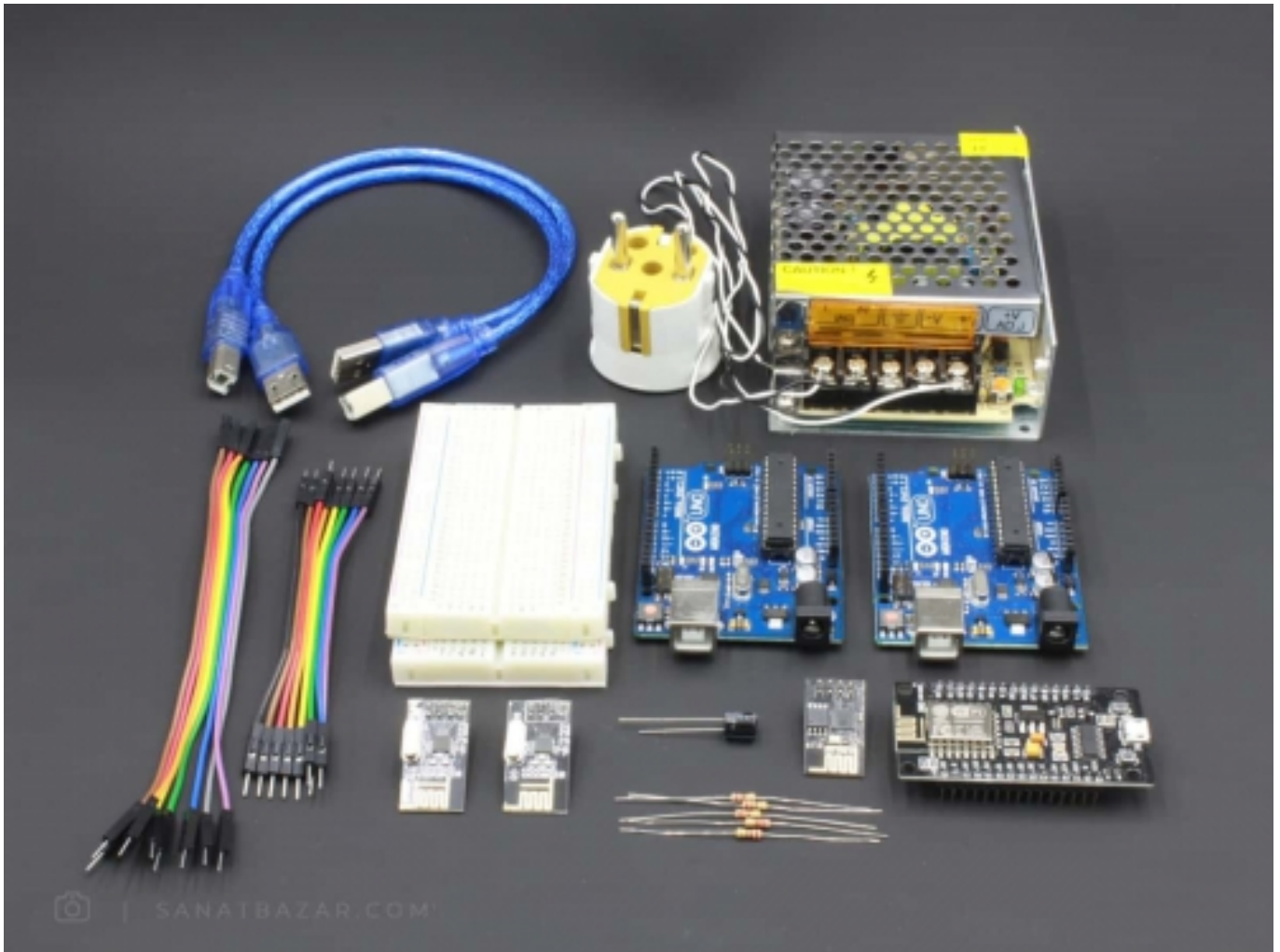








## معرفی و راه‌اندازی انواع ماژول وایرلس با آردوینو



این مطلب قسمت چهاردهم از [آموزش جامع آردوینو \(مبتدی و پیشرفته\)](#) است. در این قسمت قصد داریم به معرفی ماژول‌های بی‌سیم یا وایرلس بپردازیم. با اضافه کردن یک ماژول وایرلس به پروژه‌تان، می‌توانید کاربردهای آن را به کلی دگرگون کنید. ماژول‌های وایرلس بهترین ابزار برای تبدیل یک وسیله یا سنسور معمولی به یک پروژه IoT یا اینترنت اشیا هستند. تصور کنید که با استفاده از گوشی هوشمندتان بتوانید وسیله‌های خانه مانند روشنایی، درب پارکینگ، پرده‌ها، کولر و هر چیز دیگری را کنترل کنید. یا مثلاً فرض کنید که بخواهید دما و رطوبت گلهای خانه را زمانی که در محل دیگری هستید نظارت کرده و در صورت لزوم آنها را آبیاری کنید. همه این کارها که پروژه‌هایی از جنس اینترنت اشیا هستند، نیاز به وسیله‌ای برای ارتباط دارند که بهترین گزینه، یک ماژول وایرلس خواهد بود. در ماژول‌های وایرلس مانند خود ارتباطات بی‌سیم، تنوع زیادی وجود دارد و این موضوع دستتان را برای انتخاب یک محصول متناسب با خواسته‌تان باز می‌گذارد. در پایان این آموزش با تعدادی از ماژول‌های پرکاربرد وایرلس آشنا شده و هر کدام را راه‌اندازی خواهید کرد. با ما همراه باشید!

## قطعات مورد نیاز

	عدد 2	<a href="#">برد آردوینو UNO و کابل رابط</a>
	عدد 2	<a href="#">ماژول nRF24L01</a>
	عدد 1	<a href="#">ماژول ESP8266</a>
	عدد 1	<a href="#">برد NodeMCU</a>
	عدد 1	<a href="#">منبع تغذیه</a>
	عدد 5	<a href="#">مقاومت 220Ω</a>

1 عدد

خازن 100nF

2 بسته

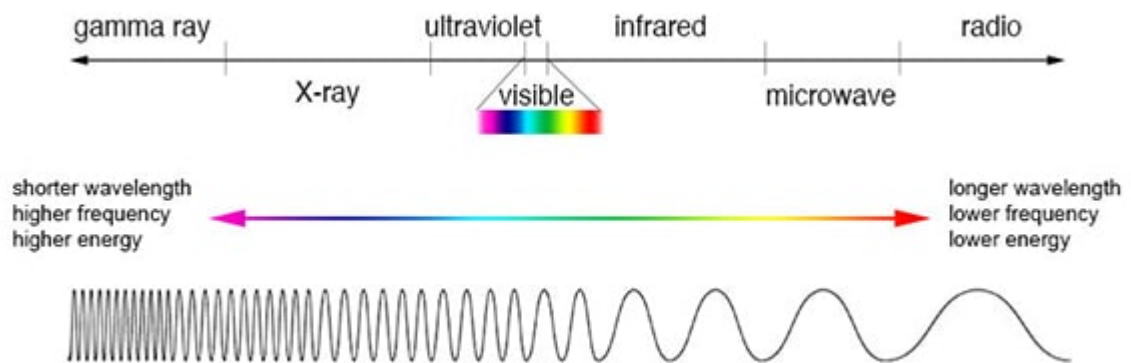
سیم جامپر

2 عدد

بردبورد

## معرفی ارتباط بی‌سیم

هرگاه صحبت از وایرلس باشد، پای طیف بسیار وسیعی از وسیله‌های الکترونیکی به میان می‌آید. در یک تعریف کلی وایرلس به ارتباطی گفته می‌شود که در آن از سیم به عنوان واسطه اطلاعات استفاده نشده باشد! این تعریف خیلی بدیهی است و به همین خاطر ارتباط وایرلس تا این حد گسترده است. تقریباً در تمام روش‌های ارتباطات بی‌سیم از امواج الکترومغناطیسی استفاده می‌شود. این امواج در اثر تغییرات میدان الکترومغناطیسی در فرستنده ایجاد می‌شود. ویژگی اصلی این نوع از امواج عدم نیاز به محیط مادی برای انتشار است. سرعت تغییرات این میدان، فرکانس موج الکترومغناطیسی را تعیین می‌کند. در تئوری، این فرکانس می‌تواند هر مقداری داشته باشد. امواج الکترومغناطیسی بر اساس فرکانس‌شان به چند دسته کلی تقسیم شده‌اند: رادیویی، مایکروویو، فرسرخ، مرئی، فرابنفش، اشعه X و اشعه گاما (به ترتیب از فرکانس کم به زیاد).



در این بین امواج رادیویی یا RF فرکانس‌های 30Hz تا 300GHz را پوشش می‌دهند و به این ترتیب می‌توان گفت تقریباً تمام روش‌های ارتباطی بی‌سیم در این محدوده قرار می‌گیرند. به همین دلیل معمولاً در ارتباطات بی‌سیم از امواج رادیویی صحبت شده و از مابقی امواج الکترومغناطیسی چشم‌پوشی می‌شود. برخلاف چیزی که شاید در اول تصور کرده باشید، امواج رادیویی مختص "دستگاه رادیو" نیست و سایر ارتباطات مانند شبکه‌های تلویزیونی، اینترنت بی‌سیم، تلفن همراه و غیره نیز از این امواج استفاده می‌کنند. هرچه فرکانس موج رادیویی بیشتر باشد، انرژی بیشتر اما طول موج و قدرت نفوذ کمتری دارد. به همین دلیل معمولاً استفاده از فرکانس‌های بالا برای کاربردهای نزدیک و فرکانس‌های پائین برای مسافت‌های دور مناسب‌تر هستند (البته نه همیشه).

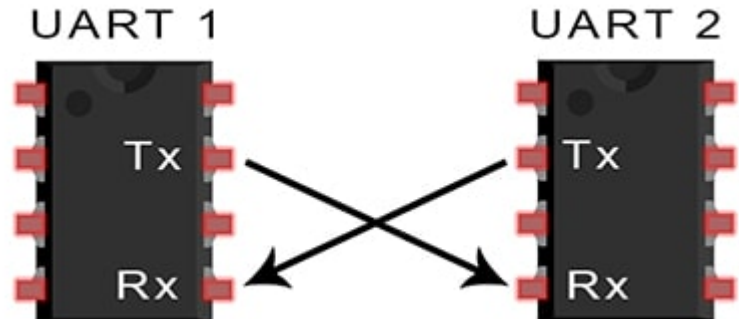
با در اختیار داشتن طیف گسترده امواج رادیویی و نیز روش‌های مختلف ارسال و دریافت اطلاعات، پروتکل‌های بسیار متنوعی برای ایجاد یک ارتباط وایرلس معرفی شده است. VHF، RFID، GSM، GPS، WiFi و ... هر کدام از پروتکل مخصوص خود برای برقراری ارتباط بهره می‌برند. بحث فنی در مورد نحوه ایجاد و دریافت امواج رادیویی و نیز نحوه ارسال و دریافت اطلاعات از این طریق، کاملاً یک بحث تخصصی است و در اینجا قصد نداریم به آن بپردازیم.

همان‌طور که قبلاً توضیح دادیم، وایرلس دسته بزرگی از انواع ارتباطات را شامل می‌شود اما معمولاً منظور از وایرلس بخش خاصی از این دسته با فرکانس 2.4GHz یا WiFi است که طبق استاندارد IEEE 802.11 b/g/n کار می‌کنند. در این بخش می‌خواهیم با چند نمونه معروف از ماژول‌های وایرلس آشنا شده و آنها را به کمک آردوینو راه‌اندازی کنیم.

## معرفی پروتکل UART

قبلاً راجع پروتکل‌های سریال و به طور خاص در مورد پروتکل SPI صحبت کردیم. یکی دیگر از پروتکل‌های پرکاربرد در ماژول‌های الکترونیکی، UART است. ویژگی‌های اصلی UART استفاده از تنها دو سیم، عدم نیاز به سیگنال کلاک و پشتیبانی از تنها یک جفت دستگاه فرستنده/گیرنده در آن واحد است. سیم‌های UART با Tx و Rx

که نشان دهنده Transmit و Recieve بوده و به معنی ارسال و دریافت هستند مشخص می شوند. نکته مهم در UART این است که Tx یک دستگاه به Rx دیگری وصل می شود. در آردوینو UNO پایه های ۰ و ۱ دیجیتال از UART پشتیبانی می کنند و کتابخانه Serial برای این اتصال نوشته شده است. لازم است بدانید که سریال مانیتور آردوینو، در واقع داده های پورت UART را دریافت و ارسال می کند و به همین دلیل اگر از سریال مانیتور استفاده می کنید، از پورت های ۰ و ۱ برای دستگاه دیگری استفاده نکنید. توضیحات تکمیلی در مورد پروتکل UART را می توانید در [آموزش آشنایی با پروتکل های ارتباطی](#) مطالعه کنید. در قسمت بعد می خواهیم در مورد ماژول ESP8266 صحبت کنیم که از پروتکل UART برای اتصال به میکروکنترلر استفاده می کند.



## راه اندازی ماژول ESP8266

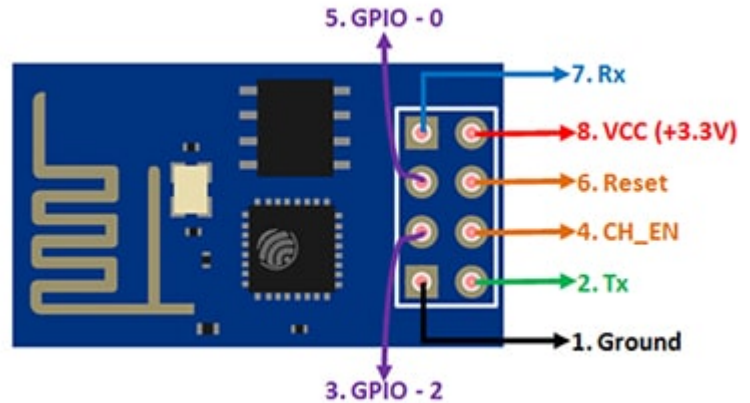
اولین ماژولی که برای ارتباط وای فای به ذهن می آید، یک قطعه معروف به نام ESP8266 ساخت شرکت چینی Espressif است. وای فای پروتکل خاصی از وایرلس است که امکان دسترسی به اینترنت و برقراری ارتباط با روترهای (Router) وای فای را فراهم می کند. این پروتکل برای انجام پروژه های IoT چه مواردی که قصد تبادل اطلاعات در شبکه اینترنت را دارید و چه مواردی که می خواهید یک شبکه محلی از سنسورها و ماژول ها داشته باشید- بسیار مناسب است. تنوع زیادی از ماژول های وای فای وجود دارد و قسمت عمده آنها از ماژول ESP8266 در درون خود استفاده کرده اند.



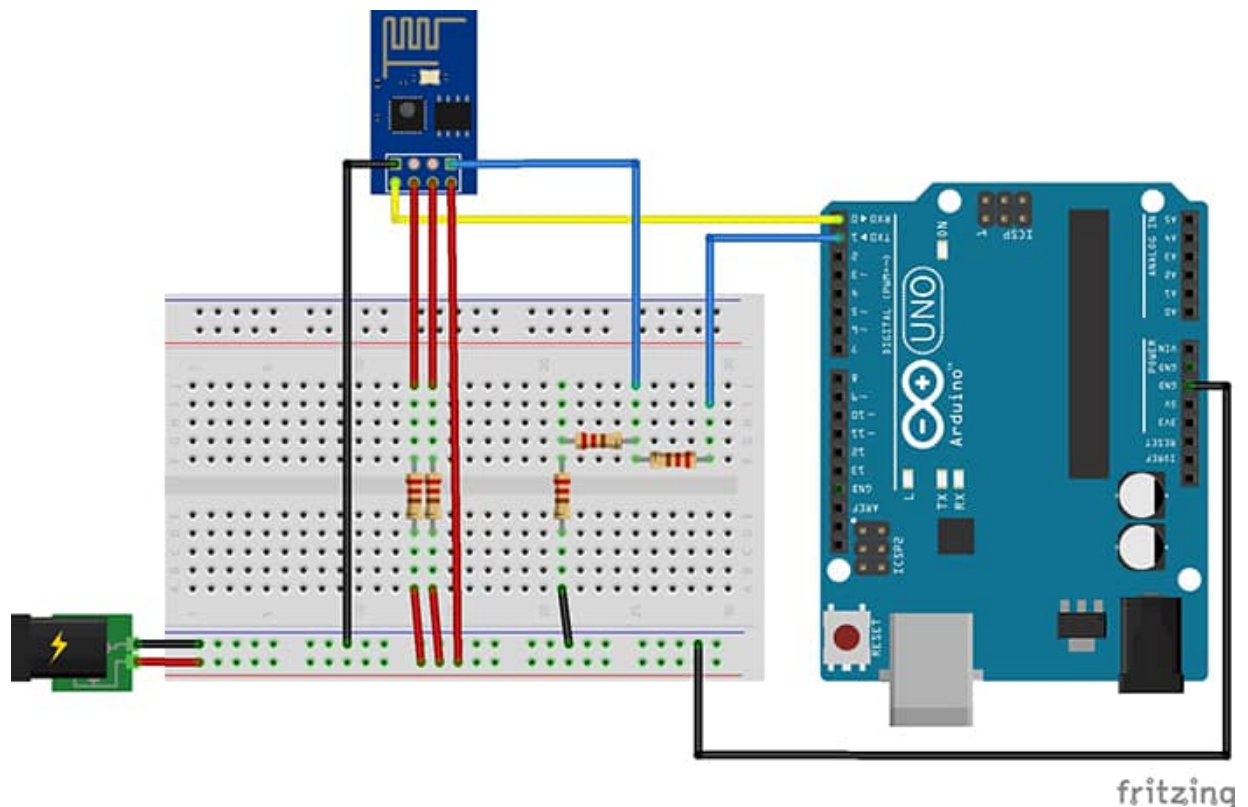
اولین و ساده ترین عضو خانواده ESP، قطعه کوچکی به نام ESP-01 است. این قطعه قابلیت های اولیه اتصال و تبادل اطلاعات در شبکه وای فای را در اختیار می گذارد. این ماژول از پروتکل UART برای ارتباط با میکروکنترلر استفاده می کند. تراشه مرکزی این قطعه (و تقریباً اکثر خانواده ESP) یک IC از نوع ESP8266 است که در حالت پیش فرض دارای سرعت پردازشی 80MHZ، حافظه عملیاتی 32KB و 80KB حافظه برای داده ها است. این مشخصات نشان می دهد که این قطعه بسیار سریعی نسبت به آردوینو است. در ارتباطات وای فای سرعت تبادل اطلاعات بسیار بالاست و به همین دلیل ماژول های آن هم باید سریع باشند. نکته با اهمیت دیگر این

است که ماژول ESP8266 از دو پروتکل مهم در ارتباطات شبکه‌ای یعنی TCP و UDP پشتیبانی می‌کند. در TCP به همراه داده‌های اصلی، داده‌هایی نیز جهت اطمینان از صحت ارسال و دریافت اطلاعات توسط فرستنده و گیرنده مبادله می‌شود. این کار از بروز خطا در اطلاعات جلوگیری می‌کند و در عین حال سرعت انتقال اطلاعات را کاهش می‌دهد. UDP با کنار گذاشتن داده‌های فرعی، سرعت انتقال را بالا برده است البته با پذیرفتن بروز خطا! این روش بیشتر در ارسال داده‌های بلادرنگ (Realtime) مثل پخش تصویری زنده کاربرد دارد.

در این قسمت می‌خواهیم ماژول ESP-01 را به صورت ساده راه‌اندازی کنیم. این ماژول ۸ پایه به صورت زیر دارد:



ابتدا ماژول را به صورت زیر به آردوینو وصل کنید:



نکته قابل توجه در اتصالات ESP-01 این است که تغذیه این ماژول با ولتاژ 3.3V انجام می‌شود. علاوه بر آن سایر پین‌های داده نیز با همین ولتاژ کار می‌کنند و تحمل ولتاژ 5V را ندارند. بنابراین نمی‌توانید آنها را مستقیماً به پین‌های آردوینو وصل کنید؛ چون که ولتاژ High آردوینو (البته بیشتر آردوینوها!) 5V است. به همین دلیل از سه مقاومت برای تقسیم ولتاژ استفاده کرده‌ایم تا ولتاژ 5V آردوینو به 3.3V تبدیل شود. این مشکل برای پین Tx ماژول وجود ندارد! دلیل این است که همانطور که گفتیم ولتاژ پین‌های ماژول 3.3V است و این مقدار از حداقل مقدار ولتاژ برای اینکه آردوینو آن را High تلقی کند (یعنی 3V) بیشتر است. نکته دیگر این است که دو پین CH-EN و Reset باید Pull-up شوند تا ماژول شروع به کار کند. دقت کنید که ماژول ESP مصرف بالایی دارد و برای تامین برق آن بهتر است از یک منبع تغذیه

خارجی استفاده کنید تا آسبایی به برد آردوینو نرسد.

برای کار با ماژول ESP دو روش وجود دارد: استفاده از دستوره‌های AT؛ استفاده از کتابخانه ESP8266. در اینجا نحوه استفاده از دستوره‌های AT را توضیح می‌دهیم و روش دوم را در قسمت راهاندازی برد NodeMCU بررسی خواهیم کرد. AT Command ها مجموعه‌ای از دستوره‌های استاندارد هستند که برای ارتباطات مخابراتی طراحی و تعریف شده‌اند. از طریق این دستورها می‌توانید مستقیماً و بدون هیچ واسطه‌ای با ماژول حرف بزنید. تعداد این دستورها بسیار زیاد است. در اینجا ما از چند دستور مهم AT استفاده کرده‌ایم. برای اطلاع از سایر دستوره‌های AT می‌توانید دفترچه راهنمای آن را مطالعه کنید.

### دفترچه راهنمای دستوره‌های AT

کد زیر را در آردوینو آپلود کنید:

```

/*
SanatBazar
Arduino Tutorial Series
Author: Davood Dorostkar
Website: www.sanatbazar.com
*/

#include <SoftwareSerial.h>
#define RX 2
#define TX 3
SoftwareSerial esp8266(RX, TX);

void setup()
{
  Serial.begin(115200);
  esp8266.begin(115200);
}

void loop()
{
  Serial.println("Please enter your AT command: ");
  while (!Serial.available())
  {
  }
  if (Serial.available() > 0)
  {
    String command = Serial.readStringUntil('\n');
    Serial.println("Entered Command: " + command);
    esp8266.println(command);
  }
  int counter = 0;
  if (esp8266.available() > 0)
  {
    while (esp8266.available() > 0)
    {
      if (counter == 0)
      {
        Serial.println("Recieved Response: ");
      }
      String response = esp8266.readStringUntil('\n');
      Serial.println(response);
      Serial.println();
      counter++;
    }

    Serial.println("=====");
    Serial.println();
  }
}

```

در مورد این برنامه چند نکته قابل ذکر وجود دارد. اول اینکه بر روی برد آردوینو تنها یک رابط سریال UART وجود دارد. در صورتی که بخواهید از سریال برای نمایش و دریافت اطلاعات در سریال مانیتور استفاده کنیم، نمی‌توانید همان پورت را به وسیله‌ای که از پروتکل UART استفاده می‌کند وصل کنید. برای حل این مشکل می‌توانید از کتابخانه SoftwareSerial استفاده کنید. این کتابخانه این امکان را به شما می‌دهد تا به صورت مجازی از پایه‌های دیجیتال دیگر برای ارتباط سریال استفاده کنید. در ابتدای این برنامه یک رابط سریال نرم‌افزاری با استفاده از پایه‌های ۲ و ۳ دیجیتال ایجاد کرده و ماژول را به این رابط وصل می‌کنیم:

```
#include <SoftwareSerial.h>
```

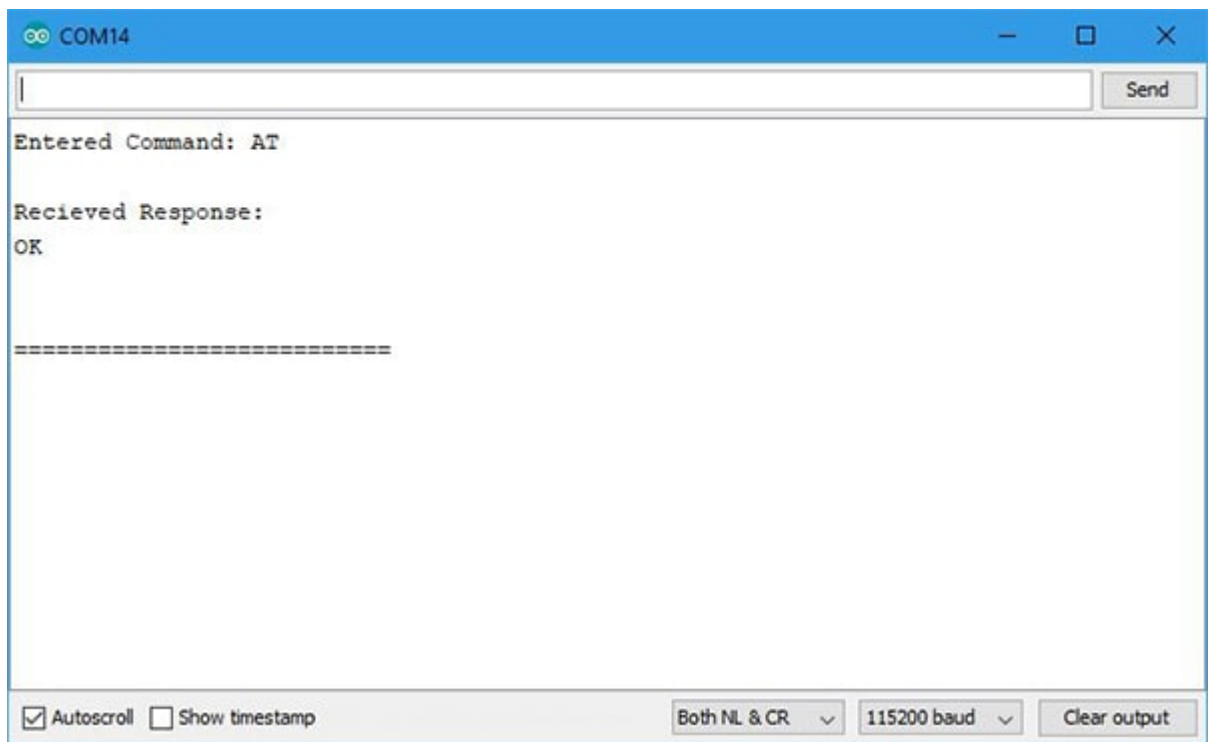


```
#define RX 2
#define TX 3
SoftwareSerial esp8266(RX, TX);
```

در ادامه، برنامه منتظر می‌ماند تا شما دستوری را از طریق سریال وارد کنید. آردوینو این دستور را به ماژول وایرلس می‌فرستد و سپس منتظر می‌ماند تا پاسخی از طرف ماژول فرستاده شود؛ در صورت دریافت پاسخ، نتیجه از طریق سریال نمایش داده می‌شود. دستورهای مورد نظر، همان فرمان‌های AT هستند. در این قسمت از دستور `readStringUntil` استفاده شده است. این دستور داده‌های دریافتی را به صورت رشته‌ای از کارکترها، تا زمانی که کاراکتر خاصی وارد شود، دریافت می‌کند. کاراکتر `\n` به معنای `newline` یا خط جدید است که وقتی کلید `Enter` را فشار می‌دهید ارسال می‌شود. بنابراین برنامه دستورات شما را دریافت کرده و منتظر می‌ماند تا کلید `Enter` را بزنید.

چند دستور اولیه برای کار با این ماژول به صورت زیر است:

دستور `AT` : در صورتی که ماژول ESP به درستی به برد وصل شده باشد، پاسخ `OK` را دریافت خواهید کرد:



دستور `AT+GMR`: این دستور مشخصات ماژول متصل شده را نمایش می‌دهد:

```

COM14
Entered Command: AT+GMR

Recieved Response:
AT+GMR
AT version:1.2.0.0(Jul  1 2016 20:04:45)
SDK version:1.5.4.1(39cb9a32)
Ai-Thinker Technology Co. Ltd.
Dec  2 2016 14:21:16
OK

=====
Autoscroll Show timestamp Both NL & CR 115200 baud Clear output

```

دستور AT+CWLAP: این دستور لیست دستگاه‌های وای‌فای قابل اتصال در دسترس را به همراه مشخصات هر کدام نشان می‌دهد.

```

COM4
Entered Command: AT+CWLAP

Recieved Response:
AT+CWLAP

=====

Response Received:
+CWLAP:(3,"Irancell-TF-001-0001_2",-01,"00:0e:54:00:01:00",1,33,0)
+CWLAP:(1,"shafiee",-01,"00:0e:54:00:01:00",1,33,0)
+CWLAP:(4,"MobinNet2400",-01,"00:0e:54:00:01:00",1,33,0)
+CWLAP:(4,"MobinNet0800",-01,"00:0e:54:00:01:00",1,33,0)
+CWLAP:(0,"PEFWAVE_B0_0000",-01,"00:0e:54:00:01:00",1,-1,0)
+CWLAP:(3,"Irancell-TF-001-0001-0001",-01,"00:0e:54:00:01:00",1,65,0)
+CWLAP:(3,"sahargrapher",-01,"00:0e:54:00:01:00",1,33,0)
+CWLAP:(2,"SanatBazar",-01,"00:0e:54:00:01:00",1,33,0)
+CWLAP:(3,"Golchin",-01,"00:0e:54:00:01:00",1,33,0)
+CWLAP:(3,"Tenda_2F0000",-01,"00:0e:54:00:01:00",1,33,0)
+CWLAP:(3,"SanatBazar",-01,"00:0e:54:00:01:00",4,21,0)
+CWLAP:(3,"akhondradeh",-01,"00:0e:54:00:01:00",4,21,0)
+CWLAP:(3,"Irancell-TF-001-0001-0001",-01,"00:0e:54:00:01:00",4,26,0)
+CWLAP:(3,"Dr.Khaliliaspurg-0001-1-0000",-01,"00:0e:54:00:01:00",8,31,0)
+CWLAP:(0,"ASUS",-01,"00:0e:54:00:01:00",1,33,0)
+CWLAP:(4,"Mohammad",-01,"00:0e:54:00:01:00",1,-1,0)
+CWLAP:(3,"Irancell-TF-001-0001-0001",-01,"00:0e:54:00:01:00",9,15,0)

Autoscroll Show timestamp Both NL & CR 115200 baud Clear output

```

دستور AT+CWLAP="Network Name","Network Password": با استفاده از این دستور می‌توانید به یک روتر وای‌فای وصل شوید:

```

COM14
Entered Command: AT+CWJAP="SanatBazzar", "
Recieved Response:
AT+CWJAP="SanatBazzar", "
OK
=====
Autoscroll Show timestamp Both NL & CR 115200 baud Clear output

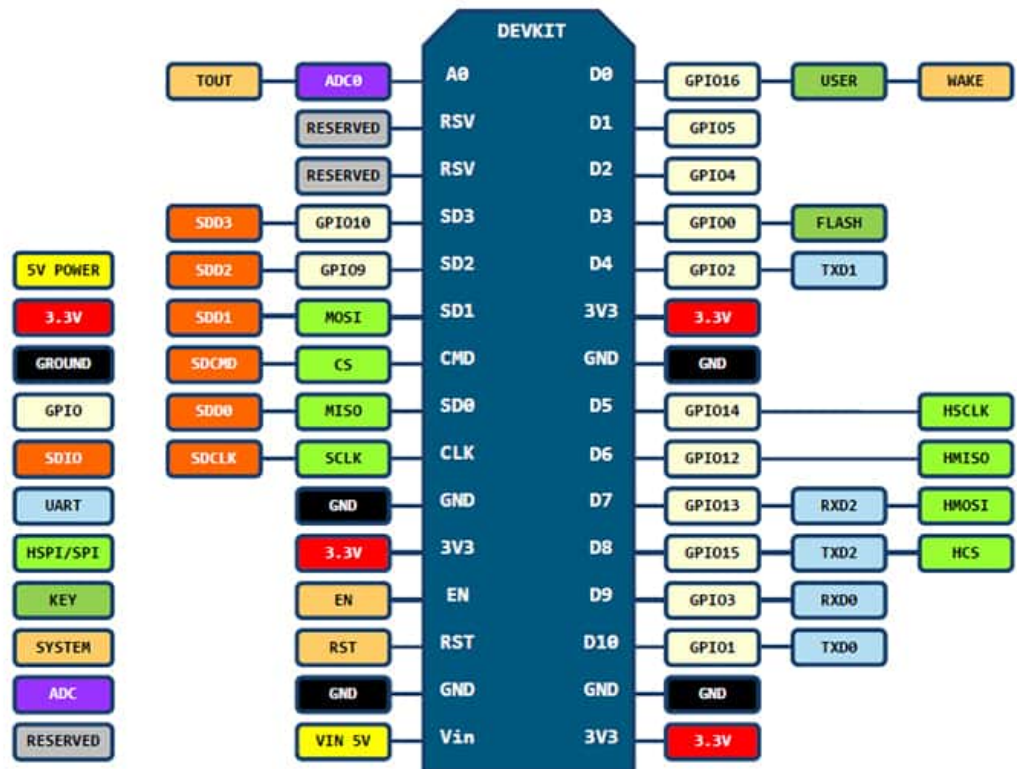
```

## راه اندازی برد NodeMCU

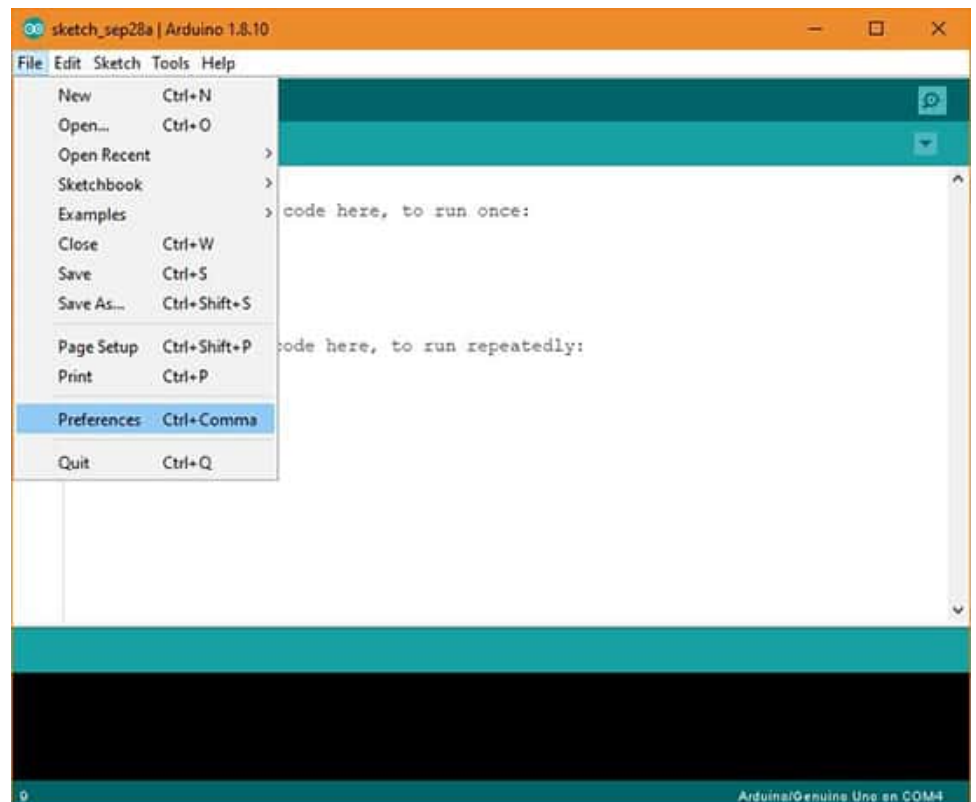
این برد بر خلاف ماژول ESP-01 یک برد توسعه‌ای کامل است. در واقع مثل این است که یک ماژول وای‌فای را با یک آردوینو ترکیب کرده باشید. NodeMCU، یک برد قدرتمند است که بر روی خود تراشه ESP8266 را دارد. علاوه بر آن دارای ۱۲ ورودی/خروجی دیجیتال، ۱ ورودی آنالوگ، 128KB حافظه موقت و 4MB حافظه ذخیره‌سازی است. همچنین NodeMCU از پروتکل‌های UART، SPI، I2C پشتیبانی می‌کند و با استفاده از رابط microUSB به رایانه وصل می‌شود.

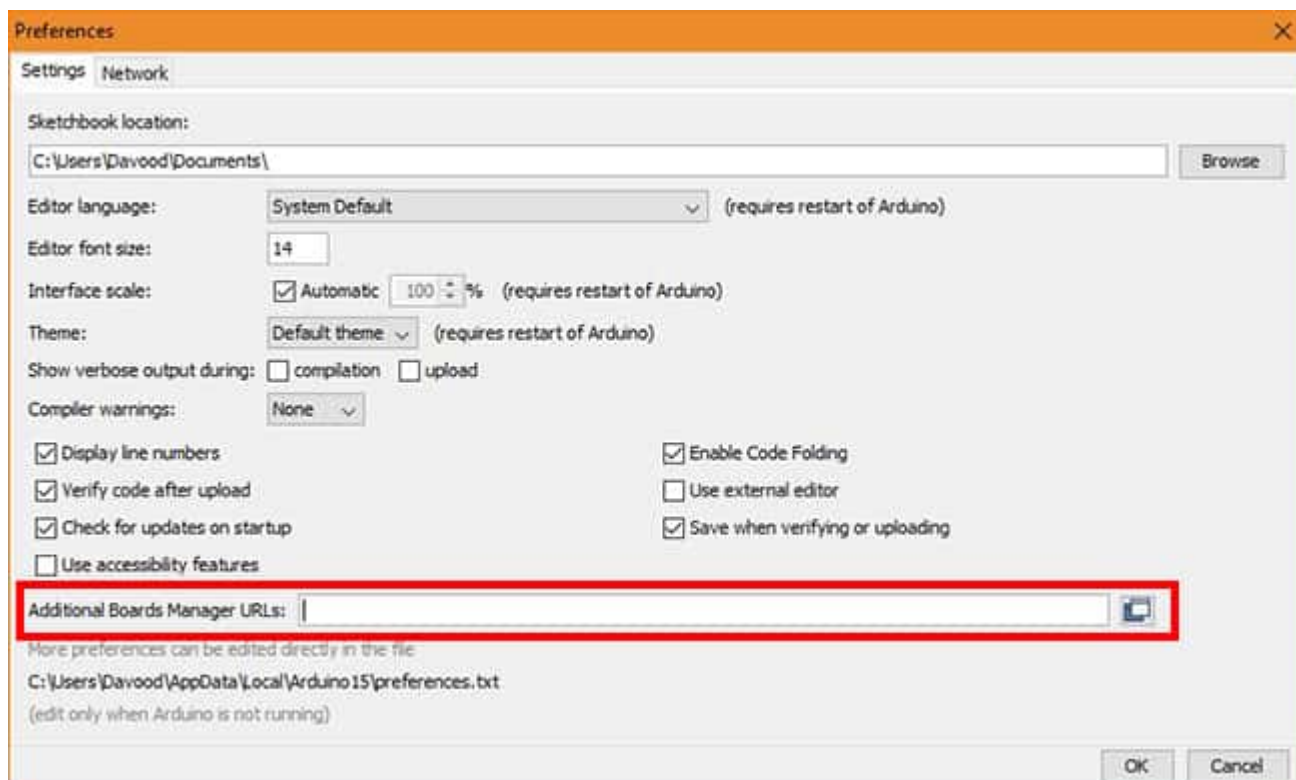




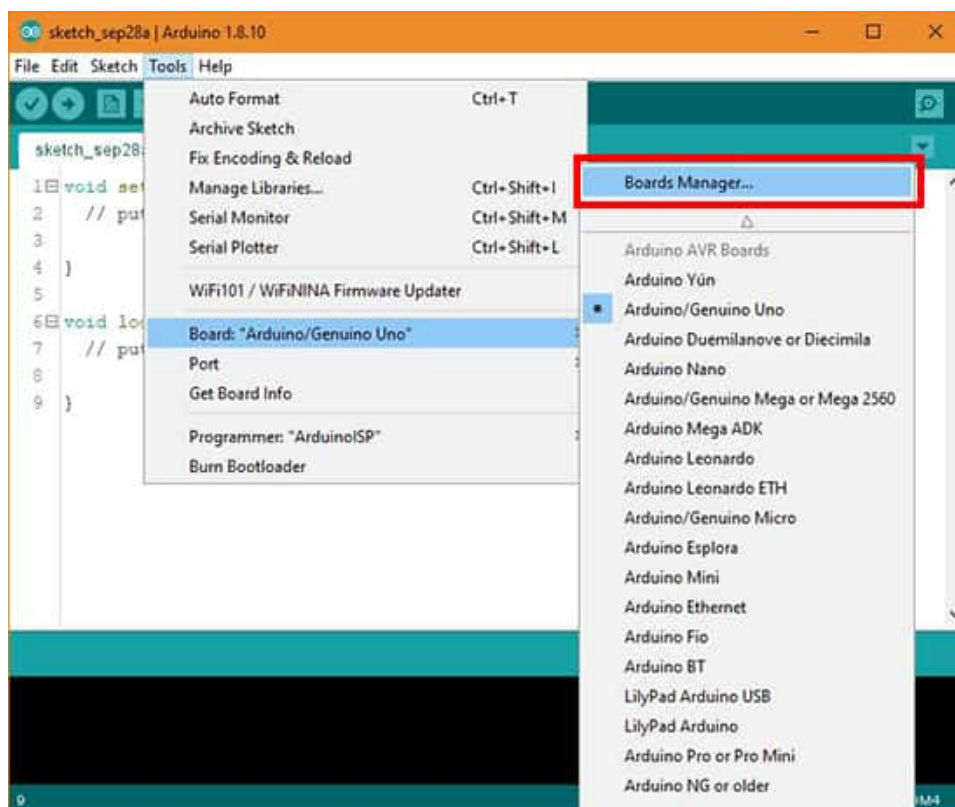


بر روی برد NodeMCU یک LED وجود دارد که به پین D4 متصل است. می‌توانید به همان روشی که برای آردوینو برنامه می‌نویسید، برای NodeMCU نیز نوشته و اجرا کنید. توجه کنید که این LED با دستور LOW روشن شده و با دستور HIGH خاموش می‌شود. برای استفاده از NodeMCU ابتدا باید برد esp8266 را در قسمت board manager آردوینو نصب کرده و نیز کتابخانه ESP6288WiFi را اضافه کنید. در نرم‌افزار آردوینو بر روی File > Preferences کلیک کرده و در قسمت Additional Boards Manager آدرس زیر را وارد کرده و OK را بزنید.





سپس از قسمت Tools → Board → Boards Manager عبارت esp8266 را جستجو کرده و کتابخانه مورد نظر را نصب کنید.

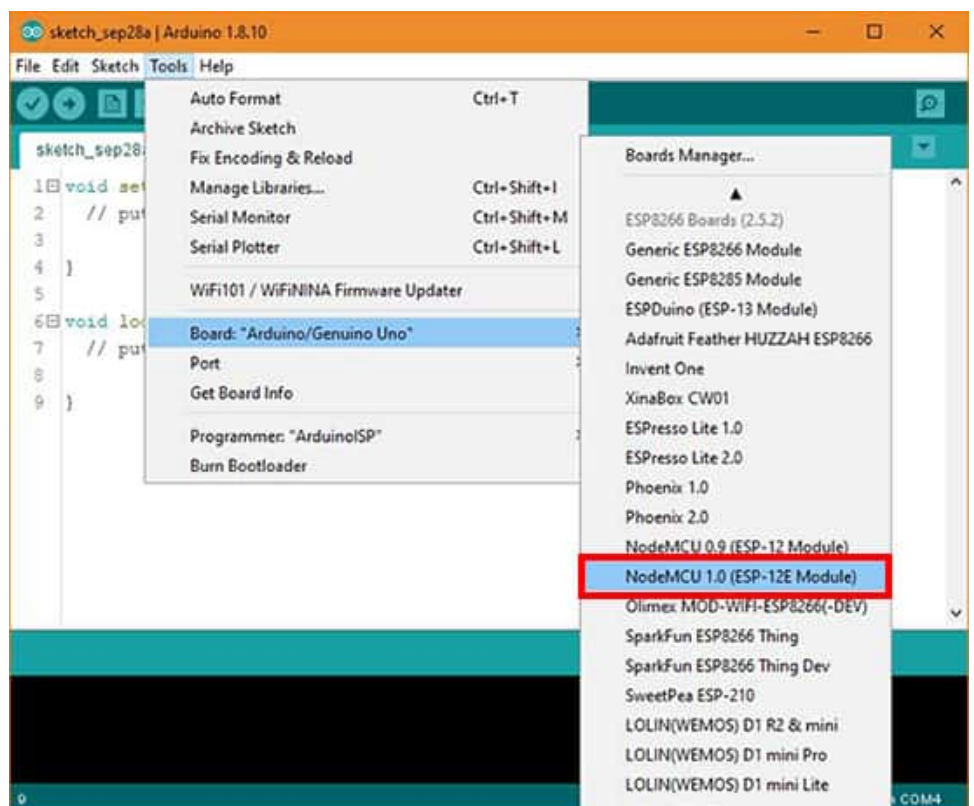




سپس کتابخانه ESP8266 را به آردوینو اضافه کنید. این کتابخانه را می‌توانید از لینک زیر نیز دانلود کنید.

#### [دانلود کتابخانه ESP8266](#)

در آخر از قسمت Tools > Board برد NodeMCU 1.0 را انتخاب کنید.



حالا برنامه زیر را بر روی برد NodeMCU آپلود کنید. معمولا آپلود کد بر روی این برد کمی بیشتر از آردوینو طول می‌کشد؛ البته نه چندان زیاد، نگران نباشید! بر روی برد NodeMCU یک LED آبی رنگ وجود دارد SanatBazar که زمانی که برنامه را آپلود می‌کنید، شروع به چشمک زدن می‌کند.

```

Author: Davood Dorostkar
Website: www.sanatzbazar.com

*/

void setup()
{
    pinMode(D4, OUTPUT);
}

void loop()
{
    digitalWrite(D4, HIGH);
    delay(500);
    digitalWrite(D4, LOW);
    delay(1000);
}

```

کار با این کتابخانه بسیار ساده است، فقط کافیسیت که توابع مهم آن را بشناسید. برای مثال برنامه زیر ماژول را به روتر وای‌فای تان وصل می‌کند:

```

#include <ESP8266WiFi.h>
/*
SanatBazar
Arduino Tutorial Series
Author: Davood Dorostkar
Website: www.sanatzbazar.com

*/

void setup()
{
    delay(1000);
    Serial.begin(115200);
    WiFi.begin("SanatBazar", "*****");
    Serial.print("Connecting ");
    while (WiFi.status() != WL_CONNECTED)
    {
        Serial.print(".");
        delay(1000);
    }
    Serial.println();
    Serial.println("Connected Successfully!");
    Serial.print("Module's IP: ");
    Serial.println(WiFi.localIP());
}

void loop()
{
}

```

سرعت پیش‌فرض کارخانه برای ماژول وای‌فای معمولاً ۱۱۵۲۰۰ است، به همین دلیل سرعت پورت سریال را هم همین مقدار قرار داده‌ایم. قبل از هر چیز برای اتصال به یک روتر باید دستور زیر را وارد کنید.

```
WiFi.begin("Network Name", "Network Password")
```

توجه: حتما نام و رمز عبور روترتان را جایگزین مقادیر بالا کنید.

دستور `WiFi.status()` وضعیت اتصال ماژول را به صورت یکی از حالات زیر نشان می‌دهد:

توضیحات	وضعیت	شماره وضعیت
زمانی که ماژول در حال تغییر بین دو حالت مختلف است	IDLE_STATUS	0
زمانی که روتر مورد نظر شما در دسترس نباشد	WL_NO_SSID_AVAIL	1
زمانی که ارتباط وای‌فای به درستی برقرار شود	WL_CONNECTED	3

زمانی که رمز عبور اشتباه وارد شده باشد	WL_CONNECT_FAILED	4
زمانی که ارتباطی برقرار نباشد	WL_DISCONNECTED	6

اگر دستور `WiFi.status()` را اجرا کنید، یکی از اعداد فوق را به شما بر می‌گرداند، اما برای راحتی کار با آن می‌توانید در کدتان از عبارتهای فوق استفاده کنید و برد NodeMCU معنای آن را می‌فهمد؛ مانند آنچه که در برنامه آمده بود.

در ادامه، برنامه منتظر می‌ماند تا به روتر شما وصل شود. زمانی که اتصال برقرار شد، پیغام موفقیت در اتصال به همراه IP ماژول نشان داده می‌شود. در هر شبکه داخلی، تعدادی وسیله مختلف به یکدیگر متصل هستند. هر کدام از این وسایل با یک آدرس به نام IP شناخته می‌شوند. IP را سرور اصلی به دستگاه‌ها اختصاص می‌دهد. این آدرس معمولاً با 192.168.1 شروع می‌شود. پنجره سریال مانیتور را باز کرده و baud rate را روی 115200 قرار دهید. در اینجا ماژول NodeMCU به روتر وصل شده و آدرس 192.168.1.15 را دریافت کرده است.

```

COM13
Connecting .....
Connected Successfully!
Module's IP: 192.168.1.15
  
```

The screenshot shows a serial terminal window titled 'COM13'. The output text reads: 'Connecting .....', 'Connected Successfully!', and 'Module's IP: 192.168.1.15'. At the bottom of the window, there are control options: 'Autoscroll' (checked), 'Show timestamp' (unchecked), 'Both NL & CR' (selected), '115200 baud' (selected), and a 'Clear output' button.

ارتباط وای‌فای برقرار شد. حالا می‌توانید تمام چیزهایی که دوست دارید (سنسورها، ماژول‌ها و برنامه‌های مختلف) را از طریق شبکه به هم وصل کنید.

برد NodeMCU می‌تواند در دو حالت مختلف کار کند: Station و Station. Access Point. دستگاهی است که برای اتصال وای‌فای نیاز به یک وسیله دیگر مثل یک روتر دارد. Access Point دستگاهی است که دستگاه‌های دیگر به آن وصل می‌شوند. معمولاً وسیله‌ای که به یک اتصال سیمی وصل شده و شبکه را از طریق وایرلس به اشتراک بگذارد را Access Point می‌گویند. در صورتی که این دستگاه از طریق سیم به جایی وصل نشده باشد و کل ارتباطات به صورت بی‌سیم باشد، به آن Soft Access Point یا Soft AP می‌گویند.



NodeMCU می‌تواند هر کدام از نقش‌های Station یا Soft AP را داشته باشد. تا اینجا باید متوجه شده باشید که برنامه قبل در حالت Station کار می‌کرد. در اینجا



نشان می‌دهیم که چطور NodeMCU را به یک Soft AP تبدیل می‌کنیم. کد زیر را در برنامه آپلود کنید:

```

/*
SanatBazar
Arduino Tutorial Series
Author: Davood Dorostkar
Website: www.sanatbazar.com

*/

#include <ESP8266WiFi.h>
String networkName = "NodeMCU";
String networkPassword = "12345678";

void setup()
{
  Serial.begin(115200);
  Serial.println();
  Serial.print("Configuring WiFi access point...");
  if (WiFi.softAP(networkName, networkPassword) == true)
  {
    IPAddress softAP = WiFi.softAPIP();
    Serial.println("Soft AP created successfully!");
    Serial.println("You can connect to the network using: ");
    Serial.println("network: " + networkName);
    Serial.println("Password: " + networkPassword);
    Serial.println("Host IP Address is: ");
    Serial.println(softAP);
    Serial.println();
  }
  else
  {
    Serial.println("Could not creat AP!");
  }
}

void loop()
{
  delay(1000);
  Serial.print("Number of connected devices (stations)");
  Serial.println(WiFi.softAPgetStationNum());
}

```

در این برنامه ابتدا باید یک AP ایجاد کنید. در صورت موفقیت در ایجاد AP مشخصات آن نمایش داده می‌شود:

```

if (WiFi.softAP(networkName, networkPassword) == true)
{
  IPAddress softAP = WiFi.softAPIP();
  Serial.println("Soft AP created successfully!");
  Serial.println("You can connect to the network using: ");
  Serial.println("network: " + networkName);
  Serial.println("Password: " + networkPassword);
  Serial.println("Host IP Address is: ");
  Serial.println(softAP);
  Serial.println();
}

```

```

COM13
Send

Configuring WiFi access point...Soft AP created successfully!
You can connect to the network using:
network: NodeMCU
Password: 12345678
Host IP Address is:
192.168.4.1

Number of connected devices (stations)0
Number of connected devices (stations)0
Number of connected devices (stations)0
Number of connected devices (stations)1
Number of connected devices (stations)1
Number of connected devices (stations)1
Number of connected devices (stations)1

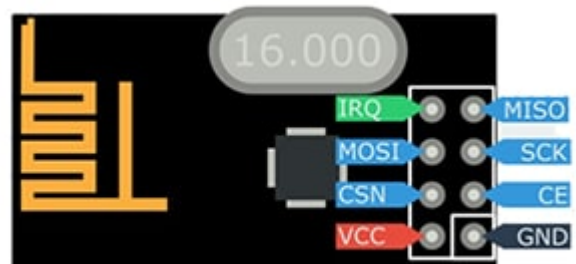
 Autoscroll  Show timestamp
Both NL & CR 115200 baud Clear output

```

پس از آن هر یک ثانیه یکبار، تعداد دستگاه‌هایی که به آن وصل شده‌اند را نشان می‌دهد. در آن واحد 5 دستگاه می‌توانند به NodeMCU وصل شوند. با این روش می‌توانید یک ماژول را به صورت AP و دستگاه دیگری را به صورت Station راه‌اندازی کرده و بدون نیاز به روتر خارجی، چند ماژول را به صورت وایرلس به هم وصل کنید.

## راه اندازی اولیه ماژول nRF

یکی دیگر از ماژول‌های وایرلس کاربردی و خوش‌ساخت، سری ماژول‌های nRF محصول شرکت نروژی nordic semiconductor هستند. این ماژول‌ها به دلیل برد زیاد و کم مصرف بودن معروف هستند. ماژول nRF24L01 یک فرستنده و گیرنده وایرلس ساده و در دسترس است. این ماژول امکان برقراری ارتباط تا فاصله 100 متر و با پهنای باندهای مختلف از 256Kbps تا 2Mbps را فراهم می‌کند. البته این فاصله برای فضای باز است و طبیعتاً در فضاهای بسته این مقدار کاهش می‌یابد. این ماژول از طریق پروتکل پرسرعت SPI با میکروکنترلر ارتباط برقرار می‌کند. این پروتکل قبلاً در [آموزش راه‌اندازی کارت SD](#) معرفی شد.

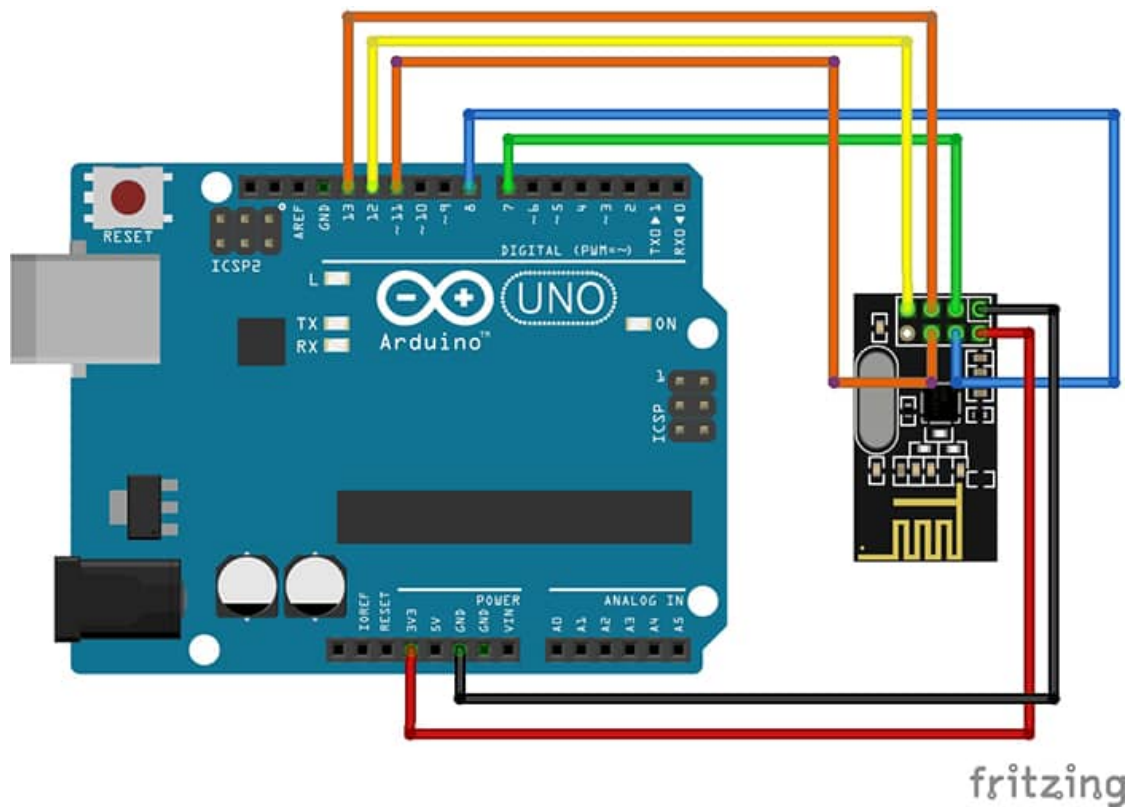


این ماژول به طرز باورنکردنی کم مصرف است. مصرف جریان nRF24L01 در حالت آماده به کار نزدیک به صفر است و تنها در حالت دریافت سیگنال با بیشترین نرخ داده (2Mbps) حدود 13.5mA جریان می‌کشد که باز هم برای یک ماژول وایرلس مقدار بسیار ناچیزی است. این ویژگی به شما کمک می‌کند به راحتی بتوانید از آن در پروژه‌های قابل حمل که انرژی از طریق باتری تامین می‌شود استفاده کنید. از این نظر nRF24L01 به هیچ وجه قابل مقایسه با ماژول‌های ESP نیست. البته فراموش نکنید که nRF یک ماژول وایرلس است که قابلیت WiFi ندارد. طبیعی است که یک ماژول وای‌فای پرمصرف باشد. nRF24L01 از یک آنتن درون خود بهره می‌برد. این آنتن کوچک که به صورت یک مسیر مارپیچ بر روی برد مشخص است برای کاربردهای معمولی کاملاً مناسب است. برای کاربردهای خاص، نمونه‌هایی از ماژول nRF وجود دارد که قابلیت نصب آنتن خارجی بر روی آن وجود دارد و به کمک آنتن مخصوص، برد آن تا 1 کیلومتر هم می‌رسد.



ولتاژ کاری این ماژول بین 1.9V تا 3.6V است. با این وجود قابلیت اتصال به ولتاژ 5V را نیز دارد. ماژول nRF در بیشترین حالت 13.5mA جریان مصرف می‌کند که این مقدار بسیار کمتر از 40mA تامین جریان در هر پایه دیجیتال آردینو و 200mA در پایه‌های ولتاژ آن است. بنابراین با خیال راحت می‌توانید پین‌های ماژول را مستقیماً به آردینو وصل کرده و نگران تامین توان آن نباشید.

در اینجا می‌خواهیم یک ارتباط ساده بین دو ماژول nRF ایجاد کرده و پیامی را بین آنها مبادله کنیم. برای این کار علاوه بر دو ماژول nRF (فرستنده و گیرنده) به دو برد آردینو نیاز داریم که پردازش اطلاعات و انجام دستورات را بر عهده دارند. اتصالات هر کدام از مدارهای فرستنده و گیرنده را مطابق شکل زیر انجام دهید:



برای کار با ماژول nRF از کتابخانه RF استفاده می‌کنیم. کار کردن با این کتابخانه ساده است، تنها باید با توابع آن آشنا شوید. بر روی یکی از آردوینوها به عنوان فرستنده برنامه زیر را آپلود کنید:

```

/*
SanatBazar
Arduino Tutorial Series
Author: Davood Dorostkar
Website: www.sanatbazar.com

*/

#include <RF24.h>
RF24 wirelessSender(7, 8); // (CE , CSN) pins
const byte address[] = "00001";

void setup()
{
    Serial.begin(9600);
    wirelessSender.begin();
    Serial.println("Initializing the wireless... ");
    wirelessSender.setPALevel(RF24_PA_MIN);
    Serial.println("The Amplification is set to: " +
String(wirelessSender.getPALevel()));
    wirelessSender.openWritingPipe(address);
    wirelessSender.stopListening();
    Serial.println("wireless initialized!");
}

void loop()
{
    char message[] = "I am sending something!";
    wirelessSender.write(&message, sizeof(message));
    delay(2000);
}

```

در ابتدا باید ماژول وایرلس را از طریق پایه‌های CE و CSN تعریف کنید. در اینجا این دو پایه به پایه‌های 7 و 8 آردوینو وصل شده‌اند. این ماژول می‌تواند در آن واحد در چندین ماژول دیگر ارتباط داشته باشد؛ بنابراین برای مشخص شدن هر ارتباط از یک آدرس استفاده می‌شود. آدرس ماژول nRF یک عدد 5 بیتی است.

```

#include <RF24.h>
RF24 wifiSender(7, 8); // (CE , CSN) pins
const byte address[] = "00001";

```

برای راهاندازی ماژول نیز چند دستور باید در حلقه setup اجرا شود. با دستور begin ماژول شروع به کار می‌کند. دستور setPALevel میزان انرژی مصرفی ماژول را مشخص می‌کند و تابع getPALevel سطح انرژی فعلی ماژول را به صورت عددی بین 0 تا 3 می‌دهد. هرچه فاصله بین دو ماژول بیشتر باشد باید انرژی بیشتری مصرف شود. این تابع از چهار حالت مختلف پشتیبانی می‌کند که از کم‌مصرف‌ترین تا پر مصرف‌ترین وضعیت به ترتیب عبارتند از:

RF24\_PA\_MIN

RF24\_PA\_LOW

RF24\_PA\_HIGH

RF24\_PA\_MAX

```

wifiSender.begin();
wifiSender.setPALevel(RF24_PA_MIN);
Serial.println("The Amplification is set to: " + String(wifiSender.getPALevel()));

```

همان طور که در کد بالا می‌بینید، می‌توان چند رشته را با عملگر + به یکدیگر وصل کرد. حالا باید ماژول را در حالت ارسال داده قرار داده و از حالت خواندن خارج کنیم:



```
wifiSender.openWritingPipe(address);
wifiSender.stopListening();
```

در حلقه loop نیز ابتدا یک متغیر کاراکتری که حاوی پیام مورد نظر است ایجاد شده و با دستور write از طریق وایرلس فرستاده می‌شود. این تابع طوری نوشته شده که ورودی اول آن آدرس پیغام ارسالی و ورودی دوم آن طول آن است. در زبان برنامه‌نویسی C++ آدرس یک متغیر به صورت زیر نشان داده می‌شود:

```
&Variable
```

بر روی آردوینوی دیگر نیز به عنوان گیرنده برنامه زیر را آپلود کنید:

```
/*
SanatBazar
Arduino Tutorial Series
Author: Davood Dorostkar
Website: www.sanatbazar.com
*/

#include <RF24.h>
RF24 wirelessReceiver(7, 8); // (CE , CSN) pins
const byte address[] = "00001";

void setup()
{
  Serial.begin(9600);
  wirelessReceiver.begin();
  Serial.println("Initializing the wireless... ");
  wirelessReceiver.setPALevel(RF24_PA_MIN);
  Serial.println("The Amplification is set to: " +
String(wirelessReceiver.getPALevel()));
  wirelessReceiver.openReadingPipe(0, address);
  wirelessReceiver.startListening();
  Serial.println("wireless initialized!");
}

void loop()
{
  if (wirelessReceiver.available())
  {
    char message[32] = "";
    wirelessReceiver.read(&message, sizeof(message));
    Serial.println(message);
  }
}
```

برنامه گیرنده تقریباً مشابه برنامه فرستنده است و تنها چند تفاوت جزئی دارد. اول اینکه در حلقه setup باید ماژول را به حالت خواندن داده‌ها تغییر دهید:

```
wifiReceiver.openReadingPipe(0, address);
wifiReceiver.startListening();
```

سپس یک متغیر کاراکتری خالی تعریف کرده و داده دریافت شده از وایرلس را در آن ذخیره کرده و در نهایت چاپ کنید:

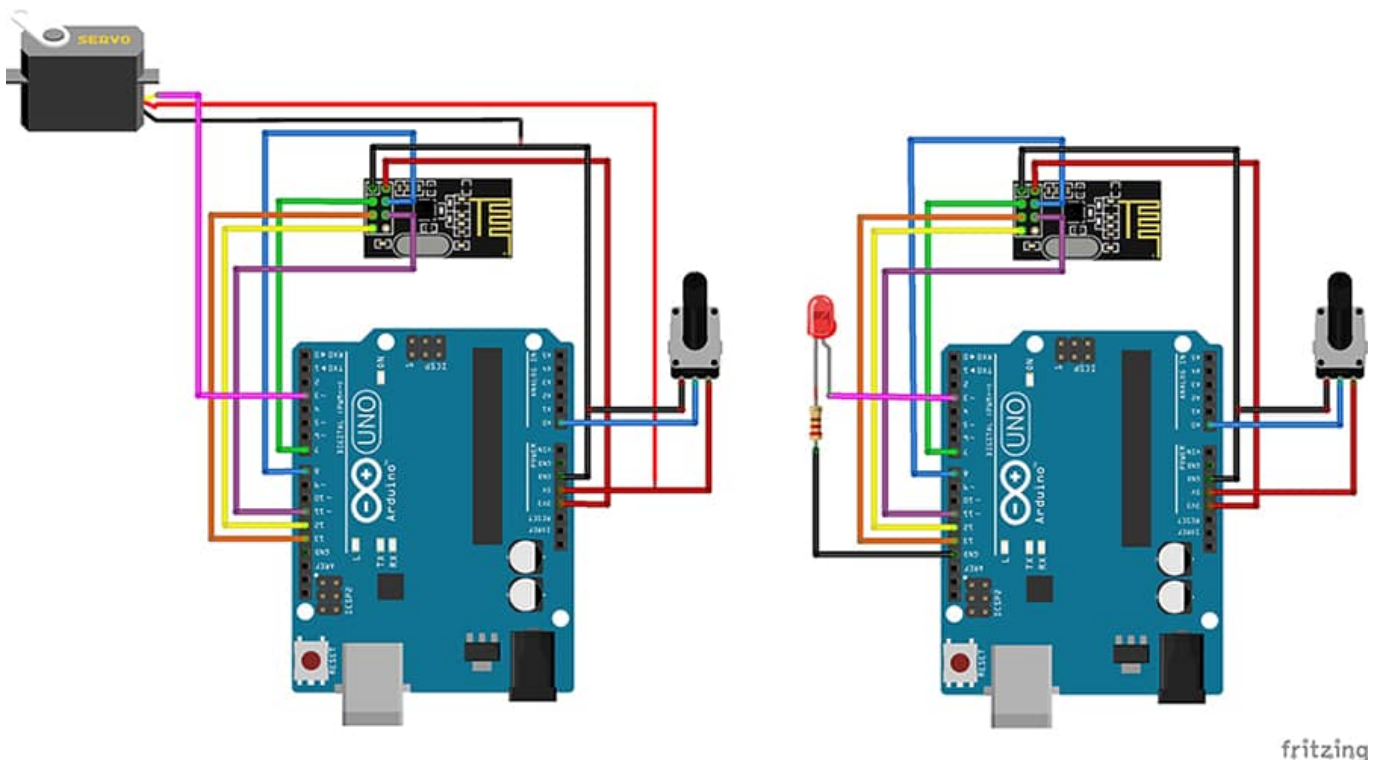
```
char message[32] = "";
wifiReceiver.read(&message, sizeof(message));
Serial.println(message);
```

## ارتباط دوطرفه ماژولها در شبکه وایرلس

یکی از کارهای جالبی که می‌توان با ماژول وایرلس انجام داد، ایجاد شبکه‌ای از این ماژول‌های متصل به هم و انتقال داده بین آنهاست. از این طریق می‌توانید مجموعه‌ای از سنسورها و عملگرهای مختلف را به هم متصل کنید و یک سیستم هوشمند (مثلا یک خانه هوشمند) بسازید. به عنوان یک مثال ساده در این قسمت قصد دارم به شما نشان دهم که چطور می‌توانید یک موتور و یک LED را از طریق وایرلس کنترل کنید. نحوه اتصال تمام قطعات را قبلا یاد گرفته‌اید و در این قسمت نکته سخت‌افزاری جدیدی وجود ندارد.

یک ماژول nRF را مانند قبل به برد آردوینو وصل کنید. به علاوه، یک سروو موتور را به پایه 3 دیجیتال و یک پتانسیومتر را به پایه A0 آنالوگ آردوینو وصل کنید. یک ماژول nRF دیگر را نیز به یک برد آردوینوی دیگر وصل کنید. همچنین یک LED را به همراه یک مقاومت  $220\Omega$  به پایه 3 دیجیتال و یک پتانسیومتر را نیز به پایه A0 آنالوگ آردوینو وصل کنید.

برد اول فرمان میزان روشنایی LED را از طریق وایرلس به برد دوم می‌فرستد و همچنین فرمان زاویه چرخش سروو موتور را گرفته و موتور را به حرکت در می‌آورد. برد دوم برعکس موارد فوق را انجام می‌دهد؛ یعنی فرمان زاویه چرخش موتور را از طریق وایرلس ارسال می‌کند و فرمان میزان روشنایی LED را دریافت کرده آن را روشن می‌کند.



fritzing

برنامه زیر را بر روی برد اول آپلود کنید:

```

/*
SanatBazar
Arduino Tutorial Series
Author: Davood Dorostkar
Website: www.sanatbazar.com

*/

#include <RF24.h>
#include <Servo.h>
#define servoPin 3
#define potentiometer A0
RF24 wireless(7, 8); // (CE , CSN) pins
Servo servoMotor;
const byte LEDAddress[] = "00001";
const byte servoAddress[] = "00002";
int servoStatus;
int LEDStatus;

void readWireless()
{

```

```

    wireless.startListening();
    wireless.read(&servoStatus, sizeof(servoStatus));
}

void writeWireless()
{
    wireless.stopListening();
    LEDStatus = map(analogRead(potentiometer), 0, 1023, 0, 255);
    if (LEDStatus < 10)
        LEDStatus = 0;
    wireless.write(&LEDStatus, sizeof(LEDStatus));
}

void setup()
{
    Serial.begin(9600);
    wireless.begin();
    Serial.println("Initializing the wireless... ");
    wireless.setPALevel(RF24_PA_MIN);
    wireless.openReadingPipe(1, servoAddress);
    wireless.openWritingPipe(LEDAddress);
    Serial.println("The Amplification is set to: " + String(wireless.getPALevel()));
    Serial.println("wireless initialized!");
    servoMotor.attach(servoPin);
}

void loop()
{
    delay(50);
    writeWireless();
    delay(50);
    readWireless();
    servoMotor.write(servStatus);
    Serial.println(servStatus);
}

```

برنامه زیر را نیز بر روی برد دوم آپلود کنید:

```

/*
SanatBazar
Arduino Tutorial Series
Author: Davood Dorostkar
Website: www.sanatbazar.com

*/

#include <RF24.h>
#define LEDpin 3
#define potentiometer A0
RF24 wireless(7, 8); // (CE , CSN) pins
const byte LEDAddress[] = "00001";
const byte servoAddress[] = "00002";
int servoStatus;
int LEDStatus;

void readWireless()
{
    wireless.startListening();
    wireless.read(&LEDStatus, sizeof(LEDStatus));
}

void writeWireless()
{
    wireless.stopListening();
    servoStatus = map(analogRead(potentiometer), 0, 1023, 0, 180);
    if (servoStatus < 10)
        servoStatus = 0;
    else if (servoStatus > 170)
        servoStatus = 180;
    wireless.write(&servoStatus, sizeof(servStatus));
}

void setup()
{
    Serial.begin(9600);
    wireless.begin();
    Serial.println("Initializing the wireless... ");
    wireless.setPALevel(RF24_PA_MIN);

```

```
wireless.openReadingPipe(1, LEDAddress);
wireless.openWritingPipe(servoAddress);
Serial.println("The Amplification is set to: " + String(wireless.getPALevel()));
Serial.println("wireless initialized!");
pinMode(LEDpin, OUTPUT);
}

void loop()
{
  delay(50);
  writeWireless();
  delay(50);
  readWireless();
  analogWrite(LEDpin, LEDStatus);
  Serial.println(LEDStatus);
}
```

برنامه آردوینو در این حالت تفاوت زیادی با حالت قبل ندارد. مهمترین تفاوت این است که باید دو مسیر انتقال داده، یکی برای انتقال دستورات حرکتی موتور و دیگری برای انتقال فرمان میزان روشنایی LED تعریف کنید. برد اول فرمان روشنایی LED را از طریق یک مسیر فرستاده و برد دوم فرمان را از همان مسیر دریافت می‌کند. مشابه همین اتفاق برای مسیر دوم (کنترل سروو موتور) نیز تکرار می‌شود. نام‌گذاری‌ها در این پروژه اهمیت زیادی دارد و در صورت وجود اشتباه، سیستم کار نخواهد کرد. در اینجا ما تنها دو ماژول را به هم وصل کردیم، اگر تعداد ماژول‌ها بیشتر شود، نام‌گذاری‌ها نیاز به دقت بیشتری نیز دارد.

```
const byte LEDAddress[] = "00001";
const byte servoAddress[] = "00002";
```

از آنجایی که ارسال و دریافت اطلاعات با وایرلس نیازمند اجرای چند دستور است، برای سادگی برنامه، دو تابع یکی برای ارسال و دیگری برای دریافت اطلاعات تعریف شده است. این دو تابع برای برد اول به صورت زیر:

```
void readWireless()
{
  wireless.startListening();
  wireless.read(&servoStatus, sizeof(servoStatus));
}

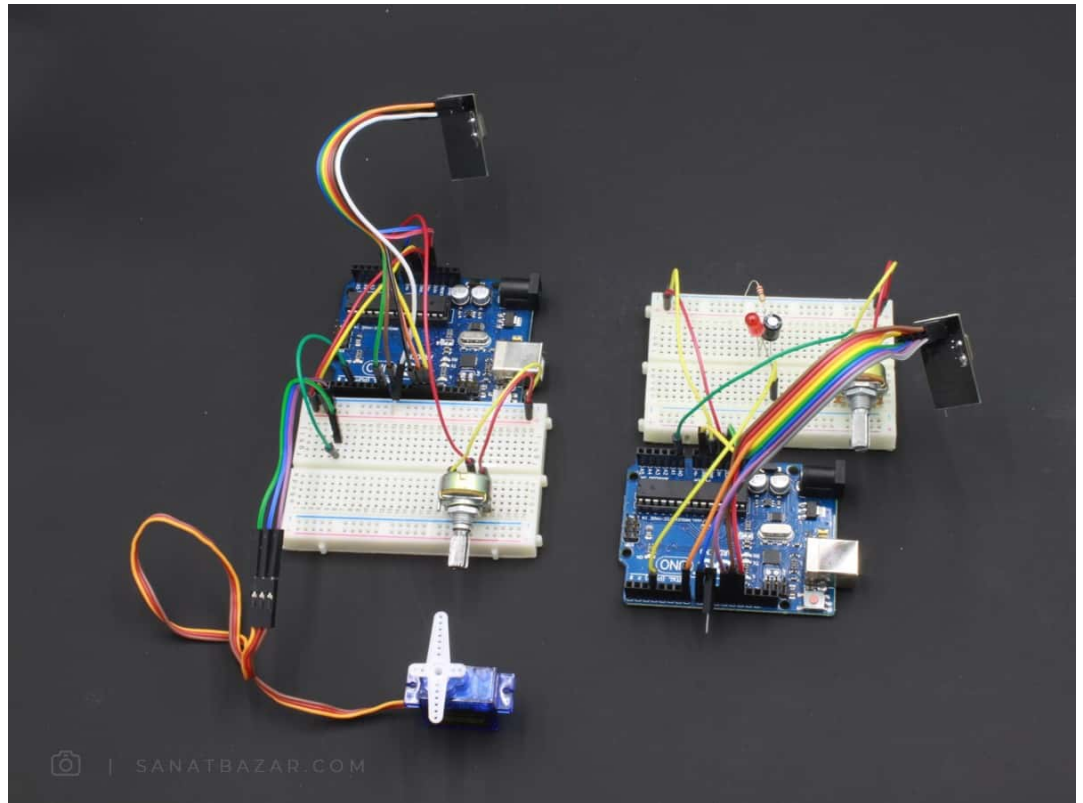
void writeWireless()
{
  wireless.stopListening();
  LEDStatus = map(analogRead(potentiometer), 0, 1023, 0, 255);
  if (LEDStatus < 10)
    LEDStatus = 0;
  wireless.write(&LEDStatus, sizeof(LEDStatus));
}
```

و برای برد دوم به صورت زیر است:

```
void readWireless()
{
  wireless.startListening();
  wireless.read(&LEDStatus, sizeof(LEDStatus));
}

void writeWireless()
{
  wireless.stopListening();
  servoStatus = map(analogRead(potentiometer), 0, 1023, 0, 180);
  if (servoStatus < 10)
    servoStatus = 0;
  else if (servoStatus > 170)
    servoStatus = 180;
  wireless.write(&servoStatus, sizeof(servoStatus));
}
```

همان طور که مشاهده می‌کنید، مقادیر مرزی سروو موتور و LED فیلتر شده است تا نتیجه بهتری به دست بیاید. همچنین برای اینکه نوسانات جریان باعث چشمک‌های ناخواسته LED نشود، می‌توانید یک خازن 100nF بین دو سر آن قرار دهید.



## نتیجه گیری

در این آموزش با مهمترین ماژول‌های وایرلس و وای‌فای آشنا شدید و مباحث مقدماتی برای کار با آنها را یاد گرفتید. همچنین با یک پروتکل سریال دیگر به نام UART تا حدی آشنا شدید.

در آموزش بعدی، [نحوه کار با ماژول بلوتوث](#) را خواهید آموخت.

نظرات شما باعث بهبود محتوای آموزشی ما می‌شود. اگر این آموزش را دوست داشتید، همین‌طور اگر سوالی در مورد آن دارید، از شنیدن نظراتتان خوشحال خواهیم شد.