
Düzce Üniversitesi

Gömülü Sistemler Uygulaması

Isı ve Nem Sensörü Uygulaması

ÖZET

Bu çalışmada görev yaptığım hastanenin sistem odası sıcaklığını kontrol etmeyi amaçlamaktayım çünkü kliması sık sık arızalanmaktadır ve sistem odasının sıcaklığı kontrol edilmez seviyelere ulaşması halinde cihaz arızaları ve önlenemez veri kayıplara oluşabilecektir.

Bu yüzden sistem odası sıcaklığı istenmeyen seviyelere ulaştığında ısı sensörü uyarı verip dışardaki kapı üzerinde bulunan kırmızı lambayı yakmayı sağlayacak ve içerde anormal bir sıcaklığın arttığı anlaşılabilecektir.

Güvenli Sistem Odası

Esp8266 modülü sayesinde Dht11 sensöründen aldığı verileri Mqtt(Mosquitto) broker üzerinden Node-Red'e yayınlamayı ve Node-Red dashboard'u üzerinden Mqtt sayesinde gönderdiğimiz mesaj ile ledi yakacağız.

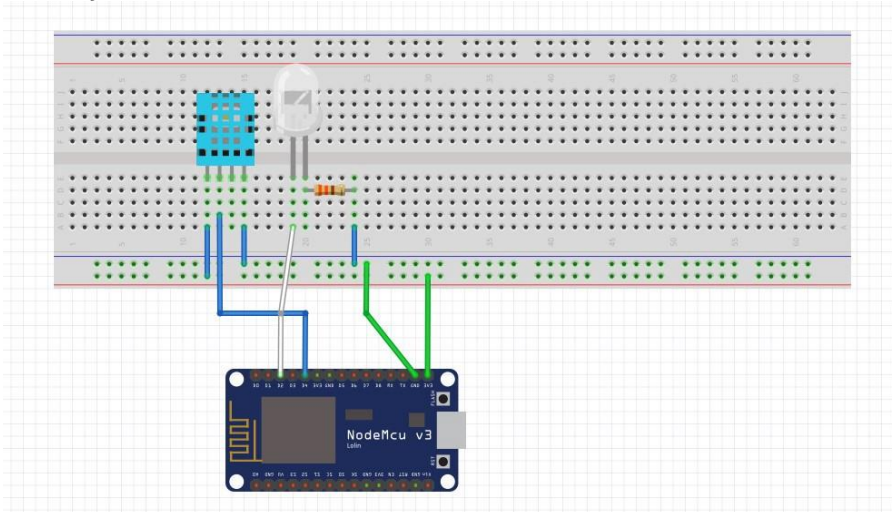
Sonrasında bu yaptığımız işlemlerin datasını influxdb aktarıp grafana üzerinde değerlerimizi izleyebileceğiz.

Keywords: Mqtt 1, Dht11 2, Esp82663

İhtiyaç listemiz

- 1 adet Esp8266 (Ben projede NodeMCU'yu kullanacağım.
- 1 adet Led
- 1 adet 330 Ohm direnç
- 7 adet Male Female Jumper Kablo
- 1 adet micro usb kablosu

Devre Şeması:



Led yerleştirirken dikkat etmeniz gereken şey ise kısa bacağa direnç bağlanacak. Ve direnç renklerini de dikkate alarak doğru bağlamalısınız.

Donanım dışında yazılım olarak ise <https://www.virtualbox.org/wiki/Downloads> adresinden indirmiş olduğum uygulama üzerine sanal ubuntu <https://ubuntu.com/download/desktop> kurdum. Ubuntu terminal ekranı açıken gereksinimleri kurmadan önce sistem güncellemesi yaptım.

```
$ sudo apt update
$ sudo apt upgrade -y
```

Sonrasında eski menü uygulamsına tercih ederek IOTstack kullandım.

```
$ git clone -b old-menu https://github.com/SensorsIot/IOTstack.git ~/IOTstack
```

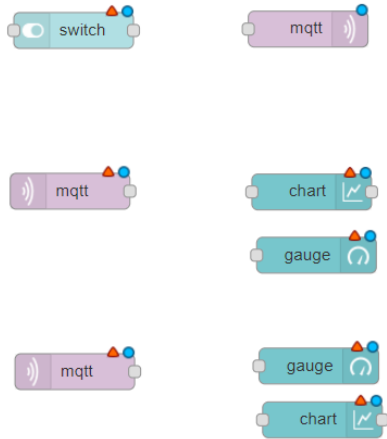
Menülerden sistemim için gereksinimleri seçip yükledim.

```
$ cd ~/IOTstack
$ ./menu.sh
```

Menülerden sistemim için gereksinimleri yükleme başlattım

```
$ cd ~/IOTstack
$ docker-compose up -d
```

Kurulum işlemlerimi tamamladıktan sonra node redi çalıştırdım terminal ekranından http ile başlayan <http://127.0.0.1:1880/> kısım node-red'in web arayüzüne giriş yaptım. Sonrasında kullanmamız gereken dashboard kısmı sol tarafta bulunmuyordu. Bunun için sağ üstte bulunan üç çizgiyi tıklayıp Ve oradan setting>palette>Install'ı seçimini gerçekleştirdim. Ara kısmına dashboard yazıp ve node-red-dashboard'u kurdum. Güncellemeler varsa burada Nodes kısmından güncellemelerini de yükledim. Sonrasında WSL terminalini yeniden başlatıp ve Node-Red'i çalıştırdım.



Minumum yukardaki görsel tarzında flowlara ihtiyaç duydum

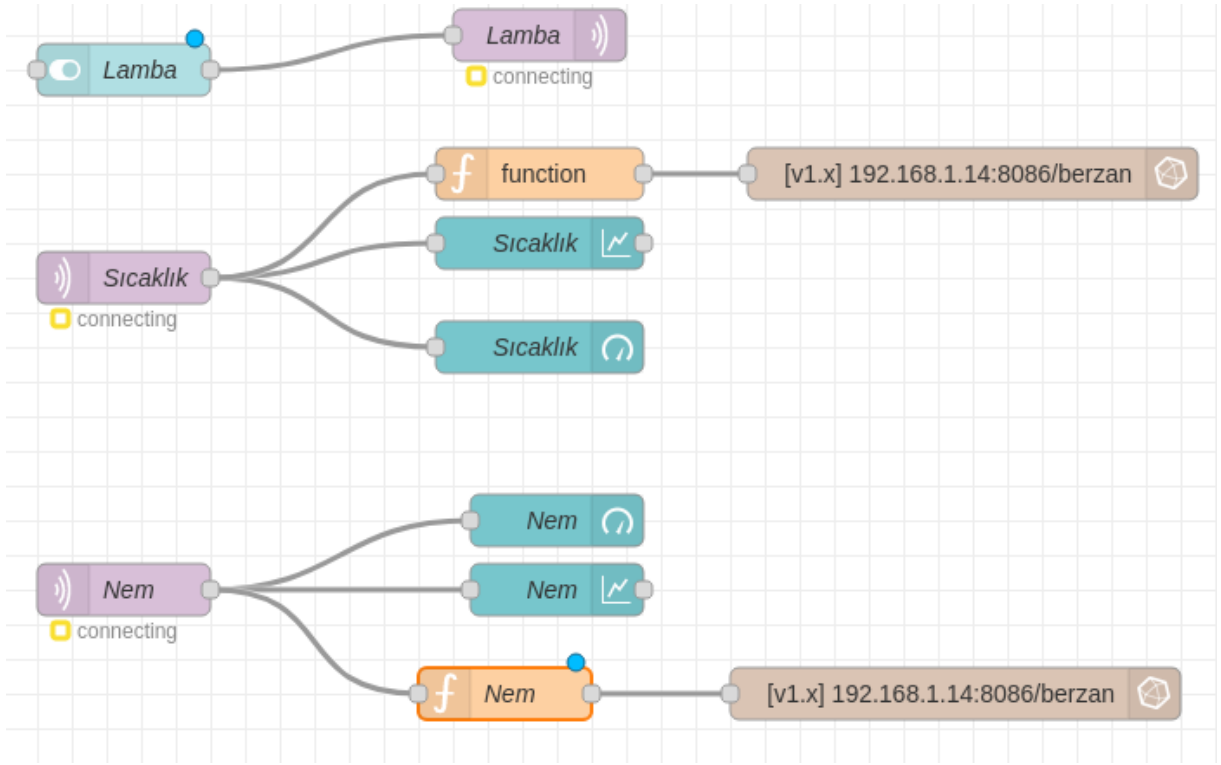
switch - ESP8266'nın çıkışını kontrol eder.

mqtt output node - ESP8266'yı Switch'in durumuna göre mesaj yayınlar

2x mqtt input nodes - Bu düğümler, ESP'den sensör verilerini almak için sıcaklık ve nem konularına Subscribe olacaktır.

chart - Sıcaklık ve Nem değerlerini grafiksel olarak göstermemizi sağlar.

2x gauge - Sıcaklık ve Nem değerlerini anlık olarak göstermemizi sağlar.



Node kurulumlarımızı yaptığımıza göre node'larımızı aşağıdaki resimdeki gibi bağlıyoruz. Oluşturduğumuz nodelerin ayarlamalarını yapmamız gerekiyor.

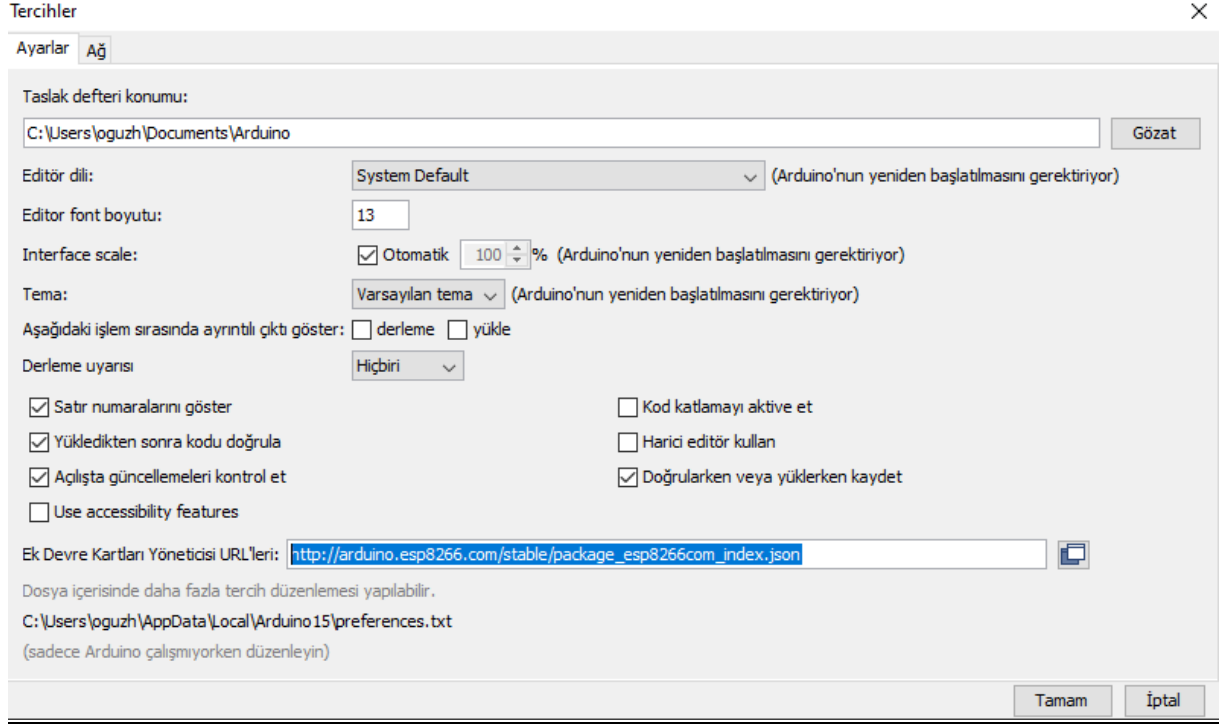
Sonrasında yaptığımız akışı sağ üstteki Deploy seçeneği ile dağıtıyoruz.



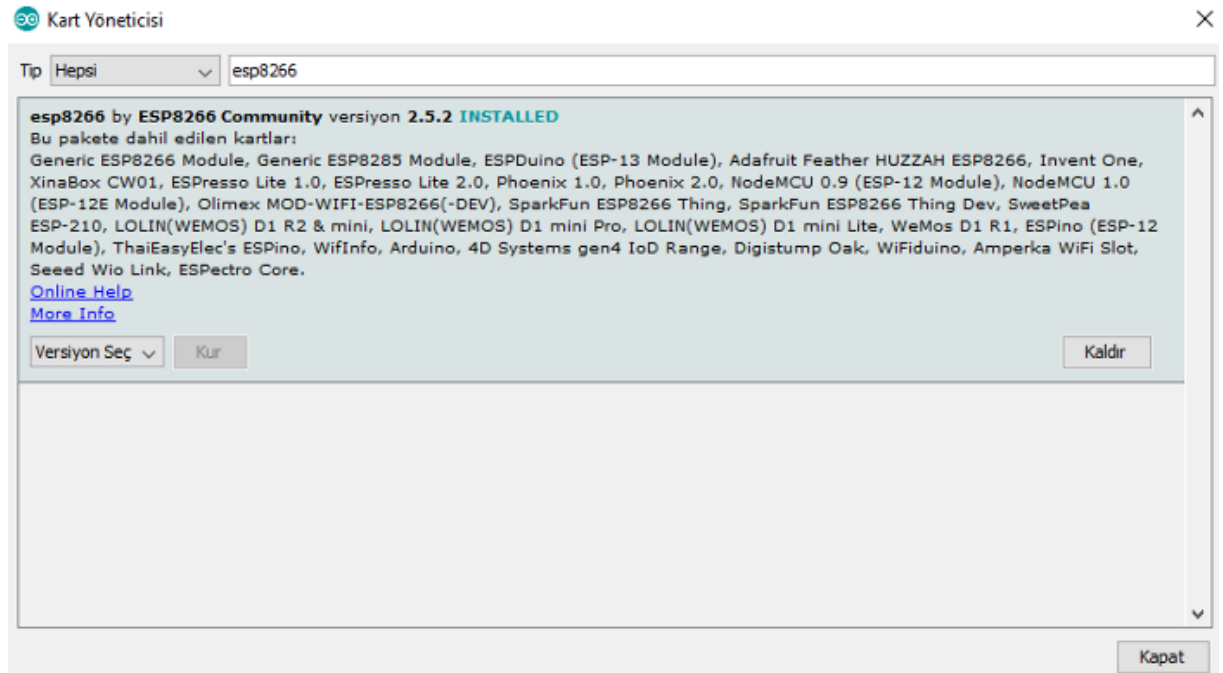
Sırada Arduino IDE ile Esp8266'mızı programlamamız gerekiyor. Bu sayede internete bağlanacak ve verilerini bilgisayarımızdaki Ubuntu'daki mqtt broker üzerinden Node-Red'i kullanarak izlememizi ve yayınlamamızı sağlar.

Arduino IDE üzerinde Dosya>Tercihler üzerinden Ek Devre Kartları Yöneticisi URL'leri kısmına aşağıdaki verilen adresi tanımlamalıyız.

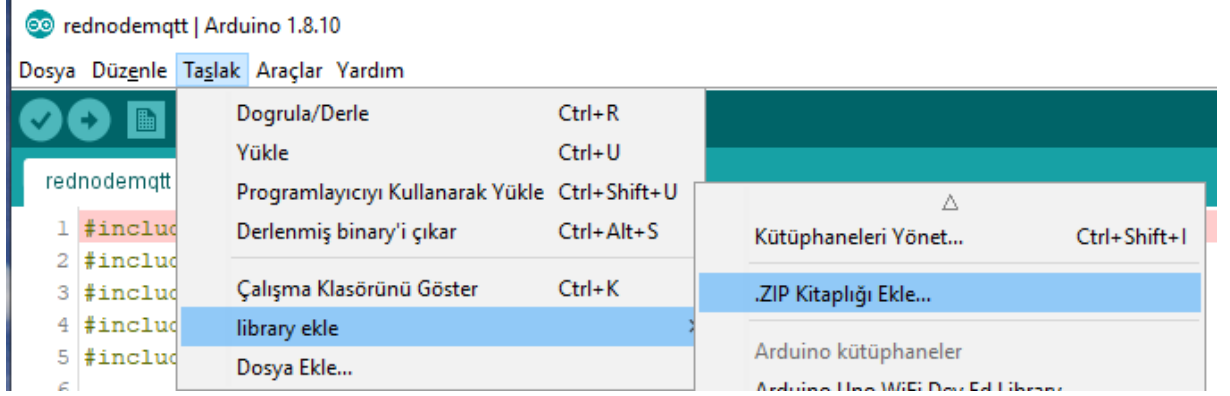
http://arduino.esp8266.com/stable/package_esp8266com_index.json



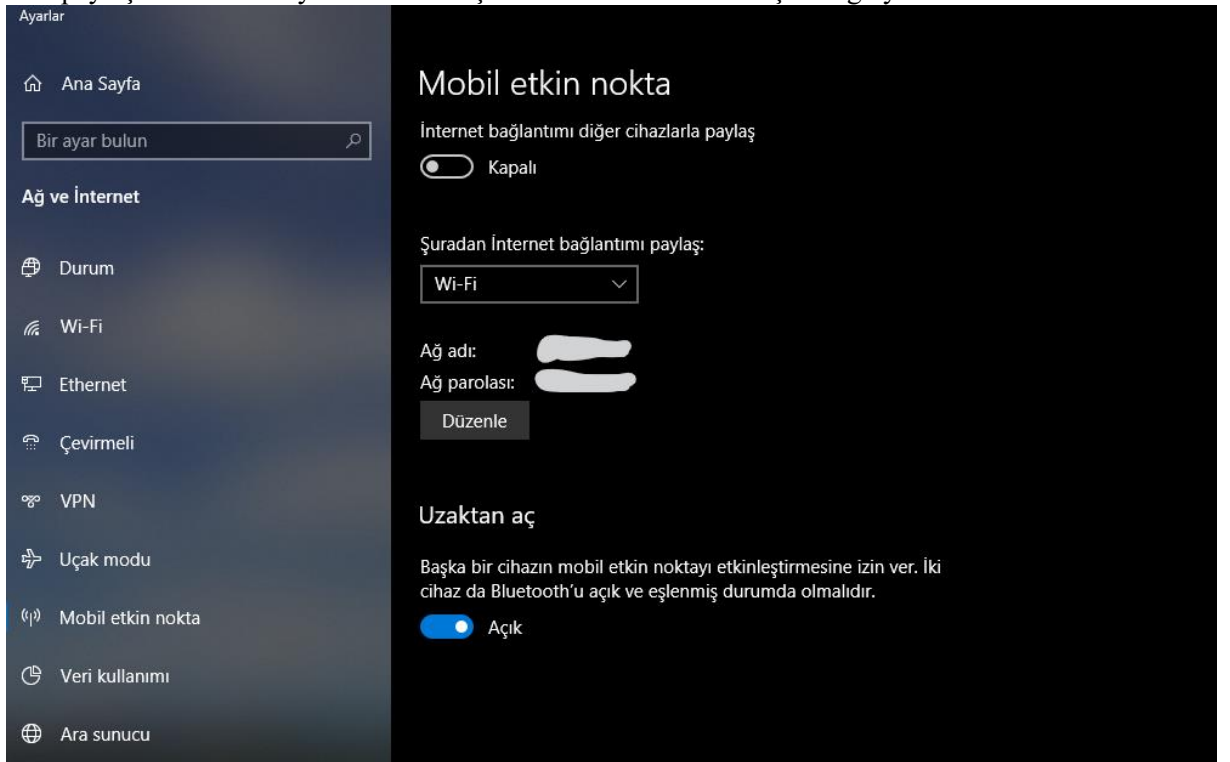
Kart Yöneticisi kısmından ara kısmına “esp8266” yazın ve aşağıdaki resimdeki gördüğünüz kütüphaneyi yüklemeliyiz.



Arduino IDE’de kullanmamız gereken önemli kütüphanelerden biri de PubSubClient kütüphanesi ile DHT11 sensörünün kütüphanesidir. Mqtt sayesinde Esp8266’dan aldığımız verileri publish ve subscribe yapmamızı sağlar. İndirdiğiniz kütüphaneyi Arduino IDE’ye taslak>library ekle>.ZIP Kitaplığı Ekle diyerek kütüphaneyi yükleyin.



Esp8266'nızı Windows 10 içerisinde bulunan Mobil Etkin Nokta sayesinde bilgisayarınızdaki internetinizi paylaşabiliriz. Wifi'yunuzun ssid ve şifresini kullanarak da erişim sağlayabilirsiniz.



Arduino IDE içerisinde Esp8266'nıza yükleyin.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
```

```
//Elinizdeki sensöre göre yorum satırını değiştir
#define DHTTYPE DHT11 // DHT 11
```

```
// Aşağıdaki kimlik bilgilerini değiştirin, böylece ESP8266'nız Wifi ağınıza bağlanır
const char* ssid = "ArenKo";
const char* password = "Aaaa1111";
```

// Aşağıdaki verileri IPv4 adresinize göre değiştirin MQTT broker oluşturacak. Cmd'ye ipconfig yazın ve Wireless LAN adapter Wi-Fi: Kısımındaki IPv4 adresinizi girin.

```
const char* mqtt_server = "192.168.1.14";
```

// espClient'i başlatır. Ev otomasyon sisteminizde çalışan birden fazla ESP varsa, espClient adını değiştirmelisiniz.

```
WiFiClient espClient;
```

```
PubSubClient client(espClient);
```

```
// DHT Sensor - GPIO 5 = D4, ESP-12E NodeMCU board
```

```
const int DHTPin = 2;
```

```
// Lamp - LED - GPIO 4 = D2, ESP-12E NodeMCU board
```

```
const int lamp = 4;
```

```
DHT dht(DHTPin, DHTTYPE);
```

```
long now = millis();
```

```
long lastMeasure = 0;
```

```
void setup_wifi() {
```

```
    delay(10);
```

```
    // Wifi ağına bağlanıyoruz.
```

```
    Serial.println();
```

```
    Serial.print("Bağlanıyor ?! ");
```

```
    Serial.println(ssid);
```

```
    WiFi.begin(ssid, password);
```

```
    while (WiFi.status() != WL_CONNECTED) {
```

```
        delay(500);
```

```
        Serial.print(".");
```

```
    }
```

```
    Serial.println("");
```

```
    Serial.print("WiFi bağlandı - ESP IP adres: ");
```

```
    Serial.println(WiFi.localIP());
```

```
}
```

```
void callback(String topic, byte* message, unsigned int length) {
```

```
    Serial.print("Mesaj topic'e ulaştı: ");
```

```
    Serial.print(topic);
```

```
    Serial.print(". Mesaj: ");
```

```
    String messageTemp;
```

```
    for (int i = 0; i < length; i++) {
```

```
        Serial.print((char)message[i]);
```

```
        messageTemp += (char)message[i];
```

```
    }
```

```
    Serial.println();
```

```
if(topic=="oda/lamba"){
```

```
    Serial.print("Odadaki lambanın durumu: ");
```

```
    if(messageTemp == "on"){
```

```
        digitalWrite(lamp, HIGH);
```

```
        Serial.print("On");
```

```
    }
```

```
    else if(messageTemp == "off"){
```

```
        digitalWrite(lamp, LOW);
```

```
        Serial.print("Off");
```

```
    }
```

```
}
```

```
    Serial.println();
```

```
}
```

```
void reconnect() {
```

```

while (!client.connected()) {
  Serial.print("MQTT bağlantısı kurulmaya çalışılıyor...");
  // bağlanmayı dene
  if (client.connect("ESP8266Client")) {
    Serial.println("MQTT bağlantısı kuruldu");
    client.subscribe("oda/lamba");
  } else {
    Serial.print("başarısız oldu, rc=");
    Serial.print(client.state());
    Serial.println(" 5 saniye içerisinde yeniden deniyor");
    //yukarıdaki 5 saniye aşağıda delay içerisinde milisaniye cinsinden değiştirebilirsiniz
    delay(5000);
  }
}

void setup() {
  pinMode(lamp, OUTPUT);

  dht.begin();

  Serial.begin(9600);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}

void loop() {
  if (!client.connected()) {
    reconnect();
  }
  if(!client.loop())
    client.connect("ESP8266Client");
  now = millis();
  // Ywni sıcaklık ve nem değerlerini her 5 saniyede bir publish eder
  if (now - lastMeasure > 5000) {
    lastMeasure = now;
    float h = dht.readHumidity();
    // Sıcaklığı Celsius olarak oku (default)
    float t = dht.readTemperature();
    // Sıcaklığı Fahrenheit olarak oku (isFahrenheit = true)
    float f = dht.readTemperature(true);

    // Herhangi bir hata durumunda hata mesajı bildirir ve yeniden dener.
    if (isnan(h) || isnan(t) || isnan(f)) {
      Serial.println("DHT sensordan okuma yapılamıyor!");
      return;
    }

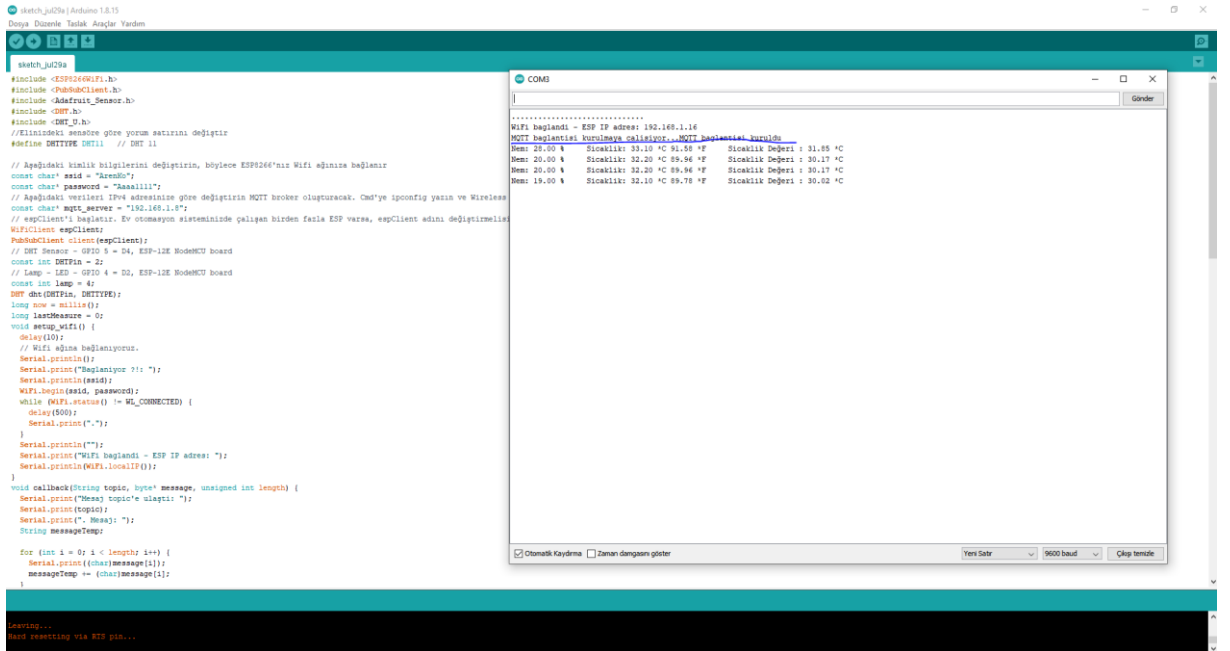
    // Celcius olarak oku
    float hic = dht.computeHeatIndex(t, h, false);
    static char temperatureTemp[7];
    dtostrf(hic, 6, 2, temperatureTemp);

    static char humidityTemp[7];
    dtostrf(h, 6, 2, humidityTemp);
    // Sıcaklık ve nem değerlerini mqtt ye publish eder.
    client.publish("oda/sicaklik", temperatureTemp);
  }
}

```

```
client.publish("oda/nem", humidityTemp);
```

```
Serial.print("Nem: ");
Serial.print(h);
Serial.print(" %\t Sıcaklik: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print(f);
Serial.print(" *F\t Sıcaklik Değeri : ");
Serial.print(hic);
Serial.println(" *C ");
// Serial.print(hif);
// Serial.println(" *F");
}
}
```



The screenshot shows the Arduino IDE interface. The sketch editor on the left contains the code for the ESP8266. The COM port monitor on the right shows the output of the sketch, which includes the MQTT connection status and the temperature and humidity readings.

```
sketch_ju29a
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT11.h>
//Elimindeki sensöre göre yorum satırını değiştir
#define DHTTYPE DHT11 // DHT 11

// Aşağıdaki kimlik bilgilerini değiştirin, böylece ESP8266'nız Wifi ağına başlar
const char* ssid = "Aresko";
const char* password = "Aaaalllll";
// Aşağıdaki verileri IPv4 adresinize göre değiştirin MQTT broker oluşturacak. Cnd'ye ipconfing yazın ve Wireless
const char* mqtt_server = "192.168.1.8";
// espClient'i başlatır. Ev otomasyon sisteminde çalışan birden fazla ESP varsa, espClient adını değiştirmelisiniz.
WiFiClient espClient;
PubSubClient client(espClient);
// DHT Sensör - GPIO 5 = D4, ESP-12E NodeMCU board
const int DHTPin = 2;
// Lamp - LED - GPIO 4 = D2, ESP-12E NodeMCU board
const int Lamp = 4;
DHT dht(DHTPin, DHTTYPE);
long now = millis();
long lastMeasure = 0;
void setup_wifi() {
  delay(10);
  // Wifi ağına başlanıyoruz.
  Serial.println();
  Serial.print("Bağlanıyor ?! ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi bağlandı - ESP IP adresi: ");
  Serial.println(WiFi.localIP());
}
void callback(String topic, byte* message, unsigned int length) {
  Serial.print("Mesaj topic'e ulaştı: ");
  Serial.print(topic);
  Serial.print(" - Mesaj: ");
  String messageTemp;
  for (int i = 0; i < length; i++) {
    Serial.print((char)message[i]);
    messageTemp += (char)message[i];
  }
}
```

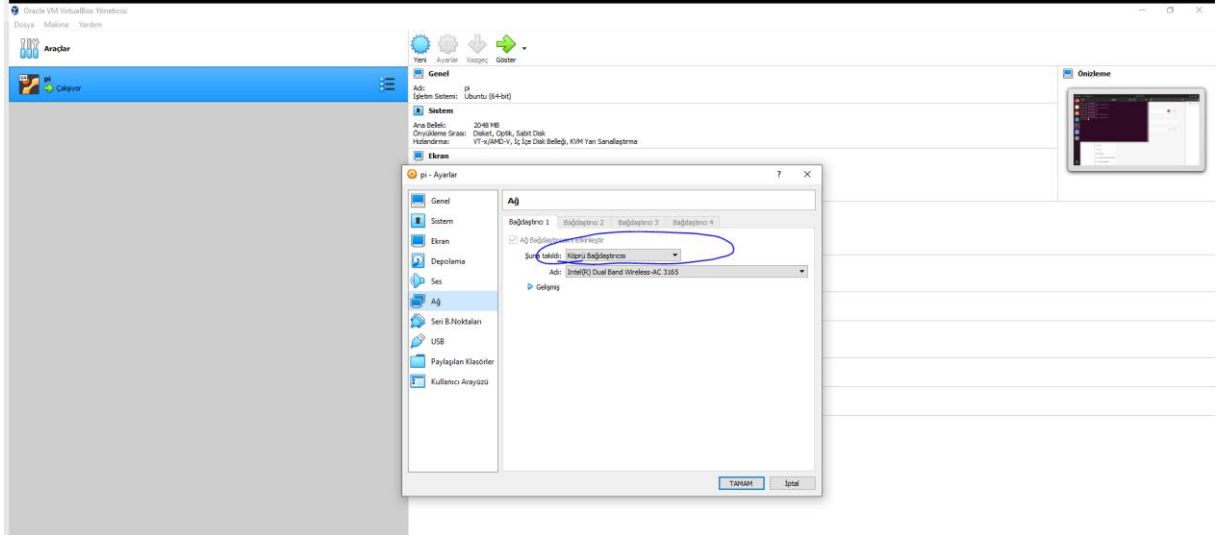
COM1

```
.....
Wifi bağlandı - ESP IP adresi: 192.168.1.16
MQTT bağlantısı kurulmaya çalışılıyor...MQTT bağlantısı yapıldı
Nem: 28.00 % Sıcaklık: 33.10 *C 91.58 *F Sıcaklık Değeri : 31.85 *C
Nem: 20.00 % Sıcaklık: 32.20 *C 89.96 *F Sıcaklık Değeri : 30.17 *C
Nem: 20.00 % Sıcaklık: 32.20 *C 89.96 *F Sıcaklık Değeri : 30.17 *C
Nem: 19.00 % Sıcaklık: 32.10 *C 89.78 *F Sıcaklık Değeri : 30.02 *C
```

Özetli Kaydırma Zamanı dengesi göster Yeni Satır 9600 baud Çıkıp terkide

Leaving...
Hard resetting via RTS pin...

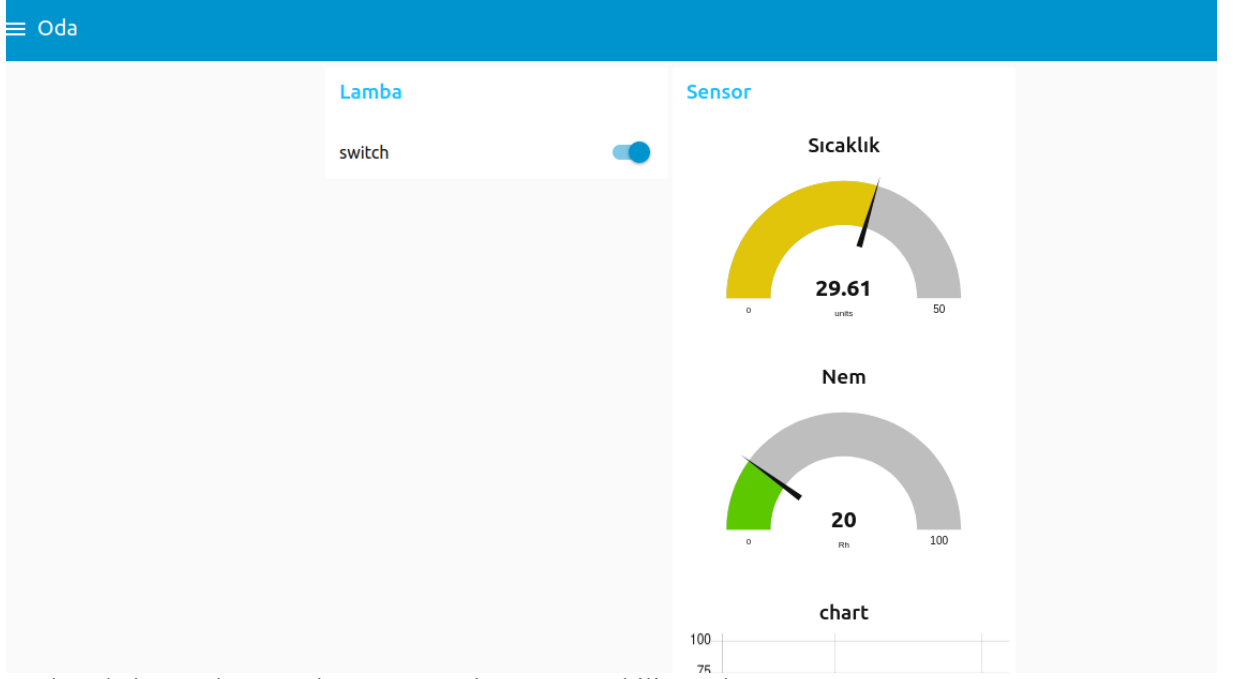
Kodumuzu çalıştırdıktan sonra esp ip bilgilerini Mqtt bağlantısının tamamlandığını ve sıcaklık nem değerlerini görebiliyoruz



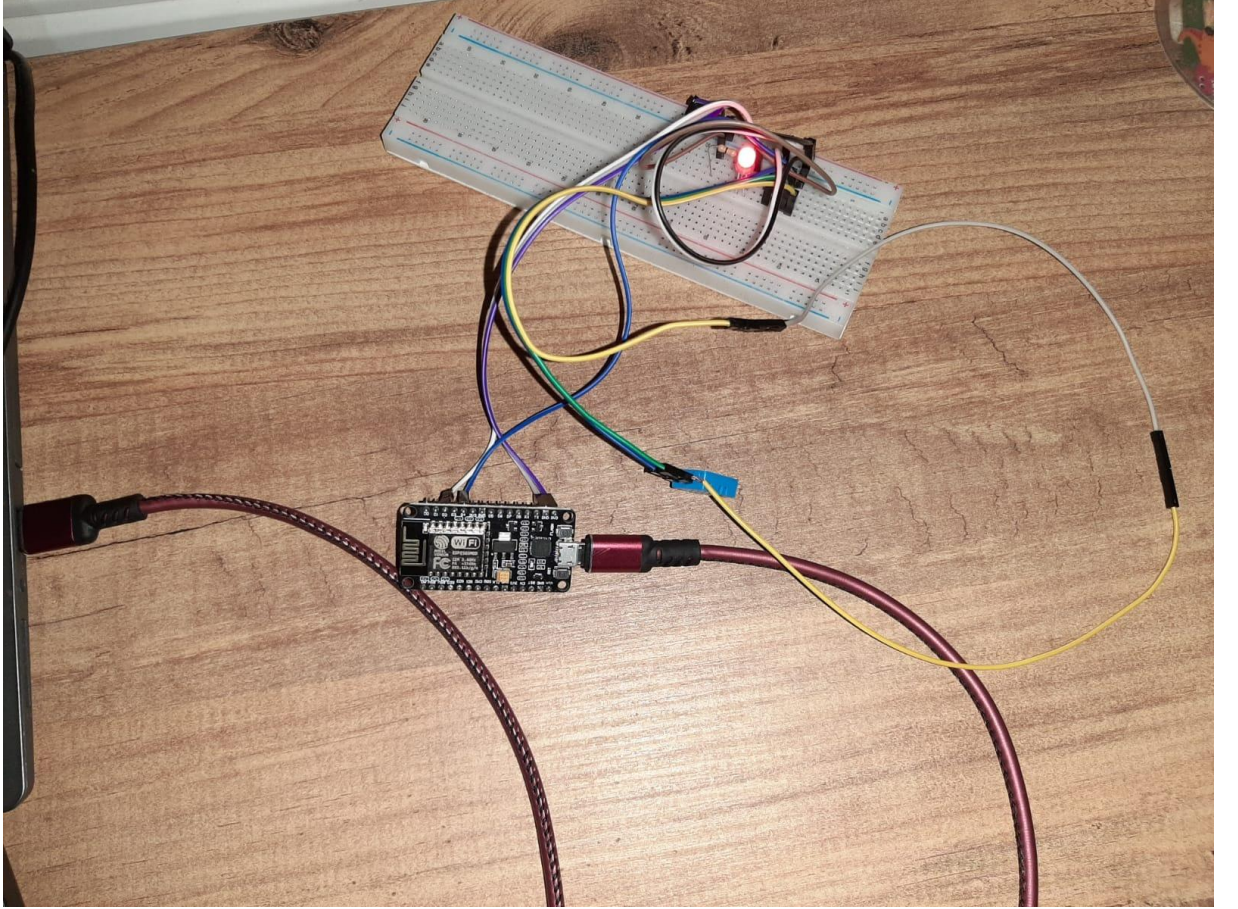
Kurulum sırasında mqtt bağlantısında sorun yaşamamak için köprüleme yapmamak gerekmektedir.

Name	State	Quick actions	Stack	Image	Created	Published Ports	Owner
nodered	healthy	[Icons]	iotstack	iotstack_nodered	2021-07-29 18:49:20	1880:1880	adm
portainer	running	[Icons]	iotstack	portainer/portainer	2021-07-29 18:30:02	9000:9000	adm
grafana	running	[Icons]	iotstack	grafana/grafana	2021-07-29 18:30:02	3000:3000	adm
mosquitto	running	[Icons]	iotstack	iotstack_mosquitto	2021-07-29 18:30:02	1883:1883	adm
influxdb	running	[Icons]	iotstack	influxdb:1.8	2021-07-29 18:30:02	2003:2003 8083:8083 8086:8086	adm

9000 portundan erişeceğimiz portainer.io uygulamalarımızı tek havuzdan yönetmemizi kolaylaştıracaktır.



Nod Red ekranından uygulamamızın çalıştığını görebiliyor olacağız.



Projemizin çalıştığına dair görüntümüz.

VERİ TABANI INFLUXDB

```
docker exec -it influxdb /bin/bash
influx
```

Komutlarıyla veri tabanına bağlandım.

```
Create database berzan
```

Komutuyla veri tabanına oluşturdum.

```
use berzan
```

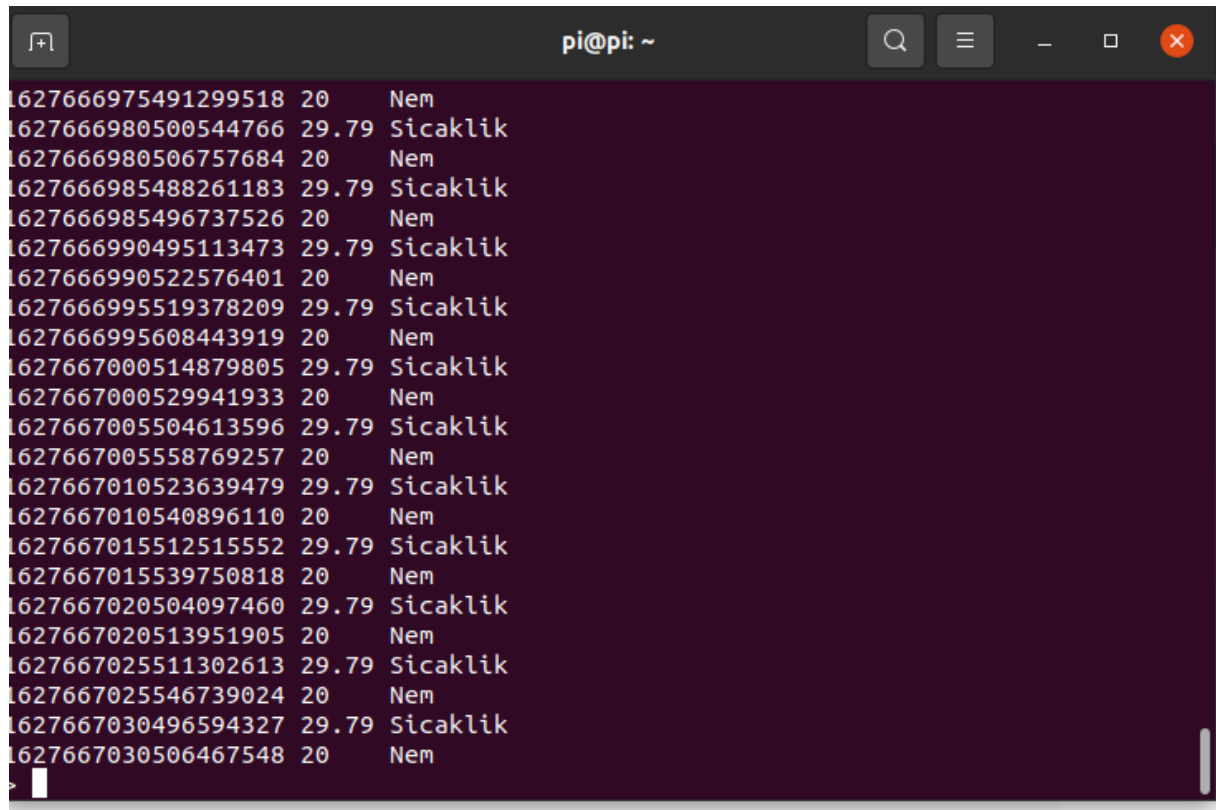
Komutuyla veri tabanına eriştim

```
show measurements
```

Komutuyla node red üzerinden bir data gönderdik ve görüntüledim.

```
Select * from heat
```

Komutuyla veri tabanındaki datamı görüntüledim.



```
pi@pi: ~
1627666975491299518 20 Nem
1627666980500544766 29.79 Sicaklik
1627666980506757684 20 Nem
1627666985488261183 29.79 Sicaklik
1627666985496737526 20 Nem
1627666990495113473 29.79 Sicaklik
1627666990522576401 20 Nem
1627666995519378209 29.79 Sicaklik
1627666995608443919 20 Nem
1627667000514879805 29.79 Sicaklik
1627667000529941933 20 Nem
1627667005504613596 29.79 Sicaklik
1627667005558769257 20 Nem
1627667010523639479 29.79 Sicaklik
1627667010540896110 20 Nem
1627667015512515552 29.79 Sicaklik
1627667015539750818 20 Nem
1627667020504097460 29.79 Sicaklik
1627667020513951905 20 Nem
1627667025511302613 29.79 Sicaklik
1627667025546739024 20 Nem
1627667030496594327 29.79 Sicaklik
1627667030506467548 20 Nem
```

Veri taban tarafındaki datalarım.

GRAFANA

3000 portu ve standart admin admin kullanıcı adı şifresi ile grafanaya eriştim.

Data Sources / InfluxDB

Type: InfluxDB

Settings

Name

InfluxDB

Default

☐

Query Language

InfluxQL

HTTP

URL

http://192.168.1.6:8080

Access

Server (default)

Help

Whitelisted Cookies

New tag (enter key to add)

Timeout

Auth

Basic auth

☐

With Credentials

☐

TLS Client Auth

☐

With CA Cert

☐

Skip TLS Verify

☐

Forward OAuth Identity

☐

Custom HTTP Headers

+ Add header

InfluxDB Details

Database Access

Setting the database for this datasource does not deny access to other databases. The InfluxDB query syntax allows switching the database in the query. For example: `SHOW MEASUREMENTS ON _internal` or `SELECT * FROM "_internal"."database" LIMIT 10`

To support data isolation and security, make sure appropriate permissions are configured in InfluxDB.

Database

berzan

User

admin

Password

configured

Reset

HTTP Method

Choose

Min time interval

10s

Görüntüleyeceğimiz database bilgilerini influxdb de oluşturdum.

