

**T.C.**  
**SAKARYA ÜNİVERSİTESİ**  
**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**BSM 498 BİTİRME ÇALIŞMASI**

**Node.js Tabanlı IoT Uygulaması**

**G151210108 – Serhat Kerem EROL**

**Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ**  
**Tez Danışmanı : Prof. Dr Celal Çeken**

**2019-2020 Güz Dönemi**

**T.C.**  
**SAKARYA ÜNİVERSİTESİ**  
**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**

**Node.js Tabanlı IoT Uygulaması**

**BSM 498 - BİTİRME ÇALIŞMASI**

Serhat Kerem

**Fakülte Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ**

**Bu tez .. / .. / ... tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.**

.....  
**Jüri Başkanı**

.....  
**Üye**

.....  
**Üye**

## ÖNSÖZ

Bu bitirme tasarım çalışması Sakarya Üniversitesi, Haşim Gürdamar Bilgisayar ve Bilişim Fakültesi, Bilgisayar Mühendisliği Lisans Programı'nda yapılmıştır.

Son zamanlarda popülaritesi her geçen gün artan arduino hayatımıza girip gerek evimizde, iş yerimizde, arabamızda, hayatımızın her noktasında bulunmaktadır. Oluşan bu nesnel ortamda nesnelerin haberleşmesi ve bu haberleşme çeşitlerinin her geçen gün daha çok önemini veri gizliliği ile de ortaya çıkmıştır. Özellikle bu teknolojinin gelişimi insanlarımızın refahını arttırırken teknoloji ortamına da gelişme imkânı sağlamaktadır.

# İÇİNDEKİLER

ÖNSÖZ iii

SİMGELER VE KISALTMALAR LİSTESİ ..... vi

ŞEKİLLER LİSTESİ ..... vii

<b>BÖLÜM 1.</b>	<b>GİRİŞ.....</b>	<b>1</b>
1.1.	Proje tanımı .....	1
1.2.	Arduino Devre Şeması .....	1
1.3.	Arduino Verileri Serial Port ile almak.....	2
1.4.	MQTT-Broker İle Verilerin Alınıp Gönderilmesi.....	3
<b>1.4.1.</b>	<b>Mqtt Broker Kurulumu.....</b>	<b>3</b>
<b>1.4.2.</b>	<b>Mqtt Broker Ayarlamaları .....</b>	<b>4</b>
1.5.	Ön Panel (Dashboard).....	4
1.6.	Database Block Diagram .....	7
1.7.	Sistem Şeması .....	7
<b>BÖLÜM 2.</b>	<b>Kullanılan Donanım Teknolojileri .....</b>	<b>8</b>
2.1.	Arduino Uno .....	8
<b>2.1.1.</b>	<b>Arduino Uno Özellikleri Nelerdir? .....</b>	<b>9</b>
<b>2.1.2.</b>	<b>Arduino Uno Pinout ve Datasheet .....</b>	<b>11</b>
<b>2.1.3.</b>	<b>Arduino Uno PWM Pin.....</b>	<b>13</b>
<b>2.1.4.</b>	<b>Arduino Uno'yu Bilgisayara Tanıtma .....</b>	<b>13</b>
2.2.	Dht11 Sıcaklık Nem Sensörü .....	14
<b>BÖLÜM 3.</b>	<b>KULLANILAN YAZILIM TEKNOLOJİLERİ.....</b>	<b>16</b>
3.1.	Node.js.....	16
<b>3.1.1.</b>	<b>Node.js Nedir? .....</b>	<b>16</b>
<b>3.1.2.</b>	<b>Node.js İle Neler Yapılabilir? .....</b>	<b>16</b>
3.2.	SerialPort .....	18
3.3.	Socket.io .....	19
	Server (Sunucu) Nedir ? .....	19
	Client (İstemci) Nedir ? .....	19
	Socket-IO kurulumu .....	20
3.4.	MQTT .....	20
<b>3.4.1.</b>	<b>MQTT Nedir? .....</b>	<b>20</b>
<b>3.4.2.</b>	<b>MQTT ÖZELLİKLERİ.....</b>	<b>22</b>

<b>3.4.3.</b>	IoT Nedir? .....	22
<b>3.4.4.</b>	MQTT ile IoT Arasındaki Bağlantı Nedir? .....	22
<b>3.4.5.</b>	Gerçek Hayatta MQTT Nerelerde Kullanıyor ? .....	23
<b>3.5.</b>	MongoDb .....	23
<b>3.5.1.</b>	MongoDb Özellikleri .....	25
<b>BÖLÜM 4.</b>	<b>SONUÇLAR VE ÖNERİLER .....</b>	<b>27</b>

## SİMGELER VE KISALTMALAR LİSTESİ

A	: Numune kesit alanı
Av	: Sıkışma katsayısı
ASTM	: Amerikan standart
Cc	: Sıkışma indisi
Cj	: Değiştirilmiş sıkışma indisi
Cr	: Yeniden yükleme indisi
Cp	: Değiştirilmiş yeniden yükleme indisi
Cv	: Konsolidasyon katsayısı
Cl	: İkincil konsolidasyon (sıkışma) katsayısı
E	: Boşluk oranı
e0	: Başlangıç boşluk oranı
Ep	: Birincil konsolidasyon sonundaki boşluk oranı
H0	: Sıkışabilir tabakanın kalınlığı
Hd	: Numune kesit yüksekliği
Ip	: Plastisite indisi
K	: Permeabilite (geçirgenlik) katsayısı
Mv	: Hacimsel sıkışma katsayısı
R0	: Başlangıç okuma değeri
R50	: %50 oturmaya karşı gelen okuma değeri
R90	: %90 oturmaya karşı gelen okuma değeri
R100	: %100 oturmaya karşı gelen okuma değeri

## ŞEKİLLER LİSTESİ

Şekil 2-Serial Port kodları .....	2
Şekil 3-Mqtt-Broker Sisteminin Genel Hiyerarşisi .....	3
Şekil 4-Mqtt-Broker Sisteminin Ayarlanması .....	4
Şekil 5-Mqtt Broker Bağlantı Örneği.....	5
Şekil 6- Ön panel giriş ekranı.....	5
Şekil 7- Sunucunun Subscribe Olması .....	6
Şekil 8- Kullanıcı Ön Paneli .....	6
Şekil 9- Veritabanı Block Diyaramı.....	7
Şekil 10 - Sistem Şeması.....	7
Şekil 11- Arduino Uno .....	8
Şekil 12 Arduino - Uno Devre Şeması.....	13
Şekil 13- Dht11 .....	14
Şekil 14- Örnek Kod Bloğu.....	17
Şekil 15- SerialPort Tanımlama Kodu .....	18
Şekil 16 - Projede Serial Port Kullanımı.....	18
Şekil 17 - Socket.io Tanımlaması .....	19
Şekil 18 - Mqtt Broker Örneklendirmesi .....	21
Şekil 19 - MongoDB Örneklendirmesi .....	24
Şekil 20 - MongoDB Çalışma Şekli .....	24
Şekil 21 - Sql ve NoSql Benzerlik Şekli .....	25
Şekil 22 - MongoDB Json Tipi Veri Saklama Tipi .....	26

## ÖZET

Anahtar kelimeler: Endüstri 4.0, Nesnelerin İnterneti, Veri Haberleştirme, Arduino

Endüstri 4.0 terminolojisinin hakim olduğu global çehrede veri oluşturabilmenin, veri haberleşmesinin, verinin güvenli iletilmesinin, ve dahi verinin anlaşılabilirliğinin önemi daha da anlaşılmış ve güncel piyasa durumlarında gerçekleşen projelere entegre edilmiş durumdadır.

Nesnelerin İnterneti kavramı akıllı cihazların birbiriyle iletişime geçmesi, haberleşmesidir. Günümüzde nesnelerin interneti küçük ev aletlerinden akıllı şehirlere kadar uzanmaktadır.

Arduino Teknolojisi enteraktif projeler geliştirmek için tasarlanan, elektronik donanım ve yazılım temelli bir geliştirme platformudur.

Arduino nesnesinden verileri mqtt ile birlikte hazırlamış olduğumuz ön panel sistemimize aktarılıp grafiklerle kullanıcıya güzel bir tasarım sunmaktayım.



Çalışmamız da arduino şeması yukarıda ki gibidir. Bu çalışmamızda arduino'nun 5V kısmından sensöre enerji aktarılıp çalışmasını sağlamaktadır. Sensörümüzün şase pini arduinoda ki GND(Ground) pinine bağlanmaktadır. Sensörümüzde oluşacak veriyi arduinoya dijital pinlerinden 8 nolu pine bağlanmaktadır.

```

#include <dht11.h>
#define DHTPin 8

dht11 DHT11;

void setup()
{
    pinMode(DHTPin, INPUT);
    Serial.begin(115200);
}

void loop()
{
    int chk = DHT11.read(DHTPin);
    Serial.println((float)DHT11.temperature, 2);
    delay(2000);
}

```

Arduino çalışma kodu

### 1.3. Arduino Verileri Serial Port ile almak

```

//Seri port işlemleri için gerekli nesneler oluşturuluyor.
const Readline = require('@serialport/parser-readline')
const seriPort = new SerialPort("COM5", { baudRate: 115200 })
const parser = new Readline()

seriPort.pipe(parser)

parser.on('data', function (gelenVeri) {
    //console.log(gelenVeri);
    var array = gelenVeri.split(":");
    console.log(array[0]);
    io.emit('SunucudanIstemcilere',{
        value:gelenVeri
    })
    client.publish(mqttParameters.topic1, array[0]);
})

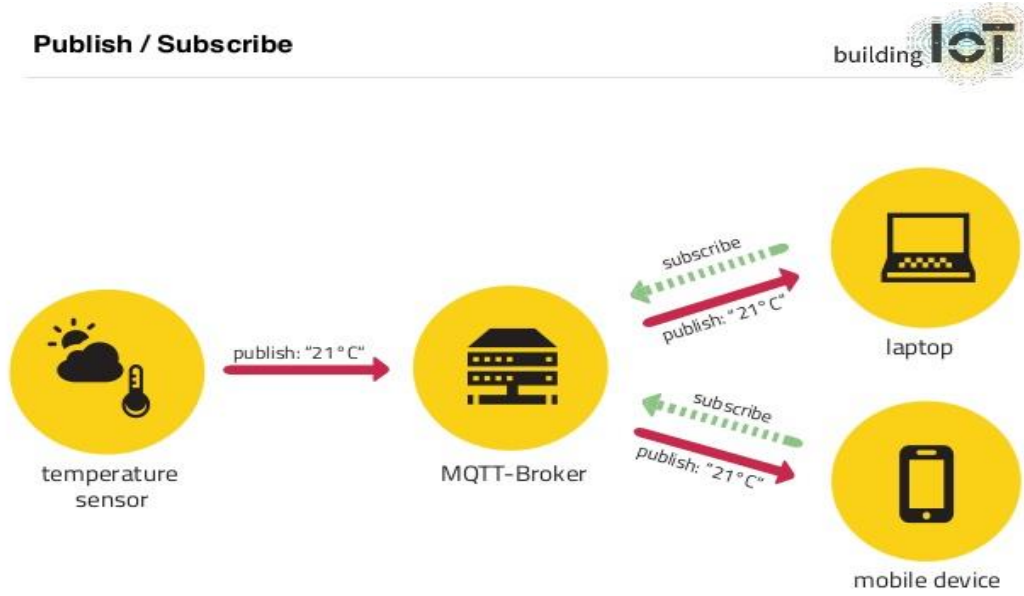
```

Şekil 2-Serial Port kodları

Arduinoda sensör vasıtası ile oluşturulan veriler 115200 baudRate hızı ile seriporta gönderiliyor. Yukarıda ki fotoğrafta belirttiğim üzere COM5 Seri port adresi dinlenip arduinodan aktarılan veriler mqtt broker adresine gönderiliyor

Mqtt broker'a verilerin gönderilmesi için belli bir topic'e ihtiyaçları vardır. Ben bu makaleyi Türkçe hazırlıyorsam okuyacak kişinin Türkçe bilmesi gibi. Arduinodan alınan veri belli bir Topic ile Mqtt broker'a iletilir. Mqtt-Broker'a bağlı olan üyelere içerisinde de veri gönderen tarafından belirlenen topic sahiplerine veriler aktarılır.

#### 1.4. MQTT-Broker İle Verilerin Alınıp Gönderilmesi



Şekil 3-Mqtt-Broker Sisteminin Genel Hiyerarşisi

Mqtt-Broker yukarıda ki fotoğrafta ifade edildiği gibi veriler sisteme gönderilir ve broker'a üye olan nesnelere iletilir. Kısaca tabir edilmesi bu şekildedir.

##### 1.4.1. Mqtt Broker Kurulumu

Nodejs üzerinde çalışacak olan Mqtt Broker'ı terminal ekranından `npm i mosca --save` diyerek kurulumunu başlatmamız gerekir. Ardından yine terminal ekranından `npm i mqtt --save` diyerek kurulumu tamamlamış bulunuyoruz.

### 1.4.2. Mqtt Broker Ayarlamaları

```
var mosca = require('mosca');

var settings = {
  port: 1884,
  http: {
    port: 3000
  },
};

var server = new mosca.Server(settings);

server.on('clientConnected', function(client) {
  console.log('client connected', client.id);
});
server.on('clientDisconnected', function(client) {
  console.log('client disconnected', client.id);
});

// fired when a message is received
server.on('published', function(packet, client) {
  console.log('Published', packet.payload.toString());
});

server.on('ready', setup);

// fired when the mqtt server is ready
function setup() {
  console.log('Mosca server is up and running');
}
```

Şekil 4-Mqtt-Broker Sisteminin Ayarlanması

Mqtt broker kurulumu gerçekleştikten Mqtt-Broker'ı ayarlamaları yapıldıktan sonra ayağı kaldırıldı. Broker'ın publish edilen port'u yani aktif olarak çalıştığı port 1884 olarak ayarlanıp Broker'a gelen verileri 3000 portu üzerinden subscribe ederek Topic sahibi olan client(üye)'lere aktarılır.

### 1.5. Ön Panel (Dashboard)

Broker'ımıza client(üye) olarak bağlandığımız ön panel sistemimizde, sistemimize giriş yaptığında bağlantı aktif edilip gerekli veriler broker üzerinden ön panel sistemimize aktarılıp, verileri kullanıcımıza grafiksel olarak yansıtmaktayız.

```

var mqtt =require("mqtt");
var client=mqtt.connect("mqtt://localhost:1884");

client.subscribe('new-user');
client.on('connect',(res,err)=>{
  console.log('connected!!');
  client.publish('new-user', 'Dashboard Connected')
  console.log(res);
});

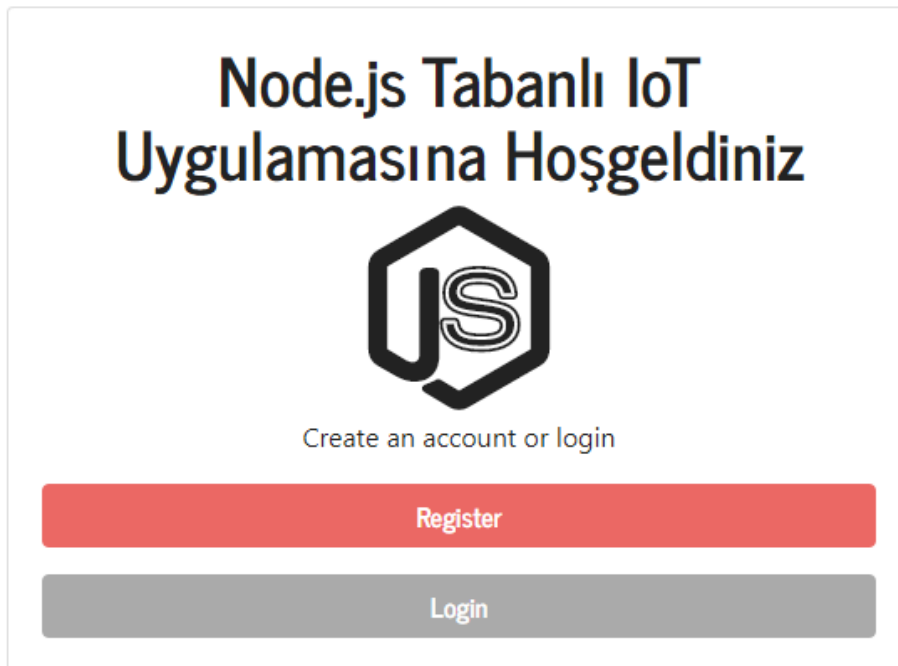
client.on('message',(topic,message)=>{
  console.log(topic,':',message.toString());
})

```

Şekil 5-Mqtt Broker Bağlantı Örneği

Ön panelimizin Mqtt-Broker serverımıza client(üye) bağlantı şekli yukarıda ki resimde ki gibidir.

Veriler sistemimize geldiği anda node.js kütüphanesi olan socket.io ile kullanıcı ekranına yansıtmaya başladık. Tabi bundan önce kullanıcımız sistemimize giriş yapmadan veriler sistemimize düşmüyordu.



Şekil 6- Ön panel giriş ekranı

Kullanıcı Sisteme girdiği anda Server'a socket.io ile birlikte sistemin aktif olduğunu veri alabileceğini bildiriyor. Ön panel sunucumuz Broker'dan istek yaparak subscribe olduğu Brokardan verileri alıyor.

```
socket.on("selamVer", data => {
  client.on("message", (topic, message) => {
    console.log(topic, ":", message.toString());
    io.emit("updateMessages", message.toString());
  });
});
```

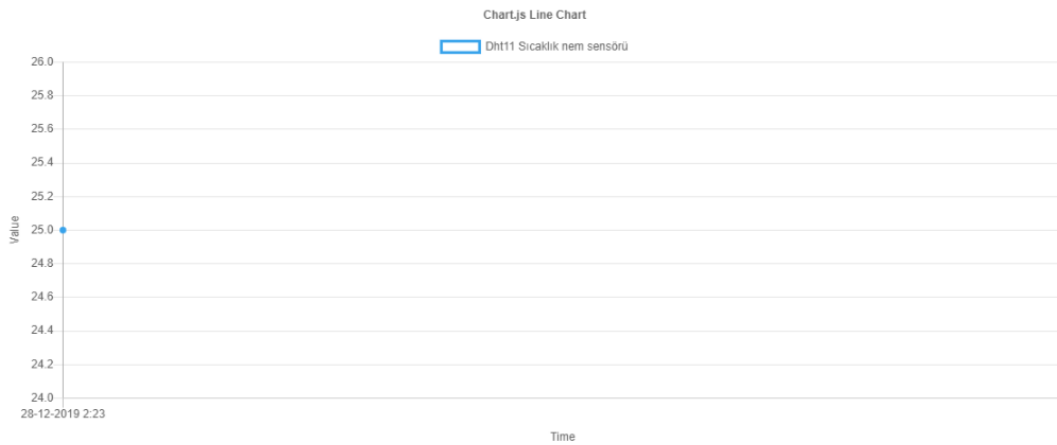
Şekil 7- Sunucunun Subscribe Olması

Gelen veriler ön panel sistemine sunucudan socket.io ile gönderilerek kullanıcı ekranına aktarılıyor. Aktarılan veriler chart.js ile birlikte grafiksel bir biçimde kullanıcının ekranına anlık olarak düşüp veritabanımıza kaydediliyor.

## Dashboard

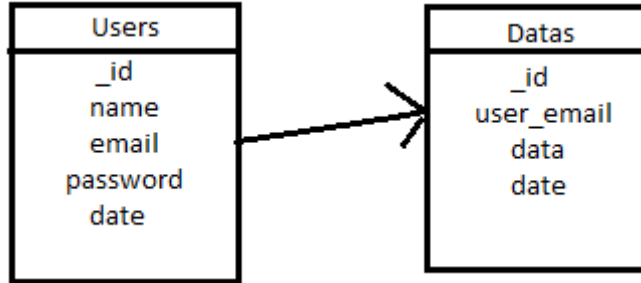
Welcome SERHAT KEREM EROL

Logout



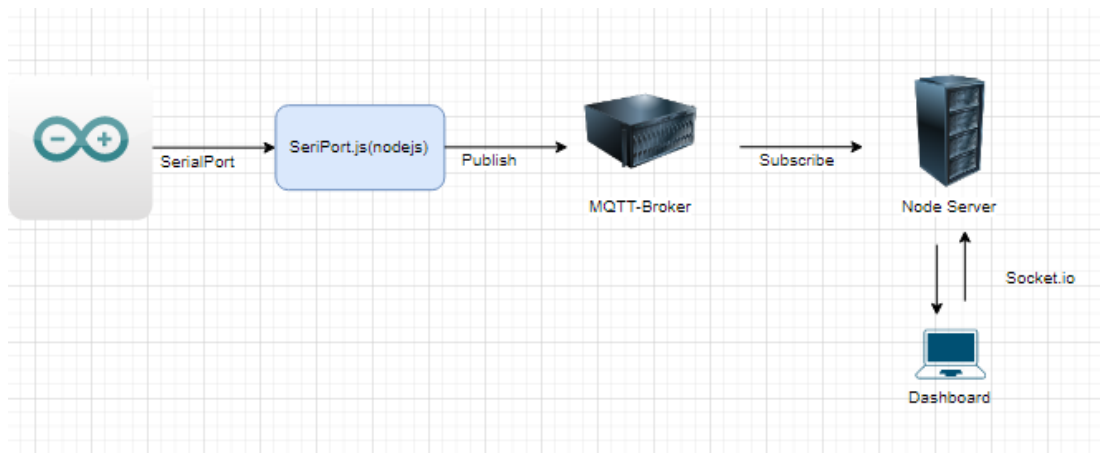
Şekil 8- Kullanıcı Ön Paneli

## 1.6. Database Block Diagram



Şekil 9- Veritabanı Block Diyaramı

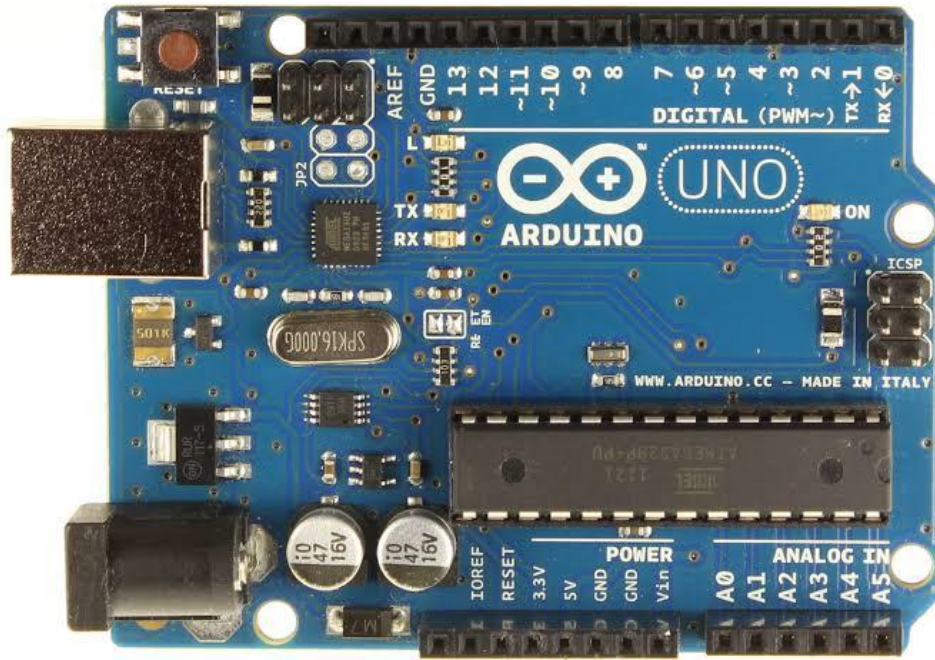
## 1.7. Sistem Şeması



Şekil 10 - Sistem Şeması

## BÖLÜM 2. Kullanılan Donanım Teknolojileri

### 2.1. Arduino Uno



Şekil 11- Arduino Uno

Arduino, elektronik donanım ve yazılım temelli bir geliştirme platformudur. Arduino ile öğrenciler de profesyoneller de çok detaylı programlama ve elektronik bilgiye sahip olmadan, temel bilgiler ile hobi amaçlı, eğitim amaçlı veya profesyonel anlamda projeler yapabilirler. Arduino Uno, en yaygın kullanılan ve en çok bilinen modelidir. 2010 yılında kullanıma sunulmuştur.

Arduino Uno ile çeşitli sensörlerden fiziksel bilgi alabilir, bu bilgiler ile çeşitli deneyler yapabilirsiniz. Ayrıca motor, LED, buzzer gibi uyarıcılardan bir çıktı elde edebilirsiniz. Bu gibi elektronik komponentleri Arduino Uno kartına bağlayarak kontrol etmek için temel bir programlama bilgisi yeterlidir. Projelerin seviyesine göre gerekli olan elektronik ve programlama bilgisi seviyesi de artacaktır. Boyut olarak çok daha küçük ve çok daha büyük modeller olsa da Arduino Uno'nun boyutu



projelere göre en standart olanıdır. 14 adet dijital çıkış pini bulunması 14 farklı dijital sensörün ve uyarıcının kontrol edilebileceği anlamına gelmektedir. Bu da birçok proje için yeterli bir sayıdır. Bu dijital çıkışlardan 5 tanesi PWM çıkışıdır. Motorların hızı, LED’lerdeki parlaklık seviyeleri gibi analog olarak kontrol edilmesi istenen uyarıcılar bu PWM pinlerine bağlanarak kontrol edilir. Arduino Uno’daki 6 tane analog giriş ise analog giriş sinyali alabildiğimiz sensörler içindir.

Arduino Uno ile LED yakıp söndürmek gibi en temel uygulamalardan drone, robot, akıllı ev otomasyonu, hırsız alarm sistemi, park sensörü gibi daha gelişmiş projeler de yapabilirsiniz. Bu tamamen ne yapmak istediğinizle alakalıdır. Kısacası Arduino Uno, standart boyutlarda bir kontrol kartı olup, basitten zora birçok uygulamada elektronik devreleri kontrol etmenizi sağlamaktadır.

### 2.1.1. Arduino Uno Özellikleri Nelerdir?

#### **Haberleşme:**

Arduino Uno birçok şekilde haberleşme işlemini gerçekleştirebilir. **RX** ve **TX** pinleri ile seri haberleşme imkanı mümkündür. **Atmega16u2** USB-seri dönüştürücü de bilgisayarda sanal bir seri port açarak Atmega328 ile bilgisayar arasında haberleşmeyi sağlar. Arduino IDE içerisinde yer alan seri monitör ile Arduino ile bilgisayar arasında metin temelli bilgilerin gönderilip alınmasını sağlar. Arduino ile bilgisayar arasında USB üzerinden bir haberleşme olduğunda Arduino üzerindeki RX ve TX yazan LED’ler yanar.

Arduino Uno’da normalde bir tane seri port bulunmaktadır fakat SoftwareSerial kütüphanesi kullanılarak bu sayı yazılımsal olarak arttırılabilir.

Atmega328 ayrıca I2C ve SPI portları da sağlamaktadır. Arduino IDE içerisinde yer alan Wire kütüphanesi I2C kullanımını, SPI kütüphanesi de SPI haberleşmesini sağlamak için kullanılır.

**Programlama:**

Arduino Uno da diğer tüm Arduino'lar gibi Arduino IDE ile programlanır. Detaylı Arduino Kurulumu yazımızı inceleyerek programı kurabilirsiniz.

Bootloader yazılımı bypass edilerek kart doğrudan mikrodenetleyicinin ICSP header kullanılarak ISP programlayıcı ile programlanabilir (\*).

**USB Kısa Devre ve Aşırı Akım Koruması:**

Arduino Uno üzerinden bulunan resetlenebilir sigorta, bilgisayarınızın USB portunu kısa devrelerden veya aşırı akım tüketimi durumlarından korumaktadır. Kart, USB portu üzerinden 500mA'den fazla akım çektiğinde otomatik olarak USB'den aldığı gücü koruma amacıyla kesmektedir. Fazla akım durumu veya kısa devre ortadan kaldırıldığında sigorta normal konuma döner ve tekrar bağlantı kurulur.

Arduino Uno'nun genel anlamda özellikleri aşağıdaki gibidir:

Mikrodenetleyici: ATmega328

Çalışma Gerilimi: 5V

Giriş Gerilimi (önerilen): 7-12V

Giriş Gerilimi (limit): 6-20V

Dijital G/Ç Pinleri: 14 (6 tanesi PWM çıkışı)

Analog Giriş Pinleri: 6

Her G/Ç için Akım: 40 mA

3.3V Çıkış için Akım: 50 mA

Flash Hafıza: 32 KB (ATmega328)

SRAM: 2 KB (ATmega328)

EEPROM: 1 KB (ATmega328)

Saat Hızı: 16 MHz

Uzunluk: 68.6 mm

Genişlik: 53.4 mm

Ağırlık: 25 g

## 2.1.2.Arduino Uno Pinout ve Datasheet

### Güç:

Arduino Uno, gücünü USB üzerinden veya adaptör girişinden alabilir. Yani bilgisayarınızın USB girişinden veya bilgisayarınızdan bağımsız olarak bir adaptör veya bataryadan güç elde edebilirsiniz. Doğrudan **Vin (+)** ve **GND (-)** pinlerinden de besleyebilirsiniz.

Harici güç kaynağı olarak 6-20V arası limit değerleri bulunmaktadır. Önerilen harici besleme gerilimi ise 7-12 V arasındır. Bunun sebebi 7V altındaki gerilimin stabil çalışmayıp, 12V üzeri gerilimin de aşırı ısınma sebebi olabilmesidir. Kart üzerinde bulunan regülatör sayesinde 7-12V arası gerilim 5V'a düşürülür ve kart bu şekilde çalışır.

**Vin:** Harici güç kaynağı için kullanılan pin.

**5V:** Regülatörden çıkan 5V çıkış gerilimini sağlar.

**3V3:** Kart üzerinde bulunan 3.3V regülatörü çıkış pinidir. Maks. 50mA çıkış verebilir.

**GND:** Toprak (-) pinleridir.

**Giriş/Çıkış (I/O):** 14 adet dijital, 6 adet analog giriş/çıkış pini bulunmaktadır. Bu pinlerin tamamının lojik seviyesi 5V'dur. Her pin maksimum 40mA giriş ve çıkış akımı ile çalışır. Ek olarak, bazı pinlerin farklı özellikleri bulunmaktadır. Özel pinler aşağıda belirtildiği gibidir:

**Seri Haberleşme- 0 (RX) ve 1 (TX):** TTL Seri veri alıp (RX), vermek (TX) için kullanılır. Bilgisayardan karta program yüklenirken veya bilgisayar-UNO arasında karşılıklı haberleşme yapılırken de bu pinlerden faydalanılır. Bu sebeple, karta program yüklendiği esnada veya kartla bilgisayar arası iletişim kurulduğunda bu pinleri kullanmamak gerekir.

**Harici Kesme (Interrupt)- 2 ve 3:** Bu pinler yükselen kenar, düşen kenar veya değişiklik kesmesi pinleri olarak kullanılabilir. Ayrıntılı bilgi için attachInterrupt() fonksiyon sayfasını inceleyebilirsiniz.

**PWM- 3,5,6,9,10 ve 11:** 8-bit çözünürlükte PWM çıkış pinleridir.

**SPI- 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK):** SPI haberleşmesi için bu pinler kullanılır.

**LED- 13:** Kart üzerinde dahili bir LED bulunmaktadır (L harfi ile gösterilmiş). Bu LED 13.pine bağlıdır. HIGH yapıldığında LED yanacak, LOW yapıldığında ise sönecektir.

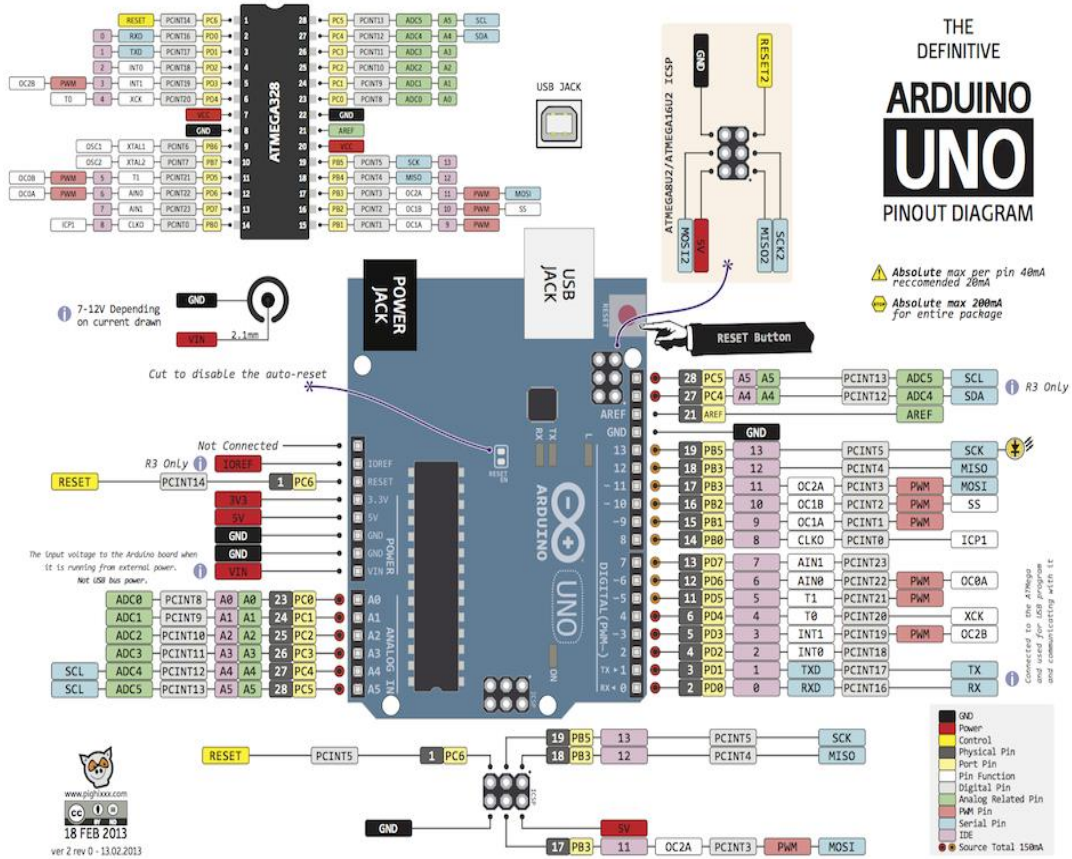
**Analog- A0,A1,A2,A3,A4,A5:** 6 adet 10-bit çözünürlüğünde analog giriş pini bulunmaktadır. Bu pinler dijital giriş ve çıkış için de kullanılabilir. Pinlerin ölçüm aralığı 0-5V'tur. AREF pini ve analogReference() foksionu kullanılarak alt limit yükseltilip, üst limit düşürülebilir.

**I2C- A4 veya SDA pini ve A5 veya SCL pini:** I2C haberleşmesi için bu pinler kullanılır.

**AREF:** Analog girişler için ölçüm referansı pini.

**Reset:** Resetleme işlemi için bu pin LOW yapılır. Bunun yerine kartta bulunan Reset butonuna da basılabilir.

**Not:** Arduino'da harici kesmede diğer pinler kullanılamaz duruma gelir.



Şekil 12 Arduino - Uno Devre Şeması

### 2.1.3.Arduino Uno PWM Pin

Yukarıda da belirttiğimiz gibi PWM pinleri **3,5,6,9,10** ve **11** numaralı pinlerdir. PWM çıkışları ile motor hız kontrolü, ışık parlaklık gibi analog kontrollere yapılabilmesi sağlanır. PWM sinyalleri daha detaylı anlattığımız Servo Motor Nedir? yazımızı inceleyebilirsiniz.

### 2.1.4.Arduino Uno'yu Bilgisayara Tanıtma

Arduino Uno ile programlamaya başlamadan önce bilgisayarınızda Arduino'nun kurulmuş olması gerekiyor. Detaylı Arduino kurulumuna buradan ulaşabilirsiniz.

Kurulum işlemlerini tamamladıktan sonra Arduino Uno'yu bilgisayara tanıtmak için aşağıdaki işlemleri sırasıyla uygulayın:

- **Araçlar>Kart>Arduino Uno** seçerek kartımızı belirtiyoruz.
- **Araçlar>Port** seçtikten sonra Arduino Uno'nun bağlı olduğu portu seçiyoruz.
- Böylelikle Arduino Uno, bilgisayarımıza tanımlanmış oluyor. **Yükle** butonuna basarak programı karta yüklüyoruz ve projemiz çalışmaya başlıyor

## 2.2. Dht11 Sıcaklık Nem Sensörü

DHT11 Sıcaklık ve Nem Sensörü, kalibre edilmiş dijital sinyal çıkışı ile karmaşık bir sıcaklık ve nem sensörüne sahiptir. Özel dijital sinyal toplama tekniğini ve sıcaklık ve nem algılama teknolojisini kullanarak, yüksek güvenilirlik ve mükemmel uzun süreli stabilite sağlar. Bu sensör, dirençli tipte bir nem ölçüm bileşeni ve bir NTC sıcaklık ölçüm bileşeni içerir ve mükemmel kalite, hızlı tepki, parazit önleyici özellik ve maliyet etkinliği sunan yüksek performanslı 8-bit mikrodenetleyiciye bağlanır.



Şekil 13- Dht11

Her DHT11 elemanı, laboratuvarında nem kalibrasyonunda son derece hassas olan kesinlikle kalibre edilmiştir. Kalibrasyon katsayıları, OTP hafızasında, sensörün

dahili sinyal tespit prosesi tarafından kullanılan programlar olarak saklanır. Tek kablolu seri arayüz, sistem entegrasyonunu hızlı ve kolay hale getirir. Küçük boyutu, düşük güç tüketimi ve 20 metreye kadar sinyal iletimi, en zorlu olanlar da dahil olmak üzere çeşitli uygulamalar için en iyi seçimdir. Bileşen 4-pin tek sıra pin paketidir. Bağlanması uygundur ve kullanıcıların isteğine göre özel paketler sağlanabilir.

Teknik özellikler:

Ölçüm aralığı: 20-90%RH 0-50 °C

Nem doğruluğu:  $\pm 5\%$  RH

Sıcaklık doğruluğu:  $\pm 2^\circ\text{C}$

Çalışma Voltajı: 3-5,5V

Çalışma Akımı: 0,5-2,5mA

Çalışma Akımı Ortalama: 0,2-1mA

Bekleme Modu Akımı: 100uA-150uA

Çözünürlük: % 1Rh

Tekrarlanabilirlik:  $\pm 1\%$

Tepkime Süresi: 6san - 15san

## BÖLÜM 3. KULLANILAN YAZILIM TEKNOLOJİLERİ

### 3.1. Node.js

Web teknolojilerinin ihtiyaçlar ve performans gereksinimleri doğrultusunda çeşitlilik, zenginlik göstermesiyle birlikte geliştirme ve işlemleri ele alış biçimlerinde de önemli değişiklikler söz konusu olmaya başladı. Özellikle front-end geliştirmelerde önemli söz hakkı bulunan JavaScript'in server-side geliştirmelerde de kullanılmasıyla birlikte hayatımıza pek çok farklı noktadan yeni bir platform daha girmiş oldu; **NodeJS (Node.js)**. Peki, nedir NodeJS.

#### 3.1.1. Node.js Nedir?

NodeJS JavaScript dili ile “Google Chrome’un v8 JavaScript Engine” kullanarak sunucu tarafında (server-side) ve ağ bağlantılı uygulamalar geliştirebilmemizi sağlayan bir ortamdır.

JavaScript ve Google Chrome v8 JavaScript Engine ilişkili önemli. Bu sayede, server tarafında JavaScript kodları çalıştırılabilmekte. Google Chrome internet tarayıcısının da üzerinde çalıştığı C, C++ ve JavaScript dilleri ile kodlanan v8 JavaScript Engine NodeJS yapısı ile JavaScript kodunu makine diline çevirmektedir.

2009 yılında, Joyent firması ile başlayan süreç şu anda milyonlarca geliştiricinin katkısıyla açık kaynak kodlu olarak sürdürülmekte.

#### 3.1.2. Node.js İle Neler Yapılabilir?

NodeJS ölçeklenebilir, event-driven (olay güdümlü) asenkron (asynchronous event driven JavaScript runtime), non-blocking I/O çalışma modelini kullanır. Dolayısıyla yüksek bir performans sunmaktadır. Ayrıca, ISS, Apache gibi web server kurulumları yerine kolaylıkla web server oluşturulabilmektedir.



```

const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});

```

Şekil 14- Örnek Kod Bloğu

Ek olarak real time (gerçek zamanlı) web uygulamaları web-socket teknolojisi ile performans öncelikli bir şekilde hazırlanabilmekte. Bu özellikler göz önünde bulundurulduğunda, NodeJS'in web uygulaması geliştirme sürecini oldukça etkili ve pratik bir hale getirdiğini söyleyebiliriz. Elbette, öncesinde NodeJS'in iş ele alış biçimine ve platformun gereksinimlerine aşinalık kazanmak gerekmekte. NodeJS'in sunduğu avantajları maddeler halinde derleyecek olursak;

- JavaScript ile neredeyse tüm gereksinimleri karşılayabilme,
- İş akışında, tek bir thread (iş parçacığı) ile bloklanmadan çalışabilme (thread sayısı arttıkça o kadar risk ve duruma göre performans gereksinimi söz konusu olabilir),
- Eşzamansız (Asenkron / Async) mimari ile kolayca istekleri / iş akışını yönetebilme,
- NPM (Node paket yöneticisi / package manager; ayrıca bkz. Yarn) ile bağımlılıkları (modül / paket) kolaylıkla yönetebilme.

İş ele alış biçimlerini NodeJS ile yeniden ele alan pek çok şirket / servis ciddi oranda performans artışı ve kaynak tasarrufu elde ettiklerini belirtmekteler.

Proje geliştirme sürecinde, bir web uygulaması için front-end JavaScript ile geliştirilmişse server-side tarafında da süreç JavaScript ile sürdürülebilir. Bu

geliştiricilerin daha küçük bir ekiple daha kısa zamanda daha etkin bir süreç izleyebilmelerine olanak sağlayacaktır.

Node.js için kullanılabilir olan modüllere / paketlere ise NPM web sitesi üzerinden ulaşılabilir.

### 3.2. SerialPort

```
const SerialPort = require('serialport')
```

Şekil 15- SerialPort Tanımlama Kodu

Bu paket portlarımızın üzerinde ki entegre olmuş devreler ile haberleşme ihtiyacımızı sağlayacak bir kütüphanedir. Yüksek düzeyde bir Akış Arayüzü , otomatik tespit bağlamaları ve bir dizi ayrıştırıcı akışı sağlar .

Tarihsel olarak bu tek paketti ve her şeyi içeriyordu. Sürüm 7'den beri, dahili parçalar kendi modüllerine ayrılmıştır ve ayrı olarak gerekli olabilir, bu da kullanıcının sadece ihtiyaç duyduklarını yüklemesine izin verir.

Bu, daha küçük kurulumlara ve alternatif arayüzlere, bağlantılara ve ayrıştırıcılara izin verir.

```
var serialport = require('serialport');
// List serial ports:
serialport.list(function (err, ports) {
  ports.forEach(function(port) {
    console.log(port.comName);
    portName=port.comName;
  });
});

//Seri port işlemleri için gerekli nesneler oluşturuluyor.
const Readline = require('@serialport/parser-readline')
const seriPort = new SerialPort("COM5", { baudRate: 115200 })
const parser = new Readline()
```

Şekil 16 - Projede Serial Port Kullanımı

### 3.3. Socket.io

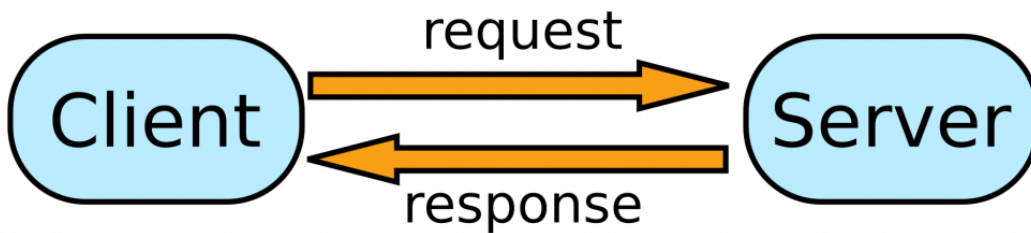
**socket-IO** bir **NodeJS** paketidir. Bu paket ile gerçek zamanlı işlemler yapabilirsiniz. **TCP** portu üzerinde kolayca anlık veri geçişi yapabilir ve projelerinizde gerçek zamanlı işlemleri kolayca gerçekleştirebilirsiniz.

#### Server (Sunucu) Nedir ?

Server , sunucu/istemci mimarisinde verilen bilgileri saklayan ve istenildiği zaman erişilebilen bir depodur .

#### Client (İstemci) Nedir ?

Client sunucu/istemci mimarisinde server'dan istekte bulunan ve server'a bilgi taşıyan ve serverdan aldığı request(yanıt)'ları taşıyan bir aracı olarak tanımlayabiliriz.



Şekil 17 - Socket.io Tanımlaması

Herhangi bir web sitesine girdiğimiz zaman Client(istemci), Server(sunucu) 'a bir sorgu gönderir. Daha sonra Server(sunucu) gelen sorguya göre bize bir sonuç verir ve istemci yanıtı bize gösterir.

Socket-IO bir node.js paketidir. NodeJS üzerinde geliştirilen bu kütüphane istemci/sunucu bağlantısının yanı sıra oturum yönetimi ve mesaj alışverişi gibi

konularda da oldukça kolaylık sağlar . Belirli bir port üzerinden açılmış bağlantının istemci tarafından dinlenmesi ve paket alışverişi mantığına dayanan bu kütüphane bir kaç satır kodla bile chat uygulaması yapımına olanak sağlar.

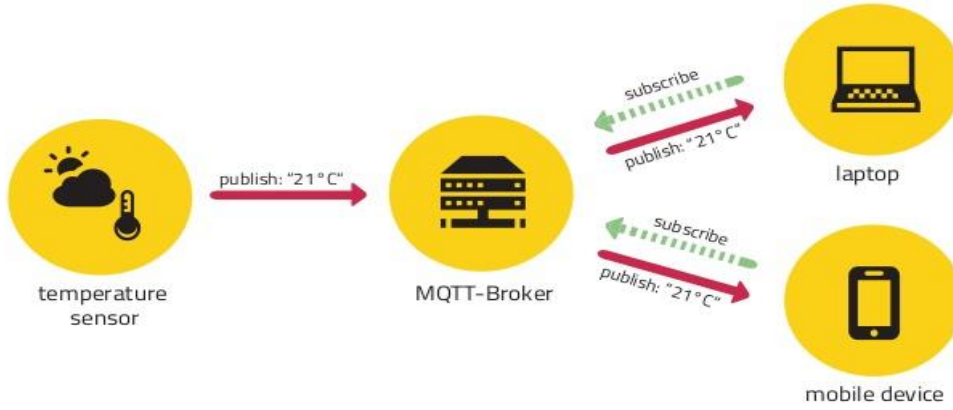
### Socket-IO kurulumu

Nodejs proje içerisinde cmd veyahut terminal ekranına **npm install socket.io –save** yazarak projenize socket.io paketini yükleyebilirsiniz

## 3.4. MQTT

### 3.4.1. MQTT Nedir?

**Message Queuing Telemetry Transport** yani MQTT mesajın karşı tarafa gönderilmesi için kullanılan bir haberleşme protokoldür. Bu haberleşme trafiğini kontrol eden yöneticiye **BROKER**, mesaj yayınına **PUBLISH** ve bu mesaj yayınına abone olanlara **SUBSCRIBE** denmektedir. Aşağıdaki görselde bu kavramlar daha iyi anlaşılacaktır.



Şekil 18 - Mqtt Broker Örneklendirmesi

MQTT de asenkron bir haberleşme kullanılmaktadır. Mesajı yayınlayan ve mesaja abone olanlar arasında veriler asenkron (eş-zamansız) olarak taşınmaktadır. Yukarıdaki görselde sıcaklık verileri (PUBLISH) haberleşme trafiğini kontrol eden yöneticiye (BROKER) gönderilir. BROKER bu verileri abone (SUBSCRIBE) online olduğu anda iletir.

İnternet üzerindeki çeşitli BROKER'lara belli konularda abone olabilirsiniz. Örnek olarak Akıllı Telefonunuzdan Uygulama Mağazasına girerek MQTT Client 'ı indirip yaşadığınız bölgedeki hava durumu için MQTT Broker 'a abone olabilirsiniz.

MQTT diğer haberleşme protokollerine göre daha basit bir yapıya sahiptir ve kolayca projelerinize entegre edilebilirsiniz. Minimum kaynak tüketimi sayesinde özellikle M2M (Machine-to-machine) haberleşmesinde kullanılmaktadır. Bu da MQTT yi IoT projeleri için vazgeçilmez bir mesajlaşma protokolü haline getirmektedir.

### 3.4.2.MQTT ÖZELLİKLERİ

- Asenkron (eş-zamansız) çalışan bir protokoldür.
- Güvenlik olarak SSL / TLS desteklemektedir.
- Minimum kaynak kullanımında bulunmaktadır.
- Broker üzerinden haberleşme temeline dayanmaktadır.
- Bilgiler MQTT protokolü üzerinden çok hızlı bir şekilde iletilebilir. (ms düzeyinde bir haberleşme)
- TCP/IP nin kullanıldığı Windows, Linux, MacOS, Android ve iOS işletim sistemlerinde çalışır.

### 3.4.3.IoT Nedir?

**IoT** yani Nesnelerin İnterneti sayesinde hayatımızda bir çok şey değişiyor. Daha da akıllı hale gelen evimizdeki elektronik cihazlar nesnelerin interneti ile birlikte farklı cihazlar ile haberleşip belirlenen kurallara göre (bazı durumlarda kendi yapay zekasını kullanarak) hayatınızı kolaylaştırmak için çalışıyorlar.

Basit bir örnek vermek gerekirse evimizdeki klima ile akıllı telefonumuz haberleşip bizim eve olan mesafemizi kontrol ediyor. Eve yaklaştığımızda klima çalışmaya başlıyor ve siz eve girdiğinizde en sevdiğiniz ideal sıcaklıkta oluyor. Veya buzdolabınız, içerisindeki yiyecekler bittiğinde sizin her gün eve dönerken kullandığınız güzergahtaki alışveriş merkezinden günlük/aylık veya haftalık tüketiminize göre sipariş veriyor, kredi kartınızla ödeme yapıyor ve size bilgi veriyor. Size kalan tek şey geçerken paketinizi almak.

### 3.4.4.MQTT ile IoT Arasındaki Bağlantı Nedir?

IoT ‘de cihazların birbirleri ile iletişimde olması sürekli veya belli aralıklarla bir haberleşme içerisinde olmasına bağlıdır. Bu noktada da MQTT daha önce belirttiğimiz özellikleri sayesinde IoT projeleri içerisinde vazgeçilmez bir haberleşme protokolüdür.

Dışarıdan bakıldığında bunu farklı protokoller ile de yapabiliriz diye düşünebilirsiniz. Ancak IoT sadece buzdolabı ve akıllı telefon arasındaki iletişim değildir. Sürekli kullandığınız anahtar, fırın, mutfak robotu, buzdolabı, araba, bisiklet, klima, müzik seti, lamba ve daha bir sürü eşyanın sürekli olarak birbiri ile iletişimde olduğunu düşünün. Ne kadar yüksek hacimde verilerin anlık olarak gidip geldiğini bir düşününce MQTT bu noktada gerçek anlamda standart bir protokol haline geliyor.

Endüstri 4.0 Devrimi IoT ile birlikte çok farklı anlamlar kazanıyor. Bu yüzden MQTT 'de Endüstri 4.0 Devrimini haberleşme protokolleri açısından olumlu yönde etkileyecek bir devrim olarak nitelendirilebilir.

### 3.4.5. Gerçek Hayatta MQTT Nerelerde Kullanıyor ?

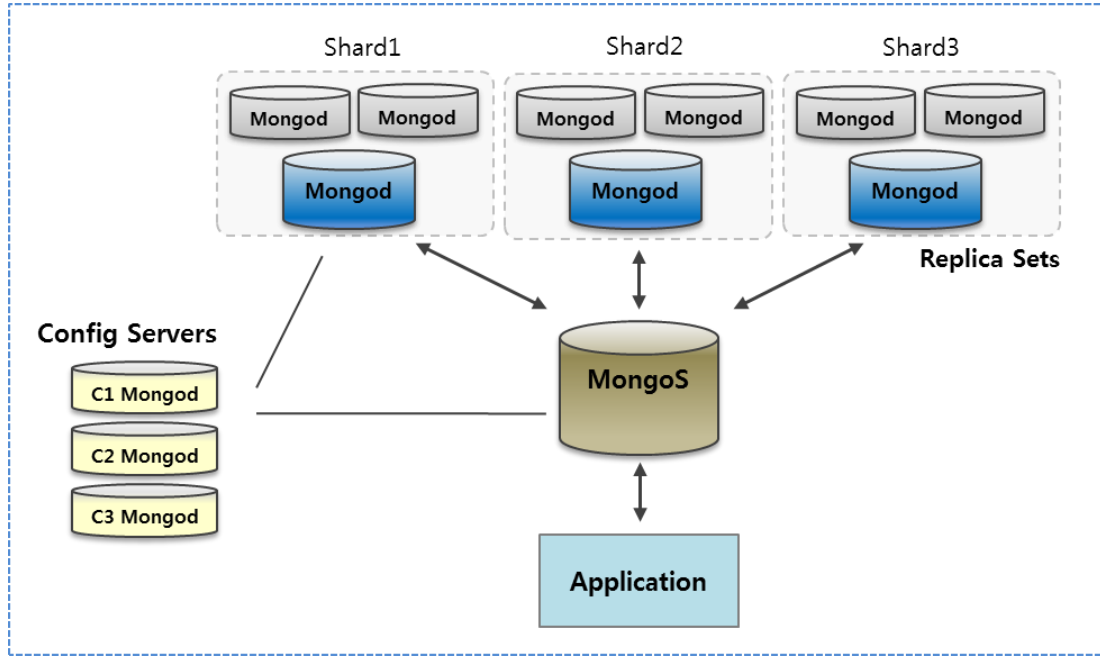
Günümüzde Akıllı Ev Kontrol Sistemleri MQTT protokolünü kullanarak ev içerisindeki bir çok (sıcaklık, nem, basınç, ışık, hareket, gaz vb...) sensöre ait veriyi anlık olarak iletmekte ve ev sahiplerini bilgilendirmektedir. Bu sistemlerin bilgilendirme fonksiyonlarının dışında olası bir kaza yada soruna karşılık erkenden önlem alabilmektedir. Ayrıca gereksiz enerji tüketimi, zaman tasarrufu ve hayatı kolaylaştırma fonksiyonları ile bu sistemler günümüzde yeni yapılan binalarda ve Akıllı Şehirlerde olmazsa olmazlar arasında yer almaktadır.

Daha yaşamın içinden bir örnek verelim. **Facebook**, online mesajlaşma uygulaması olan Facebook Messenger 'da MQTT 'nin sağladığı özellikleri kullanıyor.

Örnekler arttırılabilir tabi ki. Ancak bütün bu haberleşme sistemlerinin alt yapısında gizli kahraman MQTT.

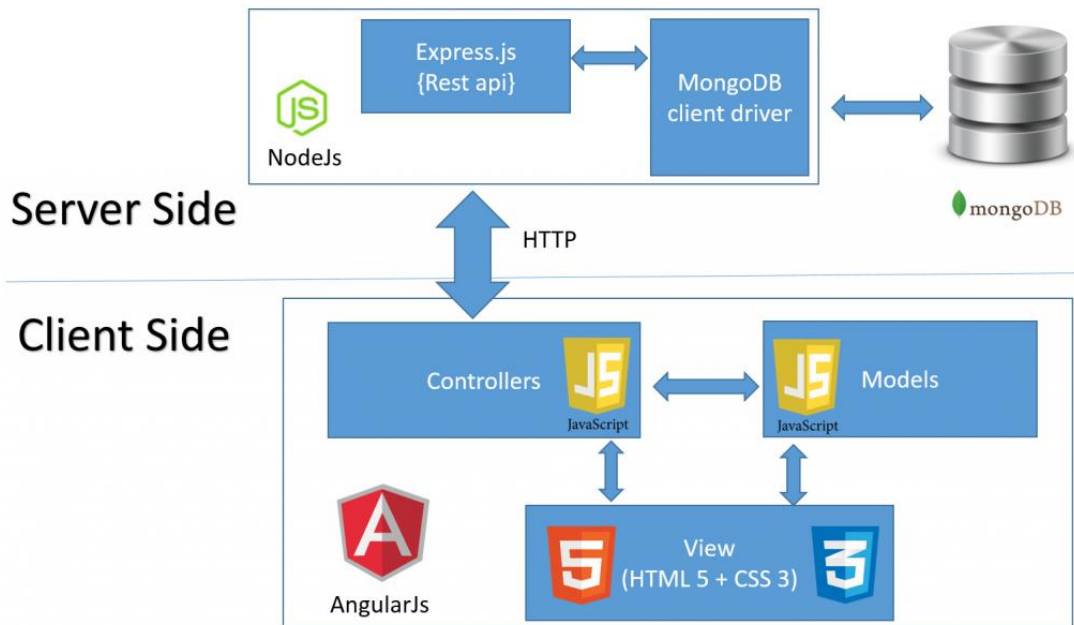
### 3.5. MongoDB

MongoDB ; NoSQL Verilerini document (belge) biçiminde saklayabilmek için kullanabileceğimiz ölçeklenebilir (scalable) bir veritabanı biçimidir.



Şekil 19 - MongoDB Örneklendirmesi

NoSQL verilerini saklayıp , son kullanıcının anlık olarak sorgulayabileceği bir sistem için MongoDB tercih nedeni olabilir.



Şekil 20 - MongoDB Çalışma Şekli



### 3.5.1.MongoDb Özellikleri

- Ölçeklenebilirdir (Scalable) . Veri boyutu arttığı durumlarda veya performans sıkıntısı yaşadığımız durumlarda makine ekleyebiliriz
- Veriler document (belge) biçiminde saklanır. Burada JSON verilerini kullanabiliriz
- Veriler JSON şeklinde saklandığı için gelen veri yapısı değişse bile kaydetme işleminde sıkıntı yaşanmaz.
- Verilerin birden fazla kopyası saklanabilir ve veri kaybı yaşanmaz (Replication)
- Veriler üzerinde index oluşturarak verilere hızlı bir biçimde ulaşabiliriz

### MongoDB TERMS

RDB	MongoDB
Table	Collection
Row	Document
Column	Field

Şekil 21 - Sql ve NoSql Benzerlik Şekli

- Database : Veritabanı
- Collection : Veritabanında kullandığımız tablo burada Collection olarak adlandırılır

- Document : Her bir kayıt(row) document olarak isimlendirilir
- Field : Tablodaki her bir kolon Field olarak adlandırılır

MongoDB üzerinde veriler JSON olarak saklanır. Bununla beraber ilişkisel veritabanı sistemlerinde farklı tablolarda saklayıp **join** yaparak kullanabileceğimiz veriler, burada sub-document olarak saklanabilir



Şekil 22 - MongoDB Json Tipi Veri Saklama Tipi

## **BÖLÜM 4. SONUÇLAR VE ÖNERİLER**

Proje istenilen çalışmaları gerçekleştirmiş gerekli sonuçlar elde edilmiştir. Elde edilen sonuçlar incelenip analiz edilip hocamız ile beraber projenin tamamlanmasına karar verilmiştir. Geliştirilebilir kısımlarına ek olarak Docker teknolojileri ve Apache kotin gibi teknolojiler eklenerek proje daha da geliştirilebilir.

## KAYNAKLAR

- [1] <https://maker.robotistan.com/arduino-uno/>
- [2] <https://www.arduino.cc/>
- [3] <https://www.youtube.com/watch?v= 2JJ29kmwRU>
- [4] <https://projeyenilik.com/dht11-nedir.html>
- [5] <https://www.robotistan.com/dht11-isi-ve-nem-sensoru-kart>
- [6] <https://www.yusufsezer.com.tr/node-js-nedir/>
- [7] <https://wmaraci.com/nedir/nodejs>
- [8] <https://nodejs.org/en/docs/>
- [9] <https://www.ebi.com.tr/blog/mqtt-nedir-iot-ile-baglantis-nedir/>
- [10] <http://mqtt.org/>

## ÖZGEÇMİŞ

Merhabalar ben Serhat Kerem EROL. 07/06/1996 tarihinde Sakarya’da dünya’ya geldim. İlköğretim ve orta öğretim hayatımı sırasıyla Bahçelievler Gazi İÖO ve Ahmet Akkoç İÖO unda tamamladım. Lise öğretimini Sakarya Fatih Anadolu Teknik Lisesi Bilgisayar Bölümü Veritabanı Programcılığı üzerine tahsil görüp. 73,08 ortalama ile mezun oldum. Lise hayatım boyunca 2 kez TÜBİTAK yarışmalarına katıldım. Birisinde c# öğretme adı altında programlama ve algoritmayı sadece lise ve üniversite de değil tüm hayatımızda kullanmak amaçlı için oluşturduğumuz şu an da bir benzeri olan udemy gibi bir sistem ile katılmıştık. İkincisinde ise evlerin rutubet alan ve güneş görmeyen kısımları için de bir prototip geliştirdik. Özel ayna ile güneş görmeyen rutubet tutan kısımları güneş görmesini sağlamıştık. 2015-2016 eğitim döneminde Sakarya Üniversitesi Bilgisayar Mühendisliği bölümünü kazanarak eğitim hayatıma devam etmiş bulunuyorum. Üniversite hayatım boyunca derslerle yetinmeyip kendimi geliştirme ortamı hazırlamaya çalıştım. 2. Sınıfta iken asp.net üzerine özel çalışmalara başladım. 2. Sınıfta iken hocalarımla önerileri ile IoT üzerine araştırmalarımı başladım. 3. Sınıfta iken Toltek Yazılım Eğitim Danışmanlık Tic. Ve San. Ltd. Şti firmasında junior developer olarak temel olarak akıllı ev sistemleri fikriyatı üzerine çalışmaya başladım. 4. Sınıfa geçmeden çalıştığım firma çalıştığım birimi kapatma kararı aldığından kaynaklı ayrılmak durumunda kaldım. Ayrıldığı dönemde donanım stajımı tamamlamak için Sakarya Üniversitesi Uzaktan Eğitim Merkezinde Donanım Stajımı tamamladım. Ekim 2018 itibari ile Argede Bilişim Teknolojileri firmasında junior developer olarak çalışmaya başladım. Aralık 2018 itibari ile okulda ki derslerime vakit ayıramadığım gerekçesi ile çalıştığım Argede Firmasından izin isteyip iş yerimden ayrıldım. Ağustos 2019 itibari ile de 20 günlük yazılım stajımı Sakarya Üniversitesi Uzaktan Eğitim Merkezinde tamamlamış bulunmaktayım. Yarım dönem okulumun uzaması sebebiyetiyle şubat 2020 itibari ile lisans mezunu olacağım. Her çalıştığım firmada kendi iş meditasyonumda “Vatanını en çok seven işini en iyi yapandır” mottosu ile işime elimden, beynimden geldiğince mesai saatlerimi aksatmayarak çalışmalarımı en iyi seviyede tutmaya çalıştım.

## BSM 498 BİTİRME ÇALIŞMASI DEĞERLENDİRME VE SÖZLÜ SINAV TUTANAĞI

KONU :Nodejs Tabanlı IoT Uygulaması

ÖĞRENCİLER (Öğrenci No/AD/SOYAD):Serhat Kerem EROL/ g151210108

Değerlendirme Konusu	İstenenler	Not Aralığı	Not
<b>Yazılı Çalışma</b>			
<b>Çalışma klavuza uygun olarak hazırlanmış mı?</b>	x	0-5	
<b>Teknik Yönden</b>			
<b>Problemin tanımı yapılmış mı?</b>	x	0-5	
Geliştirilecek yazılımın/donanımın mimarisini içeren blok şeması (yazılımlar için veri akış şeması (dfd) da olabilir) çizilerek açıklanmış mı?			
Blok şemadaki birimler arasındaki bilgi akışına ait model/gösterim var mı?			
Yazılımın gereksinim listesi oluşturulmuş mu?			
Kullanılan/kullanılması düşünülen araçlar/teknolojiler anlatılmış mı?			
Donanımların programlanması/konfigürasyonu için yazılım gereksinimleri belirtilmiş mi?			
UML ile modelleme yapılmış mı?			
Veritabanları kullanılmış ise kavramsal model çıkarılmış mı? (Varlık ilişki modeli, noSQL kavramsal modelleri v.b.)			
Projeye yönelik iş-zaman çizelgesi çıkarılarak maliyet analizi yapılmış mı?			
Donanım bileşenlerinin maliyet analizi (prototip-adetli seri üretim vb.) çıkarılmış mı?			
Donanım için gerekli enerji analizi (minimum-uyku-aktif-maksimum) yapılmış mı?			
Grup çalışmalarında grup üyelerinin görev tanımları verilmiş mi (iş-zaman çizelgesinde belirtilebilir)?			
Sürüm denetim sistemi (Version Control System; Git, Subversion v.s.) kullanılmış mı?			
Sistemin genel testi için uygulanan metotlar ve iyileştirme süreçlerinin dökümü verilmiş mi?			
Yazılımın sızma testi yapılmış mı?			
Performans testi yapılmış mı?			
Tasarımın uygulamasında ortaya çıkan uyumsuzluklar ve aksaklıklar belirtilerek çözüm yöntemleri tartışılmış mı?			
<b>Yapılan işlerin zorluk derecesi?</b>	x	0-25	
<b>Sözlü Sınav</b>			
<b>Yapılan sunum başarılı mı?</b>	x	0-5	
<b>Soruları yanıtlama yetkinliği?</b>	x	0-20	
<b>Devam Durumu</b>			
Öğrenci dönem içerisindeki raporlarını düzenli olarak hazırladı mı?	x	0-5	
<b>Diğer Maddeler</b>			
<b>Toplam</b>			

DANIŞMAN (JÜRİ ADINA):

DANIŞMAN İMZASI: