



SAKARYA ÜNİVERSİTESİ
Bilgisayar ve Bilişim Bilimleri Fakültesi
Bilgisayar Mühendisliği Bölümü

BSM 313

NESNELERİN İNTERNETİ VE UYGULAMALARI

(Internet of Things (IoT) and Applications)

NESNELERİN İNTERNETİ UYGULAMALARI GELİŞTİRMEK İÇİN
AÇIK KAYNAKLI GÖMÜLÜ SİSTEM

Arduino – NodeMCU

Prof. Dr. Cüneyt BAYILMIŞ





Arduino Açık Kaynaklı Platform

- ❑ Arduino, donanım ve yazılımın kolay kullanımına imkan tanıyan açık kaynaklı bir platformdur.
- ❑ Öne çıkan özellikleri:
 - Açık kaynaklı bir platform olup, Windows, Linux, MAC gibi farklı platformlarda çalışır.
 - Birçok sensör vb. ek donanım ile uyumluluk ve çalışabilirlik desteğine sahiptir.
 - Farklı protokol ve donanımların kolay kullanımına yönelik zengin kütüphane desteği sunar.
 - Performans ve maliyet açısından uygundur.
 - Yaygın kullanıcı sayısı nedeniyle geliştirme süreçlerinde topluluk desteğine sahiptir.
- ❑ Farklı ihtiyaçlara yönelik geliştirilmiş, mikrodenetleyici, port (pin) sayısı, boyut, çalışma gerilimi gibi farklı özelliklere sahip birçok Arduino çeşidi bulunmaktadır.
 - Arduino Uno (Başlangıç seviyesi için önerilen, en temel ve bilinen türüdür)
 - Arduino Mega (Port sayısı fazladır)
 - Arduino Nano
 - Arduino Lilypad
- ❑ Arduino donanımları çoğunlukla Atmel AVR mikrodenetleyicilerini kullanmaktadır.
- ❑ IoT uygulamaları geliştirmek amacıyla ağ ve internet bağlantıları için ESP8266 WiFi modül gibi eklentilere sahip, Arduino temelli **NodeMCU**, WeMOS gibi donanımlarda mevcuttur.



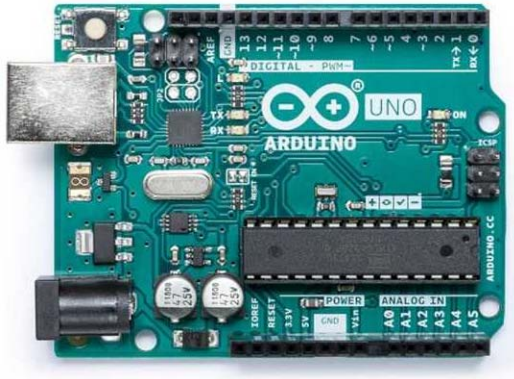
www.arduino.cc adresinden ayrıntılı bilgiye ulaşabilir ve ücretsiz olarak işletim sisteminize uygun IDE indirebilirsiniz.



Arduino Çeşitleri

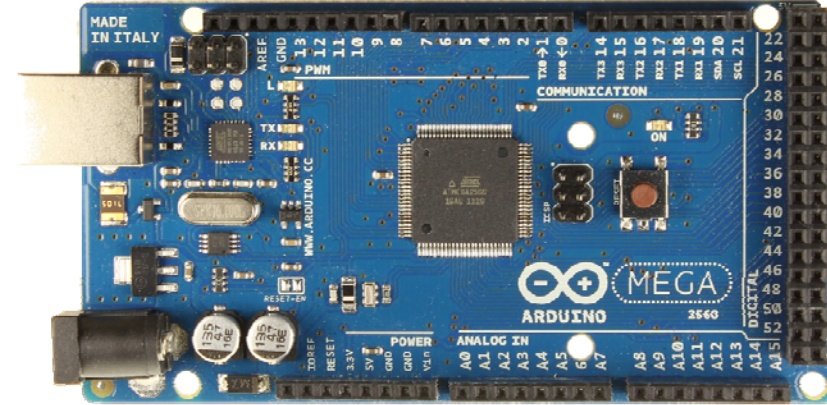
□ Uno

- Mikrodenetleyici: Atmega328
- Bellek: 32K Flash, 2K SRAM, 1K EEPROM
- Çalışma gerilimi 5v
- Giriş/Çıkış Pin Sayısı: 20 (14 Dijital, 6 Analog)
- Çalışma Frekansı: 16 MHz
- Boyut: 68,6 x 53,4 mm



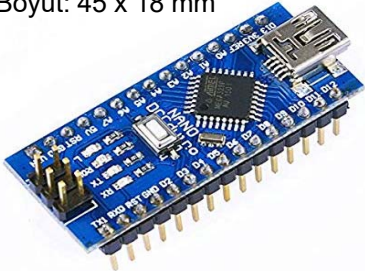
□ Mega

- Mikrodenetleyici: Atmega2560
- Bellek: 256K Flash, 8K SRAM, 4K EEPROM
- Çalışma gerilimi 5v
- Giriş/Çıkış Pin Sayısı: 70 (54 Dijital, 16 Analog)
- Çalışma Frekansı: 16 MHz
- Boyut: 101,6 x 53,4 mm



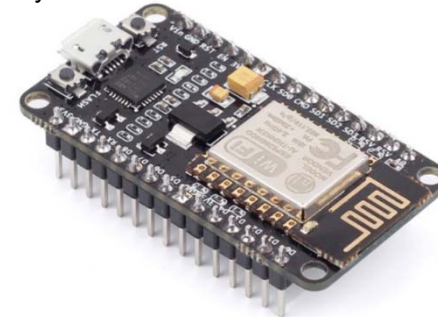
□ Nano

- Mikrodenetleyici: Atmega328
- Bellek: 32K Flash, 2K SRAM, 1K EEPROM
- Çalışma gerilimi 5v
- Giriş/Çıkış Pin Sayısı: 22 (14 Dijital, 8 Analog)
- Çalışma Frekansı: 16 MHz
- Boyut: 45 x 18 mm



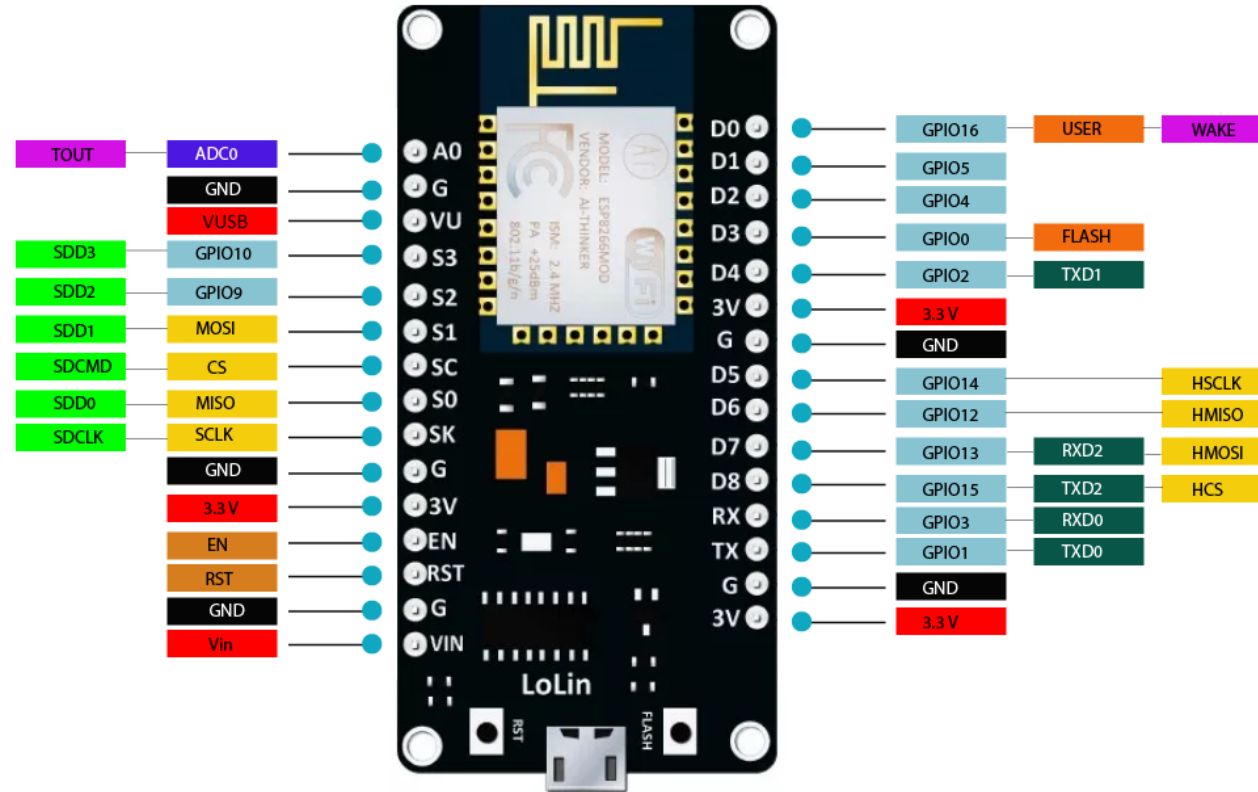
□ NodeMCU

- ESP8266 Wi-Fi modülü üzerine kurgulanmış, Lua script kullanan açık kaynaklı yerleşik yazılım ve geliştirme kitidir.
- Boyut: 48 x 24 mm



Projeleriniz için kart
seçiminde nelere dikkat
etmelisiniz!

ESP8266 WiFi Modülüne Sahip NodeMCU



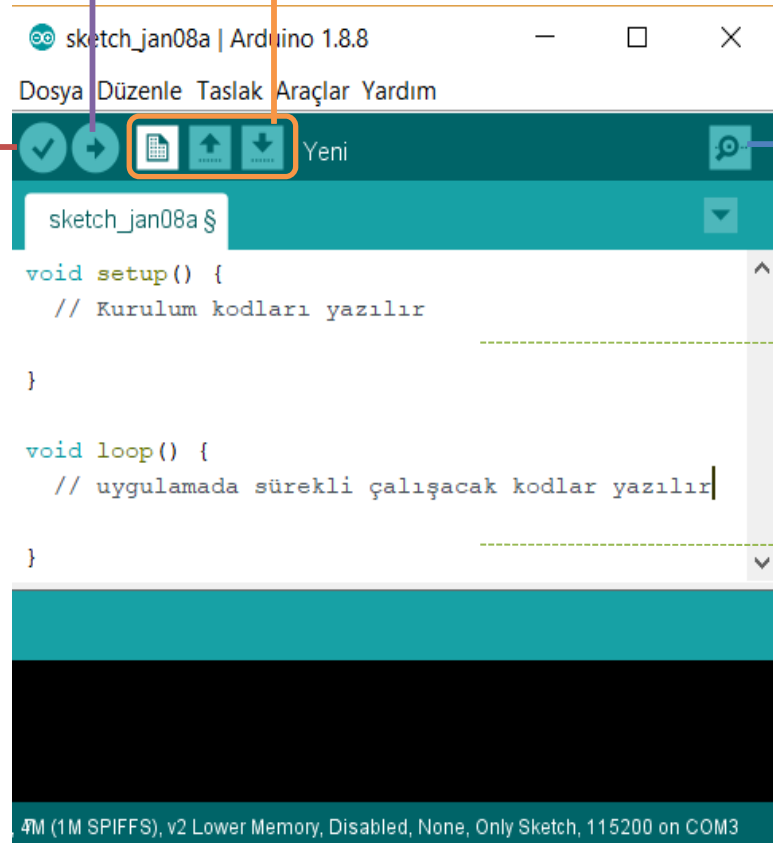
CPU (ESP8266) 32 bit	80MHz (varsayılan) veya 160MHz
Bellek	32K RAM, 512K Flash
Wireless Standart	802.11 b/g/n
Haberleşme Protokol	SPI, I2C, UART
TCP/IP Protokol Yığını	Var

Güç	3.3 VDC
Geçerli Akım Tüketimi	10 uA – 170 mA
GPIO	16 Pin (PWM pin desteği)
Analog Pin	1 adet A0 (10 bit ADC)



Arduino IDE Ortamı

- ❑ Arduino geliştirme ortamı (IDE), Arduino bootloader (Optiboot), Arduino kütüphaneleri, AVR Dude (Arduino üzerindeki mikrodenetleyici programlayan yazılım) ve derleyiciden (AVR-GCC) oluşur.
- ❑ Farklı işletim sistemlerinde çalışabilir. Programlama için C dili kullanılır. Geniş kütüphane desteğine sahiptir.
- ❑ Arduino IDE dışında Eclipse AVR, Arduino for Visual Studio gibi farklı editörler ile de program geliştirilebilir.



Bağlı Arduino karta kodu yükle (Load code to connected Arduino board)

Yeni, mevcut dosya açma (New, open existing file)

Derle (Compile)

Seri Port Ekranı Açma (Open Serial Port Monitor)

Dosya menüsünde; yeni dosya oluşturma, açma, kaydetme gibi temel dosyalama işlemleri gerçekleştirilir.

Düzenle menüsünde; kopyalama, yapıştırma gibi kod bloğu üzerindeki temel işlemler gerçekleştirilir.

Taslak menüsü; kodu derleme, yükleme, kütüphane ekleme gibi işlemleri içerir.

Araçlar menüsü; kodun yükleneceği Arduino kart türü ve bağlı olunan sanal seri port seçimi gibi işlemleri içerir.

setup () fonksiyonu giriş/çıkış pin tanımlama, seri port haberleşme hızının belirlenmesi gibi başlangıç/kurulum ayarlarına ait kod parçalarının içeren **ve başlangıçta 1 kez çalışan fonksiyondur.**

loop () fonksiyonu bir amaca yönelik olarak sürekli çalışan kod parçalarının içeren, **sürekli çalışan ana fonksiyondur.**

www.arduino.cc adresinden ücretsiz olarak işletim sisteminize uygun IDE indirebilirsiniz.

NodeMCU Programlama İçin

Arduino Komutlar Genel Bakış

- ❑ C programlama diline ait karar, döngü vb. tüm yapıları kullanabilirsiniz.
- ❑ Ayrıca, aşağıda da belirtilen Arduino'ya özgü bir takım tanımlamalar ve yapılarda mevcuttur.
- ❑ **Seri İletişim Komutları:**
 - `Serial.begin (9600);` `//setup fonksiyonu içerisinde seri iletişim baud hızını ayarlar 1200, 2400, 9600, 14400...`
 - `Serial.print ("merhaba");` `// seri port ekranına yazmak (seri porttan üzerinden göndermek) için`
 - `Serial.println ("merhaba");` `// seri port ekranına merhaba yaz ve bir alt satıra geç`
 - `Serial.end ();` `// seri iletişim sonlandırır`
 - `Serial.available ();` `// seri porttan okunacak veri olup olmadığının kontrolü`
 - `Serial.read ();` `// seri porttan veri okuma`
 - Arduino Mega gibi birden fazla seri port desteği olan kartlarda seri portlar sırasıyla `Serial`, `Serial1`, `Serial2` şeklinde tanımlanır.
- ❑ **Gecikme fonksiyonu:**
 - `delay (sure değeri);` `// mili saniye cinsinden gecikme oluşturur.`
 - `delay (100);` `// 100 ms`

NodeMCU Programlama İçin

Arduino Komutlar Genel Bakış

❑ Dijital Pin Giriş/Çıkış olarak ayarlanması:

- `pinMode` (pin no, `OUTPUT`); // setup fonksiyonu içerisinde ilgili pin **çıkış** olarak ayarlandı (örneğin LED bağlı)
- `pinMode` (10, `INPUT`); // setup fonksiyonu içerisinde 10 nolu pin **giriş** olarak ayarlandı (örneğin buton bağlı)

❑ Dijital Pin değer okuma ve yazma:

- `deger = digitalRead` (pin); // İlgili dijital pindeki değeri oku ve deger değişkenine aktar
- `digitalWrite` (pin no, `HIGH`); // İlgili pin lojik 1 olarak ayarlandı (örneğin LED'in anodu bağlı ise yandı)
- `digitalWrite` (10, `LOW`); // 10 nolu pin lojik 0 olarak ayarlandı (örneğin LED'in anodu bağlı ise söndü)

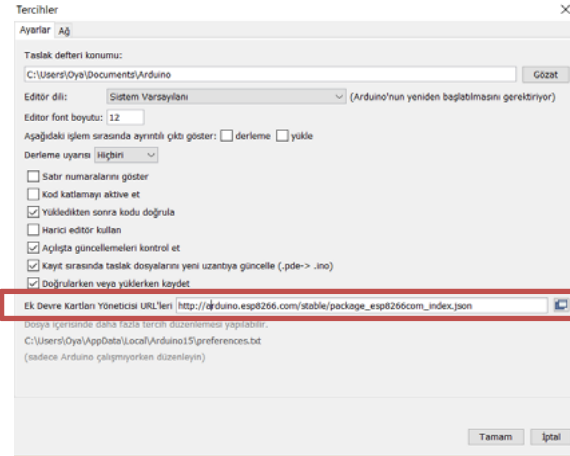
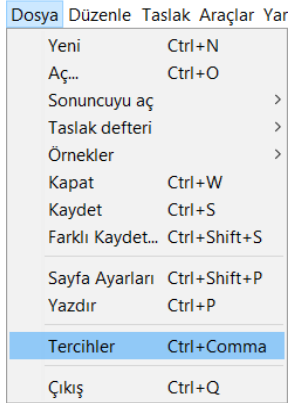
❑ Analog Pin değer okuma ve yazma:

- `deger = analogRead` (pin); // İlgili analog pindeki değeri oku ve deger değişkenine aktar (A0 analog pin 0)
- `analogWrite` (pin, deger); // İlgili analog pin'e deger değişkenindeki değer aktarılır

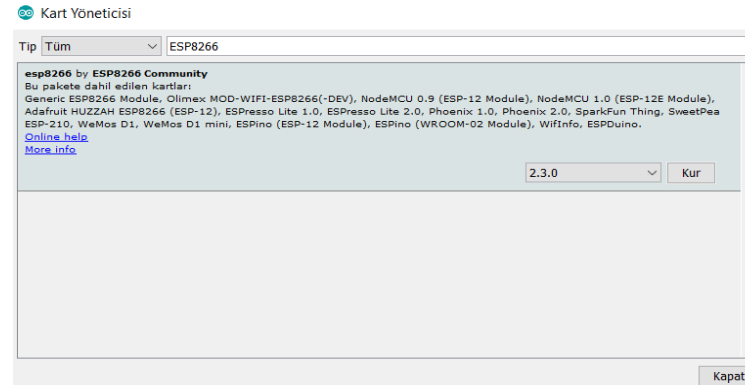


NodeMCU ile Uygulama Geliştirmek

- 1 ESP8266 Wi-Fi sahip NodeMCU vb. kartların Arduino IDE ortamında tanımlanması
- ❑ **Dosya > Tercihler** sekmesindeki ekranda “**Ek Devre Kartları Yöneticisi URL’leri**” kutusuna aşağıda verilen linki ekleyiniz.
http://arduino.esp8266.com/stable/package_esp8266com_index.json



- ❑ ESP8266 kütüphanelerini eklemek için Arduino IDE’de **Araçlar > Kart > Kart Yöneticisi** ekranından ESP8266 aratıp, kurunuz.



- 2 Kurulum sonrasında **Araçlar > Kart > NodeMCU 1.0 (ESP-12E Module)** seçeneğini olarak kartınızı işaretlemeli ve **Araçlar > Port** seçeneğinden sanal COM portunuzu seçmelisiniz.



NodeMCU ile Uygulama Geliştirmek

③ Arduino IDE ortamında **Dosya > Yeni** sekmesinden kod yazım sayfası açınız.

❑ **Blink** uygulaması (ESP8266 modülü üzerindeki mavi LED'in GPIO 2 (D4) – !!! farklı kartlarda pin numarası değişebilir)

The screenshot shows the Arduino IDE interface with the Blink code loaded. The code is as follows:

```
void setup() {  
  // put your setup code here, to run once:  
  Serial.begin(9600); // NodeMCU ile bilgisayar seri port hızı  
  pinMode (D4, OUTPUT); // GPIO2 nolu dijital pin (D4) çıkış modu (NodeMCU Lolin)  
  // NodeMCU Amica için GPIO16 - D0  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  digitalWrite (2, HIGH); // GPIO2 (D4) lojik 1  
  Serial.println("LED Yandı");  
  delay (500); // 500 ms gecikme  
  digitalWrite (D4, LOW); // GPIO2 (D4) lojik 0  
  Serial.println("LED Söndü");  
  delay (500); // 500 ms gecikme  
}
```

The serial monitor (COM4) shows the output of the code, alternating between "LED Söndü" and "LED Yandı" every 500ms.

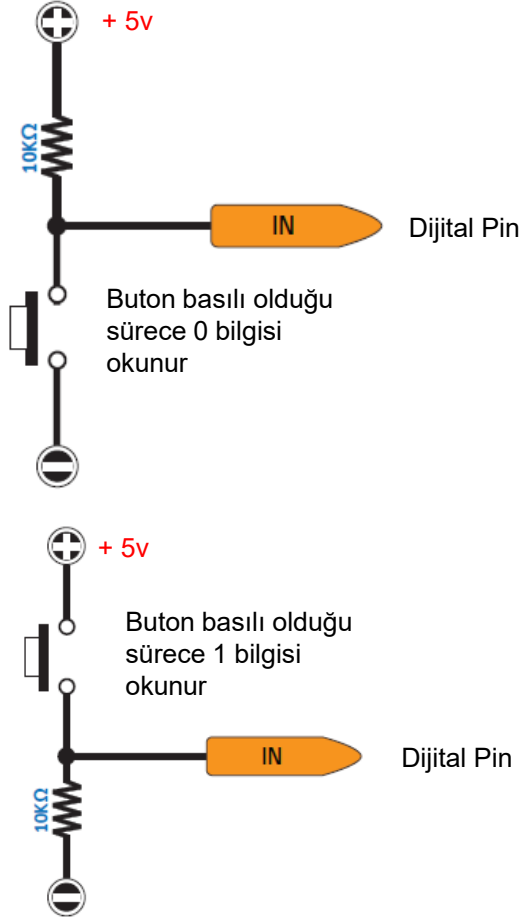
Annotations in the image:

- A red arrow points to the "Yeni" (New) button in the top toolbar, with the text "Bağlı Arduino karta kodu yükle" (Load code to connected Arduino board).
- A blue arrow points to the "Yükle" (Upload) button in the top toolbar, with the text "Yüklemeden sonra program çalışırken seri port ekranını görmek için tıkla" (Click to see the serial port screen while the program is running after uploading).

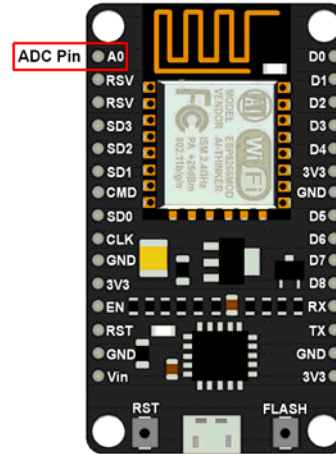
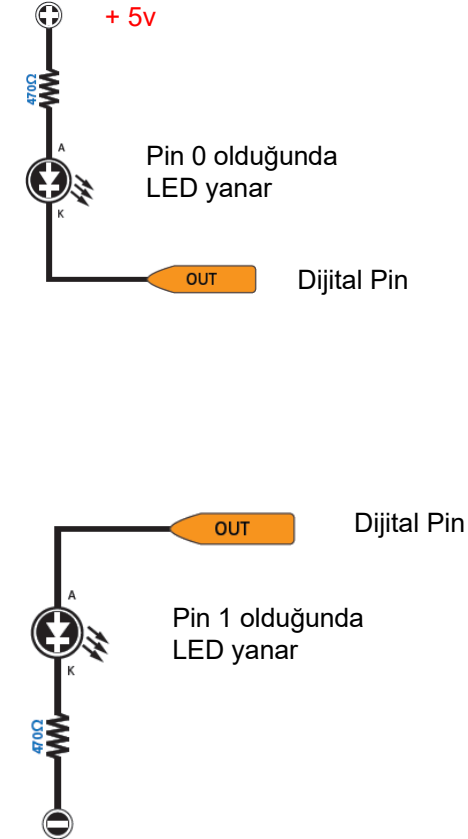
❑ Karta yükleme işlemi tamamlandıktan sonra, yüklenen kodu çalıştırmak için kart üzerindeki Reset (RST) tuşuna basınız.

NodeMcu Dijital Pin Giriş/Çıkış

❑ Dijital pin giriş (**input**) buton bağlantı örneği.



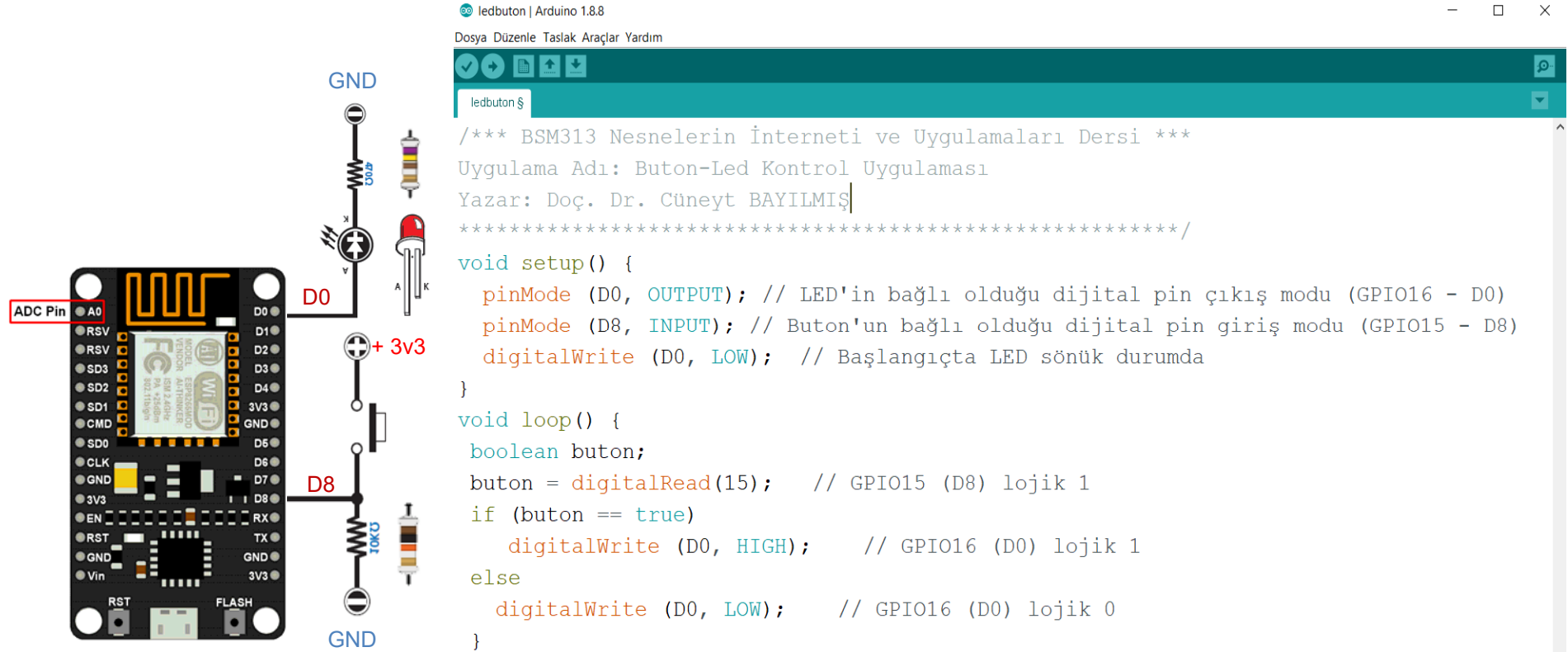
❑ Dijital pin çıkış (**output**) LED bağlantı örneği.



Arduino IDE ortamında **setup ()** fonksiyonu içerisinde, pinin kullanım şekli tanımlanmalıdır.
Giriş pini **pinMode** (pin no, **INPUT**); Çıkış pini de **pinMode** (pin no, **OUTPUT**);

Arduino Uygulama 2: Buton ile LED Kontrolü

- Buton basılı olduğu sürece (push buton) LED'i yakan uygulama



The image shows a circuit diagram and the corresponding Arduino code for a push button LED control application. The circuit diagram on the left illustrates the connection of an Arduino Uno to a push button and an LED. The push button is connected to digital pin D8 (input) and digital pin D0 (output). The LED is connected to digital pin D0 (output) and ground. The push button is also connected to a 3V3 supply and ground. The code on the right is written in the Arduino IDE and shows the setup and loop functions. The setup function configures pin D0 as an output and pin D8 as an input. The loop function reads the state of the push button (D8) and controls the LED (D0) accordingly. If the button is pressed (D8 is HIGH), the LED is turned on (D0 is HIGH). If the button is not pressed (D8 is LOW), the LED is turned off (D0 is LOW).

Devre Bağlantı Şeması

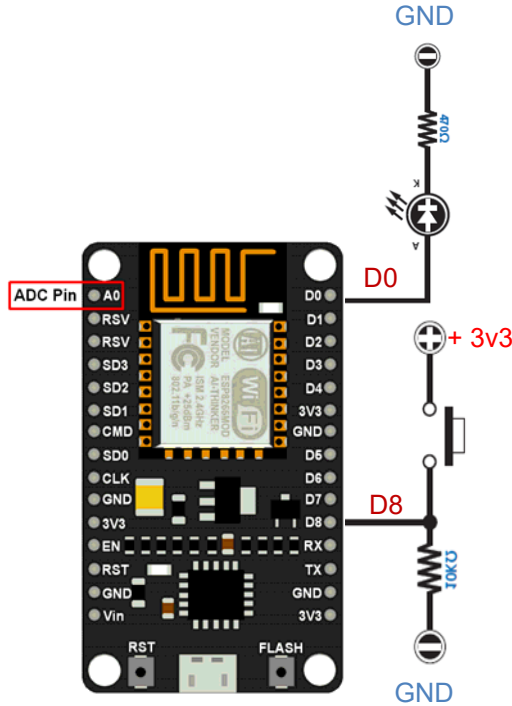
```
/** BSM313 Nesnelerin İnterneti ve Uygulamaları Dersi **  
Uygulama Adı: Buton-Led Kontrol Uygulaması  
Yazar: Doç. Dr. Cüneyt BAYILMIŞ  
*****/  
void setup() {  
  pinMode (D0, OUTPUT); // LED'in bağlı olduğu dijital pin çıkış modu (GPIO16 - D0)  
  pinMode (D8, INPUT); // Buton'un bağlı olduğu dijital pin giriş modu (GPIO15 - D8)  
  digitalWrite (D0, LOW); // Başlangıçta LED sönmük durumda  
}  
void loop() {  
  boolean buton;  
  buton = digitalRead(15); // GPIO15 (D8) lojik 1  
  if (buton == true)  
    digitalWrite (D0, HIGH); // GPIO16 (D0) lojik 1  
  else  
    digitalWrite (D0, LOW); // GPIO16 (D0) lojik 0  
}
```

- 1 Push butonlarda sıçrama (bouncing) olarak adlandırılan basılı olduğu sürece birden fazla işlem yapma durumu yazılımsal olarak nasıl düzeltilebilir.
- 2 Devredeki butonun, switch buton olarak çalışması için kod üzerinde gerekli düzenlemeleri yapınız.



Arduino Uygulama 3: Toggle Led Kontrolü

- ❑ D8 ucuna bağlı butona her basıldığında, D0 ucundaki LED durum değiştirecek.

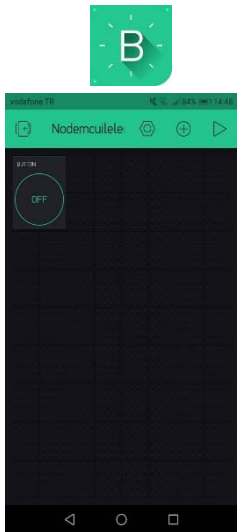


Devre Bağlantı Şeması

```
boolean oncekiDurum= LOW;
boolean sonDurum= LOW;
boolean ledDurum = LOW;
void setup() {
    pinMode (D0, OUTPUT); // LED'in bağlı olduğu dijital pin çıkış modu (GPIO16 - D0)
    pinMode (D8, INPUT); // Buton'un bağlı olduğu dijital pin giriş modu (GPIO15 - D8)
    digitalWrite (D0, LOW); // Başlangıçta LED sönük durumda
}
void loop() {
    sonDurum = digitalRead(D8);
    if (oncekiDurum != sonDurum ) {
        if (sonDurum == HIGH) {
            ledDurum = digitalRead(D0);
            ledDurum = !ledDurum;
            digitalWrite (D0, ledDurum);
        }
        oncekiDurum = sonDurum;
    }
}
```

IoT Uygulama 1: Blynk IoT Mobil Bulut Platformu ile Led Kontrolü

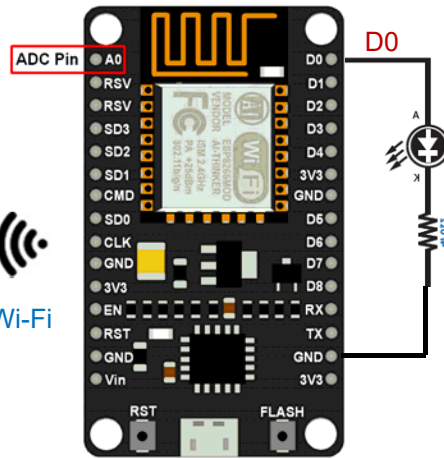
- ❑ NodeMcu IoT cihazına bağlı bir LED'in Blynk mobil IoT Bulut Platformu aracılığı ile AÇ/KAPAT kontrol işlemi gerçekleştirilecektir.
- ❑ Blynk (blynk.io) donanım bilgisi ihtiyacı olmadan, mobil uygulama aracılığıyla internet üzerinden cihaz kontrolüne izin veren bir IoT bulut platformudur.
- ❑ Player Store'dan [Blynk Uygulaması](#) indirilmeli ve Arduino IDE ortamı için [blynk](#) kütüphanesi kurulmalıdır.



Blynk IoT Mobil Bulut Platform



Wi-Fi



Devre Bağlantı Şeması

```
/*
*****
BSM313 Nesnelerin İnterneti ve Uygulamaları Dersi
Doç. Dr. Cüneyt BAYILMIŞ
Blynk Genel Uygulama | Arduino IDE Kodları
*****
*/

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

char auth[] = "BlynkAuthToken"; // Blynk uygulaması tarafından, mailinize gelen token key
char ssid[] = "KablosuzAğAdı"; // Bağlantı yapacağınız Wi-Fi adı
char pass[] = "KablosuzAğSifresi"; // Wi-Fi şifreniz

void setup()
{
  Serial.begin(9600);
  Blynk.begin(auth, ssid, pass);
}

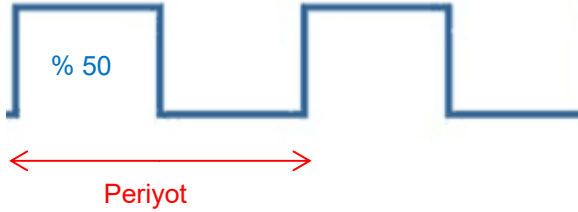
void loop()
{
  Blynk.run();
}
```



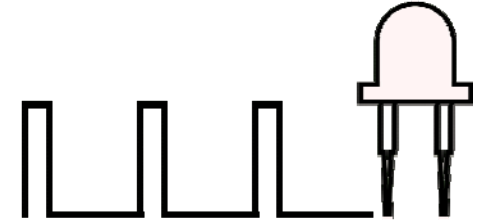
Blynk mobil IoT bulut platformunun kurulumu, kullanımı ve uygulamanın ayrıntılı aşamaları için [Blynk Led Kontrol Uygulama Föyünü](#) inceleyiniz.

NodeMCU Dijital Pin PWM Desteği

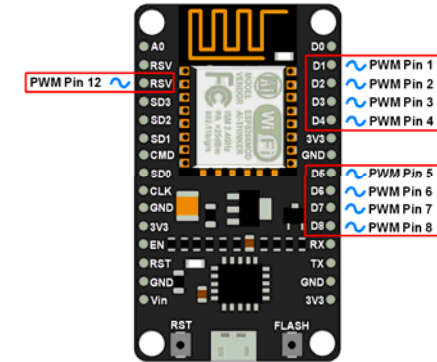
- ❑ NodeMCU, dijital pinleri/uçları (*DO Hariç*) Darbe Genişlik Modülasyon (*Pulse Width Modulation, PWM*) özelliği sunar.
- ❑ PWM, sabit dalga frekansında, darbe genişliğinin (periyot içerisinde 1 olma süresinin) değiştirildiği bir tekniktir.
- ❑ Haberleşme alanında kullanılan bir teknik olmasına karşın, uygulamalarda LED parlaklığı (dimmer), DC Motor hız kontrol gibi işlemlerde yaygın olarak kullanılır.
- ❑ Bir darbe periyodu, lojik 1 ve lojik 0 kısımlarından oluşur. Lojik 1 olma süresi *görev saykılı* (*duty cycle*) olarak adlandırılır.



$$\text{Görev Saykılı} = \frac{\text{Lojik 1 Olma Süresi}}{\text{Periyot}}$$

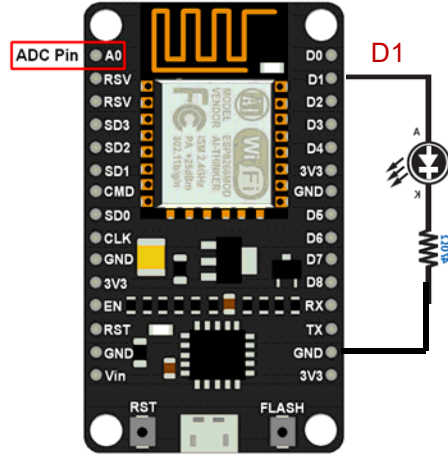


- ❑ Diğer bir deyişle, dijital uçlar üzerinden analog işlem yapmaya imkan tanır. Bu nedenle Arduino'daki analog komutlar kullanılır.
 - Örnek: *analogWrite* (pin, görevsaykılı)
- ❑ Birçok Arduino kartta PWM destekli pin ~ sembolü ile gösterilir.

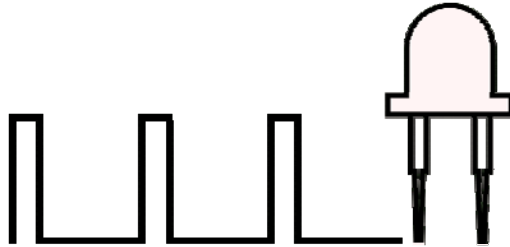


Arduino Uygulama 4: PWM Destekli Pin Kullanımı

- ❑ D1 PWM destekli uca bağlı LED'in parlaklığının değiştirilmesi.



Devre Bağlantı Şeması



Kaynak: <https://www.electronicwings.com/>

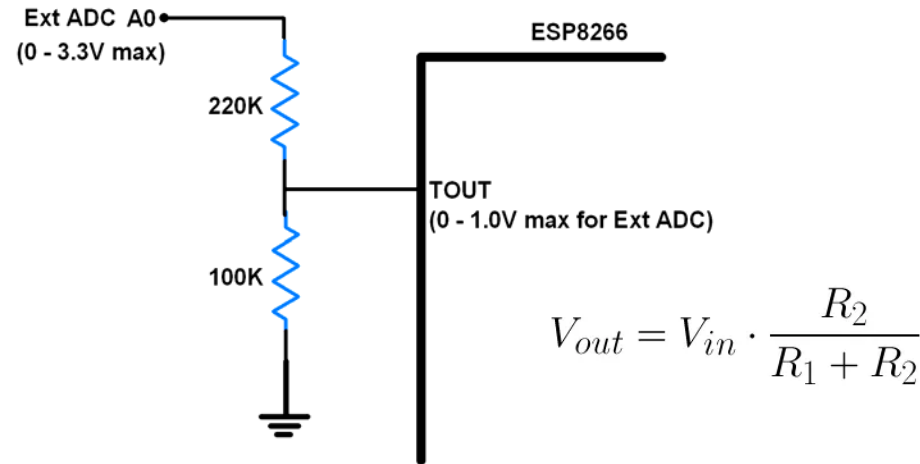
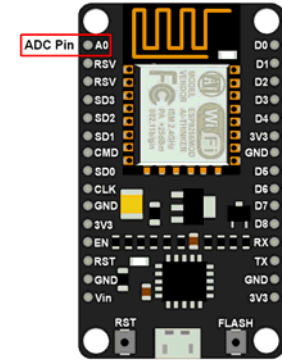
```
/** BSM313 Nesnelerin İnterneti ve Uygulamaları Dersi **  
Uygulama Adı: PWM  
Yazar: Doç. Dr. Cüneyt BAYILMIŞ  
*****/  
uint8_t LED = D1;  
uint16_t gorevSaykili = 0;  
void setup() {  
  pinMode (LED, OUTPUT); // LED GPIO5'e bağı (D1)  
  analogWrite (D1, 0); // Başlangıçta LED sönük durumda  
}  
void loop() {  
  while (gorevSaykili <=1000) {  
    gorevSaykili += 100;  
    analogWrite(LED,gorevSaykili); // LED parlaklığını ayarla  
    delay (500);  
  }  
  gorevSaykili = 0;  
  analogWrite(LED,gorevSaykili); // LED parlaklığını ayarla  
  delay (500);  
}
```



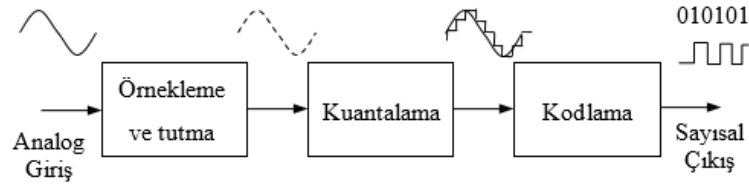
- 1 Buton ile Led Kontrol Uygulamasında olduğu gibi LED'in parlaklığını bir buton ile arttırınız.
- 2 Blynk Led Kontrol Uygulama Föyünde buton yerine **slider** kullanarak yukarıdaki uygulamayı IoT uygulaması şeklinde gerçekleştiriniz.

NodeMCU Analog Pin ve Analog Dijital Dönüştürücü (ADC)

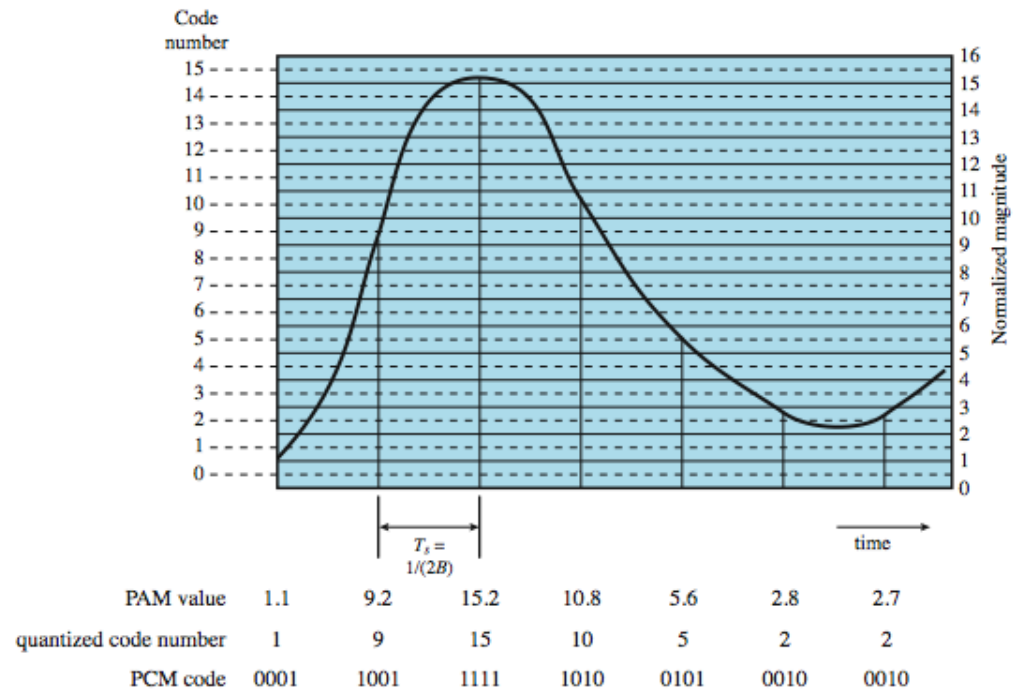
- ❑ Tek kanal (A0) ve 10 bit ADC'ye sahiptir ($2^{10} = 1024$ çözünürlük)
- ❑ Harici cihazlardan analog voltaj okumak için **A0**'ı kullanır.
- ❑ NodeMCU, ESP8266 Wi-Fi modülü üzerindeki ADC kanalı kullanır.
 - Bu nedenle, NodeMCU 3.3v ile çalışmasına karşın, ESP8266 ADC giriş pini voltaj aralığı dahili gerilim bölücü dirençler ile 0-1v aralığına çekilir.



Analog Dijital Çevirici



- ❶ **Örnekleme**; analog bilgi sinyalin örneklenerek darbe dizisi haline dönüştürülür.
- ❷ **Kuantalama**; örnekleme aşamasında elde edilen her bir örnek değer, önceden belirlenmiş seviyelerdeki değerlere yakınlştırılma işlemidir.
- ❸ **Kodlama**; her bir örnek değer, kuantalama seviyesinin bir binary dizisi (kod sözcüğü) ile kodlanmasıdır.



Kaynak: William Stallings, Data Communication and Computer Network



Analog Dijital Çevirici

- ❑ Analog girişi bulunmayan mikrodenetleyicilere harici ADC'ler bağlanabilir.
- ❑ ADC0804 piyasada yaygın olarak kullanılabilen bir 8 bitlik bir ADC'dir. 0-5 V aralığındaki analog sinyalleri ölçebilir. Daha hassas ölçümler için V_{ref} referans sinyali düşürülerek bit başına düşen gerilim düşürülebilir.
- ❑ 8 bit olduğundan 256 durum (0-255) mevcuttur. Ölçüm aralığı (Ör: 0-5 V arası) bu 256 duruma bölünerek durum başına düşen voltaj bulunur.
- ❑ $5\text{ V} = 5000\text{ mV}$ $5000 / 256 = 19,5\text{ mV}$ düşmektedir.
- ❑ Bu değer, 5 V ölçüm aralığı olan 8 bitlik bir ADC için voltaj hassasiyeti olarak ifade edilir. Diğer bir ifade ile sistem ancak 19,5 mV'dan büyük değişiklikleri hisseder.

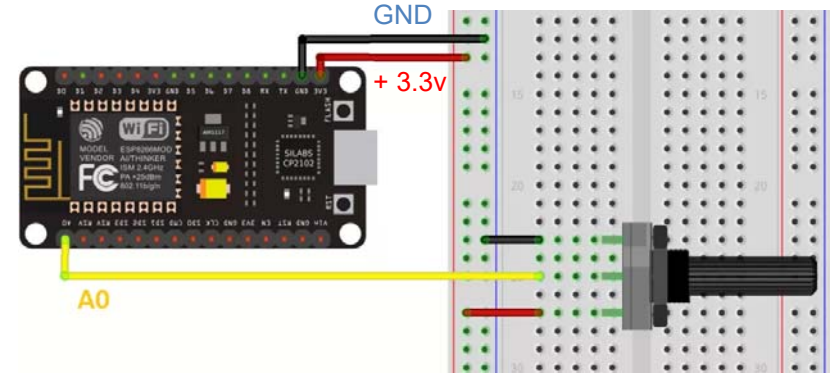
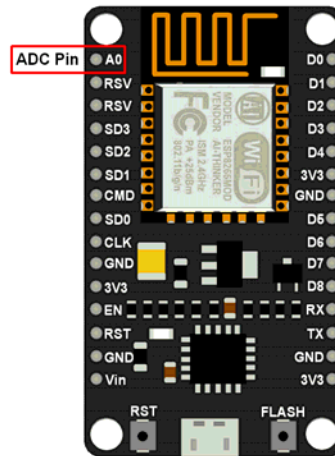
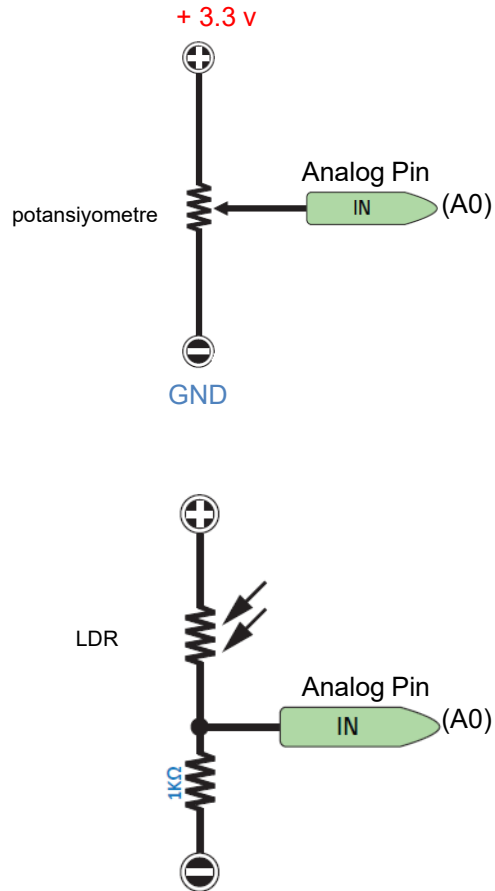
Analog Dijital Çevirici

V_{ref} Değeri ile Hassasiyet Değişimi

- ❑ ADC üzerinde bulunan V_{ref} değerine herhangi bir voltaj bağlanmazsa sistem otomatik olarak 0-5 V aralığını ölçer.
 - ❑ Diğer bir değişle, 0 durumu 0 V, 255 durumu ise 5 V
- ❑ Ancak piyasada bulunan birçok sensör çoğu zaman 0-1 V aralığında değerler üretmektedir. Bu durumda 1-5 V aralığı atıl durumda kalacak ve durumların 4/5 oranı kullanılmayacaktır.
- ❑ Bunu önlemek amacıyla V_{ref} değeri ayarlanarak tam ölçüm aralığı (*full scale*) elde edilebilir.
- ❑ $V_{ref} = 1$ V yapılırsa (*ya da bazı ADC'lerde $V_{ref} / 2$ şeklinde geçer $V_{ref} / 2 = 0,5$ V yapılırsa*), ölçüm aralığı 1000 mV olur ve 0-255 durumu (8 bit için) ile ifade edildiğinde $1000/256 = 3,9$ mV değeri elde edilir.

NodeMCU Analog Pin Kullanımı

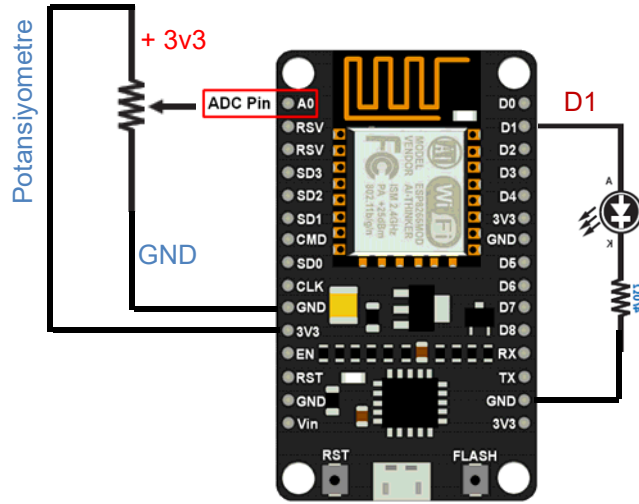
□ Analog pin giriş bağlantı örneği.



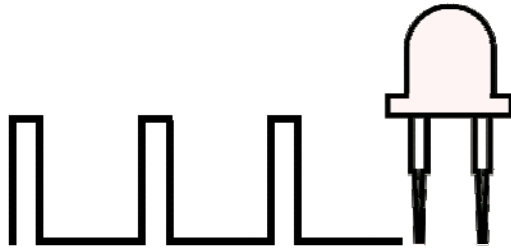
Arduino Uygulama 5: Dimmer Uygulaması

- ❑ Analog girişe bağlı potansiyometre ile D1 PWM destekli uca bağlı LED'in parlaklığının değiştirilmesi.

➤ NodeMCU 10 bit ($2^{10} = 1024$) ADC'ye sahip olduğundan PWM frekansı yaklaşık 1000 Hz alınabilir.



Devre Bağlantı Şeması



Kaynak: <https://www.electronicwings.com/>

```
uint8_t LED = D1;
uint16_t gorevSaykili;
void setup() {
  pinMode (LED, OUTPUT); // LED GPIO5'e bağıl (D1)
  pinMode (A0, INPUT); // Potansiyometre analog 0 ucuna bağıl
  analogWrite (D1, 0); // Başlangıçta LED sönük durumda
}
void loop() {
  gorevSaykili=analogRead(A0); // Potansiyometrenin değerini oku

  if (gorevSaykili > 1023)
    gorevSaykili = 1023; // Periyot sınırlama

  analogWrite(LED,gorevSaykili); // LED parlaklığını ayarla
  delay (250);
}
```

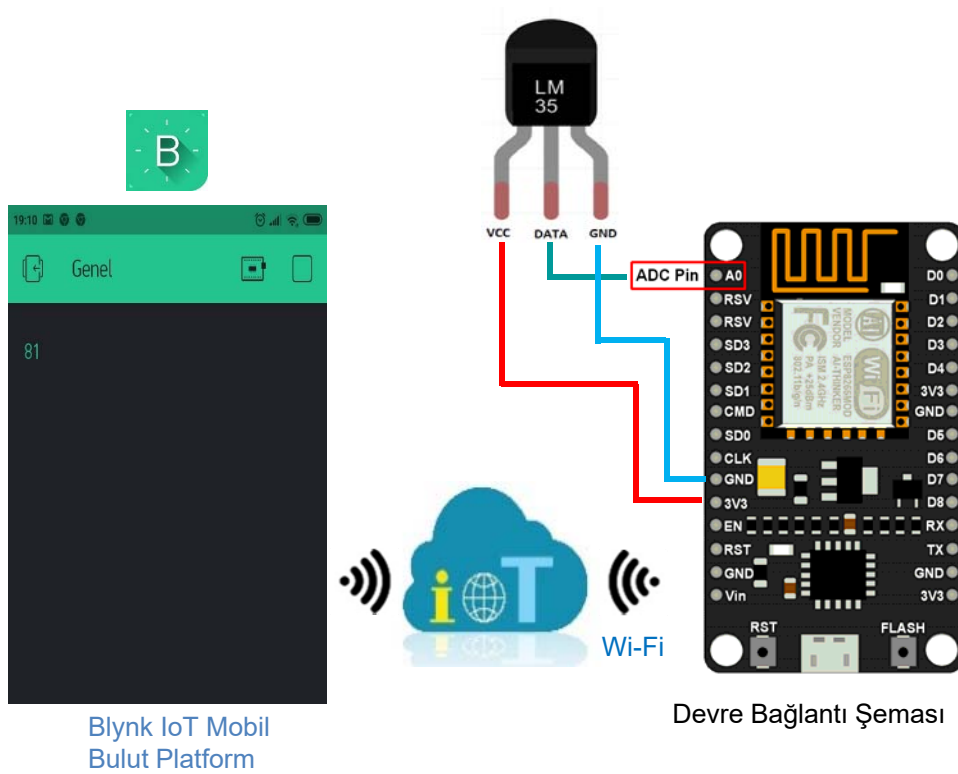


Blynk Led Kontrol Uygulama Föyünde buton yerine slider kullanarak yukarıdaki uygulamayı IoT uygulaması şeklinde gerçekleştiriniz.



IoT Uygulama 2: Blynk IoT Mobil Bulut Platformu ile Sıcaklık Ölçümü

- NodeMcu IoT cihazın, analog A0 ucuna bağlı bir LM35 sıcaklık sensöründen okunan değerin Blynk mobil IoT Bulut Platformu aracılığı ile gösterimi gerçekleştirilecektir.



LM35 sıcaklık sensörü

- 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.

- Örnek:** LM35 sensörü 240 mV ölçüyorsa kaç derece ölçülmüştür.

- $240 \text{ mV} / 10 \text{ mV} = 24^\circ \text{C}$
- 10 bitlik ADC kullanılıyorsa $2^{10} = 1024$ durum yapar.
- Sıcaklık = Ölçülen Değer * $\left(\frac{V_{ref}}{1024}\right) / 10$



IoT Uygulama 1'deki aynı kodu kullanarak, Blynk mobil uygulaması üzerinde sıcaklık değeri için **Labeled Value** göstergesi ekleyiniz ve **analog uç olarak A0** seçiniz.



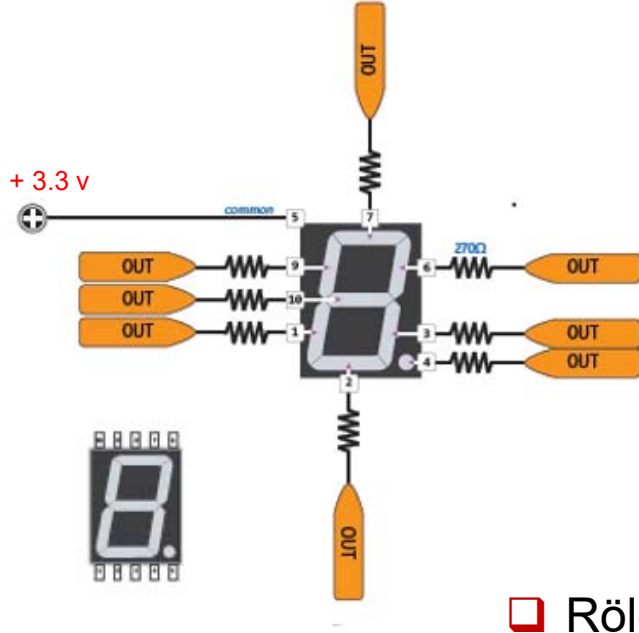
Okuduğunuz sıcaklık değeri gerçek ortam sıcaklık değerinden farklı ise sebepleri ne olabilir? Tartışınız.



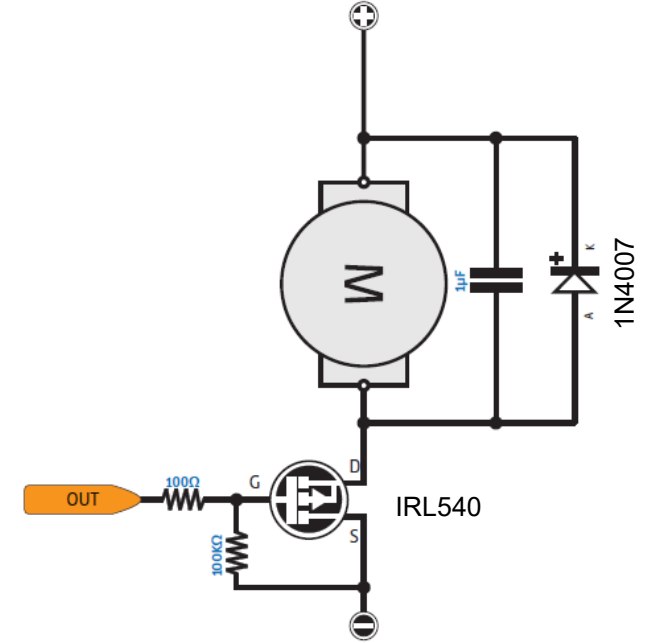
Arduino – NodeMCU

Çevresel Eleman Bağlantı Örnekleri

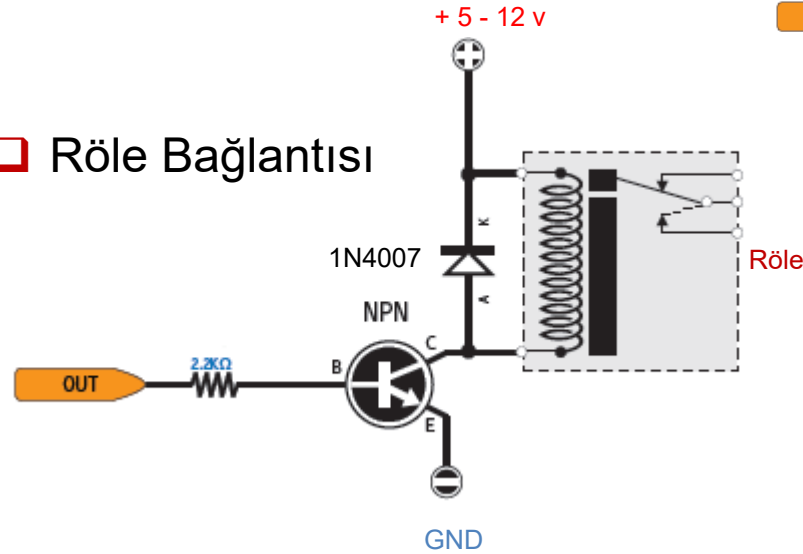
Ortak Anotlu 7 Segment Display



DC Motor Bağlantısı



Röle Bağlantısı



Kaynaklar

❖ Temel Kaynaklar

- Prof. Dr. Cüneyt BAYILMIŞ ve Prof. Dr. Kerem KÜÇÜK, “Nesnelerin İnternet’i: Teori ve Uygulamaları”, Papatya Yayınevi, 2019.

❖ Diğer Kaynaklar

- Handson Technology, “ESP8266 NodeMCU WiFi Devkit”, https://www.handsontec.com/pdf_learn/esp8266-V10.pdf
- ABC: Basic Connections – The Essential Book for Makers by PighiXXX
- <https://www.electronicwings.com/>

