



SAKARYA ÜNİVERSİTESİ
Bilgisayar ve Bilişim Bilimleri Fakültesi
Bilgisayar Mühendisliği Bölümü

BSM 451

NESNELERİN İNTERNETİ VE UYGULAMALARI

(Internet of Things (IoT) and Applications)

IoT PLATFORMLARI

Doç. Dr. Cüneyt BAYILMIŞ

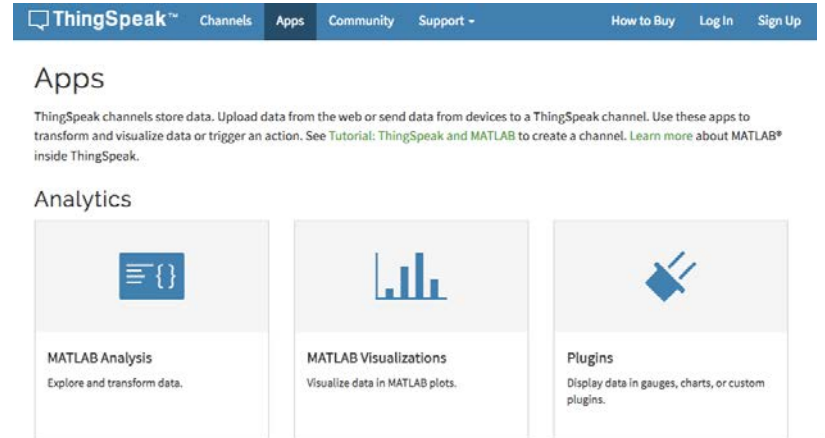


IoT Platformları

❑ IoT uygulamalarının doğası gereği nesneler tarafından üretilen/elde edilen verilerin internet ortamında depolanacağı ve görselleştirileceği/analiz edileceği ortamlara ihtiyaç vardır.

- ThingSpeak,
- adafruit,
- Firebase,
- TeMBoo,
- IBM Watson IoT,
- Microsoft Azure IoT,
- Amazon Web Services (AWS) IoT,
- ThingWorx IoT Platform,
- Carriots,

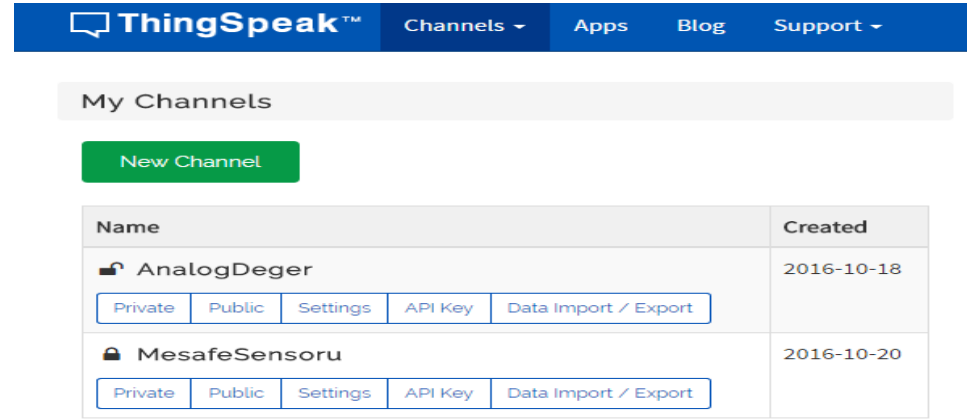
- ❑ <https://thingspeak.com>
- ❑ ThingSpeak IoT platformunu öne çıkaran özelliği veri analiz ve görselleştirme işlemlerini MATLAB ile gerçekleştirmesidir.
- ❑ Arduino, Raspberry Pi, BeagleBone, Particle Photon and Electron gibi gömülü sistemler ile birlikte çalışabilir.
- ❑ MQTT yayın desteği vardır.
- ❑ Olay programlama, uyarı/alarm oluşturma gibi özellikleri vardır.
- ❑ Twitter ile kullanılabilir.
- ❑ Ücretsiz olarak kullanılabilir.
- ❑ Açık API desteği vardır.



ThingSpeak Kullanımı

❑ <https://thingspeak.com> adresinden üye olduktan sonra veri gönderimi için kanal oluşturulur.

- Her kanaldan 8 adet veri bulunur.
- Kanal verisi JSON, XML



Name	Created
AnalogDeger	2016-10-18
MesafeSensoru	2016-10-20

❑ Gömülü sistemde veri gönderimi ve alımı için API Key elde edilir.

Write API Key

Key YUY42ACR5VYF9ZUB

Generate New Write API Key

Read API Keys

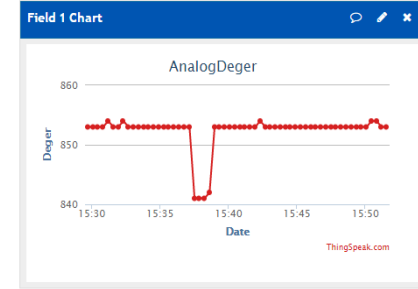
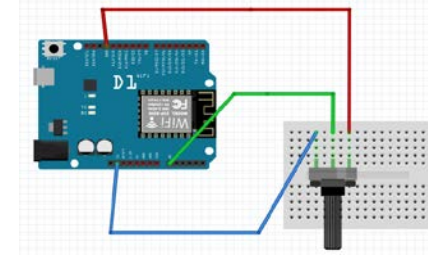
Key 4VUU0GYIJ188M8VN

Ardunio + ESP8266 WiFi (WEMOS) ile ThingSpeak Veri Gönderimi

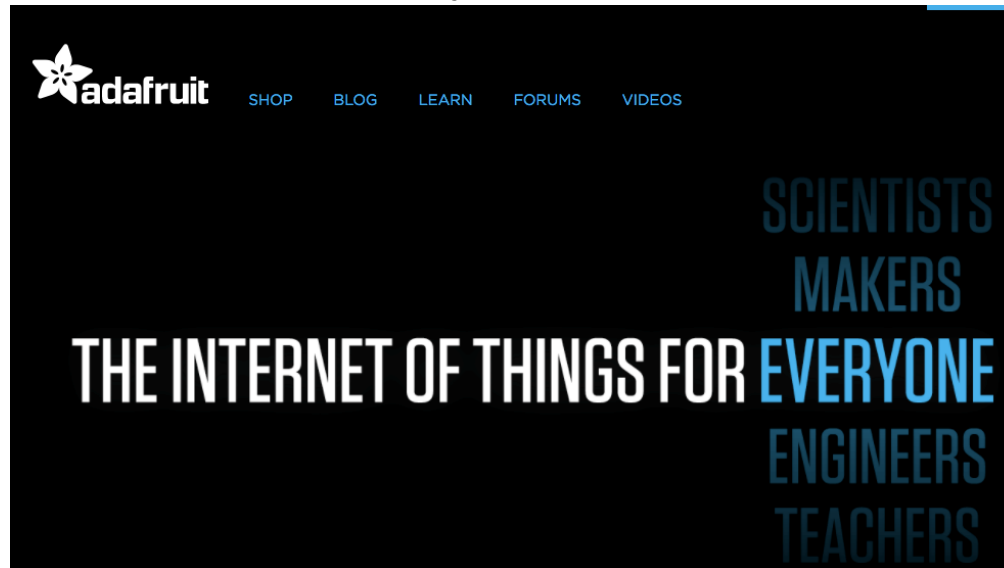
```
#include <ESP8266WiFi.h>
String apiKey = "YUY42ACR5VYF9ZUB";
const char* ssid = "TurkTelekom_IDDFA";
const char* password = "GdXm2avm";
const char* server = "api.thingspeak.com";
int deger=0;
WiFiClient client;
void setup()
{
  Serial.begin(115200);
  delay(10);
  WiFi.begin(ssid, password);

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

```
void loop()
{
  deger = analogRead(A0);
  Serial.println(deger);
  if (client.connect(server,80)) {
    String postStr = apiKey;
    postStr += "&field1=";
    postStr += String(deger);
    postStr += "\r\n\r\n";
    client.print("POST /update HTTP/1.1\n");
    client.print("Host: api.thingspeak.com\n");
    client.print("Connection: close\n");
    client.print("X-THINGSPEAKAPIKEY: "+apiKey+"\n");
    client.print("Content-Type: application/x-www-form-urlencoded\n");
    client.print("Content-Length: ");
    client.print(postStr.length());
    client.print("\n\n");
    client.print(postStr);
    Serial.print("deger: ");
    Serial.print(deger);
    Serial.println("Sending data to Thingspeak");
    client.print("\n\n");
  }
  client.stop();
  Serial.println("Waiting 20 secs");
  delay(20000);
}
```



- ❑ <https://io.adafruit.com>
- ❑ Kolay kullanım ve en az programlama ihtiyacı ile veri bağlantılarını sağlamayı amaçlamaktadır.
- ❑ REST ve MQTT API desteği vardır.
- ❑ Ücretsiz olarak kullanılabilmektedir.
- ❑ Kullanıcıya Dashboard oluşturmaya imkan verir.



Your Dashboards

NAME	DESCRIPTION	VISIBILITY	ACTIONS
Akilli Ev		Private	
Ev Otomasyonu		Public	
Welcome Dashboard	Your first dashboard.	Private	

CREATE A NEW DASHBOARD

io.adafruit.com/UyeAdi/DashboardAdi

<https://io.adafruit.com/ick1994/akilli-ev>

CREATE A NEW BLOCK

A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent.

A momentary button works similarly to a hardware push button.

The slider works well if you have a range of values you need to send.

A gauge is a read only block type that shows a fixed range of values.

A text block can be used to send data as well as view data.

A stream block can be used to view the rolling history of data for multiple feeds.

STEP 2: CHOOSE FEEDS

Add up to 1 feed

FEED/GROUP	LAST VALUE	RECORDED	ACTION
ick1994			
Alarm Sistemi	ON	5 days ago	CHOOSE
photocell	145	17 days ago	CHOOSE
alarmdurum	Ev Guvenli Durumda!	4 days ago	CHOOSE
alarmdurumu	ON	4 days ago	CHOOSE
redlight	0	17 days ago	CHOOSE
bluelight	0	17 days ago	CHOOSE
greenlight	0	17 days ago	CHOOSE
lambabuton	OFF	5 days ago	CHOOSE
sicaklikdurum	12.38	4 days ago	CHOOSE

[NEXT STEP >](#)

WEMOS ile MQTT Protokolü Kullanarak adafruit Veri Gönderimi

❑ Adafruit bağlantı kodları

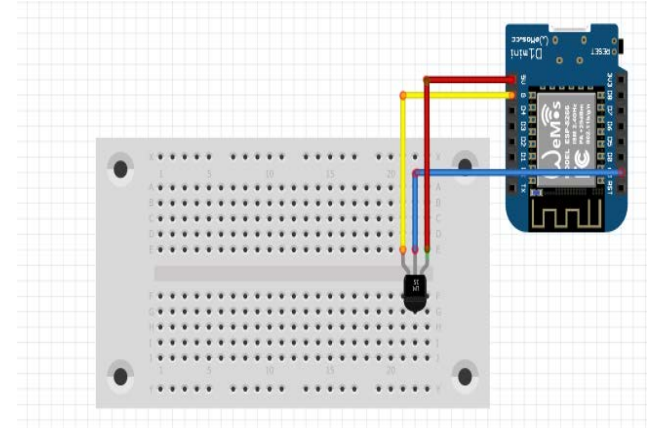
```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

/***** WiFi Access Point *****/

#define WLAN_SSID       "AndroidAP1"
#define WLAN_PASS       "sahin162"

/***** Adafruit.io Setup *****/

#define AIO_SERVER       "io.adafruit.com"
#define AIO_SERVERPORT  1883
#define AIO_USERNAME     "ick1994"
#define AIO_KEY          "4986d110c4cd4024ab8131e160ebc998"
```



❑ Adafruit MQTT fonksiyonları

```
// Create an ESP8266 WiFiClient class to connect to the MQTT server.
WiFiClient client;

Adafruit_MQTT_Client mqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
```

❑ Adafruit Feedleri

```
/***** Feeds *****/

// Notice MQTT paths for AIO follow the form: <username>/feeds/<feedname>
Adafruit_MQTT_Publish alarmdurum = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/alarmdurum");
Adafruit_MQTT_Publish sicaklikdurum = Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME "/feeds/sicaklikdurum");
// Setup a feed called 'onoff' for subscribing to changes.
Adafruit_MQTT_Subscribe onoffbutton = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/alarm-sistemi");
Adafruit_MQTT_Subscribe lambadurum = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/lambabuton");
Adafruit_MQTT_Subscribe kapidurum = Adafruit_MQTT_Subscribe(&mqtt, AIO_USERNAME "/feeds/alarmdurumu");
```


WEMOS ile MQTT Protokolü Kullanarak adafruit Veri Gönderimi

❑ MQTT Bağlantı Fonksiyonu

```
void MQTT_connect() {
    int8_t ret;

    // Stop if already connected.
    if (mqtt.connected()) {
        return;
    }

    Serial.print("Connecting to MQTT... ");

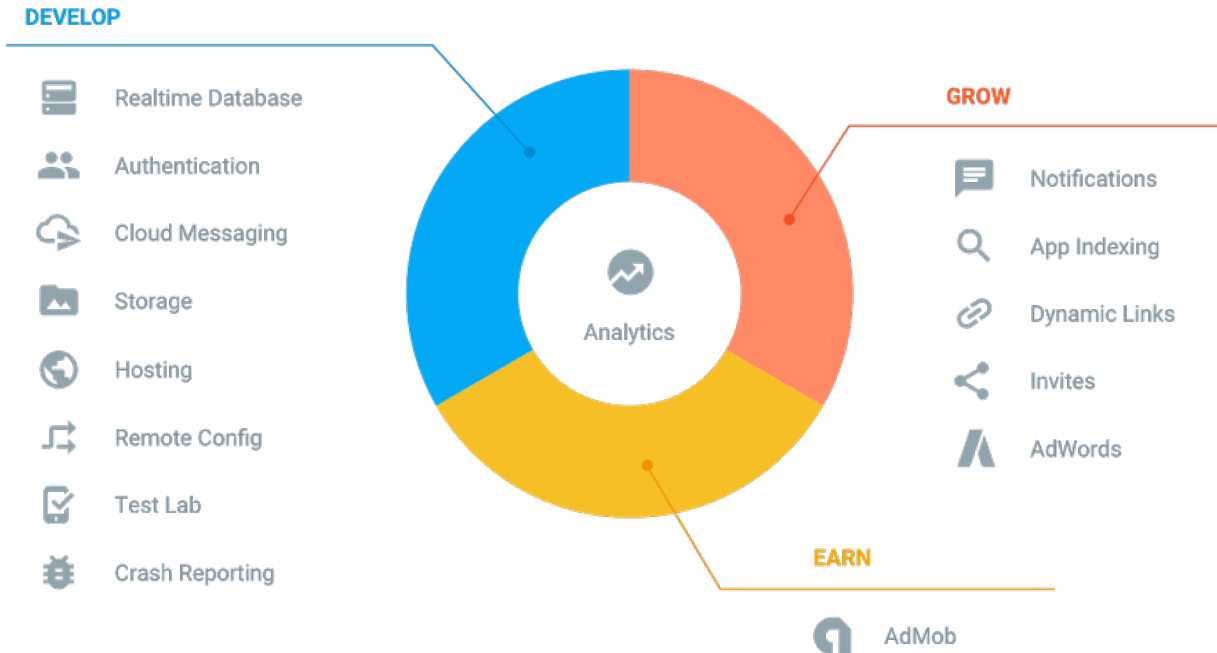
    uint8_t retries = 3;
    while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
        Serial.println(mqtt.connectErrorString(ret));
        Serial.println("Retrying MQTT connection in 5 seconds...");
        mqtt.disconnect();
        delay(5000); // wait 5 seconds
        retries--;
        if (retries == 0) {
            // basically die and wait for WDT to reset me
            while (1);
        }
    }
    Serial.println("MQTT Connected!");
}
```

```
void setup() {
    Serial.begin(115200);
    delay(10);
    // Connect to WiFi access point.
    Serial.println(); Serial.println();
    Serial.print("Connecting to ");
    Serial.println(WLAN_SSID);
    WiFi.begin(WLAN_SSID, WLAN_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println();
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

void loop() {
    MQTT_connect();
    olculendeger = analogRead(sicaklikSensor); //A0'den değeri alacak
    olculendeger = (olculendeger/1024)*5000;//değeri mV'a dönüştürecek
    sicaklik = olculendeger /12,0; // mV'u sıcaklığa dönüştürecek
    sicaklik=sicaklik-10;
    Serial.print(F("\nSending sıcaklık val "));
    Serial.print(sicaklik);
    Serial.print(" ...");
    sicaklikdurum.publish(sicaklik);
}
```



- ❑ <https://firebase.google.com/>
- ❑ Google'ın gerçek zamanlı veri depolama özelliği olan bulut tabanlı platformudur.
- ❑ Kullanım verilerinin analizi, bildirim gönderme, uygulama testi gibi işlemlerin yerine getirebilecek kontrol/yönetim paneli sunar.





- ❑ NoSQL veritabanı kullanarak kullanıcılarla cihazlar arasında verileri gerçek zamanlı olarak saklayan ve senkronize eder. JSON veri formatını kullanır.
- ❑ Güncellenmiş veri, bağlı cihazlar arasında milisaniyeler içinde senkronize edilir ve uygulamamız çevrimdışı durumdaysa veriler saklanır ve ağ bağlantısı olduğunda senkronize edilir.

The screenshot shows the Firebase console interface for a project named 'akiz-sau'. The 'Proje ayarları' (Project settings) tab is selected. The 'Proje kimliği' (Project ID) is highlighted with a red circle, showing 'akiz-sau.firebaseio.com'. The 'Web API anahtarı' (Web API key) is also highlighted with a red circle, showing 'AlzaSyCKB5O9astkA4PU-slDAv7588G0305DdkA'.



- ❑ Arduino temelli uygulamalar için <https://github.com/firebase/firebase-arduino/> adresindeki başlık dosyası kullanılabilir.
- ❑ Firebase Bağlantı kurmak için **HOST** adresi ve **AUTH** gizli anahtar eklenir.

```
String HOST = "ornek.firebaseio.com";  
String AUTH = "anahtarkodu";  
Firebase.begin(HOST, AUTH);
```

- ❑ Değer yazma “ornek.firebaseio.com/” adresinin altında sayı json verisi oluşturulur.

```
Firestore.setFloat ("sayı", 42.0);
```

- ❑ Değer okuma “ornek.firebaseio.com/sayı” adresinin altında json verisi okuyoruz.

```
sayı = Firestore.getFloat("sayı");
```

KAYNAKLAR

- A. Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, “*Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications*”, IEEE Communication Survey&Tutorials, vol. 17 (4), 2347-2376 ,2015.
- C. Tsai, C. Lai, M. Chiang, and L. T. Yang, “Data mining for Internet of Things: A survey,” IEEE Commun. Surveys Tuts., vol. 16, no. 1, pp. 77– 97, 1st Quart. 2014
- Öner ŞAHİN, İsmail Can KARAMAN, Oğuzhan TIRAŞ , "Akıllı Kiralama Sistemi", Bitirme Tezi, Sakarya Üniv., Bilgisayar Müh., Danışman Doç. Dr. Cüneyt BAYILMIŞ 2017.
- A. Ayman RADDAD, Ö. Faruk ŞAHİN, "Akıllı Priz", Bitirme Tezi, Sakarya Üniv., Bilgisayar Müh., Danışman Doç. Dr. Cüneyt BAYILMIŞ 2017.
- <https://thingspeak.com>
- <https://io.adafruit.com>
- <https://firebase.google.com/>
- <https://github.com/firebase/firebase-arduino/>