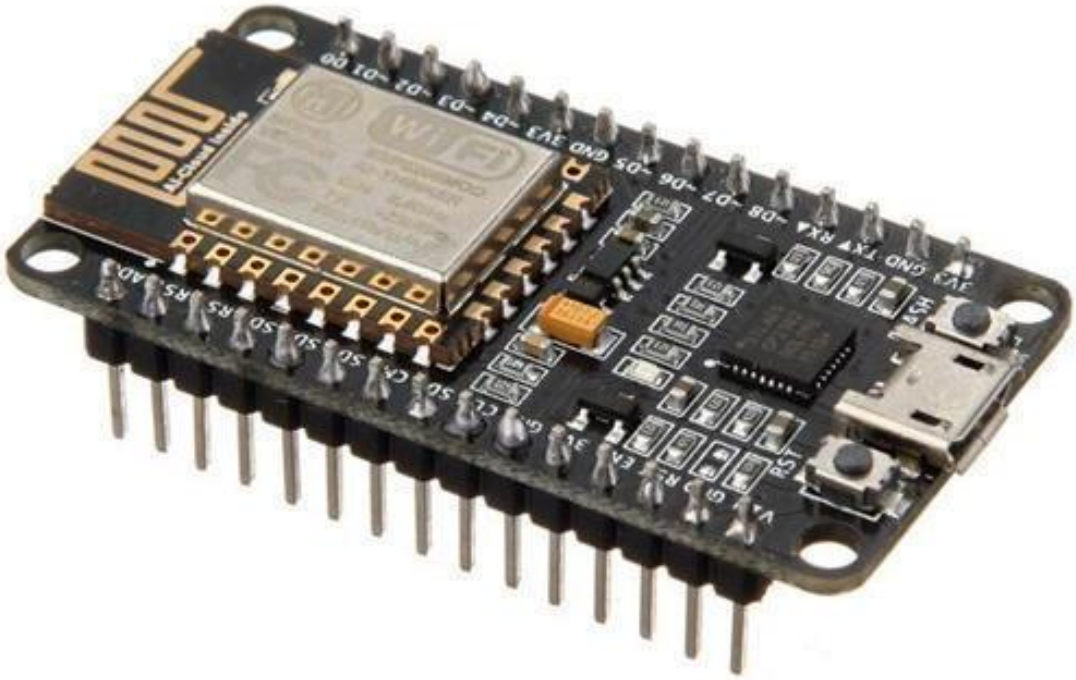
ماژول های WiFi

ESP8266 و NodeMCU



فرحان دائمی مژدهی

آکادمی رباتیک ایران

Iran Robotic Academy

معرفی تکنولوژی WiFi

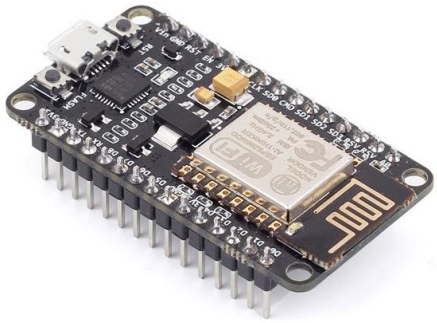
امروزه یکی از راه های ایجاد یک شبکه local network توسط چند سیستم استفاده از تکنولوژی WLAN یا Wireless Local Area Network (شبکه محلی بی‌سیم) است. این تکنولوژی که بسیار معروف است WiFi نام دارد که مخفف Wireless fidelity (وفاداری بی‌سیم) است. دسترسی به اینترنت با استفاده از این تکنولوژی توسط یک مودم یا روتر امکان پذیر است که در زندگی روزمره از آن بسیار استفاده می‌کنیم. در سال ۱۹۹۹ شخصی به نام کوین اشتون قصد داشت کاری کند که تمام اشیاء بی جان اطراف ما برای خود هویت دیجیتال داشته باشند و به کامپیوتر ها اجازه دهند آن ها را سازماندهی و مدیریت کنند. اینترنت در حال حاظر همه مردم را به هم متصل می‌کند اما او مفهوم دیگری به نام اینترنت اشیاء یا IOT (Internet of things) را خلق کرد تا اشیاء را به یکدیگر متصل کند. برای مثال ساعت دیواری خودش را از طریق اینترنت به روز رسانی کند و نیازی به تنظیم نداشته باشد و یا چراغ های خانه به صورت خودکار خاموش و روشن شوند یا صندلی ها به طور خودکار مرتب شوند.

ماژول های WiFi

ماژول های WiFi که امروزه در مبحث اینترنت اشیاء بسیار پرکاربرد هستند به مدار های الکتریکی قابلیت اتصال به یک شبکه WiFi را خواهند داد. ماژول های سری ESP8266 معروف ترین و پر استفاده ترین ماژول ها از این میان هستند. در این آموزش استفاده از ماژول های ESP8266 را به دو روش متفاوت خواهیم آموخت.

۱. استفاده از زبان ESP Basic
۲. استفاده از نرم افزار Arduino IDE

ماژول Node MCU

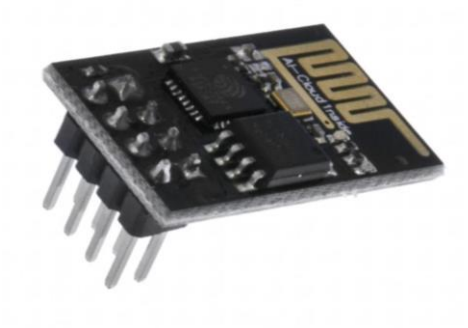


این ماژول شامل مدار کامل مورد نیاز برای برنامه ریزی یک ماژول ESP8266 را داراست و دیگر نیاز به هیچ قطعه جانبی دیگر برای برنامه ریزی آن نیست. همچنین استفاده از این ماژول بر روی برد بورد نیز امکان پذیر است. پیشنهاد می‌شود کسانی که برای اولین بار می‌خواهند با ESPها کار کنند، از این ماژول شروع کنند.

ماژول ESP8266 مدل 12F

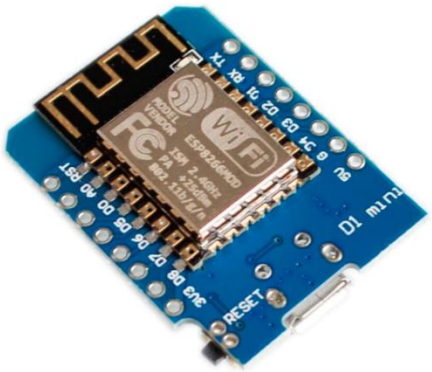
این ماژول برای اجرای پروژه ها بسیار خوب است و فقط می‌توان آن را بر روی PCB لحیم کرد. این ماژول از کیفیت و سرعت بسیار خوبی برخوردار است و برای پروژه های IOT بسیار مناسب است. برای برنامه ریزی این ماژول نیاز به یک مبدل USB به Serial خواهیم داشت.

ماژول ESP8266 مدل ESP-01



این ماژول برای کارهای سبک و ابتدایی مناسب است اما برای پروژه های سنگین و نیازمند به سرعت بالا توصیه نمی‌شود. تعداد پایه های GPIO در این ماژول بسیار کم بوده و به همین علت کاربرد آن را محدود می‌کند.

ماژول Mini NodeMCU

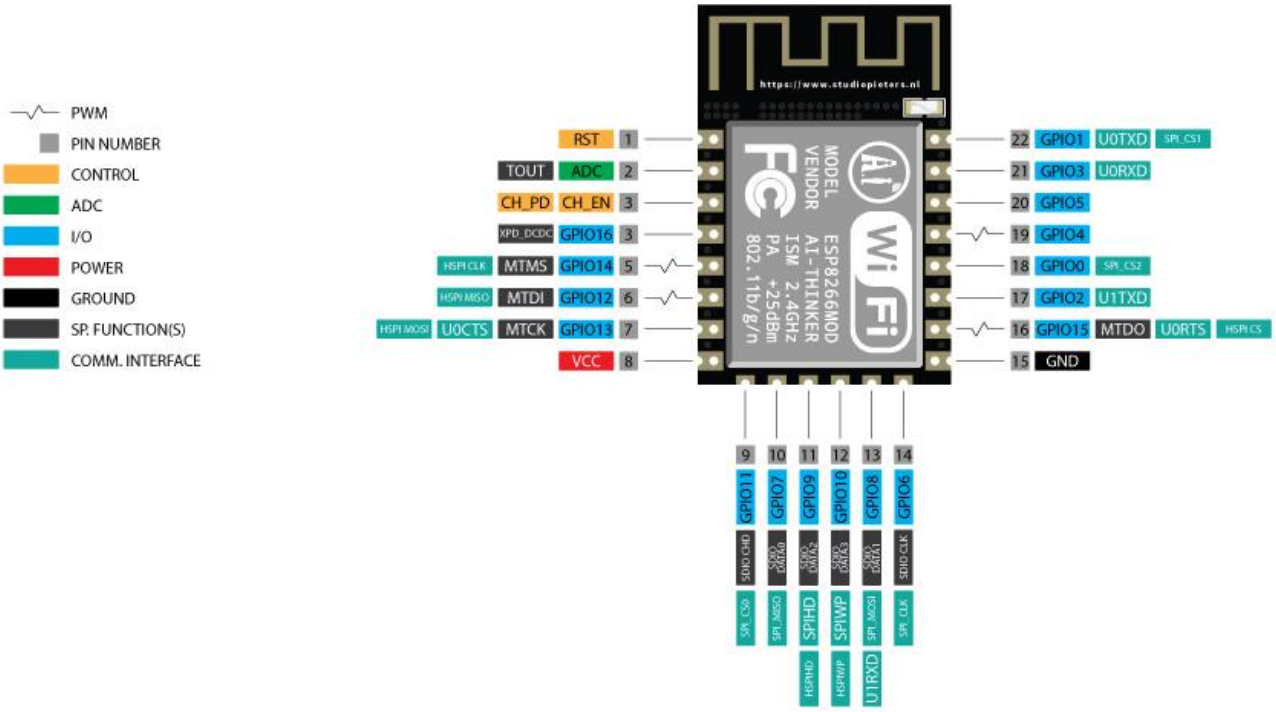


این ماژول نیز همانند Node MCU دارای مدار پروگرامر است و نیازمند هیچ گونه مدار جانبی جهت راه اندازی و برنامه ریزی نیست.

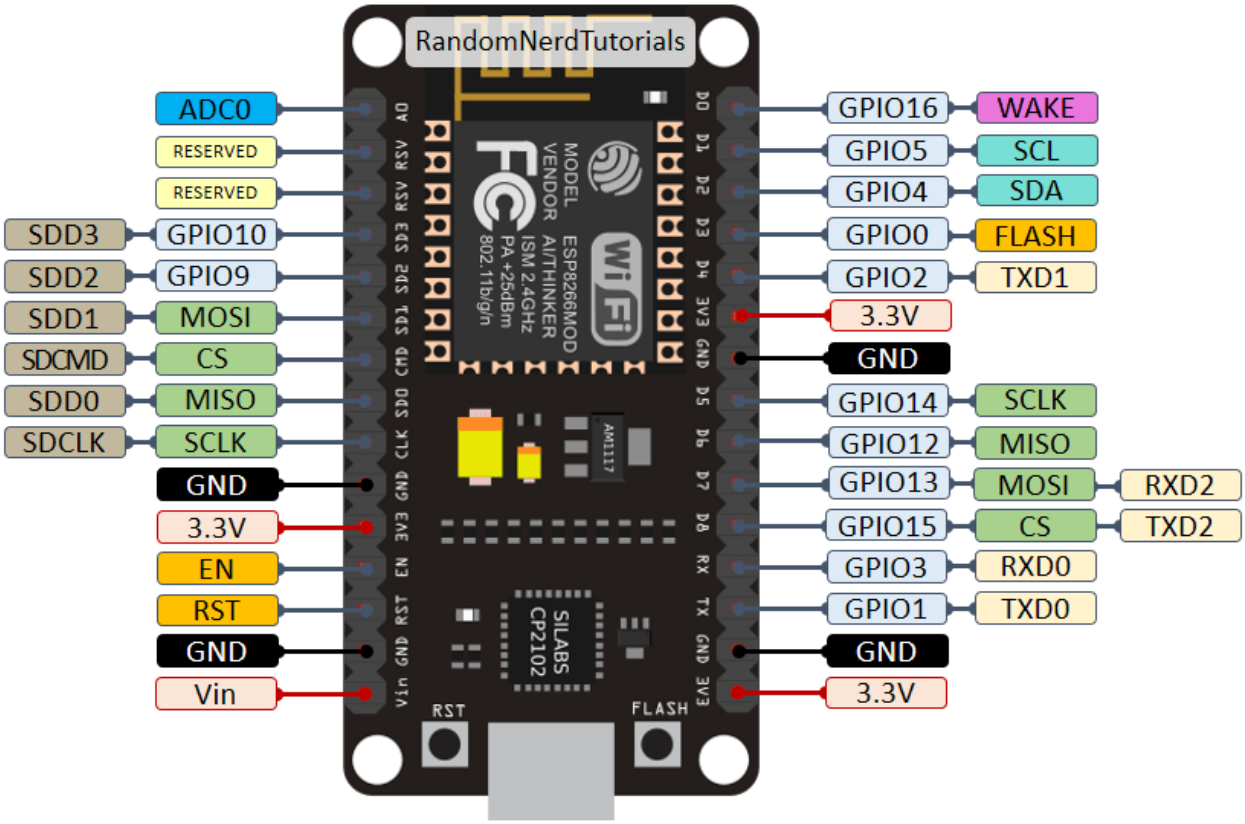
مفهوم Server و Client

به زبان ساده سرور (Server) یک کامپیوتر همیشه در دسترس می‌باشد که سرویس دهنده ی کامپیوترها و تجهیزات دیجیتالی متصل به شبکه است. کامپیوتر و تجهیزاتی که از سرویس دهنده خدمات دریافت میکنند را سرویس گیرنده یا کلاینت (Client) را میگویند. ماژول های WiFi که قرار است از آن‌ها استفاده نماییم می‌توانند در هر دو حالت Client و Server کار کنند.

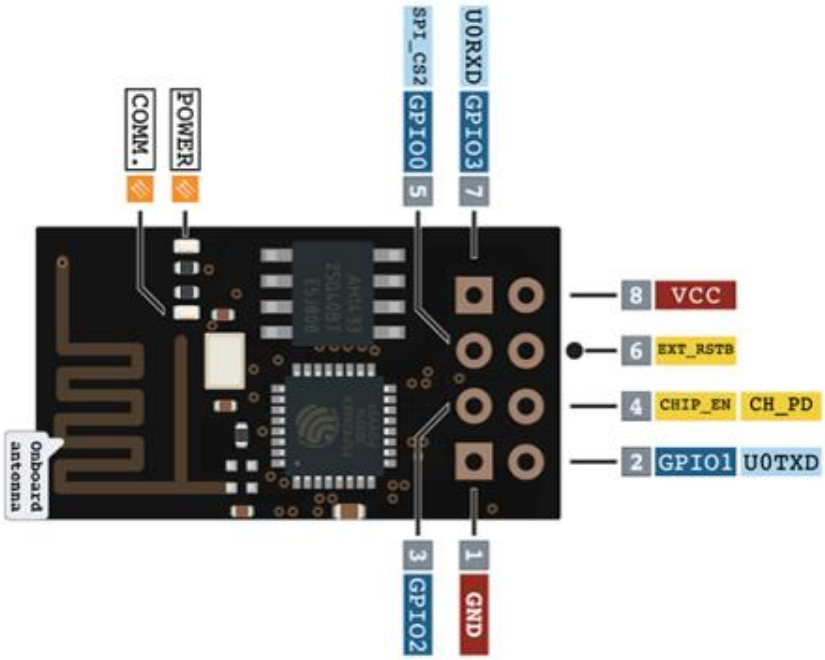
پایه های ماژول های ESP8266 مدل 12x



پایه های ماژول های Node MCU



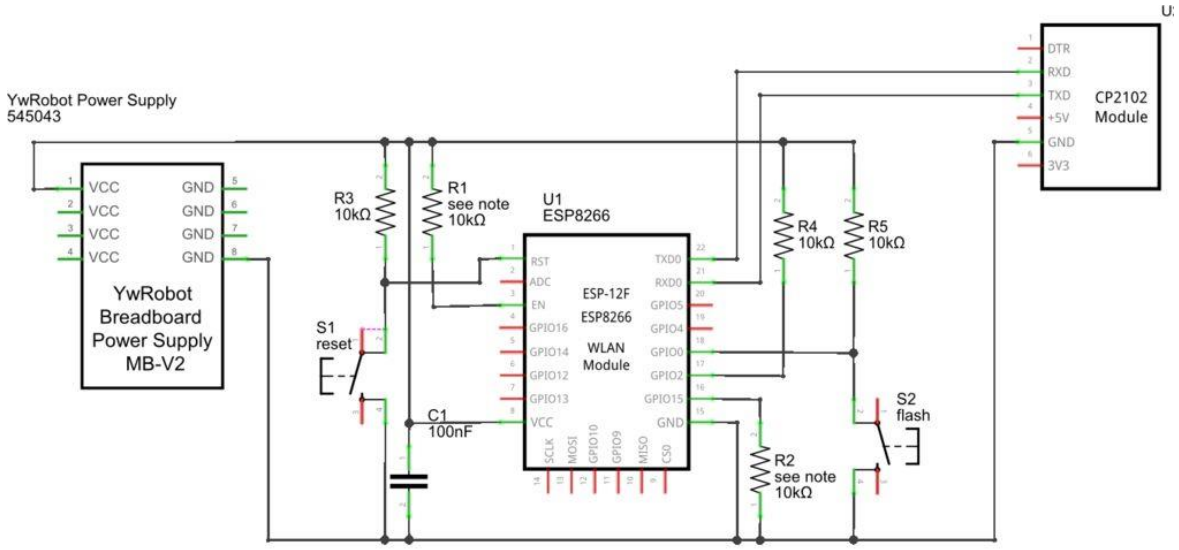
پایه های ماژول های ESP8266 مدل ESP-01



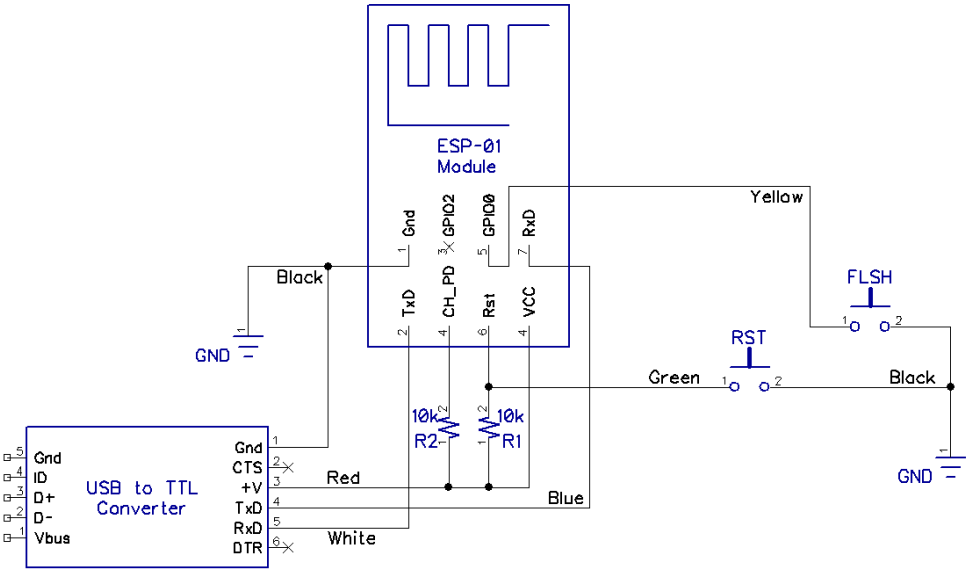
نحوه پروگرام کردن ماژول های ESP8266

پروگرام کردن ماژول های NodeMCU و Mini NodeMCU نیاز به هیچ مدار جانبی نداشته و تنها با اتصال کابل میکروUSB امکان پذیر می‌باشد. اما ماژول های دیگر ESP8266 برای پروگرام شدن نیاز به بستن مدار جانبی هستند و این مدار به صورت زیر خواهد بود. **هنگام پروگرام کردن این ماژول ها پایه GPIO0 باید به GND متصل شده و ماژول ریست شود تا ماژول در حالت برنامه پذیری قرار گیرد. به همین علت برای این پایه یک کلید قرار گرفته شده تا به هنگام نیاز با فشردن آن و سپس ریست کردن ماژول، آن را به حالت برنامه پذیری ببریم.**

ماژول ESP8266 مدل 12F

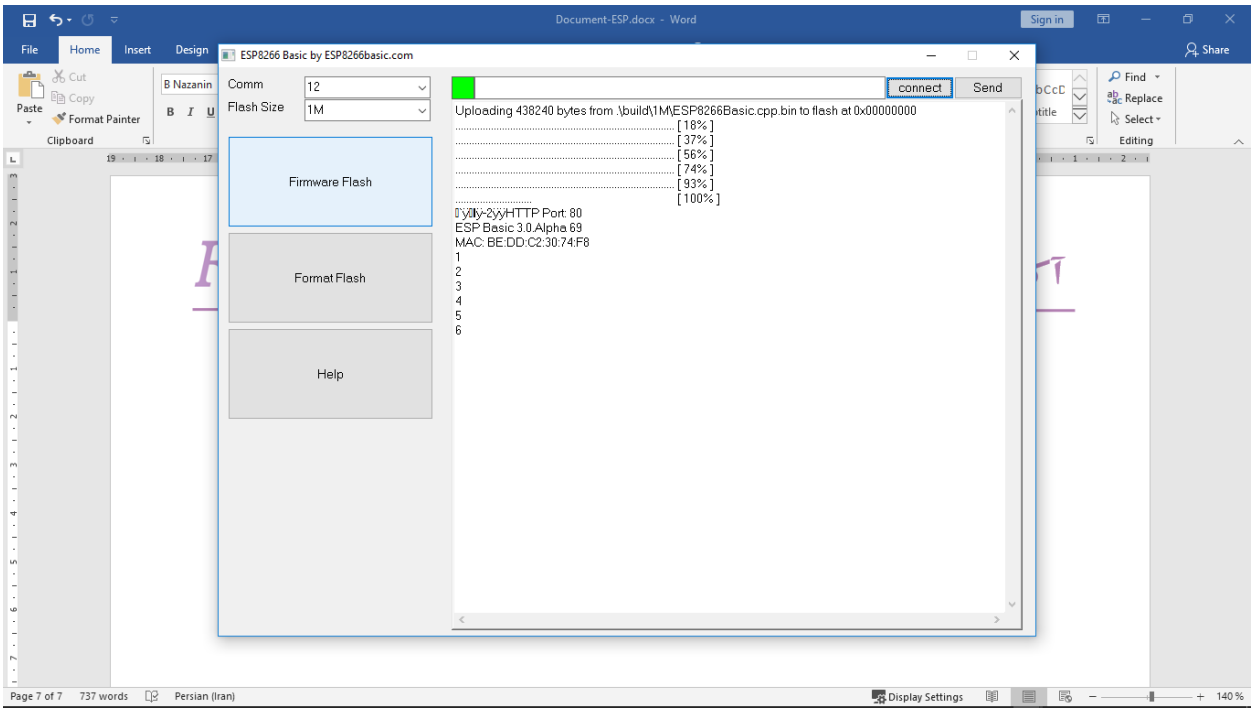


ماژول ESP8266 مدل ESP-01



پروگرام کردن ماژول با استفاده از زبان ESP Basic

توصیه می‌شود برای تکمیل اطلاعات در مورد زبان ESP Basic به آدرس www.esp8266basic.com مراجه شود. برنامه نویسی به زبان Basic برای ماژول های ESP نیاز به هیچ نرم افزاری ندارد و تنها با استفاده از یک مرورگر مانند Chrome یا Firefox می‌توان برای ماژول ESP برنامه نویسی، اجرای برنامه، اعمال تنظیمات و آپلود فایل را انجام داد. اما قبل از هر چیز لازم است بر روی این ماژول برنامه ESP Basic برای یک بار پروگرام شود. برای این کار می‌توانید نرم افزار Flasher را از وب سایت ذکر شده دانلود کرده و پس از اتصال ماژول به کامپیوتر همانند تصویر زیر COM (پورت سریال) و Flash Size (سایز حافظه داخلی) ماژول خود را مشخص کنید و سپس با زدن گزینه Firmware Flash برنامه را به ماژول منتقل کنید.

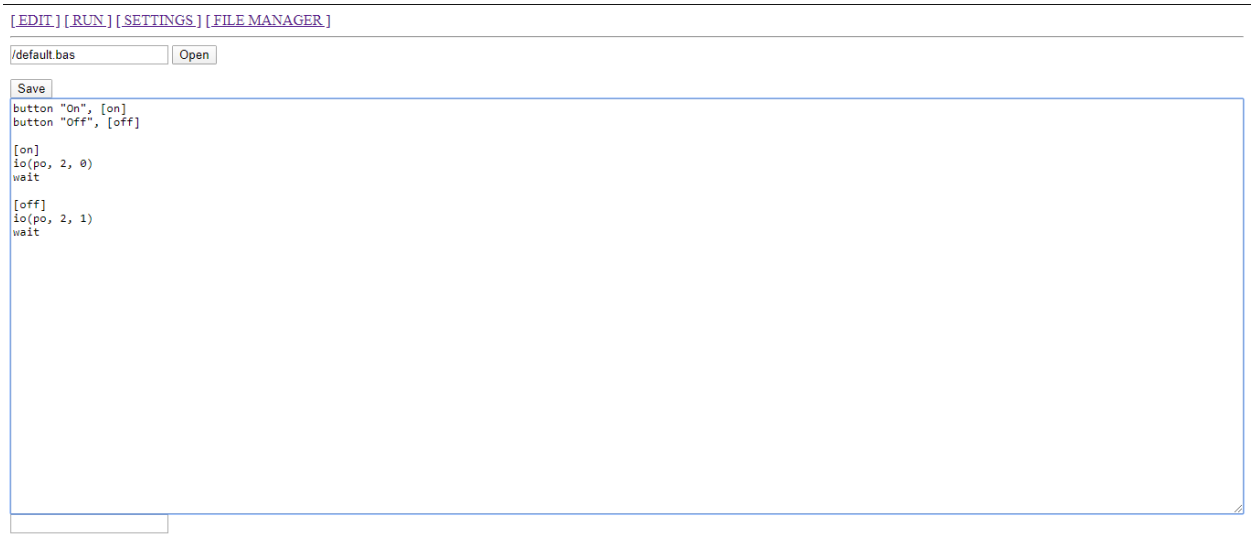


پس از اتمام آپلود، ماژول به صورت خودکار یک شبکه WiFi ایجاد خواهد کرد. با کامپیوتر خود به این شبکه متصل شوید سپس در مرورگر خود به آدرس http://192.168.4.1/edit بروید. در این آدرس شما می‌توانید فایل default.bas را ویرایش کنید. برنامه Basic در این فایل نوشته خواهد شد و هنگام انتخاب گزینه RUN برنامه ای که در این فایل ذخیره شده اجرا خواهد شد.

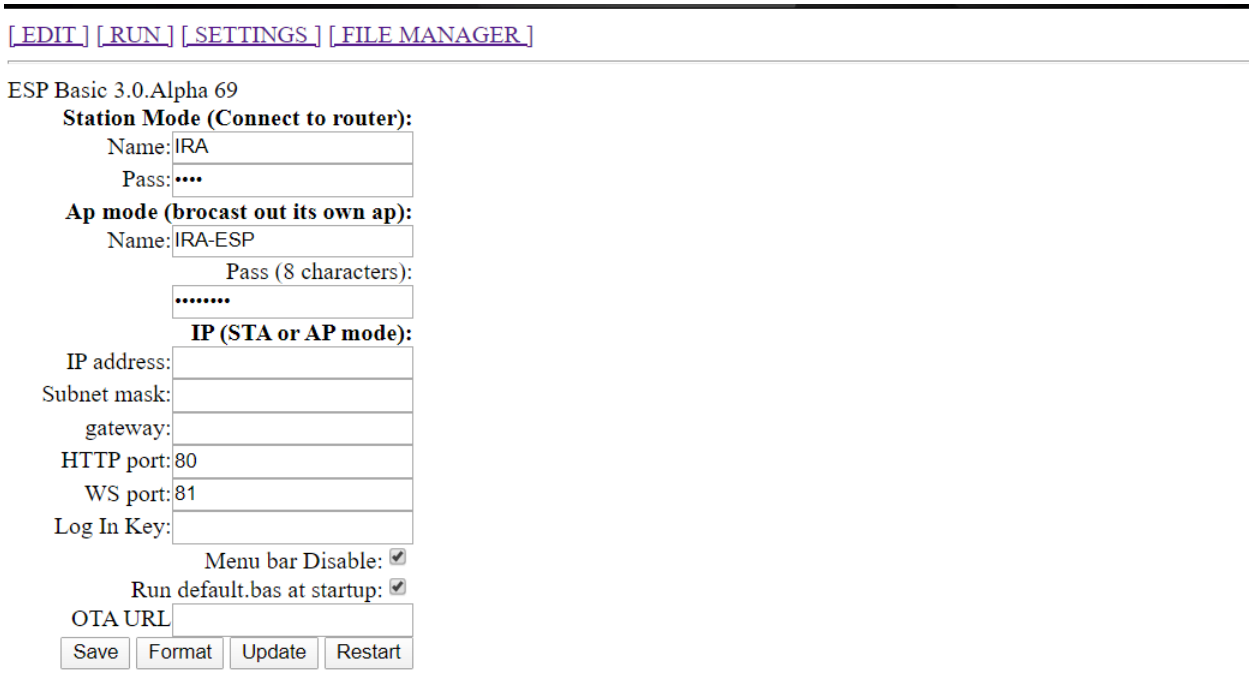
دستورات قابل استفاده در ESP Basic

کاربرد	دستور
چاپ کردن یک رشته	print “Hello”
پاک کردن کل صفحه	cls
توقف برنامه	wait
تغییر وضعیت منطقی پایه شماره ۲ به ۰	io(po, 2, 0)
تغییر PWM پایه ۲ به ۱۲۰	io(pwo, 2, 120)

نمونه برنامه برای خاموش و روشن کردن LED آبی رنگ روی ماژول.



در منوی SETTINGS شما می‌توانید تنظیمات این ماژول را مشخص نمایید. برای اینکه ماژول هنگام روشن شدن به یک مودم، یک روتر و یا یک هات اسپات متصل شود نام و کلمه عبور آن را در قسمت Station Mode مشخص نمایید. در صورتی که از مودم، روتر یا هات اسپات استفاده نمی‌کنید می‌توانید نام و کلمه عبور شبکه ای که توسط خود ماژول ایجاد می‌شود را در قسمت Ap Mode مشخص کنید. اگر تیک Menu bar disable را فعال کنید، هنگام اجرای برنامه منوی بالای آن نمایش داده نخواهد شد. اگر تیک Run default.bas at startup فعال شود، هنگام ورود به آی پی (به طور پیش فرض 192.168.4.1) فایل default.bas اجرا خواهد شد. پس از اعمال تغییرات گزینه Save را زده و ماژول را ریست کنید.



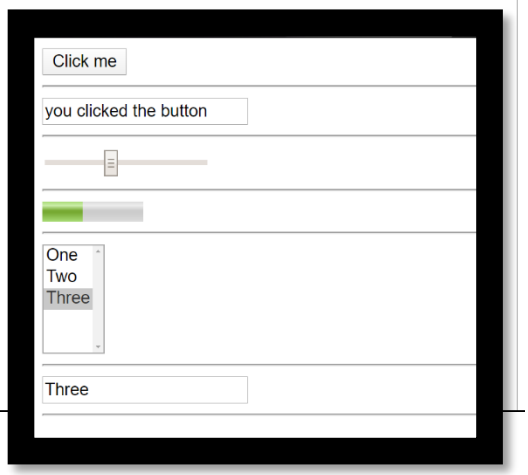
نمونه برنامه GUI

[EDIT] [RUN] [SETTINGS] [FILE MANAGER]

/default.bas Open

Save

```
button "Click me", [click]
print
textbox variablename
print
slider x, 0, 10
print
meter x, 0, 10
print
listbox selection, "One,Two,Three", 5
print
textbox selection
print
wait


[click]
Print "You Clicked me"
variablename = "you clicked the button"
x = 5
selection = "Three"
wait
```



افزودن فایل CSS به برنامه

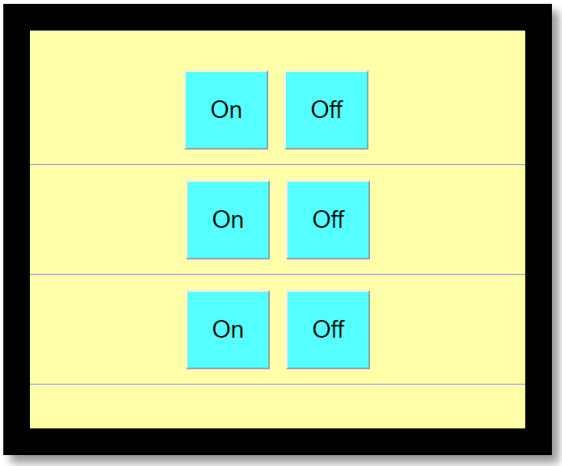
برای زیباتر ساختن برنامه می‌توانید برنامه css به آن اضافه کنید. برای اینکار ابتدا باید فایل css را در قسمت FILE MANAGER آپلود کرده و سپس برنامه را به صورت زیر بنویسیم.

[EDIT] [RUN] [SETTINGS] [FILE MANAGER]

/default.bas Open

Save

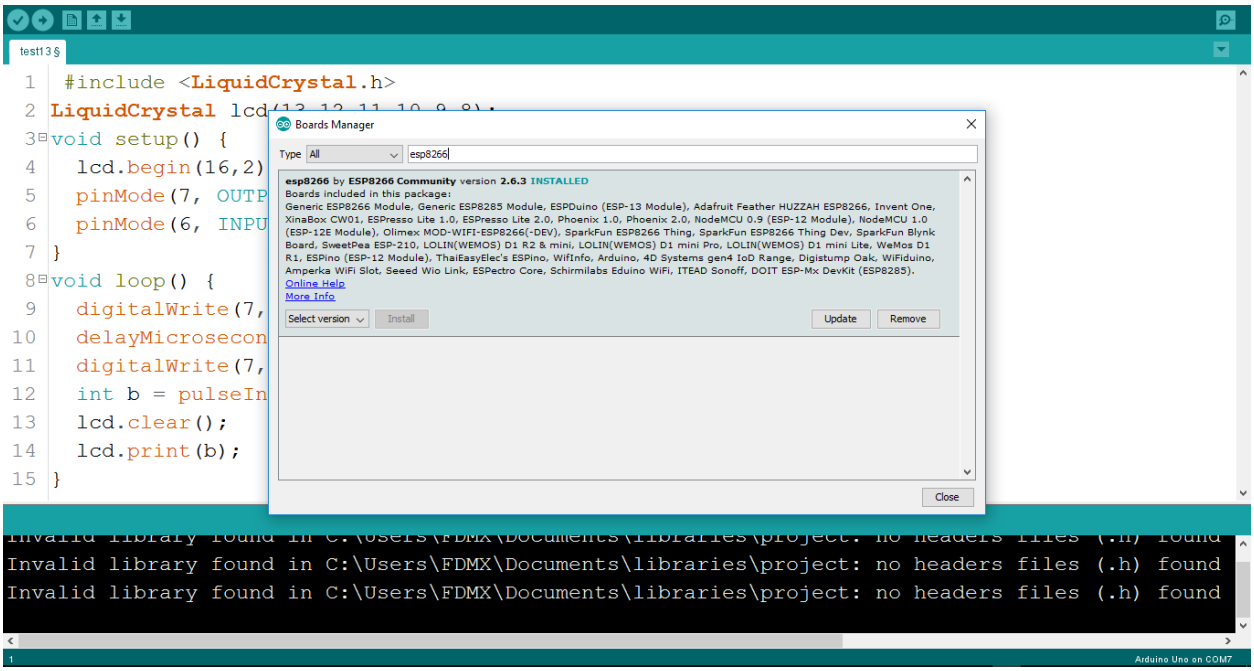
```
css css.css
button "On",[on1]
button "Off",[off1]
print " "
button "On",[on]
button "Off",[off]
print " "
button "On",[on2]
button "Off",[off2]
print " "
wait

[on]
io(po,12,1)
wait

[off]
io(po,12,0)
wait

[on1]
io(po,13,1)
wait

[off1]
io(po,13,0)
wait

[on2]
io(po,14,1)
wait

[off2]
io(po,14,0)
wait
```



راه اندازی ماژول های ESP با استفاده از Arduino IDE

قبل از هر کاری لازم است بوردها و ماژول های ESP را به لیست بوردهای نرم افزار Arduino IDE اضافه شود. برای اینکار داخل نرم افزار به قسمت File -> preferences رفته و لینک زیر را در قسمت Additional Boards Manager URLs اضافه کنید. سپس گزینه Ok را زده و به قسمت Tools -> Board -> Boards Manager رفته و کلمه ESP8266 را جستجو کنید. (دقت کنید که به اینترنت متصل باشید) سپس مجموعه بورد ها ESP را انتخاب و گزینه Install را بزنید.

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



برای پروگرام کردن ماژول کافی است در قسمت Tools -> Board بورد Generic ESP8266 Module را انتخاب نمایید، در قسمت Tools -> Port پورت مورد نظر را انتخاب کرده و گزینه Upload (ctrl + u) را بزنید. در قسمت File -> Examples نمونه برنامه های بسیار مفیدی در رابطه با این ماژول ها وجود دارد. برای روشن و خاموش کردن LED که بر روی ماژول ESP قرار دارد و به GPIO2 متصل است به صورت زیر عمل خواهیم کرد.

```
void setup() {
  pinMode(2, OUTPUT);
}

void loop() {
  digitalWrite(2, 1);
  delay(100);
  digitalWrite(2, 0);
  delay(100);
}
```

وصل شدن به یک شبکه WiFi / ایجاد یک شبکه WiFi

برای اتصال به یک شبکه WiFi موجود، ابتدا باید کتابخانه ESP8266WiFi.h را فراخوانی کنیم. ماژول های ESP در سه حالت STA (Station) و AP (Access Point) و AP_STA کار خواهد کرد. در حالت AP ماژول یک شبکه WiFi خودش ایجاد می‌کند و به دیگر دستگاه ها اجازه می‌دهد به آن وصل شوند. در حالت STA ماژول به یک شبکه ای که از قبل موجود است (مانند مودم، روتر یا هات اسپات) متصل می‌شود. در حالت AP_STA برای ماژول هر دو حالت AP و STA امکان پذیر است.

اتصال به شبکه WiFi (STA Mode)

ابتدا با استفاده از دستور WiFi.mode(WIFI_STA) ماژول را در حالت STA قرار می‌دهیم و برای وصل شدن به شبکه WiFi موجود از دستور WiFi.begin(ssid, pass) استفاده خواهیم کرد. ssid نام و pass کلمه عبور شبکه WiFi می‌باشد. سپس لازم است صبر کنیم تا زمانی که به شبکه متصل شود به همین منظور از یک حلقه While استفاده خواهیم کرد که شرط آن "وصل نبودن به شبکه" می‌باشد و این حلقه هر 500 میلی‌ثانیه تکرار می‌شود.

WiFiClient §

```
#include <ESP8266WiFi.h>
void setup() {
  pinMode(2, OUTPUT);
  digitalWrite(2,1);
  WiFi.mode(WIFI_STA);
  WiFi.begin("iranroboticacademy", "12341234");
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  digitalWrite(2,0);
}

void loop() {

}
```

Uploading...

```
Compressed 264368 bytes to 193554...

Writing at 0x00000000... (8 %)
```

18 Generic ESP8266 Module on COM12

ایجاد یک شبکه WiFi

برای ایجاد یک شبکه تنها نوشتن یک دستور کافی است. به صورت پیش فرض ESP بر روی حالت AP خواهد بود و نیازی به تنظیم مجدد آن نیست. با استفاده از دستور WiFi.softAP("SSID", "Password") می‌توان یک شبکه با نام SSID و رمز عبور Password ایجاد نماییم. آی پی این شبکه را می‌توانیم با استفاده از دستور WiFi.softAPIP() بدست آوریم و در متغیری از جنس IPAddress ذخیره نماییم.

```
WiFiAccessPoint §
1  #include <ESP8266WiFi.h>
2
3  void setup() {
4    WiFi.softAP("IranRoboticAcademy", "12341234");
5    IPAddress myIP = WiFi.softAPIP();
6  }
7
8  void loop() {
9
10 }
11
12
```

```
Done uploading.
Invalid library found in C:\Users\FDMX\Documents\libraries\project: no headers
Invalid library found in C:\Users\FDMX\Documents\libraries\project: no headers
18                                                  Generic ESP8266 Module on COM12
```

ساخت یک سرور HTTP

وظیفه یک سرور ارائه سرویس است. این سرویس می‌تواند به صورت های مختلفی مانند متن(plain/text)، صفحات وب(text/html) و .... باشند. در اینجا قصد داریم سروری بسازیم که یک برنامه کوچک HTML را به کلاینت هایی که از آن درخواست می‌کنند بفرستد. برای اینکار ابتدا کتابخانه های مربوطه را include می‌کنیم.

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

سپس با استفاده از دستور زیر یک سرور می‌سازیم.

```
ESP8266WebServer server(80);
```

پس از ساخت سرور برای ساخت یک شبکه WiFi به صورت زیر عمل میکنیم و ip ماژول را توسط ارتباط سریال چاپ می‌کنیم تا از آن برای اتصال به سرور استفاده کنیم. این آدرس ip را می‌توان بر روی یک LCD نیز نمایش داد. (این برنامه در تابع Setup نوشته می‌شود)

```
Serial.begin(115200);
WiFi.softAP(ssid, password);
IPAddress myIP = WiFi.softAPIP();
Serial.println(myIP);

server.on("/", handleRoot);
server.begin();
```

URL چیست؟

هر آدرسی که برای دسترسی به یک صفحه وب در مرورگر خود وارد می‌کنید مانند http://iranroboticacademy.com/ را یک URL می‌نامند که کوتاه شده Uniform Resource Locator می‌باشد. یک آدرس URL از سه قسمت اصلی تشکیل شده است.

نام / آدرس هاست

http://iranroboticacademy.com/home

پروتکل ارتباطی

Resource محل قرارگرفتن مورد نظر در هاست

برای هر کدام از آدرس های انتهای URL لازم است تابعی به عنوان handler (کنترل کننده) بسازیم. برای مثال اگر آدرس ip ماژول 192.168.43.88 باشد و بخواهیم برای آدرس http://192.168.43.88/ یک تابع handler بنویسیم به صورت زیر عمل می‌کنیم.

```
void handleRoot() {
  server.send(200, "text/html", "<h1>You are connected</h1>");
}
```

سپس برای سرور مشخص می‌کنیم این تابع برای / می‌باشد که همان URL بالاست (نیازی به نوشتن http:// و آدرس ip در تابع handler نیست). و در نهایت در تابع loop با دستور handleClient به درخواست تمامی کلاینت ها گوش می‌دهیم.

```
void setup() {
  Serial.begin(115200);
  WiFi.softAP(ssid, password);
  IPAddress myIP = WiFi.softAPIP();
  Serial.println(myIP);

  server.on("/", handleRoot);
  server.begin();
}

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

در نهایت پس از پروگرام و روشن کردن ماژول در مرورگر هر دستگاهی که به شبکه WiFi ایجاد شده توسط آن متصل باشد می‌تواند با وارد کردن آدرس ip ماژول به سروری که توسط ماژول ساخته شده درخواست ارسال کند و سرور در پاسخ آن درخواست توسط تابع handler یک کد کوتاه HTML ارسال می‌کند.