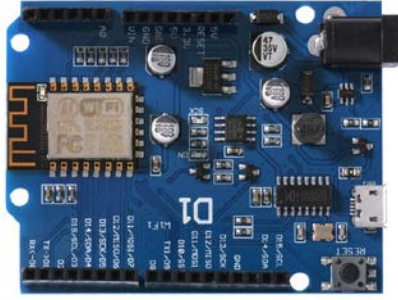


Uygulama Adı: MQTT Protokolü İle Localde IoT Uygulaması

No:

Uygulamanın Tanıtımı:

Esp8266 modülüne sahip Wemos D1 Mini IoT cihazı ile LM35 sensöründen algılanan ortam sıcaklık bilgisini MQTT protokolü ile adafruit IoT platformuna gönderen uygulama.



IoT Cihazı



Mqtt Sunucu

Şekil 1: Sistem mimarisi

Ekipman Listesi ve Kullanılan Teknolojiler:

- Wemos D1 mini ya da (Arduino + Esp8266 modülü)
- Localde çalışması için bir PC
- MQTT protokolü

Kullanılan Teknolojilere Yönelik Teknik Bilgiler:

Wemos D1 Mini

Arduino geliştirme ortamı (IDE), Arduino bootloader (Optiboot), Arduino kütüphaneleri, AVR Dude (Arduino üzerindeki mikrodenetleyici programlayan yazılım) ve derleyiciden (AVR-GCC) oluşur. Arduino yazılımı bir geliştirme ortamı (IDE) ve kütüphanelerden oluşur. IDE, Java dilinde yazılmıştır ve Processing adlı dilin ortamına dayanmaktadır. Kütüphaneler ise C ve C++ dillerinde yazılmıştır ve AVR-GCC ve AVR Libc. ile derlenmiştir.

Wemos D1 kartını Arduino IDE'nizde tanımlı kartlar arasına ekleyebilmek için **Dosya > Tercihler** sekmesindeki ekranda **"Ek Devre Kartları Yöneticisi URL'leri"** kutusuna aşağıda verilen linki ekleyiniz.

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

Esp8266

Kolayca wireless ağlara bağlanmayı sağlayan modül. esp8266-01'den başlayıp esp8266-12'ye kadar giden versiyonları bulunuyor. Kendi firmware'inizi yazıp yükleyerek başka hiçbir şeye ihtiyaç duymadan uygulama geliştirebiliyoruz. AT+ ile başlayan komutları göndererek bağlanılabilir wi-fi ağlarının listelenmesi, wi-fi adı ve

şifresinin gönderilmesiyle ağa bağlanması, ağ üzerinden bir sunucuyla tcp bağlantısı kurup istemci olarak veri alışverişi yapılması, yine tcp üzerinde server olarak kullanılması gibi işlemler yapılabilir.

Wemos D1 mini kartında ESP8266 kütüphanelerini eklemek için Arduino IDE’de **Araçlar > Kart > Kart Yöneticisi** ekranından ESP8266 aratıp, kurunuz.

MQTT

MQTT (Message Queuing Telemetry Transport), yayınlama ve abone olma mantığına dayanan telemetry mesajlaşma protokolüdür. Makineler arası haberleşmede kullanılmaktadır. Benzer protokollerden ayrılan en önemli özelliği ise hafif (lightweight) olması ve bu sayede bir çok platformda rahatlıkla kullanılabilmesidir.

MQTT Server portu 1883’tür.

MQTT haberleşme protokolü kullanarak haberleşebilmek için aşağıdaki linkte verilen kütüphaneyi Arduino uygulamamıza **Taslak > library ekle > . ZIP Kitaplığı Ekle** seçeneği ile eklemeliyiz.

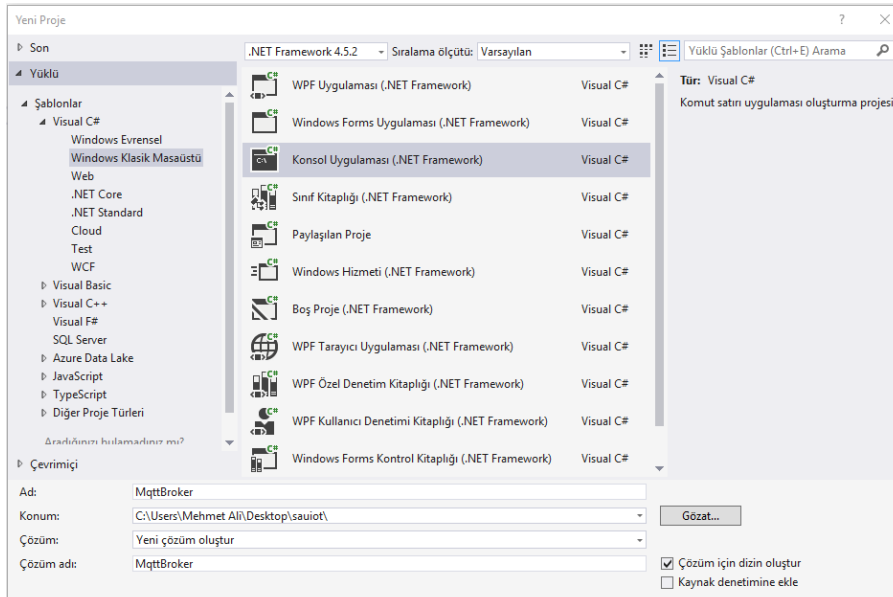
<https://github.com/knolleary/pubsubclient>

Server tarafının Mqtt Protokolü ile Kurulumu

1. Yeni bir konsol uygulaması oluşturun
2. **Nuget** paket galerisinden **MQTTnet** paketini proje başvurularına ekleyin
3. Yazılımı oluşturun.
4. Yazılımı çalıştırın. **Erişim izni** isterse **mutlaka** onaylayın.
5. Sunucu bilgisayarınızın IP adresini not edin.

1. Konsol projesinin oluşturulması:

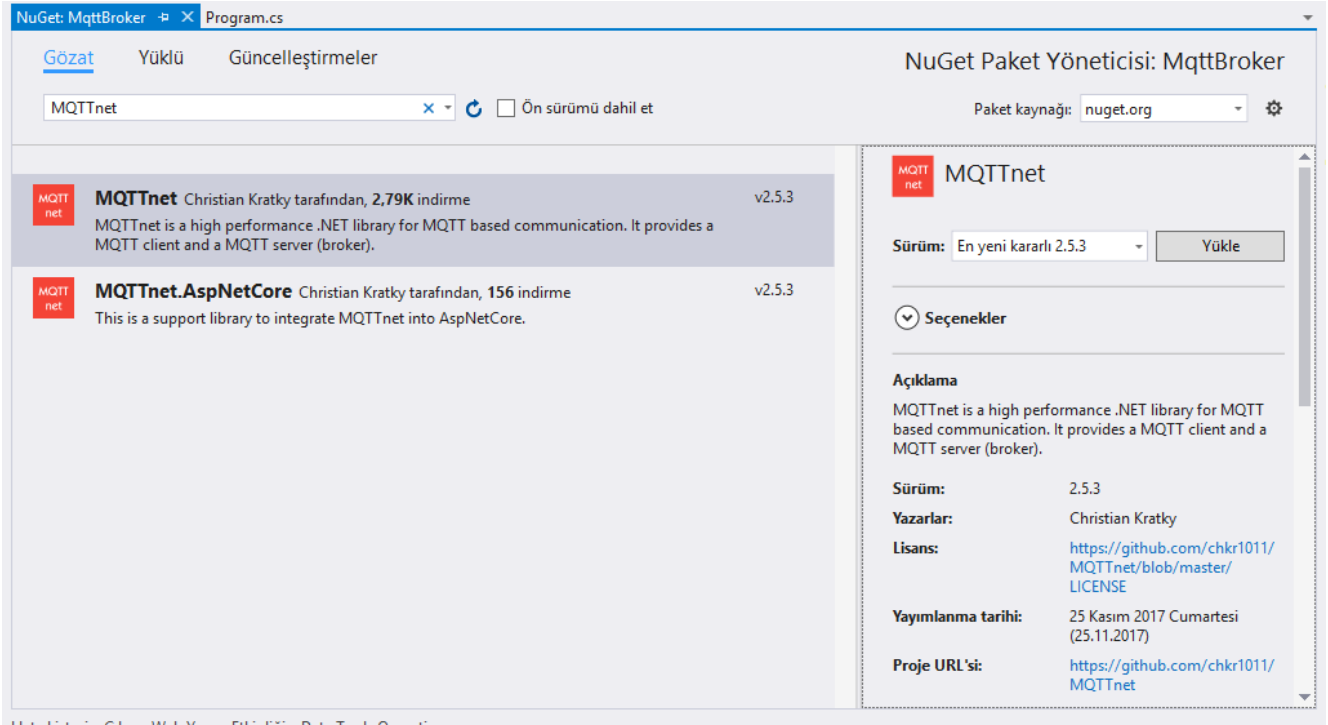
Konsol uygulaması oluşturmak için; **Dosya > Yeni > Proje** ye tıklayın. Açılan pencerede **konsol uygulamasını** seçin ve **Tamam** diyerek projeyi ekleyin.



Şekil 2: Konsol Projesinin Oluşturulması

2. MQTTnet paketini ekleyin.

Çözüm gezgininde oluşan proje üzerine sağ tıklayarak “Nuget paketlerini yönet” seçeneğine tıklayın. Açılan ekranda “Gözet” sekmesine “MQTTnet” yazarak aratın. Listelenenlerden “MQTTnet” paketini seçin “Yükle” butonuna basın.



Şekil 3: MQTTnet Paketinin Eklenmesi

3. Yazılımı Oluşturun

Aşağıdaki kodları Program.cs sınıfına yapıştırın.

```
using MQTTnet.Core.Server;
using System;
using MQTTnet;
using System.Text;

namespace MqttBroker
{
    class Program
    {
        static void Main(string[] args)
        {
            //MqttNet https://github.com/chkr1011/MQTTnet

            var options = new MqttServerOptions() { };

            int port = new MqttServerOptions().GetDefaultEndpointPort();
            Console.WriteLine("Port: " + port);

            var server = new MqttFactory().CreateMqttServer();

            server.ApplicationMessageReceived += (s, e) =>
```

```
{
    Mqtt cs = new Mqtt()
    {
        Dt = DateTime.Now,
        Konu = e.ApplicationMessage.Topic,
        Mesaj = Encoding.UTF8.GetString(e.ApplicationMessage.Payload),
        Retain = e.ApplicationMessage.Retain,
    };

    switch (e.ApplicationMessage.QualityOfServiceLevel)
    {
        case MQTTnet.Core.Protocol.MqttQualityOfServiceLevel.AtMostOnce:
            cs.ServisKalitesi = 0;
            break;
        case MQTTnet.Core.Protocol.MqttQualityOfServiceLevel.AtLeastOnce:
            cs.ServisKalitesi = 1;
            break;
        case MQTTnet.Core.Protocol.MqttQualityOfServiceLevel.ExactlyOnce:
            cs.ServisKalitesi = 2;
            break;
        default:
            break;
    }

    try
    {
        //MqttVt vt = new MqttVt();
        //vt.Ekle(cs);

        Console.WriteLine("### Mesaj alındı ###");
        Console.WriteLine($"Konu = {cs.Konu}");
        Console.WriteLine($"Mesaj = {cs.Mesaj}");
        Console.WriteLine($"Tarih = {cs.Dt}");
        Console.WriteLine($"Servis = {cs.ServisKalitesi}");
        Console.WriteLine($"Retain = {cs.Retain}");
        Console.WriteLine();
    }
    catch (Exception ex)
    {
        Console.WriteLine("### HATA OLUSTU ###");
        Console.WriteLine("hata: " + ex.Message);
        throw;
    }
};

server.StartAsync(options);

Console.WriteLine("Press any key to exit.");
Console.ReadLine();

server.StopAsync();
}

internal class Mqtt
{
    public int ID { get; set; }

    /// <summary>
```

```
/// Verinin sunucuya ulaştığı zaman
/// </summary>
public DateTime Dt { get; set; }

/// <summary>
/// Client(Arduino) tan gelen veri, bilgi ve ya mesaj
/// </summary>
public string Mesaj { get; set; }

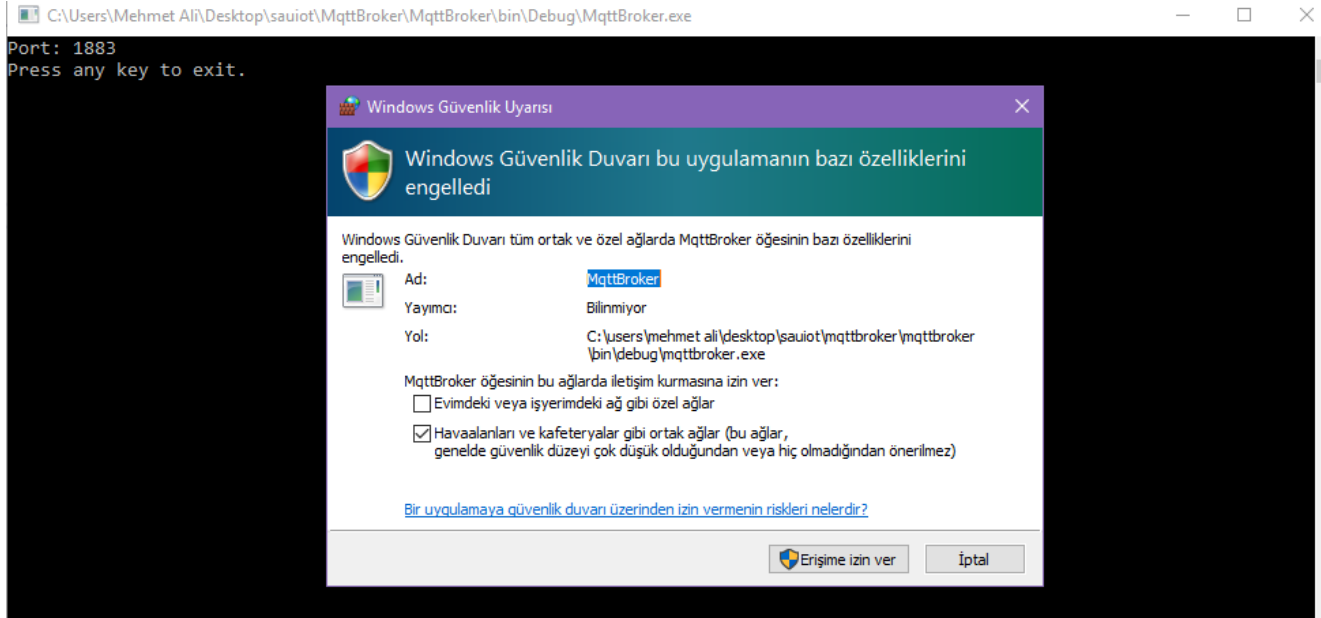
/// <summary>
/// Clienttan gelen mesajın konusu (Topic)
/// </summary>
public string Konu { get; set; }

/// <summary>
/// Mesaj teslim kalitesi
/// </summary>
public int ServisKalitesi { get; set; }

/// <summary>
/// Gelen mesajın saklanması.
/// </summary>
public bool Retain { get; set; }
}
```

4. Yazılımı Çalıştırın

Yazılımı çalıştırın ve çıkan güvenlik duvarı erişim izni uyarısına **Erişime izin ver** butonuna tıklayarak cevap verin.



Şekil 4: Yazılımın Çalıştırılması Esnasında Güvenlik Erişim İzni Verilmesi

5. IP Adresinizi Not Edin

İnternet cihazınızdan gönderilen verilerin sunucuya ulaşması için; sunucu bilgisayarın IP adresinin esp8266 Mqtt Kütüphanesine ait kodlarda ilgili yere yazılması gerekmektedir. IP adresini öğrenmek için; **Komut istemcisi**ni çalıştırın ve **ipconfig** yazarak “enter”a basın.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. Tüm hakları saklıdır.

C:\Windows\System32>ipconfig

Windows IP Configuration

Wireless LAN adapter Yerel Ağ Bağlantısı* 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

    Connection-specific DNS Suffix  . : mshome.net
    Link-local IPv6 Address . . . . . : fe80::1ddc:4992:549e:33df%4
    IPv4 Address. . . . . : 192.168.137.68
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.137.1

Tunnel adapter Yerel Ağ Bağlantısı* 11:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

C:\Windows\System32>
```

Şekil 5: IP Adresinin Elde Edilmesi

Şekilde vurgulanan IP adresidir. Not: Her bilgisayarda farklı olabilir.

Uygulamanın Wemos D1 Mini Kodları

Mqtt kütüphaneyi yükledikten sonra Arduino da Mqtt Client oluşturmak için; **Dosya > Örnekler > PubSubClient > mqtt_esp8266** ya tıklayın ve ya aşağıdaki kod örneğini yapıştırın.

Kod örneğindeki **SSID** kablosuz ağ ismini, **password** kablosuz ağ şifresini ve **mqtt_server** kısmı da sunucu adresini temsil ediyor. Bu alanların ağ bağlantınıza göre uygun şekilde doldurulması gerekiyor. Mqtt_server kısmına önceki adımda not ettiğimiz IP adresini yazmalıyız.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>

// Update these with values suitable for your network.

const char* ssid = "...ağ ismi...";
const char* password = "...ağ parolası...";
const char* mqtt_server = "...sunucu IP adresi...";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void setup()
{
    pinMode(BUILTIN_LED, OUTPUT);    // Initialize the BUILTIN_LED pin as an output
```

```
Serial.begin(115200);
setup_wifi();
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
}

void setup_wifi()
{
    delay(10);
    // We start by connecting to a WiFi network
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);

    WiFi.begin(ssid, password);

    while (WiFi.status() != WL_CONNECTED)
    {
        delay(500);
        Serial.print(".");
    }

    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length)
{
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++)
    {
        Serial.print((char)payload[i]);
    }
    Serial.println();

    // Switch on the LED if an 1 was received as first character
    if ((char)payload[0] == '1')
    {
        digitalWrite(BUILTIN_LED, LOW); // Turn the LED on (Note that LOW is the voltage
level                                         // but actually the LED is on; this is because
                                         // it is active low on the ESP-01)
    }
    else
    {
        digitalWrite(BUILTIN_LED, HIGH); // Turn the LED off by making the voltage HIGH
    }
}

void reconnect()
{
    // Loop until we're reconnected
    while (!client.connected())
    {

```

```
Serial.print("Attempting MQTT connection...");
// Attempt to connect
if (client.connect("ESP8266Client"))
{
    Serial.println("connected");
    // Once connected, publish an announcement...
    client.publish("outTopic", "hello world");
    // ... and resubscribe
    client.subscribe("inTopic");
}
else
{
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    // Wait 5 seconds before retrying
    delay(5000);
}
}
}
void loop()
{
    if (!client.connected())
    {
        reconnect();
    }
    client.loop();

    long now = millis();
    if (now - lastMsg > 2000)
    {
        lastMsg = now;
        ++value;
        snprintf(msg, 75, "hello world %ld", value);
        Serial.print("Publish message: ");
        Serial.println(msg);
        client.publish("outTopic", msg);
    }
}
```

Kaynaklar:

1 – Mehmet Ali Ebleme, “Nesnelerin İnterneti Haberleşme Protokollerinin Başarım Analizi”, Yüksek Lisans Tez Çalışması, Danışman Doç. Dr. Cüneyt BAYILMIŞ 2017.