



SAKARYA ÜNİVERSİTESİ
Bilgisayar ve Bilişim Bilimleri Fakültesi
Bilgisayar Mühendisliği Bölümü

BSM 451
INTERNET OF THINGS AND APPLICATIONS

IoT Platforms

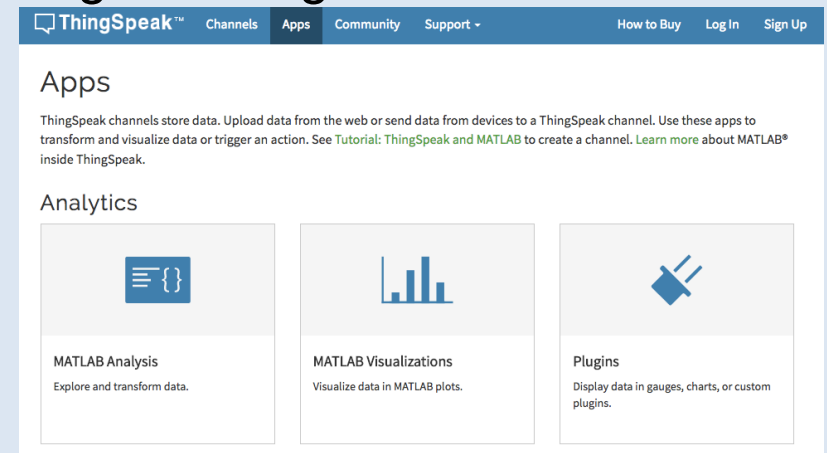
Assoc. Prof. Cüneyt BAYILMIŞ
Researcher Dr. Ünal ÇAVUŞOĞLU

IoT Platforms

❑ The nature of IOT applications requires environments where the data generated / obtained by the objects will be stored and visualized / analyzed on the internet.

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

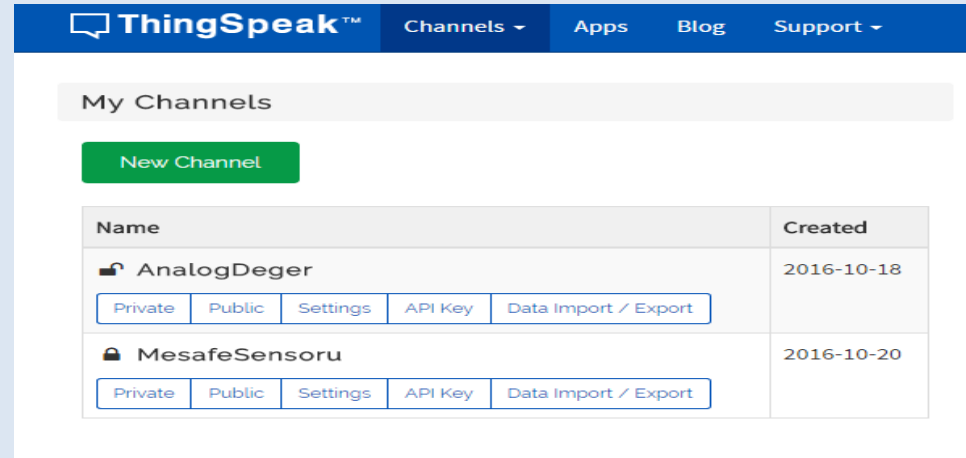
- ❑ <https://thingspeak.com>
- ❑ The ThingSpeak IoT platform is a feature that makes data analysis and visualization work with MATLAB.
- ❑ It can work with embedded systems such as Arduino, Raspberry Pi, BeagleBone, Particle Photon and Electron.
- ❑ MQTT has broadcast support.
- ❑ Olay programlama, uyarı/alarm oluşturma gibi özellikleri vardır. It has features such as event scheduling, warning / alarm generation.
- ❑ It can be used with Twitter.
- ❑ It can be used for free.
- ❑ There is open API support.



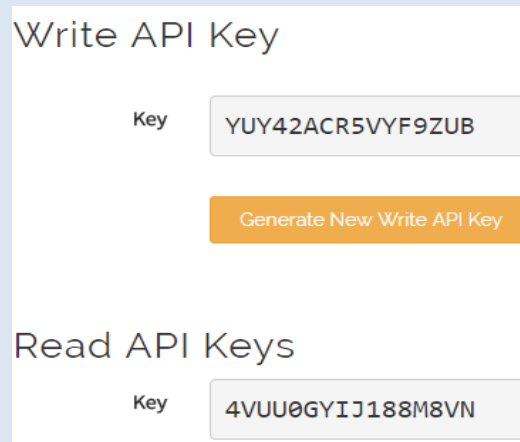
ThingSpeak Kullanımı

❑ After joining from <https://thingspeak.com> address, channel is created for sending data.

- There are 8 data from each channel.
- Channel data JSON, XML



❑ The API Key is obtained for sending and receiving data in the embedded system.

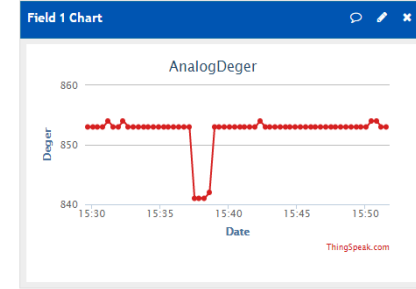
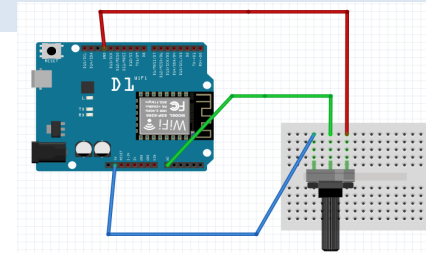


Ardunio + ESP8266 WiFi (WEMOS) with ThingSpeak send data

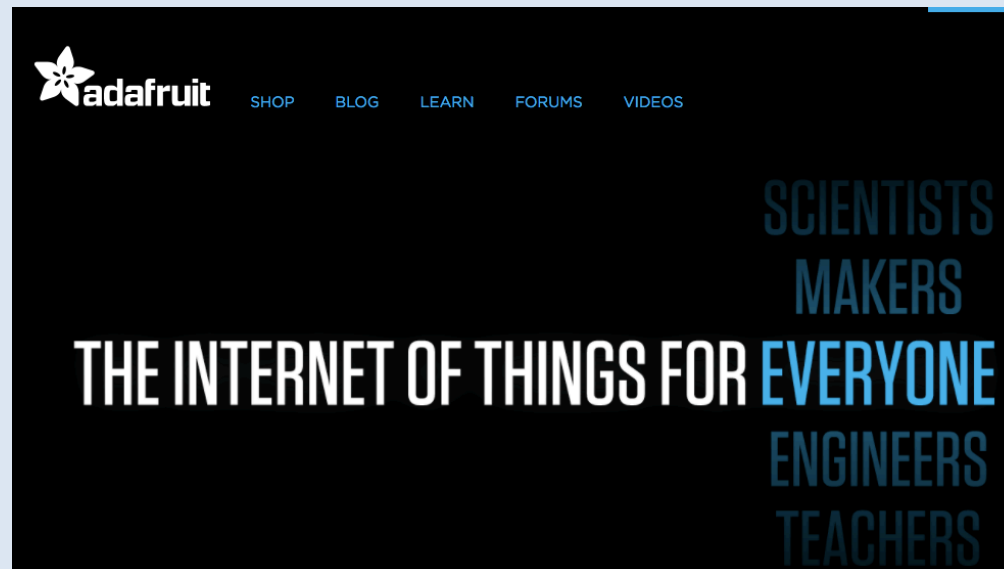
```
#include <ESP8266WiFi.h>
String apiKey = "YUY42ACR5VYF9ZUB";
const char* ssid = "TurkTelekom_TDDFA";
const char* password = "GdXm2avm";
const char* server = "api.thingspeak.com";
int deger=0;
WiFiClient client;
void setup()
{
  Serial.begin(115200);
  delay(10);
  WiFi.begin(ssid, password);

  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>
- ❑ It aims to provide data links with ease of use and minimal programming requirements.
- ❑ REST and MQTT API support.
- ❑ It can be used for free.
- ❑ Allows user to create Dashboard.



Your Dashboards

CREATE DASHBOARD

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

CREATE A NEW DASHBOARD

DASHBOARD NAME

CANCEL
CREATE DASHBOARD

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.

CREATE


A momentary button works similarly to a hardware push button.

CREATE


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

CREATE


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

CREATE


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

CREATE


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

CREATE

io.adafruit.com/UyeAdi/DashboardAdi

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

STEP 2: CHOOSE FEEDS

Add up to 1 feed

CREATE

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 >

Using WEMOS ile MQTT Protocols send data with adafruit

Adafruit connection codes

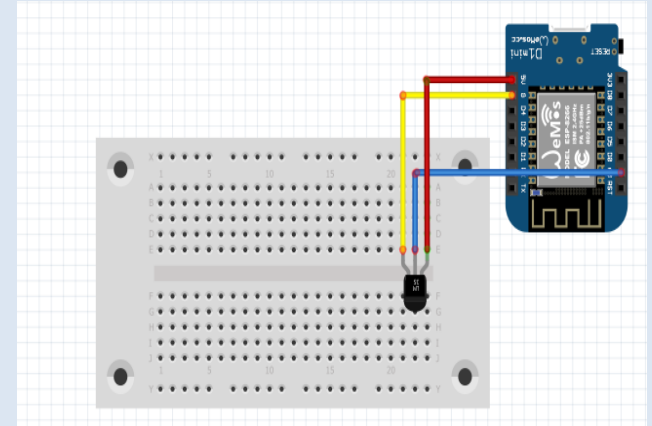
```
#include <ESP8266WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"

/***** 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 functions

```
// 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 Feeds

```
/***** 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");
```


Using MQTT Protocols send data with adafruit

MQTT connection functions

```
void MQTT_connect() {
  int8_t ret;

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

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

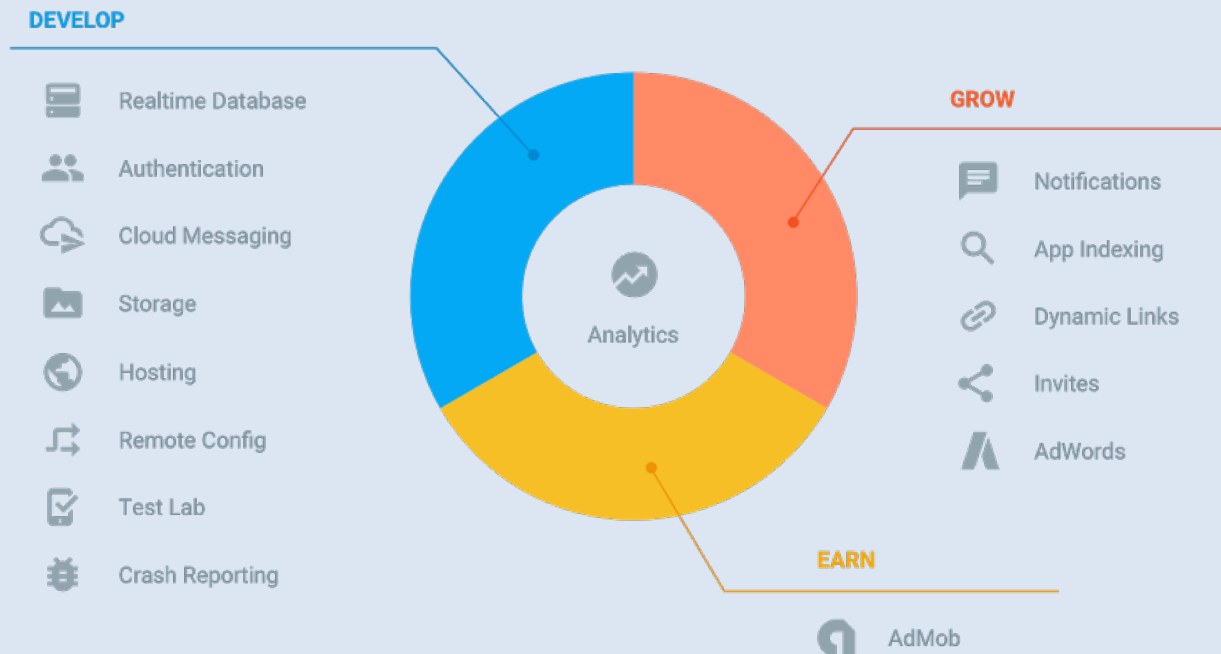
  uint8_t retries = 3;
  while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
    Serial.println(mqtt.connectErrorString(ret));
    Serial.println("Retrying MQTT connection in 5 seconds...");
    mqtt.disconnect();
    delay(5000); // wait 5 seconds
    retries--;
    if (retries == 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 sicaklığa dönüştürecek
  sicaklik=sicaklik-10;
  Serial.print(F("\nSending sicaklık val "));
  Serial.print(sicaklik);
  Serial.print("...");
  sicaklikdurum.publish(sicaklik);
}
```

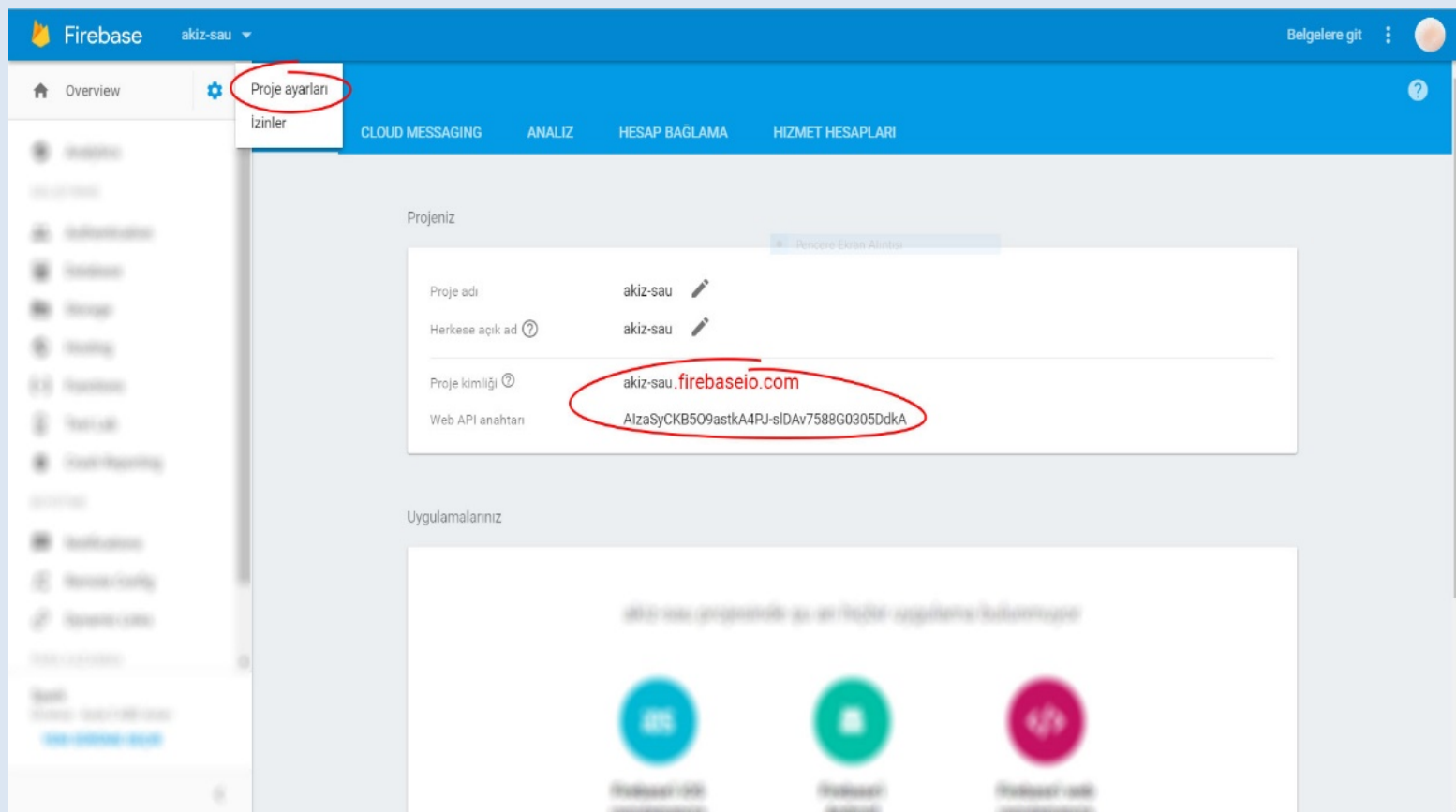


- ❑ <https://firebase.google.com/>
- ❑ It's a cloud-based platform that's Google's real-time data storage.
- ❑ A control / management panel that can perform operations such as analysis of usage data, sending notifications, application tests.





- ❑ It uses NoSQL database to store and synchronize data in real time between devices with users. It uses the JSON data format.
- ❑ The updated data is synchronized between connected devices in milliseconds, and when my application is offline, the data is stored and synchronized when there is network connectivity.





- ❑ For Arduino based applications, the header file at <https://github.com/firebase/firebase-arduino/> address can be used.
- ❑ Firebase HOST address and AUTH secret key are added to establish a connection.

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

- ❑ Write the value "json.firebaseio.com/" under the address json data is generated.

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

- ❑ Reading the value We read json data under "example.firebaseio.com/page".

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

References

- 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/>