

LM35, 3 bacaklı analog bir sıcaklık sensörüdür. -55 ile +150 derece arasındaki sıcaklıkları ölçebilir. 1° C'lik sıcaklık artışında çıkışı 10 mv artar. Giriş gerilimi olarak 4v-20v arası çalışabilir.

NodeMCU

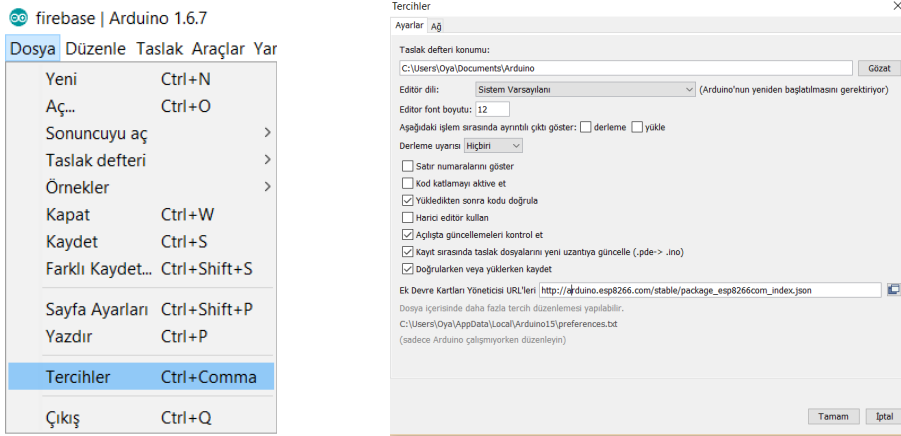
NodeMCU, ESP8266 WiFi modülüne sahip bir Arduino temelli uygulama kartıdır. Bu kart uygulamadaki dimmer işlemi için PWM uçlarına sahiptir. Uygulamada kullanılan D2 ucu aynı zamanda pwm ucudur. Darbe Genişlik Modülasyonu (PWM) ile bir kare dalga'nın lojik 1 olma süresi (genişliği) ayarlanarak LED'in parlaklık seviyesi ayarlanmaktadır.

Arduino IDE

Arduino geliştirme ortamı (IDE), Arduino bootloader (Optiboot), Arduino kütüphaneleri, AVRdude (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.

ESP8266 WiFi modülüne sahip Wemos D1, NodeMCU gibi kartları Arduino IDE'nizde tanımlı kartlar arasına ekleyebilmek için Şekil 2'de görüldüğü gibi **Dosya > Tercihler** sekmesindeki ekranda **"Ek Devre Kartları Yöneticisi URL'leri"** kutusuna aşağıda verilen linki ekleyiniz.

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

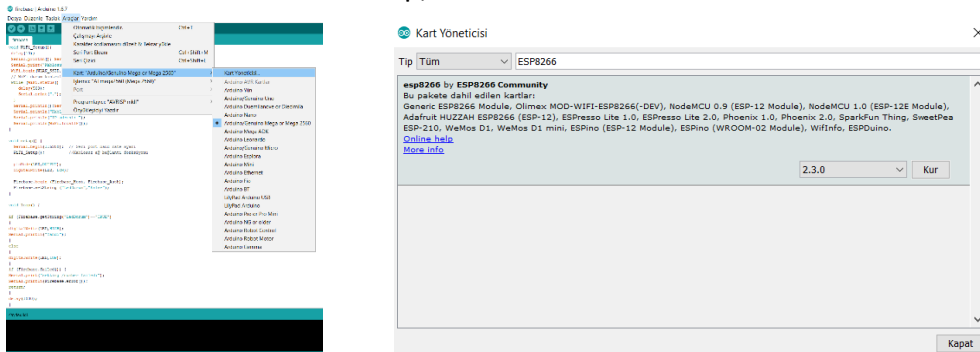


Şekil 2. ESP8266 modülünü Arduino IDE ortamında tanımlamak

Esp8266

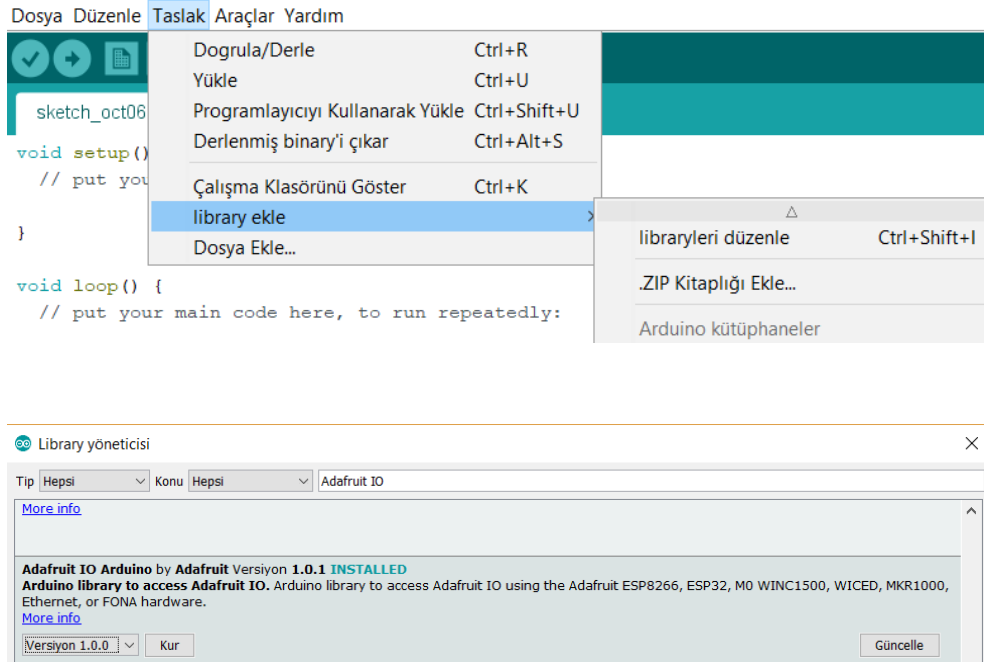
Kolayca wireless ağlara bağlanmayı sağlayan modül. esp8266-01'den başlayıp esp8266-12'ye kadar giden versiyonları bulunuyor. Kendi firmware'inizi yazıp yükleyerek başka hiçbir şeye ihtiyaç duymadan uygulama geliştirebiliyoruz. AT+ ile başlayan komutları göndererek bağlanılabilir wi-fi ağlarının listelenmesi,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

Şekil 3'de görüldüğü gibi Wemos D1 mini kartında ESP8266 kütüphanelerini eklemek için Arduino IDE'de **Araçlar > Kart > Kart Yöneticisi** ekranından ESP8266 aratıp, kurunuz.



Şekil 3. ESP8266 modülüne sahip kartları ekleme işlemi

Ardunio yazılımı ile adafruit IoT platformunu kullanabilmek için **Adafruit IO Arduino** (1.0.1 versiyonu) kütüphane dosyasını **Taslak > library ekle > libraryleri düzenle** ekranından yüklenmelidir. Kütüphane dosyası eklendiğinde **Dosya > Örnekler** sekmesinde **Adafruit IO Arduino>ESP8266** örnekleri yüklenmiş olacaktır.



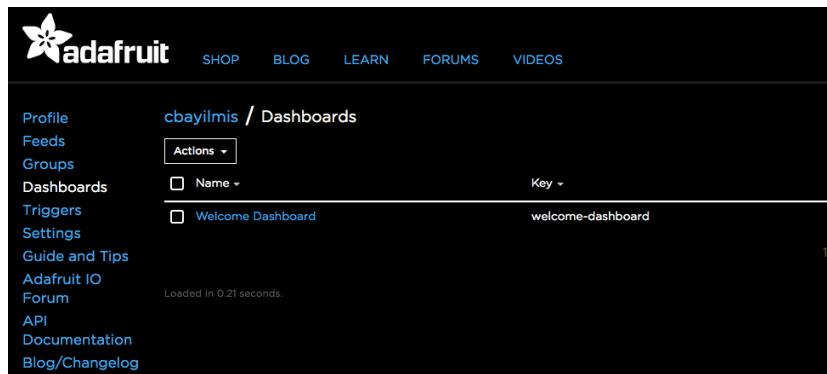
Şekil 4. Adafruit IO Arduino kütüphanelerinin eklenmesi

adafruit.io (Dashboard) IoT Bulut Platformu

Uygulamanın web üzerinden kontrolü ve kolay yönetilebilmesi için IoT platformu olarak [adafruit](https://adafruit.io) kullanacağız. [adafruit](https://adafruit.io) 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 5’de görüldüğü üzere <https://io.adafruit.com/kullaniciadi/dashboards> adresindeki arayüz aracılığıyla IoT uygulamanıza yönelik paneli (*dashboard*) oluşturabilirsiniz.



Şekil 5. 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 6. Yeni bir dashboard oluşturma

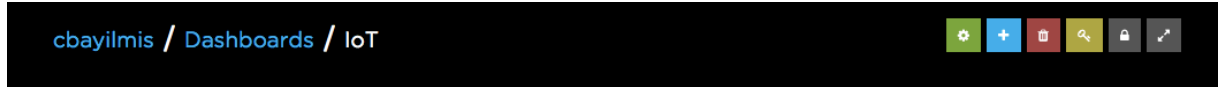
cbayilmis / Dashboards

Actions ▾


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

Şekil 7. Oluşturulan dashboard lar

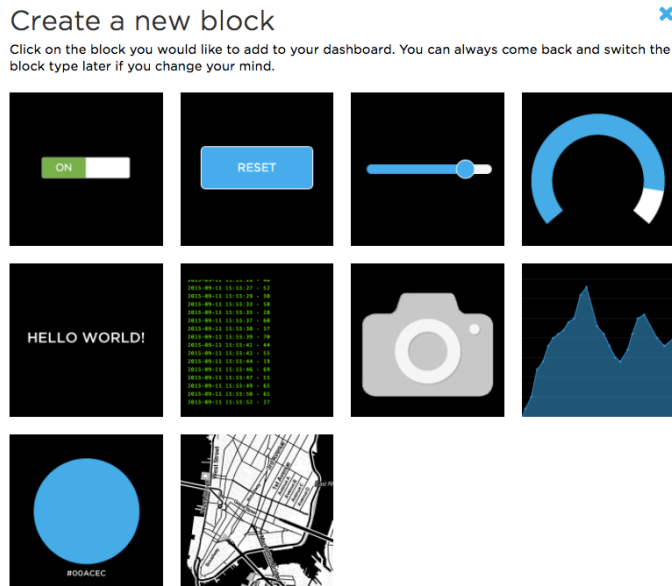
IoT dashboard sekmesini tıklayarak panelimizi uygulamamıza göre özelleştirebiliriz. (<https://io.adafruit.com/kullaniciadi/dashboards/iot>)



Şekil 8. 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 9. Oluşturulabilecek Blok Tipleri

Şekil 10'da uygulamada kullanılacak buton ve slider ekleme işlemleri görülmektedir. Önce oluşturulmak istenen Feed ismi girilir ve **Create** butonuna basılarak oluşturulur.

Choose feed

Toggle: A toggle button is useful if you have an ON or OFF type of state. You can configure what values are sent on press and release.

If you have lot of feeds, you may want to use the search field. You can also create a feed quickly below.

| Group / Feed | Last value | Recorded |
|-------------------------------|------------|-------------|
| <input type="checkbox"/> RFID | 1 | 25 days ago |

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 (optional)

Button On Text

Button Off Text

Block Preview



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

Test Value

Published Value

0 bytes

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 (optional)

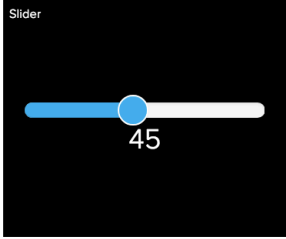
Slider Min Value

Slider Max Value

Slider Step Size

Slider Label

Block Preview



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

Test Value

Published Value

0 bytes

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 (optional)

Show History

Live (no history)

X-Axis Label

Y-Axis Label

Y-Axis Minimum

Y-Axis Maximum

Decimal Places

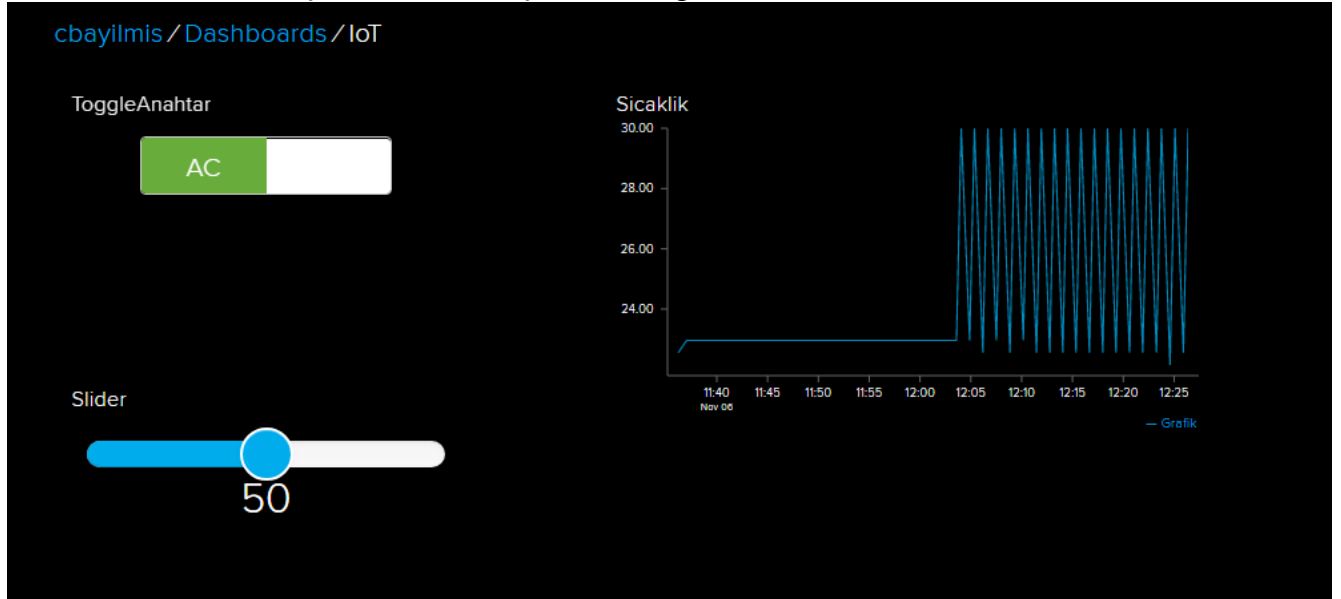
Block Preview



Line Chart The line chart is used to graph one or more feeds.

Şekil 10. Anahtar, Slider ve Grafik blok ekleme işlemleri

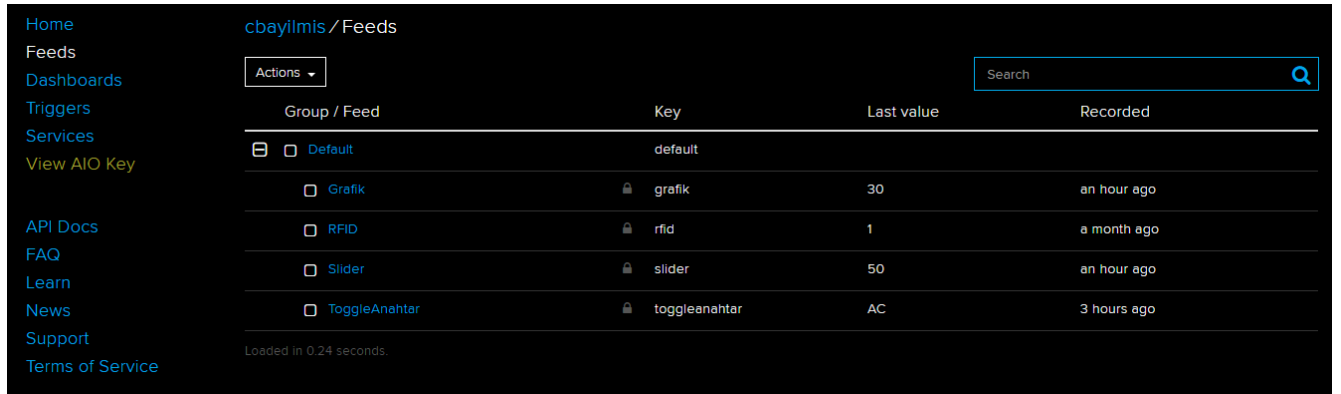
Buton, Slider ve Grafik eklenmiş IoT dashboard Şekil 11’de görülmektedir.



Şekil 11. 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.



| Group / Feed | Key | Last value | Recorded |
|---------------|---------------|------------|-------------|
| Default | default | | |
| Grafik | grafik | 30 | an hour ago |
| RFID | rfid | 1 | a month ago |
| Slider | slider | 50 | an hour ago |
| ToggleAnahtar | toggleanahtar | AC | 3 hours ago |

Şekil 12. 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 13. AIO Anahtarı

Nodemcu Arduino Kodları

```
/* *****  
BSM451 Nesnelerin İnterneti ve Uygulamaları Dersi  
Doç. Dr. Cüneyt BAYILMIŞ  
Adafruit Genel Uygulama  
***** */  
/* Kütüphane Dosyaları */  
#include <ESP8266WiFi.h>  
#include "Adafruit_IO_Client.h"  
/* Kablosuz Bağlantı Bilgileri */  
#define WLAN_SSID "KablosuzAgAdi"  
#define WLAN_PASSWORD "KablosuzAgSifresi"  
/* Adafruit Feed Erişimi İçin Key Bilgisi */  
#define AIO_KEY "adafruit key bilgisini buraya kopyalayınız"  
  
WiFiClient client;  
Adafruit_IO_Client aio = Adafruit_IO_Client (client, AIO_KEY);  
  
/* Adafruit dashboard eklenen nesne (Feed Adı RFID) */  
Adafruit_IO_Feed butonFeed = aio.getFeed("ToggleAnahtar");  
Adafruit_IO_Feed sliderFeed = aio.getFeed("Slider");  
Adafruit_IO_Feed grafikFeed = aio.getFeed("Grafik");  
/* Pin Tanımları */  
#define Led D2  
int sıcaklikSensor=A0; // LM35 Data ucu A0 pinine bağlanacak  
float sıcaklikDegeri; // Analog değeri dönüştürecek sıcaklık değeri  
float olculenDeger; // Ölçeğimiz analog değer  
  
/* ESP8266 WiFi Kurulum Fonksiyonu */  
void wifiSetup () {  
  delay (10);  
  Serial.println(); Serial.println(WLAN_SSID);  
  Serial.println(F("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 Aga Baglandi");  
  Serial.println("IP adresi: "); Serial.println(WiFi.localIP());  
}  
  
void setup() {  
  
  Serial.begin(9600); // Serihaberleşme 9600 baud hızında başlatılıyor  
  wifiSetup(); // Kablosuz ağ kurulum fonksiyonu  
  aio.begin(); // Adafruit IO client sınıfı başlatılıyor  
  
  pinMode (Led, OUTPUT); // D2 ucu Led sürmek için çıkış modunda  
  pinMode (A0, INPUT); // A0 ucu sensör okumak için giriş modunda  
  
  Serial.println(F("Kurulum Hazır"));
```

```
// digitalWrite(Led, HIGH);
analogWrite(Led, 0);
}
/* Ana (çalışan) fonksiyon */
void loop() {
    delay(500);
    butonKontrol();    // Toggle buton fonksiyonu
    // sliderKontrol();    // Slider fonksiyonu
    sicaklik();        // Sicaklik grafik fonksiyonu
}
void butonKontrol(){
    FeedData okuButonFeed=butonFeed.receive(); // io.adafruit feed değeri al
    Serial.println(okuButonFeed);

    if (strcmp(okuButonFeed,"AC")==0)
        digitalWrite(Led, HIGH);
    else
        digitalWrite (Led, LOW);
}

/* void sliderKontrol(){
    FeedData okuSliderFeed=sliderFeed.receive(); // io.adafruit Slider feed değeri al
    Serial.println(okuSliderFeed);
    int deger;
    char *oku;
    oku=(char *) okuSliderFeed; // atoi fonksiyonu için tip dönüşümü
    deger=atoi(oku);

    deger=deger * 10;
    if (deger >1023)
        deger = 1023;
    Serial.print("Duty Cycle: "); Serial.println(deger);
    analogWrite(Led, deger);
}*/

void sicaklik (){
    /* LM35 sensöründen sıcaklık değeri okuma işlemi */
    olculenDeger = analogRead (sicaklikSensor); // A0 analog ucundan değer oku
    olculenDeger = (olculenDeger/1024)*5000; // mv'a dönüşüm işlemi
    sicaklikDegeri = olculenDeger /12,0; // mV'u sıcaklığa dönüştür
    sicaklikDegeri = sicaklikDegeri-10;

    Serial.print("Sicaklik Değeri: "); Serial.println(sicaklikDegeri);

    grafikFeed.send(sicaklikDegeri); // grafik feed e sıcaklık değerini gönder
}
```