

Açık Kaynak Yazılım ve Açık Donanım ile Nesnelerin İnterneti

Mustafa Karakaplan

karakaplanm@gmail.com

İnönü Üniversitesi, Fen-Edebiyat Fakültesi, Kimya Bölümü, Malatya

Özet

Bu yazıda elektronik cihazların internete bağlanması ve uzaktan kontrolü konusunda bilgiler verilmiştir. Nesnelerin İnterneti kavramı yeni bir çağın eşiğinde olduğumuzun belirtisidir. Ticari olarak oldukça fazla ürün bulunmakla birlikte biz eğitimleri Açık Kaynak Yazılım ve Açık Kaynak Donanım ile gerçekleştireceğiz.

Anahtar Kelimeler: IOT, Nesnelerin İnterneti, Raspberry Pi, ESP8266, Arduino

1. Nesnelerin İnterneti

Neyimiz var neyimiz yok herşeyimizi internete bağlamaya başladık. Anahtarımızı veya cüzdanımızı aramaya gerek kalmıyor. Ocağı açık unututum mu acaba şüphesiyle yoldan dönmek zorunda kalmıyoruz. Gündelik yaşamda kullandığımız teknik araçlar da akıllanıyor. Teknolojinin kölesi oluyoruz serzenişleri boşuna. Zamanın, güvenliğin önemli olduğu bir çağda yaşam ve sağlık kalitesinin artması için gerekli bütün bu gelişmeler.

2. IOT Protokolleri

Nesnelerin İnterneti kavramı içerisinde araçlar birbirleriyle Cihazdan-Cihaza (D2D), Cihazdan-Sunucuya (D2S), Sunucudan-Sunucuya (S2S) olmak üzere 3 farklı şekillerde haberleşme yapılır. Nesneler, HTTP ve UDP başta olma üzere TCP/IP üzerindeki birçok iletişim protokolü ile birbirleriyle haberleşebilir. Ancak farklı cihazlar ve işletim sistemleri arasında haberleşmeye standart getirmek üzere birçok IOT protokolleri de geliştirilmiştir. Bu yazıda basitliği ve yaygın kullanımı nedeniyle sadece HTTP, UDP ve MQTT örnekleri verilecektir.

2.1 MQTT

MQTT (Message Queuing Telemetry Transport), abonelik ve yayıncılık temellerine dayanan telemetry mesajlaşma protokolüdür. Makineler arası haberleşmede kullanılır. Hafif (lightweight) olması nedeniyle tercih edilir ve bir çok platformda kullanılmaktadır.

Subscriber (Abone) birimleri mesaj almak için beklerler ve hangi mesajları alacaklarını başlangıçta bulutta bulunan Broker (Simsar, Aracı) birimine "topic" etiketiyle bildirirler. Publisher (Yayıncı) birimleri ise mesaj yayınlayan birimlerdir. Yayınladıkları mesajı topic bilgisi ile birlikte yayınlarlar.

Bir sistemde birden fazla Publisher, Subscriber ve Broker bulunabilir.

Broker makinada

```
sudo apt install mosquitto
```

Publisher ve Subscriber makinada

```
sudo apt install mosquitto-clients
```

Subscriber i bir topic e üye yapalım.

```
mosquitto_sub -t deneme -d
```

deneme topic li mesajlara üye olduk. -d ile ne olup ne bittiğini görmek için debug mesajlarını da görüntülüyoruz.

Şimdi Publisher makinadan Broker makinaya deneme topic i ile bir mesaj gönderelim.

```
mosquitto_pub -t deneme -m "Hava nemli ve sıcak"
```

2.2 XMPP

Extensible Messaging and Presence Protocol veya orjinal adıyla "Jabber" insanlar arasında text mesaj yoluyla haberleşmeyi sağlar.

2.3 DDS

Data Distribution Service, cihazlar arası veri paylaşımı sağlar.

2.4 AMPQ

Advanced Message Queuing Protocol,

3. IOT için GNU/Linux Araçları

3.1 Git

Git yazılımcılar için çok kullanılan vazgeçilmez sürüm/versiyon kontrol sistemidir.

```
$ git config --global user.name "User"
$ git config --global user.email "user@ornek.com"
```

Bu komutla içinde bulunulan dizin Git deposu haline getirilir. Bu dizin daha sonra git sunucusuna gönderilebilir.

```
$ mkdir Proje1
$ cd Proje1
$ git init
```

Localdeki bir git dizininin kopyasını çıkarmak için;

```
$ git clone /deponunu/dizin/yolu
```

Uzak bir sunucudaki git deposunun kopyasını almak için;

```
$ git clone https://github.com/karakaplanm/iot
```

Bu komutla bir dosyayı veya bir dizini git indeksine ekler. Uzak sunucuya gönderileceklerin içerisinde yer alacak dosya veya dosyaları belirlemiş oluruz.

```
$ touch benioku.txt
$ git add benioku.txt
```

Belirtilen dosya git depomuzdan silinip git indeksinden çıkarılır.

```
$ git rm benioku.txt
```

Commit komutu indekse eklenmiş dosyaları git dizini içindeki .git gizli dizinine ekler. Bu işlem local de gerçekleşir. Eklenen dosyaların uzak sunucuya gönderilmesi için push komutu kullanılmalıdır. Eklenen dosyalarda ne gibi değişiklikler yapıldığı turnak içinde verilir.

```
$ git commit -m "X degiskeni global yapildi"
```

add ve commit işlemi ile gönderilmeye hazır dosyaları uzak sunucuya göndermek için bu komut kullanılmalıdır.

```
$ git push origin master
```

Local git deposunun son durumunu gösterir. Değişim veya yeni ekleme olup olmadığı görülebilir. Uzaktaki depoda değişikliğe uğramış dosyalar veya yeni eklenmiş dosyalar varsa bunlar local e indirilir.

3.2 Minicom

Gömülü sistemlere seri port üzerinden bağlanmak için kullanılan tüm zamanların en popüler terminal programı. Kurulum için;

```
$ sudo apt install minicom
```

Seri porta (COM veya USB) bağlı bir cihaza bağlanmak için;

```
$ sudo minicom -b 9600 -D /dev/ttyUSB0
```

komutunu girmeniz gerekli. Normal kullanıcı olarak bağlanabilmek için

```
sudo chmod 777 /dev/ttyUSB0
```

şeklinde porta erişim haklarını açmalısınız. Bilgisayarınıza seri port üzerinden bağladığınız cihazın bağlanıp bağlanmadığını, bağlandı ise nereye bağlandığını dmesg komutu ile görebilirsiniz.

3.3 adb

Android Debug Bridge (ADB) Android için uygulama geliştirirken kullanılan komut kütüphanesidir. Cihaza USB ile bağlı iken uygulama yükleme ve kaldırma için kullanılır. Android Studio, Eclipse gibi IDE ler ile uygulama geliştirirken arka planda bu komutlar IDE tarafından kullanılır. Ubuntu depolarından adb yi kurmak için;

```
$ sudo apt install adb
```

adb komutuyla Android cihazınıza ulaşabilmek için cihazınızı Debug moduna almanız gereklidir. Bunun için ayarlar, cihazım, build Number a 7 defa dokunmanız gereklidir.

Android cihazını USB kablo ile bilgisayarınıza bağladıktan sonra bilgisayarınızın terminalinden

```
$ adb devices
```

komutu verdiğinizde android ekranında bağlantı izni istenecektir. İzin verdikten sonra artık ulaşabilirsiniz. apk uzantılı bir paketi Android cihaza kurmak için;

```
$ adb install uygulama.apk
```

3.4 fastboot

Fastboot, USB port aracılığıyla Android cihazların dosya sistemine bağlanmak için kullanılır. Cihazlara ROM atmak veya ROM ların kopyasını çıkarmak için kullanılır.

3.5 dd

DD ile sabit diskin, usb belleğin, cd veya dvd nin imajı alınabilir. SD kartlardan boot eden cihazların işletim sistemleri kopya edilebilir veya yenisi dd ile yazılabilir.

3.6 Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino and Genuino hardware to upload programs and communicate with them.

3.7 PlatformIO

<http://platformio.org/>

3.8 MIT App Inventor

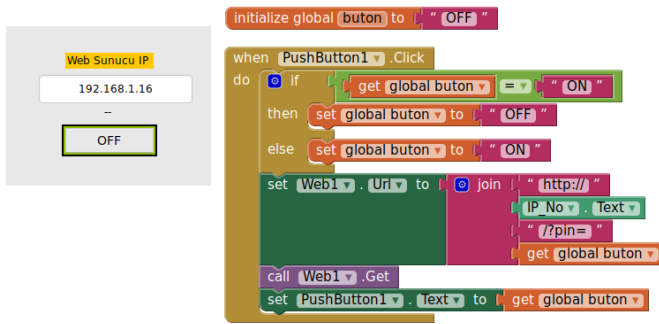


Figure 1: App Inventor

4. Açık Kaynak Donanımlar

Açık Kaynak Donanım donanımın üzerinde çalışılabilir, değiştirilebilir ve aynısından yapıp satılabilir olmasını sağlar. Bir donanımın şemalarının açık olması, çalıştırmak ve yeniden programlamak için kullanılan yazılımların açık kaynaklı olması da çoğu durumda o donanımın açık kaynak donanım olması için yeterlidir. Uluslararası kabul görmüş standartlarda, lisanssız gerektirmeyen haberleşmelerle ve diğer donanımlarla bağlantı yapabilme de açık kaynak donanımın en büyük avantajlarından biridir. Açık Donanım, Açık Kaynak Kültürünün bir uzantısıdır. Donanımı değiştirebilmek, o donanımı atmadan başka bir iş için kullanabilmeyi de sağlar.

4.1 ESP8266

ESP8266 Wifi Serial Transceiver Modül oldukça ekonomik ve kullanışlı bir Wifi modüldür. TCP/IP protokolünü desteklemektedir. ESP8266 üzerinde dahili anten bulunmaktadır. Bu sayede ortamdaki Wifi ağına rahatlıkla bağlanabilmekte, veri paketleri alıp gönderebilmektedir.

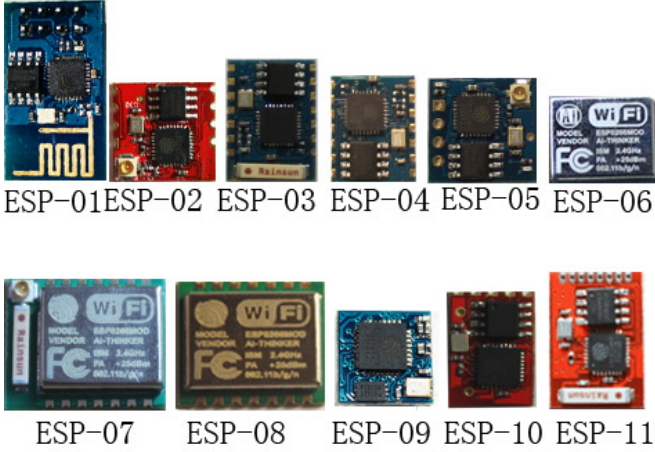


Figure 2: ESP8266

4.2 Raspberry Pi

Raspberry Pi kredi kartı boyutunda mini bir bilgisayar kartıdır. 256 MB da 1GB a kadar RAM a sahip modelleri bulunmaktadır. Mart 2016 da çıkan 3. nesil Raspberry Pi 3 modeli ise 64 bitlik işlemci barındırmaktadır.

Mikrodenetleyicinin yaptığı işler dışında yüksek kapasiteli işlemlere ihtiyaç duyduğunuzda Raspberry Pi kullanmak gerekecektir. Akıllı IOT tarzı cihazlar için uygun bir karttır.

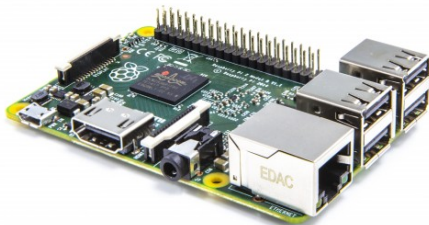


Figure 3: Raspberry Pi

4.3 Arduino

Arduino Processing/Wiring dili ile fiziksel programlama yapabileceğiniz bir mikrodenetleyici-Giriş/Çıkış kartıdır. Arduino tek başına kullanılabileceği gibi bilgisayarlar ile de haberleşebilir. Açık Donanım olması nedeniyle donanımı kendiniz üretebilir veya hazır satın alabilirsiniz.

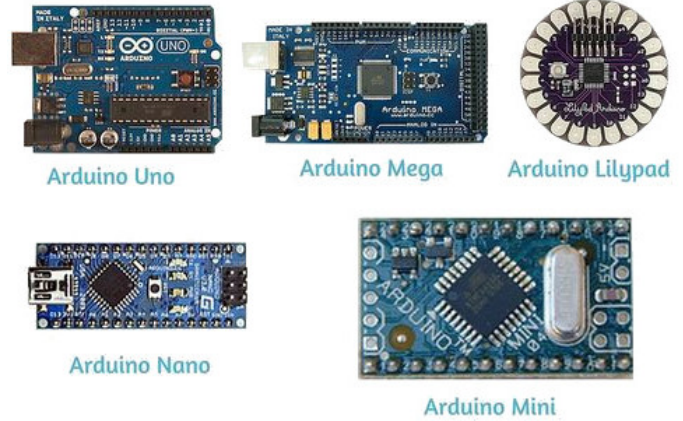


Figure 4: Arduino

4.4 Teensy

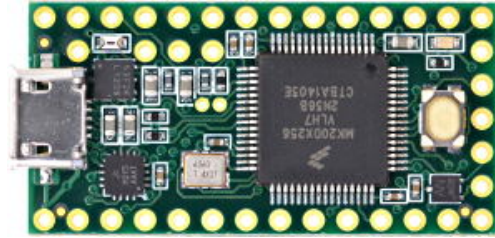


Figure 5: Teensy

Teensy kartı breadboard uyumlu bir geliştirme kartıdır. Karta varsayılan olarak bootloader yüklenmiş olduğu için direk kartın üstündeki USB portu ile programlayabilirsiniz. Harici bir programlayıcıya ihtiyaç yoktur. IDE olarak istediğiniz C program editörünü kullanabileceğiniz gibi Teensyduino harici eklentisini kurarak Arduino IDE'sini kullanabilirsiniz. Teensy üzerindeki işlemci USB'ye erişebilir ve ihtiyacımız olan her türlü USB cihazı taklit edebilir. 32bitlik işlemcisi direk hafıza ulaşımı için çoklu kanal sağlar ve birçok yüksek çözünürlüklü ADC temin eder. Ayrıca I2C dijital audio arayüzü sunar. Bunun yanında her birinin interrupt özeliği olan 4 farklı interval timer ve bir delay timer içerir. Çıkış olarak diğer cihazlar için 3.3V gerilim ve 100mA akım sağlaması da cabası. Teensy 3.2, sahip olduğu güçlü 3.3V regülatörü sayesinde ESP8266 Wifi, WIZ820io Ethernet ya da başka kartları direk besleyebilir.

5. Raspberry Pi İle IOT

5.1 Raspberry Pi Kurulumu

Raspberry İmg Yazma

```
sudo dd bs=4M if=Rasbian.img of=/dev/sdX
```

Image Dosyasını Büyüt

```
dd bs=1M count=4096 if=/dev/zero of=pi.img
$ sudo losetup -f --show raspbian.img
/dev/loop0
$ sudo losetup -f --show pi.img
/dev/loop1
```

loop0 ı loop1 de kopyala

```
sudo dd if=/dev/loop0 of=/dev/loop1 bs=1M
sudo gparted /dev/loop1
```

ile aç. Önce kullanılmayan kısma boş partition aç. Sonra küçült ve ext4 partitionu buyult.

```
sudo losetup -d /dev/loop1
sudo losetup -d /dev/loop0
```

ile loop device ları serbest bırak

5.2 Wireless

```
#/etc/wpa_supplicant/wpa_supplicant.conf
country=GB
ctrl_interface=DIR=/var/run/wpa_supplicant
GROUP=netdev
update_config=1
```

```
network={
    ssid="TCPLink"
    psk="12345678"
}
network={
    ssid="ZoZo"
    psk="zoz12345"
}
```

Kaydettikten sonra sistemin değişikliği algılayıp bağlantı yapması gerekir. ifconfig komutuyla wlan0 ın IP alıp almadığını kontrol ediniz. Eğer bağlantı sağlanmamış ise

```
$ sudo ifdown wlan0
$ sudo ifup wlan0
```

komutları ile wlan0 ı durdurup yeniden aktif ediniz.

5.3 Bluetooth

```
sudo apt-get install bluetooth bluez
sudo reboot
sudo bluetoothctl
```

```
[bluetooth]# agent on
```

```
[bluetooth]# default-agent
```

```
[bluetooth]# pair xx:xx:xx:xx:xx:xx
[bluetooth]# trust xx:xx:xx:xx:xx:xx
```

```
sudo rfcomm bind 0 xx:xx:xx:xx:xx:xx
```

5.4 i2c

sudo raspi-config interfacing Options i2c enable

```
reboot
RPi      OLED
01 3.3v  VDD 2
03 SDA1   SDA 4
05 SCL1   SCK 3
07
09 Ground GND 1
```

```
sudo i2cdetect -y 1
```

```
git clone https://github.com/hallard/ArduiPi_OLED
cd ArduiPi_OLED
sudo ./autogen.sh
sudo make
cd examples
make
sudo ./oled_demo --oled 3
```

```
git clone https://github.com/adafruit/Adafruit_Py
cd Adafruit_Python_SSD1306
sudo python setup.py install
cd examples
nano image.py # 128x32 veya 128x64 SPI veya i2c
sudo python image.py
```

5.5 QT

```
sudo apt-get install qtcreator \
    qt5-qmake qt5keychain-dev
sudo apt-get install qt5-default
```

```
./HelloWorld -platform eglfs
./HelloWorld -platform linuxfb
./HelloWorld -platform linuxfb:fb=/dev/fb1
./untitled1 -platform linuxfb:fb=/dev/fb1
    -plugin EvdevKeyboard:/dev/input/event0
./untitled1 -platform linuxfb:fb=/dev/fb1
    -plugin EvdevMouse -plugin EvdevKeyboard
```

```
LD_PRELOAD="/opt/vc/lib/libGLv2.so
/opt/vc/lib/libEGL.so"
myQtProgram -platform eglfs
```

5.6 Screen

```
#/etc/rc.local
su pi -c "screen -d -m -S pi"
```

```
# ~/.bashrc
screen -x pi
```


5.7 GPIO.OUT

```
import RPi.GPIO as GPIO
import time

# RPi pin numaralarını kullan
GPIO.setmode(GPIO.BOARD)
# 22 Nolu pini (GPIO25) çıkış yap
GPIO.setup(22, GPIO.OUT)
# GPIO25 pinini 40 kere aç/kapa
for i in range(40):
    GPIO.output(22,GPIO.HIGH)
    time.sleep(1)
    GPIO.output(22,GPIO.LOW)
    time.sleep(1)
GPIO.cleanup()
```

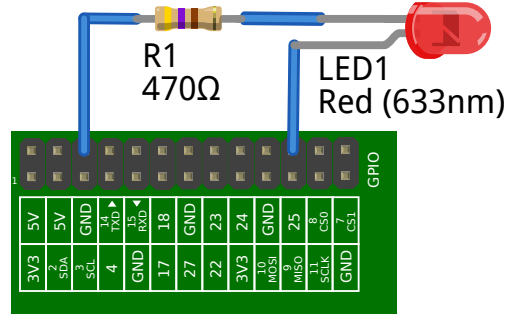


Figure 6: LED yakıp söndürme şeması

5.8 GPIO.IN

```
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.IN,
    pull_up_down = GPIO.PUD_DOWN)
GPIO.setup(24, GPIO.IN,
    pull_up_down = GPIO.PUD_UP)

while True:
    if(GPIO.input(23) ==1):
        print("1. buton basildi")

    if(GPIO.input(24) == 0):
        print("2. buton basildi")

GPIO.cleanup()
```

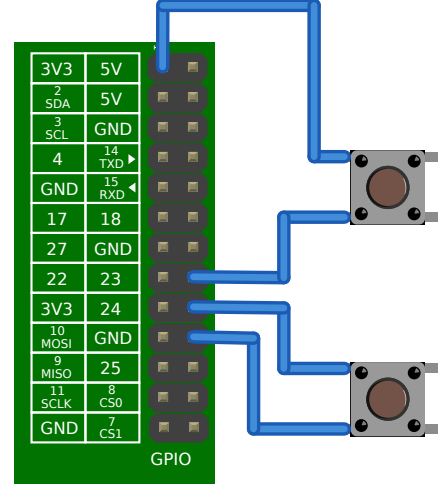


Figure 7: Giriş Çıkış Örneği

Tanım			Tanım
3.3V	1	2	5V
GPIO 0/2 I2C SDA	3	4	5V
GPIO 1/3 I2C SCL	5	6	GND
GPIO 4	7	8	GPIO 14 UART TXD
GND	9	10	GPIO 15 UART RXD
GPIO 17	11	12	GPIO 18
GPIO 21/27	13	14	GND
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
GPIO 10 SPI MOSI	19	20	GND
GPIO 9 SPI MISO	21	22	GPIO 25
GPIO 11 SPI SCLK	23	24	GPIO 8 SPI CE0
GND	25	26	GPIO 7 SPI CE1

5.9 SPI

```
VCC ->1
GND ->6
CLK (SCL) -23
DATA -> 19
CS (chip select) -> 24
D/C -> Pin 10
```

Benim OLED

6 Pins!, note SCL=CLK, SDA=MOSI, not IIC pins

```
VCC -> 1
GND -> 6
CLK -> 23
MOSI -> 19?
CS -> 24
D/C -> 10
```

6. ESP8266 ile IOT

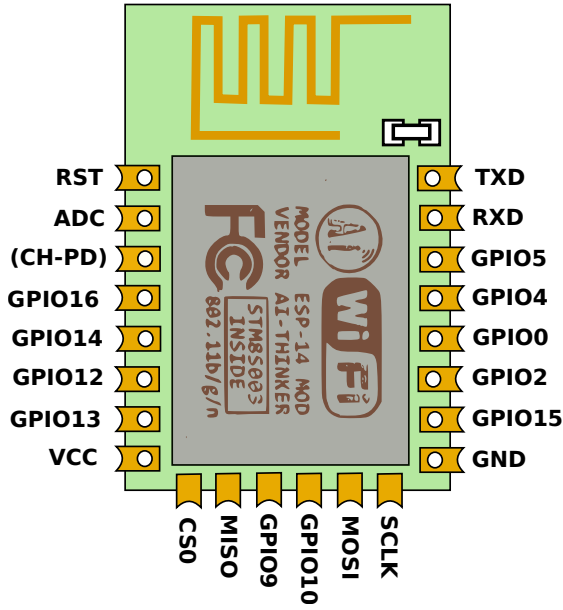


Figure 8: ESP8266 Pin Bağlantıları

```
esptool.py --port /dev/ttyUSB0 flash_id
```

Connecting...

Manufacturer: e0

Device: 4016

<https://code.coreboot.org/p/flashrom/source/tree/HEAD/trunk/flashchips.h> adresinden 4016 id chip i bulup (bu örnek için AMIC_A25LQ032) internetten arama yaparak bilgi edinebilirsiniz.

6.1 ESP8266 ve AT Komutları

www.electrodragon.com/w/ESP8266_AT-Command_firmware

Adresinden ESP8266 için AT Command firmware i indirip MCU ya atmanız gerekli. Modülün içerisinde bulunan herşeyi silmek için;

```
sudo ./esptool.py --port /dev/ttyUSB0 erase_flash
```

```
$ sudo minicom -D /dev/ttyUSB0 -b 115200
```

ile seri bağlantı yaptıktan sonra modülün reset butonuna basınız.

```
db8{lsd
Ai-Thinker Technology Co. Ltd.
ready
```

Bundan sonra AT yazdıktan sonra Ctrl+M ve Ctrl+J tuşlarına basarak AT komutunu gönderin. Ekranda,

```
AT
OK
```

gözükecektir. Diğer AT komutlarını denemek için mode=3 e geçmeniz lazım.

```
AT+CWMODE=3
```

```
OK
AT+CWJAP="KaraKarga","abc12345"
```

```
AT+CIFSR
192.168.1.135
```

```
AT+GMR
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
Ai-Thinker Technology Co. Ltd.
Jun 13 2016 11:29:20
OK
```

```
AT+CIOWRITE=12,1
```


6.2 NodeMCU ve LUA ile IOT

NodeMCU, üzerinde NodeMCU firmware yüklü ESP8266 WiFi modül barındıran bir geliştirme kartıdır.

Fazladan mikrokontrolcüye gerek olmadan GPIO, PWM, IIC, 1-Wire ve ADC bağlantıları yapılabilir. Kart üzerinde CH340 USB-Seri dönüştürücü entegre bulunduğu için microUSB ile doğrudan bilgisayarın USB portuna bağlanabilir.

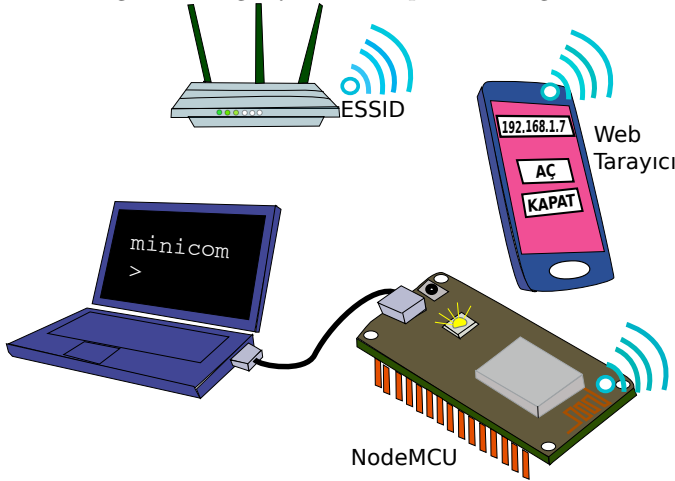


Figure 9: NodeMCU

```
$ sudo apt install python-serial
```

http://nodemcu.com/index_en.html

Nodemcu nun kaynak kodlarına
<https://github.com/nodemcu/nodemcu-firmware/>

adresinden erişilebilir.
firmware bin olarak releases dizininde bulunmaktadır.

<https://github.com/nodemcu/nodemcu-firmware/releases>

Veya

nodemcu-build.com

adresinden nodemcu modüllerini kendiniz seçerek bin dosya oluşturabilirsiniz. Oluşturulan dosyanın indirme adresi mailinize gönderilecektir.

```
$ git clone https://github.com/themadinventor/esptool
```

```
$ sudo python ./esptool.py --port /dev/ttyUSB0
write_flash 0x00000
../nodemcu-xxx.bin
```

```
sudo minicom -b 9600 -D /dev/ttyUSB0
```

Komutu ile USB kablusuyla ESP8266 modülüne terminalden bağlanıyoruz. NodeMCU üzerinde microUSB nin sol tarafında bulunan reset butonuna bastığımızda;

```
NodeMCU 0.9.6 build 20150704 powered by Lua 5.1.4
lua: cannot open init.lua
>
```

Ekrana bir mesaj yazdırarak Lua'nın çalıştığını kontrol edelim.

```
> print ("Merhaba Dünya")
```

```
Merhaba Dünya
```

```
>
```

Şimdi de NodeMCU umuzu wifi ağına bağlayalım.

```
> wifi.setmode(wifi.STATION)
> wifi.sta.config("KaraKarga","abc12345")
> print (wifi.sta.getip())
```

Ağdaki cihazların IP numaralarını ;

```
nmap -sP 192.168.1.0/24
```

şeklinde bir komut ile sorgulayabilirsiniz.

NodeMCU nun açılışta wifi ağına bağlanmasını ve başka ayarlamaları yapmak istiyorsak bu komutları `init.lua` dosyası içine yazmamız gerekecek. NodeMCU içinde script yazacak bir araç olmadığı için bu dosyayı bilgisayarımızda oluşturup cihaz içerisine göndermemiz gerekecek. Bu gönderme işlemi için `luatool` aracını kullanacağız. `luatool` aracını;

```
$ git clone https://github.com/4refr0nt/luatool.git
```

adresinden indirebiliriz. `luatool` klasörü içindeki `init.lua` dosyasındaki;

```
wifi.sta.config("KaraKarga","abc12345")
```

satırına kendi wifi ağınızın ESSID ve şifresini yazabilirsiniz. Veya kendiniz şu şekilde bir lua dosyası oluşturun.

```
--init.lua
wifi.setmode(wifi.STATION)
wifi.sta.config("KaraKarga","abc12345")
wifi.sta.connect()
tmr.alarm(1, 1000, 1, function()
if wifi.sta.getip()== nil then
print("IP yok, Bekleniyor...")
else
tmr.stop(1)
print("ESP8266 Modu : " .. wifi.getmode())
print("MAC Adresi : " .. wifi.ap.getmac())
print("IP Adresi : " .. wifi.sta.getip())
--do file ("led.lua")
end
end)
```

Dosyayı düzenledikten sonra aşağıdaki komutla `init.lua` dosyasını NodeMCU ya gönderebilirsiniz. Bu işlem sırasında `minicom` ile cihaza bağlı olmamanız gerekir. Eğer `minicom` ile bağlıysanız Ctrl+a x ile `minicom` dan çıkın. Aksi takdirde USB portu meşgul olacağından script upload edilemeyecektir.

```
$ python luatool/luatool/luatool.py
--port /dev/ttyUSB0 --src init.lua
--dest init.lua --restart
```

NodeMCU üzerinde bulunan mavi LED'i aşağıdaki kodlarla yakabilirsiniz.

```
-- led.lua
led1 = 0
gpio.mode(led1, gpio.OUTPUT)
gpio.write(led1, gpio.HIGH);
gpio.write(led1, gpio.LOW);
```

Veya bu kodu `led.lua` dosyası içerisine yazıp `luatool` ile cihaza yükleyebilirsiniz. Başlangıçta çalışmasını istiyorsanız `init.lua` dosyasına `do file ("led.lua")` satırını ekleyin. LOW açık, HIGH kapalıdır.

6.3 Lua ile Web Sunucu

İnternet üzerindeki bir nesneye ulaşmanın en kolay yolu web tarayıcıdır. Bu nedenle modüllere ulaşmak web sunucu üzerinden yapılır. Daha kolay ve erişim için kullanılan arayüz sistemden bağımsızdır. Aşağıdaki kodu NodeMCU ya init.lua adı altında gönderip web tarayıcınızın adres satırından IP numarasına bağlanmanız gerekir.

Komut satırından ise;

```
$ curl http://192.168.1.135/?pin=ON
$ curl http://192.168.1.135/?pin=OFF
```

```
-- init.lua
wifi.setmode(wifi.STATION)
wifi.sta.config("KaraKarga","abc12345")
print(wifi.sta.getip())
led1 = 0
gpio.mode(led1, gpio.OUTPUT)
srv=net.createServer(net.TCP)
srv:listen(80,function(conn)
  conn:on("receive", function(client,request)
    local buf = "";
    local _, _, method, path, vars =
      string.find(request,"([A-Z]+) (.+)?(.+) HTTP");
    if(method == nil)then
      _, _, method, path =
        string.find(request, "([A-Z]+) (.+) HTTP");
    end
    local _GET = {}
    if (vars ~= nil)then
      for k, v in
        string.gmatch(vars,"(%w+)=(%w+)&*") do
          _GET[k] = v
        end
      end
    end
    buf = buf.."<h3> ESP8266 Web Sunucu</h3>";
    buf = buf.."<p>GPIO0 <a href=\"?pin=ON\">";
    buf = buf.."<button>KAPAT</button></a>&nbsp;";
    buf = buf.."<a href=\"?pin=OFF\"><button>";
    buf = buf.."&Ccedil;</button></a></p>";
    local _on,_off = "",""
    if(_GET.pin == "ON")then
      gpio.write(led1, gpio.HIGH);
    elseif(_GET.pin == "OFF")then
      gpio.write(led1, gpio.LOW);
    end
    client:send(buf); client:close();
    collectgarbage();
  end)
end)
```

7. Arduino IDE ile ESP8266 Programlama

Arduino IDE Ubuntu paketleri arasında bulunmaktadır. Ancak sürekli yeni sürümleri yayınlanmaktadır. Bu nedenle son sürümü sitesinden indirerek kullanacağız.

1. <https://www.arduino.cc/en/Main/Software> adresinden Arduino IDE yi indirip bir dizine açınız.

`tar xvJ arduino-1.8.1-linux64.tar.xz`
2. File → Preferences → Additional Board Manager URLs girdisine http://arduino.esp8266.com/stable/package_esp8266com_index.json adresini ekleyin
3. Tools → Board → Boards Manager sekmesine gidip ESP8266 yı bularak kurulumu tamamlayın.
4. Sketch Include → Library → Manage Libraries sekmesinden ise "esp8266webserver" araması yaparak gerekli Library yi ekleyiniz.

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>

const char* ssid = "KaraKarga";
const char* password = "abc12345";

ESP8266WebServer server(80);

#define LED      D0

void handleLed() {
    if (server.arg(0)=="ON") digitalWrite(LED, LOW);
    else digitalWrite(LED, HIGH);

    String message = "<!DOCTYPE HTML>\r\n<html>";
    message += "<h3>ESP8266 Arduino Web Sunucu</h3>";
    message += "<p>GPIO0 <a href=\"led?pin=ON\">";
    message += "<button>A&Ccedil</button></a>&nbsp;";
    message += "<a href=\"led?pin=OFF\"><button>";
    message += "KAPAT</button></a></p></html>";
    server.send(200, "text/html", message);
}

void handleNotFound(){
    String message = "404 File Not Found\n\n";
    server.send(404, "text/plain", message);
}

void setup(void){
    pinMode(LED, OUTPUT); digitalWrite(LED, HIGH);
    Serial.begin(115200);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500); Serial.print(".");
    }

    Serial.print("IP adresi : ");
    Serial.println(WiFi.localIP());

    server.on("/led", handleLed);

    server.on("/", [](){
        String message = "<!DOCTYPE HTML>\r\n<html>";
        message += "<h3>ESP8266 Arduino Web Sunucu</h3>";
        message += "<a href=/led>Led Ornek</a></html>";
        server.send(200, "text/html", message);
    });

    server.onNotFound(handleNotFound);
    server.begin();
}

void loop(void){
    server.handleClient();
}
```

8. PlatformIO ile IOT Geliştirme

PlatformIO geliştirme aşamasında olduğu için Ubuntu paketleri arasında bulunmaz. pip ile kurmak gerekir pip kurulu değilse öncelikle pip i kurmak gerekir. Bunun için;

```
$ sudo apt install pip
```

PlatformIO python modüllerini kurmak için kullanacağımız komut;

```
$ pip install -U platformio
```

sudo kullanmadığımız için pio paketleri kullanıcı home dizinine kurulacaktır.

Bu aşamalardan sonra örnek bir uygulama için gerekli işlemler aşağıda verilmiştir. Daha önce ArduinoIDE ile yaptığımız WebSunucu ve led yakıp söndürme projemiz PlatformIO ile kablosuz ağ üzerinden OTA güncellemesi ile yapacak şekilde düzenlenmiştir.

```
$ platformio boards
```

Komutu ile desteklenen boardların listesini görebilirsiniz.

```
$ platformio platforms list
```

ile de desteklenen platformlar görülebilir. Şimdilik espresif8266 ve atmelavr platformları desteklenmektedir.

```
mkdir BasicOTA
cd BasicOTA
```

```
platformio init --board=nodemcu
```

.ino veya .cpp uzantılı ArduinoIDE projemizi src altında oluşturacağız veya daha önce ArduinoIDE ile oluşturduğumuz .ino dosyasını src klasörüne kopyalamamız gerekiyor. Derleyip ESP8266 modülüne atmak için komut;

```
$ pio run --target upload --upload-port /dev/ttyUSB0
```

platformio komutunu pio olarak da kısaltabiliriz.

Derlediğimiz bin dosyayı USB portundan değil de kablosuz ağ üzerinden modüle yüklemek istediğimizde yapmamız gereken /dev/ttyUSB0 yerine modülün kablosuz ağdan aldığı IP numarasını kullanmamız gerekiyor.

```
pio run -e nodemcu -t upload --upload-port 192.168.1.16
```

Şimdi modülümüz ağ üzerinden dosya göndermelerle açık hale geldi. Ancak OTA (On The Air) yüklemelerini şifreli hale getirebiliriz. Bunun için kodda verilen `ArduinoOTA.setPassword("admin");` satırını açmamız gerekir. Bunun dışında `platformio.ini` dosyası içerisine de

```
; PlatformIO Project Configuration File
```

```
upload_flags = --port=8266 --auth=admin
```

şeklinde gerekli etiketleri yazmamız gereklidir. Modülde ve ini dosyasında şifre admin olarak verilmiştir.

NodeMCU modülümüz ağda `ESP_MODULID` olarak görünür. Değiştirmek için kod içerisinde;

```
ArduinoOTA.setHostname("myesp8266");
```

komutu kullanılabilir. Yüklemeden sonra

```
$ ping myesp8266
```

ile kontrol edebilirsiniz.

```
// BasicOTA.cpp
#include <ESP8266WiFi.h>
#include <ESP8266mDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
#include <WiFiClient.h>
#include <ESP8266WebServer.h>
#define LED D0
const char* ssid = "KaraKarga";
const char* password = ".....";
ESP8266WebServer server(80);

void setup() {
  pinMode(LED, OUTPUT); digitalWrite(LED, HIGH);
  Serial.begin(115200); Serial.println("Booting");
  WiFi.mode(WIFI_STA); WiFi.begin(ssid, password);
  while (WiFi.waitForConnectResult() != WL_CONNECTED) {
    Serial.println("Connection Failed! Rebooting...");
    delay(5000); ESP.restart(); }
  //ArduinoOTA.setPort(8266);
  //ArduinoOTA.setHostname("myesp8266");
  //ArduinoOTA.setPassword("admin");
  // MD5(admin) = 21232f297a57a5a743894a0e4a801fc3
  // ArduinoOTA.setPasswordHash(
    "21232f297a57a5a743894a0e4a801fc3");
  ArduinoOTA.onStart([]() {
    String type;
    if (ArduinoOTA.getCommand() == U_FLASH) type = "sketch";
    else type = "filesystem";
    Serial.println("Start updating " + type);
  });
  ArduinoOTA.onEnd([]() {
    Serial.println("\nEnd"); });
  ArduinoOTA.onProgress([](unsigned int progress, unsigned int total) {
    Serial.printf("Pr :%u%%\r", (progress/(total/100)));
  });
  ArduinoOTA.onError([](ota_error_t error) {
    Serial.printf("Error[%u]: ", error);
    if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
    else if (error == OTA_BEGIN_ERROR) Serial.println("Begin Failed");
    else if (error == OTA_CONNECT_ERROR) Serial.println("Connect Failed");
    else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive Failed");
    else if (error == OTA_END_ERROR) Serial.println("End Failed");
  });
  ArduinoOTA.begin(); Serial.println("Ready");
  Serial.print("IP address: ");
  Serial.println(WiFi.localIP());

  server.on("/", [](){
    if (server.arg(0) == "ON") digitalWrite(LED, LOW);
    else digitalWrite(LED, HIGH);
    String message = "<!DOCTYPE HTML>\r\n<html>";
    message += "<h3>ESP8266 OTA Web Sunucu</h3>";
    message += "<p>GPIO0 (LED D0) <a href=?pin=ON?>";
    message += "<button>A&Ccedil</button></a>&nbsp;";
```

```

    message += "<a href=\"?pin=OFF\"><button>";
    message += "KAPAT</button></a></p></html>";
    server.send(200, "text/html", message);
  });
  server.begin();
}

void loop() {
  ArduinoOTA.handle(); server.handleClient(); }

```

9. OSC Send

OSC (Open Sound Control) ses aletleri, bilgisayarlar ve multimedya araçları arasında mesajlaşma ve kontrol amaçlı kullanılan bir protokoldür. Mesaj gönderme işlemi UDP (User Datagram Protocol) ile gerçekleşir.

```
$ platformio lib install "OSC"
```

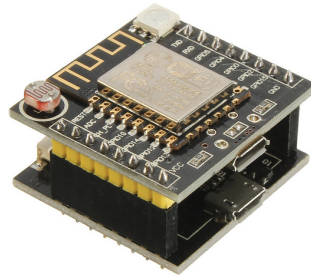


Figure 10: ESP8266-12E Modülü

IOT

```

/* github.com/karakaplanm/iot/OSCSendLDR/src/main.cpp

#include <ESP8266WiFi.h>
#include <WiFiUdp.h>
#include <OSCMessages.h>

char ssid[] = "KaraKarga";
char pass[] = "abc12345";
int ldr;
int xdata=0;
WiFiUDP Udp;
const IPAddress outIp(192,168,1,62);
const unsigned int outPort = 9000;
const unsigned int localPort = 8888;
void setup() {
  Serial.begin(115200);
  Serial.print("Baglanıyor ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");

  Serial.println("WiFi baglandi");
  Serial.println("IP adres: ");
  Serial.println(WiFi.localIP());

  Serial.println("UDP Gonderme Basliyor");
  Udp.begin(localPort);
  Serial.print("Lokal port: ");
  Serial.println(Udp.localPort());

}

void loop() {
  OSCMessage msg1("/ydata");
  ldr=analogRead(A0); msg1.add(ldr);
  Udp.beginPacket(outIp, outPort);
  msg1.send(Udp); Udp.endPacket(); msg1.empty();

  OSCMessage msg2("/xdata"); msg2.add(xdata);
  Udp.beginPacket(outIp, outPort);
  msg2.send(Udp); Udp.endPacket(); msg2.empty();
  delay(50); xdata++;
  if (xdata>100) xdata=0;
}

```

10. ESP8266-Arduino

11. Nesnelerin İnterneti İçin Komutlar

Git

```
git clone https://github.com/karakaplanm/iot
git pull
```

Seri Haberleşme

```
sudo minicom -D /dev/ttyUSB0 -b 115200
Çıkmak için: Ctrl+A, X
```

```
python -m serial.tools.miniterm <port> <baud>
Çıkmak için: Ctrl+Alt+]
```

Esptool

```
esptool.py --port /dev/ttyUSB0 erase_flash
```

LuaTool

```
luatool.py --port /dev/ttyUSB0 --src init.lua
--dest init.lua --restart
```

Platformio

```
platformio platforms list
platformio boards
pio lib search DHT22
pio lib install 115
platformio init --board=nodemcu
pio run -t upload --upload-port <port>
pio run -e nodemcu -t upload --upload-port <IP>
```

ESP8266-Arduino

```
#define led 2
pinMode(led, OUTPUT); digitalWrite(led, HIGH);

Serial.begin(115200);Serial.print("Baglnyor");

char ssid[]="KaraKarga";char pass[]="abc12345";
WiFi.begin(ssid, pass);
while(WiFi.status()!=WL_CONNECTED){delay(500);
Serial.print(".");}
Serial.println(WiFi.localIP());
```