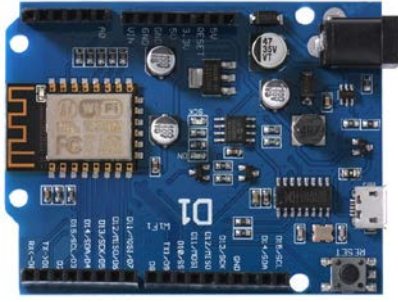


Uygulama Adı: CoAP Protokolü İle IoT Uygulaması

No:

Uygulamanın Tanıtımı:

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



IoT Cihazı



CoAP Sunucu

Şekil 1: Sistem mimarisi

Ekipman Listesi ve Kullanılan Teknolojiler:

- Wemos D1 mini ya da (Arduino + Esp8266 modülü)
- Sunucu görevini üstlenecek bilgisayar
- CoAP 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’izde 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.

CoAP

CoAp (Constrained Application Protocol – Kısıtlı Uygulama Protokolü), IETF (Internet Engineering Task Force – İnternet Mühendisliği Görev Gücü) tarafından tasarlanmış bir uygulama katmanı protokolüdür.

Adından da anlaşılacağı gibi birincil amacı kısıtlı kaynaklara sahip cihazlar üzerinde ve kısıtlı bant genişliğine sahip ağlarda çalışmaktır. CoAp, tasarımı basit tutmak için UDP üzerinde çalışır. Makineden makineye veri gönderimi sunar.

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

<https://github.com/automote/ESP-CoAP>

Kütüphaneyi uygulama olarak açmak için ise **Yeni > Örnekler > Esp-CoAP Simple library > coapclient** seçeneğini tıklamalıyız. Açılan pencerede kodlara birkaç değişiklik yaparak kullanmaya başlayabiliriz.

```
/*
ESP-COAP Client
*/
#include <ESP8266WiFi.h>
#include "coap_client.h"

//instance for coapclient
coapClient coap;

//WiFi connection info
const char* ssid = "*****"; //Kablosuz internet bağlantı adı
const char* password = "*****"; //Kablosuz internet bağlantı parolası

//ip address and default port of coap server in which your interested in
IPAddress ip(192,168,137,68);
int port = 5683; //CoAP kütüphanesinin varsayılan portu 5683 tür.

// coap client response callback
void callback_response(coapPacket &packet, IPAddress ip, int port);

// coap client response callback
void callback_response(coapPacket &packet, IPAddress ip, int port) {
```

```
char p[packet.payloadlen + 1];
memcpy(p, packet.payload, packet.payloadlen);
p[packet.payloadlen] = NULL;

//response from coap server
if (packet.type == 3 && packet.code == 0)
{
    Serial.println("ping ok");
}

Serial.println(p);
}

void setup()
{

    Serial.begin(115200);

    WiFi.begin(ssid, password);
    Serial.println(" ");

    // Connection info to WiFi network
    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);

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

    Serial.println("");
    Serial.println("WiFi connected");

    // Print the IP address of client
    Serial.println(WiFi.localIP());

    // client response callback.
    // this endpoint is single callback.
    coap.response(callback_response);

    // start coap client
```

```
coap.start();

//get request to server (arguments ip address of server,default port,resource(uri))
int msgid = coap.get(ip, port, "light");

//observe request (arguments ip address of server,deafult port,resource name,interger(0) )
//int msgid= coap.observe(ip,port,"light",0);

//reset observe cancel
//int msgid=coap.observecancel(ip,port,"resoucenam");

}

int i = 0;
char m[4];

void loop()
{
    bool state;

    sprintf(m, "%d", i);
    // Requests

    //get request
    //int msgid = coap.get(ip,port,"hello");

    //put request
    //arguments server ip address,default port,resource name, payload,payloadlength
    //int msgid =coap.put(ip,port,"resourcenam","0",strlen("0"));

    //post request
    //arguments server ip address,default port,resource name, payload,payloadlength
    int msgid = coap.post(ip, port, "hello", m, 4);

    //delete request
    //int msgid = coap.delet(ip,port,"resourcenam");

    //ping
    //int msgid=coap.ping(ip,port);

    // int msgid=coap.observe(ip,port,"obs",0);

    state = coap.loop();
    Serial.print("state=");
    Serial.println(state);
}
```

```
if (state == true)
    i = i + 1;

Serial.print("i=");
Serial.println(i);

// if (i == 3)
//{
//Serial.println("cancel observe");
//coap.observeCancel(ip,port,"resourcename");
//}

Serial.println(msgid);
delay(1000);
}
```

CoAP Sunucusu

CoAP Client' ı olan Wemos D1 mini cihazından gelen verileri okumak ve istediğimiz işlemleri gerçeklemek için bir CoAP Sunucu uygulaması oluşturmamız gerekir. Bunu Visual Studio yazılımını kullanarak yapabiliriz.

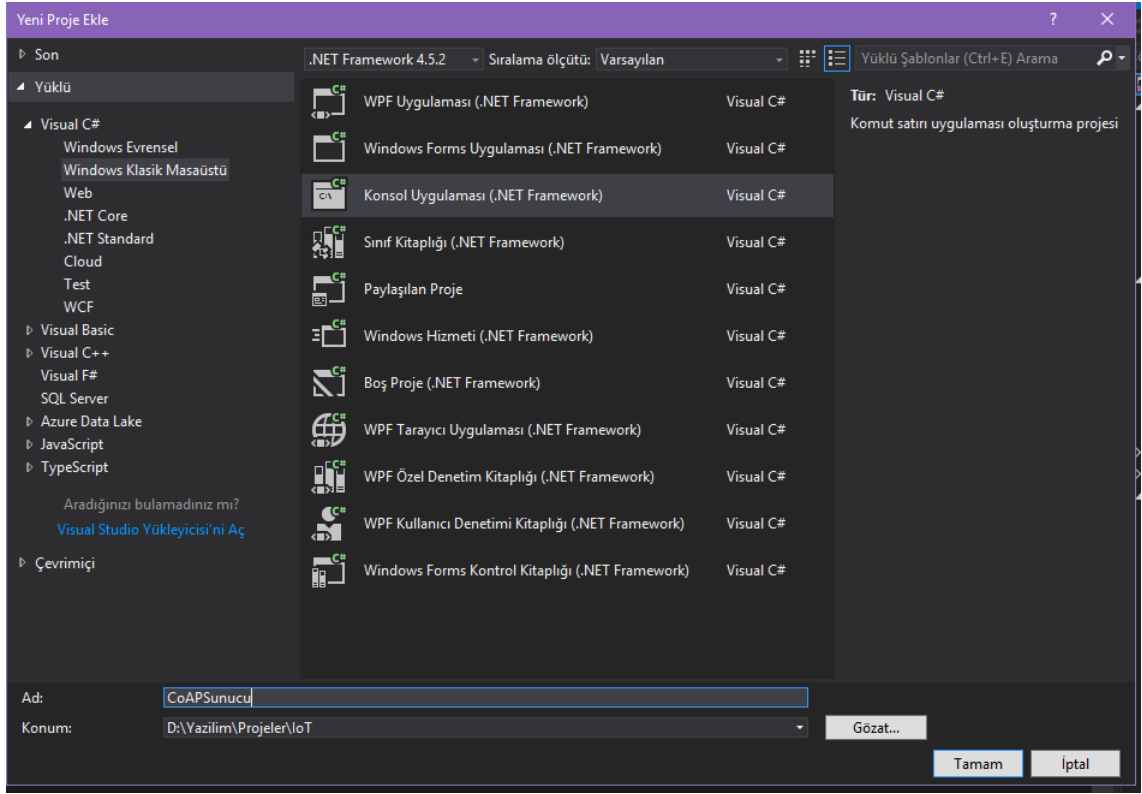
Bunun için;

1. Bir Konsol uygulaması oluşturun
2. **Nuget** paket gallerisinden “CoAP” paketini proje başvurularına ekleyin.
3. CoAP Paketlerini tanımlayan bir **resource** oluşturun
4. Resource u aktifleştiren main kodlarını yazın

1. Konsol uygulaması oluşturun.

Konsol uygulaması oluşturmak için;

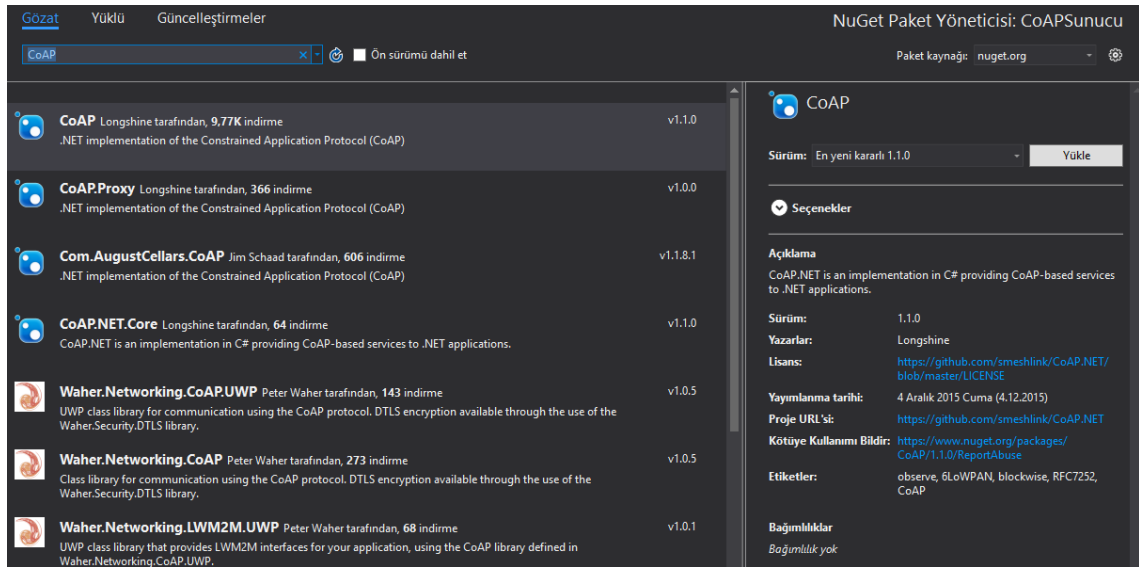
Dosya > Yeni > Proje ye tıklayın. Açılan pencerede **konsol uygulamasını** seçin ve Tamam diyerek projeyi ekleyin.



Şekil 2: Console Uygulaması Oluşturmak

2. CoAP paketini eklemek

Çözüm gezgininde oluşan proje üzerine sağ tıklayarak “Nuget paketlerini yönet” seçeneğine tıklayın. Açılan ekranda “Gözet” sekmesine “CoAP” yazarak aratın. En üst sırada gelen paket olan “CoAP” paketini yükleyin.



Şekil 3: Coap Paketini Ekleme

3. Resource Oluşturma

Resource'lar CoAP mesajlarının sunucu tarafında hangi listeye ekleneceğini ifade eder. Bir sunucuya birden fazla resource sınıfı eklenebilir. Böylece farklı işlemler için farklı resourcelar kullanılabilir. Her resource kendi içinde istenen çalışma mantığına uygun olarak özelleştirilebilir.

Resource sınıfını oluşturmak için "**Resources**" adında bir klasör ekleyin. Ardından bu klasörün içerisine "**HelloWorldResource.cs**" adında bir sınıf ekleyin. Ardından aşağıdaki kodları bu sınıfa yazın.

```
using CoAP.Server.Resources;
using System;

namespace CoAPSunucu.Resources
{
    class HelloWorldResource : Resource
    {
        public HelloWorldResource(String name)
            : base(name)
        {
            Attributes.Title = "GET a friendly greeting!";
            Attributes.AddResourceType("merhaba");
        }

        protected override void DoPost(CoapExchange exchange)
        {
            Console.WriteLine("Payload String: " + exchange.Request.PayloadString);
            Console.WriteLine(" Size: " + exchange.Request.PayloadSize);
            Console.WriteLine("");

            exchange.Respond(exchange.Request.PayloadString + " arrived");
        }
    }
}
```

4. Main kodlarının yazımı

Programın başlangıç sınıfına aşağıdaki kodları ekleyin.

```
CoapServer server = new CoapServer();

server.Add(new HelloWorldResource("hello"));

try
{
    Console.WriteLine("DefaultPort: "+ server.Config.DefaultPort);
    server.Start();

    Console.Write("CoAP server [{0}] is listening on", server.Config.Version);

    foreach (var item in server.EndPoints)
    {
        Console.Write(" ");
        Console.Write(item.LocalEndPoint);
    }
    Console.WriteLine();
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
Console.WriteLine("Press any key to exit.");
Console.ReadKey();
```

Burada ikinci satırda “hello” ifadesi `HelloWorldResource` sınıfı için **key** vazifesi görmektedir. Onun için Arduino CoAP tarafında işlem yaparken resource adı için aynı ifade kullanılmak zorundadır.

Kaynaklar:

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