin gerekse kullanıcıların güvenli bağlantı oluşturması gerekir. Bu noktada çeşitli ilkeler ve çözümler ile karşılaşılır.

- **Ağ Geçidi Güvenliği:** Cihazların yerel ağdan çıkıp internet ağına bağlanarak diğer ağlarla iletişime geçebilmesi için bir ağ geçidine (Gateway) ihtiyacı vardır. Bu noktada güvenlik ilkelerine uyulmamış bir ağ geçidi yapılandırması saldırganın rahat bir şekilde IoT nesnesine ve ağına giriş yapabilmesini sağlar.
- **Bulut Tabanlı Güvenlik:** Veriler bulut ortamında tutulduğu takdirde mutlaka sunucu cihazların fiziksel ve yazılımsal güvenliğinin sağlanması gerekir.
- **Tehdit İzleme:** Çeşitli kötü amaçlı yazılımların saldırılarına maruz kalmamak için güncelliği sağlanmış bir antivirüs sisteminin bulunması, güvenli bağlantı oluşturmanın önemli bir ilkesidir.
- **Güncel Sistem:** Bağlantı sağlayacak sistemin güncel olması ve tüm güvenlik paketlerinin sistemde yüklü olması gerekir. Güncelliğini yitirmiş ya da güncelleme almamış cihazlar her zaman tehditlere açıktır.

6.2.3. Ağ Güvenliğini Sağlama İlkesi

IoT cihazın bulunduğu ağ ortamına yalnız istenilen cihaz ve kullanıcıların bağlanması gerekir. Bu noktada ağ ortamında erişim listeleri oluşturulmalı ve ağa dâhil olması istenmeyen kişilerin giriş yapmaları engellenmelidir.

- **Güvenlik Duvarı:** Ağ ortamında bulunacak olan bir güvenlik duvarı ve güvenlik duvarı yapılandırması sayesinde IoT cihazlarının bulunduğu ağlara dâhil olacak kişiler ya da cihazların filtrelenmesi yapılmalıdır. Bu sayede IoT cihaz ve ağ ortamı güvenliği sağlanmalıdır.
- **Kimlik Doğrulaması:** Uygun şifreleme anahtarlamaları ile IoT sistemine dâhil olacak cihazların kimlik doğrulamaları yapılmalıdır. Böylelikle kötü niyetli kişilerin sisteme girmesi engellenmelidir.
- **Ağ İzlenmesi:** Ağ izleme yazılımları ile ağ ortamı anlık izlenmeli ve tehdit oluşturabilecek durumlar algılandığında kişiler uyarılmalıdır.
- **Ağ Cihazlarının Doğru Yapılandırması:** 2. katmanda ya da 3. katmanda görev yapacak ağ cihazlarının yapılandırma hatalarından arındırılması gerekir. Bu sayede gerek yerel ağdan gerekse internet ortamından gelebilecek tehditlerin önüne geçilmelidir.

ARAŞTIRMA

Akıllı sağlık hizmetlerini kullanan bir hastanede IOT aygıtları kullanılarak hastalardan nabız atış verileri toplanmaktadır. Hastane yerel ağında bulunan yapılandırma hatalı bir anahtarlama cihazına yapılacak bir saldırı hangi aksaklıklara sebebiyet verebilir? Sınıfta tartışarak güvenlik risklerini not alınız.

6.2.4. Veri Depolama Güvenliği İlkesi

IoT cihazlarından gelen anlık veriler ya da kontrol verileri belli bir alanda tutularak depolanıyorsa bu verilerin güvenliği sağlanmalıdır. Parmak izi, kredi kartı, ses ve görüntü gibi hassas veriler depolama alanlarında tutulabilir. Gerek yerel ağda sunucularda tutulacak verilerin gerekse bulut ortamında tutulacak verilerin güvenliğinin sağlanması için çeşitli ilkeler uygulanmalıdır.

- **Sunucu Güvenliği:** Verilerin tutulduğu sunucu cihaz, doğru yapılandırılmalıdır. Sunucu, dışarıya kapalı yetkisiz kişilerin girişini engellemelidir.
- **Tarama ve Raporlama:** Gerek yerel ağ ortamında gerekse bulut sisteminde bulunan güncel ve gelişmiş antivirüs sistemleri ile belli periyotlarda tarama ve raporlama yapılmalıdır.

ARAŞTIRMA

Akıllı ev sistemleri geliştiren bir firma aynı zamanda akıllı kilit sistemleri ile kullanıcılarını tanıyan kapı kilitleri geliştirmiş ve müşterilerine sunmuştur. Müşterilerin sunucuda parmak izleri tutulmaktadır. Sunucuda doğacak bir güvenlik açığının nelere yol açabileceğini araştırınız. Araştırma sonuçlarını sınıfınızda tartışınız.

6.3. IoT SİSTEM UYGULAMALARI

IoT uygulamaları tarım uygulamalarında akıllı şehir uygulamalarında aktif olarak uygulandığı gibi sağlık alanında da giderek artan bir uygulama alanına sahiptir. Uzaktan hasta izleme sistemlerinde kalp atışı, nabız, kandaki oksijen miktarı ölçümü bunlara örnek olarak gösterilebilecek temel uygulama alanlarındandır.

6.3.1. Sağlık Alanında IoT Uygulaması

1. UYGULAMA

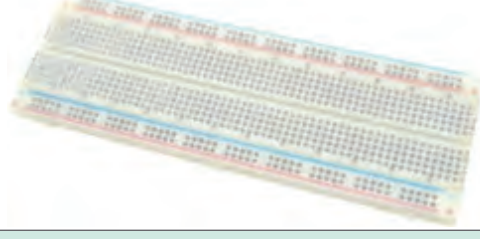
NodeMCU ve kalp atış sensörü kullanarak kalp atış hızı verilerinin web ortamında yayınlanması sağlayan uygulamayı yapınız.

Bu uygulamada, tasarım (Görsel 6.3), kurulum, program kodlarının yazılması, verilerin toplanması, verilerin uzak sisteme aktarılması ve grafiksel gösterimi yöntem olarak temel alınmıştır.

Görsel 6.3: Akıllı sağlık sistemi için blok diyagram

1. Adım : Gerekli malzemeleri Tablo 6.1'de gösterildiği gibi hazırlayınız. MX30100 sensörü hakkında bilgi edininiz.

Tablo 6.1: Sağlık Alanında IoT Uygulaması İçin Gerekli Malzemeler

1. NodeMCU V3	2. MX30100 Kalp Atış Sensörü
3. Breadboard	**4. Kablo**
	

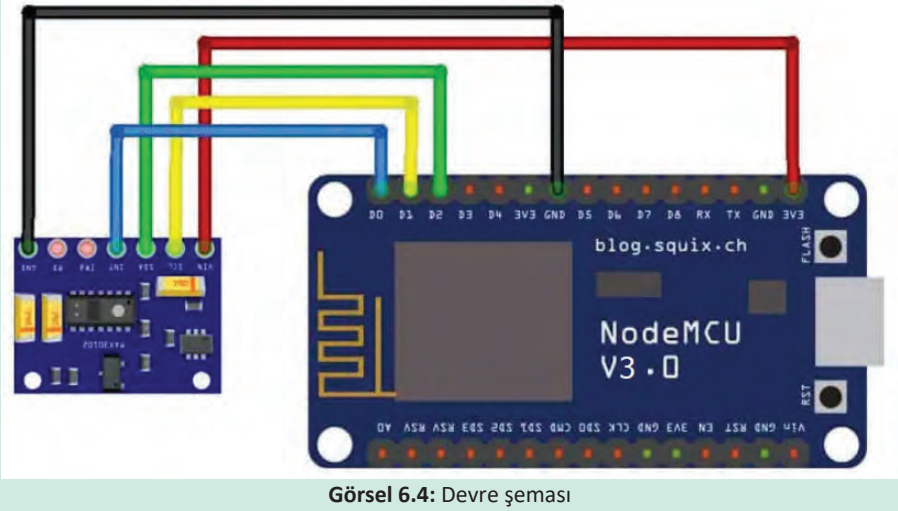
MX30100 Sensörü

Kalp-nabız sensörü, Arduino, ESP8266, ESP32 gibi kontrol kartları için tasarlanmış kalp atış sensörüdür. Canlı kalp atış hızı verileri projelerde rahatlıkla kullanılabilir. Sensör parmak ucuna veya kulak memesine takılır. Sensörün kalp logosu olan tarafı cilt ile temas eden tarafıdır. Farklı ışık koşullarında sağlıklı çalışabilmesi için ortam ışığı sensörü içerir. İçerdiği kırmızı ve kızılötesi LED'lerden ışık yayarak ve bu ışığın yansımasını ölçüm alarak çalışır. Bu LED kombinasyonu nabız ölçümü için en uygun rengi sağlar. LED'lerden gelen sinyal, bir mikroişlemci tarafından işlenir ve hedef mikrokontrolcüye I2C hattı üzerinden gönderilir. Ayrıca I2C hattı ile kullanılabilen INT pini de mevcuttur. Besleme gerilimi 3.3V'tur.

Özellikler:

- Model: Nabız Sensörü
- Besleme Gerilimi: 3V-5V
- Akım: 4mA/5V
- Çap: 16mm
- Kulak memesi veya parmaktan nabız ölçer.
- Sensörün çevresinde bulunan 3 delik sensörü giysilere montajlama imkânı sağlar.
- Dâhili yükseltici ve gürültü önleme devresi içerir.
- Giyilebilir cihaz tasarımında rahatlıkla kullanılabilir.

2. Adım : Görsel 6.4'te gösterilen devreyi kurunuz.



Görsel 6.4: Devre şeması

3. Adım : Arduino.cc programını çalıştırınız. Arduino IDE editörü, yerel makinede XAMPP sunucu ve bir metin düzenleyici kullanılacaktır. Aşağıdaki kütüphaneleri programınıza ekleyiniz ve tanımlamaları yapınız. BMP ve SpO2 adlı iki değişken tanımlaması yapılmıştır.

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <Adafruit_ADXL345_U.h>
#include "math.h"
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
```

1

```
#include <OneWire.h>
#include <DallasTemperature.h>
#include "MAX30100_PulseOximeter.h"
float BPM, SpO2;
ADC_MODE(ADC_VCC);
const char* ssid = "Senin Wifi SSID Adı";
const char* password = "Senin Wifi Şifren";
const char* serverName = "http://192.168.1.4/test.php";
String apiKeyValue = "tPmAT5Ab3j7F9";
PulseOximeter pox;
String kimlik = "777";
WiFiClient wifiClient;
```

Kod Satırı / Bloku	Açıklama
1	Uygulamayı çalıştırmak için gerekli olan kütüphanelerin yüklendiği bölümdür.
2	NodeMCU mikroişlemci kartının internete erişebilmesi için gerekli olan ağ adı ve ağ şifresinin tanımlandığı bölümdür.
3	Sensörden gelen verilerin internet tarayıcısında görüntülenmesi için gerekli olan localhost adresinin tanımlandığı kod satırıdır.
4	Bu kod (apiKeyValue) rastgele kişi tarafından yazılabilir. Güvenlik için kullanılır. Bu kod tarayıcıdan iletiliyorsa işlem yapılmakta, iletilemiyorsa işlem yapılmamaktadır.

Wi-Fi ayarlarınızı programa giriniz. SSID adı ve Wi-Fi şifrenizi giriniz. Sensörden gelen veriler pox adı altında toplanacaktır.

4. Adım : Ana program bloku (Void Setup()) içine geçiniz. Numaralı kısımlarda açıklamalar verilmiştir.

```
void setup(void)
{
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());
    if (!pox.begin()) {
    Serial.println("FAILED");
    for (;;);
```

```
  } else {
    Serial.println("SUCCESS");

  }
  delay(2000);
}
```

3

Kod Satırı / Bloku	Açıklama
1	Programın seri bağlantı hızı 115200 olarak ayarlanmıştır. Wi-Fi bağlantısı başlatılmıştır.
2	Wi-Fi'ye bağlantı işleminin kontrol edildiği bölümdür. Eğer bağlanma işlemi başarılı ise "SUCCESS" değilse "FAILED" mesajı ekranda yazacaktır.
3	2000 milisaniye (2 saniye) bekleme kodudur.

5. Adım : Sürekli olarak çalıştırılacak kısım olan LOOP bloku kodlanmasına geçiniz ve aşağıdaki kod blokunu programınıza ekleyiniz.

```
void loop(void)
{
    pox.update();
    BPM = pox.getHeartRate();
    SpO2 = pox.getSpO2();
    Serial.print("BPM: ");
    Serial.println(BPM);
    Serial.print("SpO2: ");
    Serial.print(SpO2);
    Serial.println("%");
    Serial.println("**********************************");
    Serial.println();
    if(WiFi.status()== WL_CONNECTED){
    HTTPClient http;

    http.begin(wifiClient,serverName);

    http.addHeader("Content-Type", "application/x-www-form-urlencoded");

    String httpRequestData = "kimlikno=" + apiKeyValue + "&BPM=" + BPM
                           + "&SpO2=" + SpO2 + "";
    Serial.print("httpRequestData: ");
    Serial.println(httpRequestData);
    int httpResponseCode = http.POST(httpRequestData);
    if (httpResponseCode>0) {
      Serial.print("HTTP Response code: ");
      Serial.println(httpResponseCode);

    }
    else {
      Serial.print("Error code: ");
```

1

2

3

4

```
      Serial.println(httpResponseCode);
    }
    http.end();
  }
  else {
    Serial.println("WiFi Disconnected");
  }
  delay(3000);
}
```

5

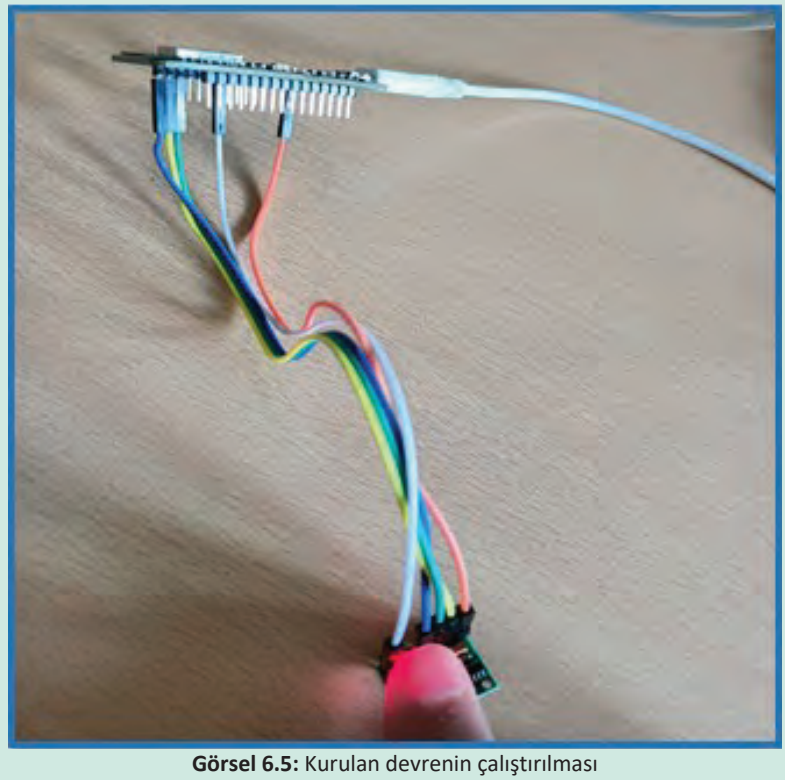
Pox.update ile sensörlerden gelen veriler yeniden hesaplanır.

Pox.getHeartRate() ile nabız sayısı alınır ve BPM değişkenine aktarılır.

Pox.getSpO2() ile kandaki oksijen miktarı alınır ve SpO2 değişkenine aktarılır.

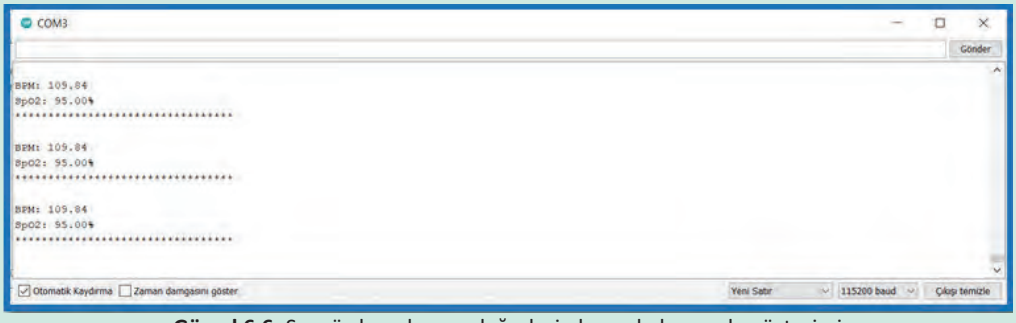
Kod Satırı / Bloku	Açıklama
1	Kalp atış sayısı (getHeartRate) ve kandaki oksijen sayısı (getSpO2) sensörlerden alınır ve konsol ekranına yazdırılır.
2	Localhost'a sensör verilerini göndermek için http servisi açılır.
3	Localhost'a veriler gönderilir. Gönderim sırasında güvenlik kontrolü için apiKeyValue'de kullanılmıştır.
4	Verilerin http tarayıcısına aktarımında bir sorun olup olmadığı kontrol edilir. Sorun yoksa toplanan veriler ekranda yazdırılır, sorun varsa ResponseCode (error code) ekrana yazdırılır.
5	Wi-Fi bağlantısı kapatılır.

6. Adım : Programı çalıştırarak test ediniz. Parmağınızı sensör üzerine yerleştiriniz (Görsel 6.5).



Görsel 6.5: Kurulan devrenin çalıştırılması

7. Adım : Toplanan verileri Arduino.cc konsolunda görüntüleyiniz (Görsel 6.6).



Görsel 6.6: Sensörden okunan değerlerin konsol ekranında gösterimi

8. Adım : Sersör verilerini http üzerindeki localhost'a aktarınız. Uygulamada NodeMCU 192.168.7.217 IP adresini almıştır. Bu IP adresini web tarayıcınıza yazınız ve tarayıcınızı çalıştırınız. Veriler Görsel 6.7'deki gibi görüntülenecektir.

Görsel 6.7: Sensör verilerinin web tarayıcıda görüntülenmesi

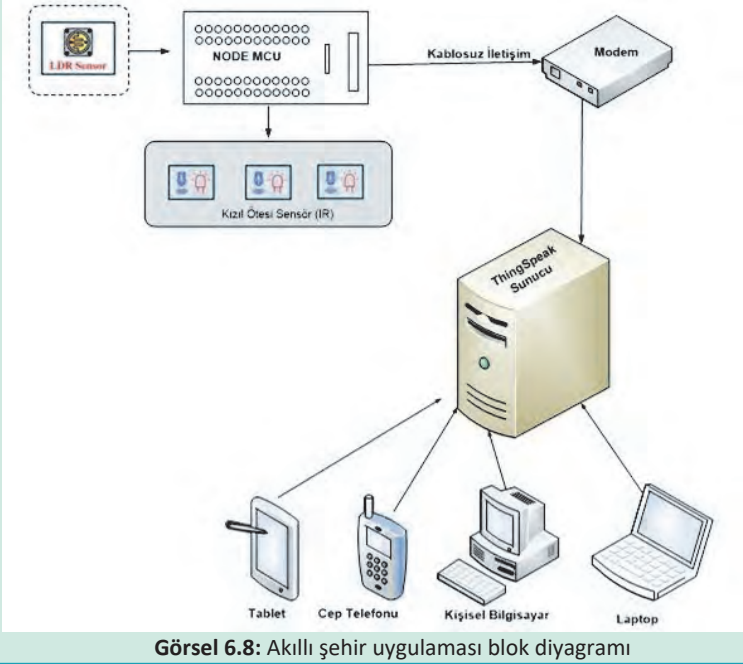
2. UYGULAMA

Trafik kontrolü sağlayan bir IoT uygulaması

Şehirlerin bütçelerinin en fazla harcandığı unsurlardan biri aydınlatmadır. Aydınlatma sistemlerini oluşturan enerji ve bakım maliyetleri oldukça yüksektir. IR sensörü, Işık Yoğunluğu sensörü ve NodeMCU (ESP-8266) kartı kullanarak Görsel 6.8'de blok şeması gösterilen IoT tabanlı sistem uygulamasını yapınız.

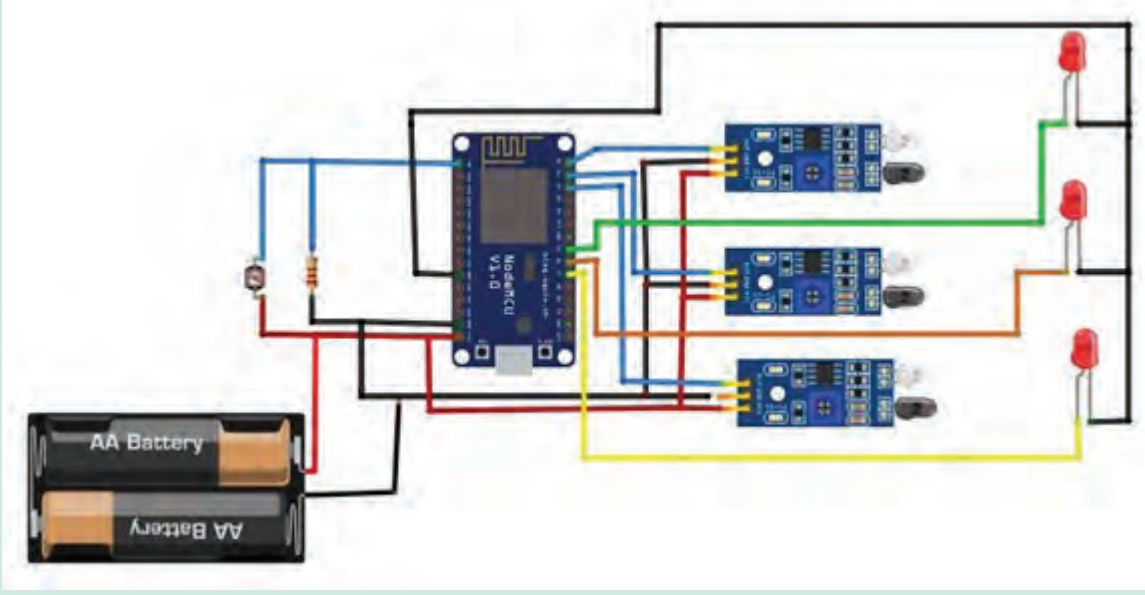
Gerekli Malzemeler

- NodeMCU (esp8266)
- IR Sensörler
- LDR Sensörü
- LED'ler
- Bord



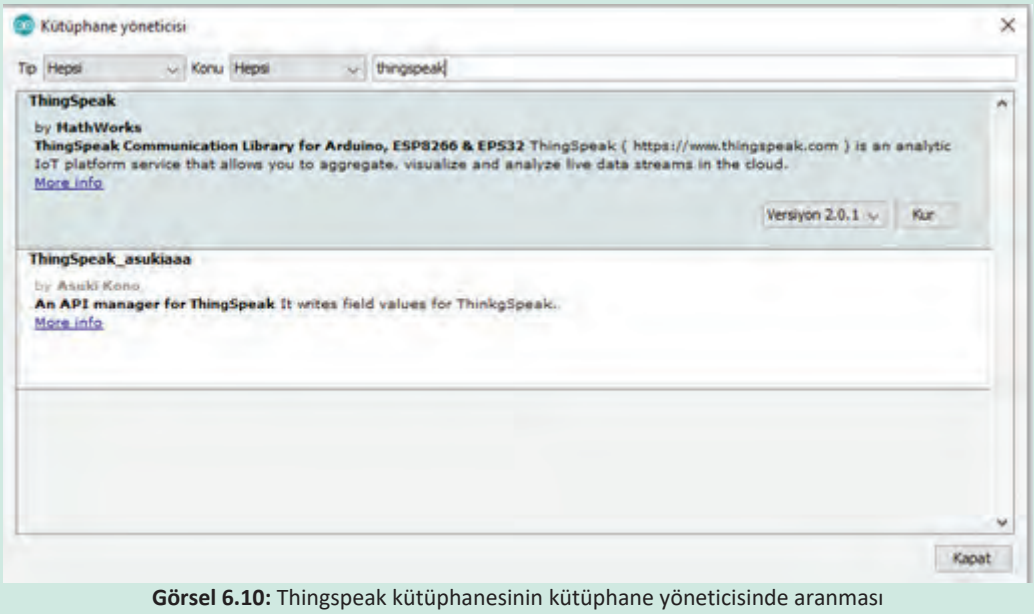
Görsel 6.8: Akıllı şehir uygulaması blok diyagramı

1. Adım : Görsel 6.9'da verilen devre şemasını bord üzerinde kurunuz.



Görsel 6.9: Akıllı şehir uygulaması devre bağlantı şeması

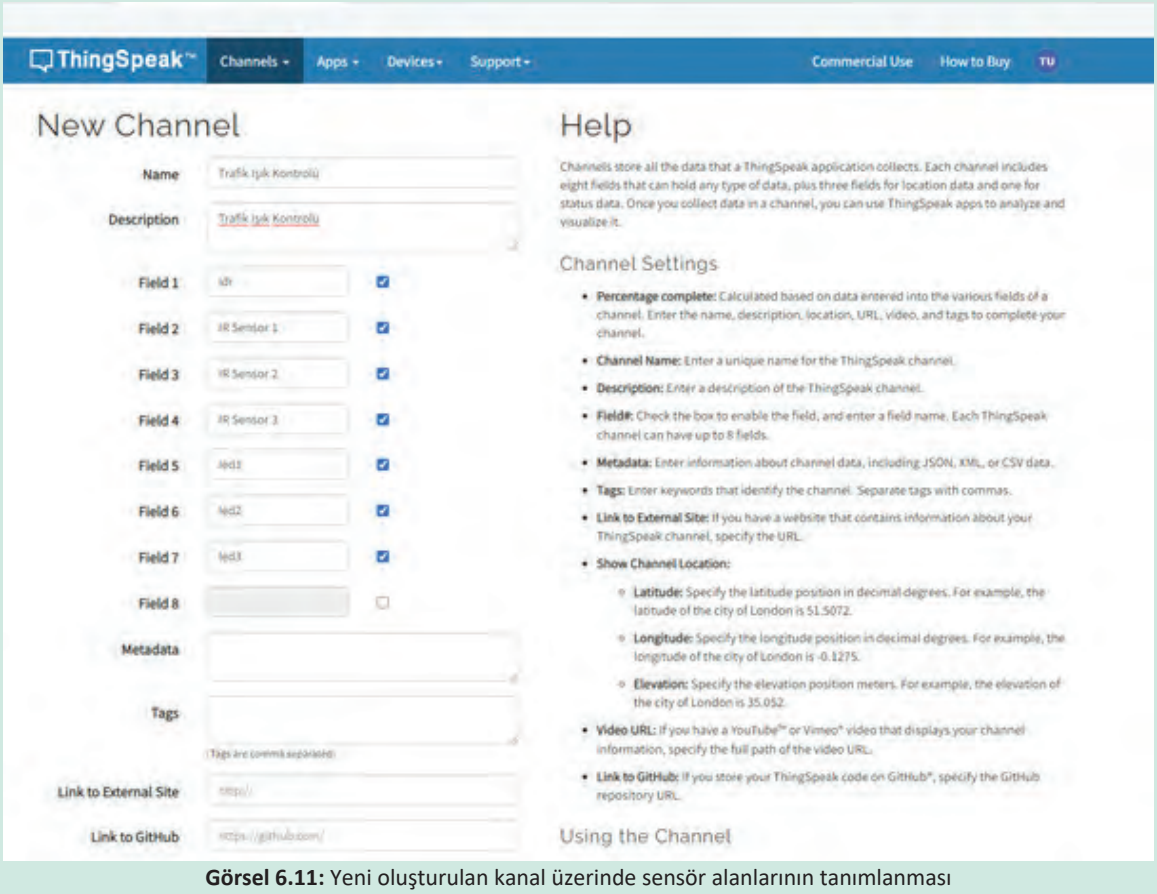
2. Adım : Arduino IDE editörünü açınız ve menüden kütüphane yöneticisi sekmesini açınız. Uygulamada Thingspeak (sensör verilerinin gönderileceği web sunucusu) kullanılacağı için kütüphanenize MathWorks tarafından geliştirilen ThingSpeak kütüphanesini ekleyiniz ve kütüphanenin kurulumunu yapınız (Görsel 6.10).



Görsel 6.10: Thingspeak kütüphanesinin kütüphane yöneticisinde aranması

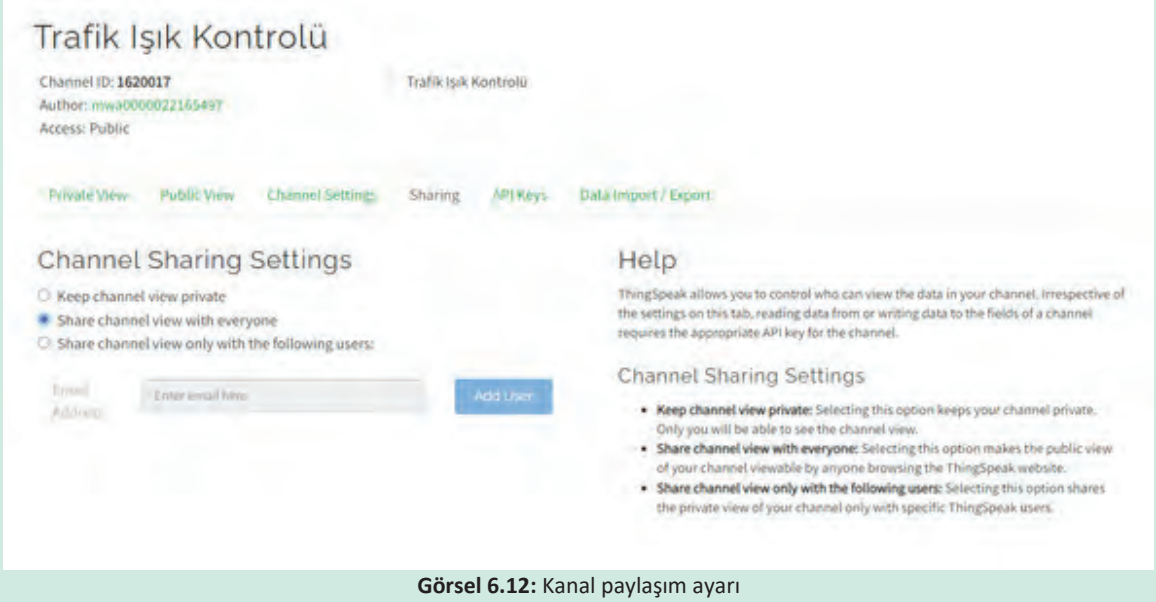
3. Adım : LDR sensörünü, ışık sensörünü ve LED verilerini ThingSpeak yüklemek için https://thingspeak.com/ adresine girerek bir hesap oluşturunuz.

4. Adım : Giriş yaptıktan sonra Channels sekmesinde "New Channel" seçeneğini tıklayarak yeni bir kanal oluşturunuz. Oluşturulan kanala ait alanları doldurunuz ve kaydet ile kanalınızı sisteme kaydediniz. Uygulamada 4 sensör kullanılacağı için 4 alana değişken isimleri tanımlayınız (Görsel 6.11).



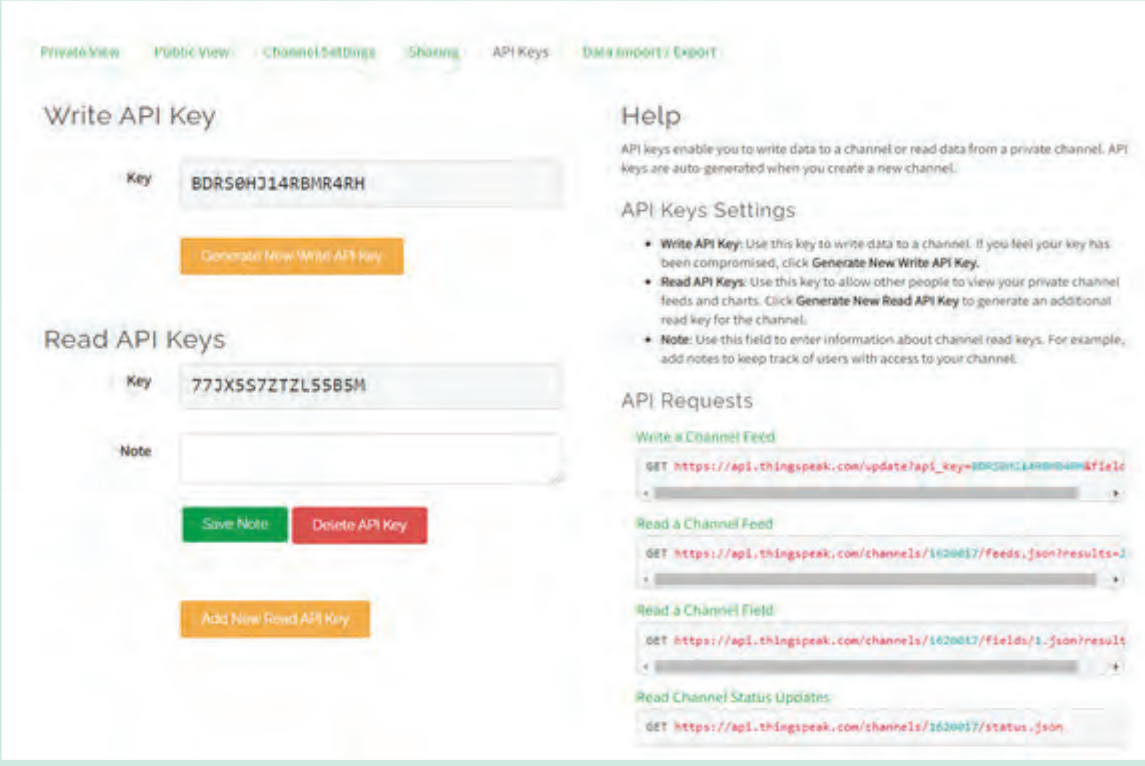
Görsel 6.11: Yeni oluşturulan kanal üzerinde sensör alanlarının tanımlanması

5. Adım : Görsel 6.12'de kanala ait paylaşım alanı gösterilmektedir. Kaydedilen kanalı, kanal görünümünü herkes ile paylaş seçeneğini seçiniz.



Görsel 6.12: Kanal paylaşım ayarı

6. Adım : API Keys seçeneğini tıklayınız. Oluşturmuş olduğunuz kanala ait API Key'ler burada yer alır (Görsel 6.13). Bu API Key'leri bir yere not ediniz. Bu Key bilgileri Arduino kodunda kullanılacaktır.



Görsel 6.13: API Key bilgilerinin okunması

7. Adım : Arduino.cc programını çalıştırınız. Aşağıdaki kütüphaneleri programınıza ekleyiniz ve tanımlamaları yapınız.

```
#include <ESP8266WiFi.h>;
#include <WiFiClient.h>;
#include <ThingSpeak.h>;

const char* ssid = "Senin WİFİ SSID Adı";
const char* sifre = "Senin WİFİ Şifren";
WiFiClient client;
unsigned long KanalNumara= 1620017;
const char * YazmaAPIAnahtari = "BDRS0HJ14RBMR4RH";
const char * OkumaAPIAnahtari = "77JX5S7ZTZL55B5M";
int Led1,Led2,Led3;
int Ir1 = D0;
int Led_1 = D7;
int Ir2 = D1;
int Led_2 = D6;
int Ir3 = D2;
int Led_3 = D5;
int Ldr = A1;
int deger =0;
```

```
void setup() {
  Serial.begin(9600);
  delay(10);
  pinMode(Ir1,INPUT);
  pinMode(Led_1,OUTPUT);
  pinMode(Ir2,INPUT);
  pinMode(Led_2,OUTPUT);
  pinMode(Ir3,INPUT);
  pinMode(Led_3,OUTPUT);
  WiFi.begin(ssid, sifre);
  ThingSpeak.begin(client);
}
```

5

Kod Satırı / Bloku	Açıklama
1	Uygulamanın çalışabilmesi için gerekli kütüphanelerin yüklendiği gerekli kod blokudur.
2	NodeMCU'nun internete bağlanması için ağ adının ve şifresinin girildiği kod blokudur.
3	Thingspeak'tan alınan API Key bilgilerinin uygulamaya tanımlandığı satırlardır.
4	LED'ler için tanımlamalar yapılmıştır.
5	Bağlantı hız ayarı yapıldıktan sonra NodeMCU üzerinde hangi sensörün ve LED'in hangi portlara bağlandığının tanımlandığı kod blokudur.

```
void loop() {
  int S1 = digitalRead(Ir1);
  int S2 = digitalRead(Ir2);
  int S3 = digitalRead(Ir3);
  s3 = not(s3);
  deger= analogRead(Ldr);
  if(deger<850)
  {
    if(S1==0)
    {
      digitalWrite(Led_1,LOW);
    }
    else
    {
      digitalWrite(Led_1,HIGH);
    }
    if(S2==0)
    {
      digitalWrite(Led_2,LOW);
    }
    else
```

1

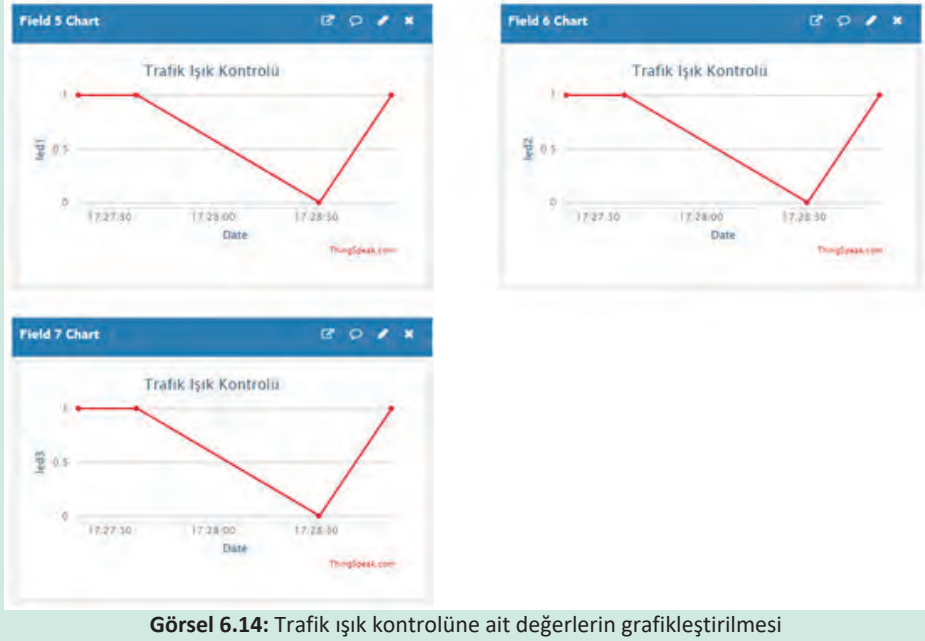
2

```
    {
      digitalWrite(Led_2,HIGH);
    }

    if(S3==0)
    {
      digitalWrite(Led_3,LOW);
    }
    else
    {
      digitalWrite(Led_3,HIGH);
    }
  }
  else
  {
    digitalWrite(Led_1,LOW);
    digitalWrite(Led_2,LOW);
    digitalWrite(Led_3,LOW);
  }
```

3

Kod Satırı / Bloku	Açıklama
1	Okunan sensör verileri S1, S2 ve S3 adlı değişkenlere aktarılır.
2	Okunan değer 0 (sıfır)'a eşitse LED kapalı, değilse LED açık yapılır.
3	Eğer LDR sensöründen okunan değer 850'den büyükse tüm LED'ler kapalı konumda tutulur.

8. Adım : Programı çalıştırarak test ediniz. https://thingspeak.com/ adresinde Private View veya Public View sekmesine tıklayarak trafik ışık kontrolüne ait değerleri görebilirsiniz (Görsel 6.14).



Görsel 6.14: Trafik ışık kontrolüne ait değerlerin grafikleştirilmesi

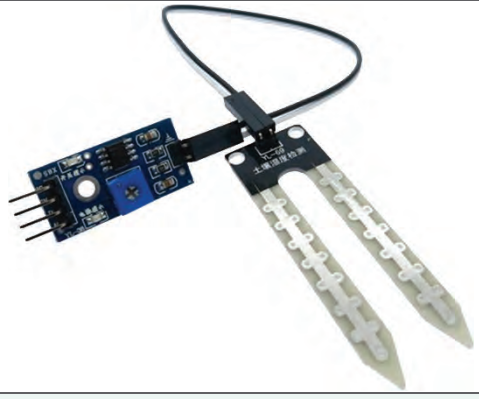
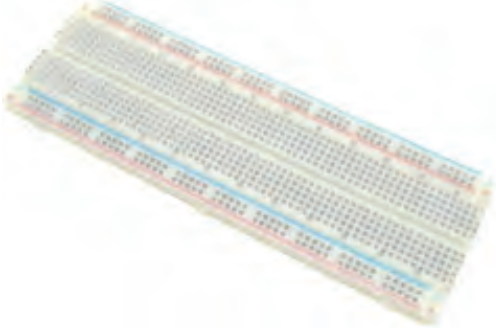
3. UYGULAMA

Arduinoblock ile Tarım Alanında Bir IOT Projesi Gerçekleştirme

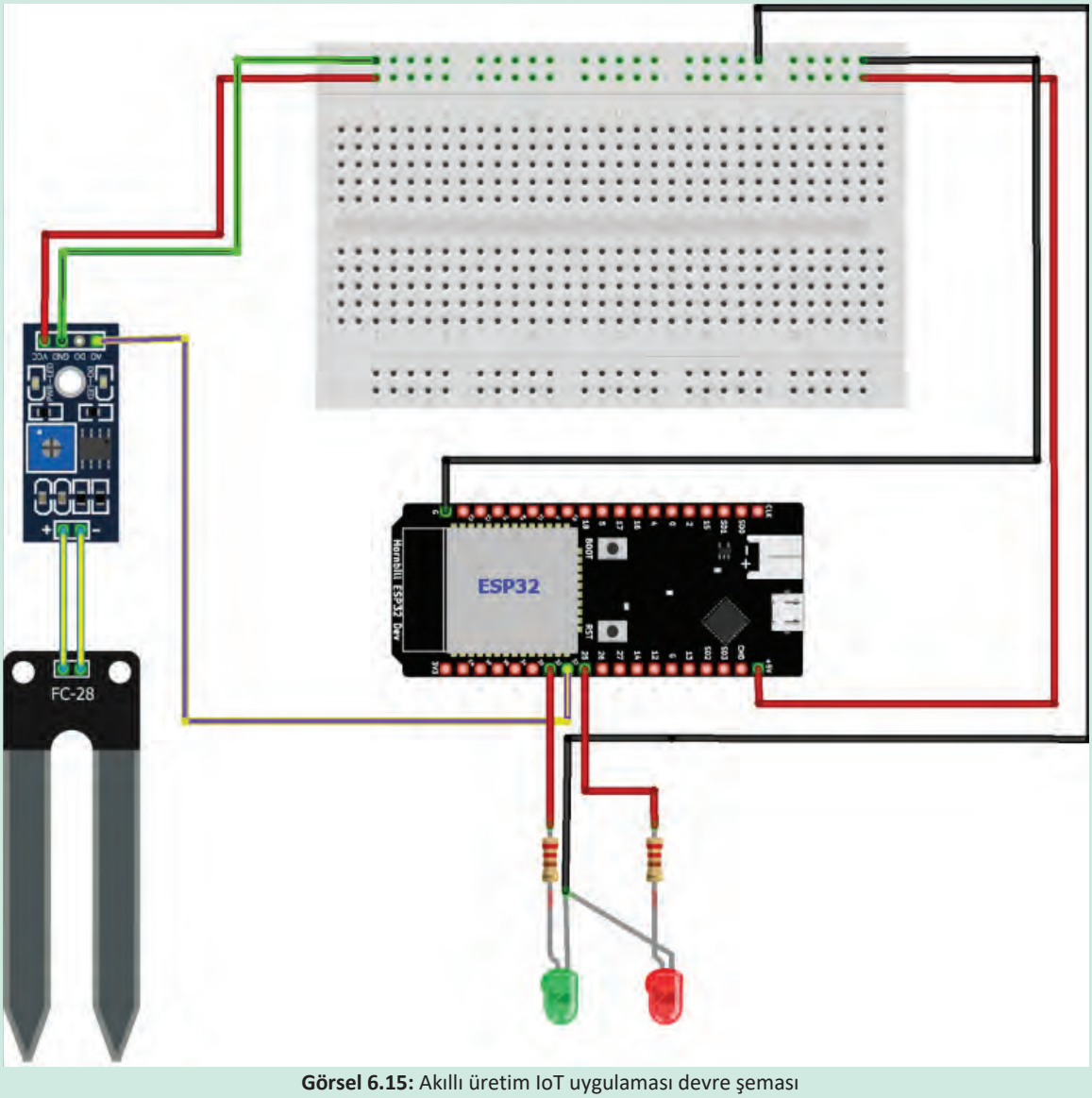
Aşağıdaki işlem adımlarına göre Arduinoblock programını kullanarak topraktaki nem miktarının sensörler aracılığı ile uzaktan takibini yapınız. Toprağın sahip olduğu nem miktarına göre uyarı vermesini ve sulama kanallarının açılmasını sağlayan projeyi gerçekleştiriniz.

1. Adım : Çalışmanın yapılması için ESP32 kartını, toprak nem sensörünü ve diğer parçaları hazır hâle getiriniz (Tablo 6.2).

Tablo 6.2: Akıllı Üretim IoT Uygulaması İçin Gerekli Malzemeler

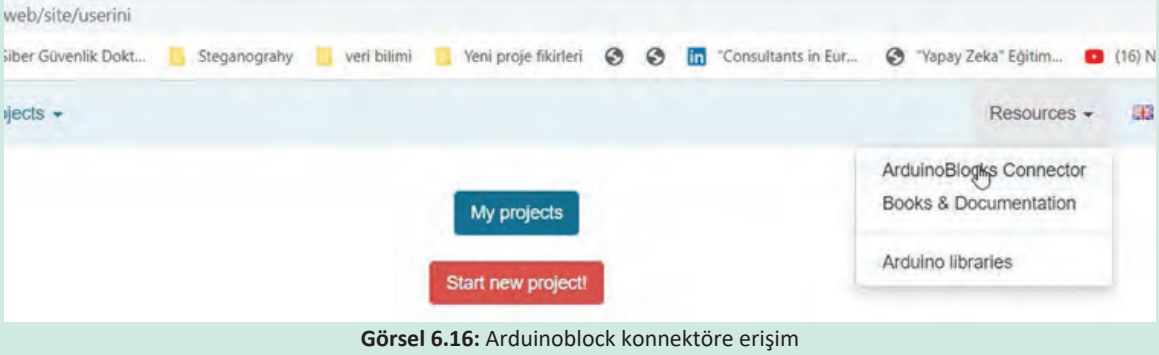
1. ESP32	**4. Toprak Nem Sensörü**
	
2. Breadboard	**5. LED**
	
3. Kablo	

2. Adım : Hazırlanacak uygulama için devre şemasını hazır hâle getiriniz (Görsel 6.15).



Görsel 6.15: Akıllı üretim IoT uygulaması devre şeması

3. Adım : Arduinoblock programının web sitesine giriş yaptıktan sonra program ile programlama kartı arasında iletişimi sağlayacak olan bir program yükleyiniz. Bunun için sağ üst tarafta bulunan "Resources" menüsüne giriniz ve "ArduinoBlocks Connector" sekmesine tıklayınız (Görsel 6.16).



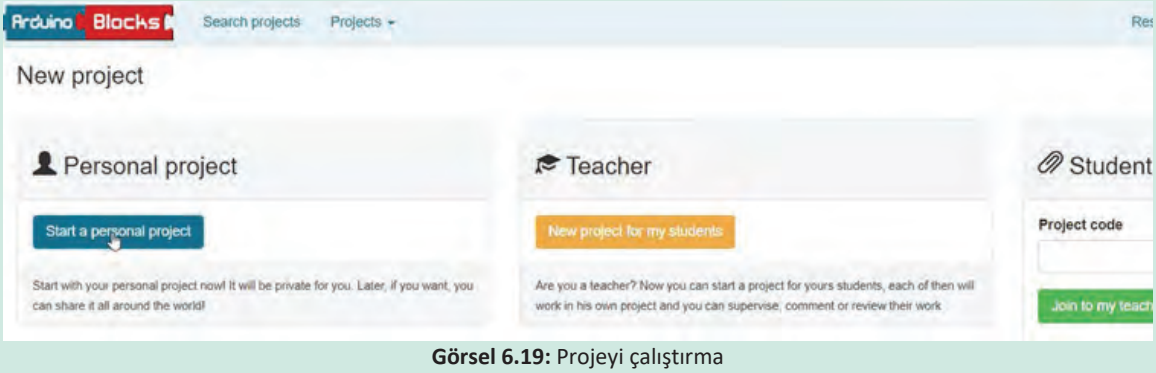
Görsel 6.16: Arduinoblock konnektöre erişim

4. Adım : Kullanılan işletim sistemine göre uygun olan sürücüyü indiriniz (Görsel 6.17).

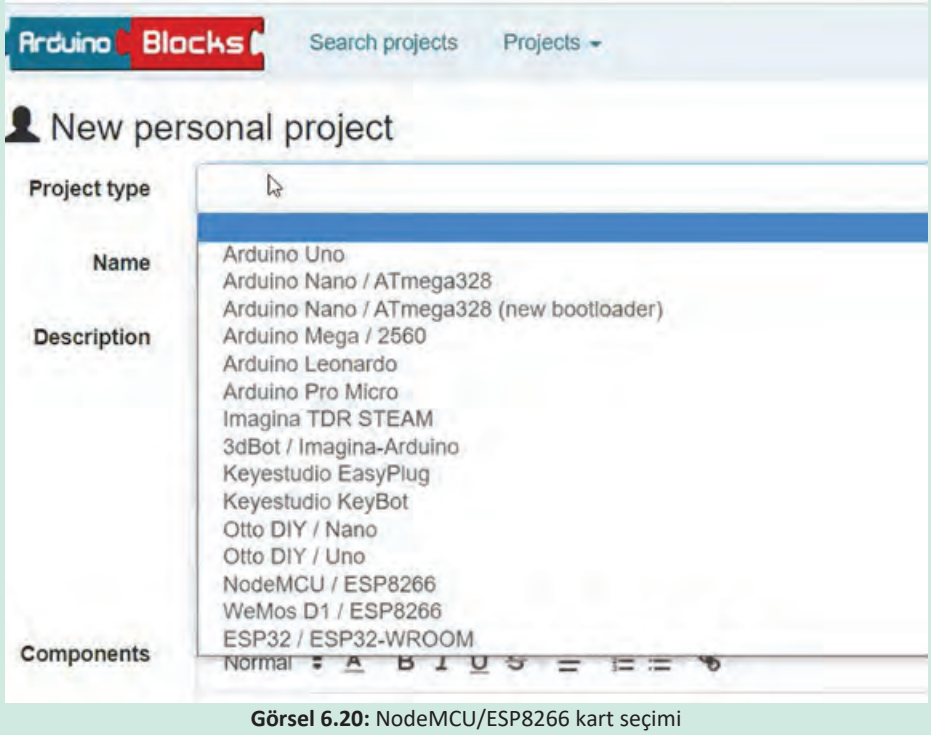
Görsel 6.17: Uygun sürücü indirme

5. Adım : İndirme işlemi tamamlanınca "Project" menüsünden "New Project" sekmesini seçiniz (Görsel 6.18). Açılan pencereden yeni bir proje başlatmak için "Start personel project" butonuna basınız (Görsel 6.19). Açılan pencereden kullanılacak devre kartını seçiniz. Uygulamada NodeMCU kullanılacağı için NodeMCU / ESP8266'yı işaretleyiniz. Name kısmına ve Description kısmına projeye özel bilgiler veriniz (Görsel 6.20). Daha sonra sayfanın altında bulunan "New Project" butonuna basarak proje oluşturunuz.

Görsel 6.18: Yeni proje oluşturma

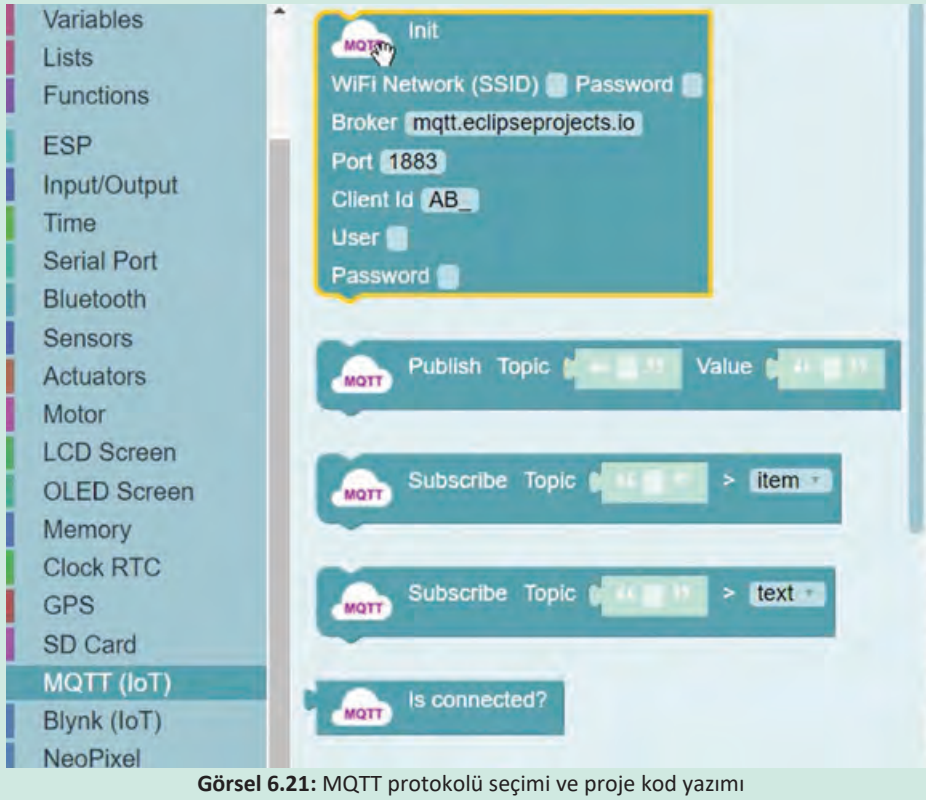


Görsel 6.19: Projeyi çalıştırma



Görsel 6.20: NodeMCU/ESP8266 kart seçimi

6. Adım : Program için öncelikle iletişim protokolünü seçerek başlatınız. Bunun için sol taraftaki menüden "MQTT (IOT)" seçiniz ve yan tarafta açılan SSID içeren kutu proje ekranına sürükle bırak yöntemiyle yerleştiriniz. Bu bloku Setup içine yerleştiriniz (Görsel 6.21).

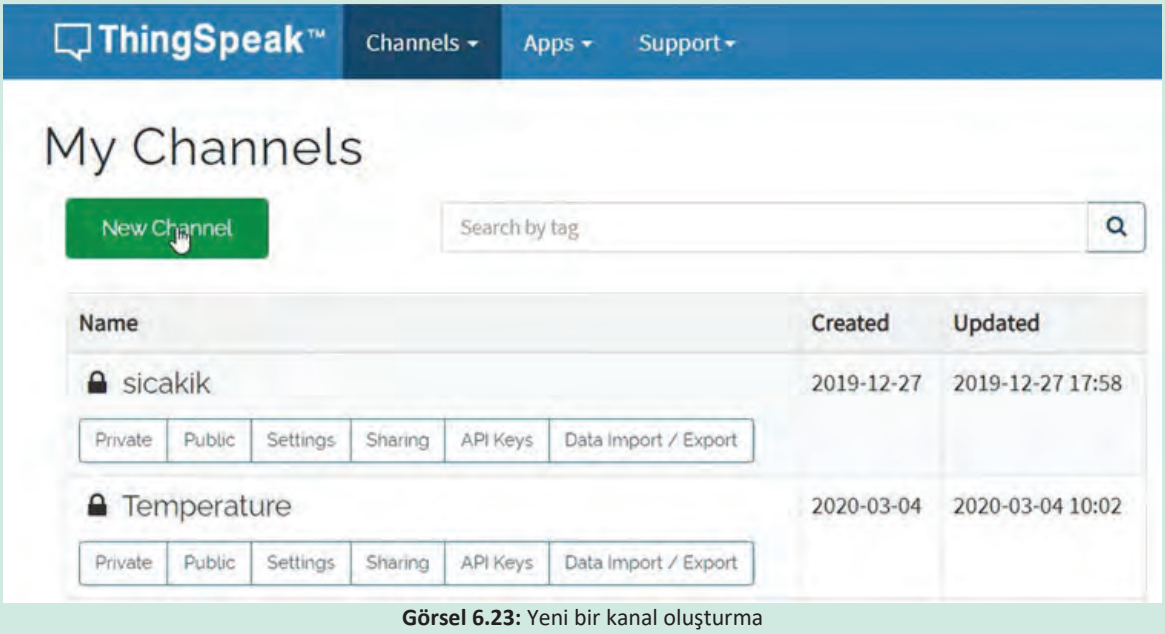


Görsel 6.21: MQTT protokolü seçimi ve proje kod yazımı

7. Adım : Bu blok bilgilerin gönderileceği uzak sunucu ayarlarının da yapıldığı kısımdır. Bu yüzden bilgilerin gönderileceği web sitesine ait bilgileri buraya giriniz. Uygulamada www.Thingspeak.com sunucusuna bilgiler gönderileceği için Thingspeak web sitesinin sağladığı bilgileri buraya giriniz. Bunun için Thingspeak sitesine gidip üye olunuz. Daha sonra bir kanal açıp sensörden toplanan verileri bu kanala aktarınız. Thikspeak sitesine üye olup giriş işlemini yapınız. Daha sonra ise "Channels" menüsünden "My Channels" sekmesine tıklayarak kendinize ait kanal oluşturunuz (Görsel 6.22).

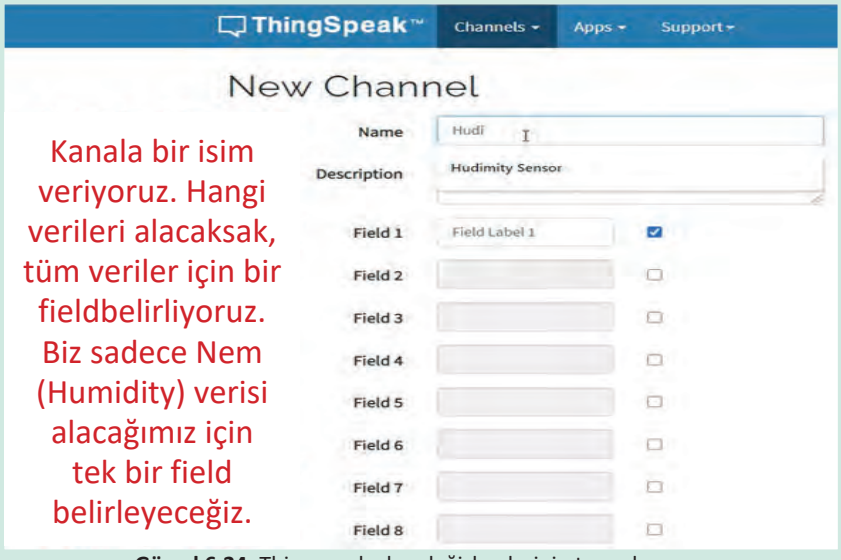
Görsel 6.22: Kişiye özel kanal listesine ulaşım

8. Adım : İhtiyaç olan yeni bir kanal için "New Channel" butonuna basınız (Görsel 6.23).



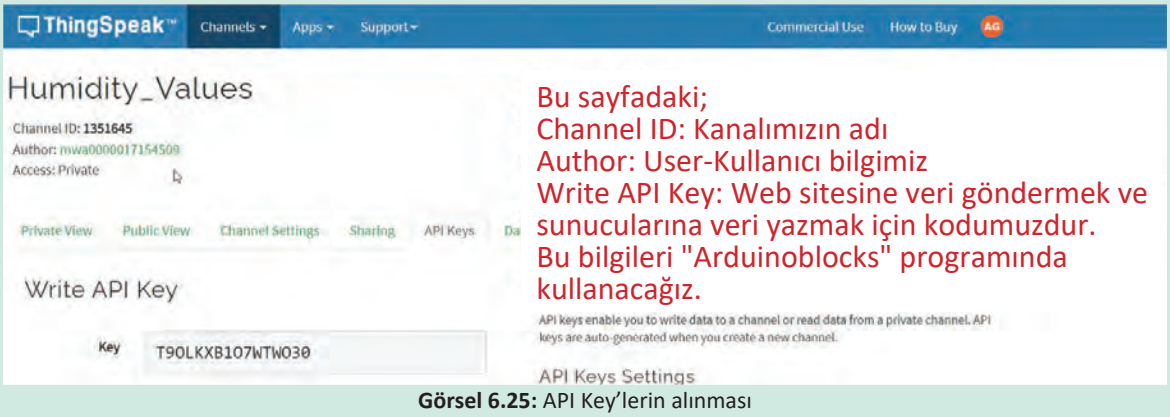
Görsel 6.23: Yeni bir kanal oluşturma

Oluşturulan yeni kanala bir isim veriniz ve sensörlerden elde edilen verilerin aktarılması için değişken alanları tanımlayınız (Görsel 6.24).

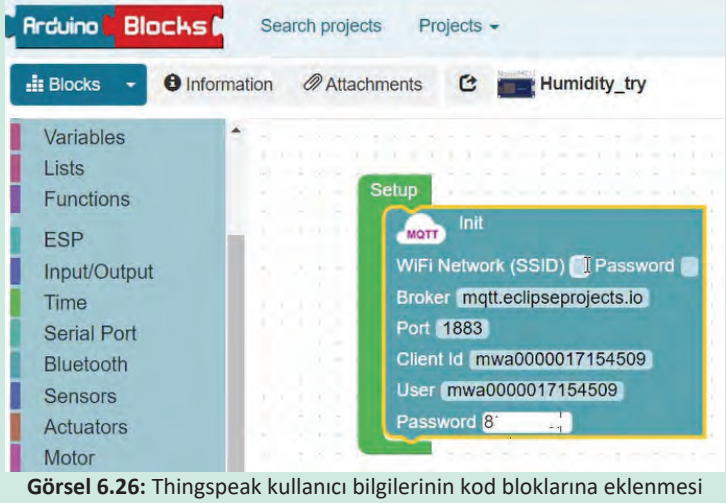


Görsel 6.24: Thingspeak alan değişkenlerinin tanımlanması

9. Adım : API Keys sekmesine girerek "Write API Key" kodunu, "Channel ID" kodunu ve "Author kodunu" not alınız (Görsel 6.25). Bu bilgileri Arduinoblocks kodlarınıza yerleştiriniz (Görsel 6.26).

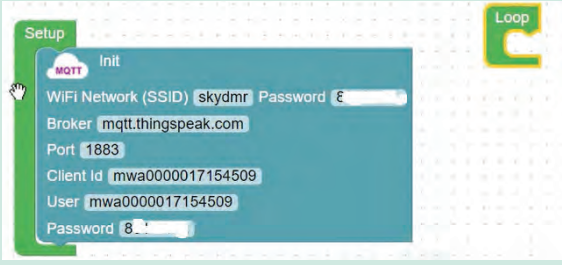


Görsel 6.25: API Key'lerin alınması



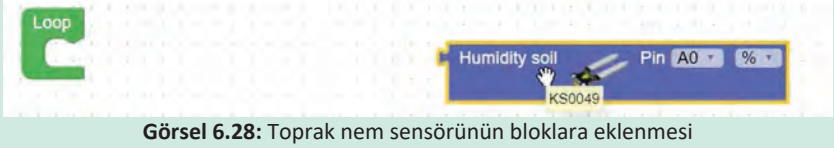
Görsel 6.26: Thingspeak kullanıcı bilgilerinin kod bloklarına eklenmesi

10. Adım : Bağlanılacak kablosuz ağ adını (SSID adını) ve ağ şifresini giriniz (Görsel 6.27).



Görsel 6.27: SSID ve parolanın kod bloklarına eklenmesi

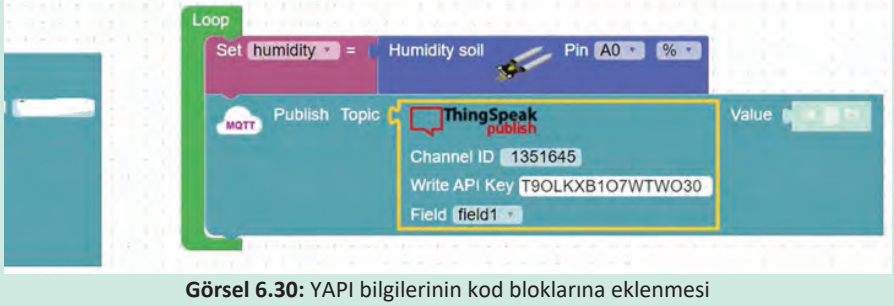
11. Adım : Toprak nem sensörünü "Sensörler" başlığından seçiniz (Görsel 6.28).



Görsel 6.28: Toprak nem sensörünün bloklara eklenmesi

12. Adım : Sensörden toplanan değerler bir değişkene aktarılacağı için değişken tanımlaması yapınız (Görsel 6.29). API Key kodlarını komutlara ekleyiniz. Field kısmına Channel oluştururken verdiğiniz ismi yazınız (Görsel 6.30). Value kısmına ise bir üst satırda kullanılan değişkenden gelen bilgiler gönderileceği için "humidity" yazınız (Görsel 6.31). Bu sayede toplanan verileri değişkene, değişkenden de oluşturulan kanaldaki field alanına aktarınız.

Görsel 6.29: Yeni değişken tanımlaması



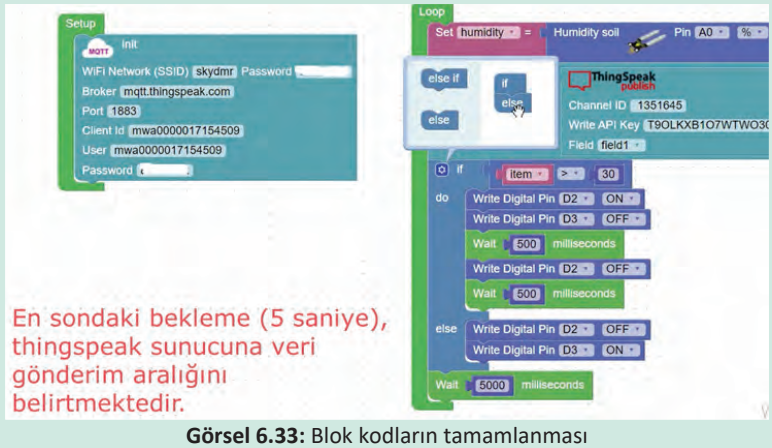
Görsel 6.30: YAPI bilgilerinin kod bloklarına eklenmesi

Görsel 6.31: Sensörden gelen verinin değişkene atanması

13. Adım : Wait kısmındaki süre milisaniye olarak verilmiştir. Yani bir saniye 1000 milisaniyedir. İlgili süreyi bloklara ekleyiniz ve nem miktarının istenilen koşul durumuna göre dijital pinlere göndereceği sinyalleri ayarlayınız (Görsel 6.32).

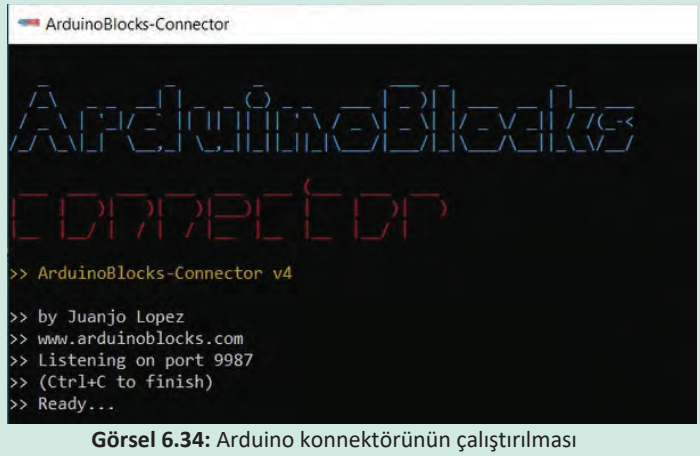
Görsel 6.32: Dijital pinlerin ON-OFF pozisyonuna geçmesi

14. Adım : Kod bloklarının Görsel 6.33'teki gibi olup olmadığını kontrol ediniz.



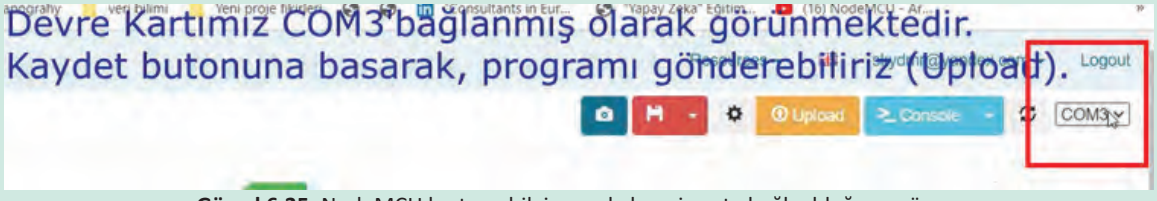
Görsel 6.33: Blok kodların tamamlanması

15. Adım : Kodlar tamamlandıktan sonra indirilen Arduinoblock Connector programını çalıştırınız (Görsel 6.34). Ready ifadesi yazınca yazılan kodların NodeMCU kartına gönderimini sağlayınız.

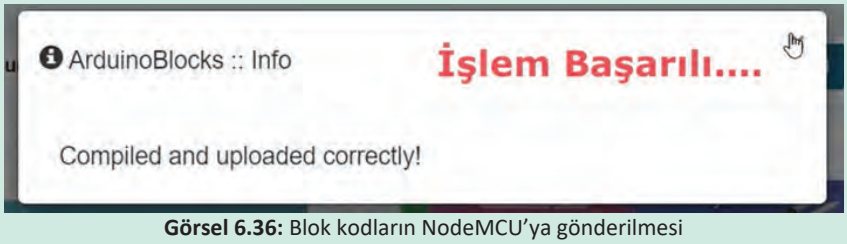


Görsel 6.34: Arduino konnektörünün çalıştırılması

16. Adım : Arduinoblocks web sitesine dönünüz ve NodeMCU kartımızın COM3 portuna bağlı olduğunu görünüz (Görsel 6.35). Projenizi kaydedip "Upload" butonuna basınız ve programınızı COM3 portuna bağlı olan NodeMCU kartına gönderiniz (Görsel 6.36).



Görsel 6.35: NodeMCU kartının bilgisayarda hangi porta bağlı olduğunu görme

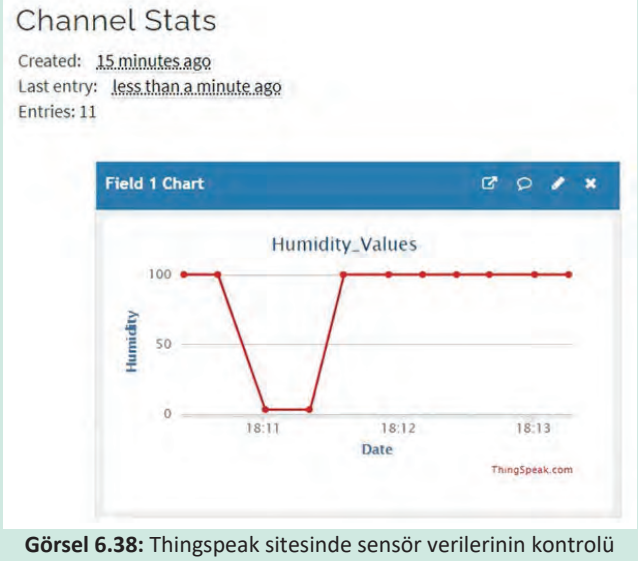


Görsel 6.36: Blok kodların NodeMCU'ya gönderilmesi

17. Adım : Thingspeak web sitesine giderek sensörün gönderdiği verilerin kontrolünü yapınız (Görsel 6.37).

Görsel 6.37: Sensör verilerinin kontrolü

18. Adım : Sonuç ekranında belirtilen sürelerde verilerin geldiği görülmektedir. Gerekli kontrolleri yaparak projeyi sonlandırınız (Görsel 6.38).



Görsel 6.38: Thingspeak sitesinde sensör verilerinin kontrolü

SIRA SİZDE

Eğitim alanında kullanılabilecek bir IoT nesnesi tasarlayarak hazırlayınız ve sınıfta sunumunu yapınız.

6.4. IoT'TA MAKİNE ÖĞRENMESİ VE YAPAY ZEKÂ

Nesnelerin İnterneti (IoT) teknolojisi, çeşitli cihazlar arasında bilgi alışverişi yapmak için çeşitli uygulamalarda kullanılır. IoT uygulamaları için güvenlik açığı ana sorunlardan biridir. Akıllı IoT tabanlı sistemler genellikle mobil cihazlarda kullanılır. Bu cihazlarda ise Android işletim sistemi kullanır. IoT'in ana sorunu olan güvenlik açığının üstesinden gelmek için makine öğrenimi algoritmaları kullanılır.

IoT ağlar yeni riskler oluşturur. Endişe verici boyutlara ulaşan bu riskler IoT ağlarda güvenlik, gizlilik ve enerji gibi bazı önemli sorunlara sebep olur. IoT ağlarda iletilen paketler bir dizi saldırılara neden olur. Bu saldırılara Sahte Kimlik Saldırısı ve Dağıtılmış Hizmet Reddi (DDoS) Saldırıları vb. örnek olarak verilebilir. Bu saldırıların geneli düğümlerin enerji seviyesini ve işlem kapasitesini olumsuz etkiler. IoT güvenliğinde saldırıları tespit etmek için birçok saldırı tespit yöntemi kullanılsa da yenilikçi ve enerji korunumlu yöntemler kullanılmalıdır. Yapay Zekâ ve Makine Öğrenmesi teknikleri kullanılarak IoT güvenliğinde saldırıların tespiti yapılır. Makine öğrenme tekniklerinden Vektör Makinesi ile DIS Flooding saldırılarının tespit edilmesi örnek olarak verilebilir.

Bazı IoT sistemleri basit olarak tasarlanmış IoT sistemlerine göre daha karmaşık ve kontrol zorluğu içerebilir. Ortam ısısına veya ışık şiddetine bağlı olarak fanların dönüş hızını ayarlayan veya lambaları açma-kapama gibi basit uygulamalara karşılık daha uygun bir eylemi başlatmak için verilerin yorumlamasını gerektiren durumlar da olabilir. Bu gibi durumlarda IoT ile beraber yapay zekânın beraber incelendiği Nesnelerin Yapay Zekâsı (AIoT) yapısı meydana gelir. AIoT, IoT'yi oluşturan her bir cihazın verileri anlaması ve IoT çevresini gözlemesi sonucunda ne yapacağına karar vermesi gibi durumları içerir. İnsan müdahalesini en aza indirmek ve karmaşık veri analizlerini yapay zekâ teknikleri kullanarak çözmek için günümüzde kullanım alanı oldukça geniştir. Örneğin trafikteki yol durumunu, araç sayısını, trafik hızını, hava durumunu (yağmurlu, karlı, rüzgârlı veya güneşli) sensörlerden gelen verilerle analiz ederek trafik kontrolünü sağlayan IoT sistemleri mevcuttur. Benzer bir örnek evler için de verilebilir. IoT sistemi; ev aletlerinden, aydınlatmadan, elektronik cihazlardan ve daha fazla çevresel faktörlerden yararlanarak ev sahibinin alışkanlıklarını öğrenip daha fazla verim ve kazanç odaklı otomatikleştirilmiş destek geliştirebilir. Makine öğrenmesi veya derin öğrenme gibi yapay zekâ tekniklerini kullanan IoT sistemleri, yeşil enerjiyi destekleyerek ve karbon salınımlarını en aza indirerek gelecek nesiller için yaşanabilir bir dünya bırakmaya yardımcı olmaktadır.

ARAŞTIRMA

IoT çevrelerinde yapay zekâ tekniklerinden faydalanarak geliştirilmiş sistemleri araştırınız. Bir örnek olarak Türkiye'nin Otomobili projesinde ve İnsansız Hava Araçları projelerinde IoT sistemlerinin kullanılıp kullanılmadığını ve bu sistemlerde makine öğrenmesi veya yapay zekâ kullanılıp kullanılmadığını araştırınız. Araştırmanızı bir rapor hâline getiriniz ve sınıf arkadaşlarınıza sunumunu yapınız.

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde parantez içine yargılar doğru ise "D", yanlış ise "Y" yazınız.

1. (.....) IoT cihazları sensörler aracılığı ile çevredeki verileri algılar.
2. (.....) Endüstride IoT projelerinin başlangıcı olarak RFID uygulanmış nesneler kabul edilmektedir.
3. (.....) Blok kodlar kullanarak IoT projesi geliştirmek mümkündür.
4. (.....) Sağlık sektöründe IoT uygulaması geliştirmek mümkündür.

B) Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

5. Aşağıdaki kavramlardan hangisi endüstride IOT projesi geliştirmeye yardımcı olur?

A) Caen Abbel
B) Crunch
C) John the Ripper
D) Medusa
E) Scada

6. Thinspeak kullanılan projeler sisteme hangi yöntemle entegre edilir?

A) API key
B) Arduinoblocks
C) Hash Code
D) IOT Thing
E) Keylog Thing

7. Aşağıdaki araçlardan hangisi IOT güvenlik ilkelerinden biridir?

A) IoT Test İlkesi
B) Nesne Kontrol İlkesi
C) Yerel Güvenlik İlkeleri
D) Antivirüs Kontrol İlkesi
E) Güvenli Bağlantı Oluşturma İlkesi

IoT ÇÖZÜMLERİ GELİŞTİRME

7. Öğrenme Birimi

KONULAR

7.1. IoT UYGULAMA TASARLAMA
7.2. IoT PROTOTİPİ

NELER ÖĞRENECEKSİNİZ?

- IoT projesinde prototip oluşturma
- IoT ile sorunlara çözüm geliştirme
- IoT bulut işlemleri yapma
- Yazılım üzerinden IoT sistemlerini kontrol etme
- IoT'ta kullanılan prototip geliştirme platformları

TEMEL KAVRAMLAR

akıllı sistemler, çözüm geliştirme, DHT11, ESP8266, IoT bulut, NodeMCU, prototip, sensör, silo

HAZIRLIK ÇALIŞMALARI

1. Prototipi geliştirilmiş bir aygıta veya ürüne bir örnek veriniz.
2. Nesnelerin interneti çalışmalarında prototip geliştirmenin avantajları neler olabilir?

http://kitap.eba.gov.tr/KodSor.php?KOD=14400

7.1. IoT UYGULAMA TASARLAMA

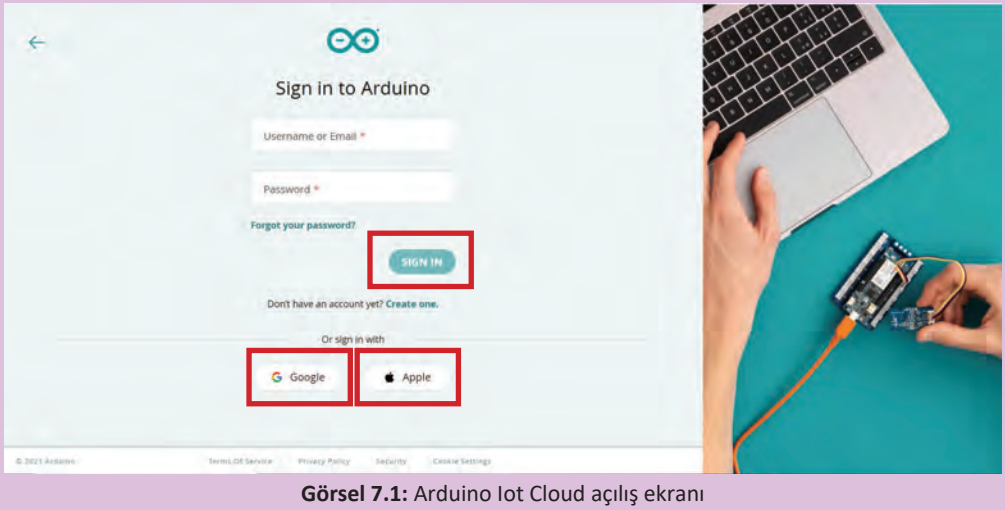
IOT cihazlarının hepsinin ortak özelliği internet üzerinden kontrol edilebilmesi veya internete veri göndermesi prensibine dayalı olmasıdır. İlk olarak bu öğrenme biriminde evdeki odalarda yer alan aydınlatmaların dünyanın herhangi bir yerinden nasıl kontrol edilebileceği anlatılacaktır.

Evdeki donanımı internet üzerinden kontrol edebilmek için sunucu hizmeti sunan bir alt yapı gereklidir. Bu öğrenme biriminde Arduino Iot Cloud kullanılacaktır. Kullanımı son derece basit olup Google hesabı ile kullanılmaya başlanabilir.

1. UYGULAMA

Arduino Iot Cloud sitesinden hesap açıp IoT nesnesi için gerekli olan temel kurulum işlemlerini gerçekleştiriniz. Bu işlemi evinizdeki bir odanın aydınlatmasını hem web hem de cep telefonu üzerinden açıp kapayabilecek şekilde kurgulayınız.

1. Adım : *https://create.arduino.cc/iot* sitesine gidiniz. Karşınıza Görsel 7.1'deki açılış ekranı gelir. Bu ekranda Arduino hesabınız var ise kullanıcı adınız veya mail adresiniz ile birlikte şifre bilgilerinizi girerek "OTURUM AÇ" butonuna tıklayınız. Eğer Arduino hesabınız yoksa ve oluşturmak istiyorsanız "Bir tane oluşturun." seçeneğine tıklayarak oluşturabilirsiniz. Google veya Apple hesabınız var ise direkt bu bilgiler ile de giriş yapabilirsiniz.

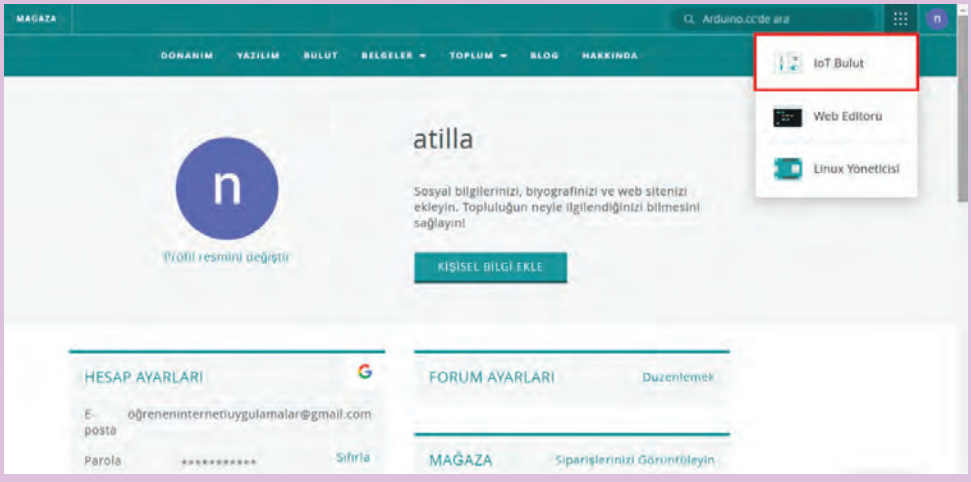


Görsel 7.1: Arduino Iot Cloud açılış ekranı

2. Adım : Giriş işlemini gerçekleştirdikten sonra Google veya Apple seçeneğini seçtiyseniz açılan sayfada (Görsel 7.2) kırmızı çerçeve içindeki alana kullanıcı adınız, onay kutularından ise gizlilik ilkesi ve hizmet şartlarını okuduğunuzu belirten kutuyu onaylayıp "HESAP OLUŞTUR" seçeneğine basınız.

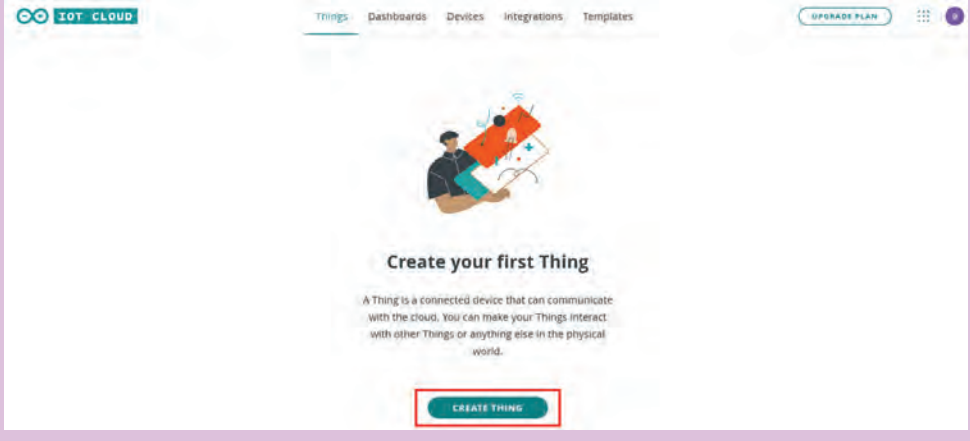
Görsel 7.2: Arduino Iot Cloud sözleşme ekranı

3. Adım : Açılan sayfada (Görsel 7.3) size ait bir profil ekranı olup burdaki kişisel bilgilerinizi, gizlilik ve güvenlik bilgilerinizi güncelleyebilirsiniz. Sol üst köşedeki "IoT Bulut" düğmesine basarak ilk projenizi oluşturmaya başlayabilirsiniz.



Görsel 7.3: Arduino Iot Cloud profil ekranı

4. Adım : Açılan sayfada (Görsel 7.4) "CREATE THING" düğmesine basarak ilk nesnenizi tasarlamaya başlayabilirsiniz.



Görsel 7.4: Arduino Iot Cloud'da nesne oluşturma

5. Adım : Oluşturulan nesneye değişken ekleyiniz (Görsel 7.5).

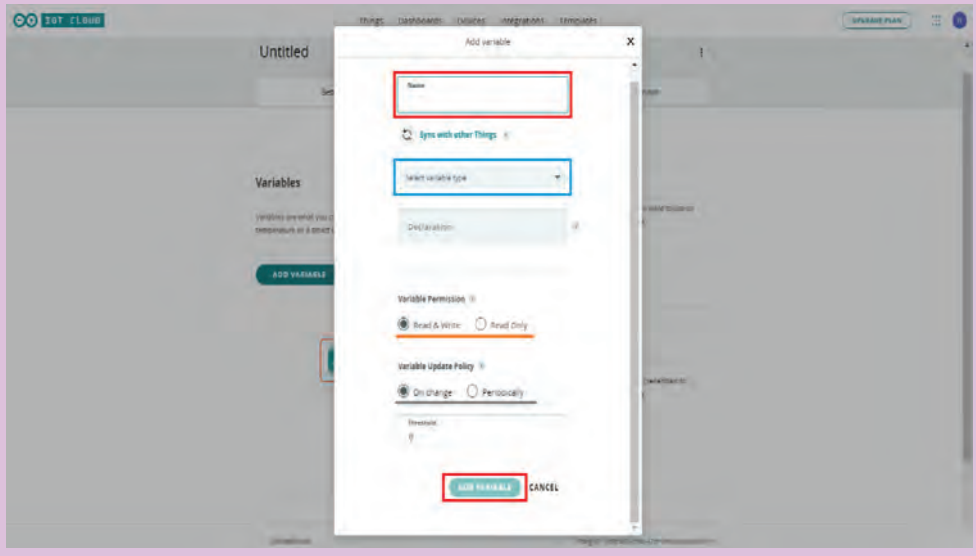
Görsel 7.5: Arduino Iot Cloud'da oluşturulan nesneye değişken tanımlama

6. Adım : Açılan sayfada Görsel 7.6'daki sayfayı göreceksiniz. Burada değişkene verilecek olan adı ve değişken tipini tanımlandıktan sonra değişken izinlerini belirleyiniz.

- **Read & Write:** Oluşturulan değişkene hem veri yazılabilir hem de veri okunabilir.
- **Read Only:** Oluşturulan değişkenden sadece veri okunabilir.

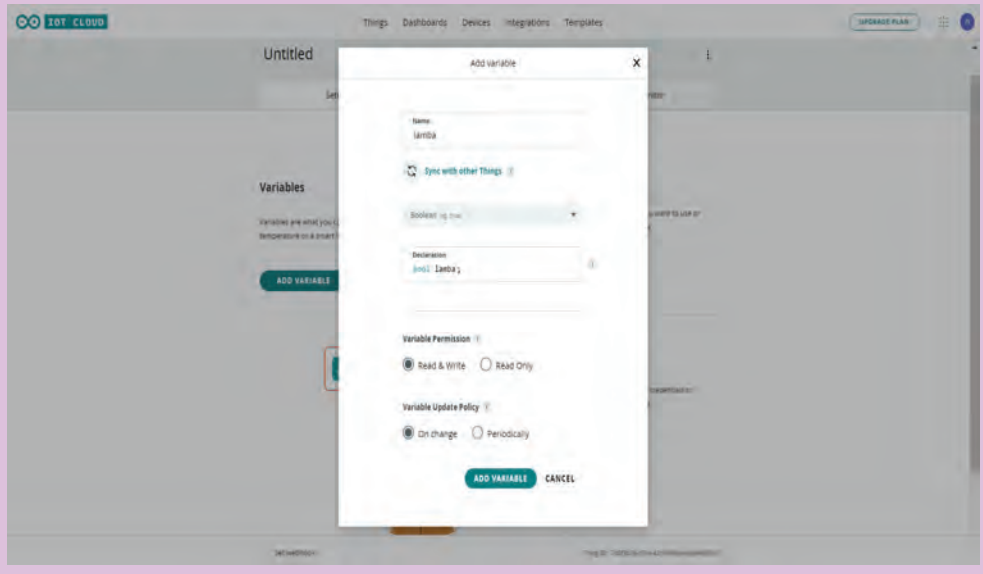
Sensörler ve durum bilgileri genellikle Read Only olurken çıkış işlemleri genellikle Read & Write olur. Değişken izinlerinin belirlenmesinden sonra veri okuma yazma işlemlerinin ne sıklıkta yapılacağının belirlenebilmesi için değişken güncelleme politikası da belirtilmelidir.

- **On change:** Var olan durum değiştiğinde veri okuma yazma işlemlerini yapar.
- **Periodically:** Belirlenen zaman değerine göre veri okuma yazma işlemlerini yapar.



Görsel 7.6: Arduino Iot Cloud'da değişken tamamlanması

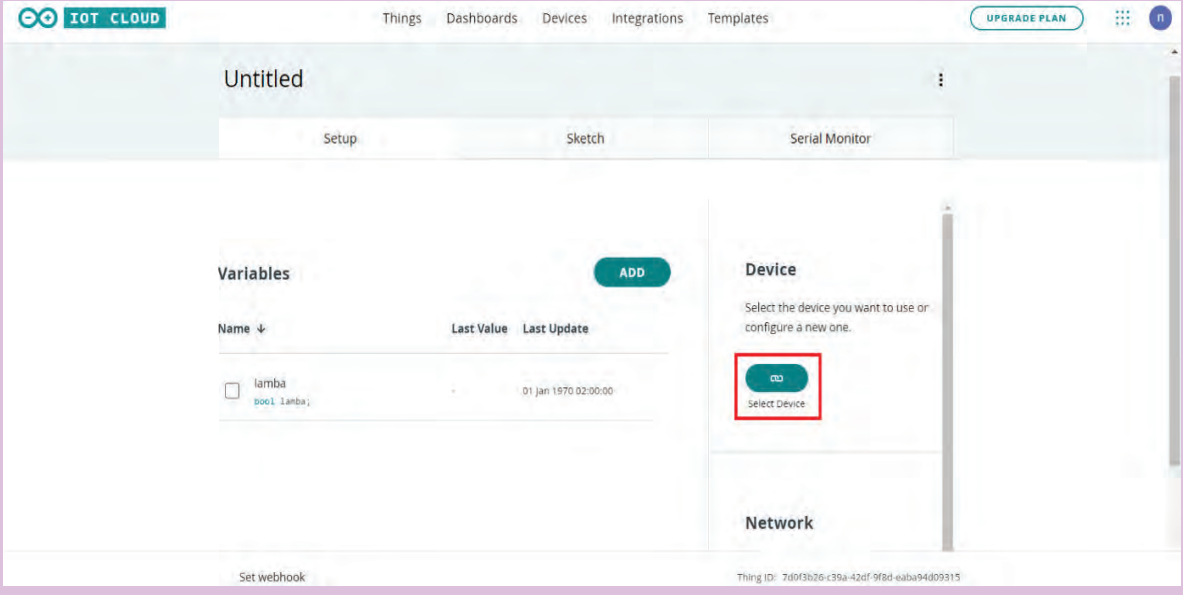
Görsel 7.7'de örnek bir değişken tanımlama ekranı görmektesiniz. Bu uygulamada sadece lamba kontrolü yapılacağı için lamba değişkenini boolean tipinde tanımlayınız.



Görsel 7.7: Arduino Iot Cloud'da örnek değişken tamamlanması

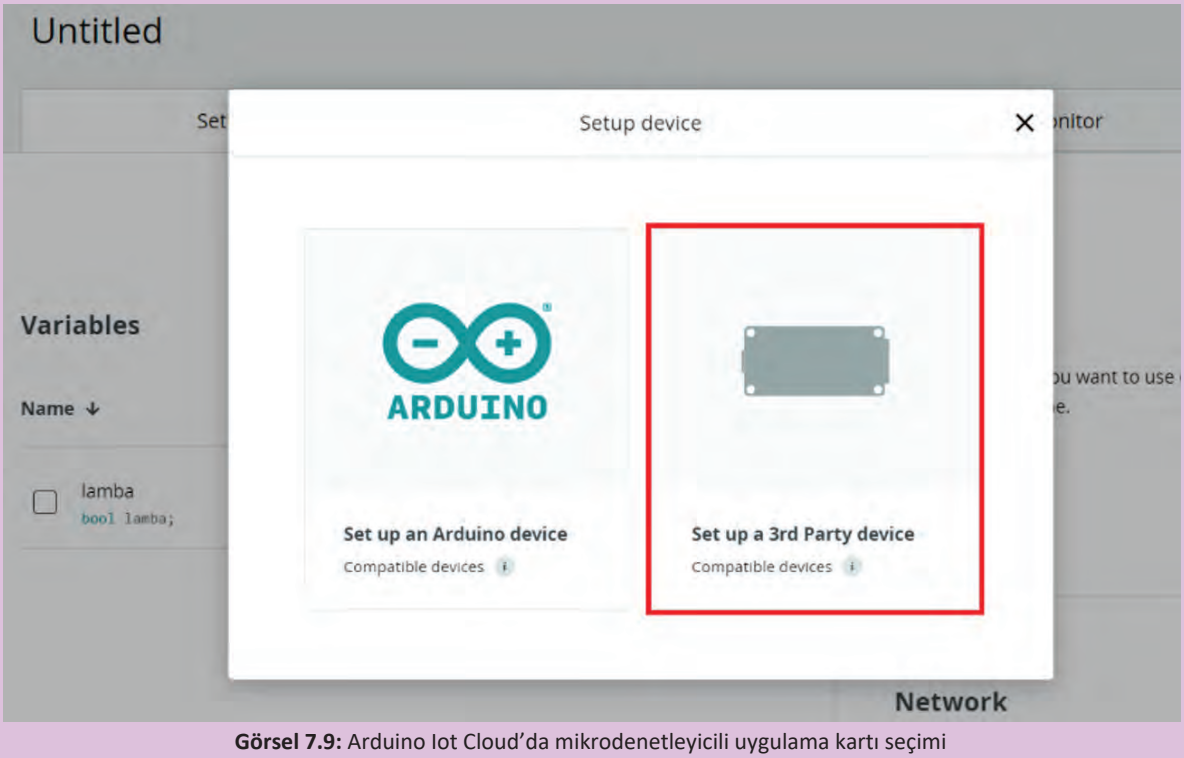
Yeni değişkenler tanımlamak isterseniz "Variables" bölümündeki "ADD" butonuna tıklayarak yeni değişkenler ekleyebilirsiniz.

7. Adım : Değişken tanımlama işlemi tamamlandıktan sonra oluşturacağınız nesnenin donanım olarak hangi alt yapıyı kullanacağını belirlemelisiniz. Bunu belirleyebilmek için Görsel 7.8'deki sayfada "Device" bölümündeki "Select Device" bölümüne tıklayınız.



Görsel 7.8: Arduino Iot Cloud'da mikrodenetleyicili uygulama kartı tamamlanması

8. Adım : Bu bölümde (Görsel 7.9) oluşturulacak nesnenin hangi donanımı temel alacağını belirleyiniz.

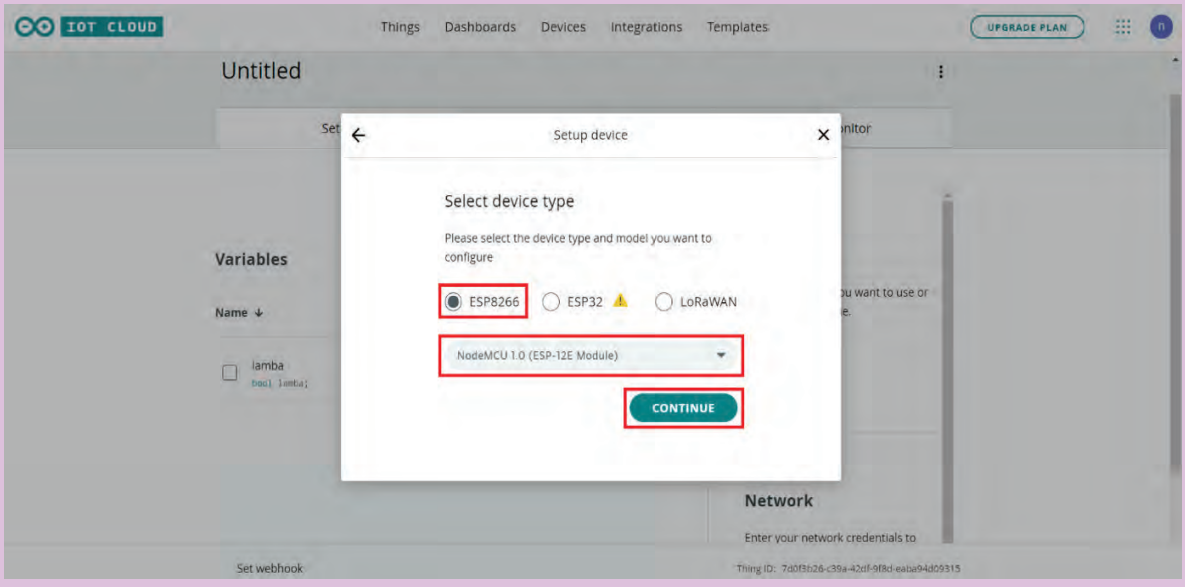


Görsel 7.9: Arduino Iot Cloud'da mikrodenetleyicili uygulama kartı seçimi

Uygulamada ESP8266 temel alan Nodemcu 1.0 12E modeli kullanılacaktır. Bu ürünün tercih edilmesindeki sebepler aşağıda belirtilmiştir.

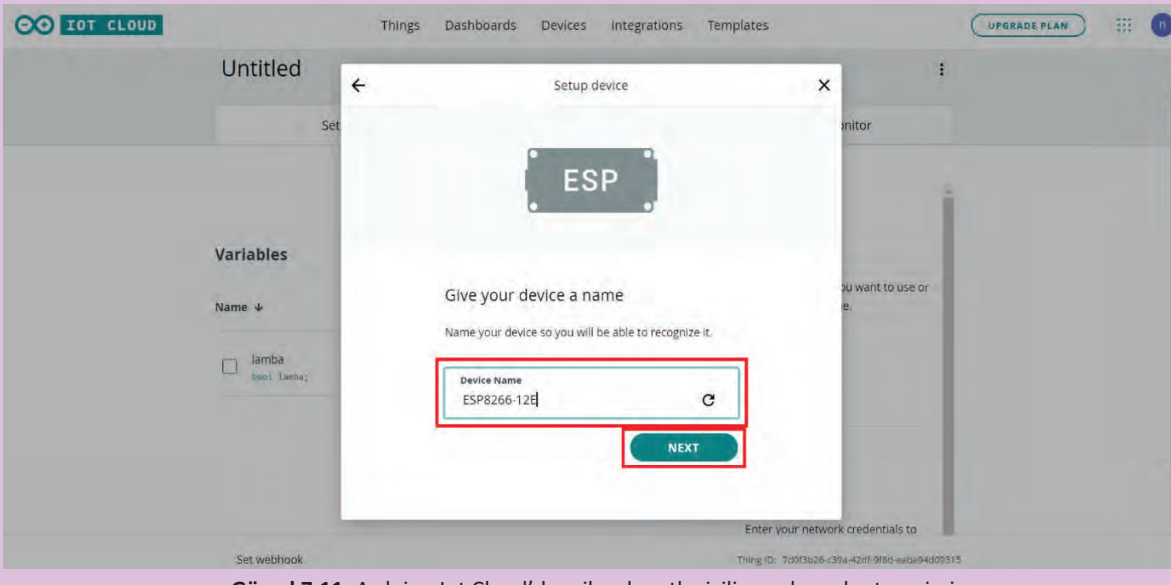
- 1 adet analog giriş (10-bit çözünürlükte)
- 11 adet dijital giriş/çıkış
- 10 adet PWM çıkış (dijital pinlerle paylaşımlı)
- Seri, I2C, SPI ve 1-Wire haberleşme protokollerine sahip karttır.

Listede arama işlemi sırasında liste incelendiğinde piyasada yaygın olarak bulunan bütün ürünlerin yer aldığını görebilirsiniz. Ekstra bir kurulum yapılmadan bütün bu kartlar kullanıma hazır durumdadır. Görsel 7.10'daki gibi seçimleri gerçekleştirip "CONTINUE" butonuna basınız.



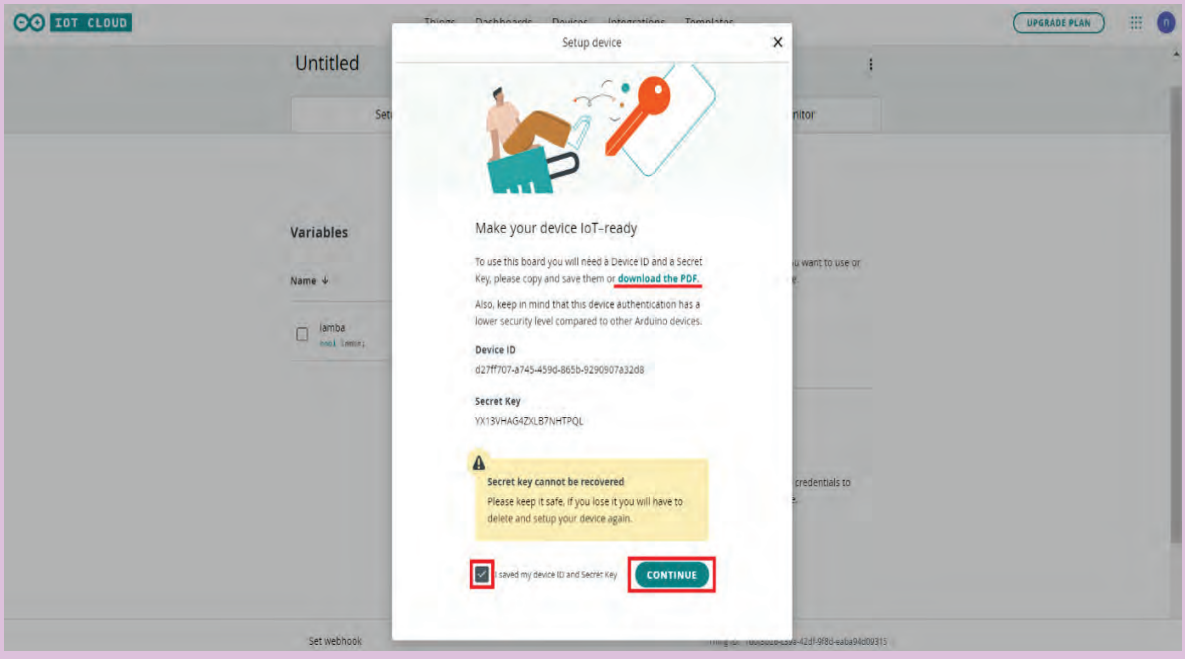
Görsel 7.10: Arduino Iot Cloud'da mikrodenetleyicili uygulama kartı seçimi

9. Adım : Kartınıza bir isim verip "NEXT" butonuna basınız (Görsel 7.11).



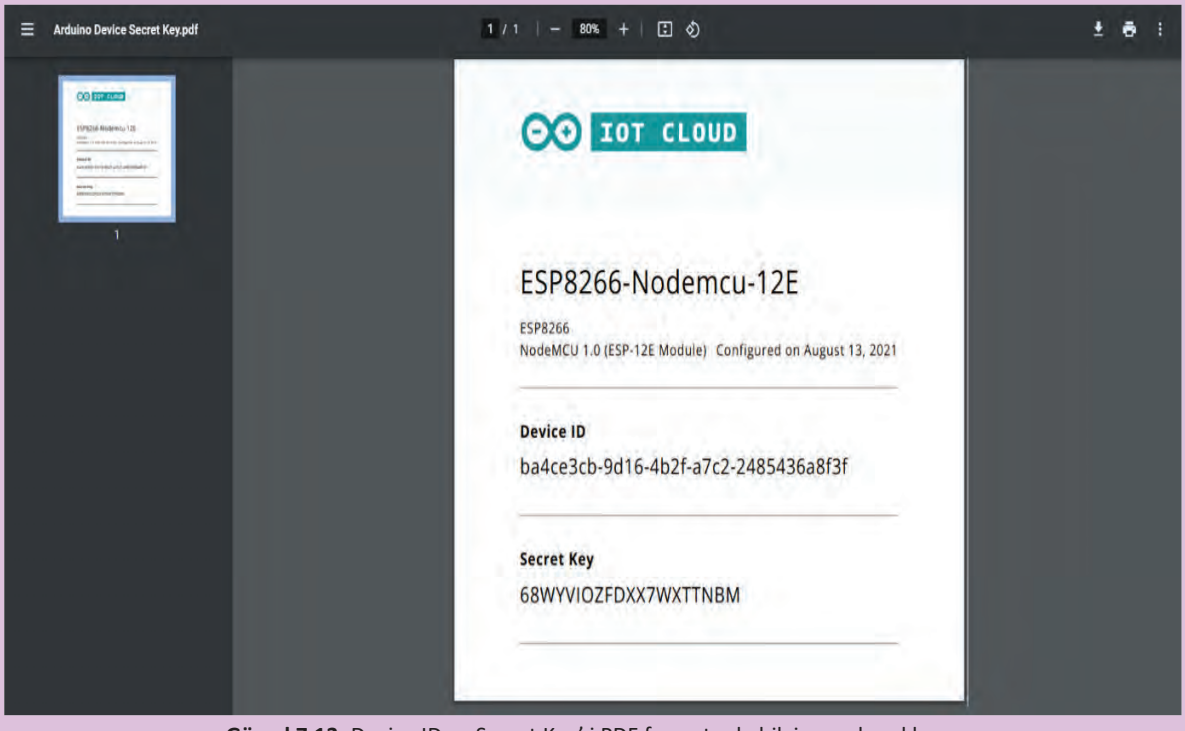
Görsel 7.11: Arduino Iot Cloud'da mikrodenetleyicili uygulama kartı seçimi

10. Adım : Görsel 7.12'deki ekranda Device ID ve Secret Key bilgileri yer almaktadır. Bu ekrandaki Secret Key bilgisi çok önemlidir. Bu bilgi ilerleyen bölümlerde Network bölümde tekrar kullanılacağından saklanmalıdır.



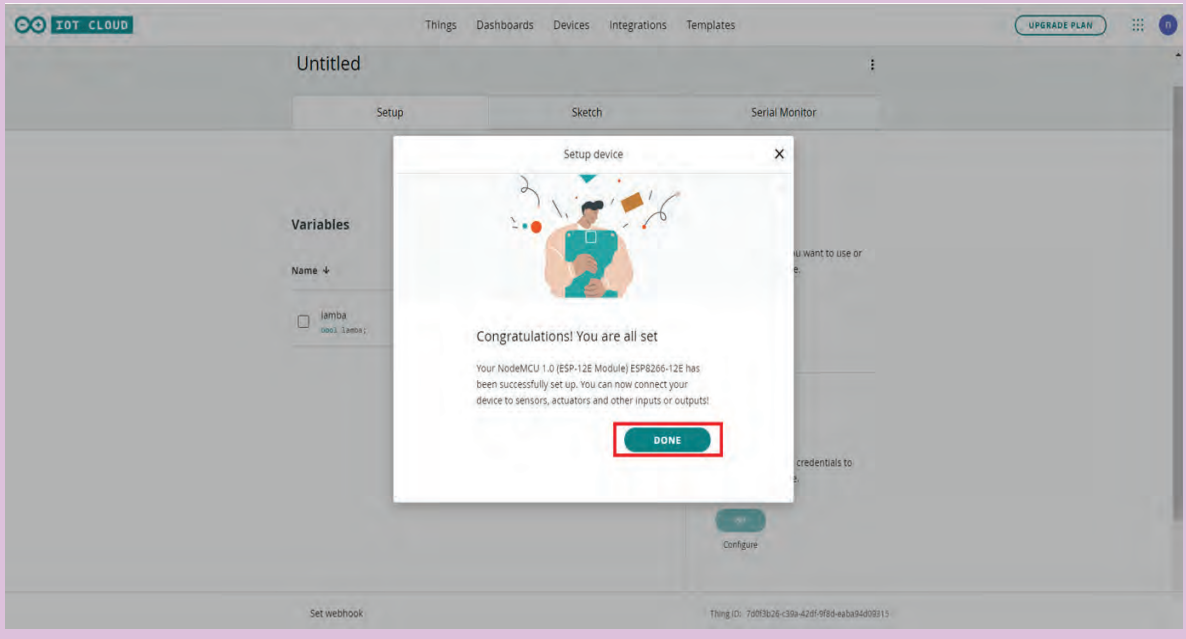
Görsel 7.12: Device ID ve Secret Key bilgi ekranı

11. Adım : Sayfa içinde "download the PDF" linkine tıklayarak bu bilgiyi PDF formatında bilgisayarınızda saklayabilirsiniz (Görsel 7.13).



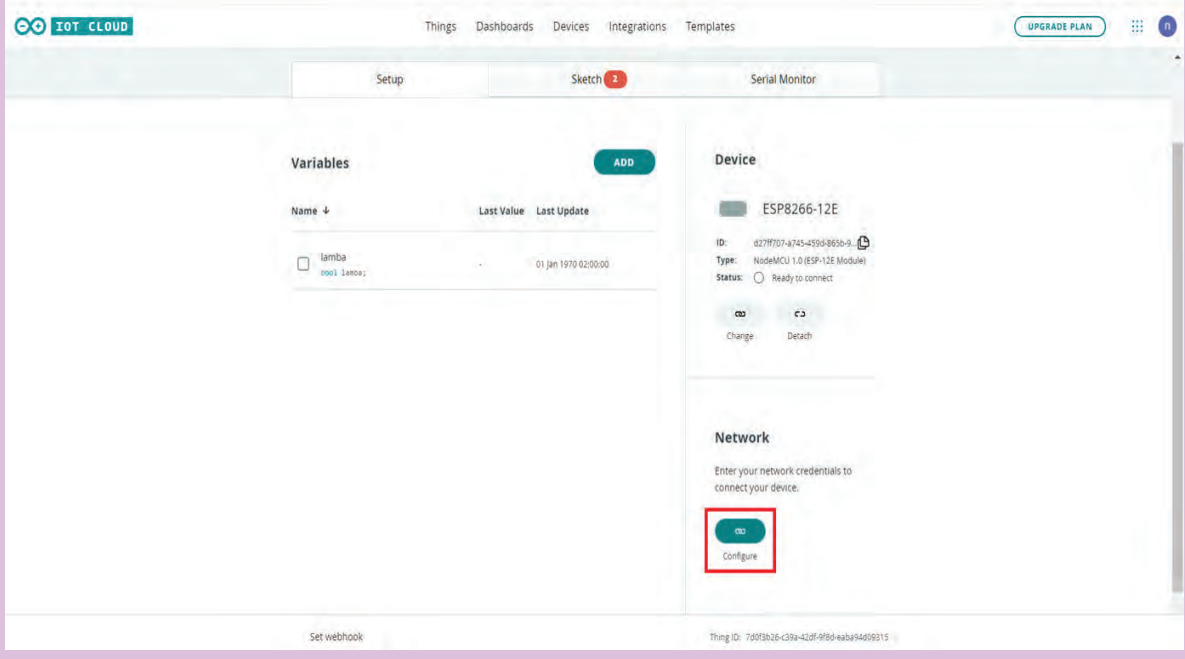
Görsel 7.13: Device ID ve Secret Key' i PDF formatında bilgisayarda saklama

12. Adım : Nesnelerin İnterneti için kullanılacak donanımın ayarlarını tamamladığınızda karşınıza Görsel 7.14'teki ekran gelir. Bu ekrana ulaşılamadıysa Device bölümüne tıklayıp oluşturduğunuz donanımı silip 7. Adımdan tekrar başlayabilirsiniz.



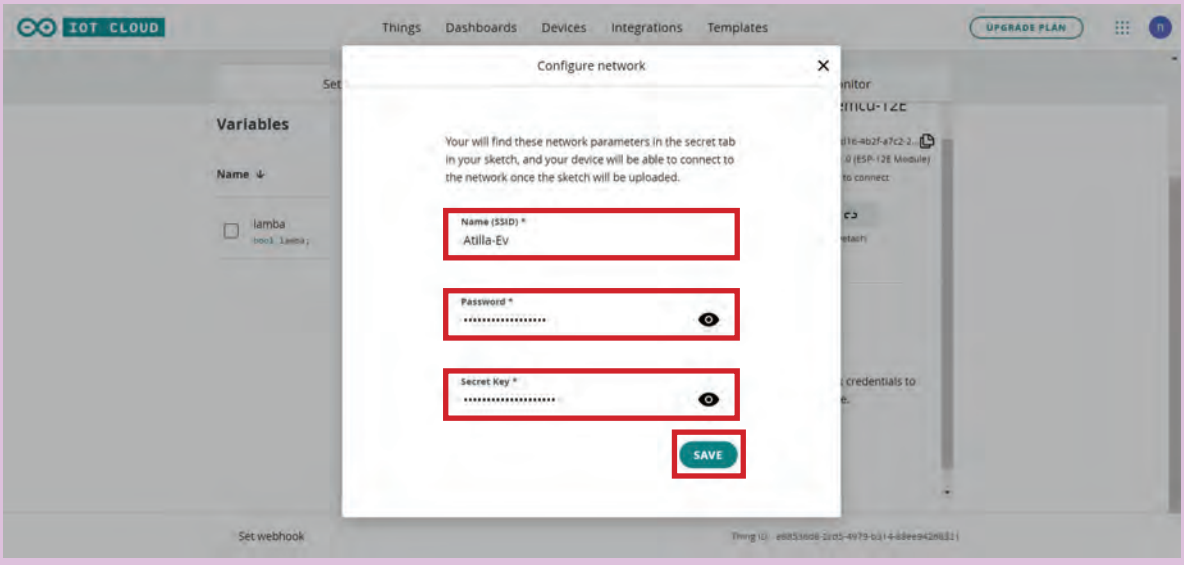
Görsel 7.14: Donanım kurulumunun başarı ile tamamlanması

13. Adım : Bu bölümde kullanılacak donanım kartı içine nesnenin kullanılacak olduğu ortamın Wi-Fi bilgisi ve daha önce Device bölümündeyken oluşturulan Secret Key bilgileri kaydedilecektir. Secret Key bilgisine sahip olan her kişi sizin donanımınızın bilgisini okuyabilir ve bilgilerinizi taklit edebilir. Bu nedenle bu bilgiyi gizli tutmalısınız. Görsel 7.15'teki "Network" bölümündeki "Configure" butonuna basınız.



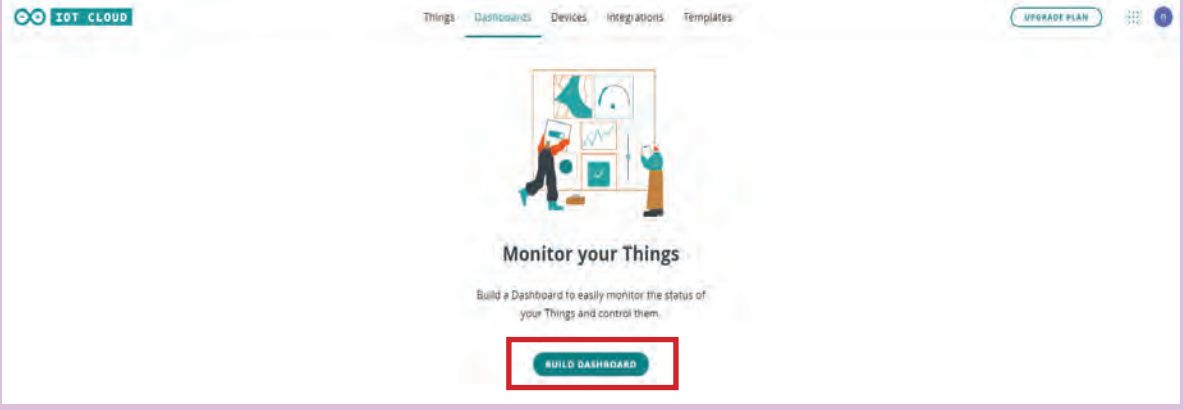
Görsel 7.15: Network ayarlarının yapılandırılması

14. Adım : Açılan ekranda (Görsel 7.16) donanımın kullanılacak olduğu yerdeki ağın adı (Name(SSID)), şifresi (Password) ve bilgisayarınızın "Device" bölümünde kaydetmiş olduğunuz kod (Secret Key) bilgilerini giriniz. Bu bilgileri doğru olarak girdiğinize emin olduktan sonra "SAVE" butonuna basarak işlemi tamamlayınız.



Görsel 7.16: Network ayarlarının yapılandırılması

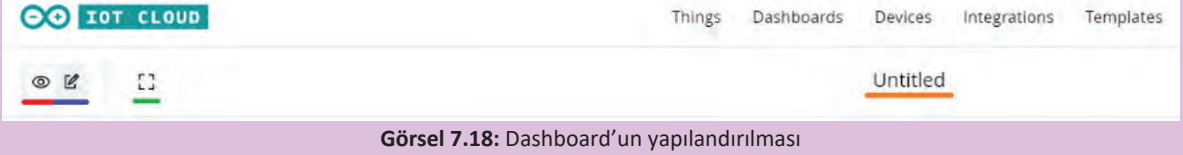
15. Adım : Oluşturulan çalışmanın web sayfası ve cep telefonundaki görünümünü ayarlayabilmek için "Dashboards" sekmesine tıklayınız (Görsel 7.17) ve "BUILD DASHBOARD" butonuna tıklayınız.



Görsel 7.17: Dashboard'un yapılandırılması

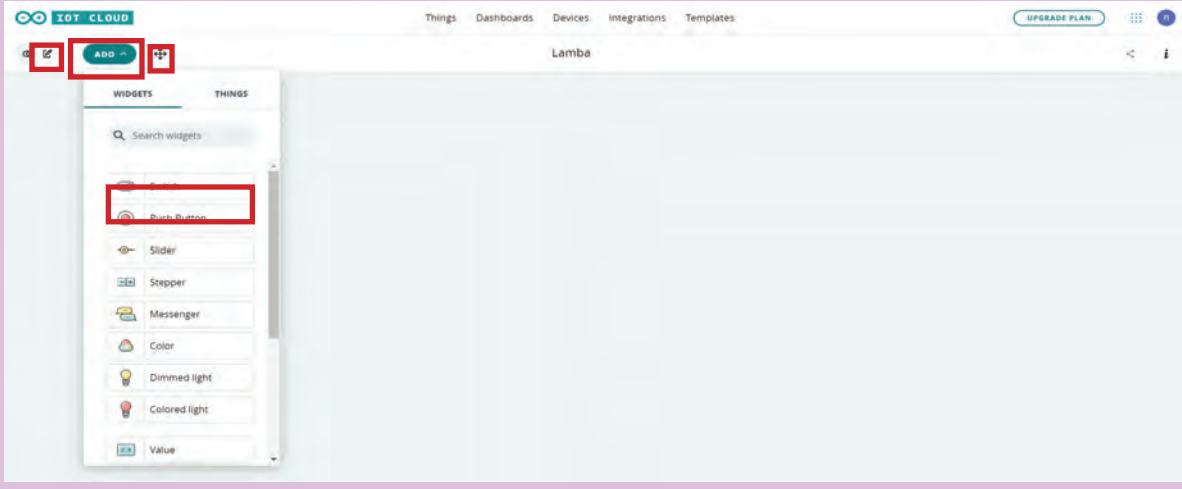
Görsel 7.18'deki renklerin açıklamaları aşağıdadır.

- ___ Bu bölüme basıldığında dashboard'ın görünümü kilitlenir.
- ___ Bu bölüme basıldığında dashboard'u düzenleme işlemleri yapılır.
- ___ Bu bölüme basıldığında tam ekran moduna geçilir.
- ___ Bu bölüme basıldığında dashboard'un ismi değiştirilir.



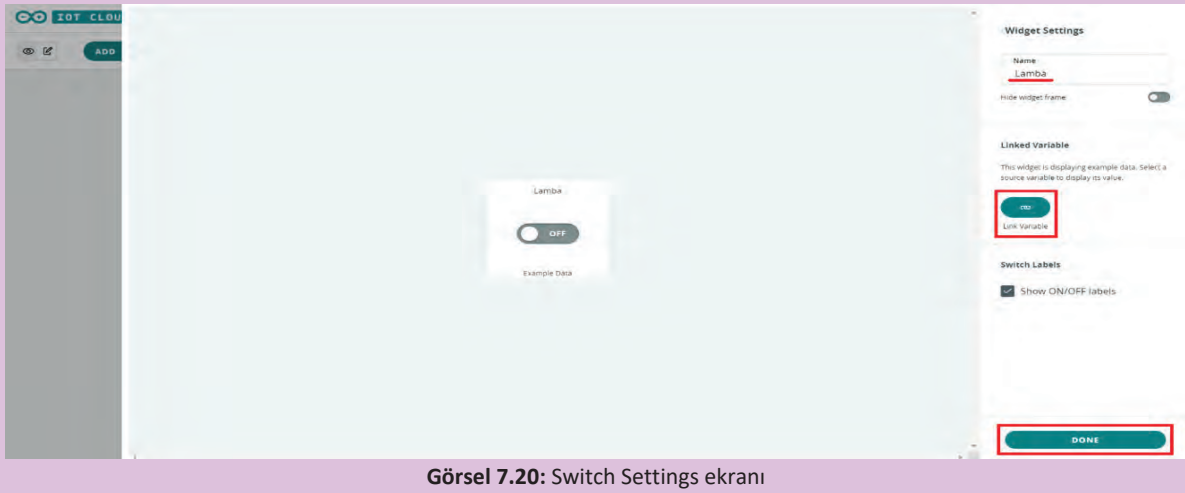
Görsel 7.18: Dashboard'un yapılandırılması

16. Adım : Dashboard'a Görsel 7.19'daki işlem sırası ile önce düzenleme butonuna basınız. Daha sonra "ADD" butonuna basarak çalışma alanına "Switch" ekleyiniz.

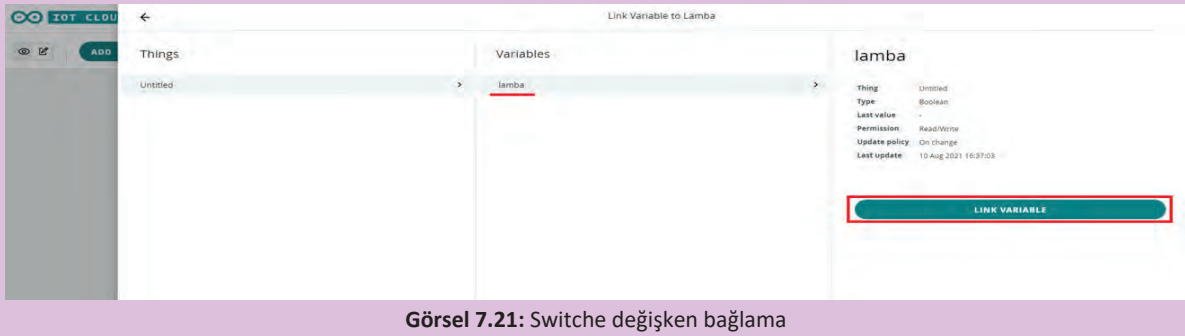


Görsel 7.19: Dashboard'a Switch ekleme

17. Adım : Switch alana eklendikten hemen sonra karşınıza Görsel 7.20'deki "Edit Settings" ekranı gelir. "Name" yazan bölümde switchin görünen adını belirleyiniz. "Link Variable" bölümünden switchi oluşturulan değişkene bağlanınız (Görsel 7.21). Açılan sayfada hangi nesneden bağlantı oluşturulacak ise önce o bağlantıyı seçip o nesne için tanımlanan uygun değişkenleri aktif şekilde listeleyiniz. Uygun değişkeni seçtikten sonra "LINK VARIABLE" butonuna basarak bağlantı oluşturunuz. Bu işlem tamamlandığında Görsel 7.20'deki ekrana geri dönüp "DONE" butonuna basarak bağlantıyı tamamlayınız.

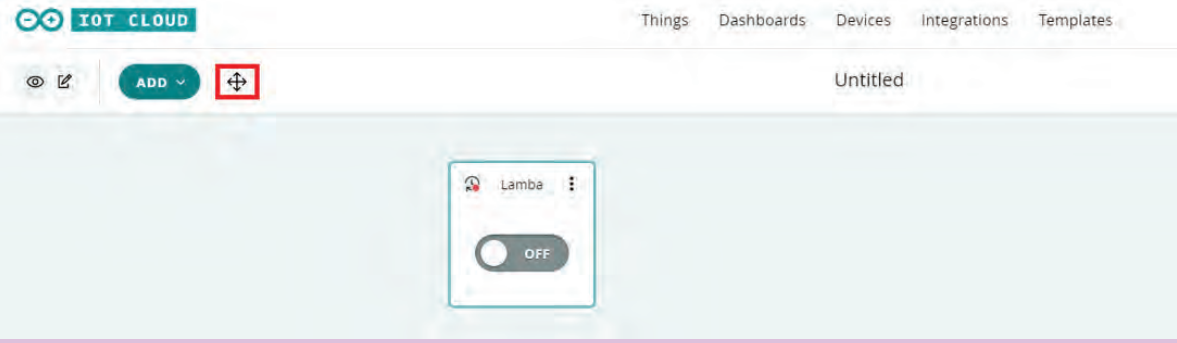


Görsel 7.20: Switch Settings ekranı



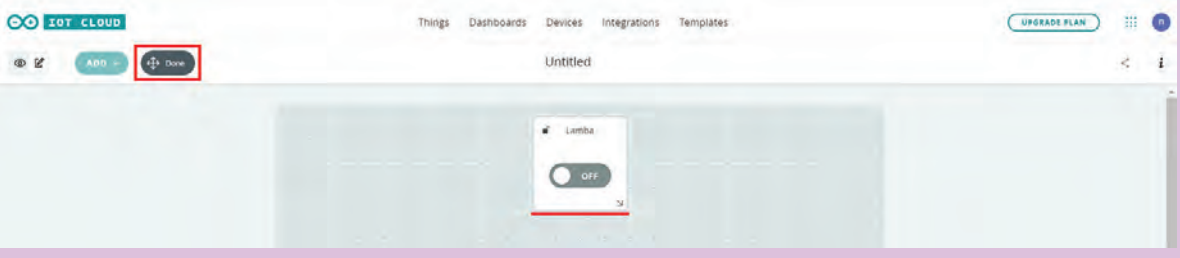
Görsel 7.21: Switche değişken bağlama

18. Adım : Eklenen Widget'i silme, kopyalama ve bilgileri değiştirme için isim yanında yer alan üç noktaya basarak bu işlemleri gerçekleştirebilirsiniz. Görsel 7.22'de yer alan kırmızı kutu içindeki butona basarak eklenen Widget'in ekranda nerede konumlanacağını ve boyutunu ayarlayınız.



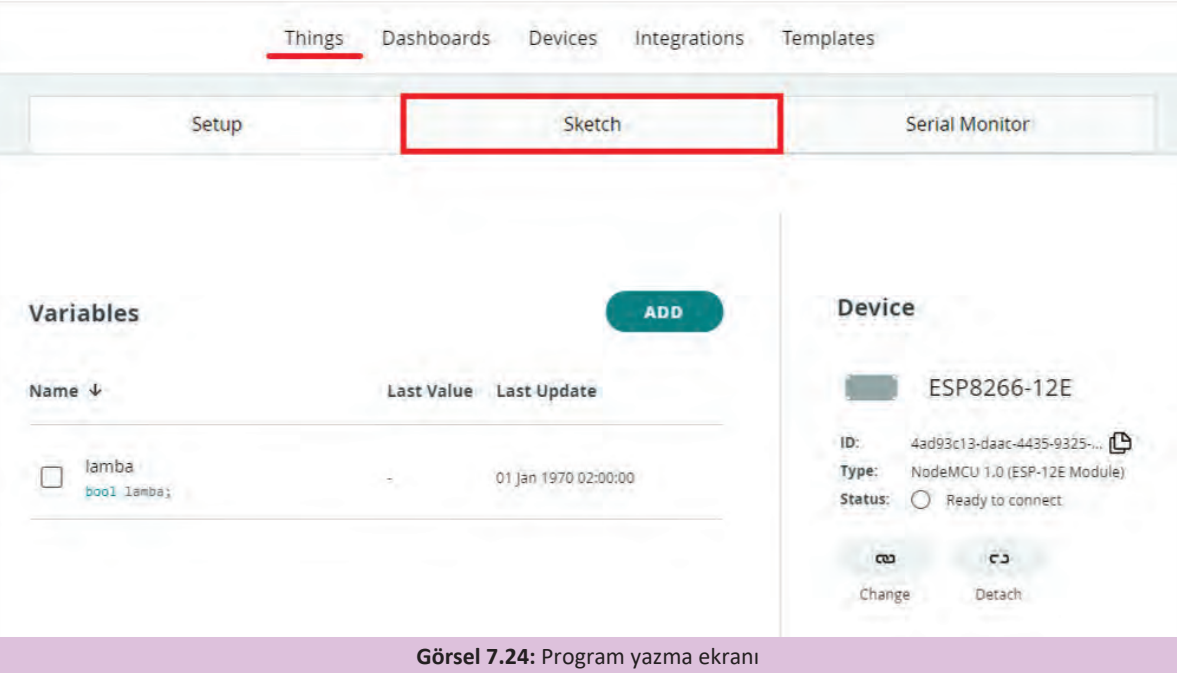
Görsel 7.22: Switch Widget'ini ekranda konumlama ve boyutunu ayarlama

19. Adım : Görsel 7.23'teki Done butonuna basarak konumlama ve boyutlandırma işlemini tamamlayınız.



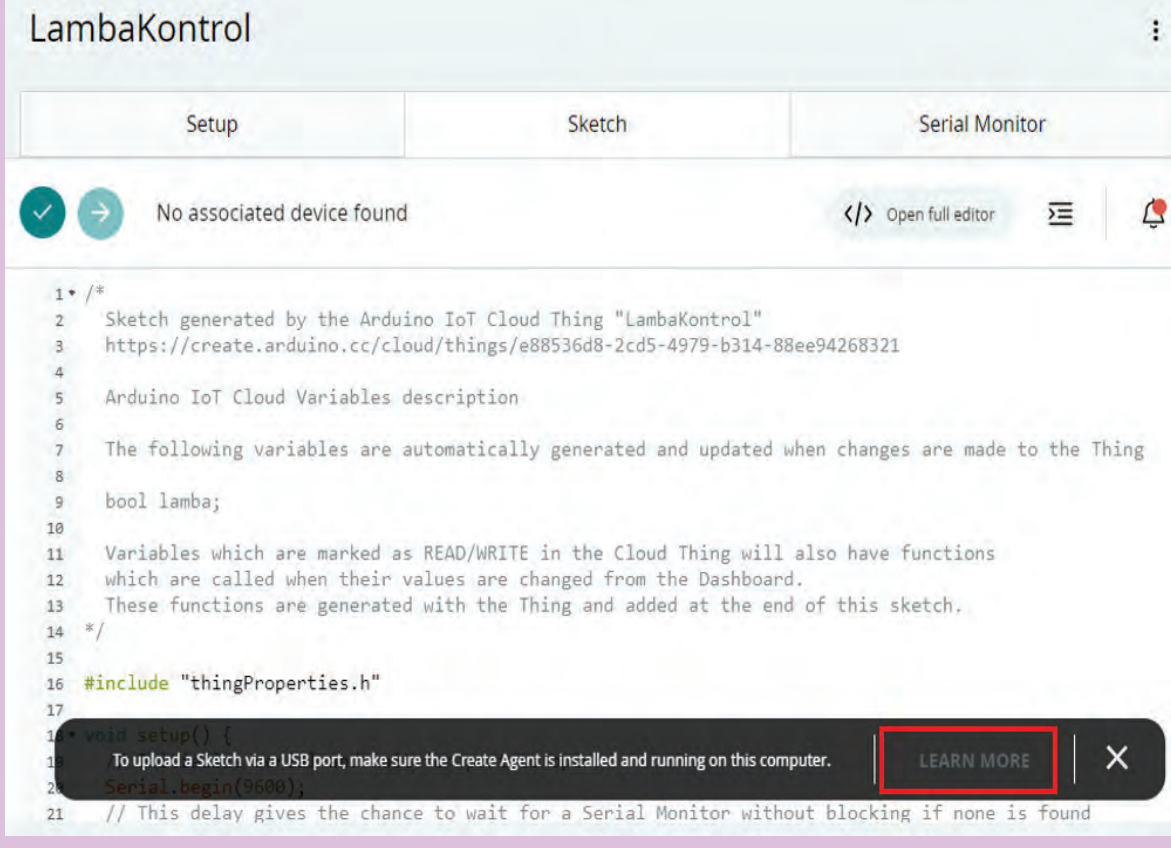
Görsel 7.23: Switch Widget'ini ekranda konumlama ve boyutunu ayarlama

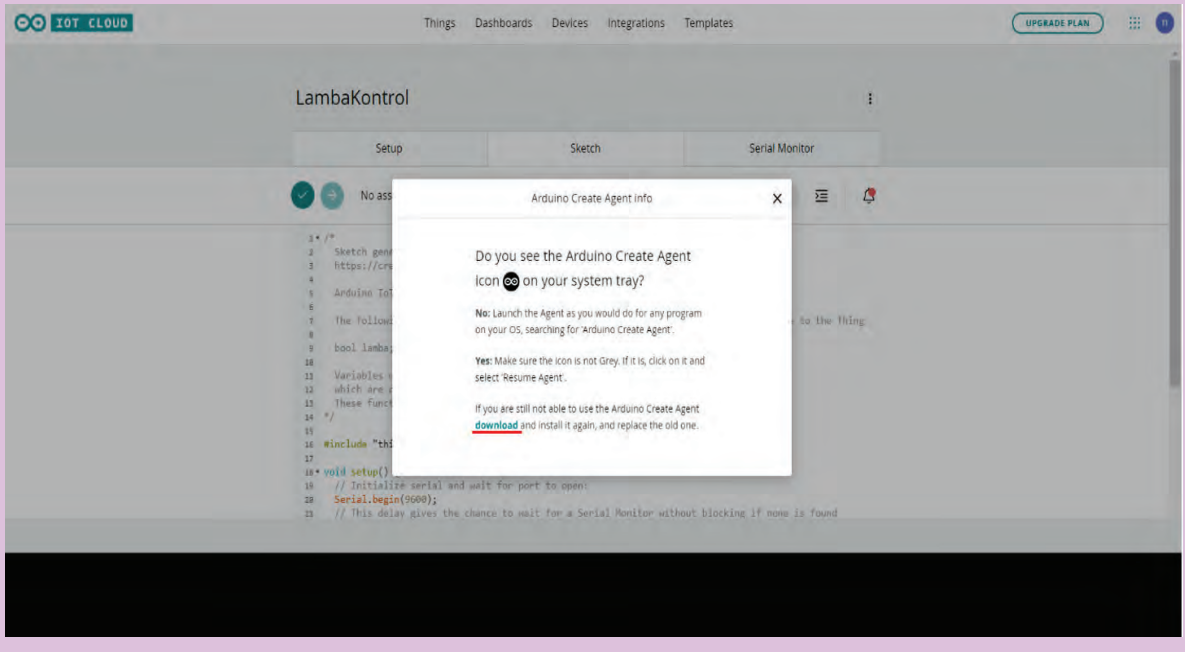
20. Adım : Nesne tanımlaması, kullanılacak değişkenler, donanım, ağ bağlantı ayarları ve web-mobil arayüz tasarımından sonra kullanılacak programı yazmak için Görsel 7.24'teki "Things" sekmesine geçip hazırlanan IoT çalışmasına tıklayınız. Daha sonra "Sketch" sekmesine geçiniz.

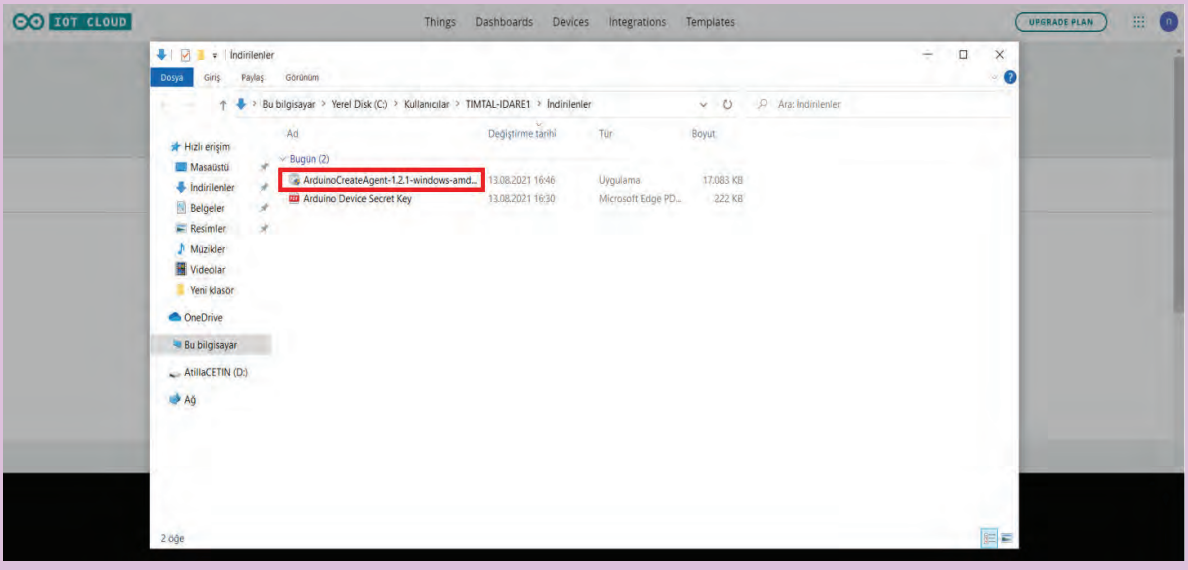


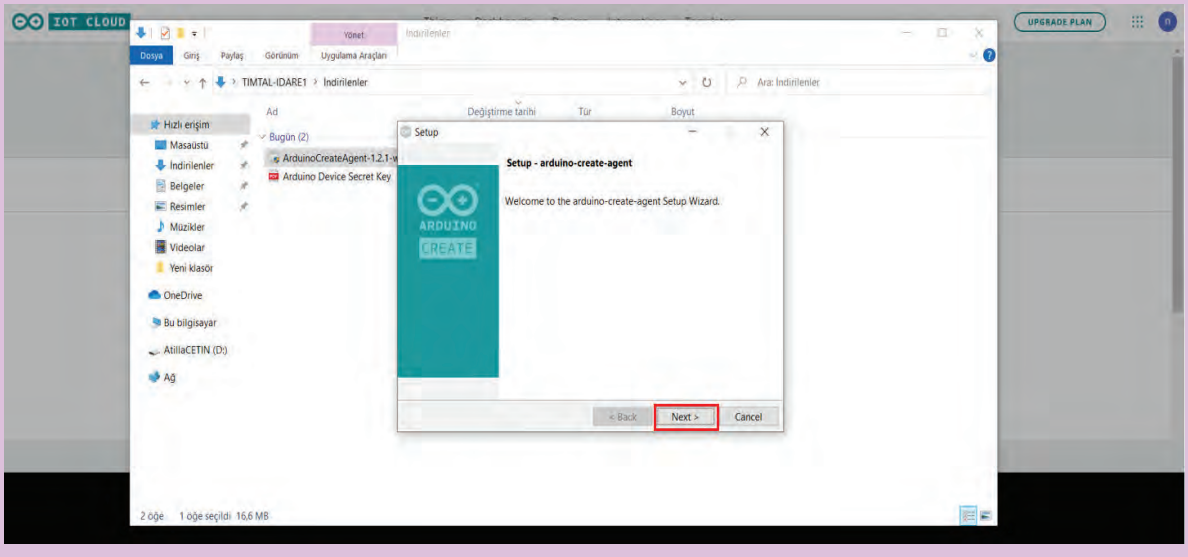
Görsel 7.24: Program yazma ekranı

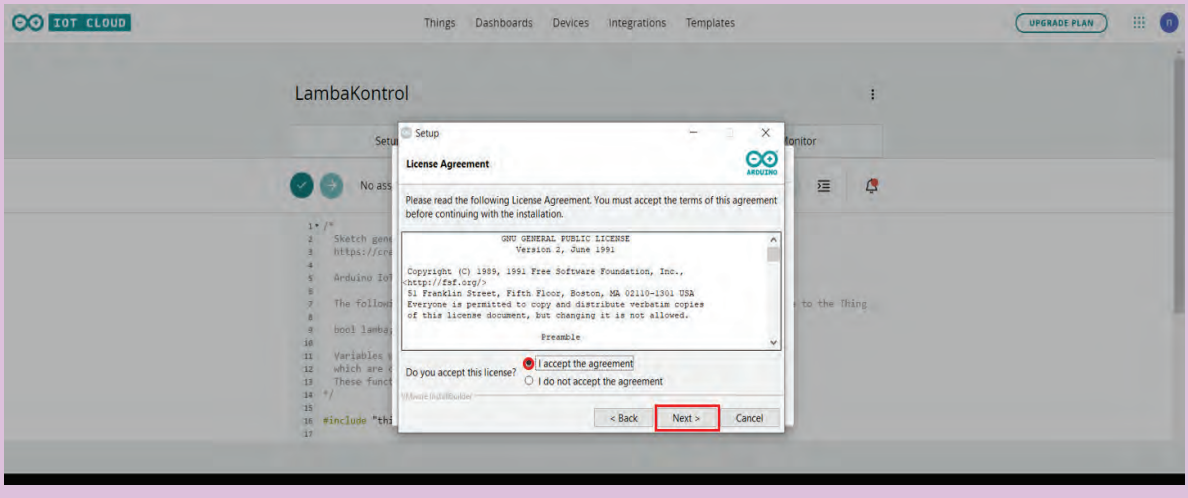
21. Adım : Bilgisayarınızda “Arduino Create Agent” programı kurulu ise Görsel 7.25’teki gibi bir ekran görüntüsü ile karşılaşırsınız. Eğer kurulu değilse Görsel 7.25’teki adımları takip ederek kurulumu tamamlayınız. Bu yardımcı program; web sayfası üzerinden hazırlanan programın derlenmesini sağlayıp karta aktarılmasını sağlar.

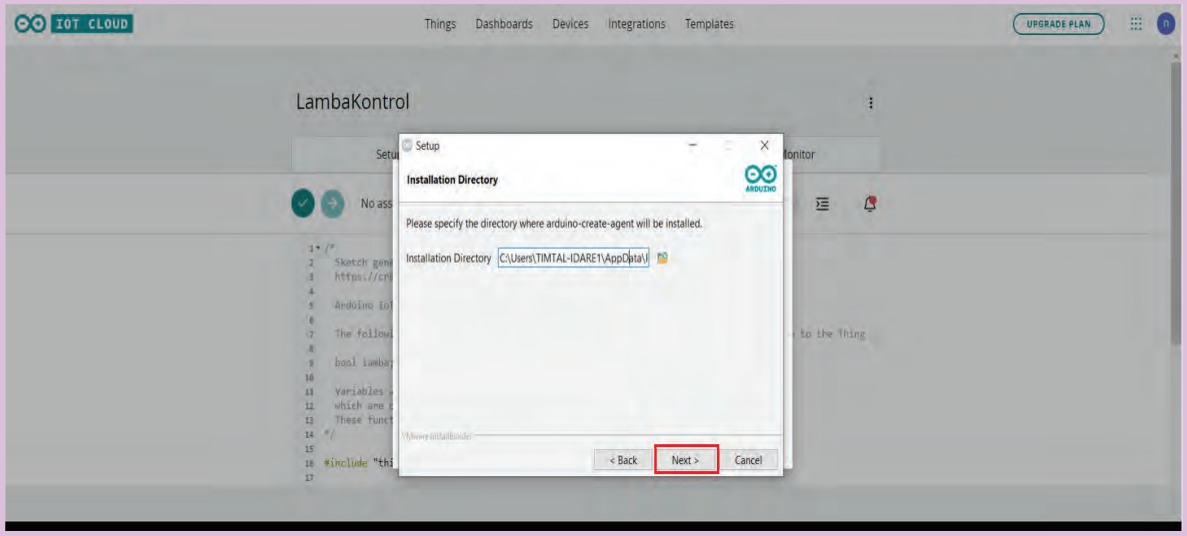


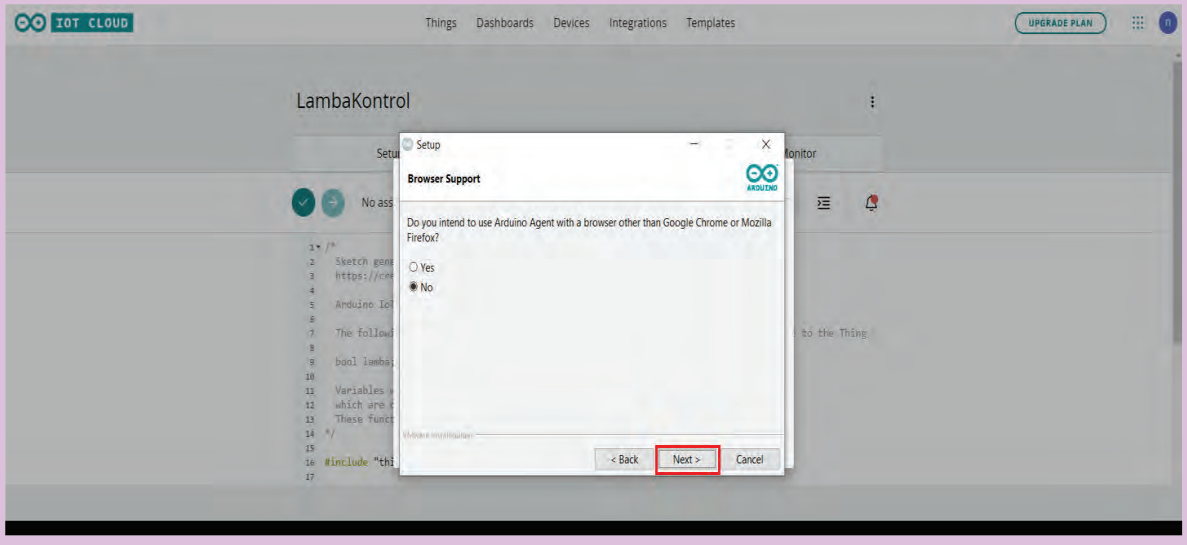


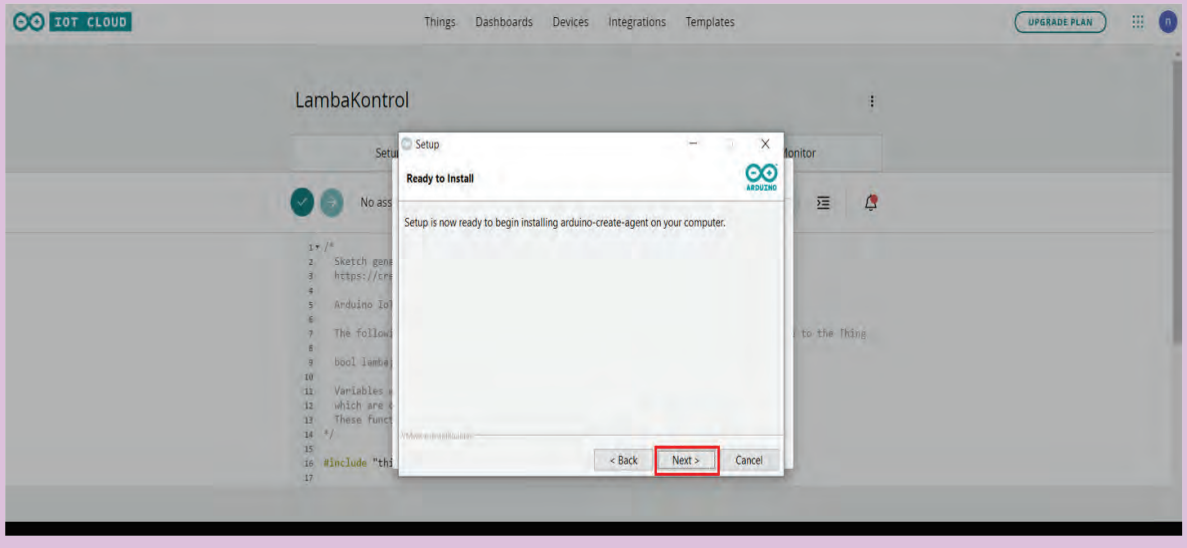

IOT CLOUD
Things
Dashboards
Devices
Integrations
Templates
UPGRADE PLAN
İndirilenler
Dosya
Giriş
Paylaş
Görünüm
Bu bilgisayar > Yerel Disk (C:) > Kullanıcılar > TIMTAL-IDARE1 > İndirilenler
Ara: İndirilenler
Ad
Değiştirme tarihi
Tür
Boyut
Hızlı erişim
Masaüstü
İndirilenler
Belgeler
Resimler
Müzikler
Videolar
Yeni klasör
OneDrive
Bu bilgisayar
AtillaCETIN (D:)
Ağ
Bugün (2)
ArduinoCreateAgent-1.2.1-windows-amd...
13.08.2021 16:46
Uygulama
17.083 KB
Arduino Device Secret Key
13.08.2021 16:30
Microsoft Edge PD...
222 KB
2 öğe

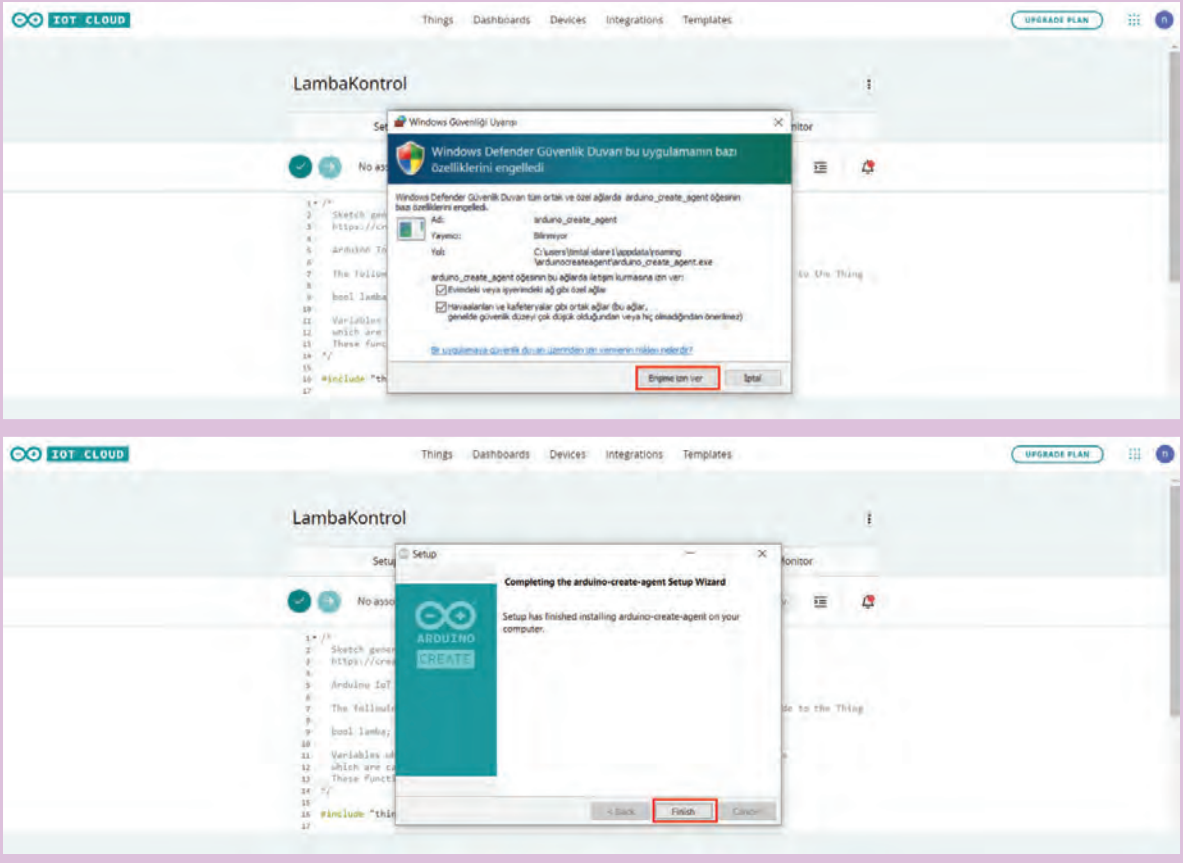

IOT CLOUD
UPGRADE PLAN
Yönet
İndirilenler
Dosya
Giriş
Paylaş
Görünüm
Uygulama Araçları
TIMTAL-IDARE1 > İndirilenler
Ara: İndirilenler
Hızlı erişim
Masaüstü
İndirilenler
Belgeler
Resimler
Müzikler
Videolar
Yeni klasör
OneDrive
Bu bilgisayar
AtillaCETIN (D:)
Ağ
Bugün (2)
ArduinoCreateAgent-1.2.1-w
Arduino Device Secret Key
Setup
Setup - arduino-create-agent
Welcome to the arduino-create-agent Setup Wizard.
ARDUINO
CREATE
< Back
Next >
Cancel
2 öğe
1 öğe seçildi 16,6 MB


IOT CLOUD
Things
Dashboards
Devices
Integrations
Templates
UPGRADE PLAN
LambaKontrol
Setup
License Agreement
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
GNU GENERAL PUBLIC LICENSE
Version 2, June 1991
Copyright (C) 1989, 1991 Free Software Foundation, Inc.,
<http://fsf.org/>
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.
Preamble
Do you accept this license?
I accept the agreement
I do not accept the agreement
< Back
Next >
Cancel


IOT CLOUD
Things
Dashboards
Devices
Integrations
Templates
UPGRADE PLAN
LambaKontrol
Setup
Installation Directory
Please specify the directory where arduino-create-agent will be installed.
Installation Directory
C:\Users\TIMTAL-IDARE1\AppData\
< Back
Next >
Cancel


IOT CLOUD
Things
Dashboards
Devices
Integrations
Templates
UPGRADE PLAN
LambaKontrol
Setup
Browser Support
Do you intend to use Arduino Agent with a browser other than Google Chrome or Mozilla Firefox?
Yes
No
< Back
Next >
Cancel

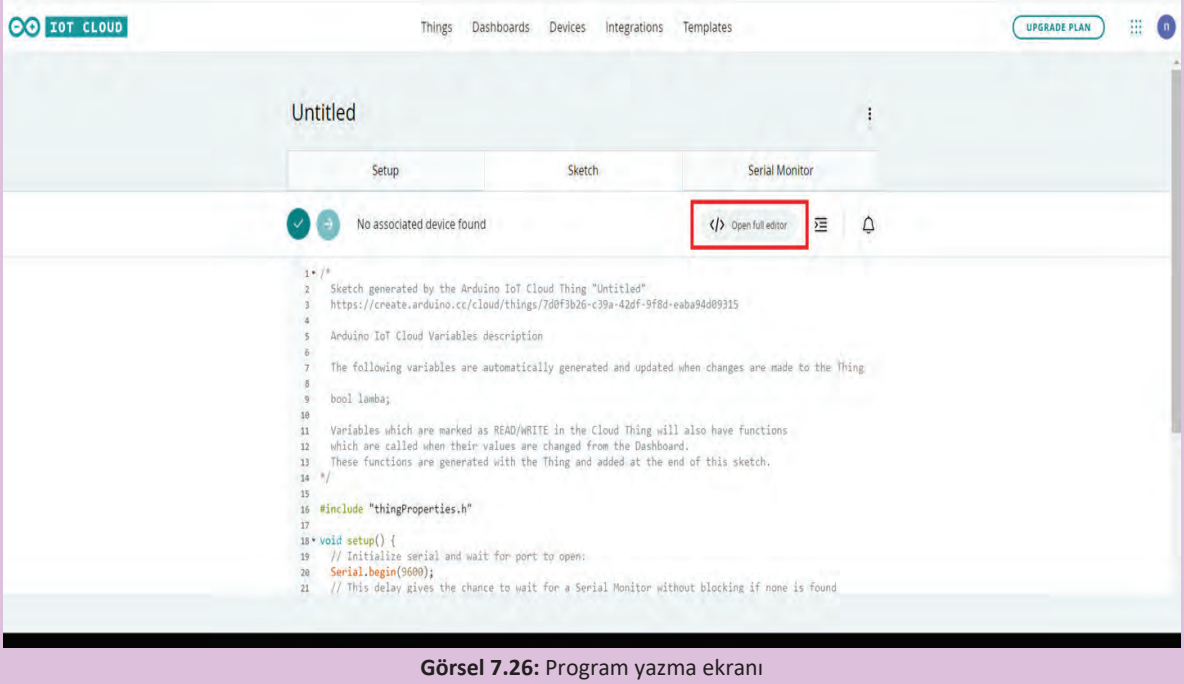

IOT CLOUD
Things
Dashboards
Devices
Integrations
Templates
UPGRADE PLAN
LambaKontrol
Setup
Ready to Install
Setup is now ready to begin installing arduino-create-agent on your computer.
< Back
Next >
Cancel



Görsel 7.25: Arduino Create Agent yardımcı programının kurulması

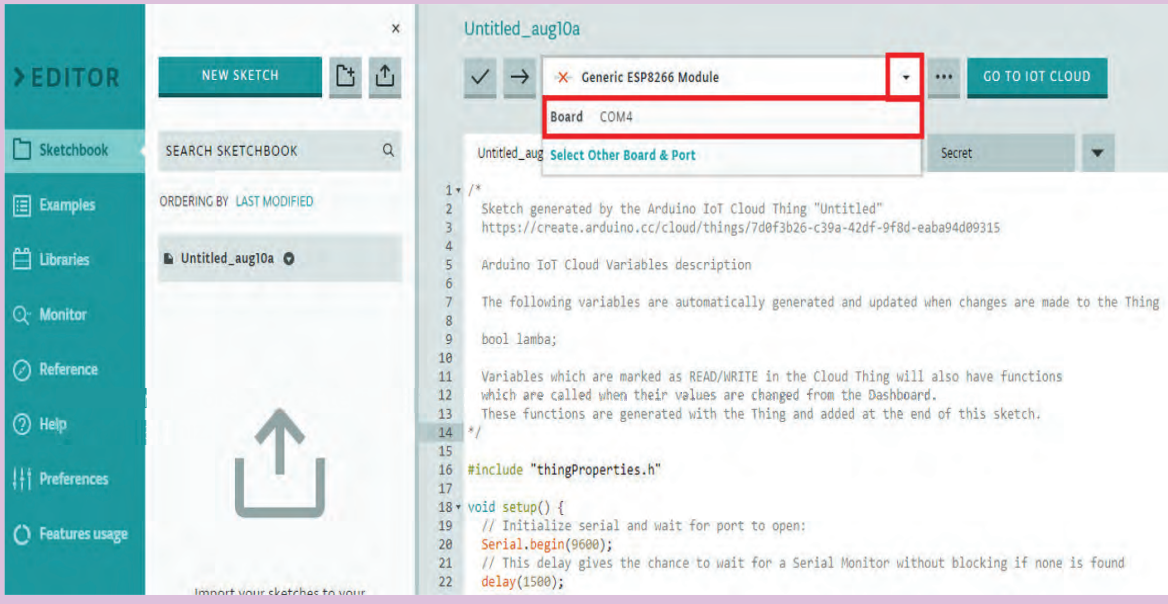
Yukarıdaki adımları izleyerek kurulumu tamamlayınız.

"Open full editor" butonuna basarak program yazabileceğiniz editöre geçiş yapınız (Görsel 7.26).



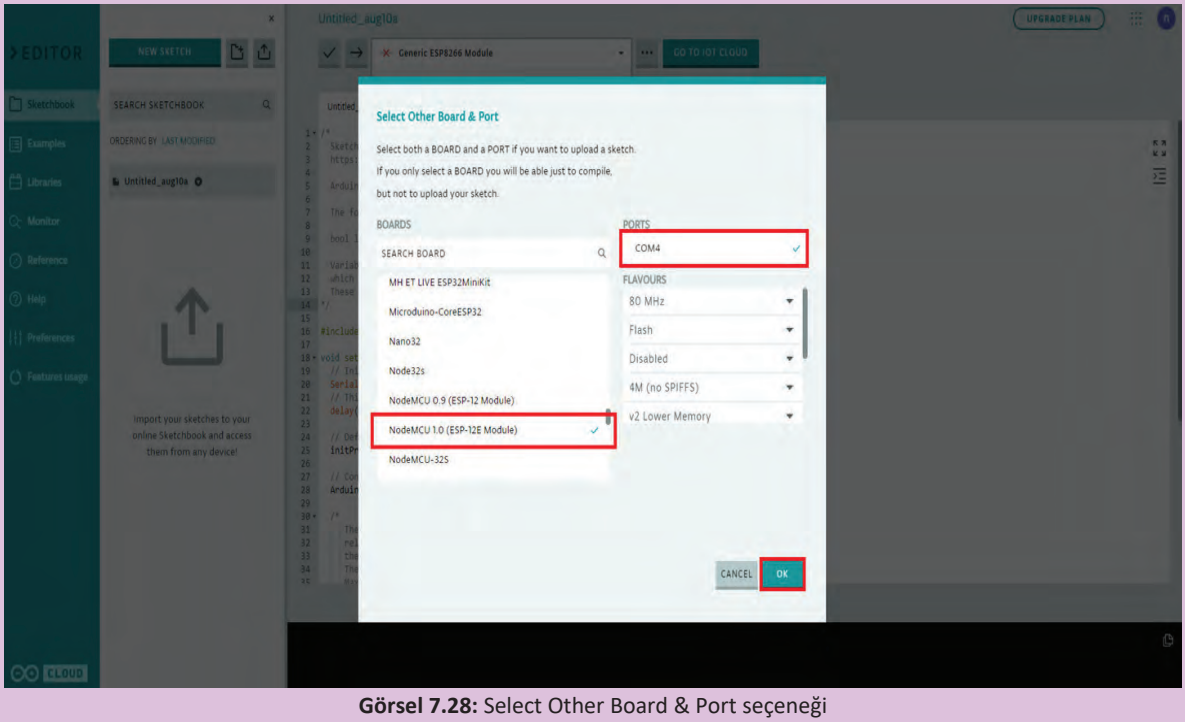
Görsel 7.26: Program yazma ekranı

22. Adım : Full editör moduna geçildiğinde kullanılacak uygulama kartını bilgisayarınızda boş olan herhangi bir USB portuna bağlayınız. Görsel 7.27'deki kırmızı kutu içinde yer alan aşağı oka basıldığında uygulama kartının hangi portta bağlı olduğu görülecektir. O portu seçiniz.
Eğer herhangi bir port numarası gözükmüyor ise bilgisayarınıza öncelikle Arduino ide programını ve sonrasında ch341 kütüphanesini kurmanız gerekmektedir.



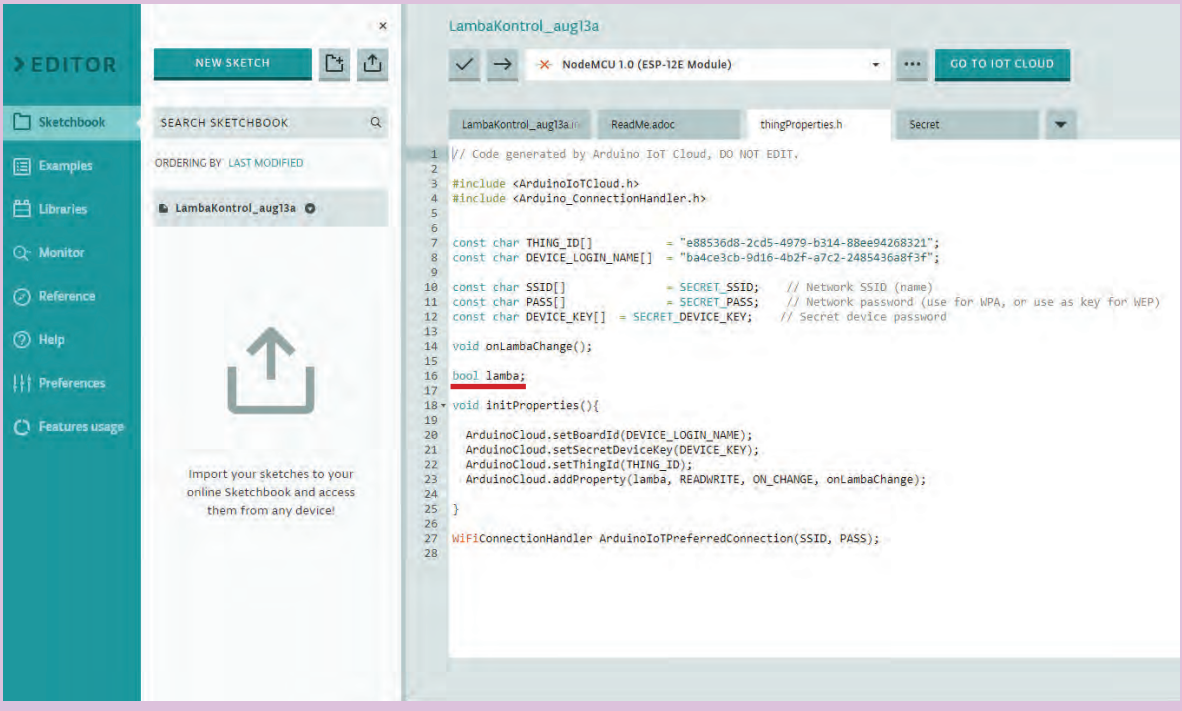
Görsel 7.27: Uygulama kartına USB portu üzerinden bağlanma

23. Adım : Başka kart veya porttan bağlanmak için "Select Other Board & Port seçeneğine tıklayarak yaygın olarak bulunan bütün kartlar içinden kendi kartınızı ve bağlı olduğu port numarasını seçerek "OK" düğmesine basıp işlemi tamamlayınız (Görsel 7.28).



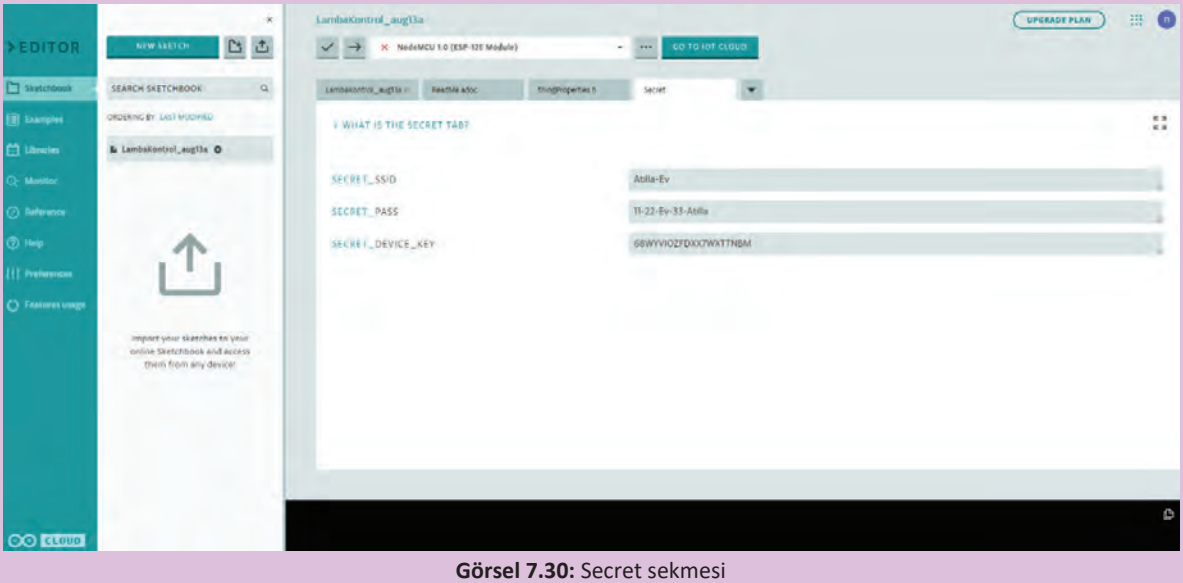
Görsel 7.28: Select Other Board & Port seçeneği

24. Adım : Program yazımına geçilmeden öncelikle kontrol edilmesi gereken iki sekme bulunur. Bu sekmelerden ilki thingProperties.h sekmesidir (Görsel 7.29). Bu sekmede uygulama için tanımlanan değişkenin yazım şekline dikkat ediniz. Çünkü program içinde bu sekmede nasıl yazılmışsa aynı şekilde yazarak kullanmalısınız.



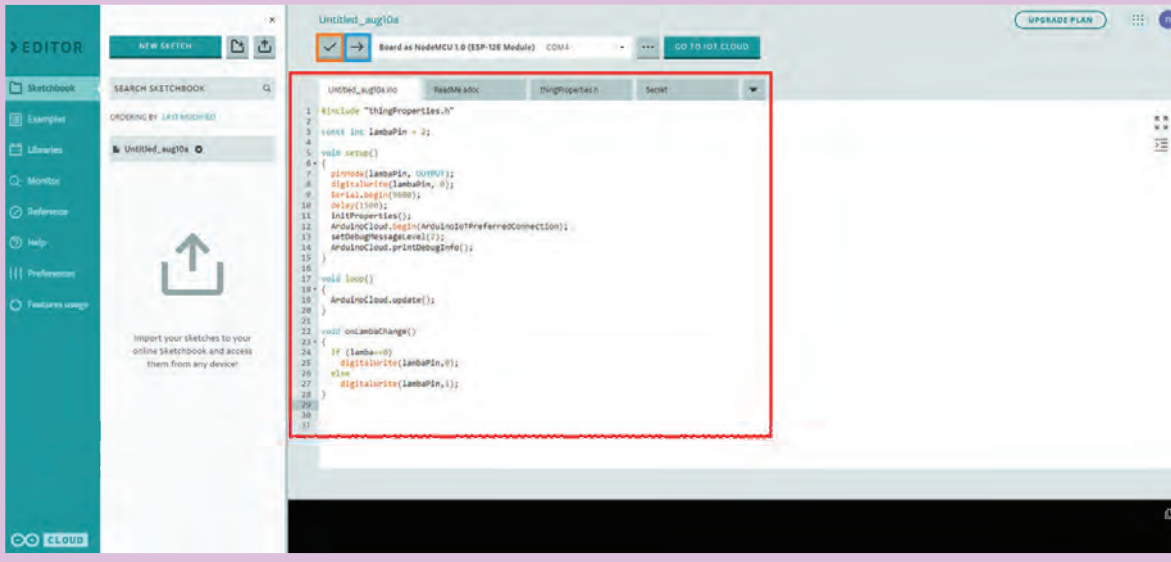
Görsel 7.29: UthingProperties.h sekmesi

25. Adım : Bir diğer kontrol edilmesi gereken sekme ise Secret sekmesidir (Görsel 7.30). Bu sekmede ise hazırlanan IoT cihazının kullanılacak olduğu yerdeki Wi-Fi adı ve şifresi ile birlikte "Device" sekmesinde PDF olarak indirilen Secret Key yer alır. Doğruluğunu programa başlamadan önce kontrol etmelisiniz.



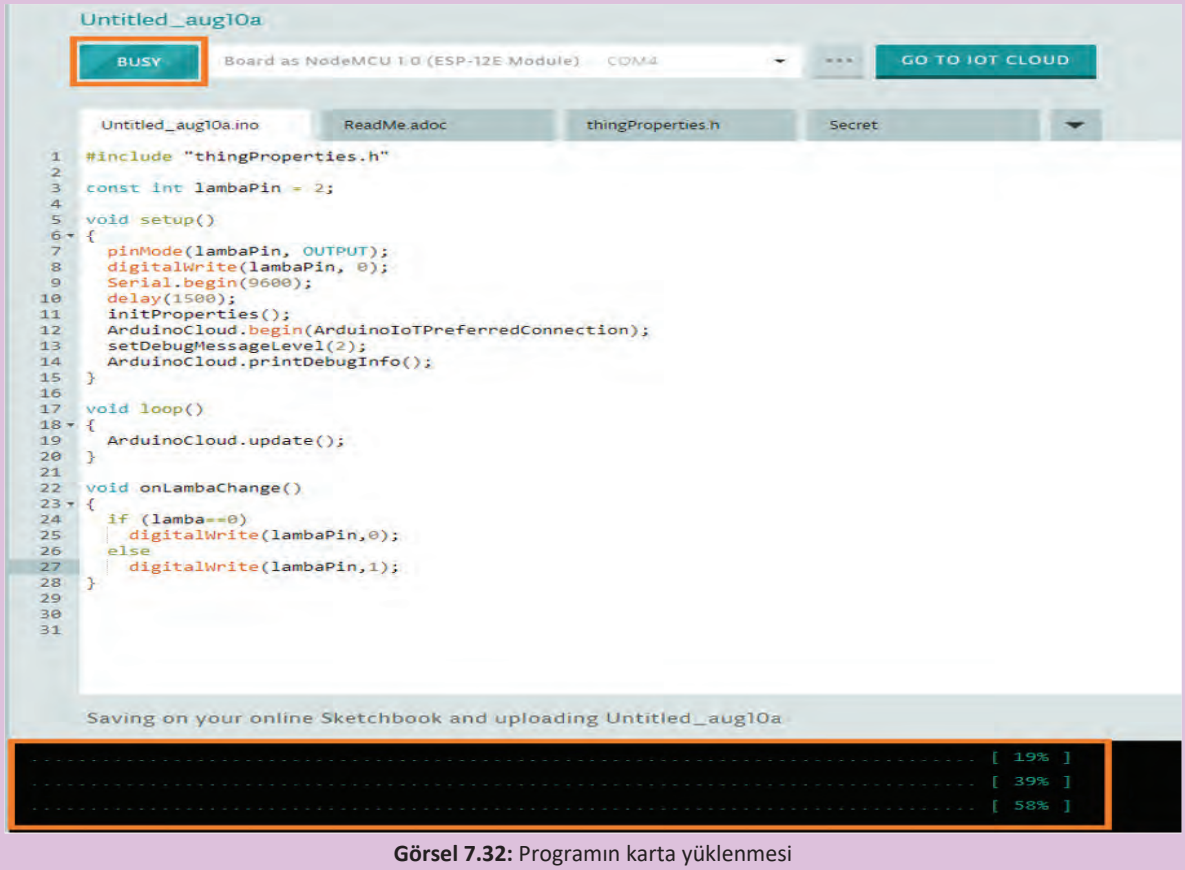
Görsel 7.30: Secret sekmesi

26. Adım : Görsel 7.31'deki programı çalışma alanına yazınız. Öncelikle turuncu kutu içindeki düğmeye basarak çalışmanızın derlenmesi sağlayınız. Böylelikle programda yazım yanlışı olup olmadığını görebilirsiniz. Mavi kutu içindeki buton ile de yapılan çalışmayı hem derleyebilir hem de USB portunda bağlı olan karta programı yüklemeye başlayabilirsiniz.

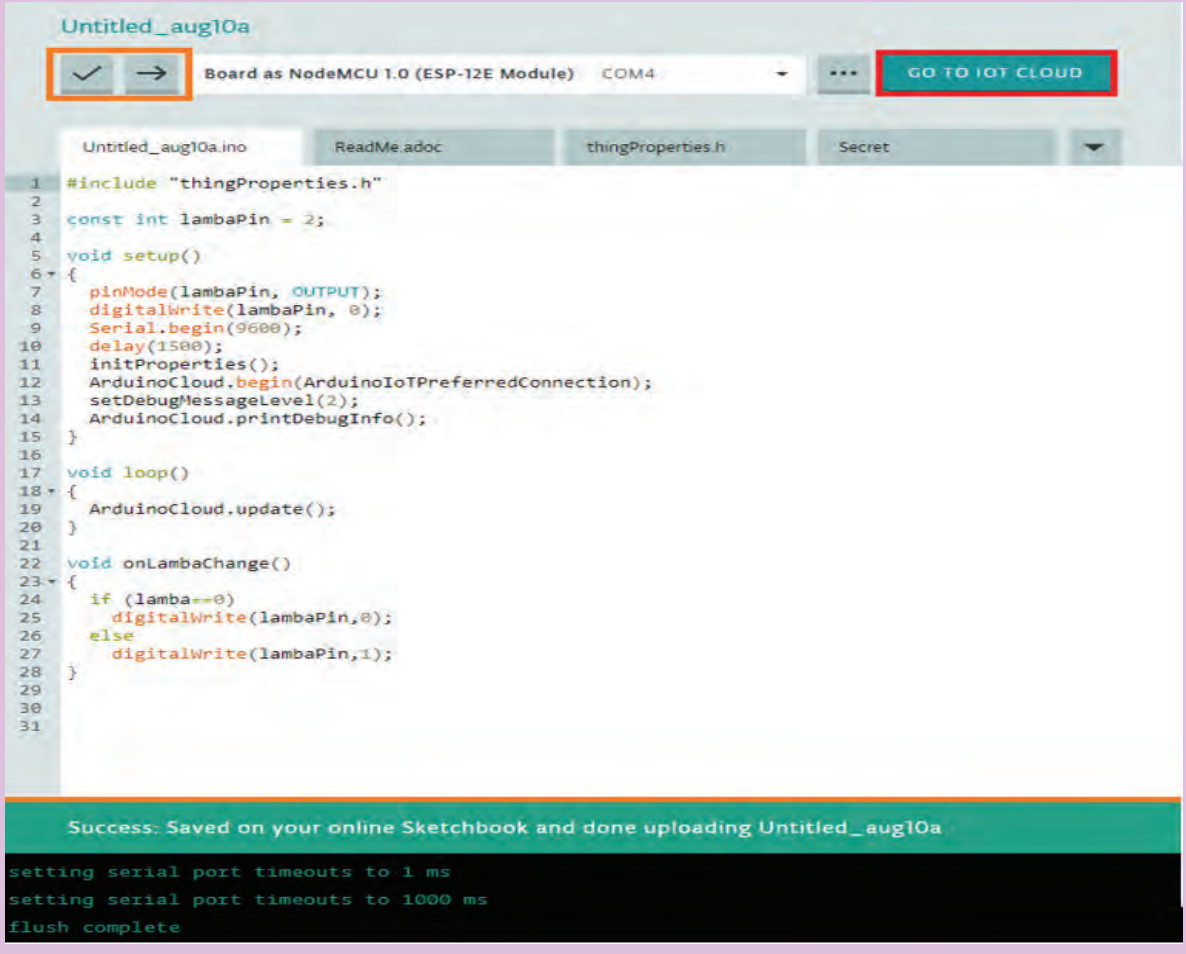


Görsel 7.31: Programın yazılması

27. Adım : Program yüklenmeye devam ederken Görsel 7.32'deki gibi BUSY yazar. Yüklenme tamamlandığında Görsel 7.33'teki ekran karşınıza gelir.

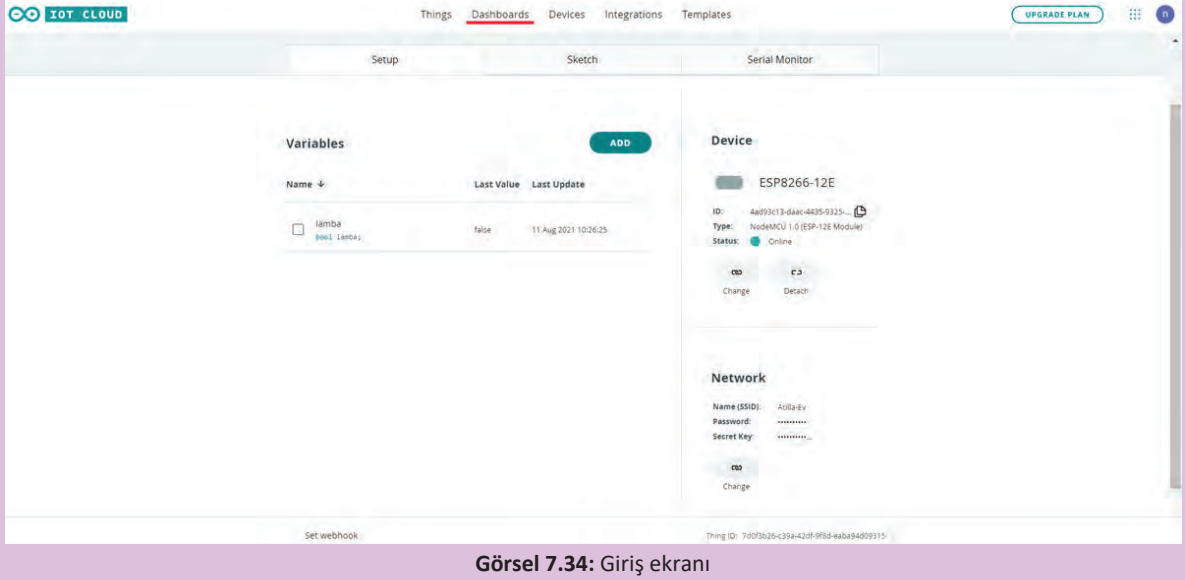


Görsel 7.32: Programın karta yüklenmesi



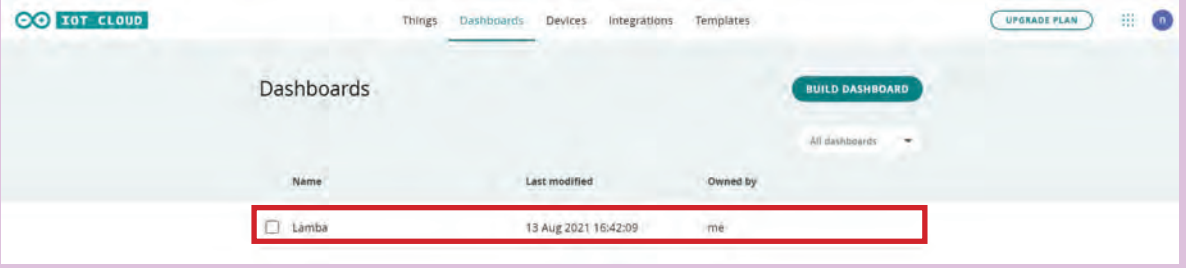
Görsel 7.33: Programın karta yüklenmesi

28. Adım : Görsel 7.33'te yer alan kırmızı kutu içindeki "GO TO IOT CLOUD" düğmesi ile Görsel 7.34'teki giriş ekranına geri dönülür. Görsel 7.33 ve 7.34'teki adımları takip ederek hazırlanan Dashboard'u açınız.



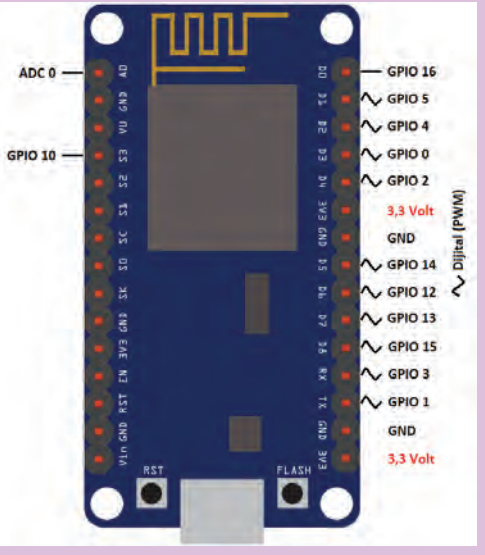
Görsel 7.34: Giriş ekranı

Hazırlanan Dashboard Görsel 7.35'te gösterilmiştir.



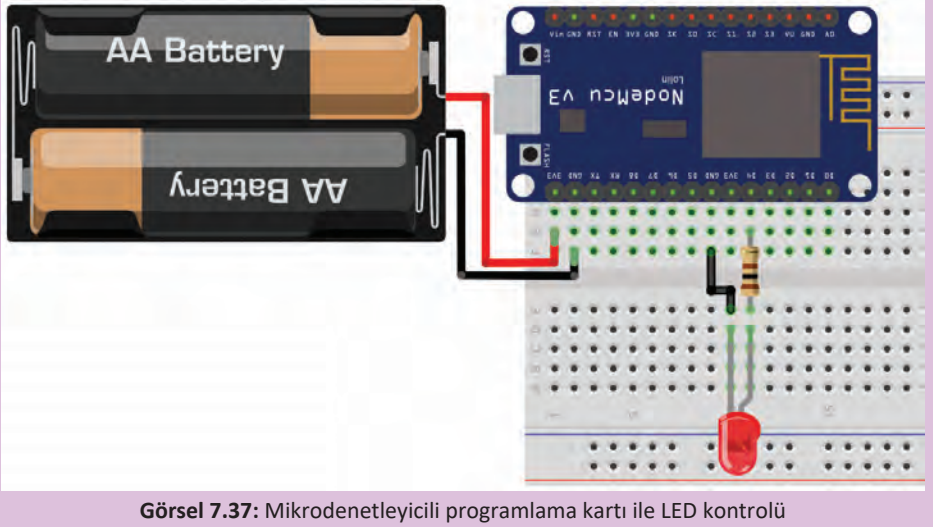
Görsel 7.35: Hazırlanan Dashboard

29. Adım : Mikrodenetleyicili programlama kartının bu yöntemle programlanması sonucunda üzerinde yazan pin numaralarında değişiklik gerçekleşir. Programı yazarken ve devreyi hazırlarken bu pin numaralarına dikkat etmelisiniz (Görsel 7.36).

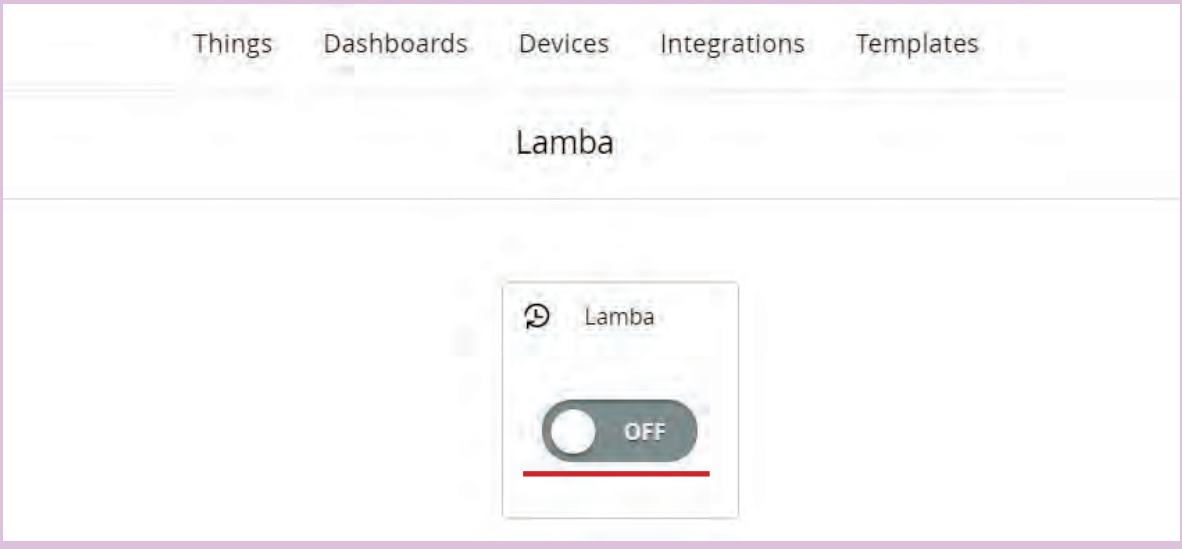


Görsel 7.36: Mikrodenetleyicili programlama kartının pin yapısı

30. Adım : Program yazarken lamba için ayırdığınız 2 numaralı pinin karta program yüklendikten sonra D4 pini yazan yerde olduğu görülmektedir. Görsel 7.37'deki devreyi breadbord üzerinde kurup Görsel 7.38'deki web arayüzünden switch'i açıp kapatarak LED'in durumunu gözlemleyiniz. Bu devre üzerinden lamba kontrolü yapabilmek için LED yerine 2.Öğrenme birimindeki optokuplör devresi ve röle devresi birleştirilerek kullanılabilir. Ancak 220 volt ile çalışmak tehlikeli olduğu için bu devrede lamba LED ile temsil edilmiştir.



Görsel 7.37: Mikrodenetleyicili programlama kartı ile LED kontrolü



Görsel 7.38: Uygulamayı çalıştıracak web arayüzü

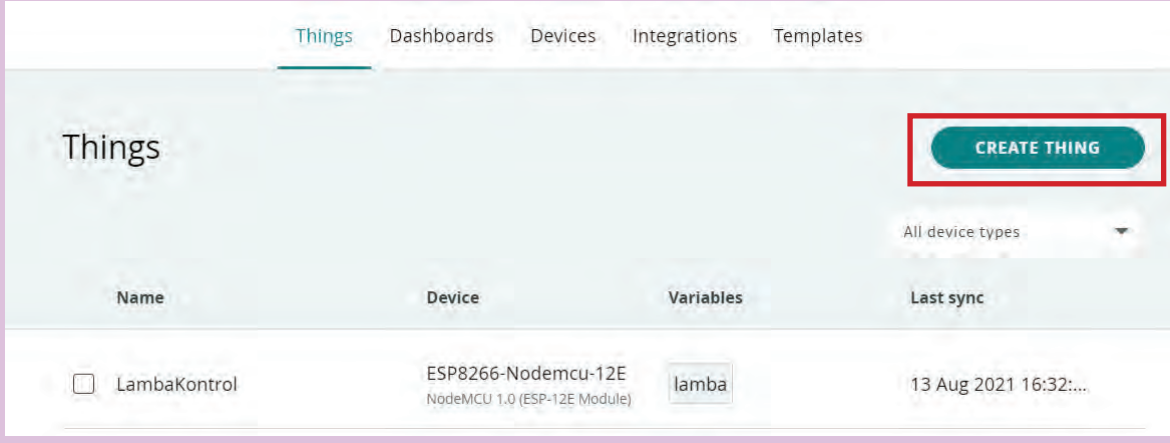
31. Adım : Uygulamayı cep telefonu üzerinden çalıştırabilmek için cep telefonunuzun uygulama mağazasına girip IoT Remote programını indiriniz. Program ilk açıldığında hesabınızı ne şekilde açtıysanız aynı şekilde giriş yaparak Dashboard'unuzu görebilirsiniz (Görsel 7.39).

Görsel 7.39: Uygulamanın hem web hem de mobil arayüzünden çalıştırılması

2. UYGULAMA

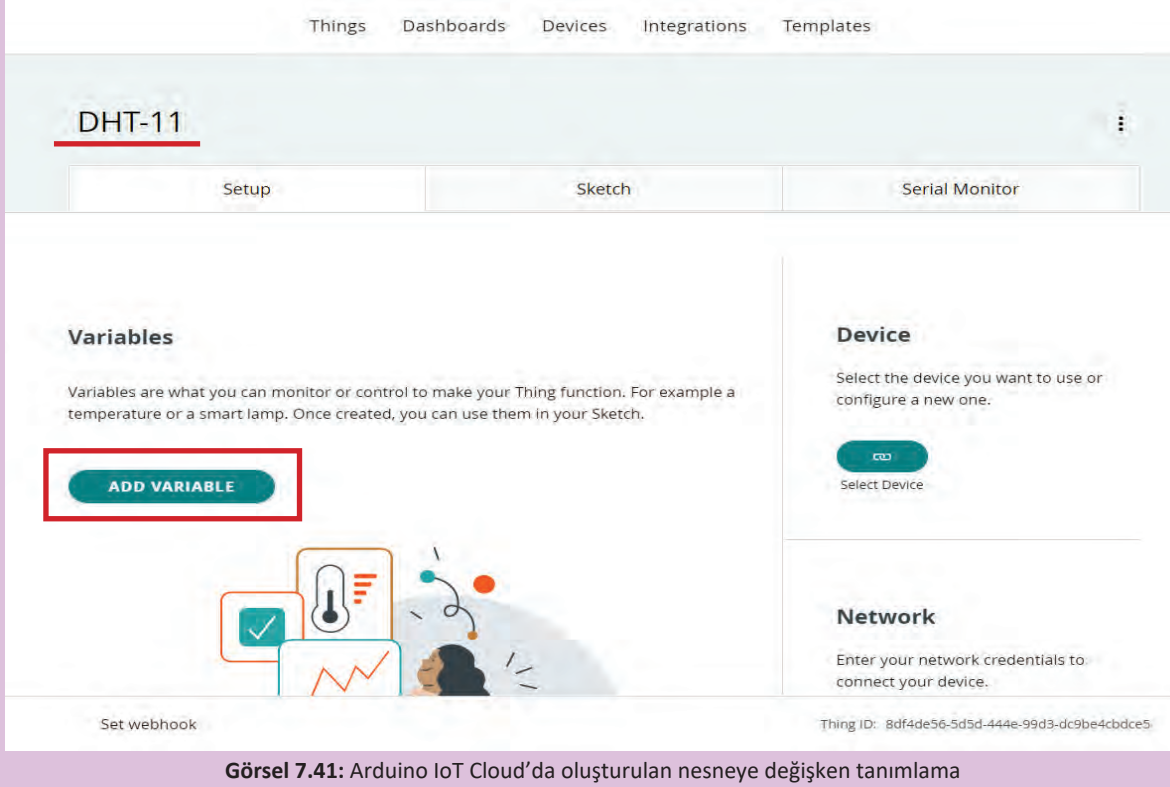
DHT-11 sensörü kullanılarak bulunduğu ortamın sıcaklık ve nem bilgisini Arduino IoT Cloud'a aktaran ve sonucu web sayfası ile mobil uygulama üzerinden takip edilebilecek IoT nesnesini tasarlayınız.

1. Adım : Arduino IoT Cloud'a giriş yapınız. Görsel 7.40'taki sayfadan "CREATE THING" butonuna basarak yeni nesne tasarlama işlemine başlayınız.



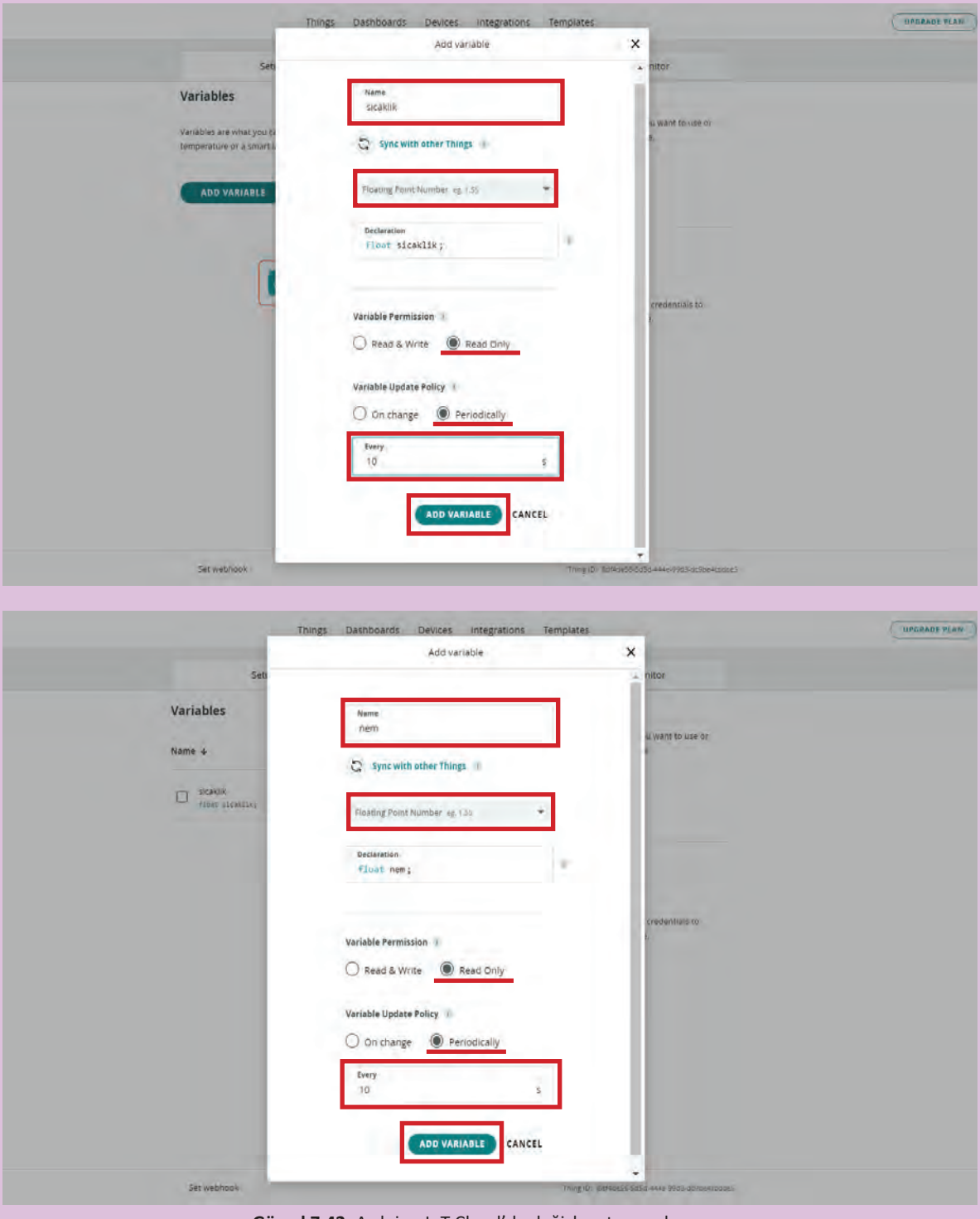
Görsel 7.40: Arduino IoT Cloud'da yeni nesne oluşturma

2. Adım : Açılan sayfada (Görsel 7.41) altı çizili bölümden oluşturulan yeni nesneye isim veriniz. Variables bölümünden oluşturulacak nesnede kaç adet değişken kullanılacak ise "ADD VARIABLE" butonuna basarak ekleyiniz.



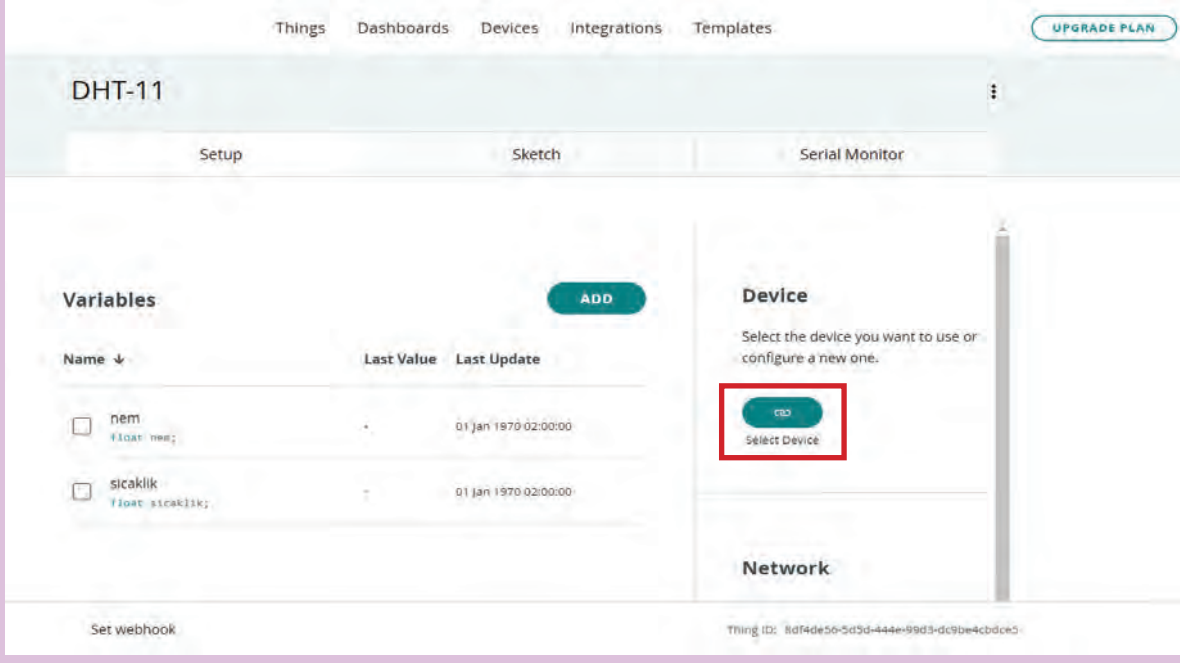
Görsel 7.41: Arduino IoT Cloud'da oluşturulan nesneye değişken tanımlama

3. Adım : Görsel 7.42'de görüldüğü gibi çalışmada iki adet değişken kullanılacaktır. Bu değerler virgüllü olduğundan değişken tipini float olarak tanımlayınız. Tanımlanan bu değişken üzerinden sadece ortam sıcaklığı ve nem bilgileri okunacağından değişken izinlerinden "Read Only"yi tercih ediniz. Değişkenin içinde yer alan verinin 10 saniyede bir güncellenmesi için "Periodically"i tercih edip "Every" bölümünün içine 10 yazınız. Bu düzenlemeleri tamamladıktan sonra "ADD VARIABLE" butonuna basarak değişkeni nesneye ekleyiniz.



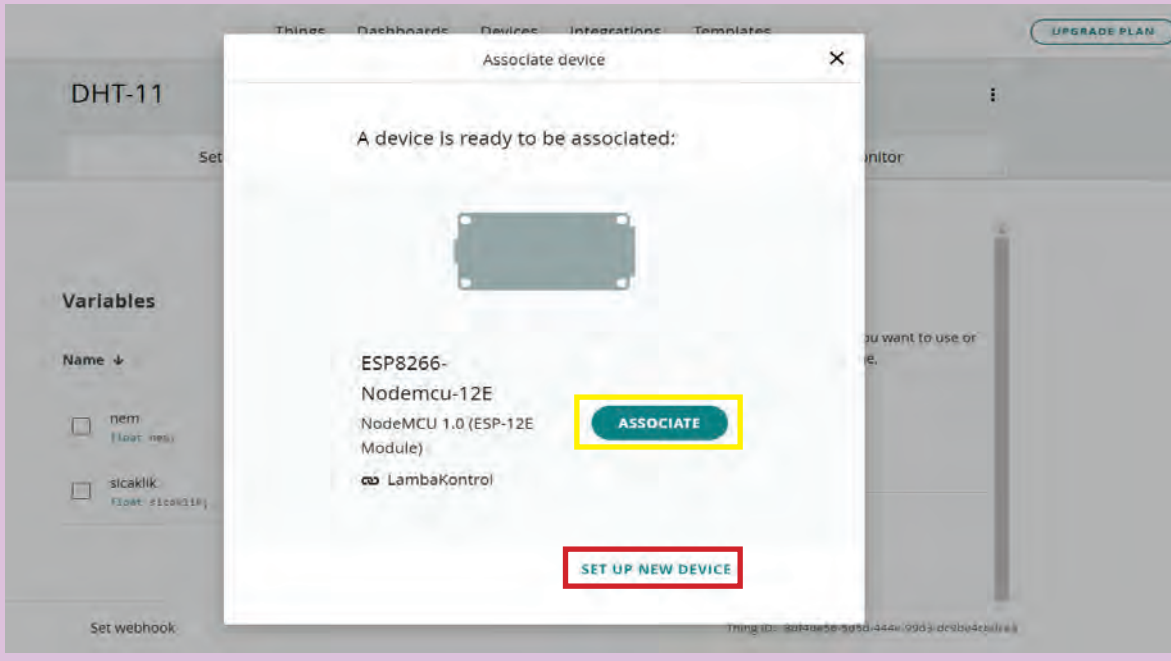
Görsel 7.42: Arduino IoT Cloud'da değişken tamamlanması

4. Adım : Değişken tanımlama işlemini tamamladıktan sonra oluşturacağınız nesnenin donanım olarak hangi alt yapıyı kullanacağını belirleyiniz. Bunu belirleyebilmek için Görsel 7.43'teki sayfada "Device" bölümündeki "Select Device" butonuna basınız.



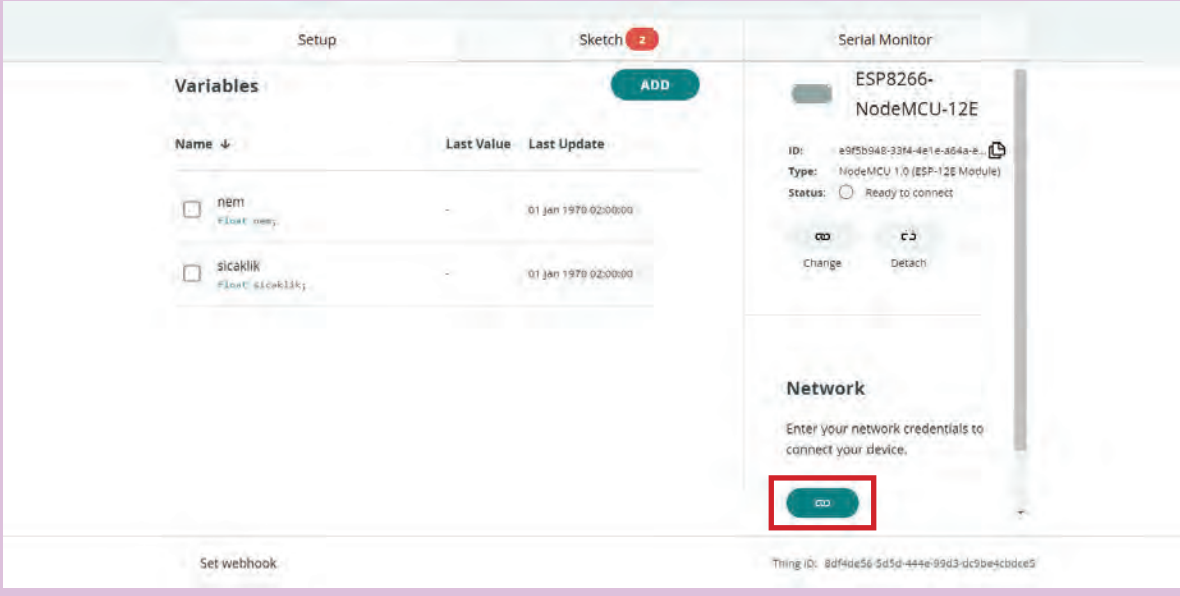
Görsel 7.43: Arduino IoT Cloud'da mikrodenetleyicili uygulama kartı tamamlanması

5. Adım : Görsel 7.44'te görüldüğü gibi daha önceki çalışma için hazırlanmış kart bilgisini bu çalışmada da kullanmak için "ASSOCIATE" butonuna basınız. Ancak daha önceki çalışmanızın üzerine bu işlemleri yapacağından önceki çalışma bu işlemden sonra çalışmayabilir. Bu nedenle "SET UP NEW DEVICE" butonuna basarak 1. Uygulamadaki kart tanımlama adımlarını takip ediniz. Görsel 7.9'dan Görsel 7.14'e kadar olan adımları uygulayınız.



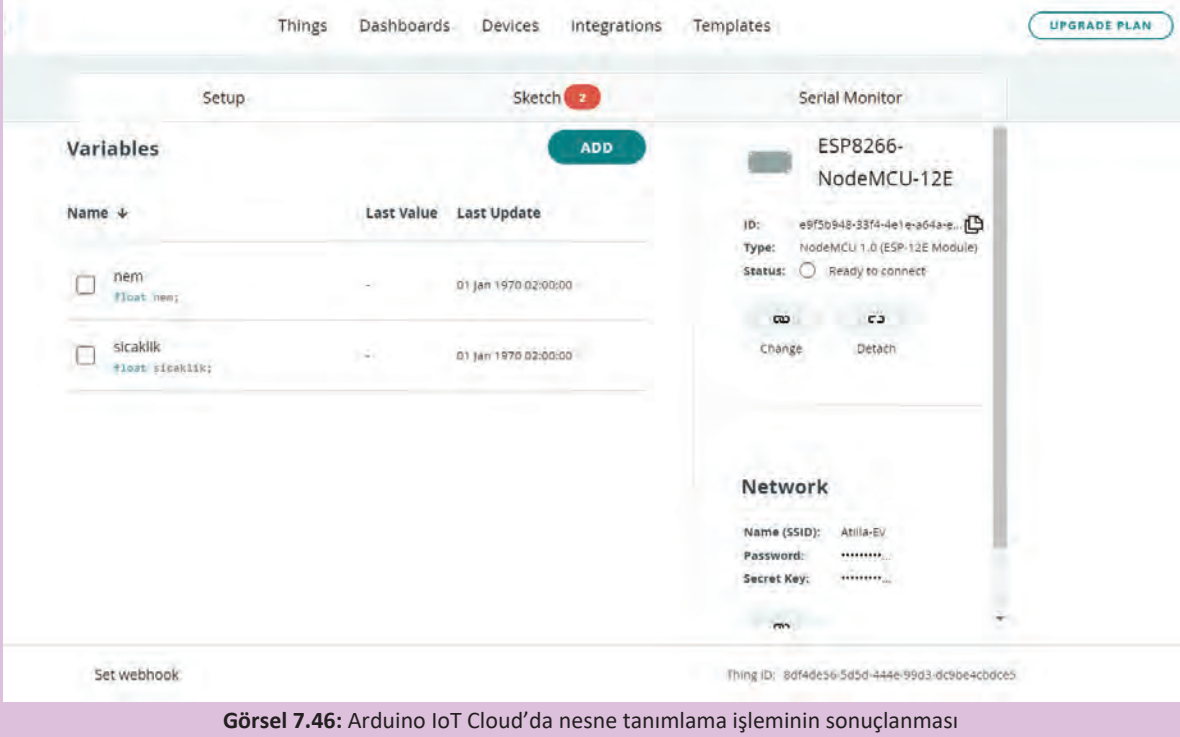
Görsel 7.44: Arduino IoT Cloud'da mikrodenetleyicili uygulama kartı tamamlanması

6. Adım : Kart tanımlama işlemini tamamladıktan sonra "Network" bölümünden "Configure" butonuna basınız (Görsel 7.45). Açılan ekranda donanımın kullanılacak olduğu yerdeki ağın adı (Name(SSID)), şifresi (Password) ve bilgisayarınızın "Device" bölümünde kaydetmiş olduğunuz kod (Secret Key) bilgilerini giriniz. Bu bilgileri doğru olarak girdiğinize emin olduktan sonra "SAVE" butonuna basarak işlemi tamamlayınız.



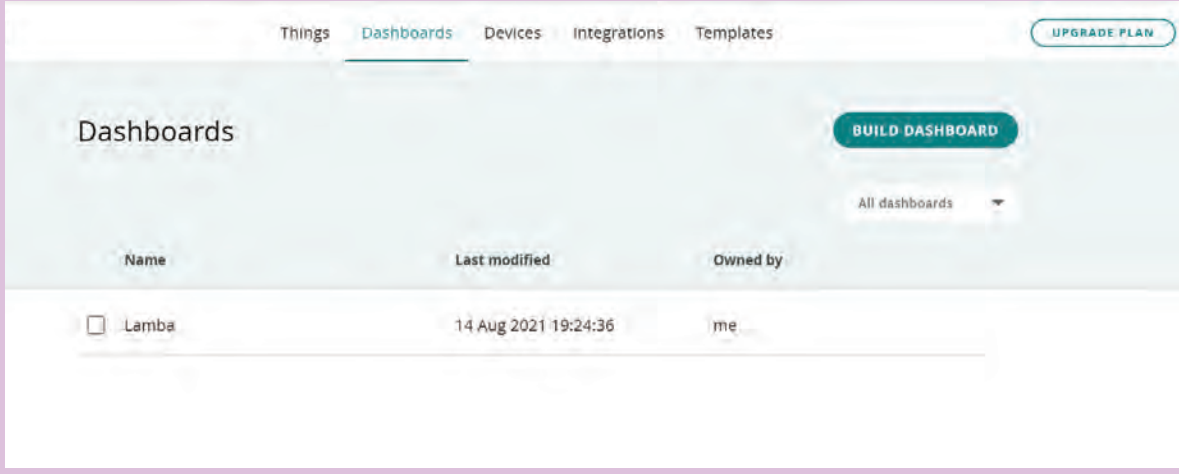
Görsel 7.45: Arduino IoT Cloud'da oluşturulan nesnenin ağ ayarları

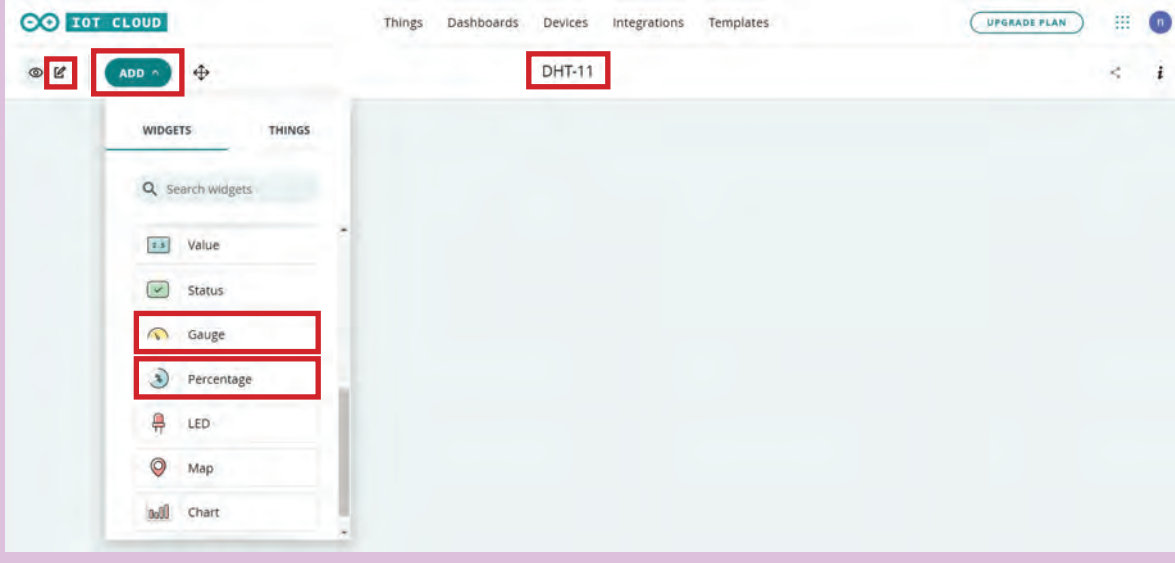
7. Adım : Ağ tanımlaması yapıldıktan sonra nesne tanımlaması tamamlanmış olur. Yapılan tüm ayarların özeti Görsel 7.46'da olduğu gibi bu ekranda görülür.

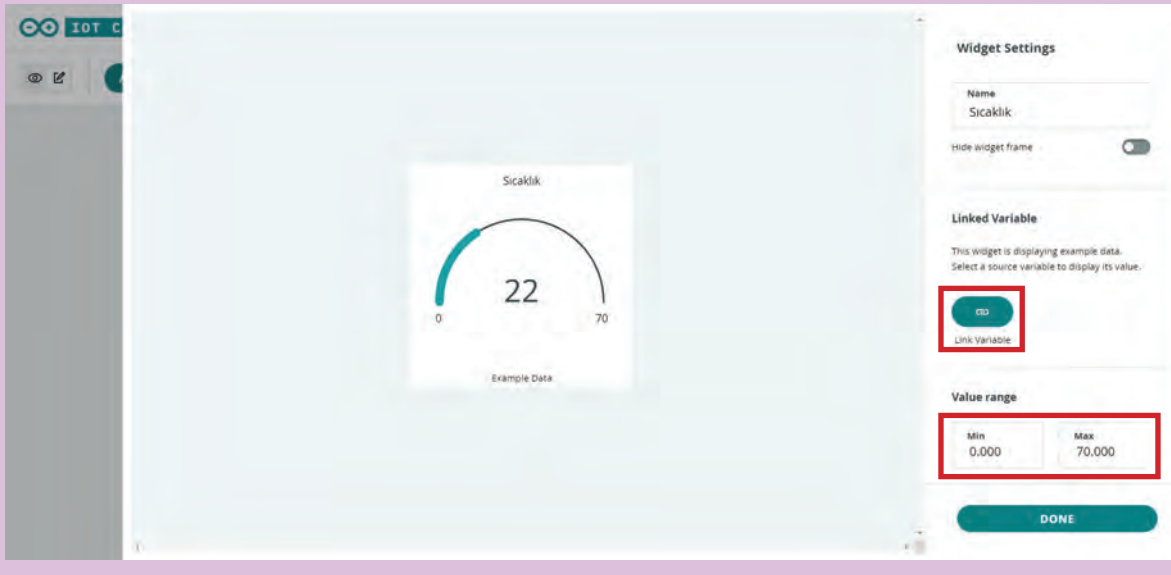


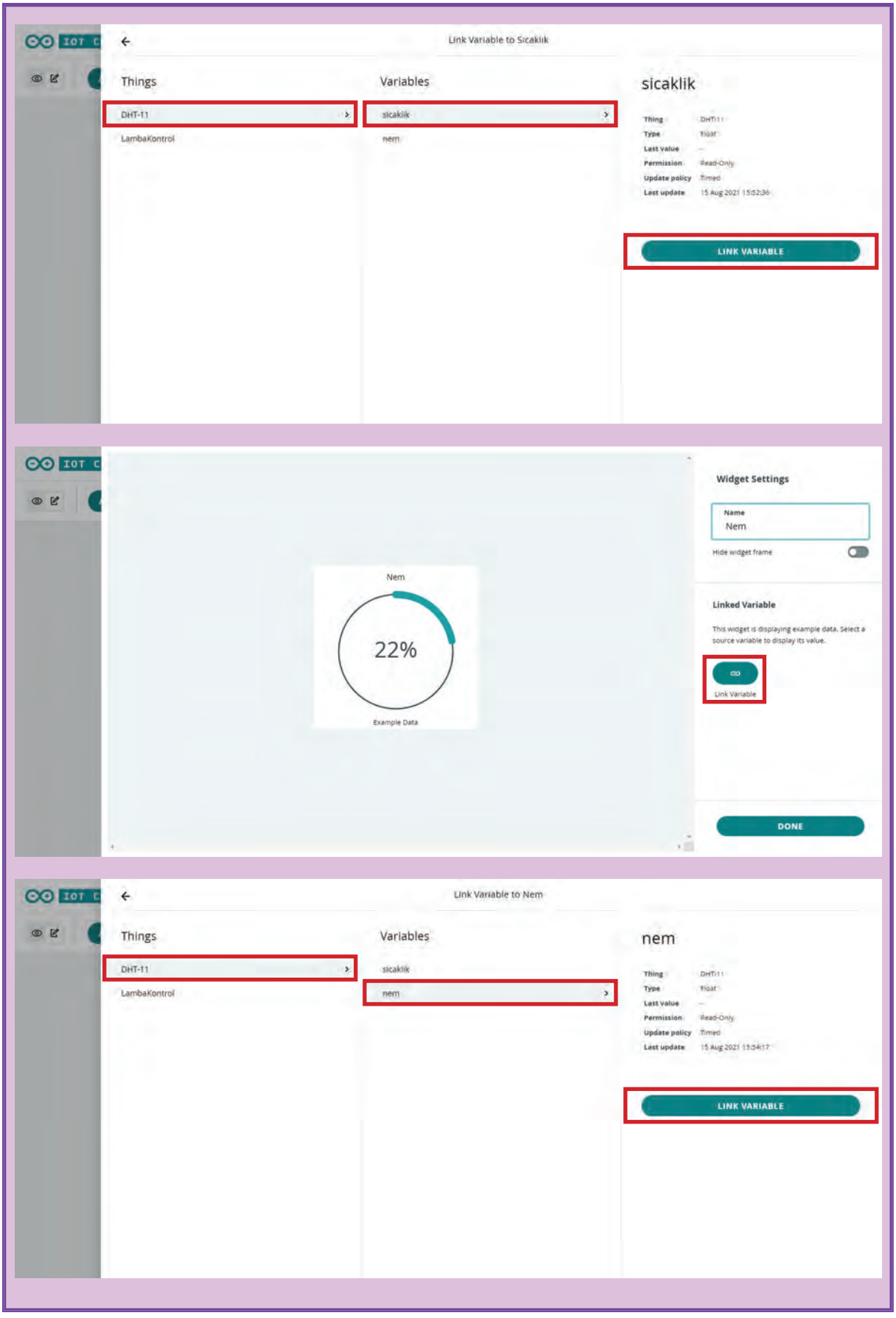
Görsel 7.46: Arduino IoT Cloud'da nesne tanımlama işleminin sonuçlanması

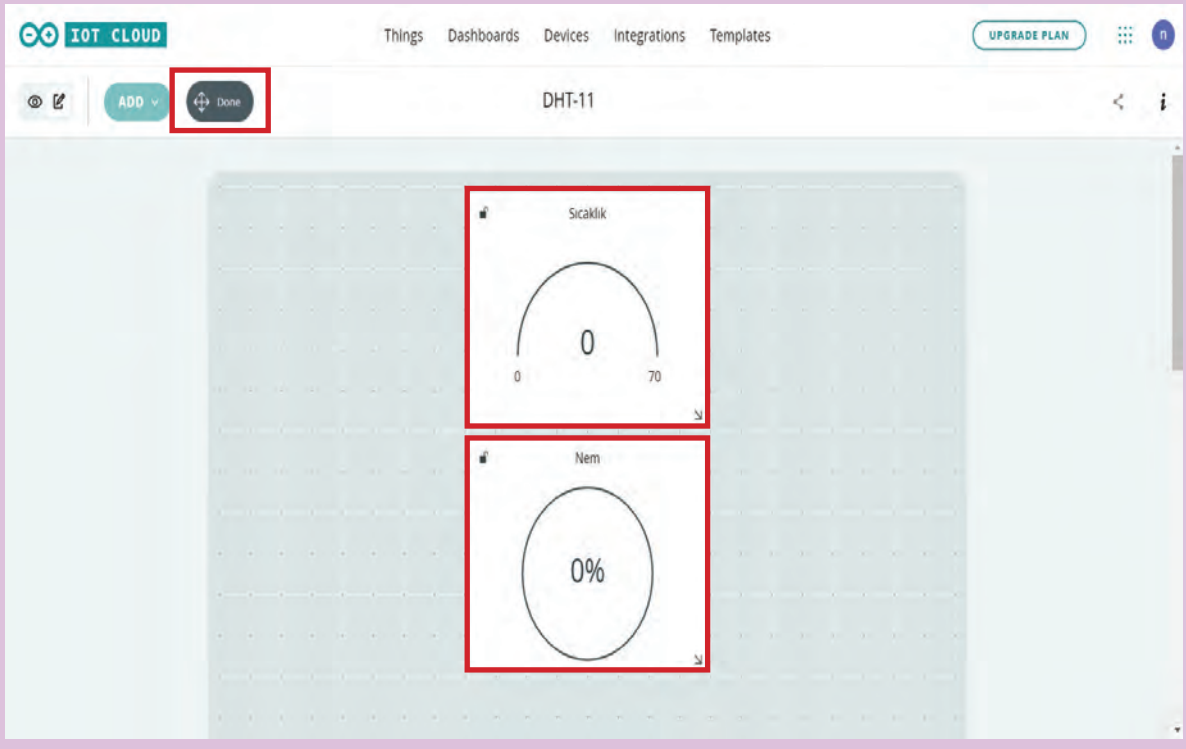
8. Adım : Kullanıcının göreceği arayüzü tanımlamak için Görsel 7.47'deki işlemler dizisini takip ediniz.





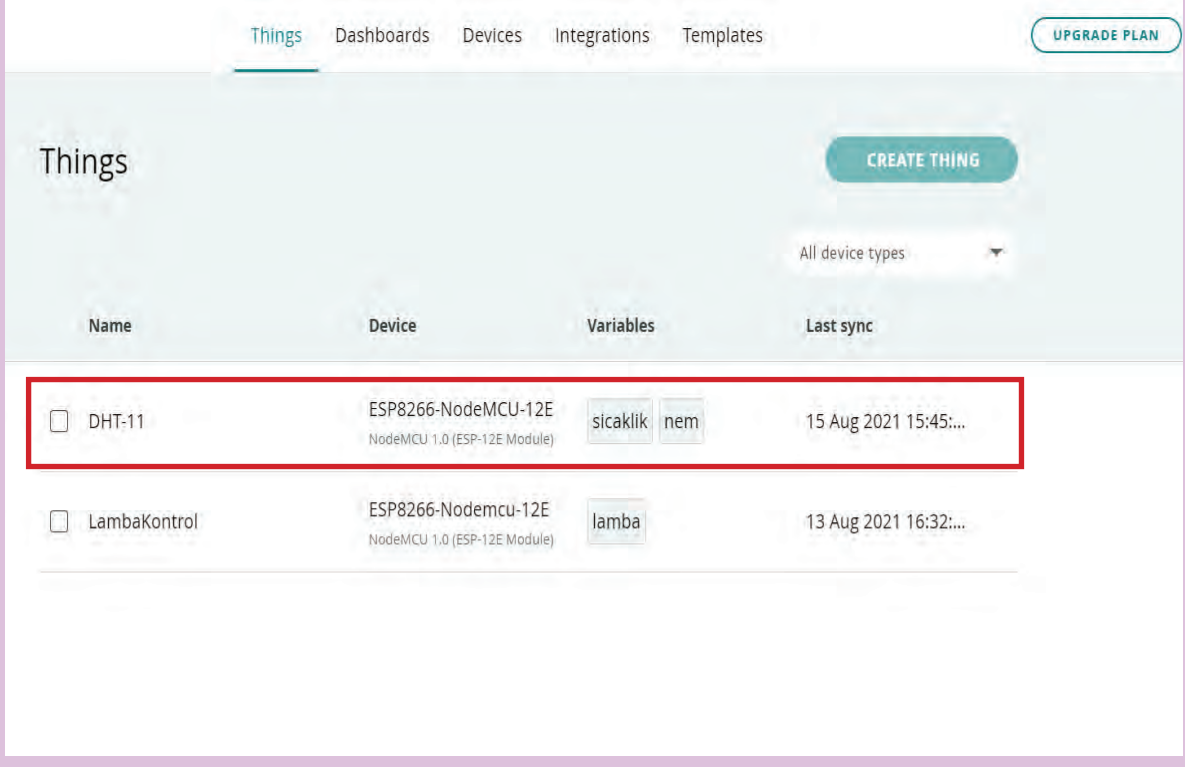



Link Variable to Sicaklik
Things
Variables
DHT-11
LambaKontrol
sicaklik
nem
sicaklik
Thing
Type
Last value
Permission
Read-Only
Update policy
Timed
Last update
15 Aug 2021 15:52:36
LINK VARIABLE
Nem
22%
Example Data
Widget Settings
Name
Nem
Hide widget frame
Linked Variable
This widget is displaying example data. Select a source variable to display its value.
Link Variable
DONE
Link Variable to Nem
Things
Variables
DHT-11
LambaKontrol
sicaklik
nem
nem
Thing
Type
Last value
Permission
Read-Only
Update policy
Timed
Last update
15 Aug 2021 15:54:17
LINK VARIABLE



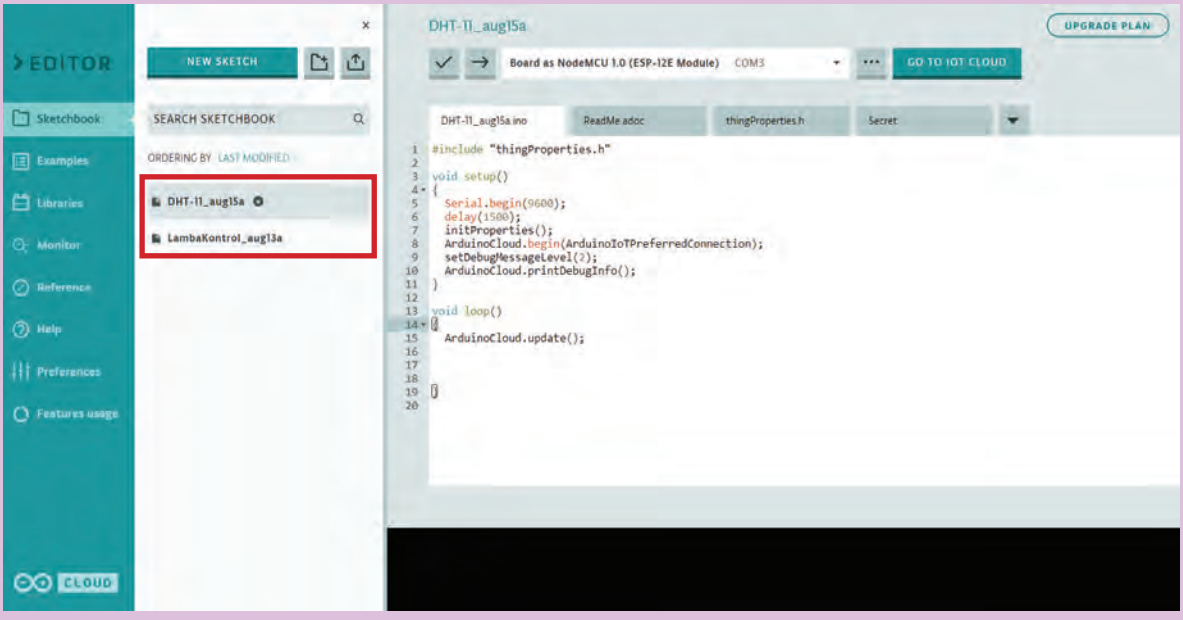
Görsel 7.47: Uygulamada kullanılacak dashboard tasarımı işlem dizisi

9. Adım : Arayüz tasarımını tamamladıktan sonra "Things" sekmesine geçiniz. Bu sekme içinde DHT-11 adı verilmiş olan bölüme tıklayarak bu nesneyi seçiniz (Görsel 7.48).



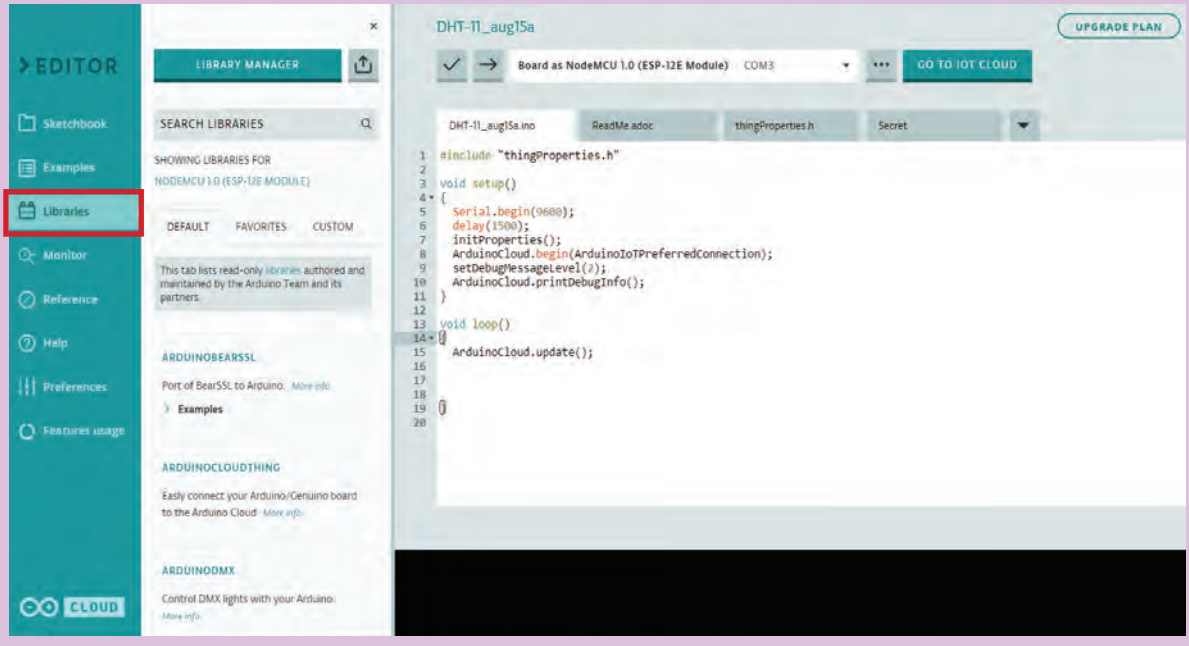
Görsel 7.48: Uygulamada kullanılacak dashboard tasarımı işlem dizisi

10. Adım : Açılan sayfada "Sketch" sekmesinden "Full Editor" bölümüne basarak program yazma editörüne ulaşınız. Programdaki açıklama satırlarını silerek daha sade bir görünüme kavuşturabilirsiniz. Görsel 7.49'daki seçili alan incelendiğinde daha önce oluşturulan çalışmanın programını görebilirsiniz. Program yazarken DHT-11 yazılı olan programın seçili olduğuna dikkat ediniz. Mikrodenetleyicili uygulama kartını bilgisayara bağlayıp ilgili portu seçerek programlamaya hazır hâle getiriniz.



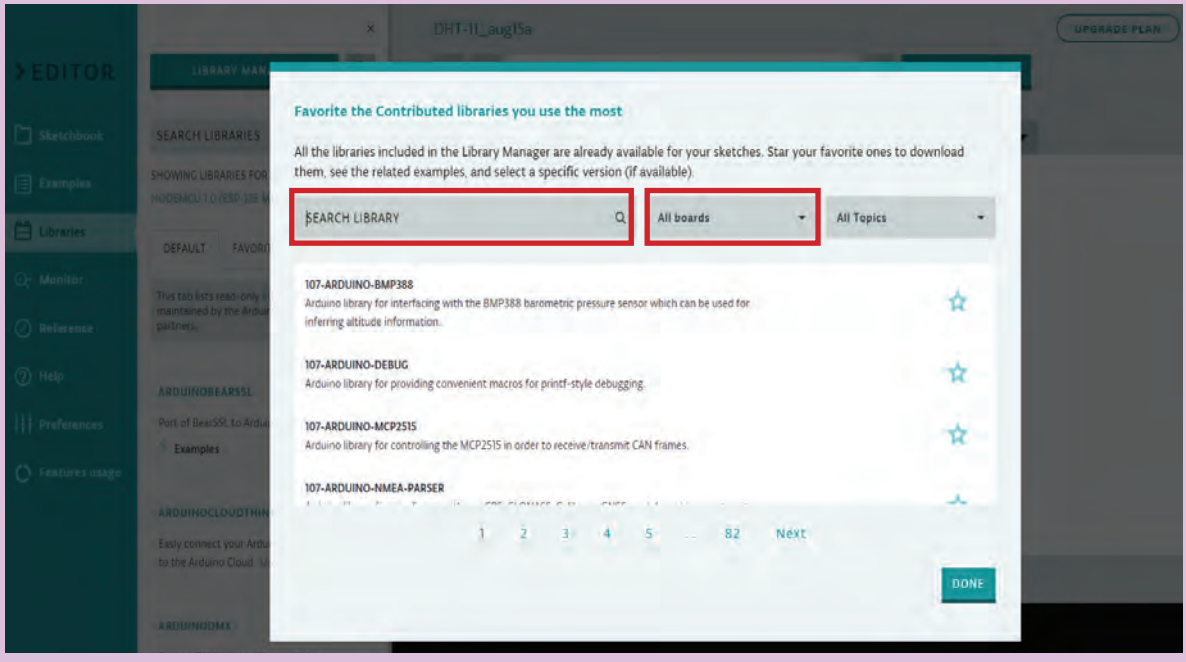
Görsel 7.49: Programın yazılması

11. Adım : DHT-11 sensörünün kullanılabilmesi için yazılan programa DHT-11 sensörünün kütüphanesinin eklenmesi gerekir. Kütüphane eklemek için Görsel 7.50'deki "Libraries" sekmesine geçiniz ve "LIBRARY MANAGER" butonuna basınız.



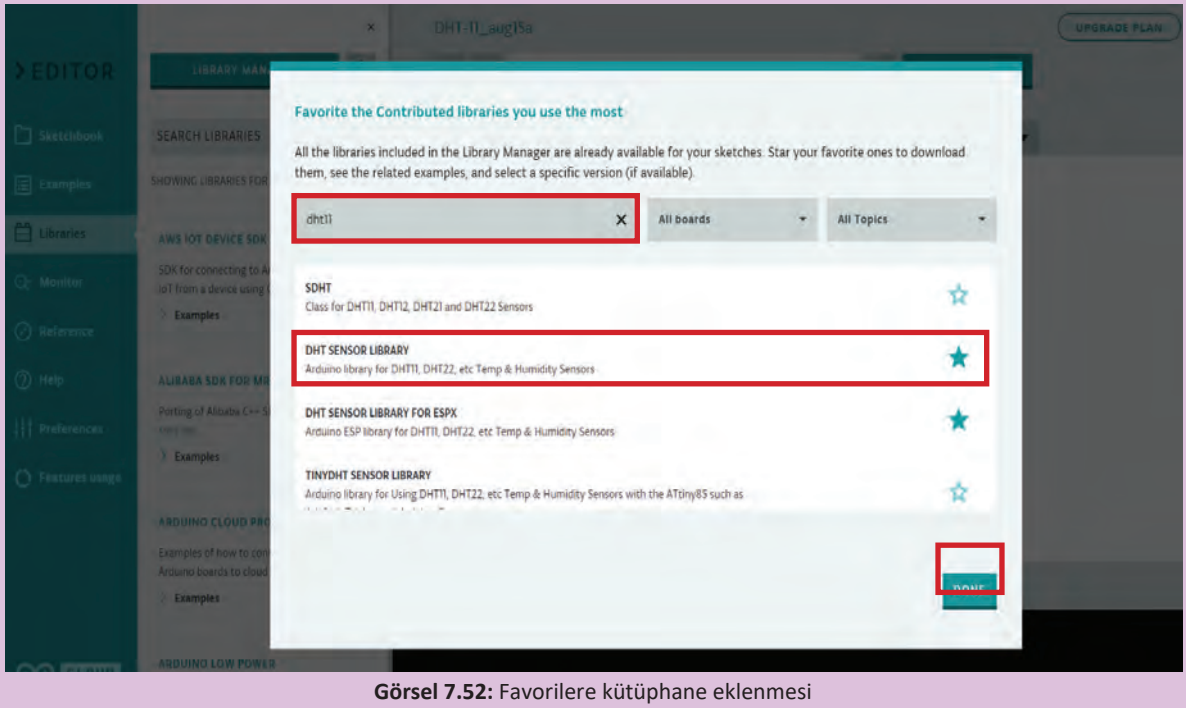
Görsel 7.50: Programa kütüphane eklenmesi

12. Adım : Açılan sayfa incelendiğinde piyasada bulunan neredeyse bütün sensörlerin kütüphanelerinin burada bulunduğunu görebilirsiniz. Eksik olan kütüphaneler ise hızlı bir şekilde eklenmeye devam eder. "SEARCH LIBRARY" bölümüne aramak istediğiniz sensörün ismini, "All boards" bölümünden ise istediğiniz mikrodenetleyicili programlama kartını seçebilirsiniz (Görsel 7.51).



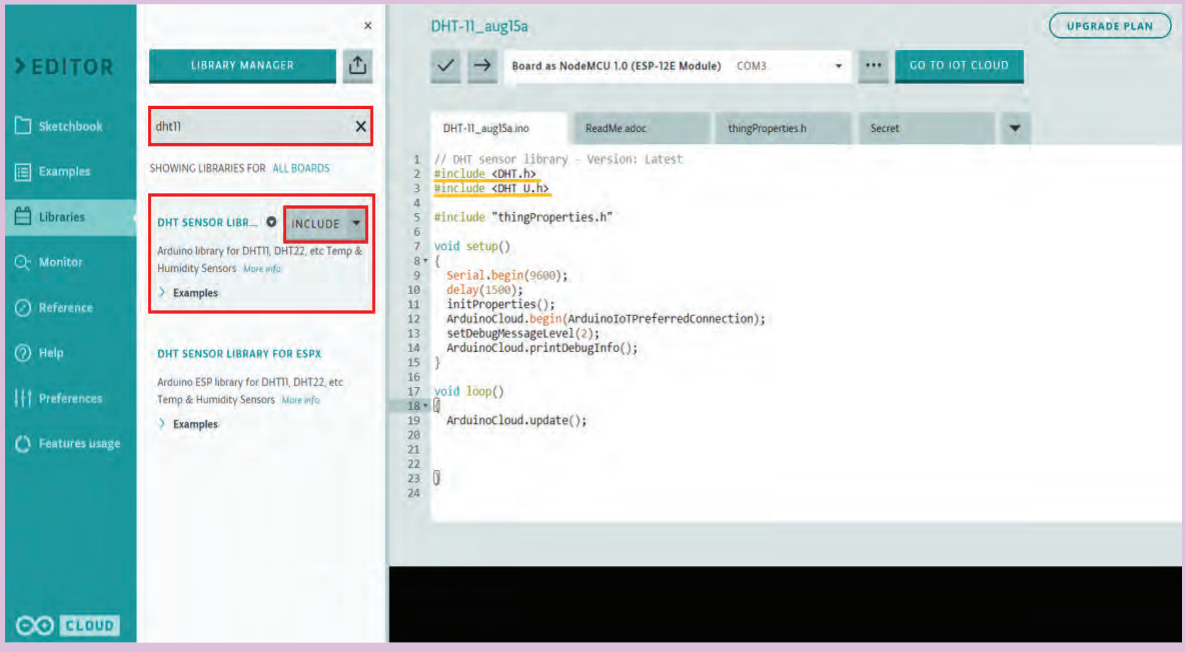
Görsel 7.51: Kütüphane içinde sensör arama

13. Adım : "SEARCH LIBRARY" bölümüne "dht11" yazıp arama işlemi yaptığınızda gelen sonuçlardan "DHT SENSOR LIBRARY" seçeneğinin sonundaki yıldızı seçerek favorilere kütüphaneyi ekleyiniz. "Done" butonuna basılarak arama ekranını kapatınız (Görsel 7.52).



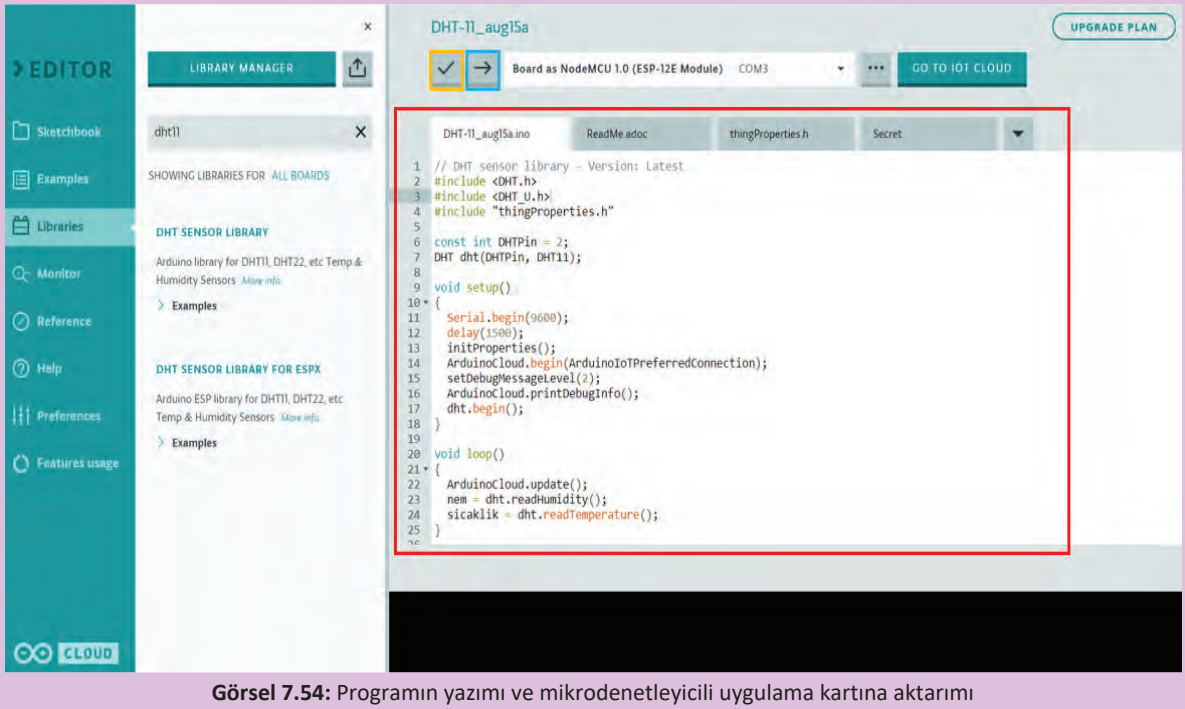
Görsel 7.52: Favorilere kütüphane eklenmesi

14. Adım : Program editöründe "Libraries" sekmesinde iken "SEARCH LIBRARIES" bölümüne dht11 yazarak sonuçları listeleyiniz. Bu sonuçlar içinde favorilere eklenen "DHT SENSOR LIBRARY" seçeneğini seçip "INCLUDE" butonuna basınız ve iki kütüphanenin program içine dâhil olduğunu gözlemleyiniz (Görsel 7.53).



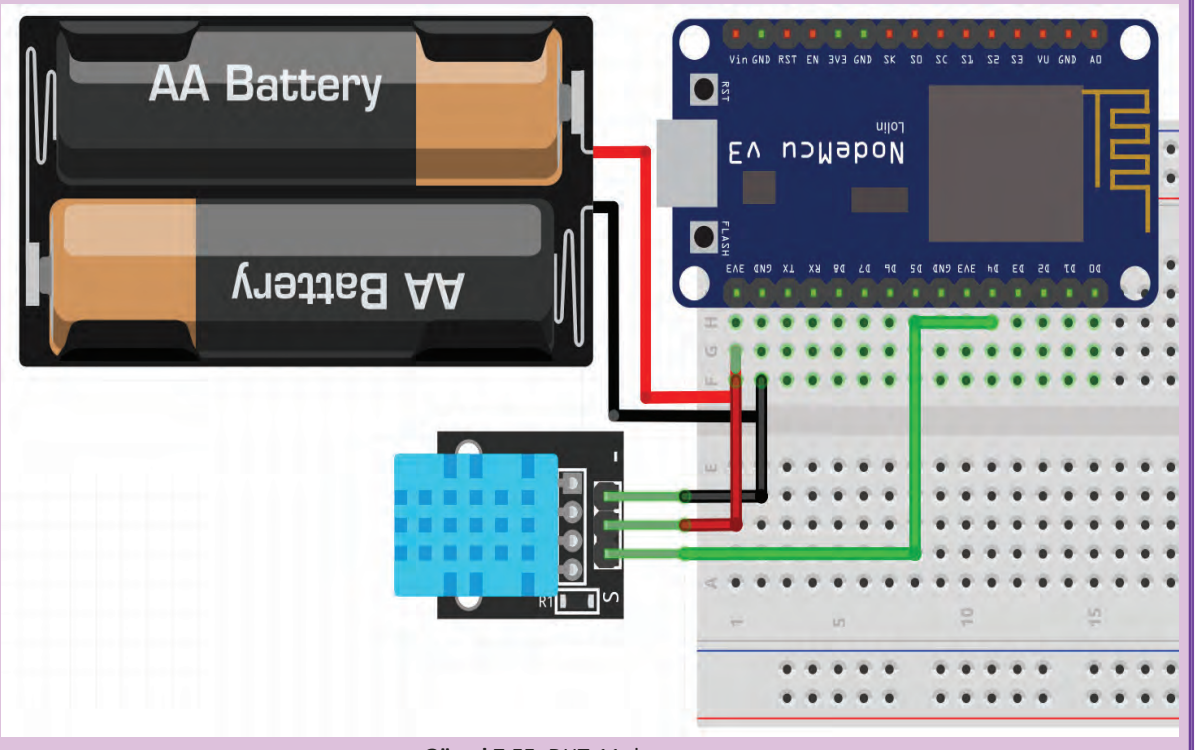
Görsel 7.53: Programa kütüphane eklenmesi

15. Adım : Görsel 7.54'te yer alan kırmızı çerçeve içindeki programı yazınız. Yazma işlemi tamamlandıktan sonra turuncu kutu içindeki butona basarak programın derlenmesini sağlayınız. Hata var ise bu sırada size hatanın nerede olduğu gösterilir. Hata yok ise mavi kutu içindeki butona basarak yazılan programı kart içine gönderiniz.

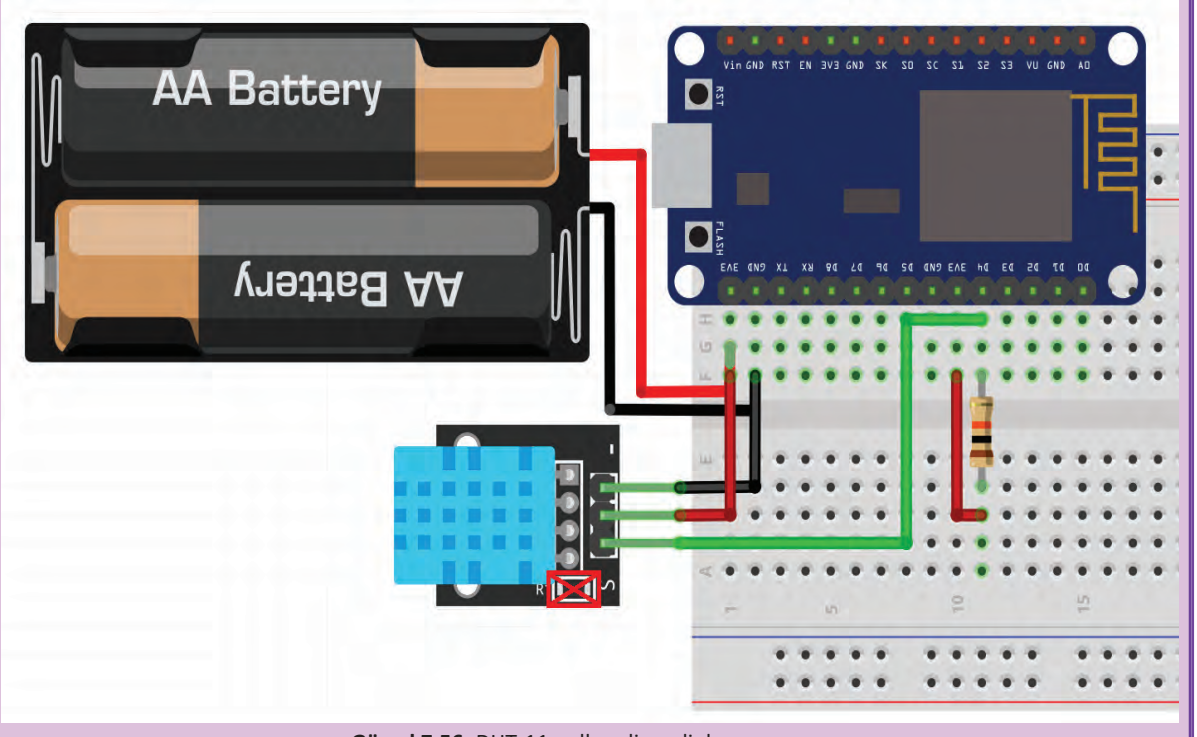


Görsel 7.54: Programın yazımı ve mikrodenetleyicili uygulama kartına aktarımı

16. Adım : Görsel 7.55'teki devre şemasını breadboarda kurunuz. Devre kurulumunu yaparken DHT-11 sensörünüzün üzerinde Görsel 7.55'teki gibi direnç yok ise veri hattına Görsel 7.56'daki gibi 10KΩ'luk pullup direnci bağlamalısınız.

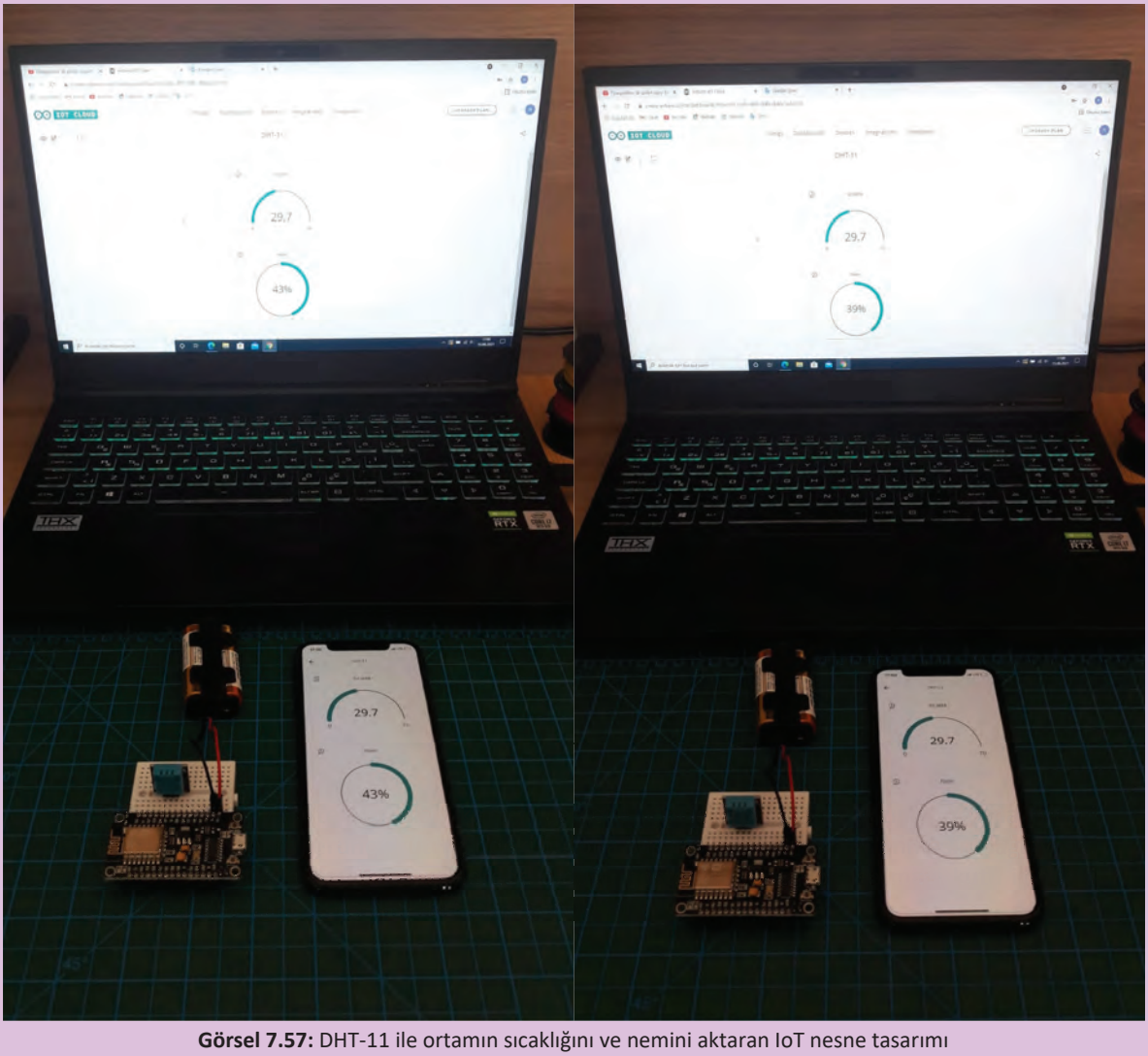


Görsel 7.55: DHT-11 devre şeması



Görsel 7.56: DHT-11 pullup dirençli devre şeması

17. Adım : Devreyi kurup çalıştırdığınızda Görsel 7.57'deki gibi bir sonuç elde edersiniz.



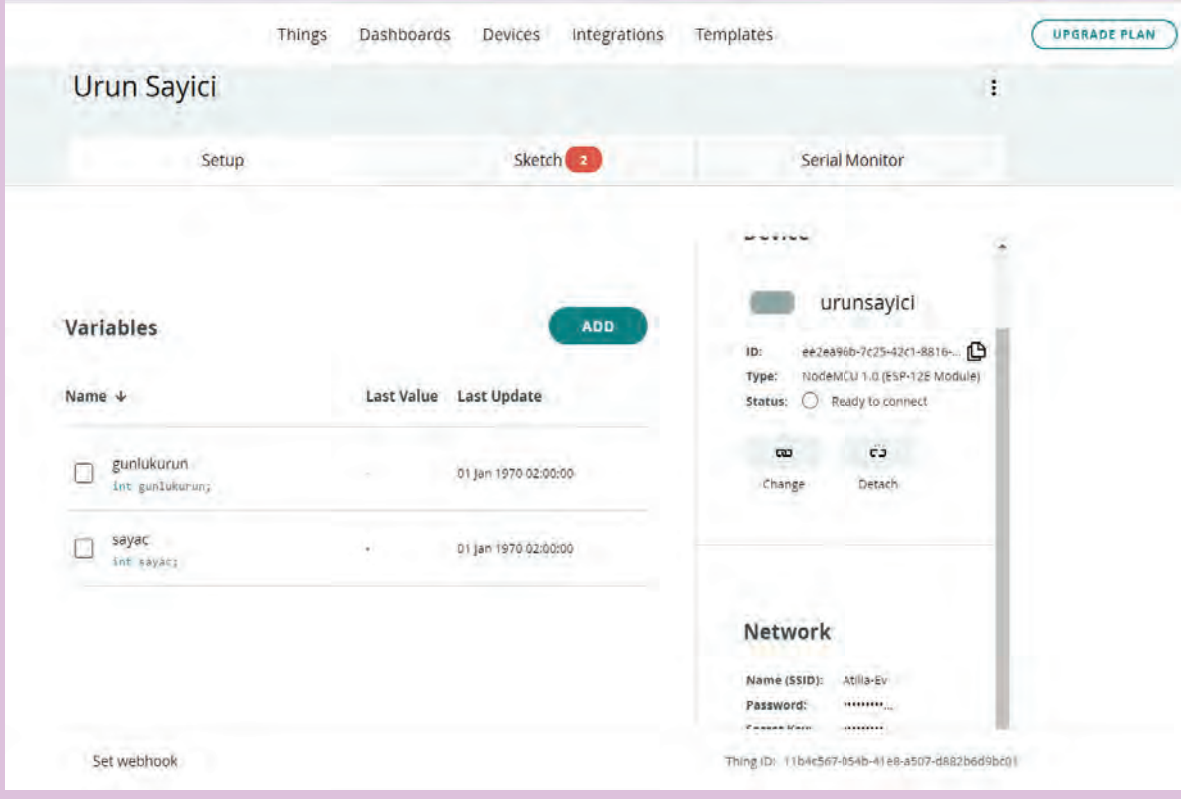
Görsel 7.57: DHT-11 ile ortamın sıcaklığını ve nemini aktaran IoT nesne tasarımı

3. UYGULAMA

Kişiye özel plastik ürünler üreten bir işletmenin üretim hattında hata oluşması sonucu üretim bandı durmaktadır. İşletme sahibi bu durumun önüne geçmek için bandın durumunu ve anlık olarak kaç ürün üretildiğini gösteren bir tasarım istemektedir.

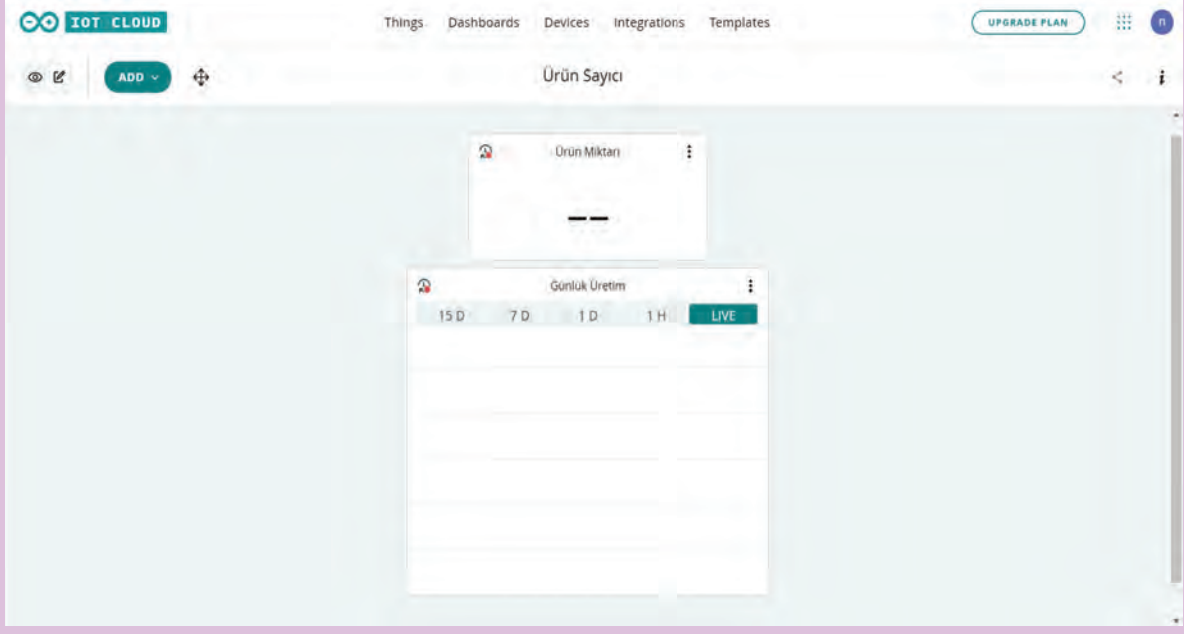
Banttan üretilen ürünleri saymak için KY-032 türünde bir adet infrared alıcı ve verici ile sensör önünden geçen nesneler sayılacaktır. Bu bilgiyi sıfırlamak için ise sistem üzerinde yer alan buton kullanılarak istenilen zamanda bu sayma işlemi sıfırlanabilecektir. Bu tasarımı IoT nesnesi olarak tasarlayınız.

1. Adım : Görsel 7.58'deki gibi değişkenler tanımlayıp sistemde kullanılacak kartı seçiniz. Sistemin kullanılacak olduğu ortamın Wi-Fi bilgilerini kaydediniz.



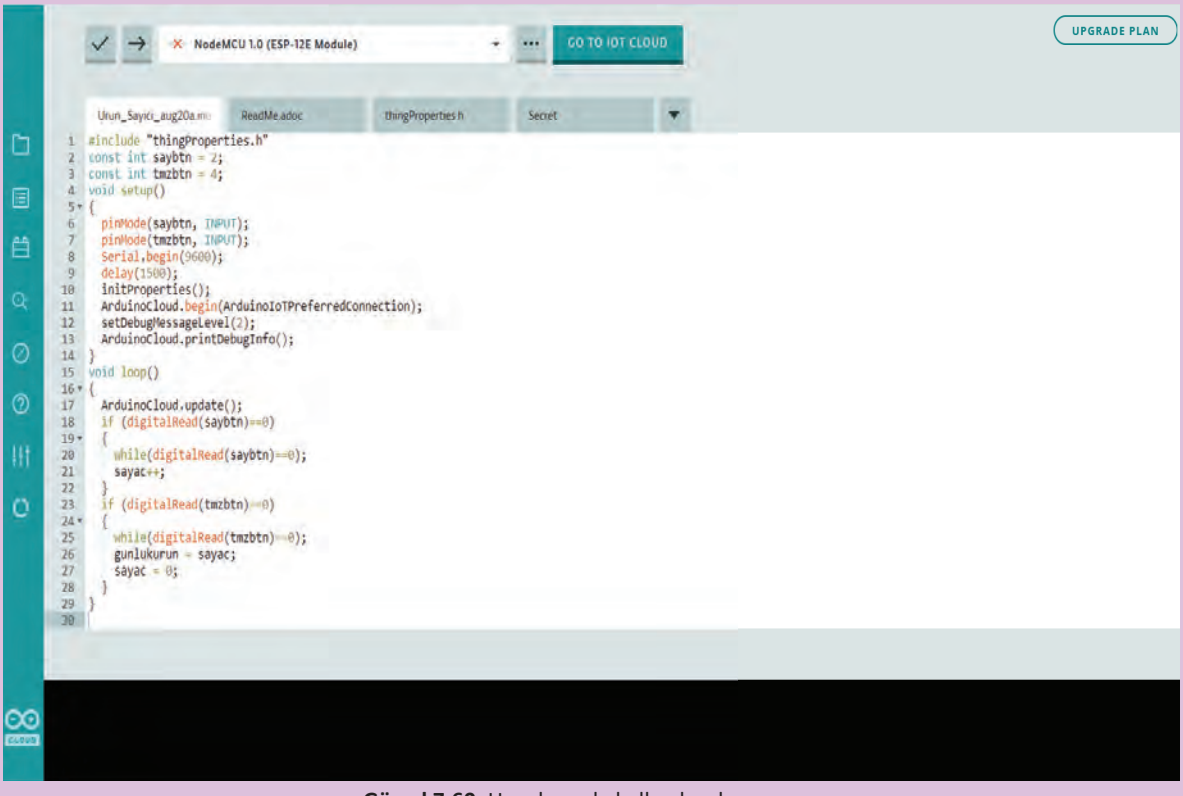
Görsel 7.58: Uygulamada kullanılacak değişken tamamlanması

2. Adım : Görsel 7.59'daki kullanıcı arayüzünü tasarlayınız ve gerekli değişken bağlantılarınızı yapınız.

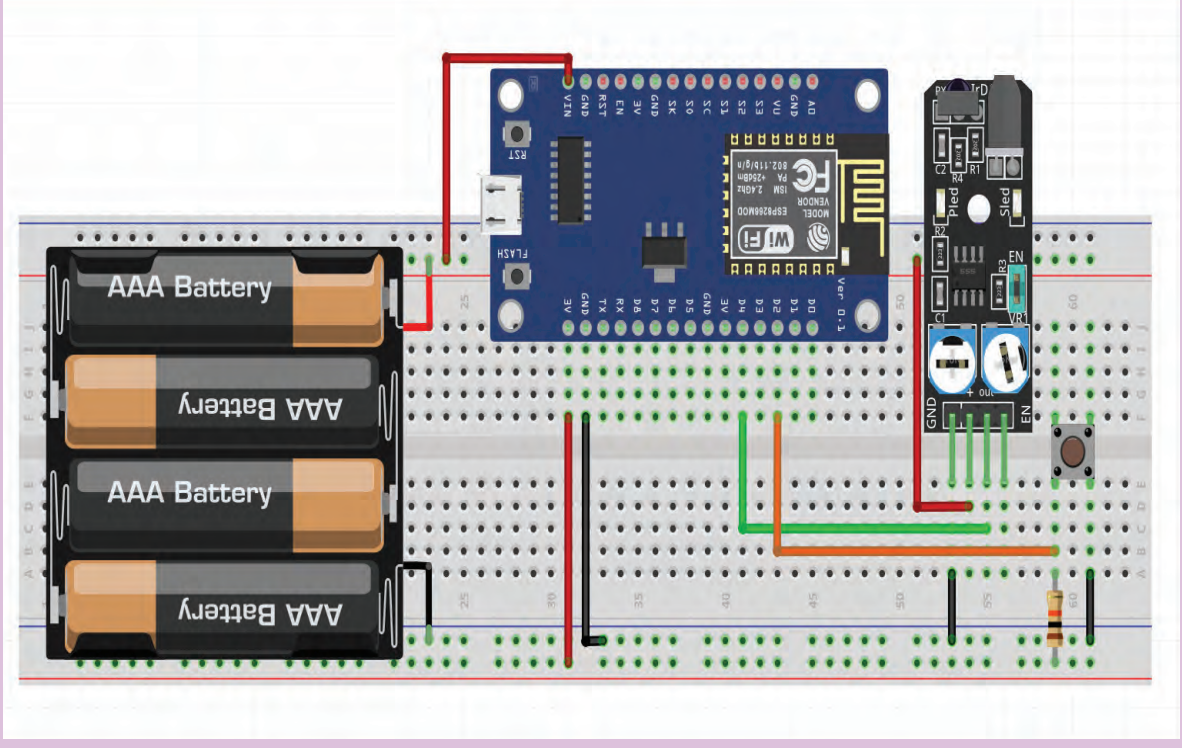


Görsel 7.59: Uygulamada kullanılacak dashboard tasarımı

3. Adım : Görsel 7.60'taki programı yazınız ve Görsel 7.61'deki devreyi kurarak sistemin çalışmasını test ediniz.



Görsel 7.60: Uygulamada kullanılacak program



Görsel 7.61: Uygulamada kullanılacak devre şeması

SIRA SİZDE

Evde yalnız kalmak zorunda olan yaşlıların acil durumlarda yakınlarına haber verebilmek için bir tasarım istenmektedir.

Bu tasarımda bir adet buton kullanılarak acil durum oluştuğunda yaşlı kişinin bu butona bir kere basarak bildirimi aktif hâle getirmesi ve aynı butona iki kere bastığında ise bildirimi iptal edecek şekilde çalışması istenmektedir. Bu tasarımı IoT nesnesi olarak tasarlayınız.

Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

Ölçütler	Evet	Hayır
1. Doğru sensör seçimi yapıldı.		
2. Elektronik devre kurulumu doğru şekilde tamamlandı.		
3. Butona bir kere basıldığında bildirim vermesi için kod yazımı başarılı şekilde tamamlandı.		
4. Butona iki kere basıldığında bildirim iptali için kod yazımı başarılı şekilde tamamlandı.		

SIRA SİZDE

Doktor Melike'nin evinde Hınzır adında doymak bilmeyen bir kedisi vardır. Melike nöbet günlerinde kedisini evde yalnız bırakmak durumundadır. Hınzır acıktığında besleyebilmek için bir sistem tasarlanmasını istemektedir.

Hınzır acıktığı zaman mama kabının etrafında miyavlamaktadır. Miyavlama sesini ses sensörü ile okuyup sesin belli bir değerin üstüne çıktığı durumları saydırarak bu sayaç durumuna göre hastaneden mama kabına mama koyacak bir IoT nesnesi tasarlayınız.

SIRA SİZDE

Kişiye özel dondurucu tasarlayan bir firmanın soğutucu kanallarının içine son derece yanıcı ve zehirli bir gaz dolumu yapılması gerekmektedir. Bu dolumun yapıldığı ortamı kalite kontrol birimi ile takip etmek istemektedir. Bu işlem için gaz sensörü ile alev sensörü kullanılarak ortamdaki bilgiler kaydedilecektir. Bu işlemleri yapacak IoT nesnesi tasarlayınız.

7.2. IoT PROTOTİPİ

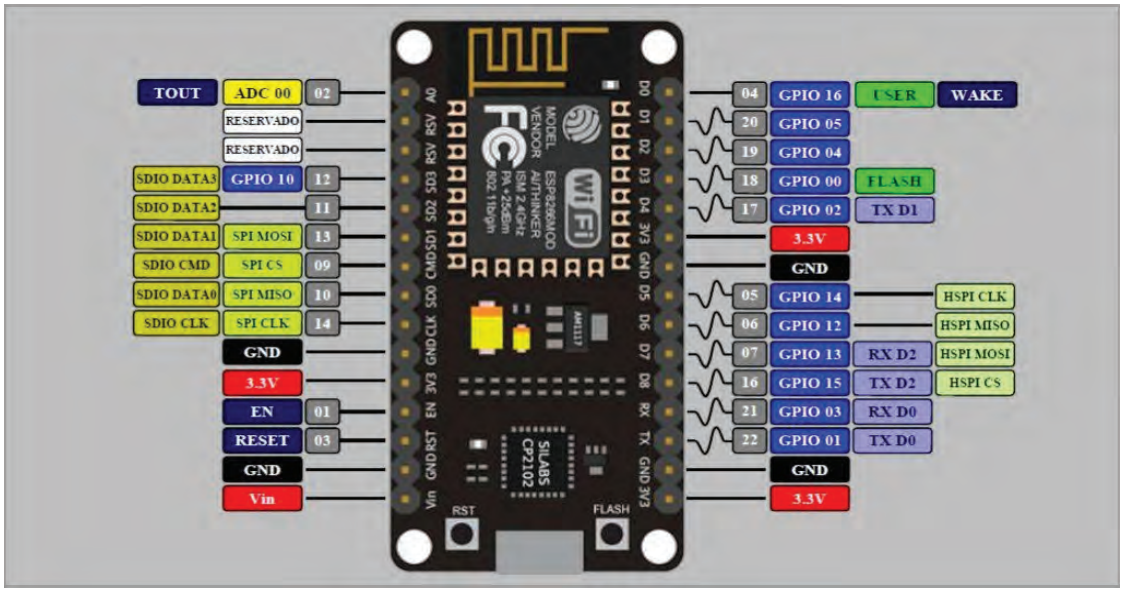
Prototip, bir ürünün üretim aşamasından önce fiziksel veya bilgisayar ortamında bitmiş hâline en yakın şekilde hazırlanan bir örneği ya da modelidir. Bir ürünün geliştirilmeden önce neye benzeyeceği ve hangi özelliklere sahip olacağını gösterdiğinden genellikle üretim öncesinde hazırlanır. IoT çevrelerinde de planlanan sistemin bir prototipinin yapılması, ürünün ortaya çıkarıldıktan sonra neleri yapacağını ve hangi sorunlara çözüm geliştireceğini belirlemek için tercih edilir.

Geliştirilecek prototipte silolardaki nem ve sıcaklık probleminden yola çıkılarak Nesnelerin İnterneti tabanlı prototip bir sistem geliştirilerek siloların gerçek zamanlı web ortamında izlenip kontrol altında tutulması

gerçekleştirilecektir. Silolar, kullanılan en güvenli depolama alanlarından olsa da kontrolü zordur. Özellikle dolu bir siloda alt bölgedeki ürünlerin kontrolü için teknolojiye başvurmak gerekir. Silolardaki problemlerin başında nem ve sıcaklık gelir. Kontrol altında olmayan bir nem, ürünün çürümesine kadar giden bir dizi soruna neden olur. Aynı şekilde sıcaklık problemi de özellikle haşere problemine sebep olabilir. Bu sıcaklık ve nem problemlerini gidermek amacıyla bir prototip oluşturulmuştur.

IoT tabanlı oluşturulan sistem için gömülü mimari ve Wi-Fi modüle sahip NodeMCU mikrodenetleyici kullanılmıştır. Silo içinde sıcaklık bilgileri DS18B20 sıcaklık sensörü tarafından alınmıştır. Silo içindeki nem değerleri ise DHT 11 sıcaklık ve nem sensörü ile alınmıştır. Kullanılan materyaller şunlardır.

a) **NodeMCU:** Üzerinde Wİ-Fİ özelliğine sahip olması ile modüler yapıya sahiptir. Programlanması sayesinde Nesnelerin İnterneti uygulamalarını düşük maliyet ile yapar. Analog giriş, PWM çıkış, dijital giriş/çıkış birimleri ve haberleşme desteği sağlar. NodeMCU mikrodenetleyicisine ait pin bağlantı yapısı Görsel 7.62'de gösterilmektedir.



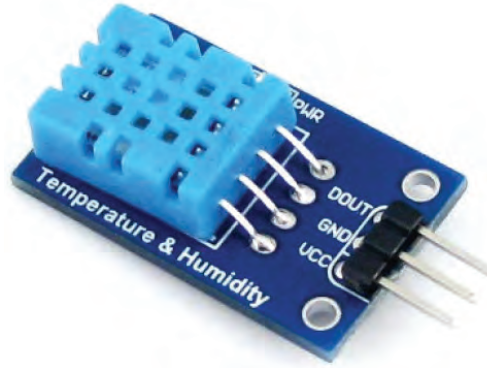
Görsel 7.62: NodeMCU bağlantı yapısı

NodeMCU, yaygın kullanılan, dâhilî Wi-Fi modüle sahip bir Nesnelerin İnterneti denetleyicisidir. 160 MHz'de çalışan 32 bitlik bir RISC Tensilica LX106 mikrodenetleyicisi içerir. NodeMCU'yu programlamak için C++ dili kullanılarak Arduino IDE derleyicisi kullanılmıştır. NodeMCU'ya ait teknik özellikler Tablo 7.1'de belirtilmiştir.

Tablo 7.1: NodeMCU Teknik Özellikleri

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

b) **DHT 11 Sıcaklık ve Nem Sensörü:** Bu çalışmada silonun ortam nem değerlerini ölçmek için bir adet DHT 11 sıcaklık ve nem sensörü kullanılmıştır. Kullanılan sensör Görsel 7.63'te gösterilmiştir ve özellikleri Tablo 7.2'de belirtilmiştir.



Görsel 7.63: DHT 11 Sensörü

Tablo 7.2: DHT 11 Teknik Özellikleri

Özellikler	Değer
Çıkış Tipi	Dijital Sinyal
Çalışma Gerilimi	3V ~ 5.5V (Tipik: 5V)
Çalışma Akım (mA)	0.5 ~ 2.5.
Sıcaklık Algılama Aralığı (°C)	0 ~ +50.
Nem Algılama Aralığı (%RH)	20 ~ 90.
Sensör Sıcaklık Hassasiyeti	±2 °C.

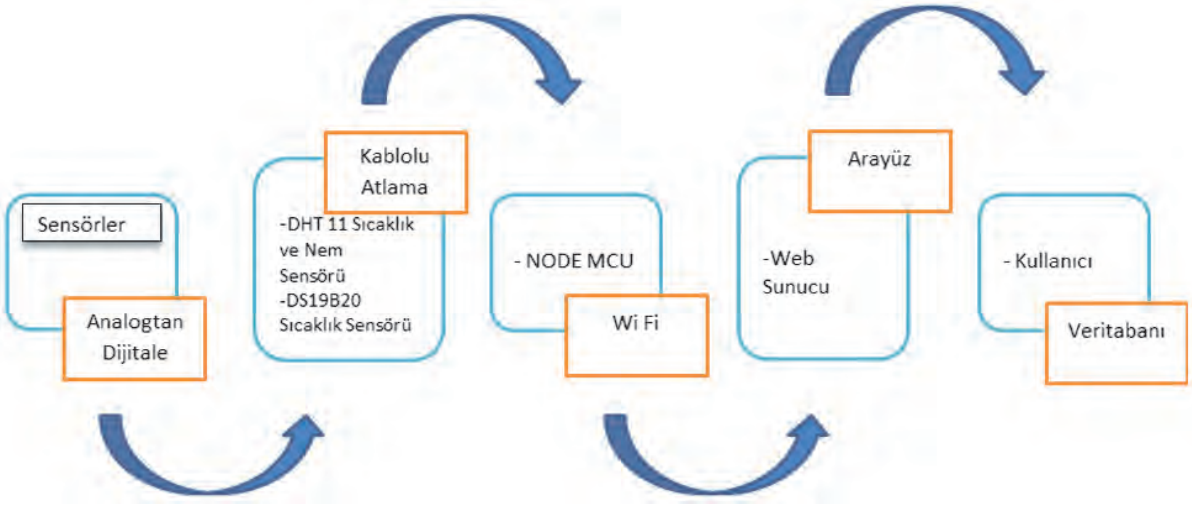
c) **DS18B20 Sıcaklık Sensörü:** Mikrodenetleyici ile 1 Wire arayüzünü kullanarak tek hat üzerinde haberleşir. Her sensör ROM hafızasında üretim esnasında belirlenen ve tek olan 64 bitlik seri koda sahiptir. Bu kod sayesinde aynı hat üzerinden birden fazla sensör haberleşir. Görsel 7.64'te DS18B20 sensörü gösterilmiş ve Tablo 7.3'te teknik özellikleri belirtilmiştir.

Görsel 7.64: DS18B20 Sensörü

Tablo 7.3: DS18B20 Teknik Özellikleri

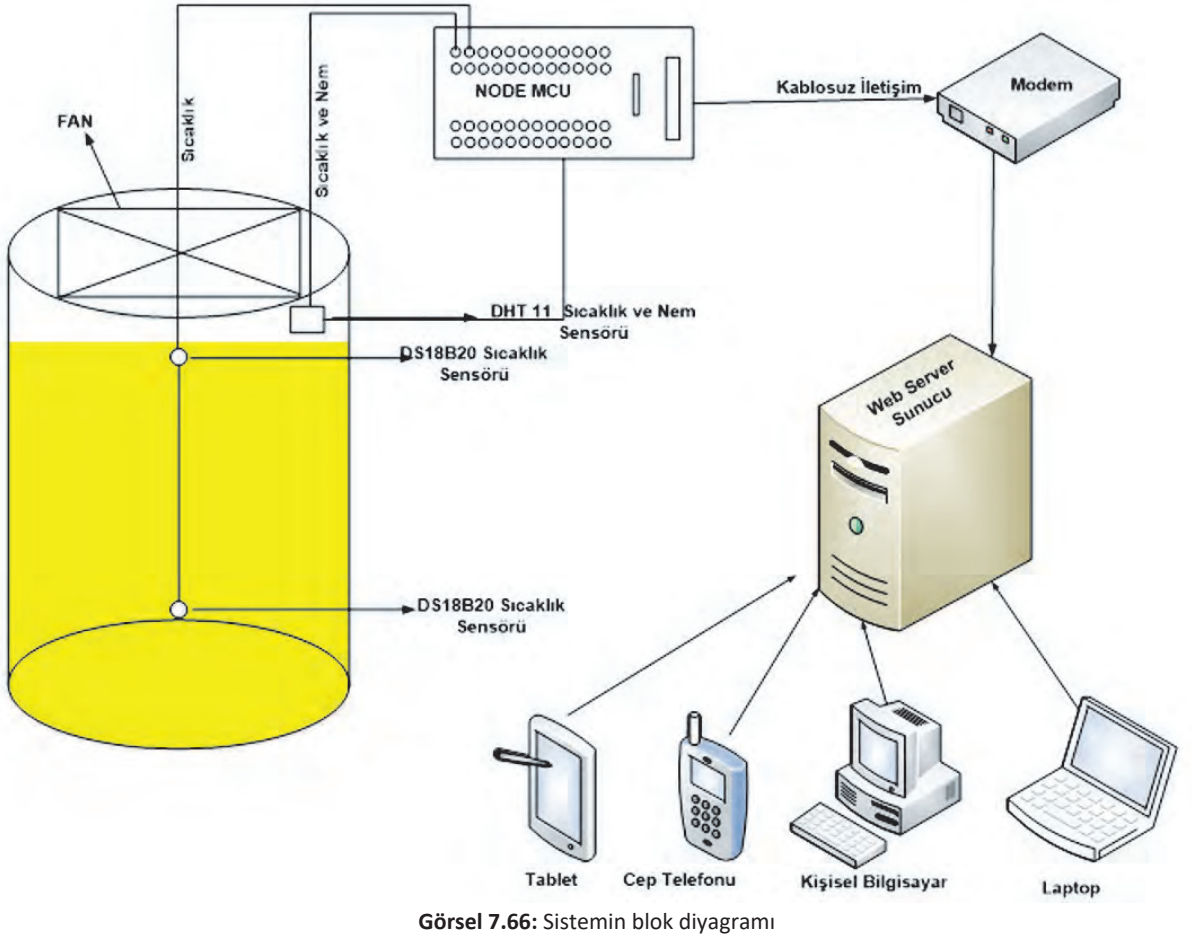
Özellikler	Değer
Çalışma Gerilimi	3V ~ 5.5V
Sıcaklık Algılama Aralığı (°C)	-55 ~ +125
Sensör Sıcaklık Hassasiyeti	±0.5 °C.

Oluşturulan prototipte nem ve sıcaklık kontrolünü sağlamak için izlenen metodoloji Görsel 7.65'te gösterilmiştir.



Görsel 7.65: Sistem için belirlenen yöntem

Belirlenen yöntem çerçevesinde oluşturulan IoT tabanlı prototip sisteme ait blok diyagramı Görsel 7.66'da gösterilmiştir.



Görsel 7.66: Sistemin blok diyagramı

1. Adım : Arduino.cc programını çalıştırınız. Yeni bir dosya oluşturarak IoT programının kodlamalarını aşağıda gösterildiği gibi yazınız.

```
#include <Wire.h>
#include "DHT.h"
#include "math.h"
#include <LM35.h>
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#define DHTTYPE DHT11
#define ONE_WIRE_BUS D2
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
const int DHTPin = 2;
DHT dht(DHTPin, DHTTYPE);
float tempSensor1, tempSensor2, tempSensor3;
uint8_t sensor1[8] = { 0x10, 0x42, 0x12, 0x60, 0x01, 0x08, 0x00, 0x52  };
uint8_t sensor2[8] = { 0x10, 0x37, 0x53, 0x80, 0x00, 0x08, 0x00, 0xE7  };
uint8_t sensor3[8] = { 0x28, 0xF6, 0x1D, 0x43, 0x98, 0x03, 0x00, 0x7E  };
ADC_MODE(ADC_VCC);
const char* ssid = "Sizin Wifi SSID Adınız";
const char* password = "Sizin Wifi Şifreniz";
const char* serverName = "http://192.168.1.4/test.php";
String apiKeyValue = "tPmAT5Ab3j7F9";
static char sicaklik[7];
static char nem[7];
```


Kod Satırı/Bloku	Açıklama
1	Uygulamayı çalıştırmak için gerekli olan kütüphanelerin yüklendiği bölümdür.
2	Sıcaklık ve nem sensörlerinin tanımlandığı kod blokudur.
3	NodeMCU mikroişlemci kartının internete erişebilmesi için gerekli olan ağ adı ve ağ şifresinin tanımlandığı bölümdür.
4	Sensörden gelen verilerin internet tarayıcısında veri tabanına kaydedildiği localhost adresin tanımlandığı kod satırıdır.
5	Bu kod (apiKeyValue) rastgele olarak kişi tarafından yazılabilir. Güvenlik için kullanılır. Bu kod, tarayıcıdan iletiliyorsa işlem yapılır, iletilmiyorsa işlem yapılamaz.

```
void setup(void)
{
  dht.begin();
  Serial.begin(115200);
 WiFi.begin(ssid, password);
  Serial.println("Connecting");
  while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("Connected to WiFi network with IP Address: ");
  Serial.println(WiFi.localIP());

}
```

Kod Satırı/Bloku	Açıklama
1	Dht11 sensörü okuması başlatan kod satırıdır.
2	Programın seri bağlantı hızı 115200 olarak ayarlanmıştır. Wi-Fi bağlantısı başlatılmıştır.
3	Wi-Fi'ye bağlantı işleminin kontrol edildiği bölümdür. Eğer bağlanma işlemi başarılı ise IP adresi ile seri port ekrana yazdırmaktadır.

```
void loop(void)
{
  sensors.requestTemperatures();
  tempSensor1 = sensors.getTempC(sensor1);
  tempSensor2 = sensors.getTempC(sensor2);
  tempSensor3 = sensors.getTempC(sensor3);
  float nemdeger = dht.readHumidity();
  float sicaklikdeger = dht.readTemperature();
  if(WiFi.status()== WL_CONNECTED){
    HTTPClient http;
    http.begin(serverName);
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    String httpRequestData = "kimlikno=" + apiKeyValue + "&sicaklik1=" + tempSensor1
                           + "&sicaklik2=" + tempSensor2 + "&ortamsicaklik=" + sicaklikdeger + "&nem=" + nemdeger + "";
    Serial.print("httpRequestData: ");
    Serial.println(httpRequestData);
    int httpResponseCode = http.POST(httpRequestData);
    if (httpResponseCode>0) {
      Serial.print("HTTP Response code: ");
      Serial.println(httpResponseCode);
    }
    else {
      Serial.print("Error code: ");
      Serial.println(httpResponseCode);
    }
    http.end();
  }
  else {
    Serial.println("WiFi Disconnected");
  }
  delay(500000);
```

Kod Satırı/Bloku	Açıklama
1	Sıcaklık ve nem değerleri sensörlerden alınır.
2	Localhosta sensör verilerini göndermek için http servisi açılır.
3	Sensörlerden okunan sıcaklık ve nem değerleri için Http istek verisi oluşturulur. Gönderim sırasında güvenlik kontrolü için apiKeyValue'de kullanılmıştır.
4	Localhost'a veriler gönderilir.
5	Verilerin http tarayıcısına aktarımında bir sorun olup olmadığı kontrol edilir. Sorun yoksa toplanan veriler ekranda yazdırılır, sorun varsa ResponseCode (error code) ekrana yazdırılır.
6	Wi-Fi bağlantısı kapatılır.
7	Bu döngü işlemi 5 dakika arayla gerçekleşir.

2. Adım : Sunucu tarafında çalışacak program için aşağıdaki kodları herhangi bir HTML editörünü açarak yazınız. Bunun için Notepad, Notepad++ gibi ücretsiz HTML editörlerini kullanabilirsiniz.

```php
<?php
$servername = "localhost";
$dbname = "silo";
$username = "root";
$password = "1";
$api_key_value = "tPmAT5Ab3j7F9";
$kimlik= $sicaklik1 = $sicaklik2 = $ortamsicaklik = $nem= "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $kimlik = test_input($_POST["kimlikno"]);
    if($kimlik == $api_key_value) {
        $sicaklik1 = test_input($_POST["sicaklik1"]);
        $sicaklik2 = test_input($_POST["sicaklik2"]);
        $ortamsicaklik = test_input($_POST["ortamsicaklik"]);
        $nem = test_input($_POST["nem"]);
        $conn = new mysqli($servername, $username, $password, $dbname);
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }
        $sql = "INSERT INTO tablo (id,sicaklik1,sicaklik2,ortamsicaklik,nem)
        VALUES ('" . $kimlik . "', '" . $sicaklik1 . "', '" . $sicaklik2 . "','" . $ortamsicaklik . "','" . $nem . "')";
        if ($conn->query($sql) === TRUE) {
            echo "Yeni Kayıt Başarılı Bir Şekilde Oluşturuldu";
        }
        else {
            echo "Error: " . $sql . "<br>" . $conn->error;
        }
        $conn->close();
    }
    else {
        echo "Wrong API Key provided.";
    }
}
else {
    echo "No data posted with HTTP POST.";
}
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
```

Kod Satırı/Bloku	Açıklama
1	PHP'de veri tabanı bağlantısı ayarları yapılır. Server adı, veri tabanı adı ve şifresi tanımlanır.
2	Veri tabanına kaydedilecek değişkenlerin tanımlandığı kod blokudur.
3	IoT cihazından gönderilen sensör verilerinin eşitlendiği kod blokudur.
4	Veri tabanına kayıt işlemi yapılır.
5	Bu fonksiyon IoT cihazından gönderilen verilerin POST'ta ayrıştırıldığı kod blokudur.

```
<!DOCTYPE html>
 <html>
 <head>
 <title>Silo Sıcaklık ve Nem Kontrolü</title>
<meta name='viewport' content='width=device-width, initial-scale=1.0'>
 <link href='https://fonts.googleapis.com/css?family=Open+Sans:300,400,600'
rel='stylesheet'>
<style>
html { font-family: 'Open Sans', sans-serif; display: block; margin: 0px auto;
text-align: center;color: #444444;}
  body{margin-top: 50px;}
  h1 {margin: 50px auto 30px;}
.side-by-side{display: table-cell;vertical-align: middle;position: relative;}
  .text{font-weight: 600;font-size: 19px;width: 200px;}
 .temperature{font-weight: 300;font-size: 50px;padding-right: 15px;}
 .living-room .temperature{color: #3B97D3;}
  .bedroom .temperature{color: #F29C1F;}
  .kitchen .temperature{color: #26B99A;}
  .superscript{font-size: 17px;font-weight: 600;position: absolute;right: -
5px;top: 15px;}
  .data{padding: 10px;}
  .container{display: table;margin: 0 auto;}
  .icon{width:82px}
 </style>
  </head>
  <body>

 <?php
$servername = "localhost";
$dbname = "silo";
$username = "root";
$password = "1";
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT id, sicaklik1,sicaklik2,ortamsicaklik,nem,tarih,saat FROM tablo
ORDER BY tarih DESC LIMIT 1";
if ($result = $conn->query($sql)) {
    while ($row = $result->fetch_assoc()) {
        $row_id = $row["id"];
        $row_sicaklik1 = $row["sicaklik1"];
    $row_sicaklik2 = $row["sicaklik2"];
    $row_ortamsicaklik = $row["ortamsicaklik"];
        $row_nem = $row["nem"];
        $row_tarih = $row["tarih"];
    $row_saat = $row["saat"];
```

1

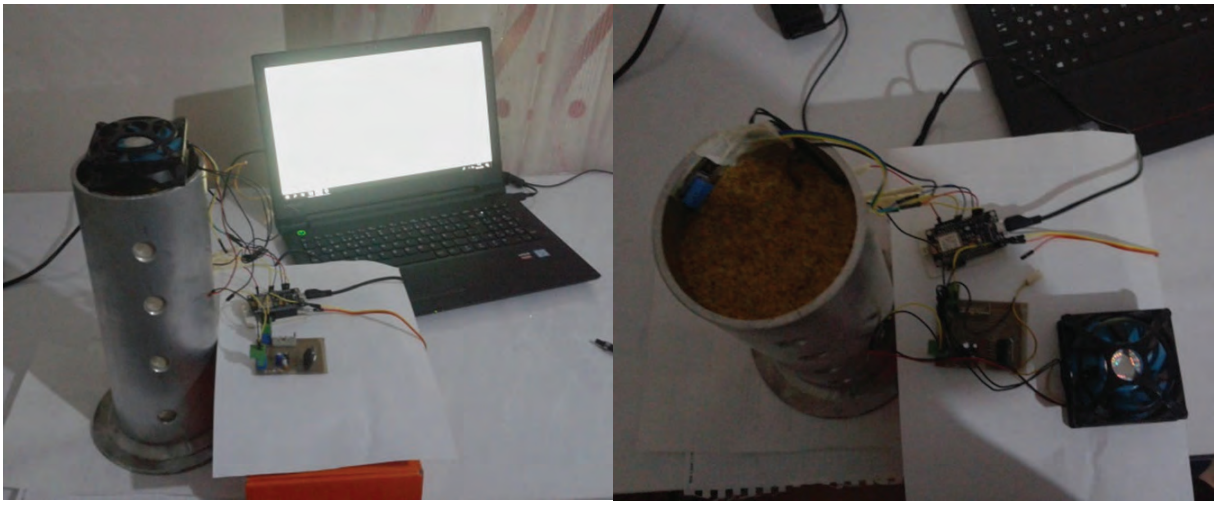
1

2

```
  echo '<h1>Silo Sıcaklık ve Nem Kontrolü</h1>
  <div class='container'>
  <div class='data living-room'>
  <div class='side-by-side icon'>
 <svg id="Layer_1" enable-background="new 0 0 496 496" height="50" viewBox="0
0 496 496" width="50"></svg>
  </div>
  <div class='side-by-side text'>1 Nolu Sıcaklık Sensörü</div>
  <div class='side-by-side temperature'>'
  . $row_sicaklik1 .
  '<span class='superscript'>&deg;C</span></div>
  </div>
  <div class='data bedroom'>
  <div class='side-by-side icon'>
  <svg id="Layer_1" enable-background="new 0 0 496 496" height="50" viewBox="0
0 496 496" width="50">
  <circle cx="284" cy="48" fill="#ff6e29" r="16"/></svg>
  </div>
  <div class='side-by-side text'>2 Nolu Sıcaklık Sensörü</div>
  <div class='side-by-side temperature'>'
  . $row_sicaklik2 .
  '<span class='superscript'>&deg;C</span></div>
  </div>
  <div class='data kitchen'>
  <div class='side-by-side icon'>
 <svg id="Layer_1" enable-background="new 0 0 496 496" height="50" viewBox="0
0 496 496" width="50"></svg>
  </div>
  <div class='side-by-side text'>Ortam Sıcaklığı</div>
  <div class='side-by-side temperature'>'
  .$row_ortamsicaklik.
  '<span class='superscript'>&deg;C</span></div>
  </div>

  <div class='data kitchen'>
  <div class='side-by-side icon'>
 <svg version="1.1" id="Capa_1"></svg>
</div>
  <div class='side-by-side text'>Nem</div>
  <div class='side-by-side temperature'>'
  .$row_nem.
  '<span class='superscript'>&deg;C</span></div>
  </div>
  </div>'
  ?>
  </body>
</html>
```

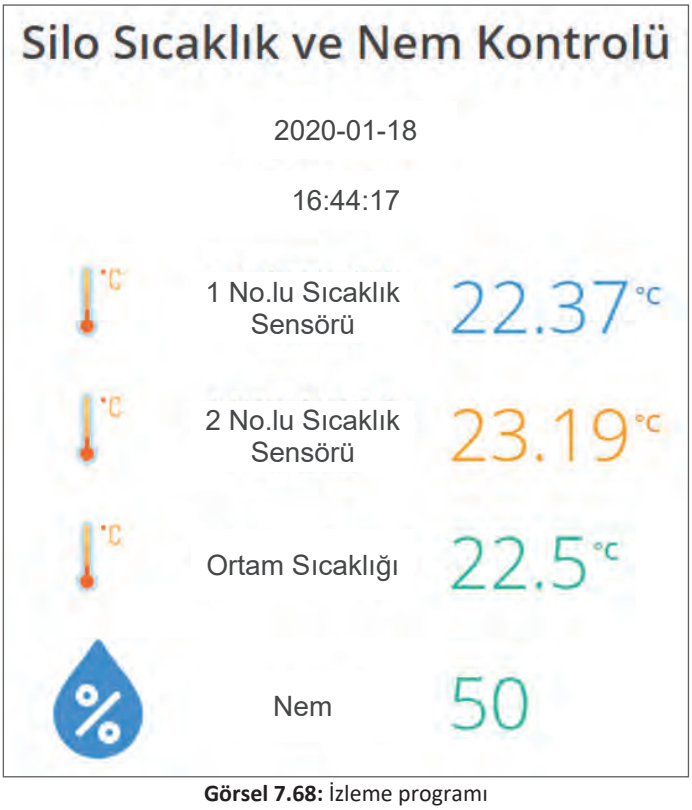
Kod Satırı/Bloku	Açıklama
1	Web sayfası stillerinin tanımlandığı kod blokudur.
2	Veri tabanı server, kullanıcı ve şifre tanımlaması yapılır. Ayrıca veri tabanı bağlantısı yapılarak veri tabanına kayıtlı en son verinin getirildiği kod blokudur.
3	Veri tabanından gelen sıcaklık ve nem değerlerinin hmtl sayfasında gösterildiği kod blokudur.

3. Adım : Siloların kontrolü için oluşturulan sistemin prototipi Görsel 7.67'de gösterilmektedir.



Görsel 7.67: Sistemin protipi

4. Adım : Programı çalıştırarak sensörlerden okunan değerleri web tarayıcınızda izleyiniz (Görsel 7.68).



Görsel 7.68: İzleme programı

SIRA SİZDE

Gaz sensörü ve alev sensörü kullanarak ortamdaki yanıcı ve zehirli gaz bilgilerini kaydetmesi amacıyla tasarladığı iz IoT nesnesinin bir prototipini geliştiriniz.

Çalışmalarınızı yaparken değerlendirme ölçütlerini dikkate alınız.

Ölçütler	Evet	Hayır
1. Kullanılacak Iot cihazları doğru seçti.		
2. Devre şeması oluşturdu.		
3. Devre şemasına göre elektronik devre kartı üzerine devreyi doğru bir şekilde kurdu.		
4. Devre elemanları arasındaki bağlantıyı doğru şekilde yaptı.		
5. Prototipe ait kodları doğru şekilde yazdı.		
6. Yazılan kodların Iot prototipine aktarımını başarılı bir şekilde yaptı.		
7. Prototip, amacına uygun şekilde sensörlerden toplanılan verileri istenilen yere depoladı.		

ÖLÇME VE DEĞERLENDİRME

A) Aşağıdaki cümlelerde parantez içine yargılar doğru ise "D", yanlış ise "Y" yazınız.

1. ( ) Bir ürünün geliştirilmeden önceki örneği, modeli ya da sürümüne prototip denir.
2. ( ) IoT prototipi oluşturmak için sensörlere ihtiyaç yoktur.
3. ( ) DHT11 sensörü ışık şiddetini ölçen bir sensördür.
4. ( ) IoT cihazları internete veri gönderebilir ancak internet üzerinden kontrol edilemez.
5. ( ) IoT cihazlar için internet olmadan da etkileşim kurulabilir.

B) Aşağıdaki soruları dikkatlice okuyarak doğru seçeneği işaretleyiniz.

6. Aşağıdaki değişken izinlerinden hangisi Arduino IoT Cloud projesinde veri yazma ve okuma amacıyla kullanılır?

A) Execute
B) Periodically
C) Read Only
D) Read & Write
E) Write Only

7. Arduino IoT Cloud arayüzünde gerçekleştirilen bir projeye Slider ve Switch nesneleri yerleştirilmek istenmektedir. Bu nesneleri projenin web sayfasına yerleştirmek için hangi sekme kullanılır?

A) Dashboards
B) Devices
C) Integrations
D) Templates
E) Things

8. Görseldeki Arduino IoT Cloud nesnenin adı aşağıdakilerden hangisidir?

A) Chart
B) Gauge
C) Map
D) Status
E) Value

CEVAP ANAHTARLARI

1. ÖĞRENME BİRİMİ					
Doğru/Yanlış - Boşluk Doldurma - Çoktan Seçmeli					
1	Yanlış	**6**	Sensörler	**11**	C
2	Doğru	**7**	Abone	**12**	E
3	Yanlış	**8**	B	**13**	E
4	Doğru	**9**	B	**14**	C
5	Nesnelerin İnterneti (IoT)	**10**	D	**15**	

2. ÖĞRENME BİRİMİ					
Boşluk Doldurma - Çoktan Seçmeli					
1	Akımı - gerilimi	**7**	E	**13**	D
2	ROM	**8**	D	**14**	I - II - III - IV - V
3	A	**9**	B	**15**	0,37 KΩ
4	D	**10**	C	**16**	250 Ω
5	A	**11**	E		
6	A	**12**	D		

3. ÖĞRENME BİRİMİ					
Doğru/Yanlış - Boşluk Doldurma - Çoktan Seçmeli					
1	Yanlış	**10**	Kullanıcı arayüzü	**19**	D
2	Doğru	**11**	Tek kartlı bilgisayar, boyutu, ucuz	**20**	B
3	Yanlış	**12**	GPIO.BOARD	**21**	E
4	Yanlış	**13**	D	**22**	D
5	Doğru	**14**	A	**23**	B
6	Girdi, işleme ve çıktı	**15**	C	**24**	A
7	Sınıflandırma, sıralama ve hesaplama	**16**	A	**25**	A
8	API	**17**	C	**26**	E
9	GET, POST, PUT, DELETE	**18**	B	**27**	

4. ÖĞRENME BİRİMİ					
Doğru/Yanlış - Boşluk Doldurma - Çoktan Seçmeli					
1	Yanlış	**8**	Broker	**15**	B
2	Doğru	**9**	XML	**16**	B
3	Doğru	**10**	Sis bilişim	**17**	D
4	Yanlış	**11**	D	**18**	B
5	Doğru	**12**	A	**19**	A
6	Doğru	**13**	C	**20**	A
7	LoRAWAN	**14**	A		

5. ÖĞRENME BİRİMİ	
Doğru/Yanlış - Çoktan Seçmeli	
1	Doğru
2	Yanlış
3	Yanlış
4	B
5	E
6	A

6. ÖĞRENME BİRİMİ	
Doğru/Yanlış - Çoktan Seçmeli	
1	Doğru
2	Doğru
3	Doğru
4	Doğru
5	E
6	A
7	E

7. ÖĞRENME BİRİMİ	
Doğru/Yanlış - Çoktan Seçmeli	
1	Doğru
2	Yanlış
3	Yanlış
4	Yanlış
5	Doğru
6	D
7	A
8	B

KAYNAKÇA

1. ÖĞRENME BİRİMİ

- https://www.cisco.com/
- https://toolbox.googleapps.com/apps/messageheader/

2. ÖĞRENME BİRİMİ

- L293D : Sensörün datasheet'inden faydalanıldı.
- Mq-2 : Microsoft Word - MQ-2 new.doc (pololu.com)
- Mq-3 : MQ-3.doc (sparkfun.com)
- Mq-4 : MQ-4.doc (sparkfun.com)
- Mq-5 : MQ-5.doc (elektronikhobi.net)
- Mq-6: MQ-6.doc (sparkfun.com)
- Mq-7: MQ-7.doc (sparkfun.com)
- Mq-9: Microsoft Word - MQ-9 New.doc (pololu.com)

3. ÖĞRENME BİRİMİ

- https://www.restapitutorial.com/httpstatuscodes.html)
- http://sistem41.com/blog/nesnelerin-interneti-iot-guvenligi/
- ioturkiye.com/2020/09/iot-saldirilari-ve-onerilen-onlemler/
- https://medium.com/coding-in-simple-english/iot-best-security-practices-c95231807cb9
- https://iot-analytics.com/understanding-iot-security-part-1-iot-security-architecture/
- https://www.raspberrypi.org
- https://www.raspberrypi.com/documentation/computers/os.html#gpio-and-the-40-pin-header
- Cisco Packet Tracer

4. ÖĞRENME BİRİMİ

- https://appinventor.mit.edu/
- https://io.adafruit.com/
- https://www.kaggle.com/
- Özdoğan, Erdal. (2020). Kampüs ağlarında nesnelerin interneti için genel mimari ve protokol tasarımı (Doktora tezi). Yükseköğretim Kurulu Ulusal Tez Merkezi (650764).

5. ÖĞRENME BİRİMİ

- Majid Meghdadi, Suat Özdemir, İnan Güler. Kablosuz Algılayıcı Ağlarında Güvenlik: Sorunlar ve Çözümler, Bilişim Teknolojileri Dergisi, Cilt: 1, Sayı: 1, Ocak 2008
- Oğuzhan TAŞ1, Farzad KIANI. Nesnelerin İnterneti (IoT) ve Kablosuz Algılayıcı Ağların Güvenliğine Yapılan Saldırıların Tespit Edilmesi ve Önlenmesi. Gazi Üniversitesi Politeknik Dergisi, ISSN: 2147-9429 (ÇEVRİMİÇİ)
- Furkan Yusuf YAVUZ, Devrim ÜNAL, Ensar GÜL. Nesnelerin İnternetinde Yönlendirme Saldırılarının Tespiti için Derin Öğrenme, International Journal of Computational Intelligence Systems, Cilt. 12 (2018) 39-58

6. ÖĞRENME BİRİMİ

- www.arduino.cc
- www.arduinoblocks.com
- www.thingspeak.com

7. ÖĞRENME BİRİMİ

- https://create.arduino.cc/iot
- Basçıftçı, F. , Ünlü, T. & Dasdemir, A. (2021). IoT Tabanlı Kontrol Sistemi ile Tahıl Silolarını Anında İzleme. Avrasya Bilim Teknoloji Mühendisliği ve Matematik Bildiriler Kitabı, 14, 15-23. DOI: 10.55549/epstem.1050154

GÖRSEL KAYNAKÇASI

http://kitap.eba.gov.tr/karekod/Kaynak.php?KOD=2423