



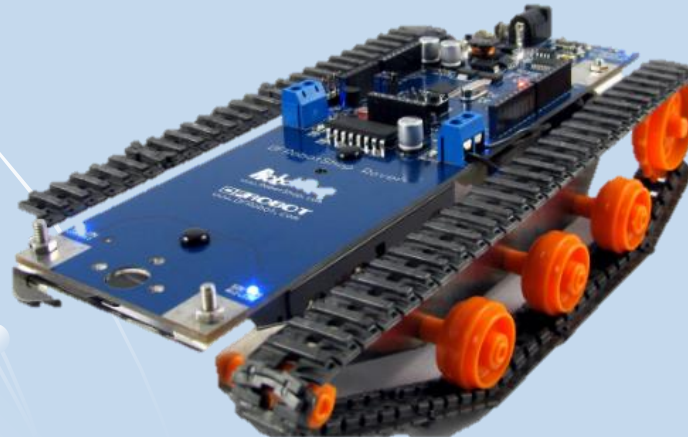
دانشکده مهندسی کامپیوتر

درس برنامه نویسی تجهیزات اینترنت اشیا

نمیسال دوم ۱۴۰۲

برنامه نویسی ماژول وای فای

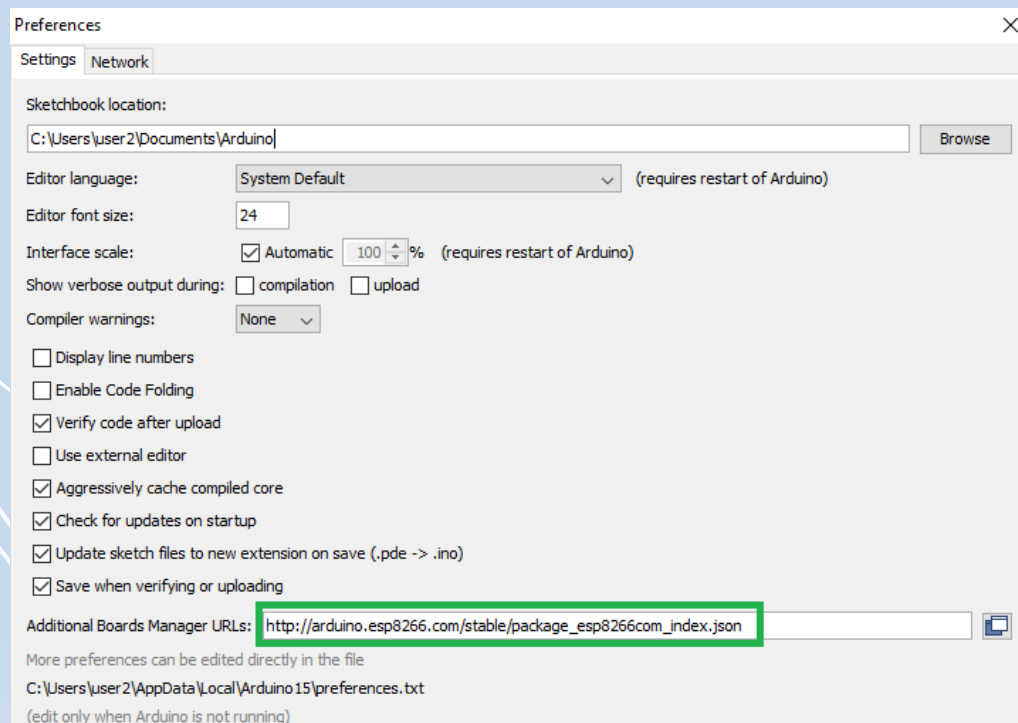
دکتر علی بهلولی



# نحوه برنامه نویسی ماژول وای فای

# نصب کتابخانه های لازم برای ماژولهای ESP8266

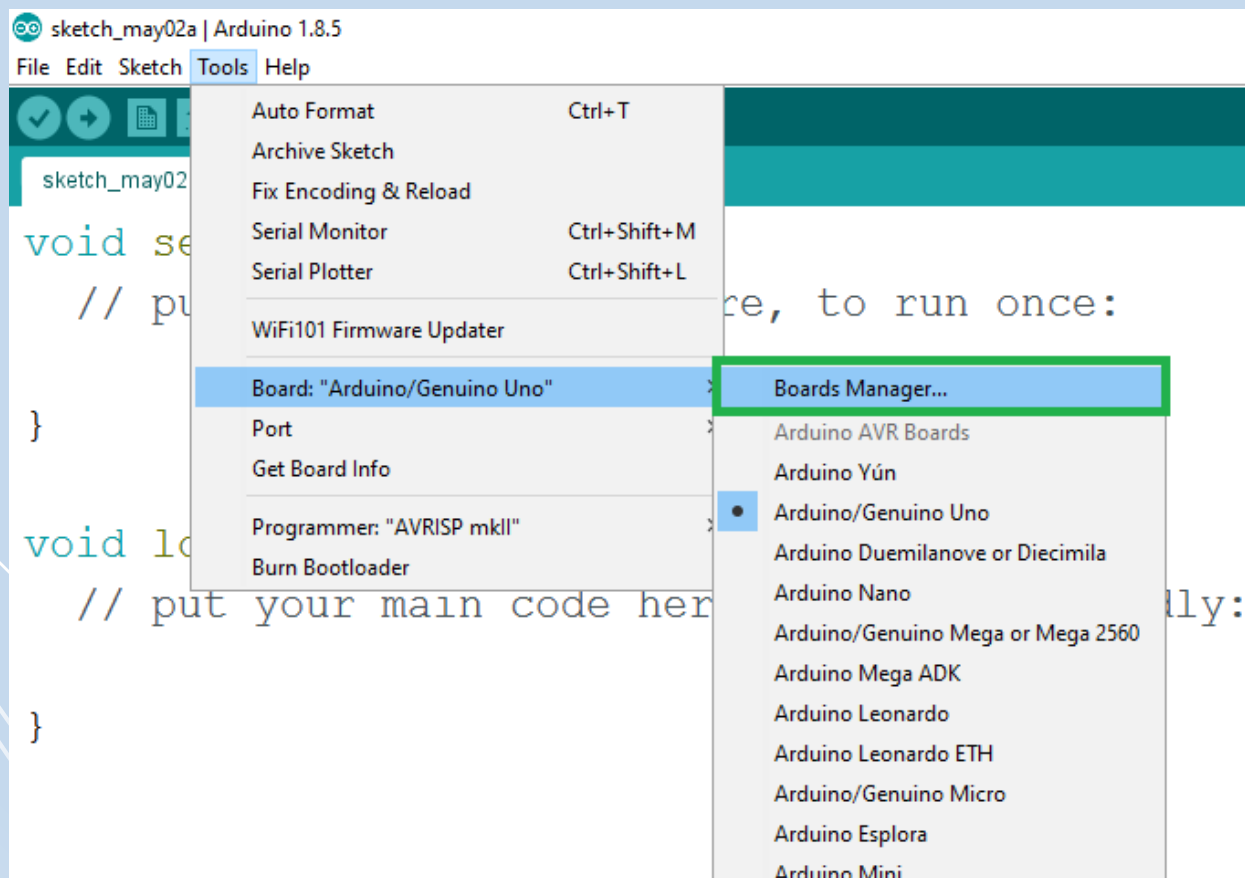
- انتخاب گزینه preferences از منوی فایل
- درج لینک مربوط به ماژول در محل مشخص شده
- [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



# نصب کتابخانه های لازم برای ماژولهای

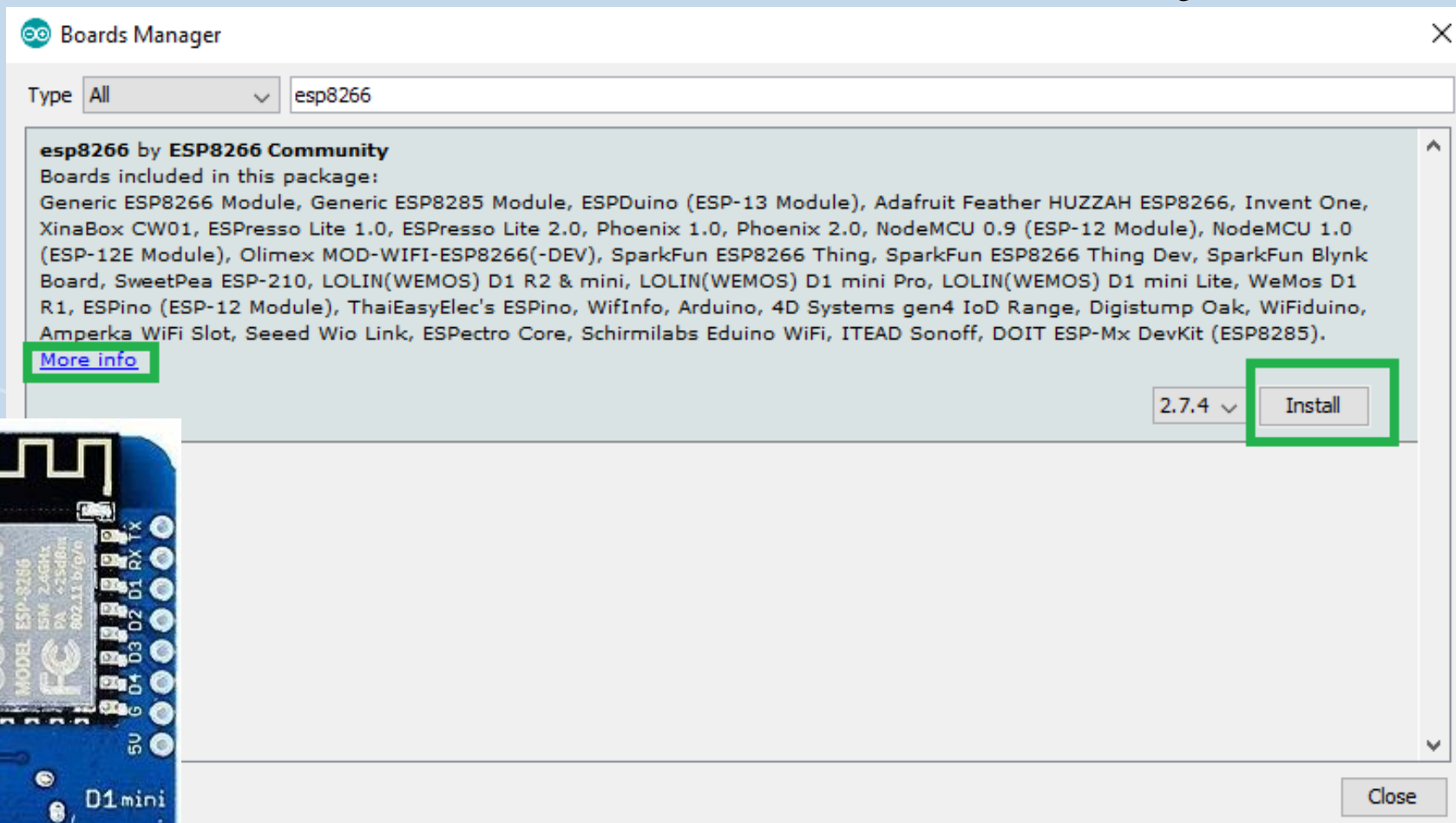
## ESP8266

### • انتخاب گزینه Boards Manager



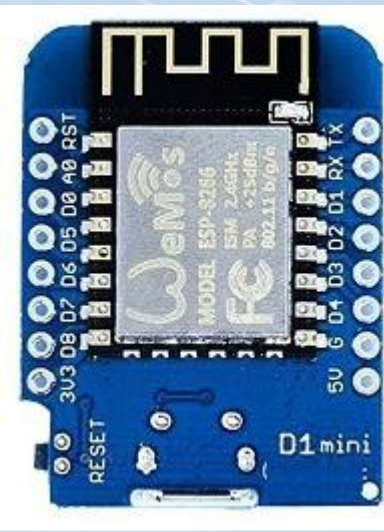
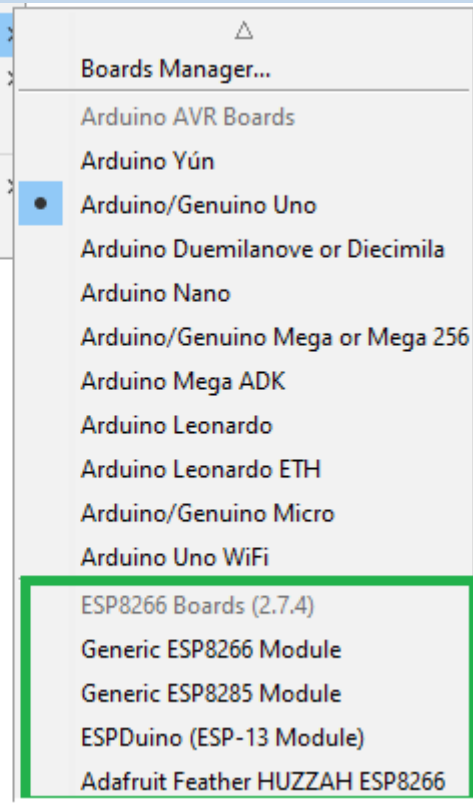
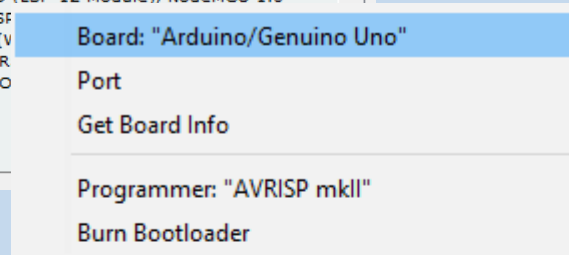
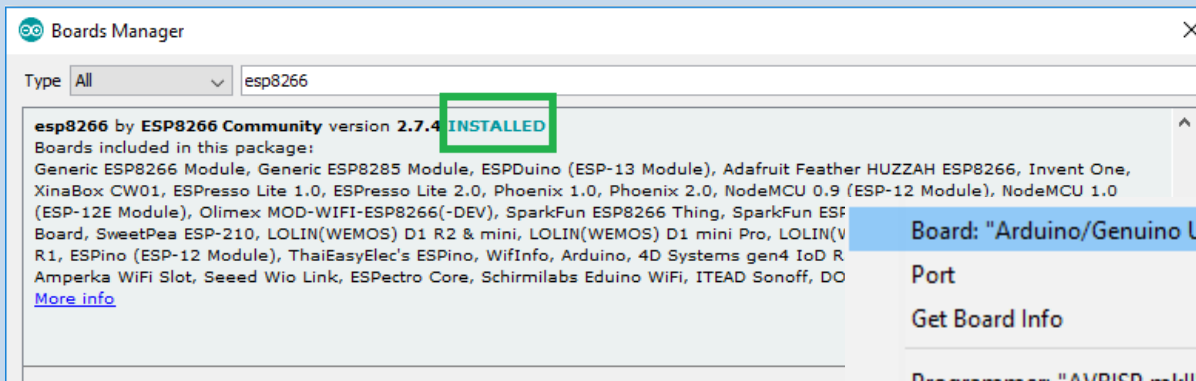
# نصب کتابخانه های لازم برای ماژولهای ESP8266

● جستجوی ماژول esp8266



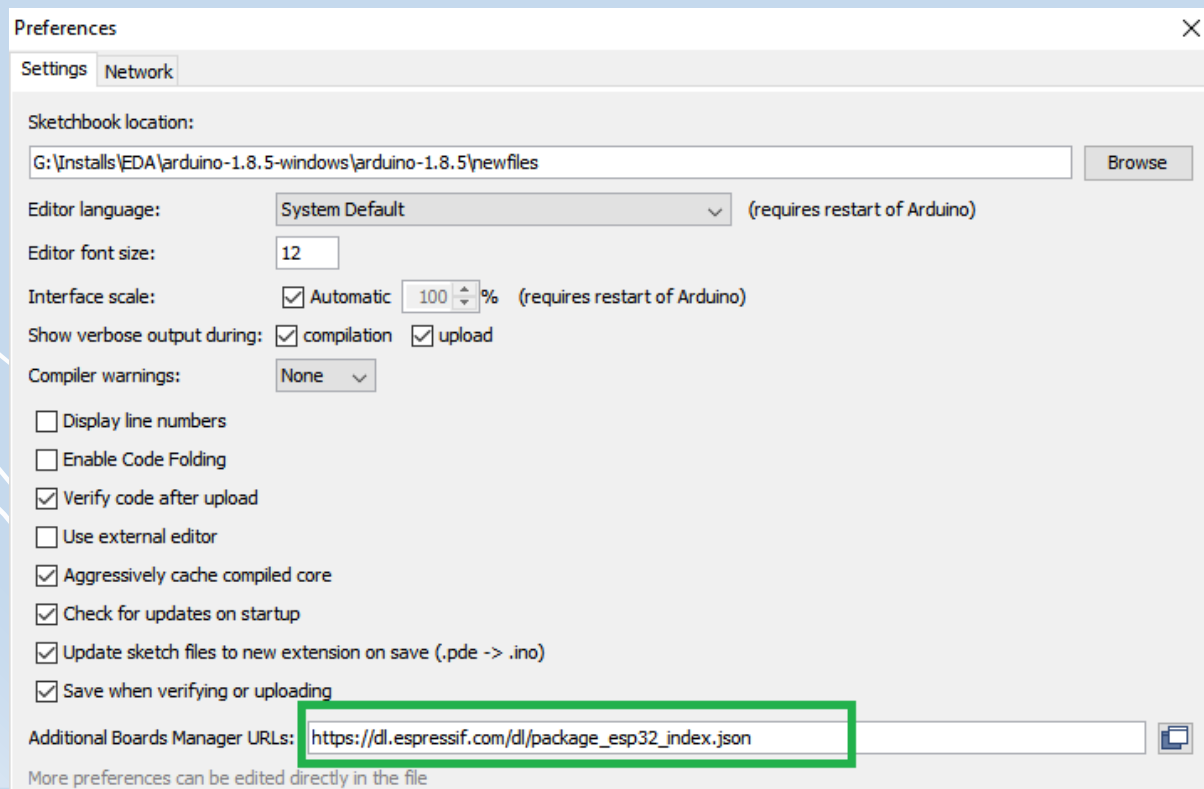
# نصب کتابخانه های لازم برای ماژولهای ESP8266

● چک کردن موفقیت در نصب



# نصب کتابخانه های لازم برای ماژولهای ESP32

- انتخاب گزینه preferences از منوی فایل
- درج لینک مربوط به ماژول در محل مشخص شده
- [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)

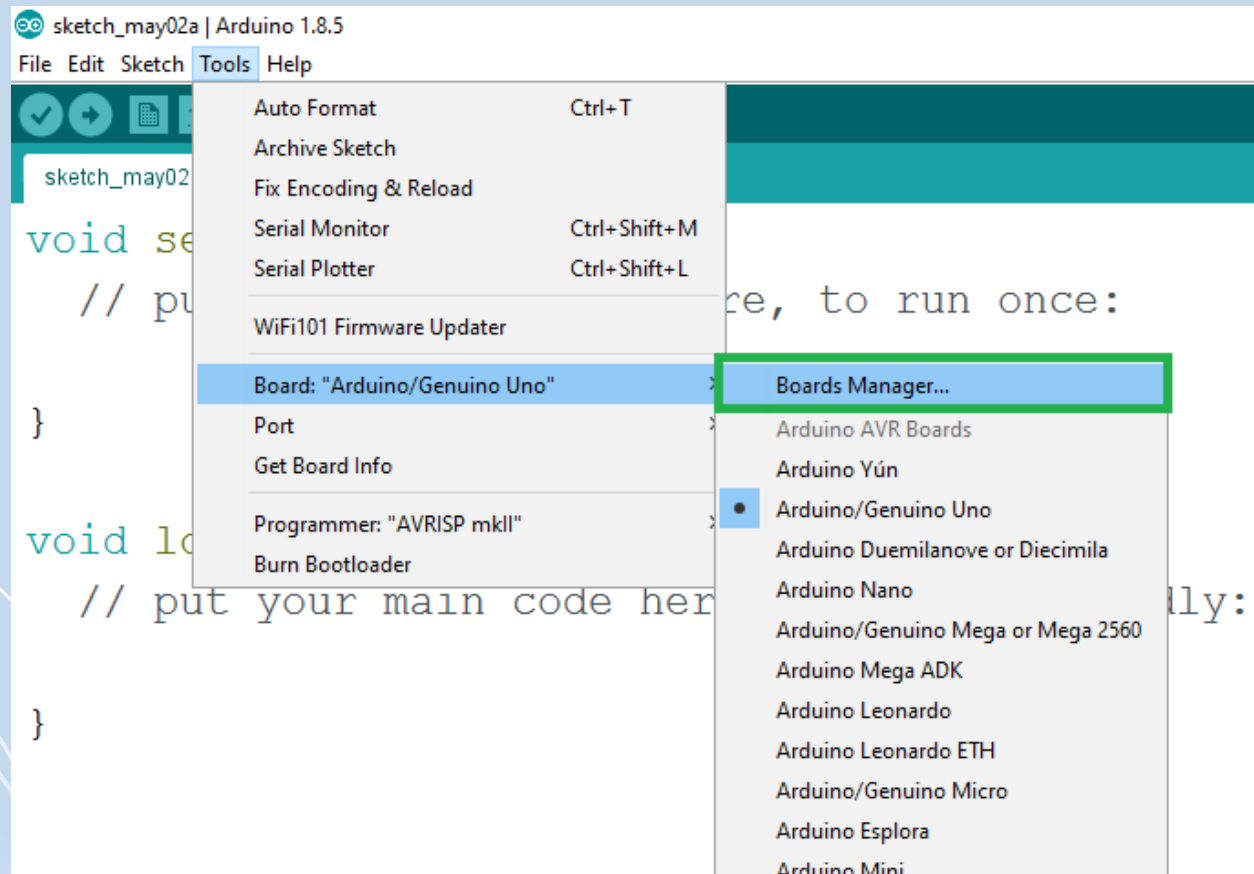




# نصب کتابخانه های لازم برای ماژولهای

## ESP32

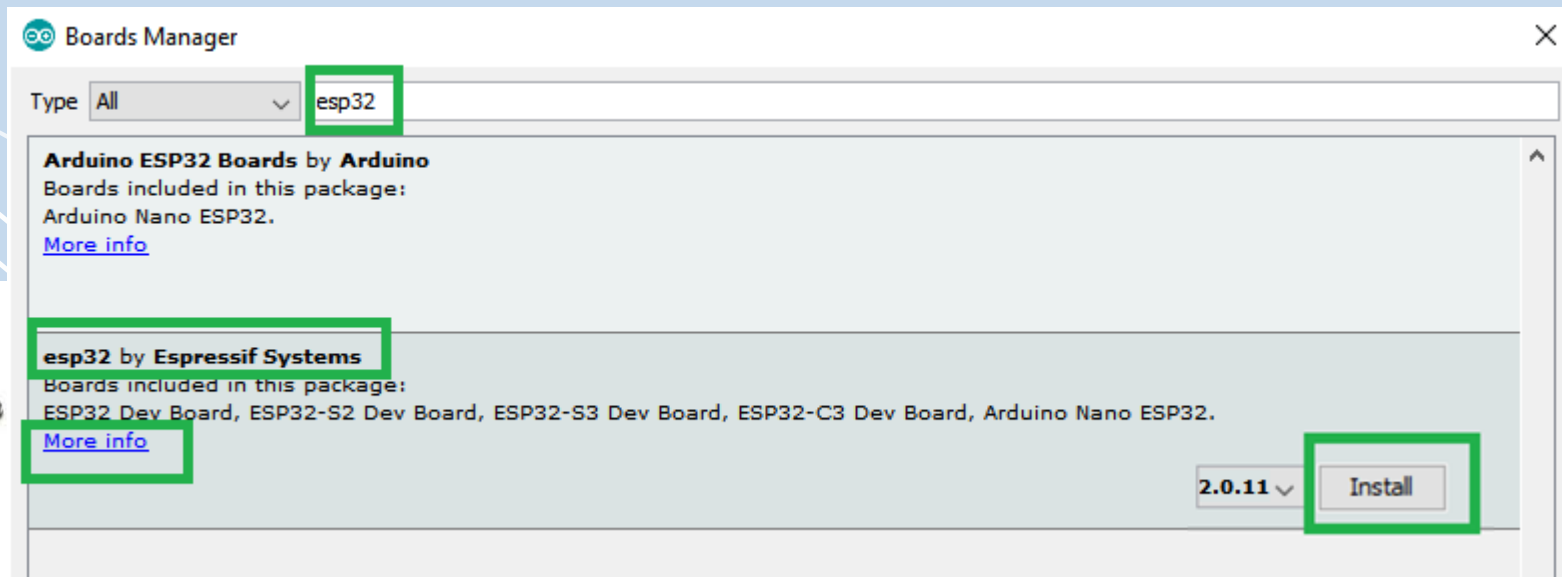
### • انتخاب گزینه Boards Manager





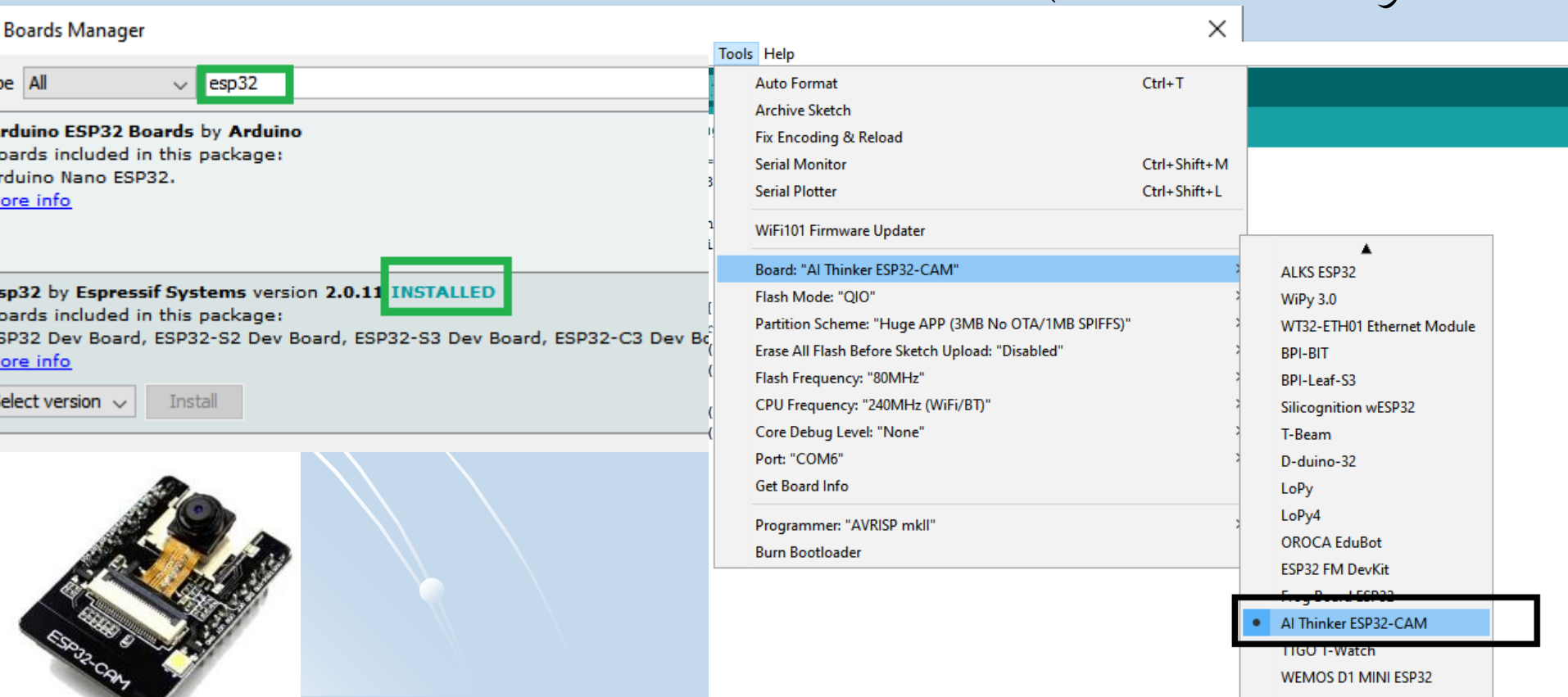
# نصب کتابخانه های لازم برای ماژولهای ESP32

- جستجوی ماژول esp32
- در قسمت Espressif گزینه more Info را بزنید
- گزینه Install را انتخاب تا نصب شود



# نصب کتابخانه های لازم برای ماژولهای ESP32

- چک کردن موفقیت در نصب (نمایش Board Installed در قسمت Manager) و (اضافه شدن بورد AI Thinker ESP32-CAM در قسمت Board)



The screenshot shows the Arduino IDE Boards Manager window. The 'Board' dropdown is set to 'esp32'. The list of boards includes 'AI Thinker ESP32-CAM' which is marked as 'INSTALLED'. A dropdown menu is open, showing the selected board and a list of other boards. The 'AI Thinker ESP32-CAM' board is highlighted in the dropdown menu.

Boards Manager

Board: All

Arduino ESP32 Boards by Arduino

boards included in this package:

Arduino Nano ESP32.

[More info](#)

ESP32 by Espressif Systems version 2.0.11 **INSTALLED**

boards included in this package:

ESP32 Dev Board, ESP32-S2 Dev Board, ESP32-S3 Dev Board, ESP32-C3 Dev Board

[More info](#)

Select version

Install

Tools Help

Auto Format Ctrl+T

Archive Sketch

Fix Encoding & Reload

Serial Monitor Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

WiFi101 Firmware Updater

Board: "AI Thinker ESP32-CAM"

Flash Mode: "QIO"

Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)"

Erase All Flash Before Sketch Upload: "Disabled"

Flash Frequency: "80MHz"

CPU Frequency: "240MHz (WiFi/BT)"

Core Debug Level: "None"

Port: "COM6"

Get Board Info

Programmer: "AVRISP mkII"

Burn Bootloader

ALKS ESP32

WiPy 3.0

WT32-ETH01 Ethernet Module

BPI-BIT

BPI-Leaf-S3

Silicognition wESP32

T-Beam

D-duino-32

LoPy

LoPy4

OROCA EduBot

ESP32 FM DevKit

ESP32 Dev Board

AI Thinker ESP32-CAM

TTGO T-Watch

WEMOS D1 MINI ESP32

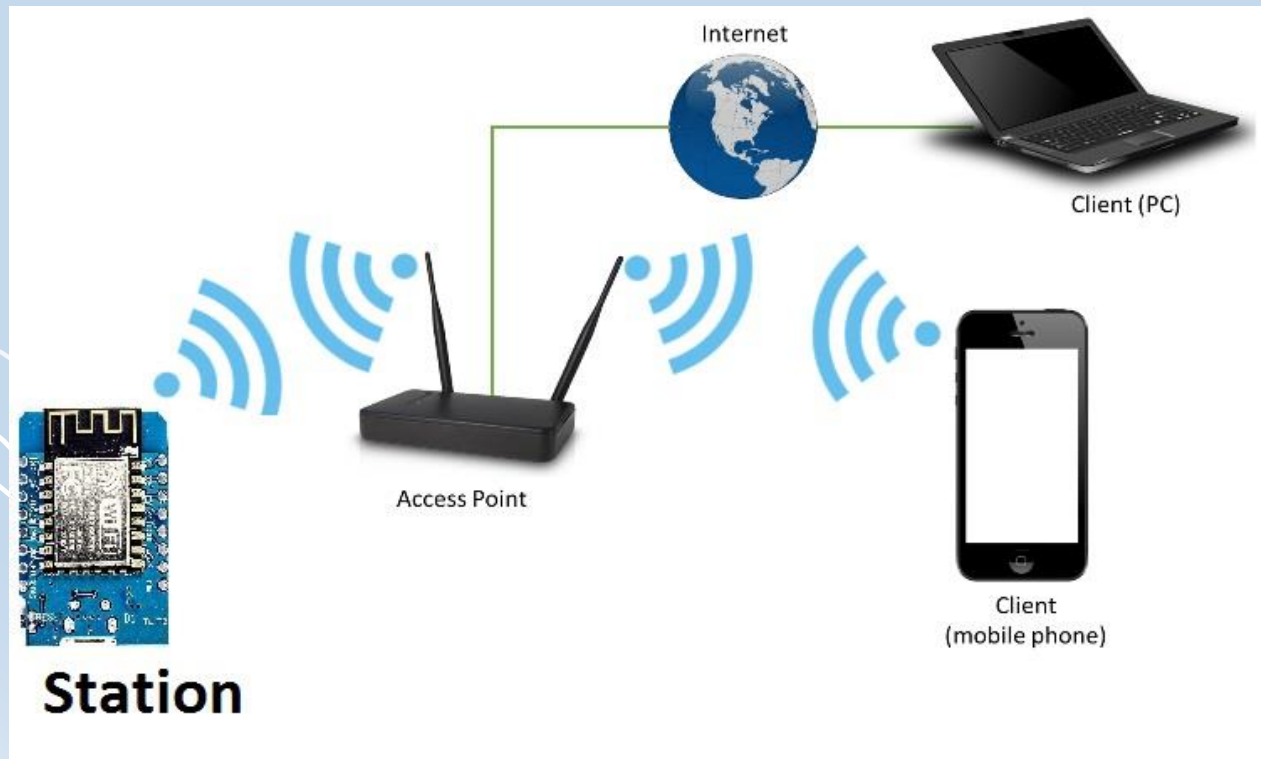
# نحوه برنامه نویسی ماژول وای فای

- کار با ماژول به دو صورت انجام می شود:
- اتصال آن به یک میکروکنترلر و فرمان دادن به آن از طریق UART

- برنامه نویسی مستقیم ماژول (بدون نیاز به میکروکنترلر دیگر)

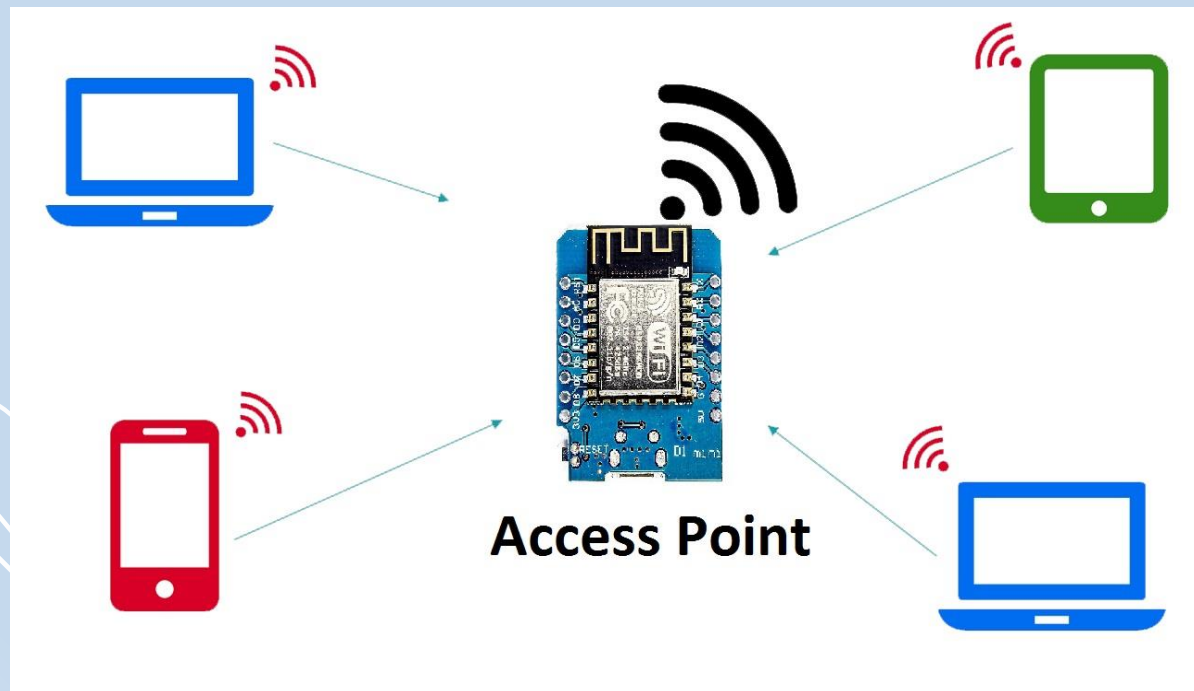
# مدهای کاری مازول

## STAtion •



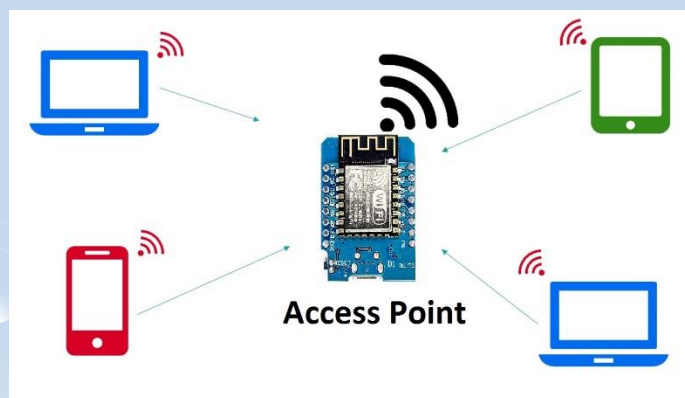
# مدهای کاری مازول

## (AP) Access Point ●



# مدهای کاری مازول

● مد BOTH



# اتصال به ماژول از طریق UART

- دستورات AT
- فرامین به صورت رشته هایی توسط پورت سریال برای دستگاه ارسال خواهد شد
- انتهای هر دستور دو کاراکتر `\r` و `\n` باید باشد
- سرعت پیشفرض 115200 است
- مثلاً فرمان زیر برای تست اتصال مورد استفاده قرار می گیرد:
- `AT\r\n`
- در پاسخ، ماژول پیام `OK` را برمی گرداند



# دسته بندی فرامین AT

- فرامین را می توان در سه دسته زیر تقسیم کرد:
- Basic AT Commands
- WiFi AT Commands
- TCP/IP AT Commands

# اتصال به ماژول از طریق UART

## ● دستورات پایه (Basic AT Commands)

### دستور AT

عمل کرد	پاسخ برنامه	دستور	نوع دستور
ماژول را تست می کند	OK	AT	Execute

### دستور AT+RST

عمل کرد	پاسخ برنامه	دستور	نوع دستور
تمام تنظیمات ماژول را ریست می کند	OK	AT+RST	Execute

### دستور AT+GMR

عمل کرد	پاسخ برنامه	دستور	نوع دستور
ورژن ماژول را نمایش می دهد	version, OK	AT+GMR	Execute

# اتصال به مازول از طریق UART

## ● دستورات مربوط به وای فای (WiFi AT Commands)

این دستورات برای کنترل ویژگیهای وای فای مورد استفاده قرار می گیرد (از قبیل: تعیین مود عملکرد، دریافت لیست APها، اتصال به اکسس پوینت، تعیین آدرس MAC و...)

● جزئیات دستورات در لینک زیر قابل

مشاهده است:

<https://www.electronicshub.org/esp8266-at-commands/>

### WiFi AT Commands

AT+CWMODE

AT+CWJAP

AT+CWLAP

AT+CWQAP

AT+CIPSTA

AT+CWSAP

AT+CWLIF

# اتصال به مازول از طریق UART

## ● انواع دستورات AT

نوع	مثال	توضیح
Test	?=AT+CIPSTART	مقدار مقادیر مجازی که پارامترهای هر دستور می‌تواند دریافت کند
Query	?AT+CMD	مقدار پارامتر هر دستور را مشخص می‌کند
Set	AT+CMD=Parameter	به پارامترهای هر دستور ، مقداری نسبت داده می‌شود
	AT+CMD	دستور را بدون پارامترهایش اجرا می‌کند

# اتصال به ماژول از طریق UART

## ● برخی از دستورات

### دستور AT+GSLP

عملکرد	پاسخ برنامه	دستور	نوع دستور
مقدار زمانی که ماژول در حالت sleep قرار می‌گیرد	time ,OK	AT+GSLP=time	set

### دستور AT+CWMODE – WIFI mode

عملکرد	پاسخ برنامه	دستور	نوع دستور
مقدارهای مجاز برای پارامتری این دستور را نمایش می‌دهد	CWMODE:+ (۱-۳)OK	?=AT+CWMODE	Test
مدی که ماژول در آن قرار دارد را برگرداند. (مد کلاینت ، هاست یا هردو)	CWMODE:mode+  OK	?AT+CWMODE	Query
مد ماژول را تعیین می‌کند	OK	AT+CWMODE=mode	Execute

# اتصال به مازول از طریق UART

● برخی از دستورات

## دستور AT+CWJAP – Connect to AP

عمل کرد	پاسخ برنامه	دستور	نوع دستور
نام اکسس پوین (مودم) که مازول به آن متصل است را بر میگرداند	+ CWJAP:ssid OK	?AT+CWJAP	Query
به وسیله این دستور به مودم وای فای یا اکسس پوینتی که رمزش را داریم متصل می شویم	OK	AT+CWJAP=ssid,pwd	Execute

# اتصال به مازول از طریق UART

- دستورات مربوط به ارتباطات

(TCP/IP AT Commands)

این دستورات برای ایجاد ارتباطات شبکه و ارسال و دریافت داده ها استفاده می شود.

- جزئیات دستورات در لینک زیر قابل

مشاهده است:

<https://www.electronicshub.org/esp8266-at-commands/>

## TCP/IP Commands

AT+CIPSTATUS

AT+CIPSTART

AT+CIFSR

AT+CIPMUX

AT+CIPSERVER

AT+CIUPDATE



# پراگرام کردن خود تراشه در آردوینو

- مرحله اول: نصب کتابخانه 8266 در نرم افزار آردوینو

- مرحله دوم: انتخاب بورد وای فای از منوی **Tools** بخش **Board**

- استفاده از دستورات زیر:

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

```
const char* ssid = "XXX";
```

```
const char* password = "YYYYYYYYYY";
```

# پراگرام کردن خود تراشه در آردوینو

● استفاده از دستورات:

```
WiFi.begin(ssid, password);  
WiFiServer server(80);  
server.begin();  
WiFiClient client =server.available();  
client.readStringUntil
```

## ledOnOff §

```
#include <ESP8266WiFi.h>
const char* ssid = "WifiTest";
const char* password = "12345678";
int ledPin = D4;
WiFiServer server(80);
void setup() {
    Serial.begin(115200);
    delay(10);
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, LOW);
    // Connect to WiFi network
    Serial.println();
    Serial.println();
    Serial.print("Connecting to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED)
    {
        delay(1000);
        Serial.print(".");
        digitalWrite(ledPin, !digitalRead(ledPin));
    }
```

```
Serial.println("");  
Serial.println("WiFi connected");  
// Start the server  
server.begin();  
Serial.println("Server started");  
// Print the IP address  
Serial.print("Use this URL : ");  
Serial.print("http://");  
Serial.print(WiFi.localIP());  
Serial.println("/");  
}
```

```

void loop() {
// Check if a client has connected
WiFiClient client = server.available();
if (!client) {
return;
}
// Wait until the client sends some data
Serial.println("new client");
while(!client.available()){
delay(1);
}
// Read the first line of the request
String request = client.readStringUntil('\r');
Serial.println(request);
client.flush();
// Match the request
int value = LOW;
if (request.indexOf("/LED=ON") != -1) {
digitalWrite(ledPin, LOW);
value = LOW; }
if (request.indexOf("/LED=OFF") != -1){
digitalWrite(ledPin, HIGH);
value = HIGH;
}
// Return the response
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // do not forget this one
client.println("<!DOCTYPE HTML>");

```

```
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.print("Led pin is now: ");
if(value == HIGH) {
client.print("On");
digitalWrite(ledPin, LOW);
} else {
client.print("Off");
digitalWrite(ledPin, HIGH);
}
client.println("<br><br>");
client.println("Click <a href=\""/LED=ON\">here</a> turn the LED on pin 5 ON<br>");
client.println("Click <a href=\""/LED=OFF\">here</a> turn the LED on pin 5 OFF<br>");
client.println("</html>");
delay(1);
Serial.println("Client disconnected");
Serial.println("");
}
```