

# Arduino Nano ve ESP8266 ile Wifi üzerinden Sıcaklık – Kontrol Web Sunucu

## İçindekiler

Abstract .....	1
Motivation .....	1
Neden? .....	1
Parçalar .....	2
Başlarken .....	2
Yanlış Giden bazı işler.....	3
1.Aşama – ESP 8266 –Arduino Bağlantısı.....	4
2.Aşama ESP8266'nın Arduino üzerinden Test Edilmesi.....	6
3.Aşama DHT11'in Bağlanması .....	7
4.Aşama – DHT11'den sıcaklık ve nem okuması .....	8
5.Aşama – Web Sunucusunun Çalıştırılması .....	10
6.Aşama – Arduino Nano üzerindeki LED'in açılıp kapanması.....	13
Sonuç .....	14

Kaynak kod/Source code:

<http://www.barissanli.com/electronics/arduino-esp-webserver-barissanli.zip>

## Abstract

In this tutorial, a very simple wireless webserver that can serve a single page with ambient temperature and humidity through DHT11 is described. Also a button is added on the page to turn on and off LED on Arduino Nano. ESP8266 has not been modified, the commands are low level. The tutorial is in Turkish, when ready English version can be reached from here <link not ready>:

## Motivation

When I go out to buy a temperature and humidity monitor, I realized that I can make a wireless one with the same price, so I tried it.

## Neden?

Eve bir sıcaklık ve nem göstergesi almak için gezdiğimde, fiyatların yüksekliği kafamı karıştırdı. Evdeki bazı parçalardan, evdeki her noktadan sıcaklık ve nemi görebileceğim, bir de yanında bazı aparatları kontrol edebileceğim bir sistem tasarlama ihtiyacı duydum. Artık cep telefonumdan her daim ev sıcaklığı ve nemini kontrol ediyor, ışık açıp kapayabiliyorum.

## Parçalar

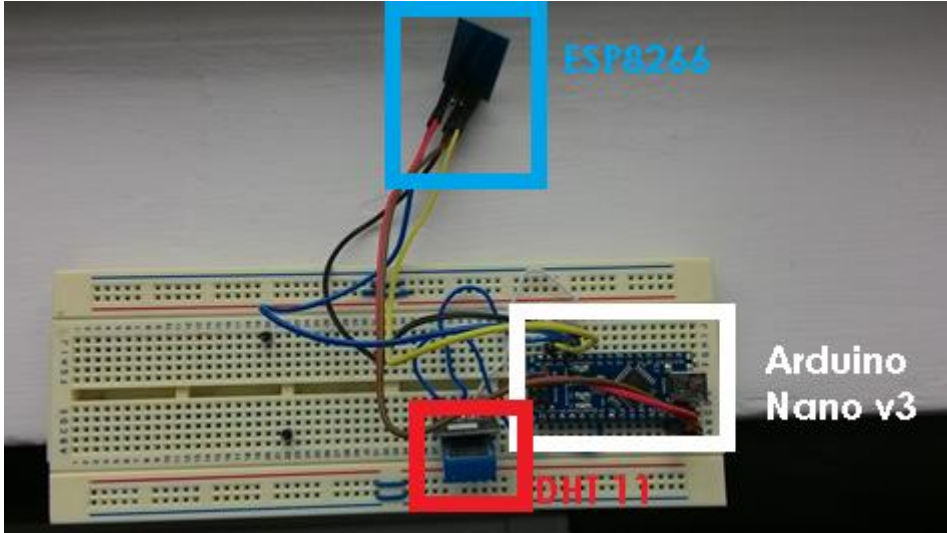
- Arduino Nano v3 ... 7\$ (30 TL-21 TL 14 Haziran 2015 – [www.gittigidiyor.com](http://www.gittigidiyor.com))
- DHT11 sıcaklık ve nem sensörü (11 TL-5 TL [www.gittigidiyor.com](http://www.gittigidiyor.com))
- Breadboard (Mini de olabilir, 10 TL-5 TL)
- 8 tane kablo (ben lehim vs kullanmadım, 5 tanesi dişi-dişi olsa iyi olur, 3.54 TL)

Toplam: 55 TL -35 TL , kargo ücretleri dahil değil...

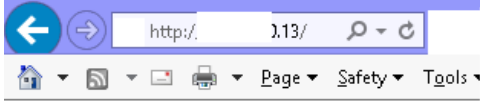
## Başlarken

Başlarken, böyle bir proje aklımda yoktu, sadece Arduino ile en ucuz WiFi opsiyonunu arıyordum.Yıllarca önce PIC kullanmış biri olarak, gerek Raspberry Pi, gerek Arduino'nun neler yapabildiğini ve ne kadar düşük fiyata yapabildiğini görmek oldukça ilginç...

Sonunda elde edeceğimiz ise Resim 1'deki devre ve Resim 2'deki ekran görüntüsü



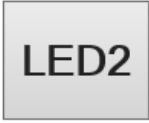
Resim 1- Arduino Webserver (sunucu Arduino'da çalışıyor)



## Arduino!

**Nem: 34.00%**

**Sicaklik 26.00 C**



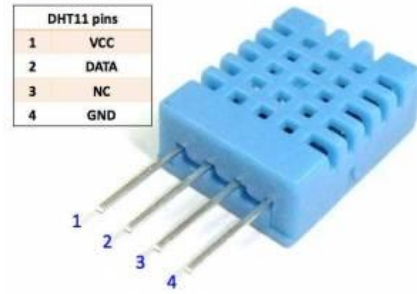
Resim 2- Arduino sunucumuza eriştiğimizde gelen web sayfası

### Yanlış Giden bazı işler....

Resim 3'de gördüğünüz benim Robotistan aldığım shield'li (zarflı) DHT 11 sensörü... İnternette bakarsanız DHT 11 sensörü altındaki devre yok (Resim 4). Bu devrenin amacı aslında bir 10k direnç ekleyerek sensörün arduino'ya direkt bağlanmasını sağlamak....



Resim 3 – DHT 11 sensörü – Robotistan .com



Resim 4- DHT 11 sensörü – Gittigidiyor.com

Diğer yanlış giden husus ise ESP8266'ı çalıştırmanın biraz zor olmasıydı. Baud Rate 9600 üzerinde başarı sağlayamadım. Bir de USB-TTL converter kullanmayıp işi basit tuttuğum için biraz uğraştırdı. Ayrıca Arduino nano biraz eski olduğundan geçmişte pinlerini yakmış olabileceğimden farklı pinler kullandım. İnternette İngilizce sitelerde ESP'nın RX pininin Arduino'nun dijital RX pinine bağlanacağı söylense de, ben RX ile TX yani çapraz bağlantının başarılı olduğunu gördüm.

Sıralama

1. ESP 8266'nın Arduino'ya bağlanması
2. ESP 8266'nın Test edilmesi
3. Sonra DHT 11'in bağlanması
4. DHT 11 ile sıcaklık okuma
5. Web sunucunun çalıştırılması
6. LED aç/kapa

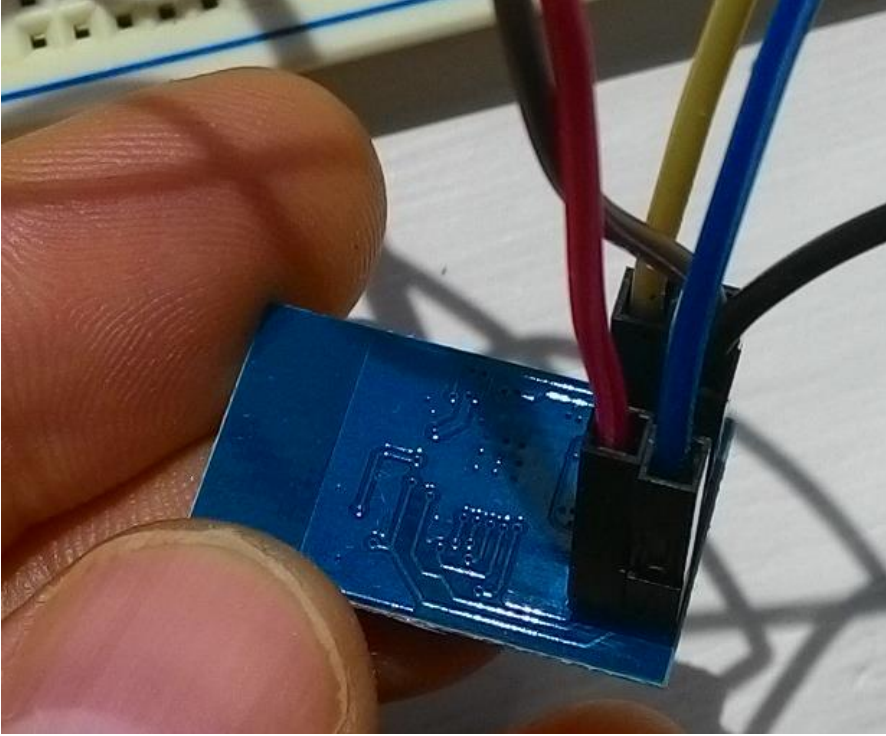
İnternette kullandığım siteler ve kodlar:

<http://www.hobbyist.co.nz/?q=documentations/wiring-up-dht11-temp-humidity-sensor-to-your-arduino>

<http://allaboutee.com/2014/12/30/esp8266-and-arduino-webserver/>

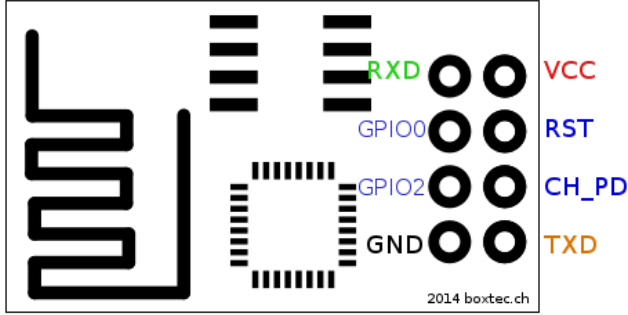
### 1.Aşama – ESP 8266 –Arduino Bağlantısı

ESP8266'da arkadaki 5 pini kullanacağız. Bu pinlerin renkli kablolar ile bağlantısı resim 5'te verilmiştir. Normalde, kırmızı VCC siyah GND için kullanmama rağmen aşağıda renk kodlarına dikkat edilmeden verilmiş resim bulunmaktadır.



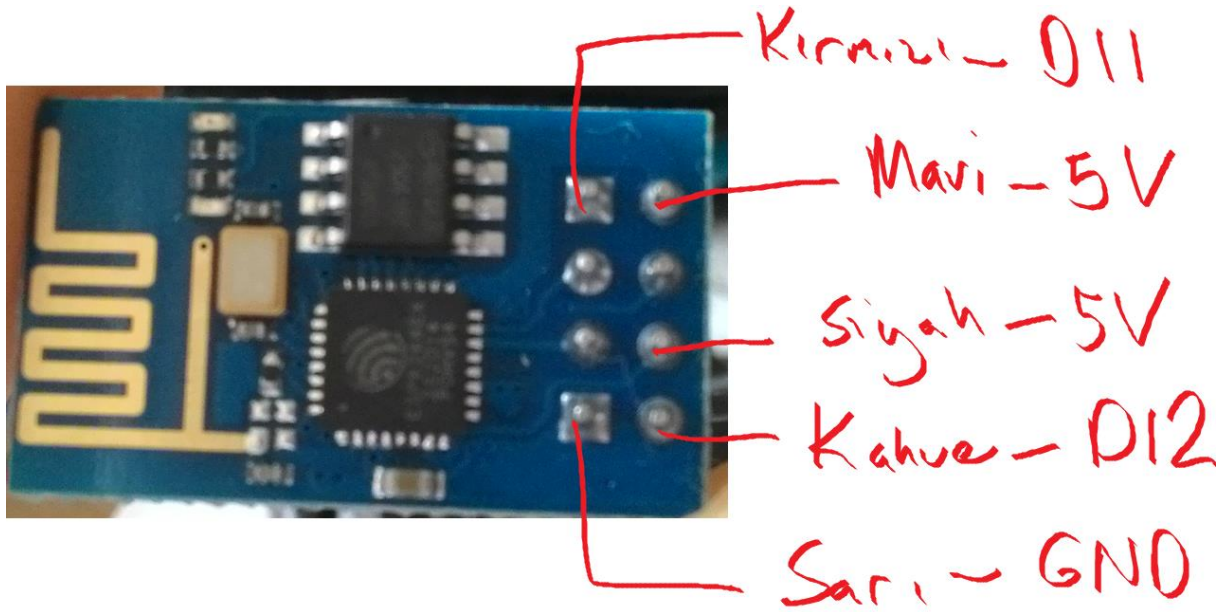
Resim 5 – ESP8266, bağlantıların arkadan görünüşü

Şematik olarak önden (devre elemanlarını gören şekilde bakınca) Resim 6'da verilmektedir.



Resim 6 – ESP8266 şematik

Kabloların bağlantısının kablo renkleri çizili hali ise şu şekildedir.

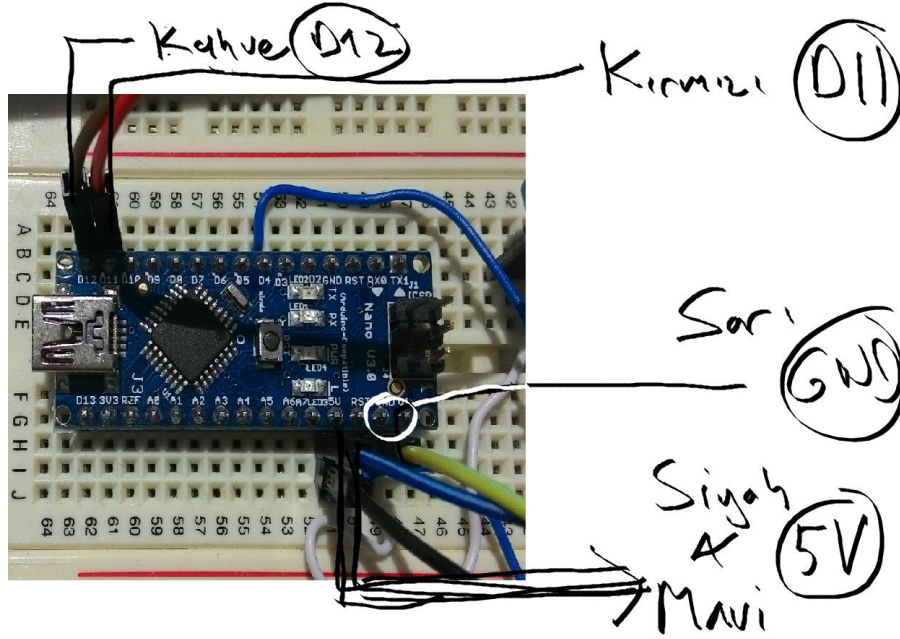


Resim 7 – Benim bağlantı kodlarım ve Arduino üzerinde bağlandıkları noktalar

ESP8266	Kablo Rengi	Arduino
RxD	Kırmızı	D11
VCC	Mavi	5V
CH_PD	Siyah	5V
GND	Sarı	GND
TxD	Kahve	D12

Tablo 1 – ESP 8266 ve Arduino bağlantısı

Bu pinlerden 5V ve GND olanlar ESP'nin besleme ve diğer ayarları için yapılırken, TxD ve RxD veri alışverişi için kullanılmaktadır. Tüm bu bağlantıların Arduino üzerindeki görüntüsü de şu şekildedir.



Resim 8 – Kabloların Arduino'daki bağlantı durumları

## 2.Aşama ESP8266'nın Arduino üzerinden Test Edilmesi

İlk programımız sadece ESP8266'ya bağlantıyı test edecek. Programımızın ismi esp826.ino. Bu program sadece ama sadece Arduino'ya USB üzerinden gönderdiğimiz komutların ESP'ye iletilip ESP'den gelen cevabın gene USB üzerinden ekranımıza geri gelmesini sağlıyor.

Kullanacağımız kütüphane "SoftwareSerial.h", Arduino standart kütüphanesidir. Kodumuzun en başında esp8266(Rx, Tx) ile alıcı ve verici pinlerimizi bildiriyoruz. Ben yukarıdaki resimlerden de görüleceği üzere 12 ve 11'inci dijital pinleri (D12 ve D11)'i belirledim. Baud rate 9600 olsun.

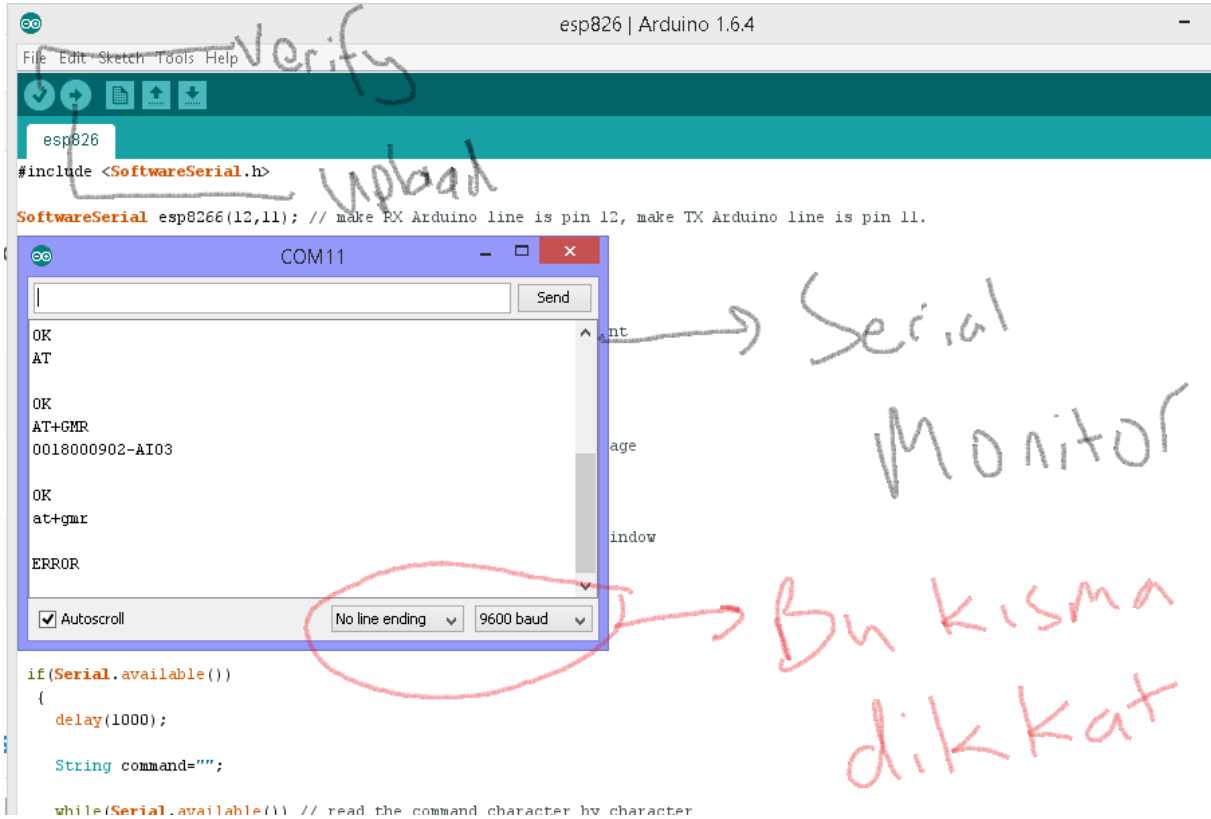
```
SoftwareSerial esp8266(12,11);
```

Programı verify ve upload ettikten sonra Tools menüsünden Serial Monitor'u açıyoruz. Serial Monitor'de en altta 9600 ve "No line ending" seçili olmasına dikkat ediyoruz. İnternette "Both NL & CR" seçeneğini önermesine rağmen ben bu konuda yazdığım komutların arkasına garip karakterler eklediğini gördüğümünden yapmadım.

Şimdi resim 9'daki gibi bazı komutları deneyelim:

Yazdığım Komut	ESP8266'nın cevabı Cevabı
AT	OK
AT+GMR	0018000902-AI03
at+gmr	ERROR

Tablo 2 - ESP bağlantısını yaptıktan sonra Resim 9'daki gibi bazı komutlar ve cevapları



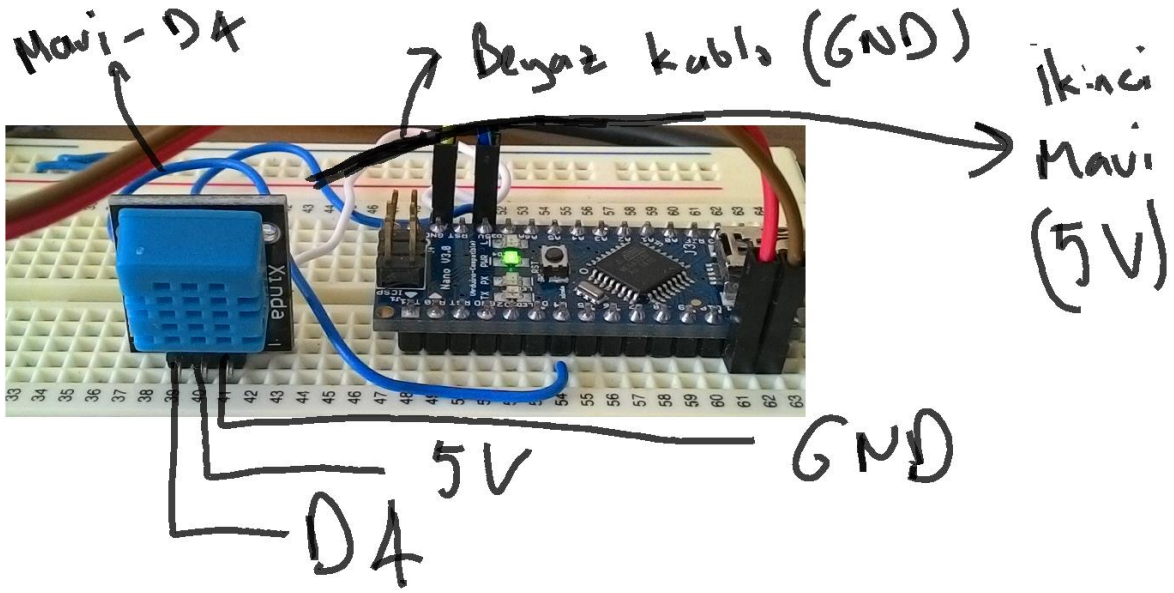
Resim 9 – ESP8266 ile ilk temas

Dikkat etmemiz gereken birinci nokta ESP8266'ya gönderilen komutların büyük harf olması gerektir. ESP8266'ya gönderebileceğiniz komutların listesi şu adreste mevcut:

<https://room-15.github.io/blog/2015/03/26/esp8266-at-command-reference/>

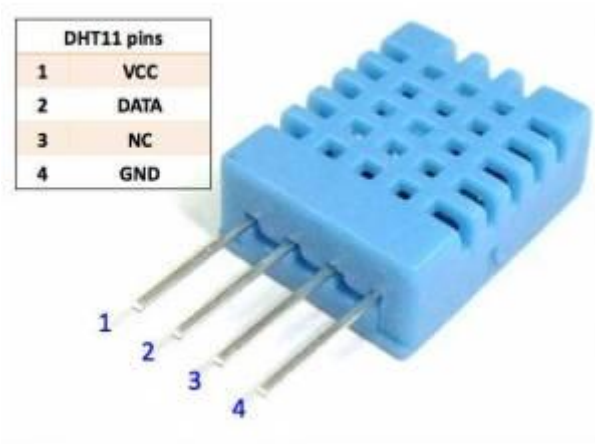
### 3.Aşama DHT11'in Bağlanması

DHT11'in bağlanması için ise internetteki kaynakları mutlaka inceleyin, benim aldığım sensör 3 bacaklı olduğundan benim devreye bakarak 4 bacaklı DHT11'in nasıl 3 bacaklı hale getirildiğine bakmam gerekiyordu. Robotistan'dan aldığım parçada kenarda bir R gördüğümden normalde Data pin'inine bağlamam gereken pull-up resistörü ihmal ettim. Bağlantı Resim 10'daki gibi oldu.



Resim 10 – DHT11'in(4 pin'den 3pine dönüşüm devresi ile) Arduino'ya bağlantısı

Eğer devresiz 4 bacaklı dht11'iniz var ise aşağıdaki şekilde pinleri kullanın. DHT11'in 2 nolu DATA pin'i Arduino D4'e bağlanacak.Sensörünüz aşağıdaki şekilde ise **DATA ile VCC arasına mutlaka 10 k Ohm direnç koyuyorsunuz.**



Resim 11 – DHT11 orijinal sensör 4 bacaklı.

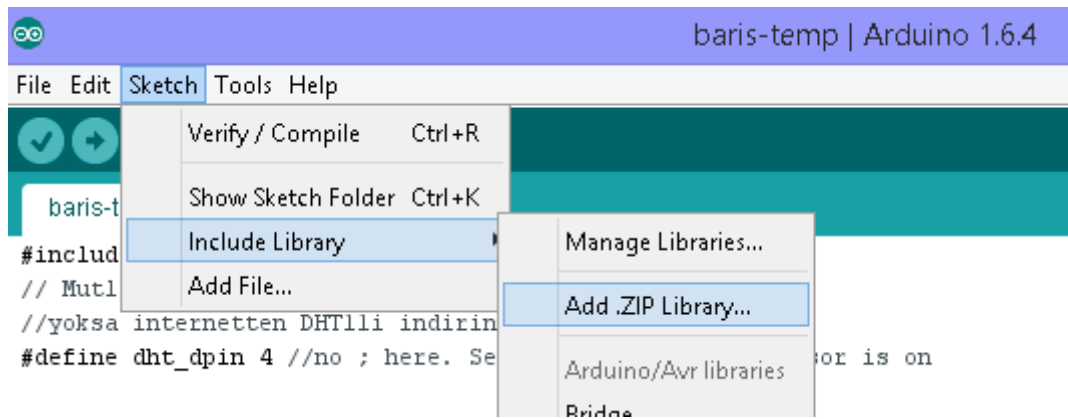
#### 4.Aşama – DHT11'den sıcaklık ve nem okuması

İnternette birkaç kütüphane var ve bunlar ya DHT11 ya da DHT kütüphaneleridir. Ben DHT diye basit ve az hafıza gerektiren bir kütüphane buldum. Bu kütüphaneyi internetten aşağıdaki linkten bulabilirsiniz.

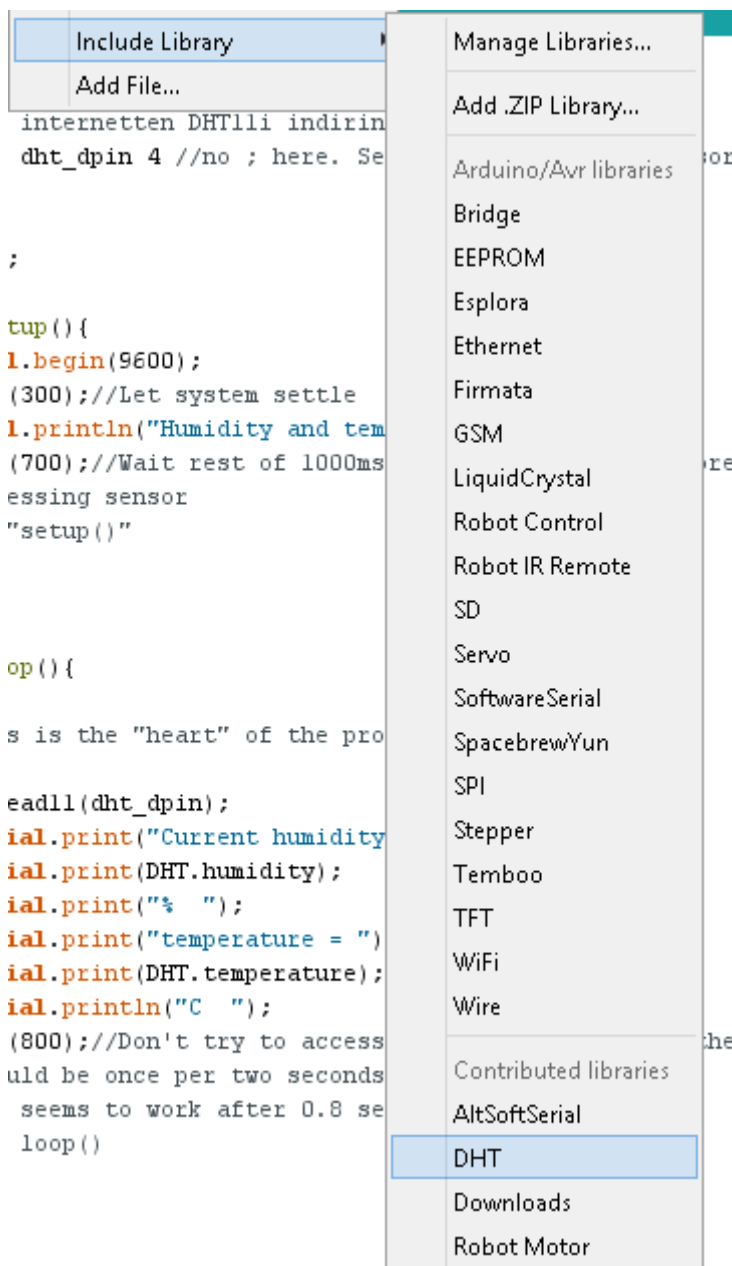
<http://playground.arduino.cc/Main/DHTLib>

Kütüphane'yi download ettikten sonra Resim 12'deki gibi zip dosyasını önce kütüphanenize ekleyin. Ekledikten sonra DHT kütüphanesi Resim 12'deki menünün en altında DHT olarak görülecektir.



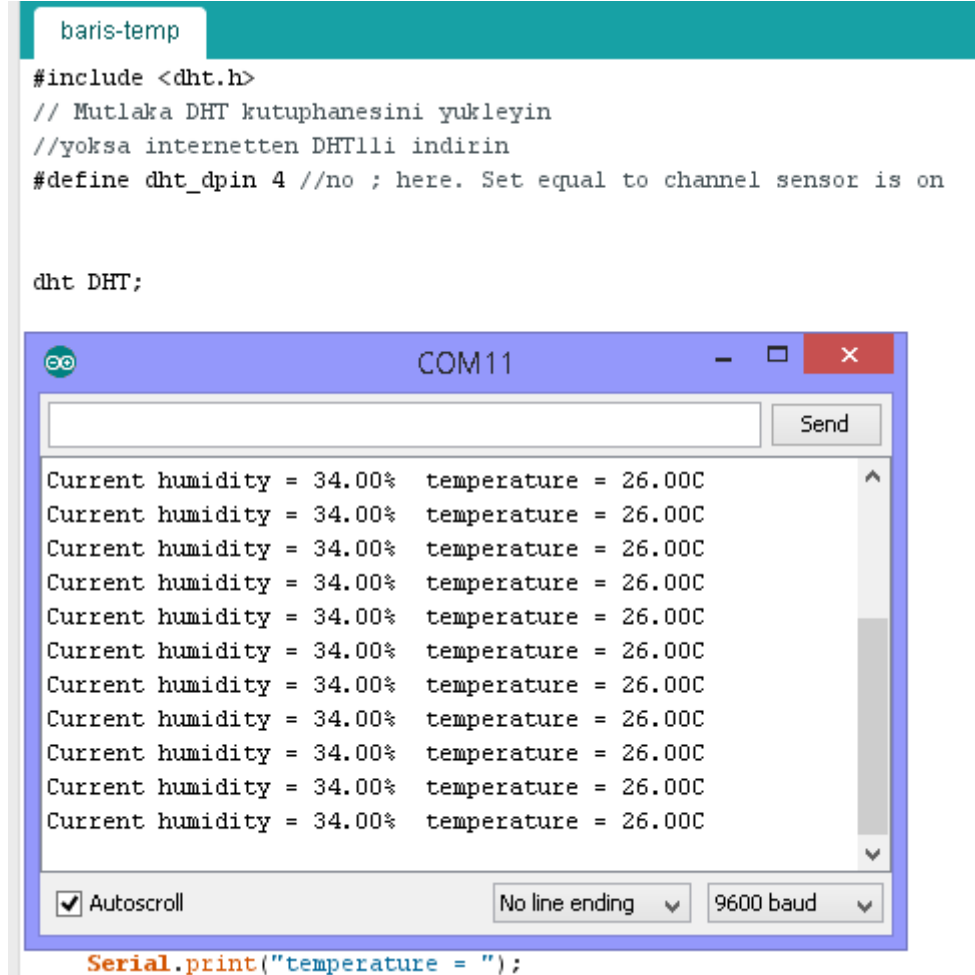


Resim 12 – DHT.zip’i kütüphane olarak ekleyin



Resim 13- DHT kütüphanesini kullanmak için ekledikten sonra DHT'yi seçin

Programın ismini baris-temp olarak değiştirdim. Programı tekrar verify&upload yaptıktan sonra Serial Monitor'ü açtığımızda ekranımıza aşağıdaki şekilde bilgiler akmaya başlayacaktır(Resim 14).



Resim 14- Seri port'tan sıcaklık ve nem okuması

## 5.Aşama – Web Sunucusunun Çalıştırılması

İşte asıl bu nokta biraz zorlayıcı oldu. Bu yüzden bu kısımda kodu uzun uzun anlatma ihtiyacı duyacağım. İnternetteki birkaç kodu birbirine karıştırdım. Ama çok güzel kodlardan birisi şu adreste:

<http://allaboutee.com/2015/01/02/esp8266-arduino-led-control-from-webpage/>

Şimdi sırası ile yapacaklarımız şunlar

- İlgili Kütüphaneyi ekle
 

```
#include <dht.h>
#include <SoftwareSerial.h>
```
- Hem ESP8266 ile hem de DHT11 ile bağlanacak pinleri belirle, kullanacağımız değişiklikleri belirle

```
#define DEBUG true
```

```
SoftwareSerial esp8266(12,11); // RX Arduinoda pin 12ye,TX Arduinoda pin 11e.
```

```
int status=0;
```

```
#define dht_dpin 4 //no ; here. Set equal to channel sensor is on
```

```
dht DHT;
```

c. Setup kısmında

```
void setup()
```

```
{
```

- a. Önce iletişim hızını 9600 yap

```
Serial.begin(9600);
```

```
esp8266.begin(9600);
```

- b. Sonra ESP'yi resetle

```
sendData("AT+RST\r\n",2000,DEBUG); // reset module
```

- c. ESP'yi bir kablosuz noktaya bağlanacak hale getir.

```
sendData("AT+CWMODE=1\r\n",1000,DEBUG);
```

- d. Bağlanacağımız kablosuz modemin ismi ve şifresini gir ve bağlanmak için biraz bekle.

Burada komutunuz aşağıdaki şekilde olmalıdır: \" karakteri ise tırnak işaretinin

karakter olarak algılanması içindir.

```
sendData("AT+CWJAP=\"Kablosuz nokta\", \"Sifre\"\r\n",5000,DEBUG);
```

```
// configure
```

```
sendData("AT+CWJAP=\"",\",\", \"\r\n\",5000,DEBUG);
```

```
delay(3000);
```

- e. IP adresini al

```
sendData("AT+CIFSR\r\n",3000,DEBUG); // get ip address
```

- f. Çoklu bağlantıya ayarla

```
sendData("AT+CIPMUX=1\r\n",1000,DEBUG); // configure for multiple connections
```

- g. Sunucu olarak 80nci portu aç

```
sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG); // turn on server on port 80
```

d. Loop Kısmında

```
void loop()
```

```
{
```

- a. DHT'ye bağlan

```
DHT.read11(dht_dpin);
```

- b. ESP8266 hazır ise

```
if(esp8266.available()) // check if the esp is sending a message
```

```
{
```

- c. Biraz bekle

```
delay(1000);
```

- d. connectionID'yi al

```
int connectionId = esp8266.read()-48;
```

- e. Web sayfası verisini hazırla, DHT'den aldığın sıcaklık (DHT.temperature) ve nem

(DHT.humidity) değerlerini concat komutu ile ekle. En sona da bir link ile komut ekle

```
String webpage = "<h1>Arduino! - BARIS SANLI</h1><br><h1>Nem: ";
webpage.concat(DHT.humidity);
webpage+="<h1>Sicaklik ";
webpage.concat(DHT.temperature);
webpage+=" C </h1>";
webpage+="<br><a href=\"?pin=on\"><button><h1>LED2</h1></button></a>";
```

f. Hazırladığın web sayfasını gönder

```
String cipSend = "AT+CIPSEND=";
cipSend += connectionId;
cipSend += ",";
cipSend +=webpage.length();
cipSend += "\r\n";
```

```
sendData(cipSend,1000,DEBUG);
sendData(webpage,1000,DEBUG);
```

g. Ve kapat

```
String closeCommand = "AT+CIPCLOSE=";
closeCommand+=connectionId; // append connection id
closeCommand+="\r\n";
```

```
sendData(closeCommand,3000,DEBUG);
delay(1000);
```

Bu programı indirdiğiniz zaman sendData fonksiyonunu da göreceksiniz. sendData fonksiyonu, kendisine verilen komutları yazıyor ve sistemden cevabını bekliyor. Gelen cevabı kullanıcıya String şeklinde geri dönüyor.(Resim 15). Eğer debug değeri doğru ise komutları aynen Seri port'a da yazıyor. Yani "Serial Monitor'den neler döndüğünü öğrenebilirsiniz. (Resim 16)

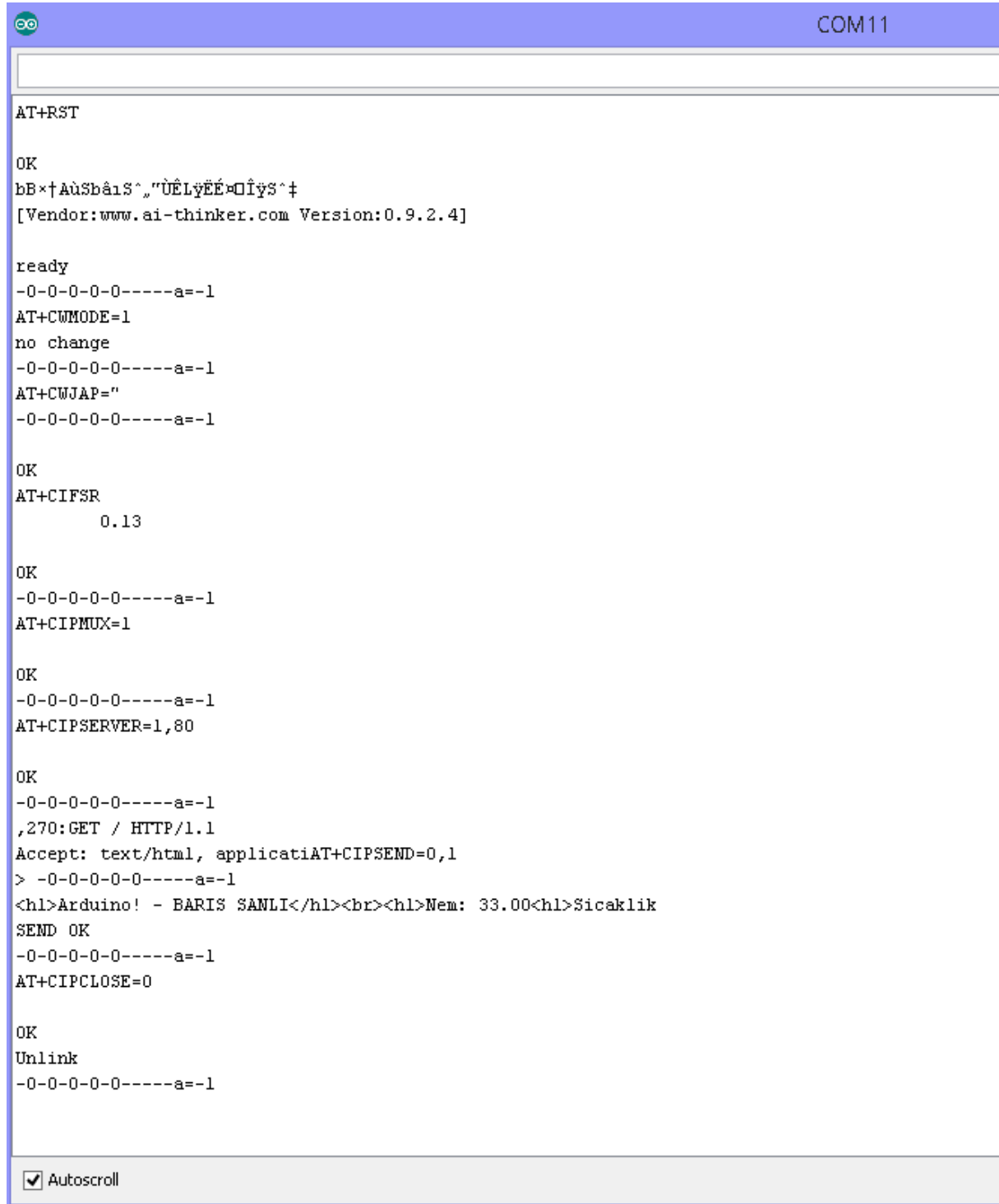
```
String sendData(String command, const int timeout, boolean debug)
{
    String response = "";

    esp8266.print(command); // send the read character to the esp8266

    long int time = millis();

    while( (time+timeout) > millis())
    {
        while(esp8266.available())
        {
            // The esp has data so display its output to the serial window
            char c = esp8266.read(); // read the next character.
            response+=c;
        }
    }
}
```

Resim 15- Send data komutu



```

COM11
AT+RST

OK
bB×†AùSbâ1S^,"ÜËLÿËË»QÏÿS^†
[Vendor:www.ai-thinker.com Version:0.9.2.4]

ready
-0-0-0-0-0-----a=-1
AT+CWMODE=1
no change
-0-0-0-0-0-----a=-1
AT+CWJAP=""
-0-0-0-0-0-----a=-1

OK
AT+CIFSR
      0.13

OK
-0-0-0-0-0-----a=-1
AT+CIPMUX=1

OK
-0-0-0-0-0-----a=-1
AT+CIPSERVER=1,80

OK
-0-0-0-0-0-----a=-1
,270:GET / HTTP/1.1
Accept: text/html, applicatiAT+CIPSEND=0,1
> -0-0-0-0-0-----a=-1
<h1>Arduino! - BARIS SANLI</h1><br><h1>Nem: 33.00<h1>Sicaklik
SEND OK
-0-0-0-0-0-----a=-1
AT+CIPCLOSE=0

OK
Unlink
-0-0-0-0-0-----a=-1
 Autoscroll

```

Resim 16 – ESP8266'nın debug değişkeni TRUE olduğunda serial monitör üzerinden haberleşmenin izlenmesi. (AT+CWJAP kısmında şifreler olduğu için silindi)

## 6.Aşama – Arduino Nano üzerindeki LED'in açılıp kapanması

Evet, sunucumuz hazır şu aşamada ekrana sıcaklık ve nem değerlerini yazan bir sunucu elde ettik. Fakat LED2 isimli düğmeye basıldığından ne yapılacağını belirlemedik.LED2 düğmesine basıldığından sistem, <http://adres/?pin=on> komutunu çağırıyor. Serial Monitor'den gözükten bu diziyi gelen metinde aratarak lamba açık ise kapalı, kapalı ise açık hale getireceğiz.

```
,308:GET /?pin=on HTTP/1.1
Accept: text/html, apAT+CIPSEND=0,1
> -0-0-0-0-0-----a=4
```

Resim 17 – LED2 düğmesine basıldığında ESP üzerinden Arduino'ya gelen mesaj

LED2'ye basıldığında, ESP'ye "GET /?pin=on HTTP/1.1" mesajı geliyordu. Ben de bu mesajdaki kısmı ".indexOf" komutu ile aratıyorum. Eğer gelen mesajda bu kısım yok ise a değişkeni "-1" değerini alıyor. Ama eğer bulursa cevap 4 oluyor. Ben işi sağlama almak için, 3'ten büyük tüm cevapları kabul ettim. Eğer cevap 3'ten büyük ise, dijital 13.pin ki Arduino nano üzerindeki kırmızı LED'e işaret eder, status'e göre açar veya kapatır(Resim 18).

```
int a=response.indexOf(":GET /?pin=on");
Serial.print("-0-0-0-0-0-----a=");
Serial.println(a);
if(a>3){
  pinMode(13, OUTPUT);
  if(status==0){
    digitalWrite(13,HIGH);
    status=1;
  }else{
    digitalWrite(13,LOW);
    status=0;
  }
}
```

Resim 18 – Gelen cevapta ?pin=on ara

## Sonuç

Evet Programımızı Arduino'ya yükleyip çalıştırınca resim 19'daki sonuç gözlenir. LED2'ye tıklayınca da Arduino üzerindeki LED yanıyor, söner, sönüyorsa yanar. Bu tip programlarda LED yakmak amaç değildir. Amaç Röle çalıştırmak olduğundan, bu konuya sonra geleceğiz. Şimdilik sadece röle çalıştırın.



Resim 19 – Arduino – ESP8266 sıcaklık suncusu

14 Haziran 2015 – Barış Sanlı

Sorular: [barissanli2@gmail.com](mailto:barissanli2@gmail.com)

Kodların adresi: <http://www.barissanli.com/electronics/arduino-esp-webserver-barissanli.zip>

