

# ESP8266 VE ARDUINO İLE NESNELERİN İNTERNETİ

---

HASBİ SEVİNÇ

# İÇİNDEKİLER

<b>BÖLÜM 1: GİRİŞ</b>	<b>1</b>
Ne Demektir Bu Nesnelerin İnterneti?	1
Yerel Ağdan İnternet Ağına Çıkma	2
Neler Öğrendik?	4
<b>BÖLÜM 2: ESP8266 İLE WIRELESS HABERLEŞMEYE GİRİŞ</b>	<b>7</b>
ESP8266 Nedir?	8
ESP8266 Modülleri	10
Arduino'nun Seri Haberleşme Modülü Olarak Kullanılması	11
ESP8266 Baud Rate Ayarı	12
ESP8266 Firmware Güncellemesi	13
AT Komutları	15
Neler Öğrendik?	16
<b>BÖLÜM 3: ARDUINO İLE ESP8266</b>	<b>19</b>
SoftwareSerial Kütüphanesi	20
ESP8266 ile Arduino Uno mu Mega mı Kullanmalı?	20
Arduino Mega ile İnternet Sitesine Bağlanma	21
Arduino ile Hava Durumu	25
ThingSpeak.com Nedir?	30
Arduino ile Oda Sıcaklığını İnternete Yükleme	30
Arduino ile Tweet'leme	35
Arduino ile Server Yapımı	38
İnternet Üzerinden LED Kontrolü	39
İnternet Üzerinden Şifreli Lamba Kontrolü	44

Mağazaya Giren Kişi Sayısını İnternette Kontrol Edelim	50
Arduino ile TCP Server'a Bağlanma	55
TCP Server Programı	55
Arduino ile Chat Odası	56
İnternet Üzerinden Kontrol Edilebilen Araç Yapımı	60
Neler Öğrendik?	67

## **BÖLÜM 4: NODEMCU İLE ESP8266** **69**

Giriş	70
LUA Programlamaya Giriş	70
Koşul Yapıları	71
NODEMCU ile LUA Diline Eklenen Fonksiyonlar	72
ESP8266'nin NodeMCU Firmware Güncellemesi	73
FTDI Dönüştürücüsü ile Güncelleme	74
Arduino'nun Bypass Edilerek ESP8266'nın Güncellenmesi	75
NodeMCU-Lua İçin IDE Seçimi	76
ESP8266'nın Harici Beslenmesi	77
Yanıp Sönen LED	78
ESP8266 ile Web Sayfası Oluşturma	80
ESP8266 ile Çoklu Web Sayfası Oluşturma	81
İnternet Üzerinden LED Kontrolü	82
İnternet Kontrollü RGB LED ve Şerit	85
ESP8266 ile RGB Şerit LED Kullanımı	89
ESP8266 ile Tweet Atma	89
DS18B20 ile Sıcaklık Ölçümü	92
Buluta Veri Yollama	94
DS18B02 ile Oda Sıcaklığını Buluta Yükleme	96

Mail Bildirimli Hırsız Alarmı	99
Pushingbox.com Üzerinden Apı Alma	99
ESP8266 ile Pushingbox Bildirim Yollama	100
TCP Server Üzerinden Chat Yapma	102
Neler Öğrendik?	104
<b>BÖLÜM 5: ESP-07 MODÜLÜ</b>	<b>107</b>
ESP-07 Modülüne NodeMCU Yükleme	108
NodeMCU ile İnternet Üzerinden Araç Kontrolü	108
STATION Modunda Araç Kontrolü	110
Access Point Modunda Araç Kontrolü	112
Son Söz	115
İndeks	116

# 2

## ESP8266 İLE WIRELESS HABERLEŞMEYE GİRİŞ

### BU BÖLÜMDE

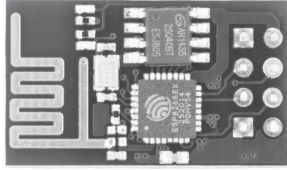
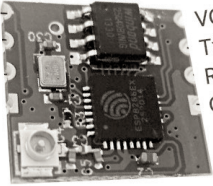

ESP8266 Nedir?	8
ESP8266 Modülleri	10
Arduino'nun Seri Haberleşme Modülü Olarak Kullanılması	11
ESP8266 Baud Rate Ayarı	12
ESP8266 Firmware Güncellemesi	13
AT Komutları	15
Neler Öğrendik?	16

Arduino'yu internete bağlamanın en ucuz yolu ESP8266 Wifi modülü kullanmaktır. Bu modül ortamda bulunan kablosuz ağlara bağlanabileceği gibi, kendi internet ağını yayarak diğer cihazların bu ağa bağlanabilmesine de imkan sağlamaktadır. Bölüm 2'de ESP8266'nın teknik detaylarını ve bu modülü kullanmadan önce yapmamız gereken ayarlamaları göreceğiz.

## ESP8266 NEDİR?

ESP8266, TTL (Seri Haberleşme) ile kablosuz internet ağına bağlanabilen bir modüldür. Ucuz ve kolay kullanıma sahip olmasından dolayı nesnelerin interneti projelerinde yaygın olarak kullanılmaktadır. ESP8266 modülü 3.3 Volt ile çalışmaktadır. Yani modülün çalışması için Arduino'nun 3.3 Volt çıkışı kullanılabilir. Fakat ağa bağlanma ve veri alışverişi sırasında anlık olarak Arduino'nun sağlayabileceği akımdan fazla akım çekebilmektedir. Bu yüzden ESP8266'yı harici bir 3.3 Volt gerilim kaynağından besleyerek kullanmak, modülün daha istikrarlı çalışmasını sağlayacaktır. Unutulmamalıdır ki harici 3.3 Volt kaynağının ve Arduino'nun toprak hatları birbirine bağlanmalıdır.

Yapacağımız uygulamalarda ESP-01 modülü kullanılacaktır. Bu modül ve diğer modüller arasındaki fark giriş ve çıkış pinlerinin sayısıdır. Kod kısmında bir değişiklik olmadan diğer ESP modülleri de kullanılabilir. Yeri gelmişken belirtelim, ESP-01'in seçilme nedeni diğer modüllere göre pin bağlantılarının daha kolay olmasıdır. Modül üzerinde VCC (3.3 Volt Besleme), GND (Toprak), TX, RX, RST (Reset), CH\_PD, GPIO0 ve GPIO2 olmak üzere 8 adet pin bulunmaktadır.

<p><b>ESP-01</b></p> <ul style="list-style-type: none"> <li>- PCB Anten</li> <li>- 3 Giriş/çıkış pini: GPIO0/2/16</li> <li>- Diğer pinler; UART (URXD/UTXD)</li> </ul> <p>RESET, CH_PD (Güç kapatma) ve VCC/GND</p> <p>- Boyut: 14.3 mm x 24.8 mm</p>	 <p>Rx VCC GPIO0 RST GPIO2 CH_PD GND Tx</p>
<p><b>ESP-02</b></p> <ul style="list-style-type: none"> <li>- Extra anten için U-FL konnektörü bulunur</li> <li>- 2 Giriş/çıkış pini: GPIO0/2</li> <li>- Diğer pinler: UART (URXD/UTXD)</li> </ul> <p>RESET, CH_PD (Güç kapatma) ve VCC/GND</p> <p>- Boyut: 14.2 mm x 14.2 mm</p>	 <p>GPIO15 VCC GPIO0 Tx GPIO2 Rx RST GND</p>
<p><b>ESP-03</b></p> <ul style="list-style-type: none"> <li>- Seramik Anten</li> <li>- 7 Giriş/çıkış pini: POIO/2/12/13/14/15/18</li> <li>- Diğer pinler: UART (URXD/UTXD)</li> </ul> <p>RESET, CH_PD (Güç kapatma) ve VCC/GND</p> <p>- Boyut: 12.1 mm x 17.3 mm</p>	 <p>VCC Anten GPIO14 CH_PD GPIO12 GPIO18 GPIO13 Rx GPIO15 Tx GPIO2 Boş GPIO0 GND</p>

ESP8266'nın baud rate hızının 9600'e çekilmesi için aşağıdaki komutu kullanınız. Eğer OK mesajını gördüyseniz yapmanız gereken işlem başarıyla gerçekleşmiştir.

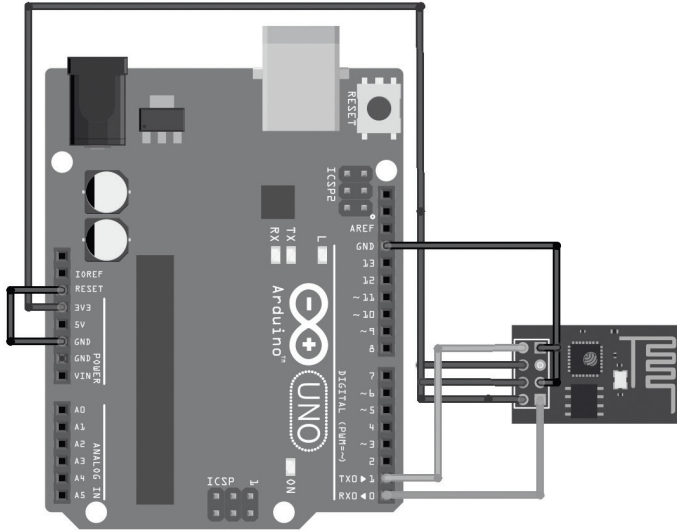
```
AT+CI0BAUD=9600
```

Bu işlem sonrasında tekrardan veri yollayabilmek için artık ekranın sağ altından baud rate hızınızı 9600 yapmayı unutmayınız.

## ESP8266 FIRMWARE GÜNCELLEMESİ

ESP8266 kablosuz internet modülüne, ilk üretildiği günden beri bir çok yazılım güncellemesi geldi. Bu güncellemeler sayesinde modül daha stabil çalışmaya başladı. Bu yüzden projelerimizi yapmadan önce modülün yazılımını güncellemeliyiz.

Güncelleme için Arduino'yu daha önce öğrendiğimiz şekilde bypass yapmalıyız. Daha sonra modülün güncelleme moduna girebilmesi için GPIO0 pinini toprağa bağlamalıyız. Aşağıdaki resimde güncelleme için kurulması gereken devre şeması gösterilmiştir. (ESP-12 kullanıcıları, GPIO0 ve GPIO15 pinlerini toprağa ve GPIO1 pinini 3.3 Vo1ta bağlamaları gerekmektedir.)



# 3

## ARDUINO İLE ESP8266

### BU BÖLÜMDE

SoftwareSerial Kütüphanesi	20
ESP8266 ile Arduino Uno mu Mega mı Kullanmalı?	20
Arduino Mega ile İnternet Sitesine Bağlanma	21
Arduino ile Hava Durumu	25
ThingSpeak.com Nedir?	30
Arduino ile Oda Sıcaklığını İnternete Yükleme	30
Arduino ile Tweet'leme	35
Arduino ile Server Yapımı	38
İnternet Üzerinden LED Kontrolü	39
İnternet Üzerinden Şifreli Lamba Kontrolü	44
Mağazaya Giren Kişi Sayısını İnternette Kontrol Edelim	50
Arduino ile TCP Server'a Bağlanma	55
TCP Server Programı	55
Arduino ile Chat Odası	56
İnternet Üzerinden Kontrol Edilebilen Araç Yapımı	60
Neler Öğrendik?	67

Şu ana kadar ESP8266'nın nasıl çalıştığını ve modülün Arduino ile nasıl haberleşeceğini öğrendik. Bu bölümde ESP8266'yı Arduino ile kullanarak çeşitli nesnelerin interneti projeleri geliştireceğiz. Bu bölümdeki örnekler üzerinde değişiklikler yaparak yeni projeler üretebileceğinizi unutmayın.



```

"\r\n",10000,true); // wifi bağlantısı yapıyor
String durum = "";
durum = ATKomuduYolla("AT+CIFSR\r\n",8000,true);
// bağlantı bilgileri
if(durum.indexOf("FAIL") != -1 ){
// bağlantı başarı durumu test ediliyor
return false;
}
return true;
}

```

## ARDUINO İLE TWEET'LEME

Bir önceki uygulamamızda nasıl **thingSpeak.com** adresine yollamak istediğimiz verileri POST metodu ile yolladıysak, aynı şekilde tweet'lerimizi de **Twitter.com** adresine kendi kullanıcı adımız ile yollayabiliriz. Twitter'ın mesajlarımızı kabul etmesi için öncelikle **Token** denilen biletlere sahip olmamız gerekir. Bu biletleri sağlayan ve mesaj yollamamızı kabul eden Twitter uygulamaları mevcut.

Bu uygulamalara örnek olarak:

<http://arduino-tweet.appspot.com/oauth/twitter/login>

Öncelikle linke tıklayınız ve link yardımıyla Twitter'a giriş yapınız. Buradan giriş yaptığınızda uygulamaya izin vermenizi isteyecektir. Uygulamaya izin verdikten sonra sayfada size ait **Token** yani bilet numaranız görünecektir. Bu bileti birazdan Arduino kodumuza yazacağız. Bu bileti bir yere kaydetmeyi unutmayınız.

Twitter, çok sık ve benzer mesajlar attığınızda sizin mesajlarınızı filtrelemektedir. Bu yüzden bir sonraki tweet'inizi atmadan önce bir süre beklemeyi ve bir önceki tweet'inizden farklı bir tweet atmayı unutmayın.

### Kullanılacak Malzemeler

1 x Arduino

1 x ESP8266 (ESP-01)

### Arduino Kodu

```

#include <SoftwareSerial.h>

SoftwareSerial wifiBaglantisi(10, 11);

const String wifiAdi = " WIFI_ADI ";

```

```
const String wifiSifresi = " WIFI_SIFRE ";

const String TOKEN = "TOKEN_BURAYA ";

void setup()
{
  Serial.begin(9600);
  wifiBaglantisi.begin(9600);
  delay(500);
  delay(1000);

  if(!internetBaglan()){
    Serial.println("internet baglanilamadi");
  }
  else {
    Serial.println("Modul internete baglandi");
  }
}

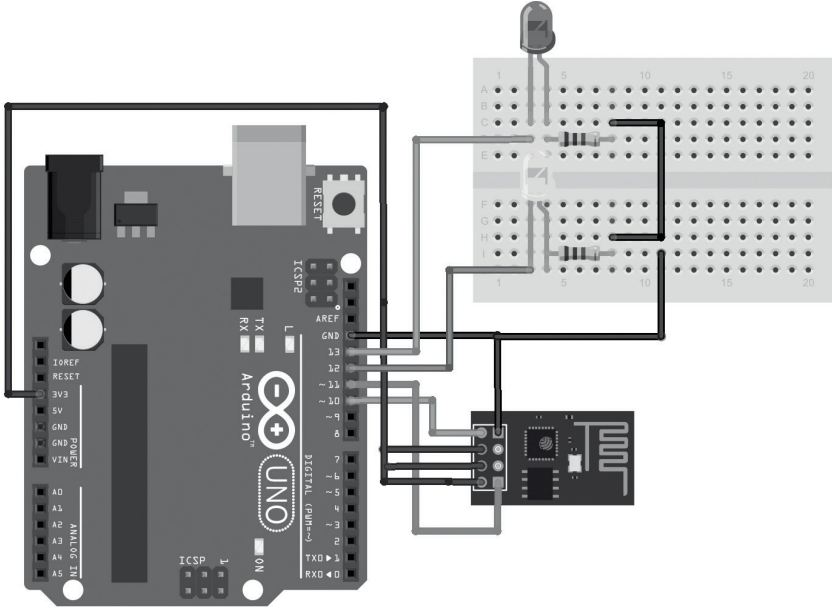
void loop()
{
  String mesaj = "ESP ile deneme mesaji";
  tweetle(mesaj);

  Serial.println("Tweet islemi tamamlandi");
  while(1);
}

void tweetle(String tweet){
  String mesaj = "token="+TOKEN+"&status="+tweet;
  String istek;
  istek = "POST http://arduino-tweet.appspot.com/update HTTP/1.0\r\nHost:
arduino-tweet.appspot.com" "\r\nContent-Length: "+String(mesaj.length())+
"\r\n\r\n";

  wifiBaglantisi.print("AT+CIPSTART=\r\nTCP\r\n,\r\narduino-tweet.appspot.com\r\n,80\r\n");
  wifiBaglantisi.find("OK");

  Serial.println("Baglanti istegi yollandi");
```



fritzing

### Arduino Kodu

```
#include <SoftwareSerial.h>
SoftwareSerial wifiBaglantisi(10, 11);
/* Arduino'nun 10. pini ESP'nin Tx, 11. pini ESP'nin Rx pinine bağlandı */

const String wifiAdi = " WIFI_ADI ";
const String wifiSifresi = " WIFI_SIFRE ";

#define DEBUG true

const int LED1 = 13, LED2 = 12; // LED pinleri

void setup()
{
  Serial.begin(9600); // Seri haberleşme bağlantısı kuruldu
  wifiBaglantisi.begin(9600); // ESP bağlantısı kuruldu
  delay(500);
  pinMode(LED1,OUTPUT); // LED pinleri çıkış olarak tanımlandı
  pinMode(LED2,OUTPUT);
}
```

```

    return cevap;
}

boolean interneteBaglan(){
    ATKomuduYolla("AT+CWMODE=1",1000,true);
    // internete bağlanacağımız için 1. moda alıyoruz
    ATKomuduYolla("AT+CWJAP=\"" + wifiAdi + "\",\""+ wifiSifresi + "\",10000,true);
    // wifi bağlantısı yapılıyor
    String durum = "";
    durum = ATKomuduYolla("AT+CIFSR",8000,true); // bağlantı bilgileri
    if(durum.indexOf("FAIL") != -1 ){
    // bağlantı başarı durumu test ediliyor
        return false;
    }
    return true;
}

```

Yukarıdaki kodu Arduino'ya yükledikten sonra **Serial Monitör**'ü açalım. Burada öncelikle ESP8266'nın internete bağlandığını söyleyen mesajı görmeliyiz. Eğer bağlantı başarı ile gerçekleştirildi ise Arduino Server'ı oluşturacaktır. Server oluşturulduktan sonra ESP8266'nın ağ içerisinde aldığı yerel IP adresi ve kodda belirtilen port numarası ile Server'a bağlanabiliriz. Örneğin: 192.168.2.224:80

Buradaki önemli notlardan birisi Server kullanıcıya bağlantının yükleme sonrasında kapatıldığını belirten `Connection: close\r\n` komutunu yollamasıdır. ESP8266'nın bağlantı sonlandırma kodu bulunmaktadır. Fakat bu kod zaman zaman yeni bağlantıları engellemektedir.

Yukarıdaki kodlar üzerinde değişiklikler yaparak siz de kendinize ait bir Server oluşturabilir ve evinizdeki elektronik eşyaları bu Server üzerinden kontrol edebilirsiniz.

## İNTERNET ÜZERİNDEN ŞİFRELI LAMBA KONTROLÜ

Bir önceki örnekte ESP8266 ve Arduino yardımıyla basit bir Server'ın nasıl kurulacağını öğrendik. Örnekte sadece LED'ler kontrol edilmiştir. Bu uygulamada farklı olarak röle kontrolü yapılacaktır. Bu röleye bağlanan lamba, LED'ler gibi internet üzerinden kontrol edilebilecektir. Örnekte röleye lamba bağlanmıştır. Siz de evdeki diğer elektronik eşyalarınızı lamba yerine bağlayarak internet üzerinden kontrol edebilirsiniz.

## ARDUINO İLE CHAT ODASI

TCP Server Arduino ile internete bağlı diğer cihaz arasında hızlı haberleşmeyi sağlamak için kurulmaktadır. Böyle bir bağlantı kurulduğunda ilk yapmamız gereken gönderilen mesajların karşı tarafa düzgün ulaşip ulaşmadığını test etmektir. TCP haberleşme mantığının kavranması ve sistemin çalışıp çalışmadığını test etmek için Bilgisayar programı ve Arduino'nun katıldığı bir chat (*sohbet*) odası kuracağız.

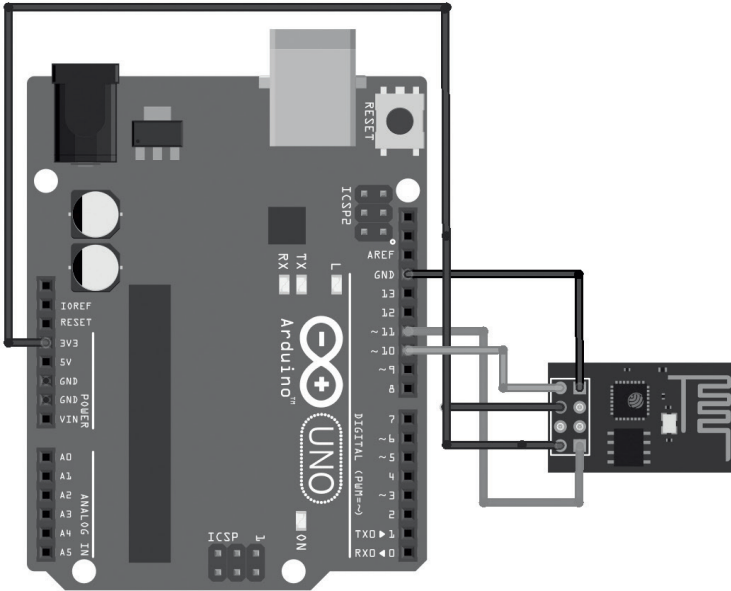
Bu uygulamada bilgisayar programı olarak bir önceki konuda tanıttığımız programı kullanacağız. Program açıldığı gibi TCP Server kurulmaktadır. Kurulan TCP Server'in IP ve port numaraları programın altında yazmaktadır. Bu sayıları Arduino kodumuzda güncellememiz gerektiğini unutmayalım.

### Kullanılacak Malzemeler

1 x Arduino

1 x ESP8266 (ESP-01)

Devre şeması aşağıdaki resimde gösterilmiştir. Bu devrede ESP8266 daha önceki devre şemalarında olduğu gibi bağlanmıştır.



# 4

## NODEMCU İLE ESP8266

### BU BÖLÜMDE

Giriş	70
LUA Programlamaya Giriş	70
NODEMCU ile LUA Diline Eklenen	
Fonksiyonlar	72
ESP8266'nin NodeMCU Firmware	
Güncellemesi	73
NodeMCU-LUA İçin IDE Seçimi	76
ESP8266'nin Harici Beslenmesi	77
Yanıp Sönen LED	78
ESP8266 ile Web Sayfası Oluşturma	80
ESP8266 ile Çoklu Web Sayfası Oluşturma	81
İnternet Üzerinden LED Kontrolü	82
İnternet Kontrollü RGB LED ve Şerit	85
ESP8266 ile Tweet Atma	89
DS18B20 ile Sıcaklık Ölçümü	92
Bulutla Veri Yollama	94
DS18B02 ile Oda Sıcaklığını Buluta Yükleme	96
Mail Bildirimli Hırsız Alarmı	99
TCP Server Üzerinden Chat Yapma	102
Neler Öğrendik?	104

### Bu Bölümde

ESP8266'yı AT komutları ve Arduino IDE'si ile kullanabileceğimiz gibi, hiç Arduino olmadan sadece ESP8266'ya kod atarak da çalıştırabiliriz. Böylece Arduino'ya ihtiyaç duymadan yapmak istediğimiz istediklerimizi ESP8266'yı bir mikrokontrolcü olarak gerçekleştirebiliriz.

## Giriş

ESP8266'nın Arduinoya ihtiyaç duymadan LUA programlama dilinde çalışması için NodeMCU platformu geliştirilmiştir. Bu platform ücretsiz olarak internet üzerinde paylaşılmıştır. Bu platformun çalışması için öncelikle ESP8266'ya bu platform yüklenmelidir. Platform sayesinde, ESP8266 üzerinde bulunan pinler giriş ve çıkış olarak kullanılabilir. Bu platform pinlerin sadece I/O olarak kullanılmasından daha fazlasını, ADC, PWM, UART... gibi özellikler ile de kullanılmasını sağlamaktadır. Platform Lua programlama dilinde çalıştığı için öncelikle bu programlama dilini öğrenmemiz gerekir. Dil öğrenilmesi en kolay dillerden birisidir. Bu dile NodeMCU platformu ile ESP8266 için özelleştirilmiş yeni fonksiyonlar da eklenmektedir.

**NodeMCU'nun resmi web sayfası:**

[www.nodemcu.com/index\\_en.html](http://www.nodemcu.com/index_en.html)

## LUA PROGRAMLAMAYA Giriş

ESP8266 yeni programlarımızda Lua programlama dilinde programlayacağız. Bu dil ile ilk defa tanışanlar için kod yapısı biraz farklı gelse de aslında dil Arduino dili kadar kolaydır.

### Yorum Bloğu

Diğer programlama dillerinden farklı olarak Lua'da yazılan koda açıklama eklemek için "--" ifadesini kullanılır. Bu çift tire işaretinden sonra yazılan ifadeler ESP8266 tarafından işleme alınmayacaktır.

### Değişken Tanımlama

Lua dilinde kullanacağımız değişkenin türünü belirtmeye gerek yoktur. String formatında bir değişken tanımlamak için `a = "mesaj"`, tam sayı formatında bir değişken tanımlamak için `i = 5` şeklinde tanımlamalar yapılabilir

### Ekrana Yazdırma

Bulduğumuz sonuçları ve mesajlarımızı ekrana yazdırmak için `print()` fonksiyonu kullanılır.

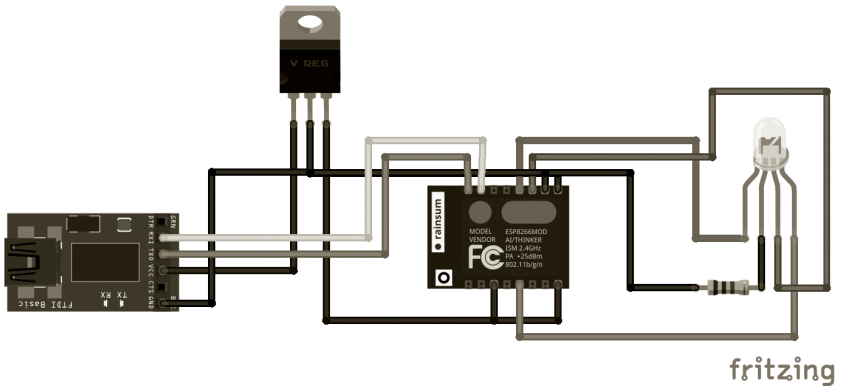
#### Kod

```
i = 10
print("i değişkeninin degeri")
print(i)
```





ESP8266 (ESP-07)	Devre
GND	GND (LED) ve GND (FTDI)
GPI00	Kırmızı (LED)
GPI02	Mavi (LED)
GPI014	Yeşil (LED)
GPI015	GND
VCC	3.3 Volt (regülatör)
CH_PD	3.3 Volt (regülatör)
Tx	Tx (FTDI)
Rx	Rx (FTDI)



### Kod

```

led1 = 3
led2 = 4
led3 = 5

function baglan()
  wifi.setmode(wifi.STATION) -- aga baglanti için ayarlandı
  wifi.sta.config ( " WIFI_ADI " , " WIFI_SIFRE " ) -- internet agina katil
  print(wifi.sta.getip()) -- Aga baglanti kurulduysa IP bilgilerini yazdır
end
gpio.mode(led1,gpio.OUTPUT)
gpio.mode(led2,gpio.OUTPUT)
gpio.mode(led3,gpio.OUTPUT)

pwm.setup(led1,500,1023)
pwm.setup(led2,500,1023)

```

## DS18B20 İLE SICAKLIK ÖLÇÜMÜ

Daha önceki uygulamalarımızda LM35 sıcaklık sensörünün çıkışındaki analog veriyi okuyarak sıcaklık bilgisini almıştık. ESP8266'nın bazı modüllerinde (örneğin: ESP-01) analog pini bulunmadığı için bu uygulamamızda LM35 yerine DS18B20 sıcaklık sensörünü kullanacağız. Bu sensör tek kablo üzerinden dijital olarak haberleşme yapmaktadır.

NodeMCU'nun bize sağladığı kolaylıklardan birisi de `ow` nesnesidir. Bu nesne DS18B20 gibi tek kablo üzerinden haberleşen cihazların kontrolünü sağlamaktadır. Öncelikle `ow.setup(pin)` metodu ile sensörün bağlı olduğu pini belirtmeliyiz. Daha sonra sensörün adresini taramalıyız. Sensörün adresi bulunduktan sonra sıcaklık verisi ölçülmeye başlanabilir. Uygulamada `t1` değişkeni sıcaklığın tam sayı kısmını, `t2` değişkeni ise sıcaklığın virgülden sonraki kısmını tutmaktadır.

Devre bağlantısı aşağıdaki resimde gösterilmiştir. Sensör devreden uzaklaştığında hatta kayıplar meydana gelmektedir. Bu yüzden 4.7 K ohm değerindeki direnç pull-up olarak devreye bağlanmıştır.

