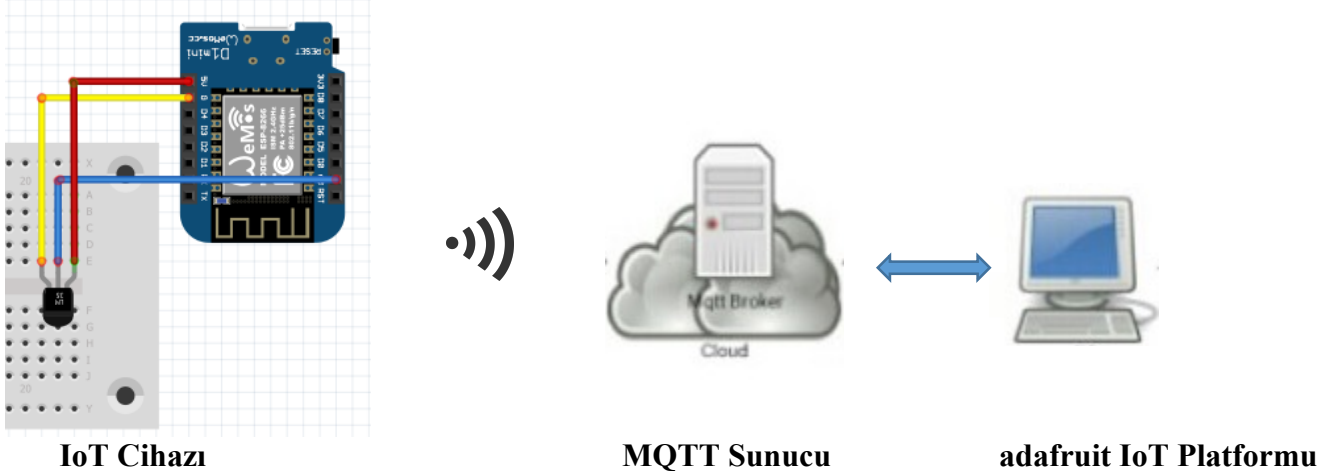


Uygulama Adı:	MQTT Protokolü İle IoT Uygulaması	No:	
---------------	-----------------------------------	-----	--

### 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.



Şekil 1. Sistem mimarisi

### Ekipman Listesi ve Kullanılan Teknolojiler:

- Wemos D1 mini ya da (Arduino + Esp8266 modülü)
- LM35 sıcaklık sensörü
- BreadBord
- Jumper Kablo
- Adafruit IoT platformu
- 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 Arduinio IDE'nizde tanımlı kartlar arasına ekleyebilmek için **Dosya > Tercihler** sekmesindeki ekranda **"Ek Devre Kartları Yöneticisi URLleri"** kutusuna aşağıda verilen linki ekleyiniz.

[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](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.

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

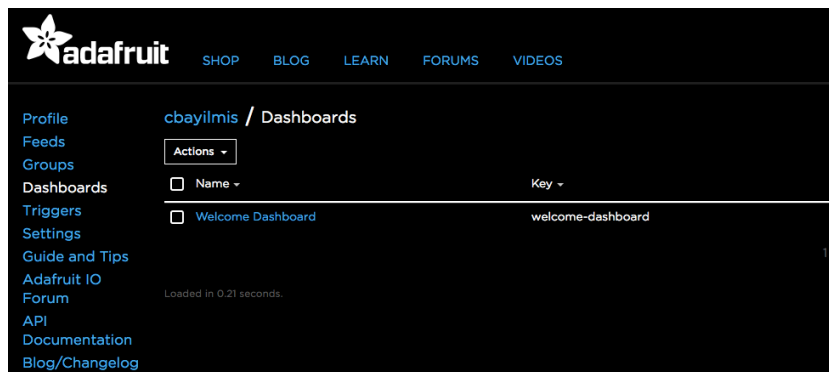
[https://github.com/adafruit/Adafruit\\_MQTT\\_Library](https://github.com/adafruit/Adafruit_MQTT_Library)

## adafruit.io (Dashboard) IoT Platformu (Web Servis Teknolojisi)

Uygulamanın web üzerinden kontrolü ve kolay yönetilebilmesi için IoT platformu olarak **adafruit** kullanacağız. **adafruit** IoT platformu grafik, buton, harita, resim vb. arayüzlerin hızlı bir şekilde kullanılabilmesini sağlamaktadır.

MQTT gibi IoT haberleşme (web servis) protokollerini destekler.

io.adafruit.com adresinden üye olunduktan sonra Şekil ....'de görüldüğü üzere <https://io.adafruit.com/kullaniciadi/dashboards> adresindeki arayüz aracılığıyla IoT uygulamanıza yönelik paneli (*dashboard*) oluşturabilirsiniz.



Şekil 2. Adafruit.io ilk giriş Dashboard arayüzü

IoT uygulamamızın kontrolü için yeni bir panel (dashboard) oluşturmak için **Action** sekmesinden **Create a New Dashboard** ile yeni dashboard'un adını "**IoT**" olarak tanımladık. Bu arayüzden yeni dashboard oluşturmak mümkünken mevcut dashboard üzerinden düzenlemeler yapabilirsiniz.

Create a new Dashboard ✕

Name  
IoT

Description  
|

Cancel Create

Şekil 3. Yeni bir dashboard oluşturma

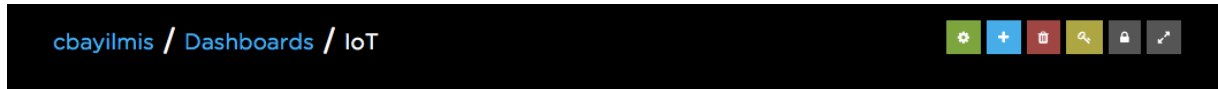
cbayilmis / Dashboards

Actions ▾


<input type="checkbox"/> Name ▾	Key ▾
<input type="checkbox"/> IoT	iot
<input type="checkbox"/> Welcome Dashboard	welcome-dashboard

Şekil 4. Oluşturulan dashboard lar

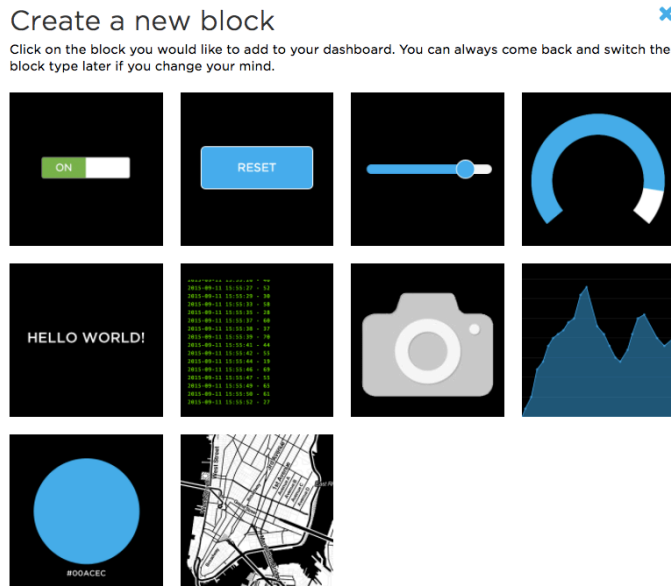
IoT dashboard sekmesini tıklayarak panelimizi uygulamamıza göre özelleştirebiliriz. (<https://io.adafruit.com/kullaniciadi/dashboards/iot>)



Şekil 4. Oluşturulan IoT dashboard

 **Kilit** sekmesi, panelin görünürlük (**visibility**) değerini göstermektedir. Kapalı kilit bu değer **private** (sadece kullanıcı tarafından erişilebilir) olduğunu gösterir. İlgili butona tıklayıp **public** (herkes tarafından erişilebilir) olarak ayarlanabilir.

 **artı** sekmesi ile ise dashboard ta görülmesini istediğimiz buton, grafik, slider, harita vb. bloklar eklenebilir.



Şekil 5. Oluşturulabilecek Blok Tipleri

Grafik (chart) blok ekleme işlemleri Şekil 6’da görülmektedir.

Block settings

In this final step, you can give your block a title and see a preview of how it will look. Customize the look and feel of your block with the remaining settings. When you are ready, click the "Create Block" button to send it to your dashboard.

Block Title  
sicaklik

Hours of History (0 for realtime)  
0

X-Axis Label  
Zaman

Y-Axis Label  
Sicaklik

Y-Axis Minimum  
0

Y-Axis Maximum  
50

Block Preview

sicaklik

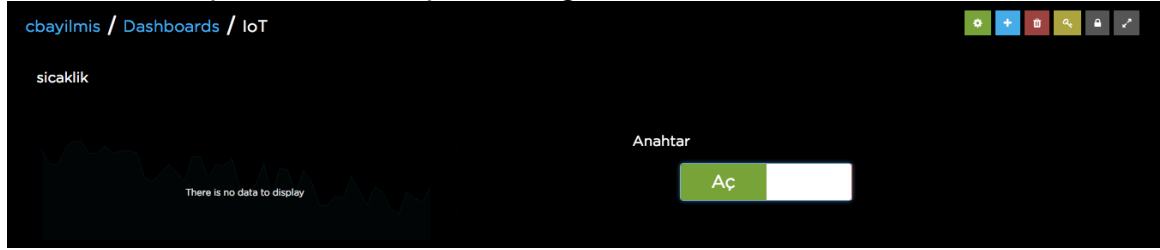
40  
20  
0

Jan 03 Jan 05 Jan 07 Jan 09 Jan 11  
-- Sample

< Previous step Create block

Şekil 6. Grafik blok ekleme işlemi

Grafik ve buton eklenmiş IoT dashboard Şekil 7’de görülmektedir.



Şekil 7. IoT dashboard eklenen bloklar

 Edit sekmesi ile panele yerleştirilmiş, bloklar düzenlenebilir.

Adafruit kullanıcı sayfasımızdan Feeds sekmesinden bloklarımıza ait feed (besleme) isimlerini görebiliriz. Feed isimlerini blokları oluştururken veriliyordu. Feed isimleri Arduino kod kısmında IoT panelimize veri göndermek ya da veri almak için kullanılacaktır.


Profile  
Feeds  
Groups  
Dashboards  
Triggers  
Settings  
Guide and Tips  
Adafruit IO Forum  
API Documentation  
Blog/Changelog

cbayilmis / Feeds

Actions

Name	Key	Last Value	Recorded
Welcome Feed	welcome-feed	62	2 months ago
sicaklik	sicaklik	No Data Available	2 months ago
buton	buton	Kapa	an hour ago

Şekil 8. Feed işlemleri

 Anahtar sekmesi ile ise IoT panelimize erişmek üzere bize özgü AIO Anahtara erişilir.

## YOUR AIO KEY

Your Adafruit IO key should be kept in a safe place and treated with the same care as your Adafruit username and password. People who have access to your AIO key can view all of your data, create new feeds for your account, and manipulate your active feeds.

If you need to regenerate a new AIO key, all of your existing programs and scripts will need to be manually changed to the new key.

Active Key

REGENERATE AIO KEY



Şekil 9. AIO Anahtarı

### Uygulamanın Wemos D1 Mini Kodları

```

/*****
BSM 451 Nesnelerin İnterneti ve Uygulaması Dersi
Doç. Dr. Cüneyt BAYILMIŞ
Adafruit MQTT Uygulaması (Wemos D1 Mini)
*****/

/**** ESP8266 WiFi Kütüphane Dosyası ****/
#include <ESP8266WiFi.h>

/**** Adafruit Kütüphane Dosyaları ****/
#include <Adafruit_MQTT.h>
#include <Adafruit_MQTT_Client.h>

/**** Kablosuz Ağ Bilgileri ****/
#define WLAN_SSID "KablosuzAğAdi" // "Kablosuz Ağ Adı"
#define WLAN_PASSWORD "KablosuzAğSifresi" // "Kablosuz Ağ Şifresi"

/***** Adafruit.io Kurulumu *****/
#define AIO_SERVER "io.adafruit.com"
#define AIO_SERVERPORT 1883 // MQTT Portu
#define AIO_USERNAME "kullaniciadi" // Kullanıcı Adı
#define AIO_KEY "fla1f63970044bfd97a6bc789800903c" // adafruit türetilen KEY

/***** MQTT Sunucuya Bağlantı Ayarları *****/
WiFiClient client; // ESP8266WiFiClient sınıfından bağlantı nesnesi
// adafruit MQTT bağlantı kurulumu
Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);

// Publish ve Subscribe için Feed Ayarları
Adafruit_MQTT_Publish sıcaklik = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/sicaklik");
// Adafruit_MQTT_Subscribe buton = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/buton");

/**** Değişken ve Pin Tanımlamaları ****/
int sıcaklikSensor= A0;
float sıcaklikdegeri; //Analog değeri dönüştürecekimiz sıcaklık değeri
float olculendeger; //Ölçecekimiz analog değer

/**** MQTT Bağlantı Fonksiyonu ****/
void MQTT_connect() {
  int8_t ret;
  // Bağlantı kurulmuş ise dur
  if (mqtt.connected()) {
    return;
  }
  Serial.print("Connecting to MQTT... ");
  //uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // 0 dönerse bağlanmıştır
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000); // wait 5 seconds
    // retries--;
    // if (retries == 0) {
    //   // basically die and wait for WDT to reset me
    //   while (1);
    // }
  }
  Serial.println("MQTT Connected!");
}

```

```
/** ESP8266 WiFi Kurulum Fonksiyonu */  
void WiFi_Setup() {  
    delay(10);  
    Serial.println(); Serial.print(WLAN_SSID);  
    Serial.print("Kablosuz Agina Baglaniyor");  
    WiFi.begin(WLAN_SSID, WLAN_PASSWORD);  
    // WiFi durum kontrolü  
    while (WiFi.status() != WL_CONNECTED) {  
        delay(500);  
        Serial.print(".");  
    }  
    Serial.println(); Serial.print(WLAN_SSID);  
    Serial.println("Kablosuz Agina Baglandi");  
    Serial.println("IP adresi: ");  
    Serial.println(WiFi.localIP());  
}  
  
void setup() {  
    Serial.begin(115200); // Seri port baud rate ayarı  
    WiFi_Setup(); //Kablosuz ağ bağlantı fonksiyonu  
}  
  
void loop() {  
  
    MQTT_connect(); // MQTT bağlanma fonksiyonu  
  
    // LM35 sıcaklık değeri okuma  
    olculendeger = analogRead(sicaklikSensor); //A0'den değeri alacak  
    olculendeger = (olculendeger/1024)*5000;//değeri mV'a dönüştürecek  
    sicaklikdegeri = olculendeger /12,0; // mV'u sıcaklığa dönüştürecek  
    sicaklikdegeri=sicaklikdegeri-10;  
  
    // MQTT server sıcaklık değeri gönderiliyor  
    Serial.print(F("\n Sicaklik Degeri : "));  
    Serial.print(sicaklikdegeri);  
    Serial.print("...");  
    sicaklik.publish(sicaklikdegeri);  
}
```

**Soru:** Bu uygulamada kazandığınız bilgileri kullanarak aşağıda verilen problemi çözen bir uygulama geliştiriniz. IoT platformuna bir buton ekleyiniz, bu buton aracılığı ile Wemos cihazınıza bağlı bir LED'i yakınız, tekrar basıldığında LED'i söndürünüz (toggle). Wemos cihaz, buton bilgisine MQTT Subscriber olarak işlem yapacaktır.

İpuçları: **Setup** fonksiyonu içerisinde abone olunacak topic (anahtar) bilgisi eklenmelidir.

*mqtt.subscribe(&anahtar);*

**Loop** fonksiyonu içerisinde subscriber işlemi için abone nesnesi tanımlanmalıdır.

*Adafruit\_MQTT\_Subscribe \*abone;*

*while ( ( abone = mqtt.readSubscription(1000))) {*

*if (abone == &anahtar) {*

*Serial.print(F("Durum: ")); // anahtar verisi*

*Serial.println((char\*)anahtar.lastread); // Serial.println((char\*) anahtar.lastread);*

*}*

*}*

**Kaynaklar:**

1 – Öner ŞAHİN, İsmail Can KARAMAN, Oğuzhan TIRAŞ , "Akıllı Kiralama Sistemi",Tasarım Tezi, Danışman Cüneyt BAYILMIŞ 2017.