



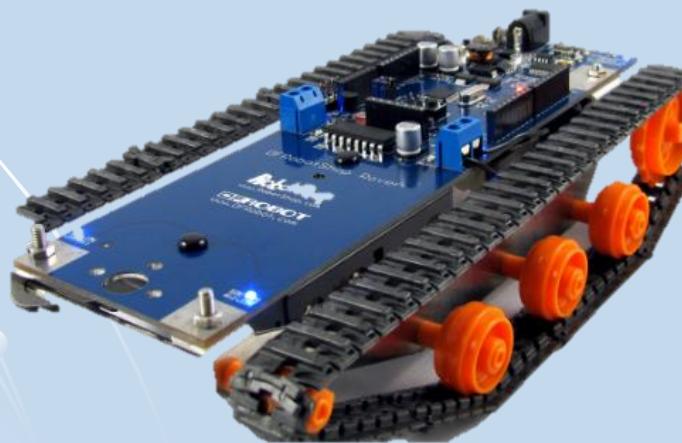
دانشکده مهندسی کامپیوتر

درس برنامه نویسی تجهیزات اینترنت اشیا

نمیسال دوم ۱۴۰۲

آردوینو (ارتباط سریال)

دکتر علی بهلوانی



مقدمه

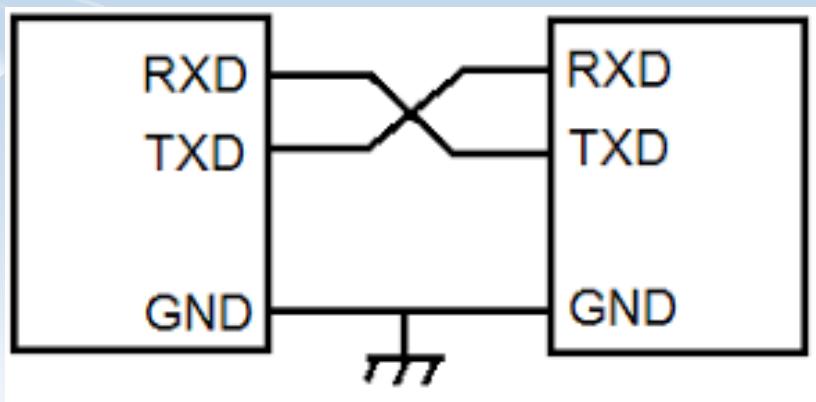
- چگونه می توان ورودی/خروجیهای بورد آردوینو را گسترش داد؟(فقط کلید و led نباشد)
- چگونه می توان بین دو سیستم دیجیتال دیتا رد و بدل کرد؟
- یک بورد آردوینو با بورد دیگر
- بورد آردوینو و کامپیوتر
- بورد آردوینو و ماژولهایی نظیر وای فای، کارت حافظه و

پروتکلهای استاندارد ارتباطی

- پروتکل UART
- پروتکل SPI
- پروتکل I²C
-

پروتکل UART

- سه رشته سیم مورد نیاز است
- ارتباط به صورت کاملاً دو طرفه برقرار خواهد شد
- مهمترین پارامتر: سرعت ارسال و دریافت



پروتکل UART

- دستورات برنامه نویسی پورت سریال
- تعریف پورت سریال

`Serial.begin(int Bitrate);`

- نوشتن در پورت سریال
- `Serial.print("Hello World");`

- خواندن از پورت سریال

`Serial.read();`

`Serial.available();`

تعريف پورت سریال

- تعیین سرعت ارسال/دریافت
- اطمینان از آماده شدن سخت افزار/نرم افزار

```
void setup() {  
    //Initialize serial and wait for port to open:  
    Serial.begin(9600);  
    while (!Serial);  
}
```

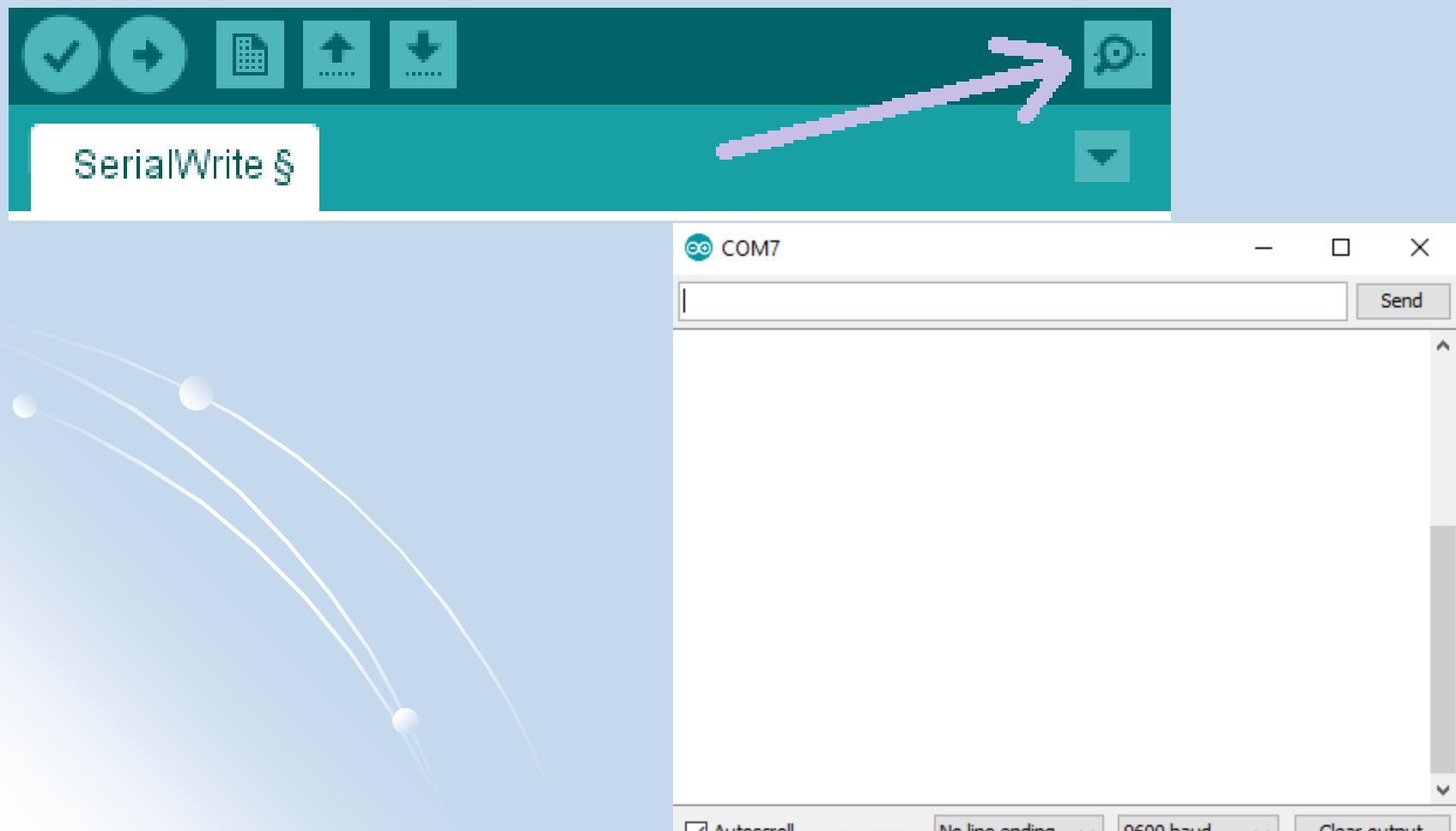
دستورات مربوط به نوشتن در پورت

```
Serial.print(x);  
Serial.print(x, DEC);  
Serial.print(x, HEX);  
Serial.print(x, OCT);  
Serial.println(x, BIN);
```

● سوال: خروجی کجا نشان داده خواهد شد؟

خروجی کجا نمایش داده خواهد شد؟

- باز کردن پنجره ترمینال



دستورات مربوط به خواندن از پورت

مثال:

```
int incomingByte = 0; // for incoming serial data

void setup() {
  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
  while (!Serial);

}

void loop() {
  // send data only when you receive data:
  if (Serial.available() > 0) {
    // read the incoming byte:
    incomingByte = Serial.read();

    // say what you got:
    Serial.print("I received: ");
    Serial.println(incomingByte, DEC);
  }
}
```

نحوه فریم بندی داده ها برای ارسال

- چون در **UART** ارتباط سنکرون نیست، بنابراین طول بخش دیتا نمیتواند بیش از ۸ بیت باشد.
- پیکربندی پورت **UART** معمولاً به صورت زیر کد می شود:
- Examples
 - “9600-N-8-1” → <baudrate><parity><databits><stopbits>
 - “9600-8-N-1” → <baudrate><databits><parity><stopbits>

پورت سریال در کامپیووتر

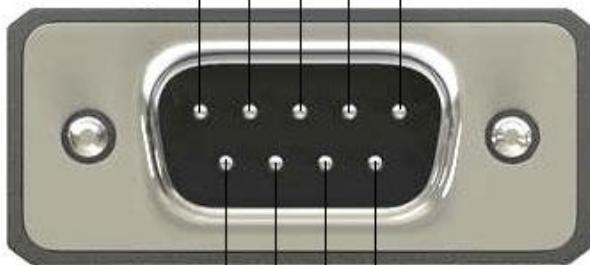
- پورت سریال استاندارد روی کامپیوترهای قدیمی

- پورت RS232

- استفاده از ولتاژهای ۱۲+ و ۱۲- برای ارسال و دریافت بیتها

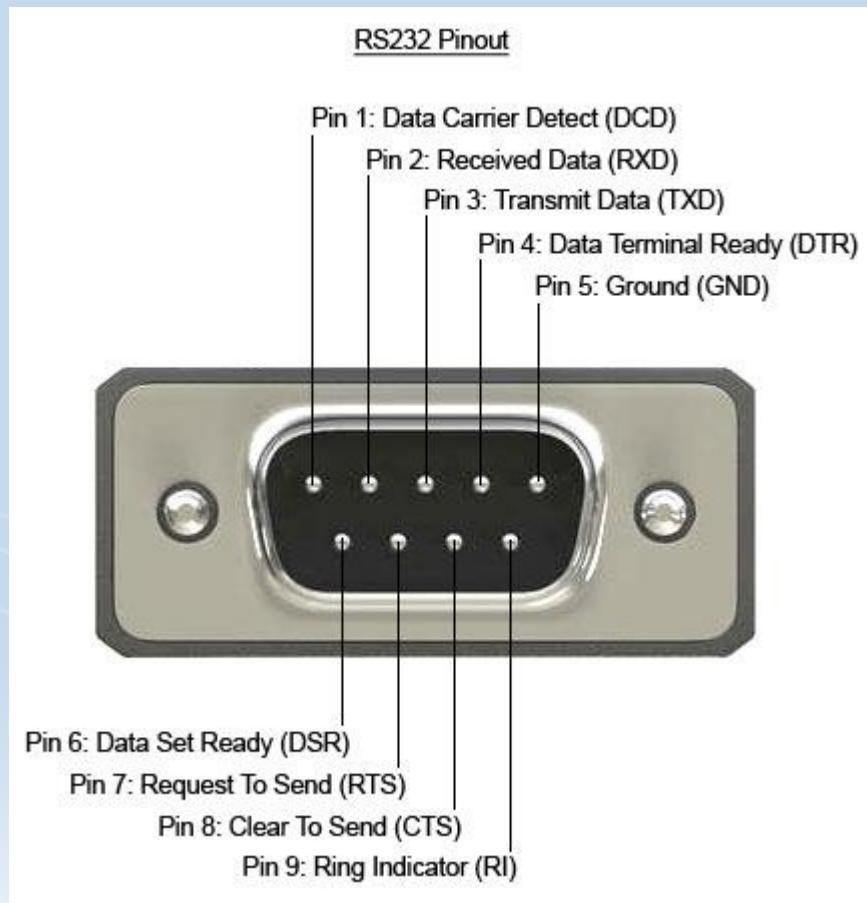
RS232 Pinout

Pin 1: Data Carrier Detect (DCD)
Pin 2: Received Data (RXD)
Pin 3: Transmit Data (TXD)
Pin 4: Data Terminal Ready (DTR)
Pin 5: Ground (GND)



Pin 6: Data Set Ready (DSR)
Pin 7: Request To Send (RTS)
Pin 8: Clear To Send (CTS)
Pin 9: Ring Indicator (RI)

پورت سریال در کامپیووتر



- پایه های مهم مورد استفاده:
 - پایه شماره ۲ : دریافت دیتا
 - پایه شماره ۳: ارسال دیتا
 - پایه شماره ۵: زمین مشترک

پورت سریال در کامپیووتر

- پورت سریال با استفاده از مبدل USB



مشابه پورت سریال استاندارد (RS232)

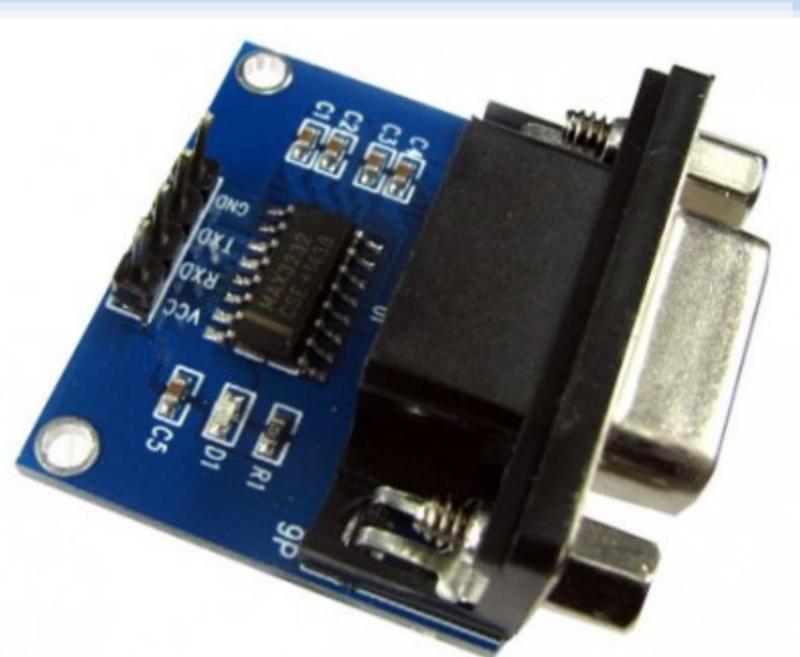


■	+5V
■	GND
■	RXD
■	TXD

سطوح ولتاژ ۰ و ۵ ولت برای ارسال
و دریافت استفاده می شود (TTL)

ماژولهای پورت سریال

- ماژول مبدل RS232 به TTL
- قیمت: ۳۶۰۰۰ تومان (۱۴۰)



(digikala)

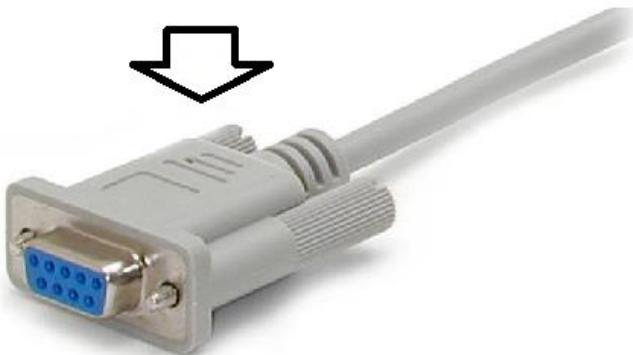
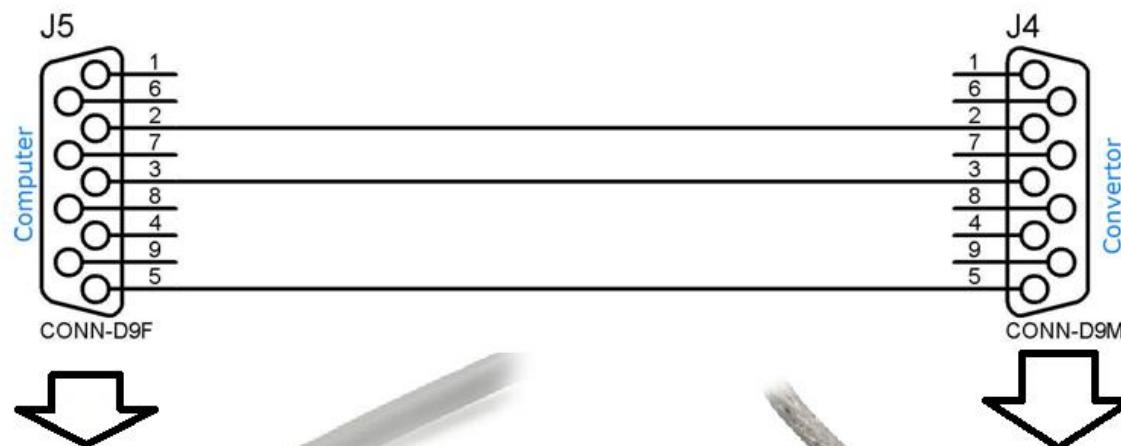
کابلها و کانکتورهای مرتبط با پورت سریال



کابل رابط، اتصال مستقیم



• به صورت Male-Female

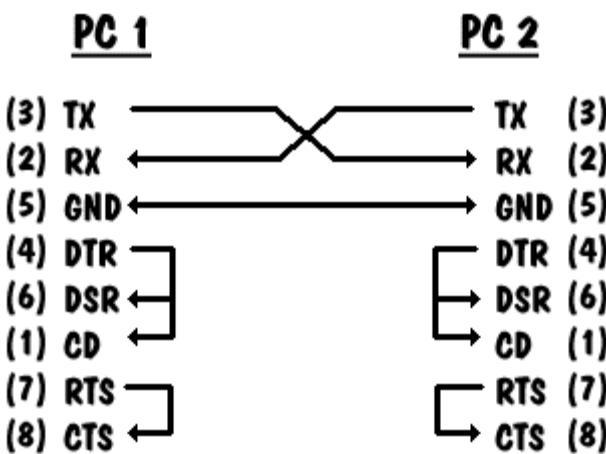


کابل رابط، اتصال کراس

- به دو صورت Female-Female و Male-Male وجود دارد



NULL MODEM



برنامه نویسی UART در C#

- کامپونت Serial Port را روی فرم قرار دهید و ..
- ```
foreach (string item in System.IO.Ports.SerialPort.GetPortNames())
 comboBoxCOMPort.Items.Add(item);
```

```
serialPort.PortName = "COM1";
```

```
serialPort.BaudRate = 9600;
```

باز کردن پورت برای شروع به عملیات خواندن یا نوشتن;

خواندن یک بایت از بافر پورت، منتظر می ماند تا وقتی که دیتا بیاید;

خواندن یک بایت از بافر پورت، اگر دیتا بیاید نباشد منتظر نمیماند;

ارسال یک کاراکتر به پورت;

<http://csharp.simpleserial.com/>

مرجع

راهنمای کامل برنامه نویسی پورت سریال در سی شارپ

آموزش-سریال-پورت-در-سی-شارپ/<https://asainterface.com/>

# برخی توابع مفید برای کار با پورت سریال

- <https://arduinogetstarted.com/reference/arduino-serial>

## Functions

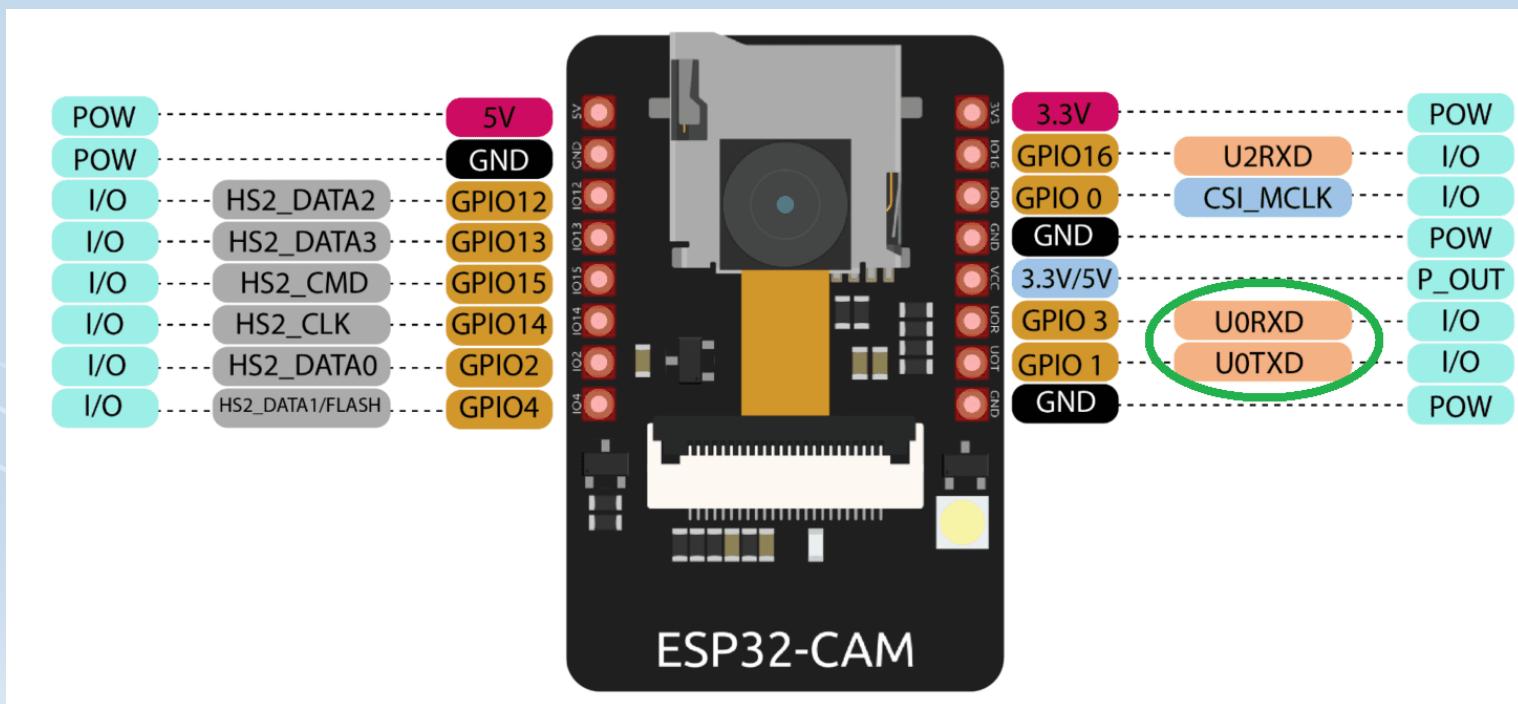
- ◆ Serial.if(Serial)
- ◆ Serial.available()
- ◆ Serial.availableForWrite()
- ◆ Serial.begin()
- ◆ Serial.end()
- ◆ Serial.find()
- ◆ Serial.findUntil()
- ◆ Serial.flush()
- ◆ Serial.parseFloat()
- ◆ Serial.parseInt()
- ◆ Serial.peek()
- ◆ Serial.print()
- ◆ Serial.println()
- ◆ Serial.read()
- ◆ Serial.readBytes()

- ◆ Serial.readBytesUntil()
- ◆ Serial.readString()
- ◆ Serial.readStringUntil()
- ◆ Serial.setTimeout()
- ◆ Serial.write()
- ◆ Serial.serialEvent()

# انواع پورت سریال‌های آردوینو

- پورت سریال سخت افزاری
- یک پورت سریال واقعی (پایه صفر و یک)
- پورت سریال نرم افزاری

# پورت سریال سخت افزاری



# پورت سریال نرم افزاری

- امکان استفاده از سایر پایه های آردوینو برای پورت سریال
- استفاده از کتابخانه SoftwareSerial.h
- File->Examples->SoftwareSerial

# مثال از پورت سریال نرم افزاری

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial(10, 11); // RX, TX

void setup() {
 // Open serial communications and wait for port to open:
 Serial.begin(57600);
 while (!Serial) {
 ; // wait for serial port to connect. Needed for native USB port only
 }

 Serial.println("Goodnight moon!");

 // set the data rate for the SoftwareSerial port
 mySerial.begin(4800);
 mySerial.println("Hello, world?");
}

void loop() { // run over and over
 if (mySerial.available()) {
 Serial.write(mySerial.read());
 }
 if (Serial.available()) {
 mySerial.write(Serial.read());
 }
}
```

## مثال از پورت سریال نرم افزاری

- پورت سریال نرم افزاری محدودیتها باید ایجاد می کند که ممکن است روی عملکرد برنامه تاثیر بگذارد، بنابراین استفاده از آن در حالت کلی توصیه نمیشود
- برای ایجاد پورت سریال نرم افزاری، از همه پایه های نمیتوان استفاده کرد و در مستندات باید محدودیتها را مطالعه کرد



دانشکده مهندسی کامپیوتر

درس برنامه نویسی تجهیزات اینترنت اشیا

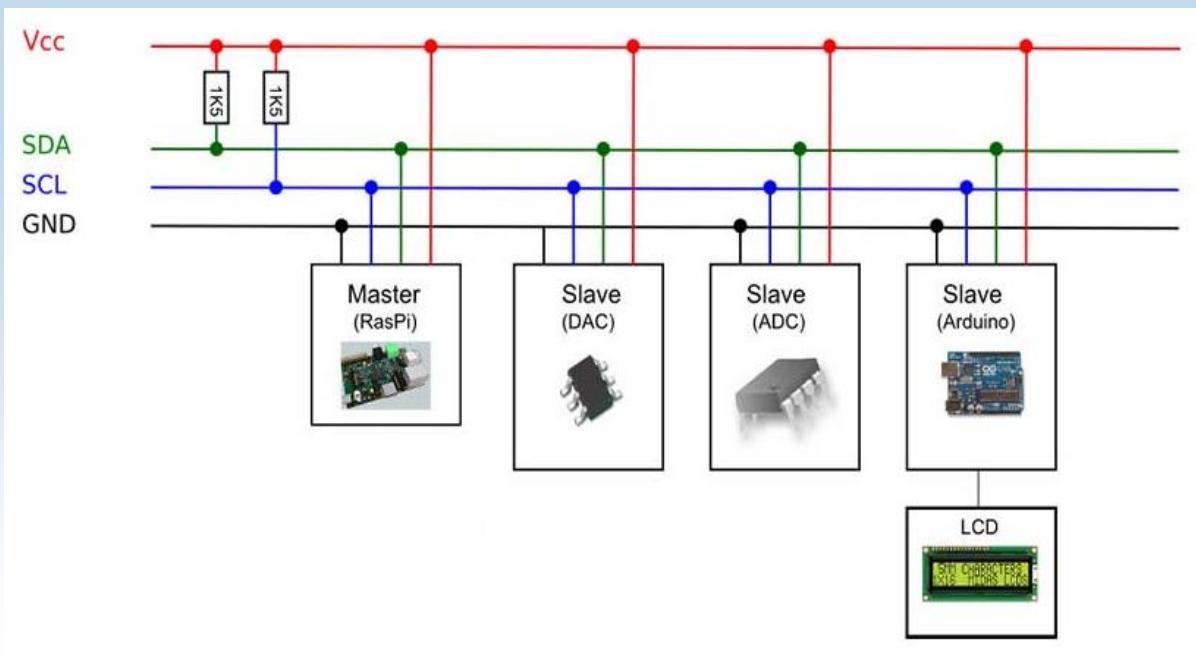
نمیسال دوم ۱۴۰۲

پروتکل‌های I<sup>2</sup>C و SPI

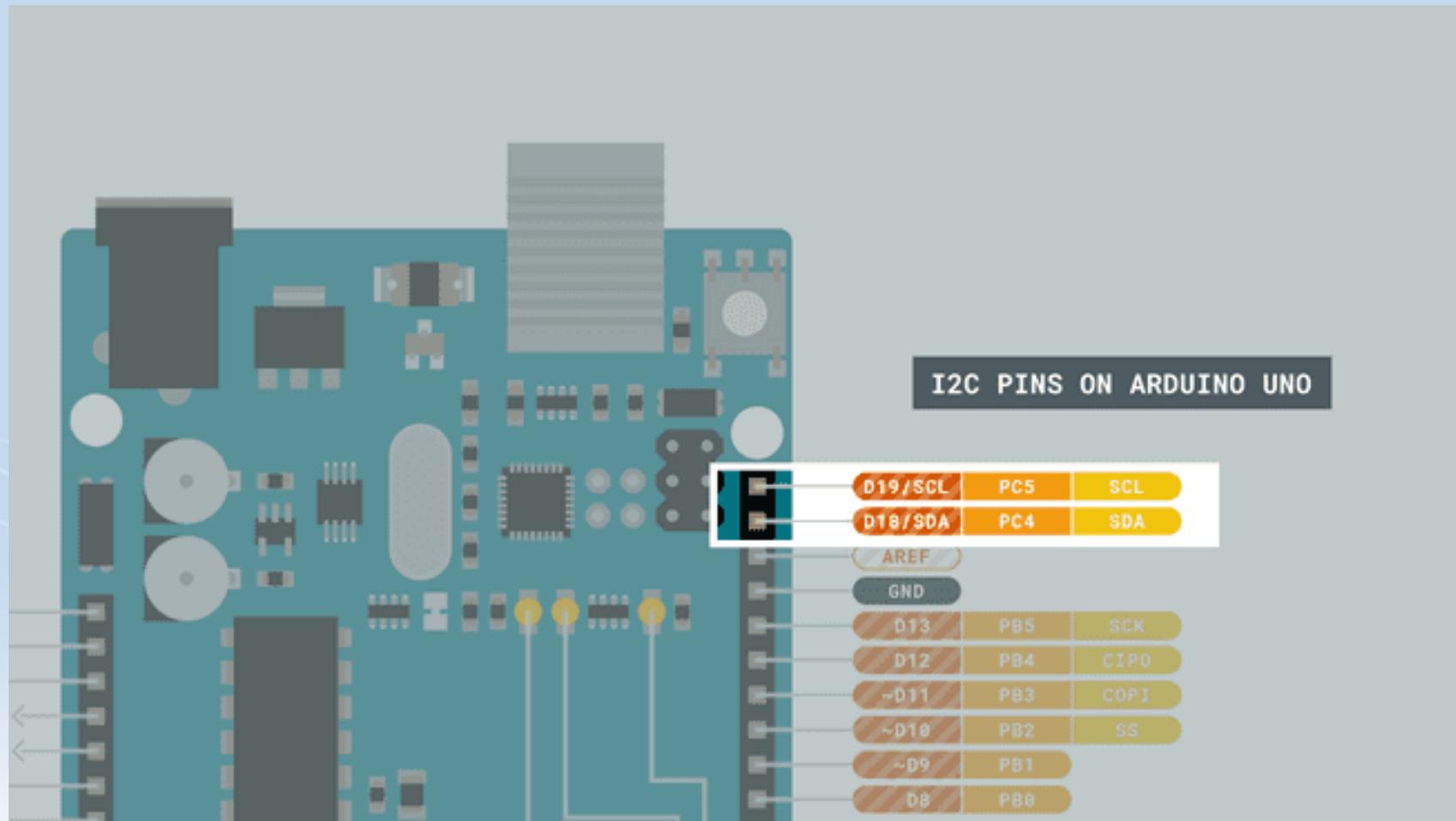
دکتر علی بهلوانی

# I<sup>2</sup>C پروتکل

- سه رشته سیم مورد نیاز است
- ارتباط به صورت نیمه دو طرفه برقرار خواهد شد.
- ارتباط سنکرون



# I<sup>2</sup>C پروتکل



# I<sup>2</sup>C پروتکل

## ● نمونه کد Master

```
#include <Wire.h>

void setup() {
 Wire.begin(); // join i2c bus (address optional for master)
 Serial.begin(9600); // start serial for output
}

void loop() {
 Wire.requestFrom(8, 6); // request 6 bytes from peripheral device #8

 while (Wire.available()) { // peripheral may send less than requested
 char c = Wire.read(); // receive a byte as character
 Serial.print(c); // print the character
 }

 delay(500);
}
```

# I<sup>2</sup>C پروتکل

## Slave کد نمونه ●

```
#include <Wire.h>

void setup() {
 Wire.begin(8); // join i2c bus with address #8
 Wire.onRequest(requestEvent); // register event
}

void loop() {
 delay(100);
}

// function that executes whenever data is requested by master
// this function is registered as an event, see setup()
void requestEvent() {
 Wire.write("hello "); // respond with message of 6 bytes
 // as expected by master
}
```

# I<sup>2</sup>C ماظول نمونه

• ماظول رزوليشن 128x64 سفید OLED 1.3 inch I<sup>2</sup>C

display.clearDisplay() – all pixels are off

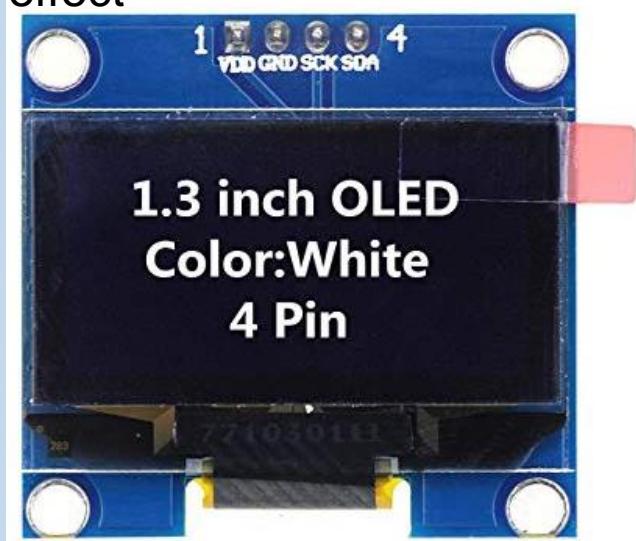
display.drawPixel(x,y, color) – plot a pixel in the x,y coordinates

display.setTextSize(n) – set the font size, supports sizes from 1 to 8

display.setCursor(x,y) – set the coordinates to start writing text

display.print("message") – print the characters at location x,y

display.display() – call this method for the changes to make effect



# پروتکل SPI

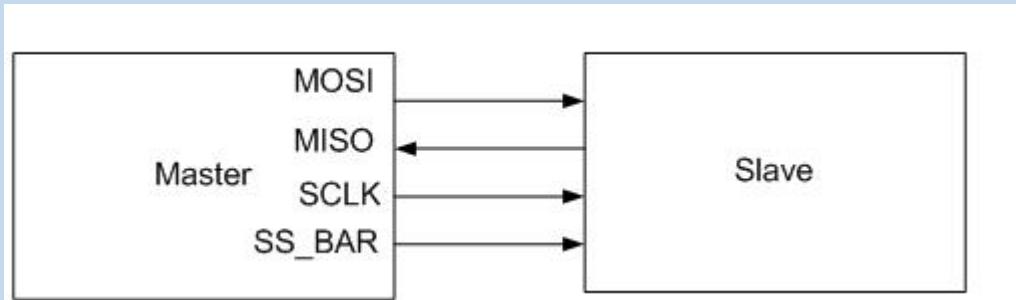
# کلیات پروتکل SPI

- پروتکل سریال از نوع سنکرون
- ساده، سریع و سادگی در استفاده
- به علت سادگی هر دستگاهی میتواند این واسط ارتباطی را داشته باشد



# قابلیتهاي پروتکل SPI

- ارتباط به صورت کاملاً دو طرفه می باشد(FullDuplex)
- ارتباط دو طرفه در هر لحظه می تواند رخ دهد
- سرعت ارسال داده ها در حدود چندین مگابیت است(۱ تا ۲۰ مگابیت بر ثانیه)
- داده ارسالی در هر مرحله، بین ۴ تا ۱۶ بیت است.

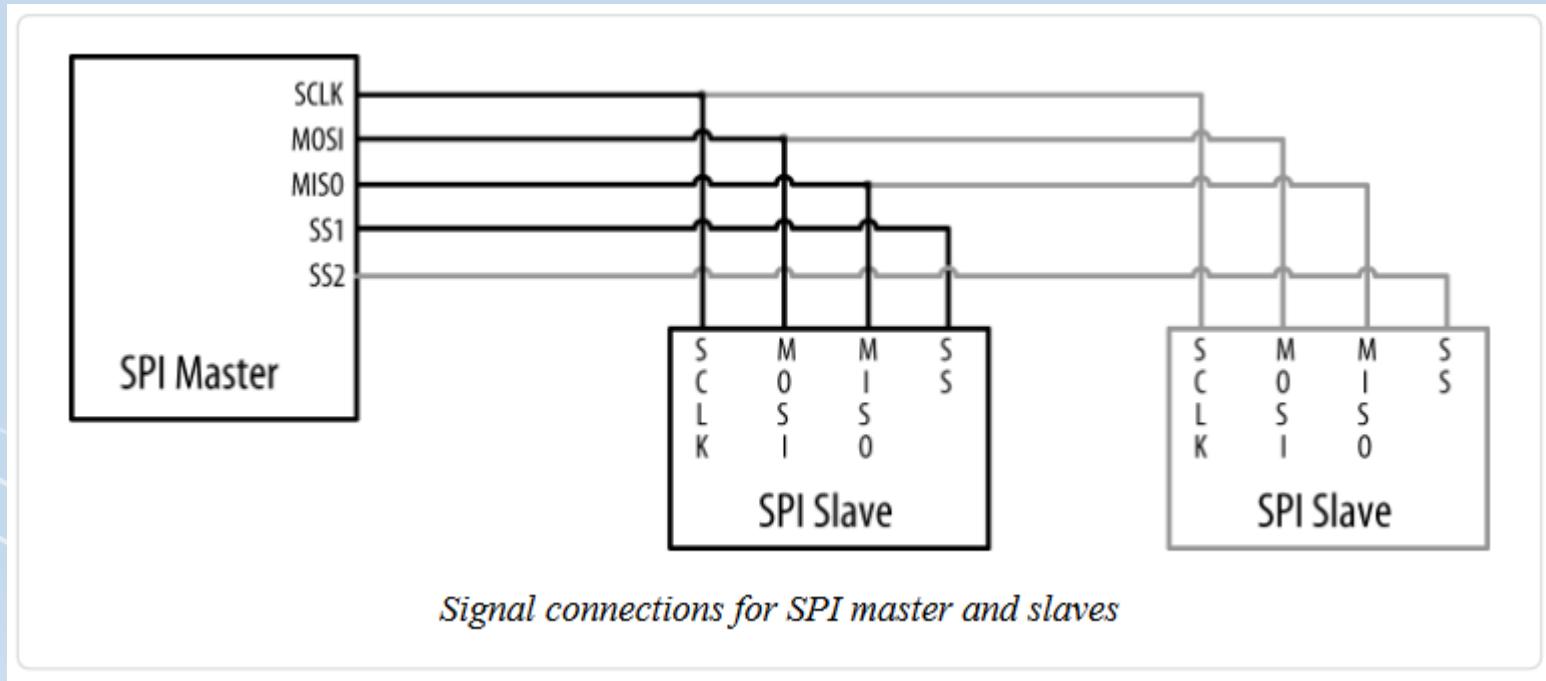


رشته سیم‌های مورد استفاده:

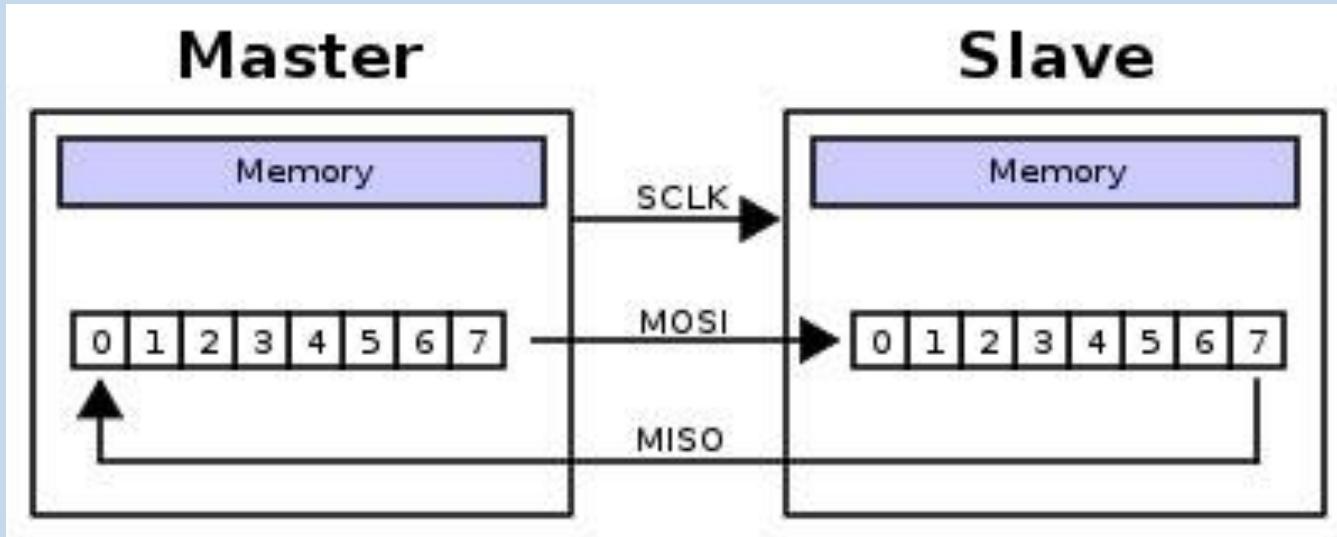
- Master Out Slave In (MOSI)
- Master In Slave Out (MISO)
- System Clock (SCLK)
- Slave Select 1...N

- 1)Master Set Slave Select low
- 2)Master Generates Clock
- 3)Shift registers shift in and out data

# نحوه اتصال چند دستگاه



# نحوه تبادل دیتا بین ارباب و برده



Master shifts out data to Slave, and shift in data from Slave

[http://upload.wikimedia.org/wikipedia/commons/thumb/b/bb/SPI\\_8-bit\\_circular\\_transfer.svg/400px-SPI\\_8-bit\\_circular\\_transfer.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/b/bb/SPI_8-bit_circular_transfer.svg/400px-SPI_8-bit_circular_transfer.svg.png)

- عملیات خواندن یا نوشتن باید توسط ارباب صورت پذیرد.
- برده این امکان را ندارد که خودش شروع کننده ارتباط باشد
- ارباب برای خواندن دیتا از برده، باید یک عملیات نوشتن الکی انجام دهد

# انواع مودهای ارتباطی در SPI

| Mode      | Clock Polarity (CPOL) | Clock Phase (CPHA) | Output Edge | Data Capture |
|-----------|-----------------------|--------------------|-------------|--------------|
| SPI_MODE0 | 0                     | 0                  | Falling     | Rising       |
| SPI_MODE1 | 0                     | 1                  | Rising      | Falling      |
| SPI_MODE2 | 1                     | 0                  | Rising      | Falling      |
| SPI_MODE3 | 1                     | 1                  | Falling     | Rising       |

# UNO در آردوینو SPI

*Arduino digital pins used for SPI*

| SPI signal                | Arduino Uno | Arduino Mega |
|---------------------------|-------------|--------------|
| SCLK (clock)              | 13          | 52           |
| MISO (data out)           | 12          | 50           |
| MOSI (data in)            | 11          | 51           |
| SS/CS (slave/chip select) | 10          | 53           |

# برنامه نویسی SPI در آردوینو

```
#include <SPI.h>
```

```
const int slaveSelectPin = 10;
```

```
SPI.beginTransaction(SPISettings(14000000, MSBFIRST, SPI_MODE0));
```

```
SPI.begin(); // حالت ساده شده شروع
```

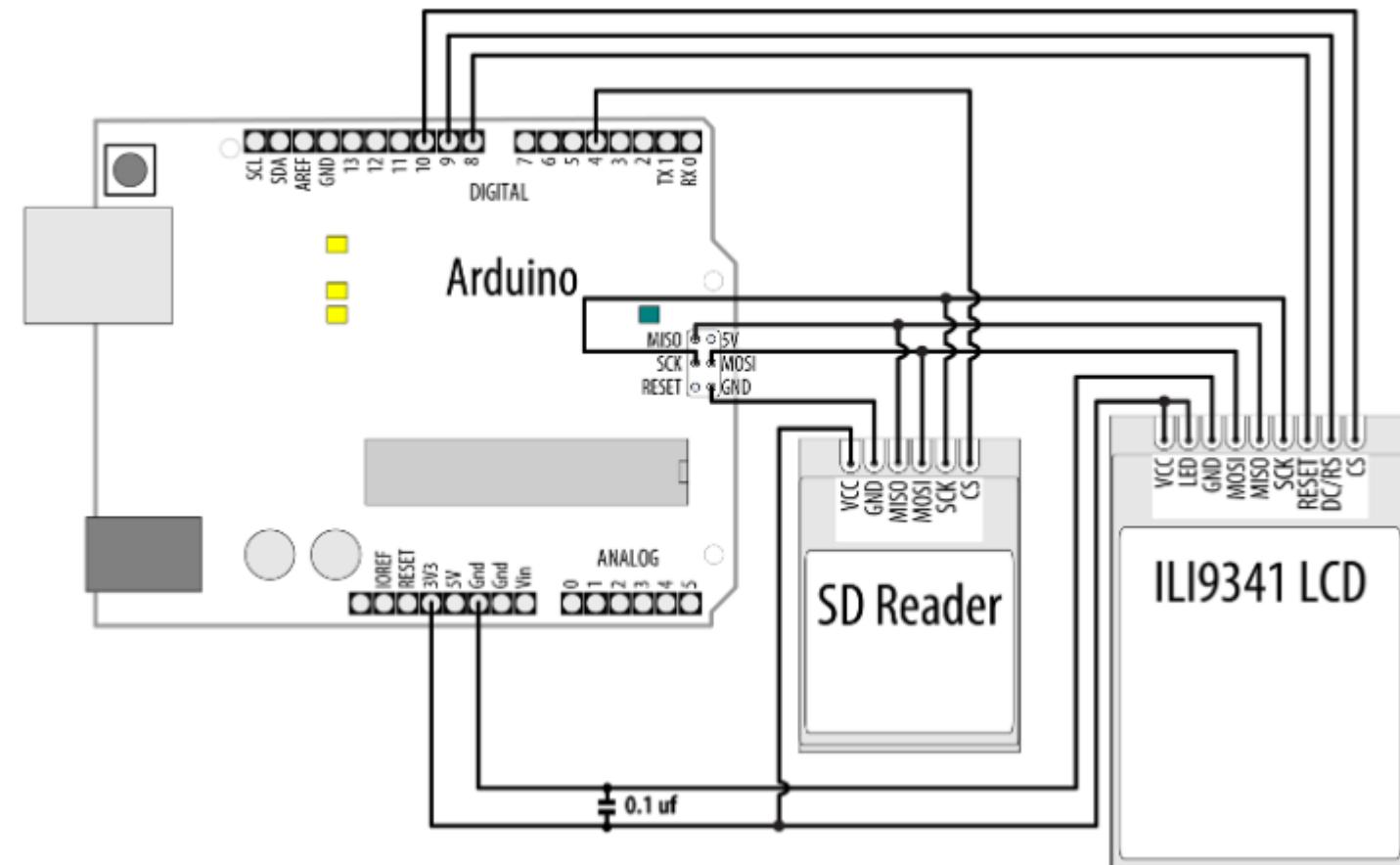
```
digitalWrite(slaveSelectPin, LOW); // فعال کردن دستگاه
```

```
SPI.transfer(x); // نوشتن (ارسال)
```

```
Y= SPI.transfer(0); // خواندن (دریافت)
```

```
digitalWrite(slaveSelectPin, HIGH); // غیر فعال کردن دستگاه
```

# مثال کاربردی





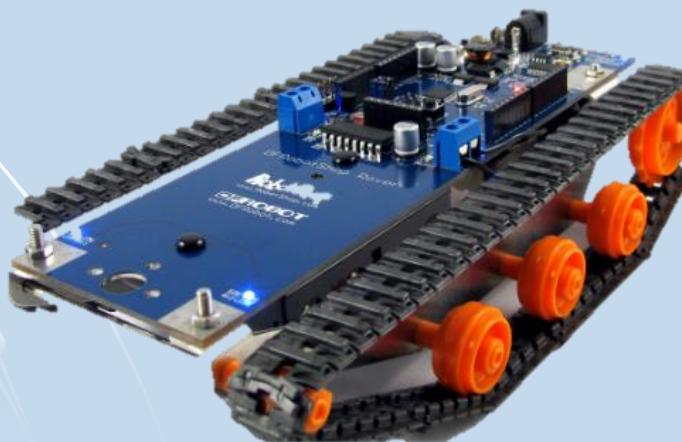
دانشکده مهندسی کامپیوتر

درس برنامه نویسی تجهیزات اینترنت اشیا

نمیسال دوم ۱۴۰۲

ماژول وای فای

دکتر علی بهلوانی



# چرا وای فای WiFi



- استفاده بسیار گسترده در همه تجهیزات وایرلس
- سرعت انتقال بالا
- ارزان قیمت بودن چیپهای آن نسبت به سایر تکنولوژی های بیسیم
- پلی برای ارتباط با اینترنت

# چیپ ESP8266

- میکرو کنترلر + فرستنده و گیرنده WiFi
- دارای پردازنده با قدرت پردازشی ۳۲ بیت
- پشتیبانی از پروتکل TCP/IP
- محدوده دمای کاری از  $-40$  تا  $125$  درجه
- ولتاژ کاری بین  $3$  تا  $3.6$  ولت
- پشتیبانی از دستورات AT Command
- فرکانس کاری  $2.4$  گیگاهرتز

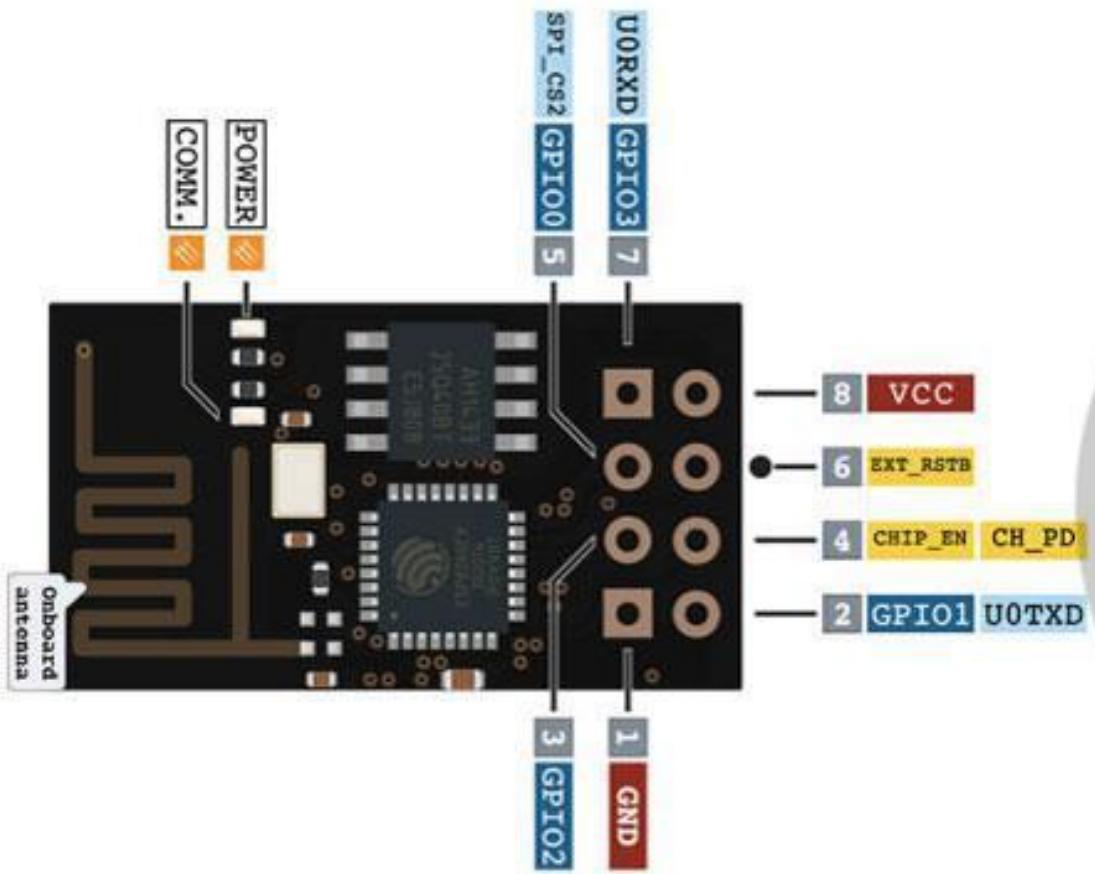
# انواع ماژولهای ساخته شده بر اساس 8266



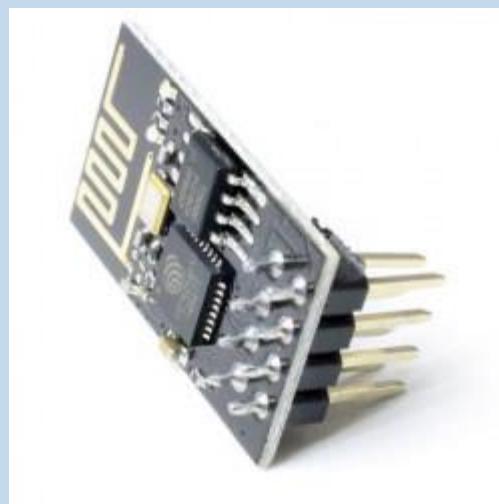
# تفاوت مازولها

- قابلیت نصب روی برد بورد
- پروتکل ارتباطی با بورد(سریال، یو اس بی)
- تعداد پایه های ورودی/خروجی
- ابعاد بورد
- امکان یا عدم امکان نصب آنتن
- قابلیت استفاده در صنعت

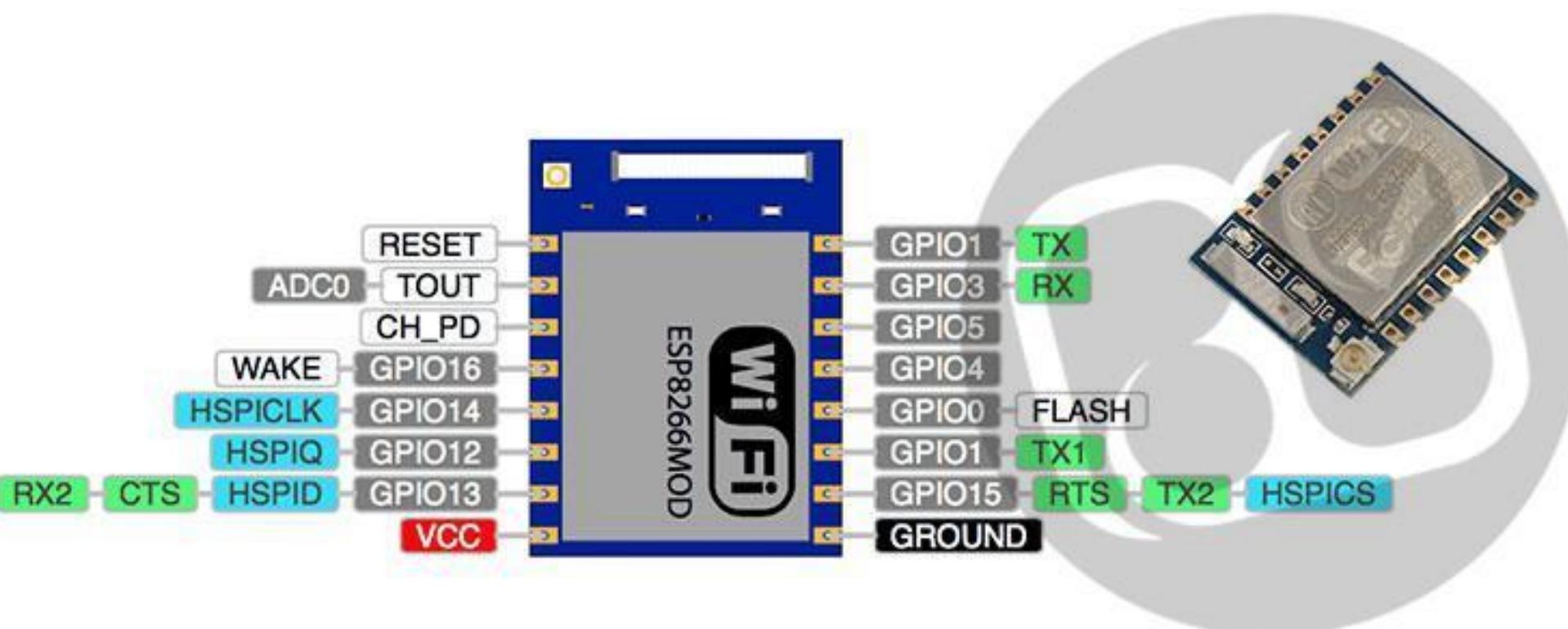
# ESP8266-01S



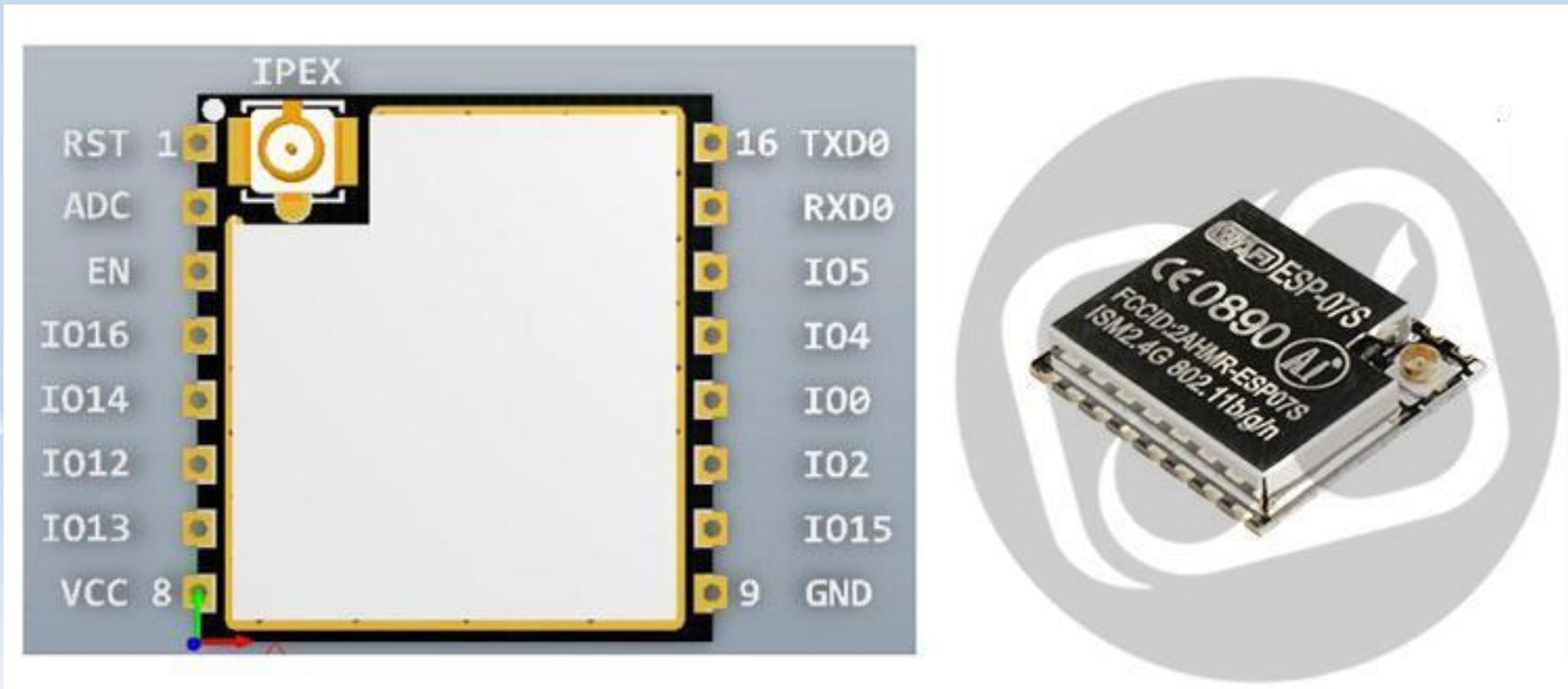
# Wifi ESP-01, ESP8266-01E



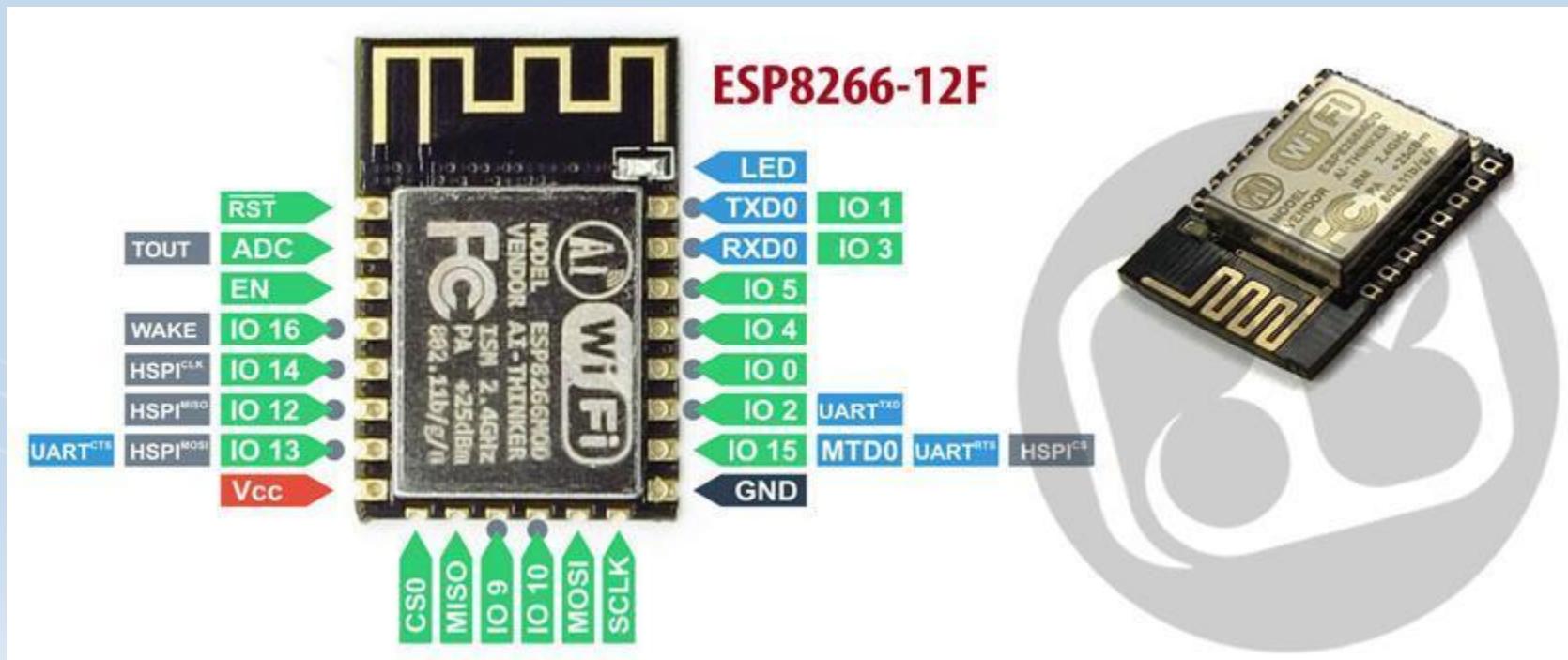
# ESP8266-07



# ESP8266-07S



# ESP8266-12F

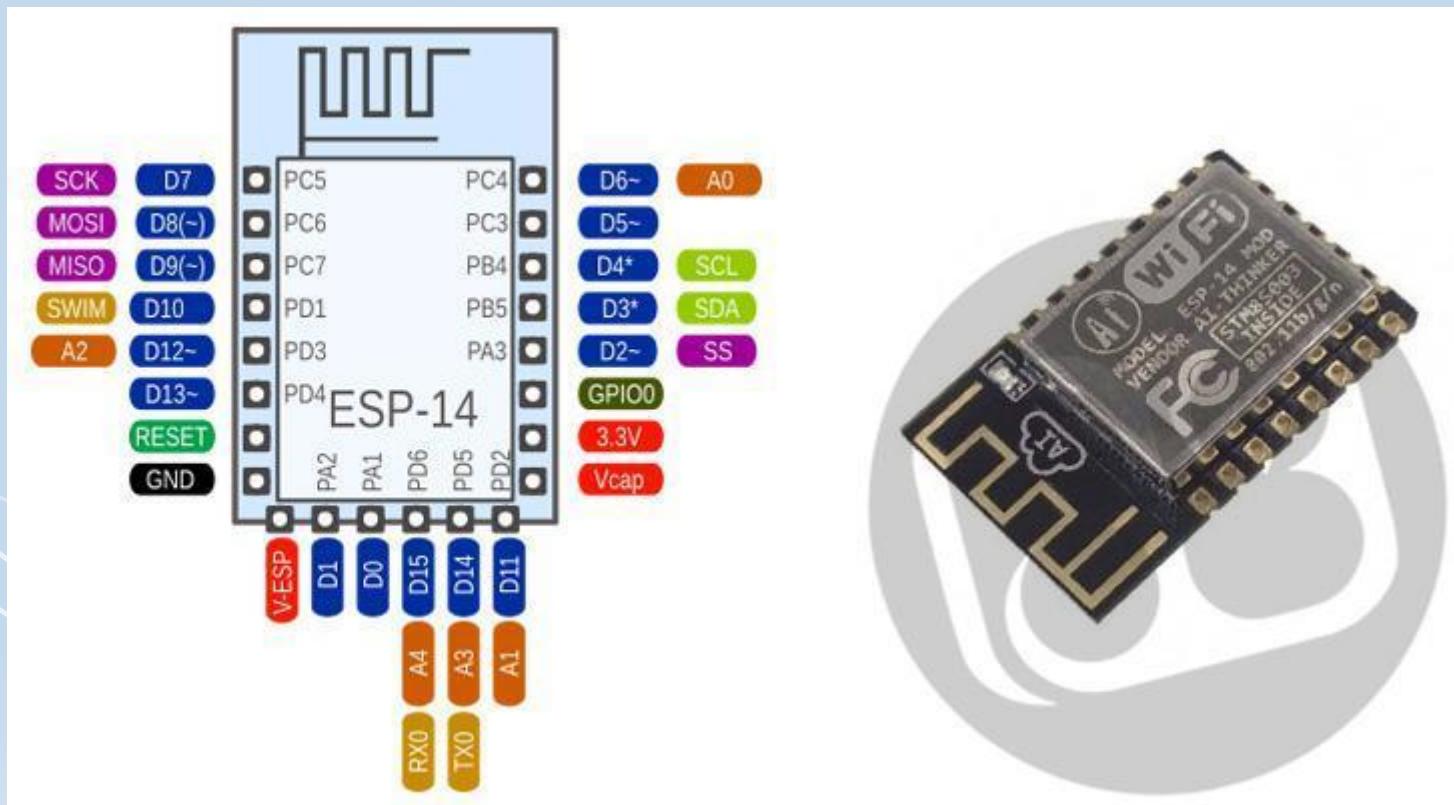


# ESP8266-12E

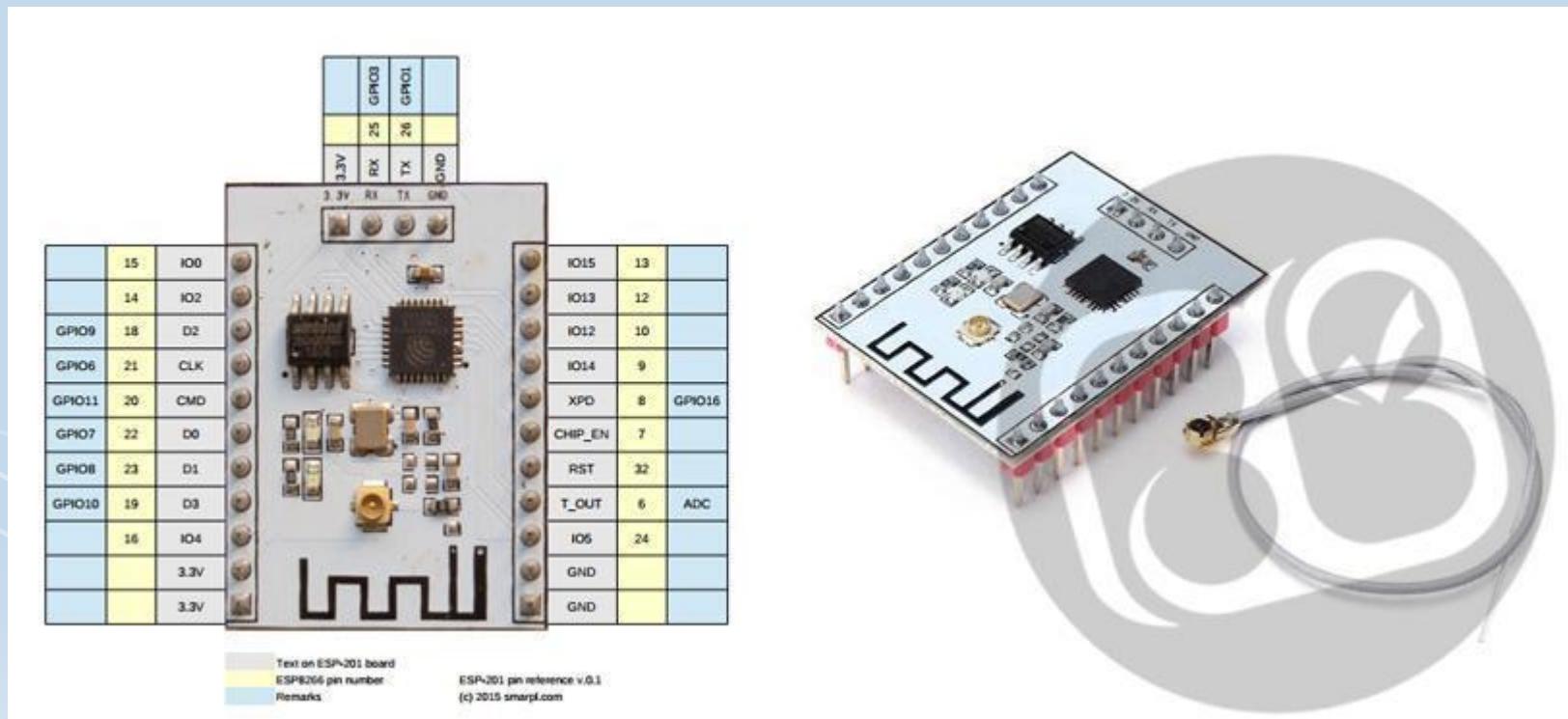
ماژول وای فای صنعتی Wifi



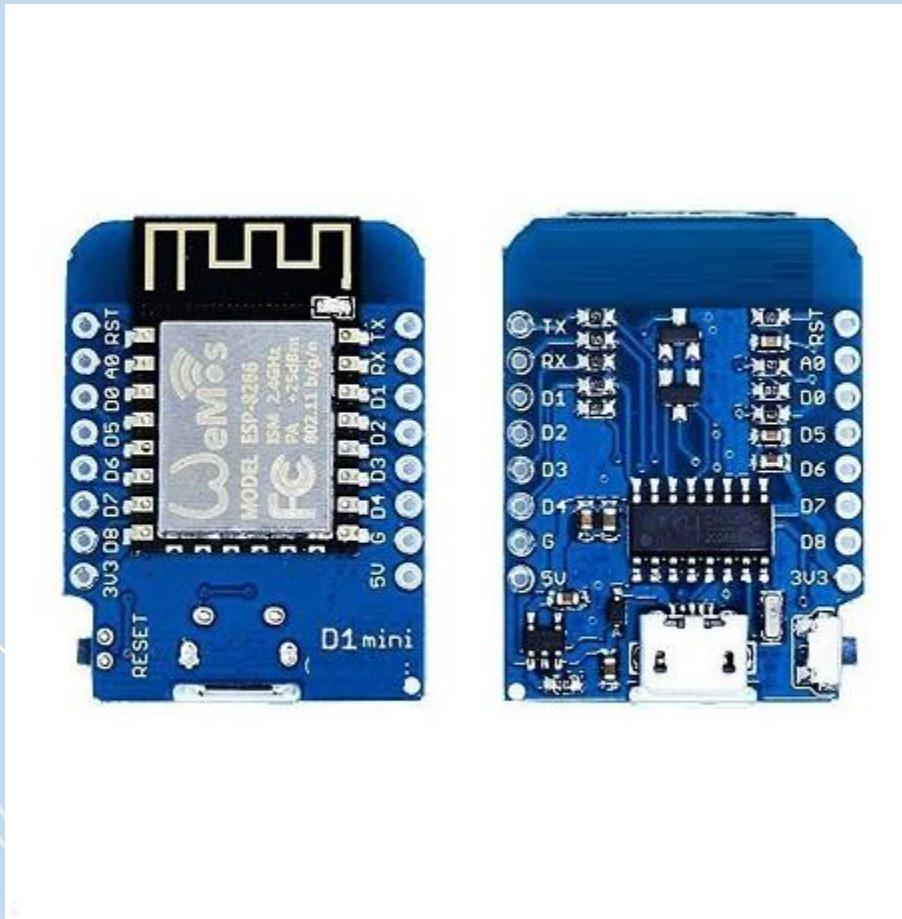
# ESP8266-14



# ESP8266-ESP201



# Wemos D1 Mini



# NODEMCU WEMOS



ماژول NodeMCU  
محصول  
دارای هسته WeMos  
و مبدل سریال  
ESP8266  
CP2102

# ESP8266 NODEMCU CH340

CP2102  
CH340



# ESP8266 NODEMCU CP2102



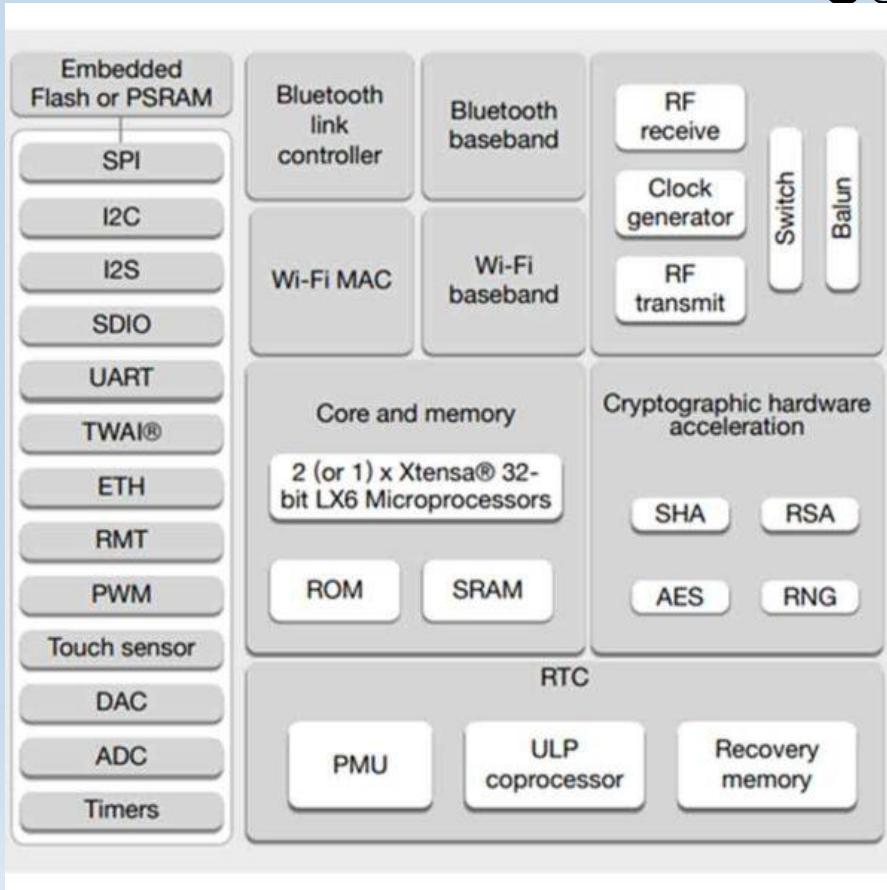
# چیپ ESP32

- نسل تکامل یافته ESP2866
- قابلیت وای فای و بلوتوث
- CPU and Memory: Xtensa® 32-bit LX6  
Dual-core processor, up to 600 DMIPS  
4 MByte SPI Flash  
448 KByte ROM  
520 KByte SRAM  
16 KByte SRAM in RTC

# نسلها و مدل‌های مختلف ماژول وای فای



# قابلیتهای عمدۀ ماژول ESP-32



## قابلیتهای جدید ESP-32

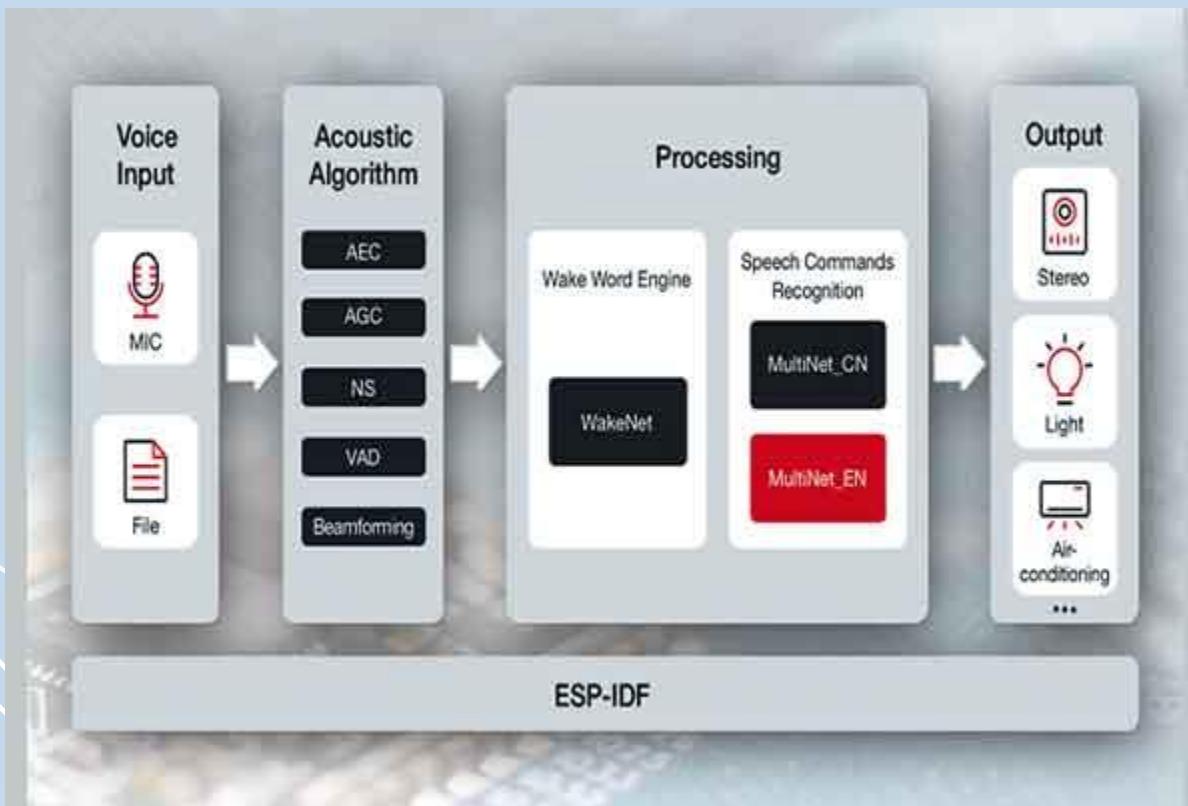
- اضافه شدن قابلیت بلوتوث
- اضافه شدن شتاب دهنده های رمزگننده
- اضافه شدن پروتکلهای ارتباطی بیشتر

# مقایسه نسخه های مختلف مازول وای فای

ESP8266 vs ESP32 vs ESP32-S2

| FEATURES              | ESP8266                        | ESP32                                                                  | ESP32-S2                                                                                              |
|-----------------------|--------------------------------|------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| Release Year          | 2014                           | 2016                                                                   | 2019                                                                                                  |
| Microcontroller       | Xtensa single-core 32-bit L106 | Xtensa single/dual-core 32-bit                                         | Xtensa single-core 32-bit LX7                                                                         |
| Clock Frequency       | 80 MHz                         | 160/240 MHz                                                            | 240 MHz                                                                                               |
| Co-processor          | ✗                              | ULP                                                                    | ULP (RISC-V)                                                                                          |
| SRAM                  | 160KB                          | 520KB                                                                  | 320KB                                                                                                 |
| RTC Memory            | ✗                              | 16KB                                                                   | 16KB                                                                                                  |
| External SPIRAM       | Up to 16MB                     | Up to 16MB                                                             | Up to 128MB                                                                                           |
| External Flash        | ✗                              | ✗                                                                      | Up to 1G                                                                                              |
| Wi-Fi (802.11 b/g/n)  | HT20                           | HT20                                                                   | HT20                                                                                                  |
| ESP-MESH              | ✓                              | ✓                                                                      | ✓                                                                                                     |
| Bluetooth             | ✗                              | BT 4.2, BR/EDR, BLE                                                    | ✗                                                                                                     |
| Ethernet              | ✗                              | 10/100 Mbps                                                            | ✗                                                                                                     |
| CAN                   | ✗                              | 2                                                                      | ✗                                                                                                     |
| Time of Flight        | ✗                              | ✗                                                                      | ✓                                                                                                     |
| GPIO (total)          | 16                             | 34                                                                     | 43                                                                                                    |
| Touch Sensors         | ✗                              | 10                                                                     | 14                                                                                                    |
| SPI                   | 2                              | 4                                                                      | 4 (OSPI)                                                                                              |
| I2C                   | 1 (soft)                       | 2                                                                      | 2                                                                                                     |
| I2S                   | 2                              | 2                                                                      | 1                                                                                                     |
| UART                  | 2 (1.5 actually)               | 3                                                                      | 2                                                                                                     |
| ADC                   | 1(10-bit)                      | 18 (12-bit)                                                            | 20 (12-bit)                                                                                           |
| DAC                   | ✗                              | 2 (8-bit)                                                              | 2 (8-bit)                                                                                             |
| PWM (soft)            | 8                              | 16                                                                     | 8                                                                                                     |
| SDMMC                 | ✗                              | ✓                                                                      | ✗                                                                                                     |
| USB OTG               | ✗                              | ✗                                                                      | ✓                                                                                                     |
| LCD Interface         | ✗                              | ✗                                                                      | ✓                                                                                                     |
| Camera Interface      | ✗                              | ✗                                                                      | ✓                                                                                                     |
| Temperature Sensor    | ✗                              | ✓                                                                      | ✓                                                                                                     |
| Hall sensor           | ✗                              | ✓                                                                      | ✗                                                                                                     |
| Security              | ✗                              | Secure boot Flash encryption 1024-bit OTP<br>AES, SHA-2, RSA, ECC, RNG | Secure boot Flash encryption 4096-bit OTP<br>AES-128/192/256, SHA-2, RSA, RNG, HMAC, Digital Signatur |
| Crypto                | ✗                              |                                                                        |                                                                                                       |
| Low Power Consumption | 20uA                           | 10uA deep sleep                                                        | 5uA in idle mode, 24uA at 1% duty cycle                                                               |

# نمونه کاربرد مژول



# بوردهای توسعه ESP32

- ESP32-WROOM-32 •
- ESP32-WROOM-32 •
- ESP32-WROOM-32 •
- Node-MCU •
- ESP32-DevKitC •
- ESP32CAM •

# ESP32CAM



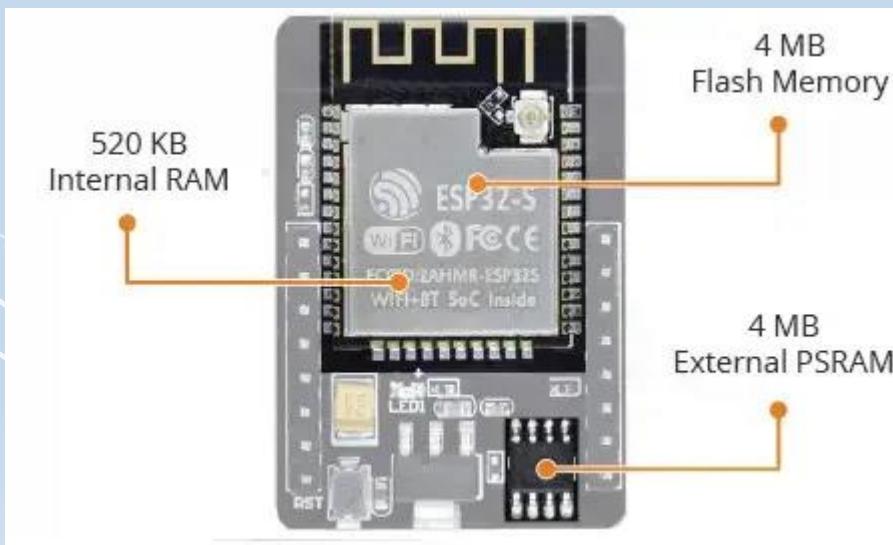
# ESP32CAM

نوع S از نوع ESP32

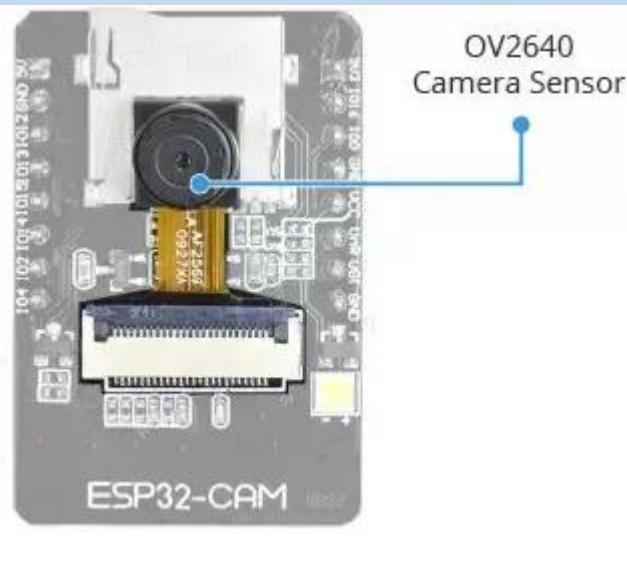


# ESP32CAM

حافظه ها



# ESP32CAM



- دوربین **OV2640**
- رزولوشن ۲ مگاپیکسل
- $1600 \times 1200$  پیکسل
- سرعت انتقال تصویر ۱۵ تا ۶۰ فریم بر ثانیه
- دارای فرمت های خروجی **YUV422**, **RGB555**, **RGB565**, **YUV420** و داده های فشرده ۸ بیتی

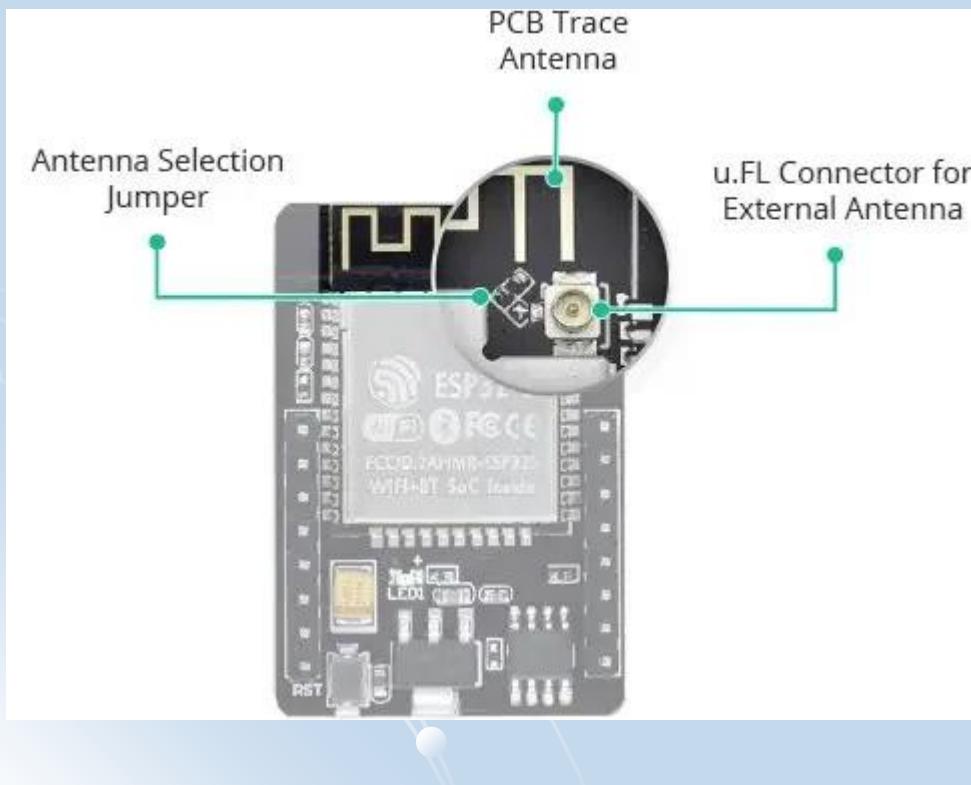
# ESP32CAM

• حافظه جانبی



# ESP32CAM

• آنتن



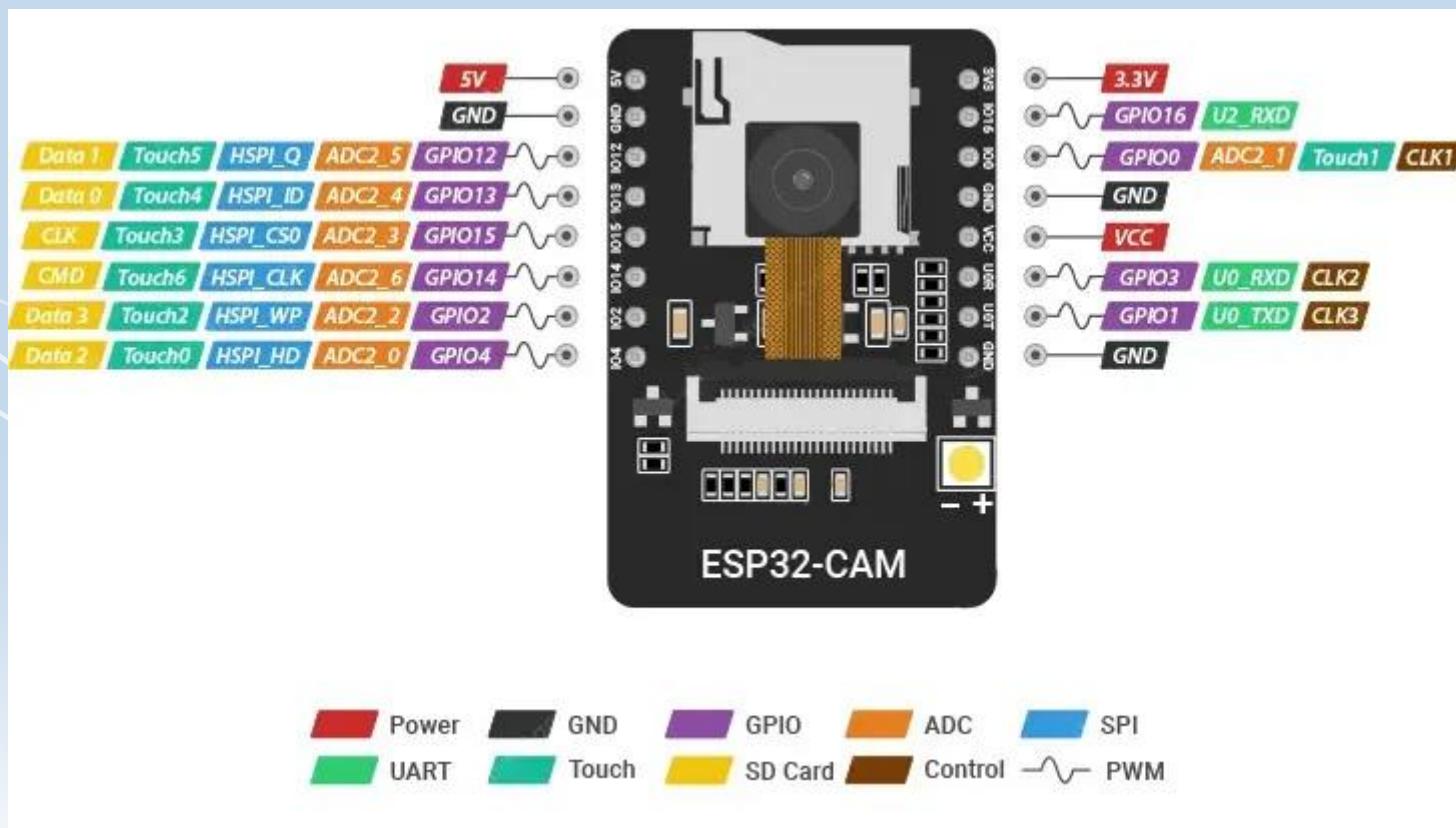
# ESP32CAM

• LED‌های بورد



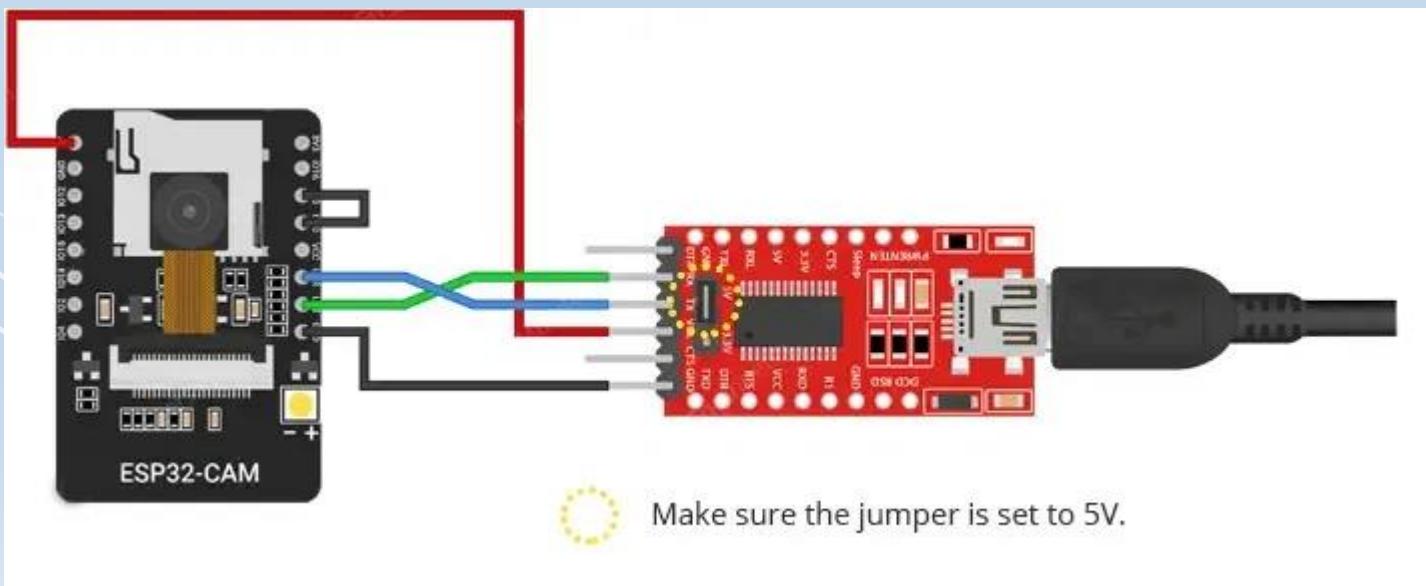
# ESP32CAM

پایه های در دسترس



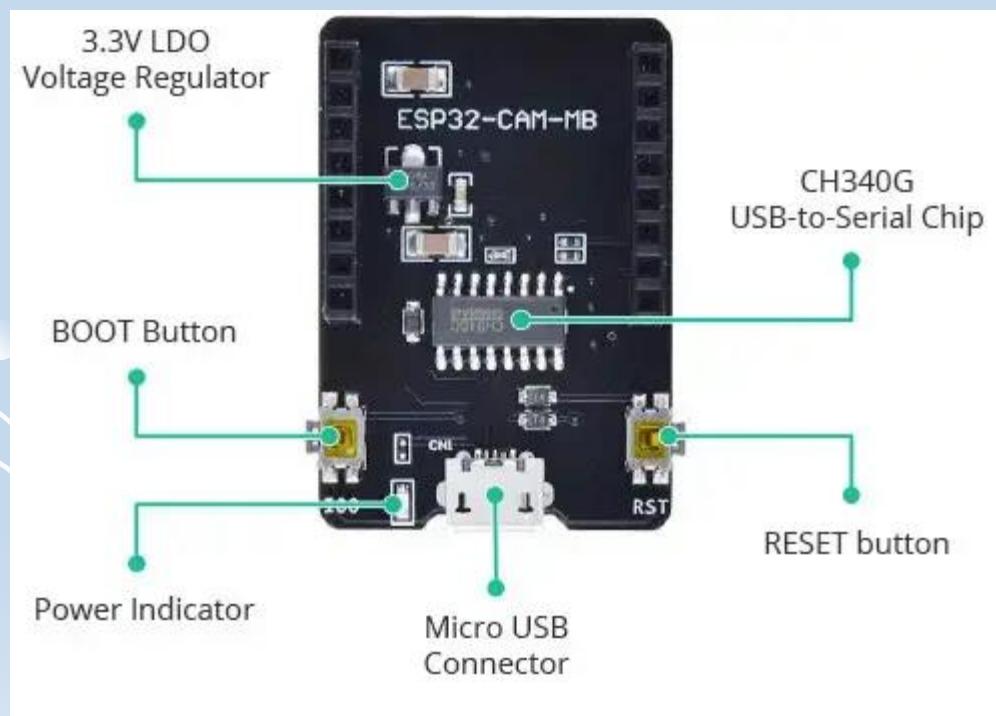
# ESP32CAM

• نحوه پردازش کردن(روش اول)



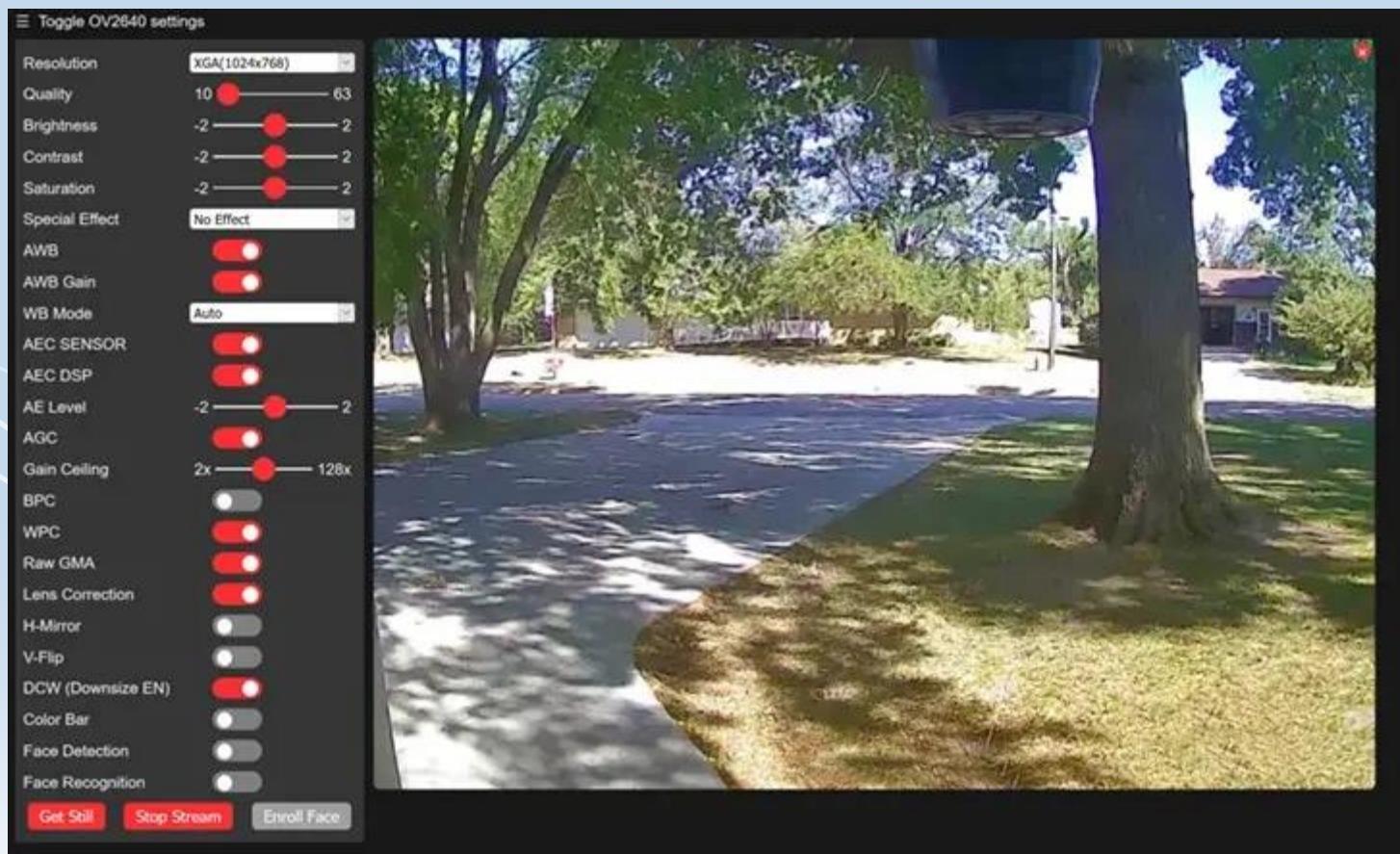
# ESP32CAM

• نحوه پردازش داده های دوربینی (روش دوم)



# ESP32CAM

نمونه برنامه





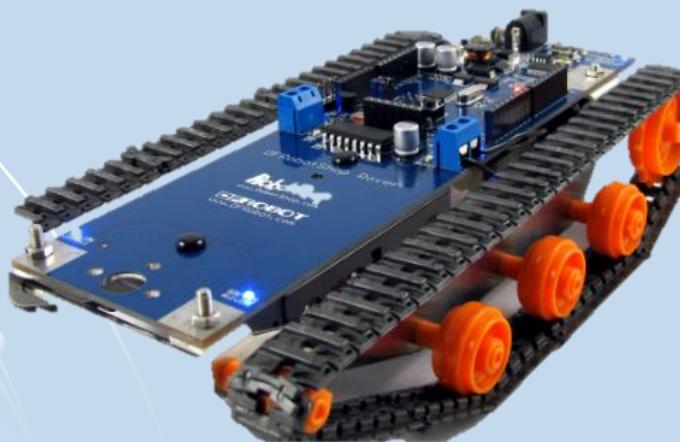
دانشکده مهندسی کامپیوتر

درس برنامه نویسی تجهیزات اینترنت اشیا

نهمیال دوم ۱۴۰۲

برنامه نویسی ماژول وای فای

دکتر علی بهلوانی



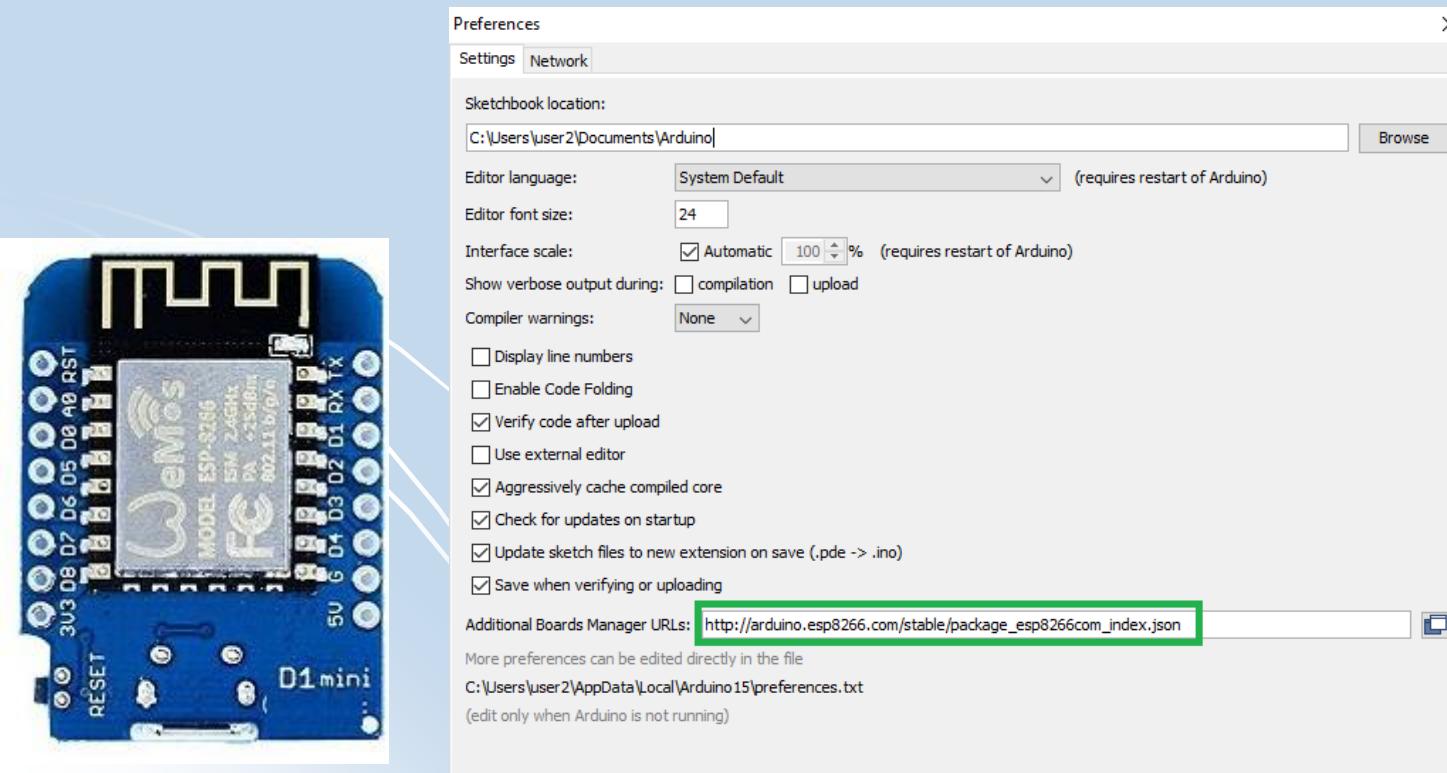
# نحوه برنامه نویسی ماژول وای فای



# نصب کتابخانه های لازم برای مازولهای

## ESP8266

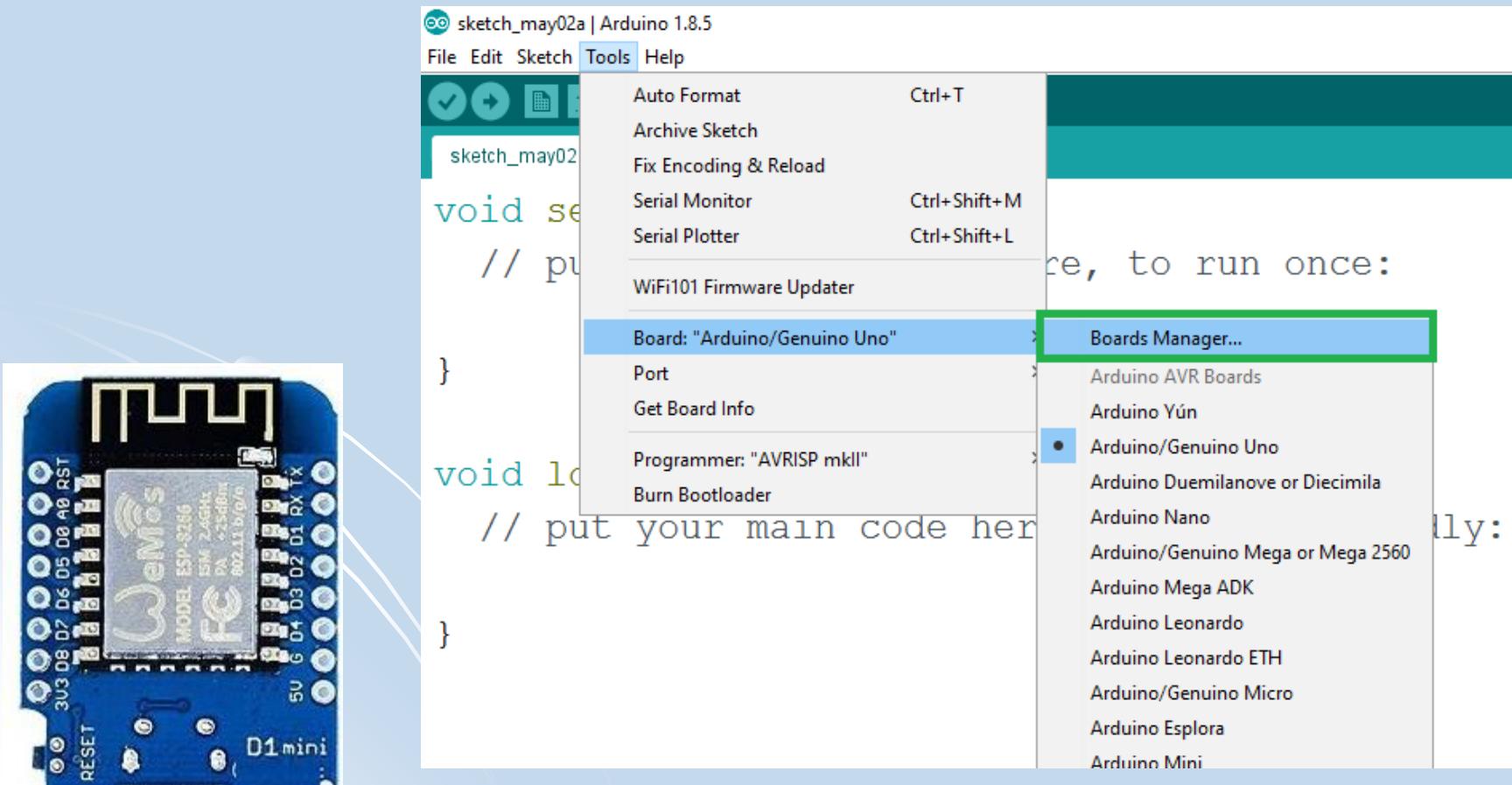
- انتخاب گزینه preferences از منوی فایل
- درج لینک مربوط به مازول در محل مشخص شده
- [http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)



# نصب کتابخانه های لازم برای مازولهای

## ESP8266

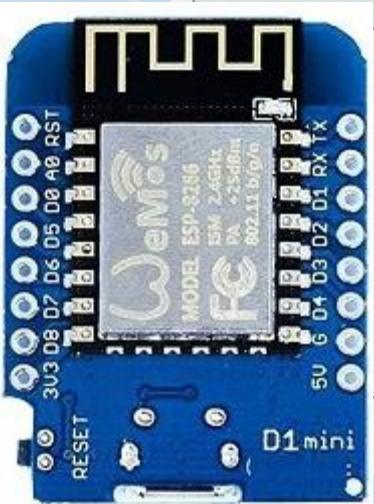
### • انتخاب گزینه Boards Manager



# نصب کتابخانه های لازم برای مازولهای

## ESP8266

● جستجوی مازول esp8266



Boards Manager

Type All esp8266

**esp8266 by ESP8266 Community**

Boards included in this package:

Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, ESPresso Lite 1.0, ESPresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SparkFun Blynk Board, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(WEMOS) D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Systems gen4 IoT Range, Digistump Oak, WiFiduino, Amperka WiFi Slot, Seeed Wio Link, ESPectro Core, Schirmilabs Eduino WiFi, ITEAD Sonoff, DOIT ESP-Mx DevKit (ESP8285).

[More info](#)

2.7.4

Install

Close

# نصب کتابخانه های لازم برای مازولهای

## ESP8266

● چک کردن موفقیت در نصب

The image shows the Arduino IDE's Boards Manager interface. A search bar at the top has 'esp8266' typed into it. Below the search bar, a package titled 'esp8266 by ESP8266 Community version 2.7.4' is listed with the status 'INSTALLED'. A green box highlights the 'INSTALLED' status. To the right of the main window, a sidebar titled 'Boards Manager...' lists various Arduino boards. The 'Arduino/Genuino Uno' board is currently selected, indicated by a blue background. Other boards listed include Arduino AVR Boards, Arduino Yún, Arduino/Genuino Uno, Arduino Duemilanove or Diecimila, Arduino Nano, Arduino/Genuino Mega or Mega 256, Arduino Mega ADK, Arduino Leonardo, Arduino Leonardo ETH, Arduino/Genuino Micro, Arduino Uno WiFi, and Adafruit Feather HUZZAH ESP8266. A green box highlights the 'ESP8266 Boards (2.7.4)' section.

Boards Manager

Type All esp8266

esp8266 by ESP8266 Community version 2.7.4 **INSTALLED**

Boards included in this package:

Generic ESP8266 Module, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, Invent One, XinaBox CW01, Espresso Lite 1.0, Espresso Lite 2.0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex MOD-WIFI-ESP8266(-DEV), SparkFun ESP8266 Thing, SparkFun ESP Board, SweetPea ESP-210, LOLIN(WEMOS) D1 R2 & mini, LOLIN(WEMOS) D1 mini Pro, LOLIN(V1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WiFiInfo, Arduino, 4D Systems gen4 IoT R. Amperka WiFi Slot, Seeed Wio Link, ESPectro Core, Schirmilabs Eduino WiFi, ITEAD Sonoff, DO More info

Board: "Arduino/Genuino Uno"

Port

Get Board Info

Programmer: "AVRISP mkII"

Burn Bootloader

Boards Manager...

Arduino AVR Boards

Arduino Yún

● Arduino/Genuino Uno

Arduino Duemilanove or Diecimila

Arduino Nano

Arduino/Genuino Mega or Mega 256

Arduino Mega ADK

Arduino Leonardo

Arduino Leonardo ETH

Arduino/Genuino Micro

Arduino Uno WiFi

ESP8266 Boards (2.7.4)

Generic ESP8266 Module

Generic ESP8285 Module

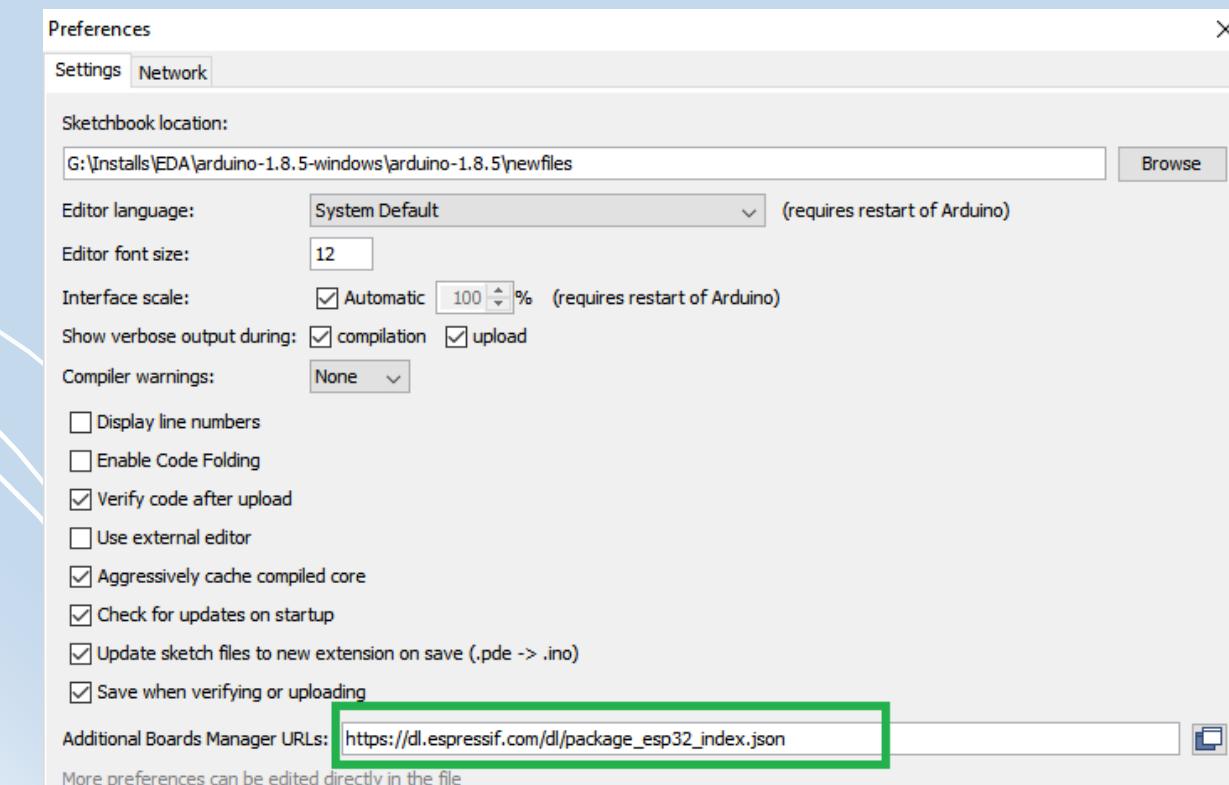
ESPDuino (ESP-13 Module)

Adafruit Feather HUZZAH ESP8266

# نصب کتابخانه های لازم برای مازولهای

## ESP32

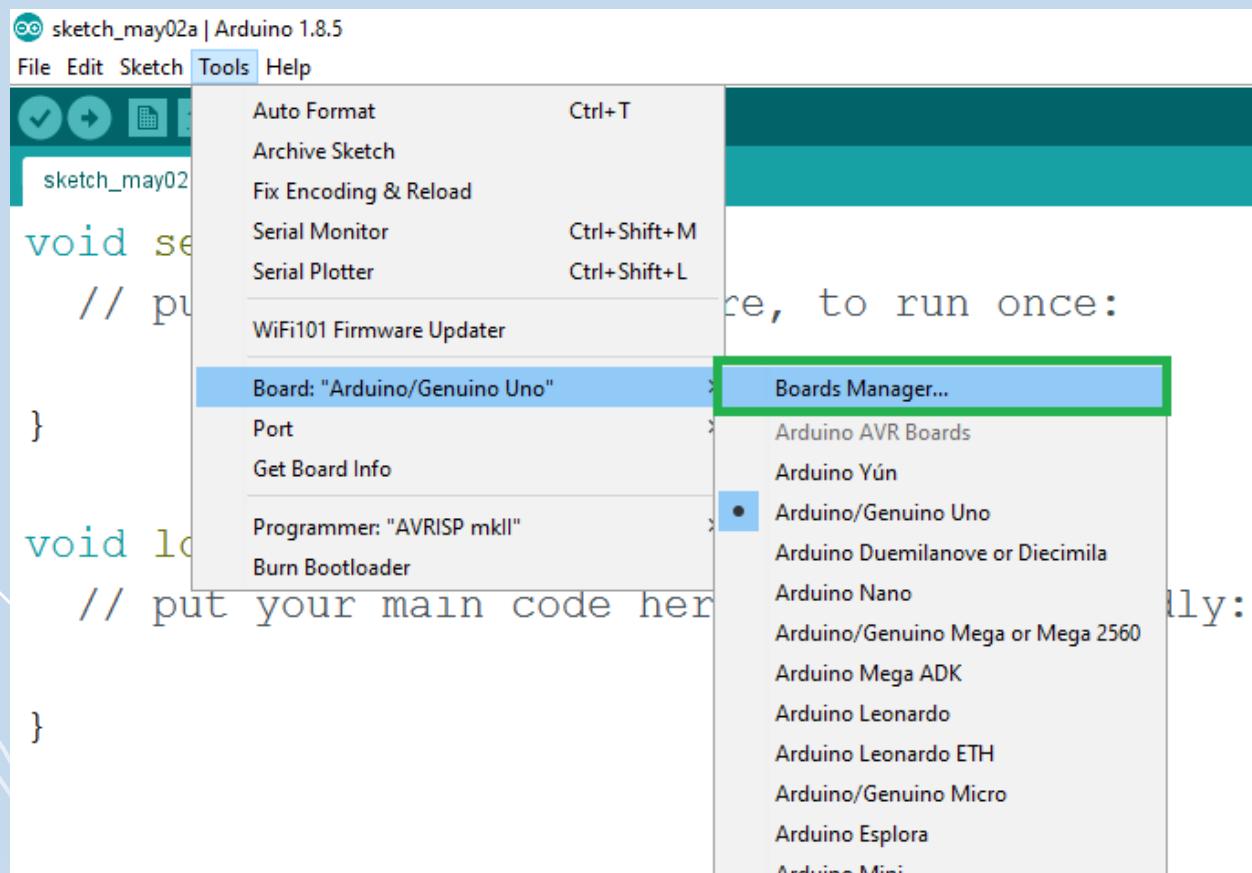
- انتخاب گزینه preferences از منوی فایل
- درج لینک مربوط به مازول در محل مشخص شده
- [https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)



# نصب کتابخانه های لازم برای مازولهای

## ESP32

### • انتخاب گزینه Boards Manager



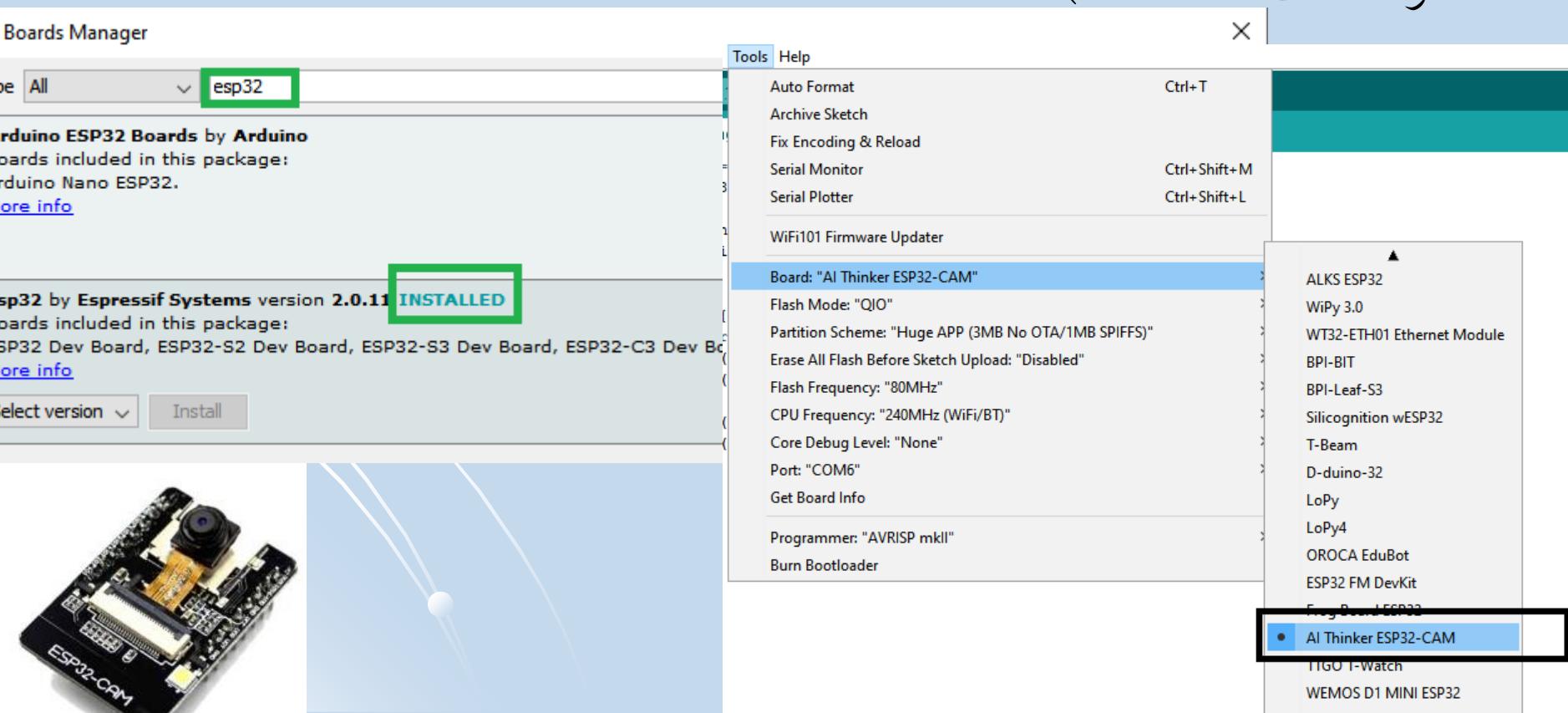
# نصب کتابخانه های لازم برای مازولهای ESP32

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



# نصب کتابخانه های لازم برای مازولهای ESP32

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

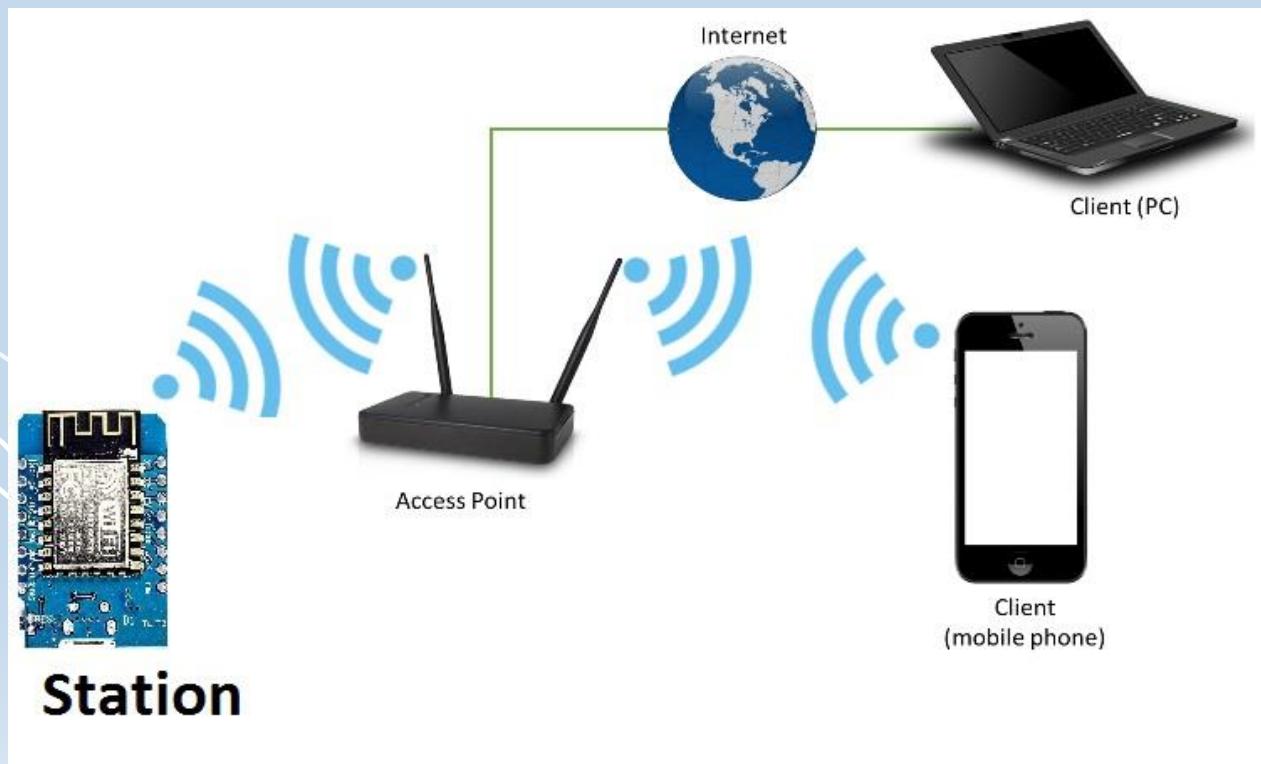


# نحوه برنامه نویسی مازول وای فای

- کار با مازول به دو صورت انجام می شود:
- اتصال آن به یک میکروکنترلر و فرمان دادن به آن از طریق **UART**
- برنامه نویسی مستقیم مازول(بدون نیاز به میکروکنترلر دیگر)

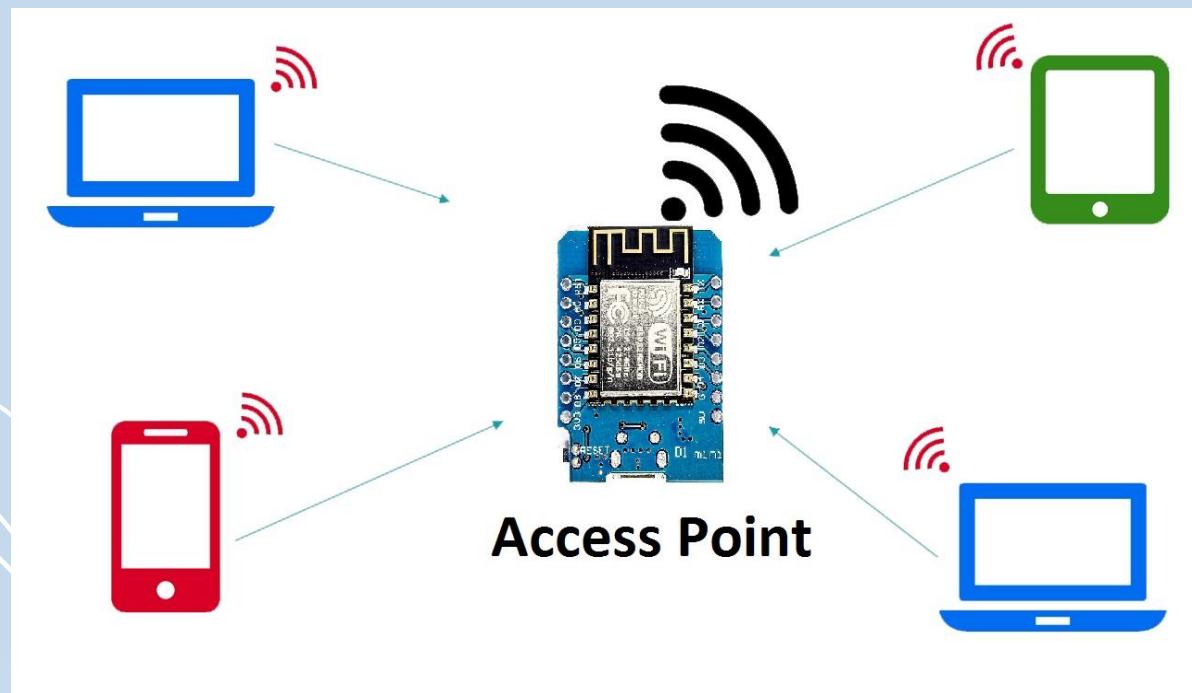
# مدھای کاری ماژول

STAtion •



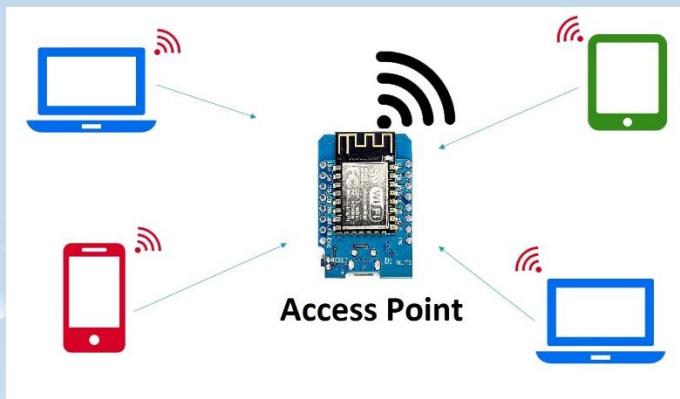
# مدھای کاری ماژول

(AP) Access Point •



# مدهای کاری مازول

BOTH مد ●



# 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
- مثلا فرمان زیر برای تست اتصال مورد استفاده قرار می گیرد:
  - AT\r\n
  - در پاسخ، ماژول پیام OK را برمی گرداند

# AT دسته بندی فرামین

- 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

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

### دستور AT+RST

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

### دستور AT+GMR

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

# اتصال به ماژول از طریق **UART**

- دستورات مربوط به وای فای (**WiFi AT Commands**)

## WiFi AT Commands

AT+CWMODE

AT+CWJAP

AT+CWLAP

AT+CWQAP

AT+CIPSTA

AT+CWSAP

AT+CWLIF

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/>

# اتصال به ماژول 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

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

### دستور AT+CWMODE – WIFI mode

| نوع دستور | دستور          | پاسخ برنامه         | عمل کرد                                                                   |
|-----------|----------------|---------------------|---------------------------------------------------------------------------|
| Test      | ?=AT+CWMODE    | CWMODE:+<br>(1-۳)OK | مقدارهای مجاز برای پارامتری این دستور را نمایش می‌دهد                     |
| Query     | ?AT+CWMODE     | CWMODE:mode+<br>OK  | مدی که ماژول در آن قرا ردارد را بر می‌گرداند. ( مد کلاینت ، هاست یا هردو) |
| Execute   | AT+CWMODE=mode | OK                  | مد ماژول را تعیین می‌کند                                                  |

# اتصال به ماژول از طریق UART

## ● برخی از دستورات

### دستور AT+CWJAP – Connect to AP

| نوع دستور | دستور             | پاسخ برنامه           | عمل کرد                                                                        |
|-----------|-------------------|-----------------------|--------------------------------------------------------------------------------|
| Query     | ?AT+CWJAP         | +<br>CWJAP:ssid<br>OK | نام اکسس پوینت (مودم) که ماژول به آن متصل است را بر میگرداند                   |
| Execute   | AT+CWJAP=ssid,pwd | OK                    | به وسیله این دستور به مودم وای فای یا اکسس پوینتی که رمزش را داریم متصل میشویم |

# اتصال به ماژول از طریق UART

## TCP/IP Commands

AT+CIPSTATUS

AT+CIPSTART

AT+CIFSR

AT+CIPMUX

AT+CIPSERVER

AT+CIUPDATE

- 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/>

# پرائگرام 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 ";
```

# پرائگرام 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("

");
client.println("Click here turn the LED on pin 5 ON
");
client.println("Click here turn the LED on pin 5 OFF
");
client.println("</html>");
delay(1);
Serial.println("Client disconnected");
Serial.println("");
}
```



دانشکده مهندسی کامپیوتر

درس برنامه نویسی تجهیزات اینترنت اشیا

نمیسال دوم ۱۴۰۲

راه اندازی دوربین

دکتر علی بهلوانی

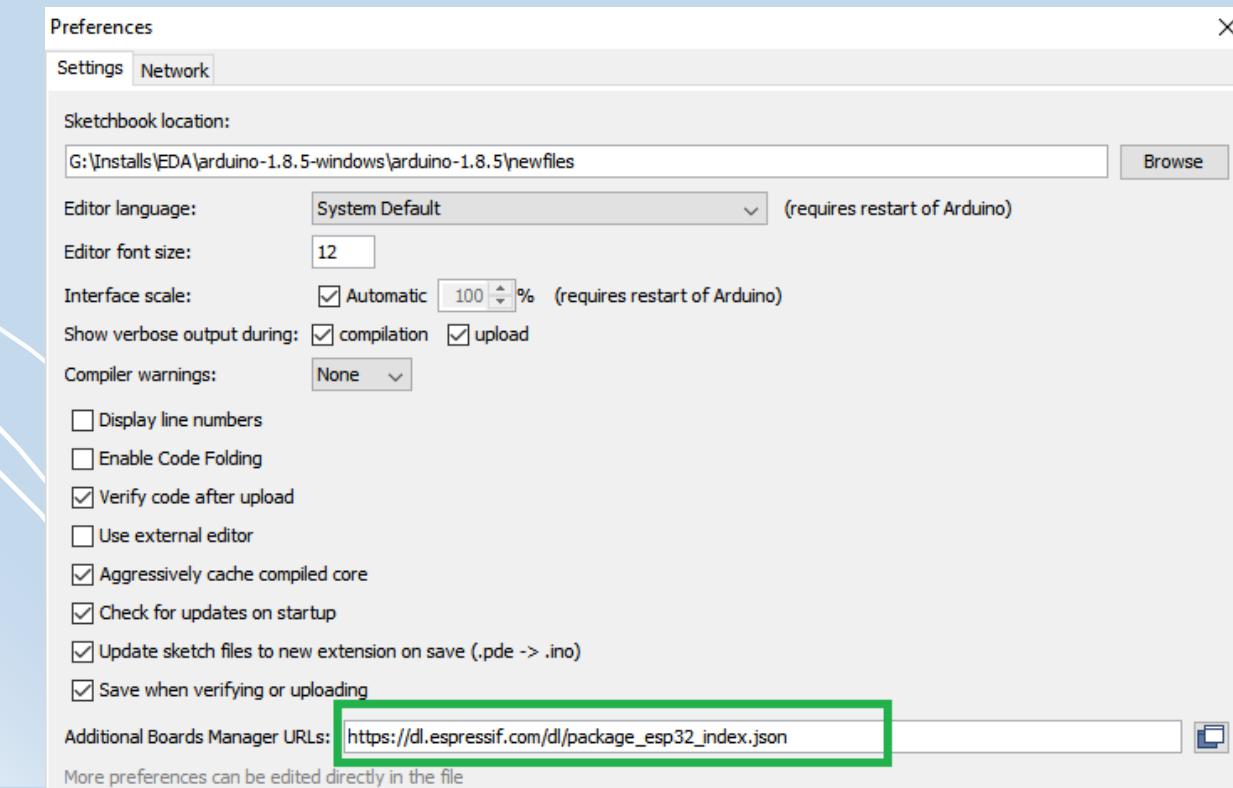


# مراحل راه اندازی دوربین ESP32-CAM

- نصب کتابخانه ESP32
- باز کردن یکی از پروژه های آماده مربوط به دوربین
- انجام تغییرات روی پروژه
- تنظیم بورد روی AI-Thinker ESP32-CAM کامپایل و آپلود آن روی بورد
- باز کردن ترمینال و تنظیم سرعت آن روی 115200
- ریست کردن بورد
- مشاهده IP اختصاص داده شده در ترمینال
- زدن آدرس IP در مرورگر و مشاهده صفحه و انتخاب گزینه Start Stream

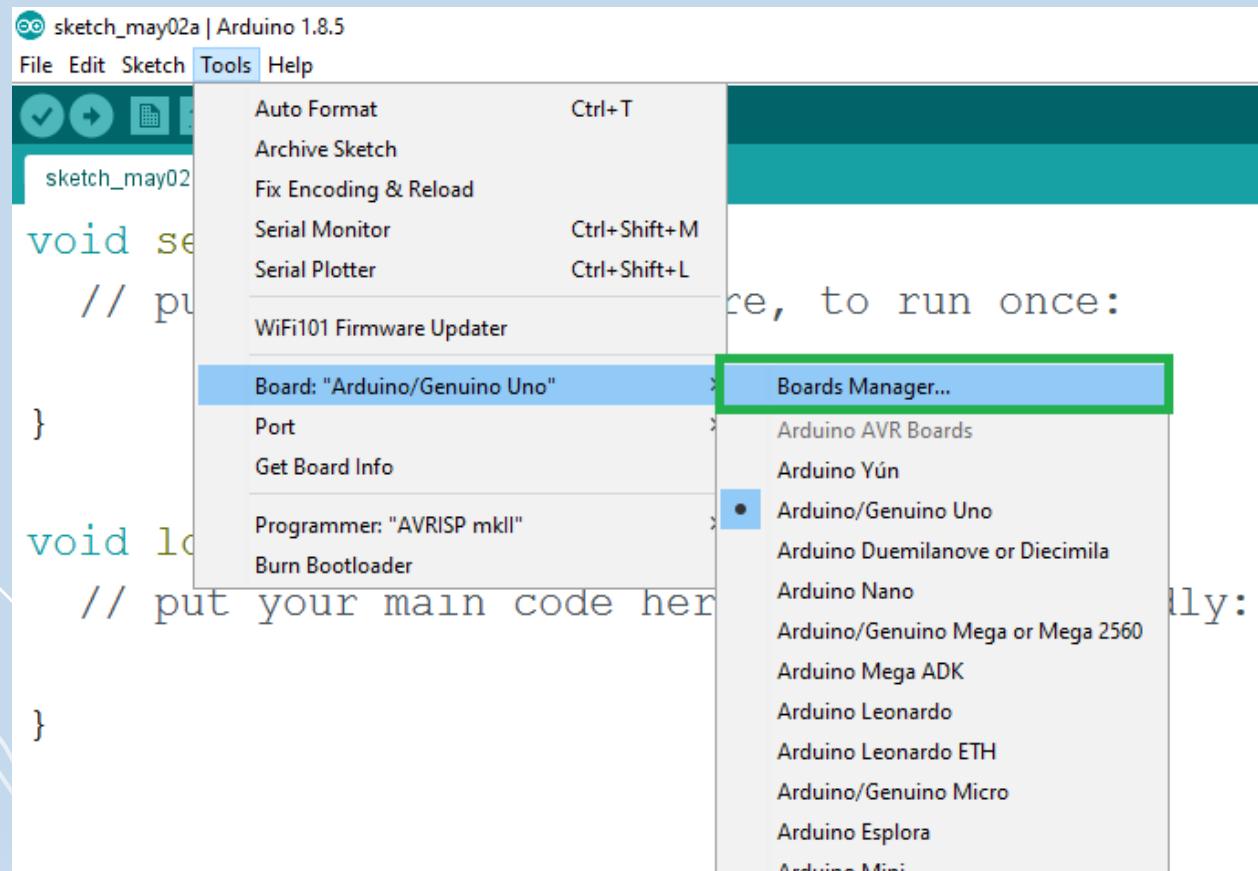
# نصب کتابخانه ESP32

- 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)



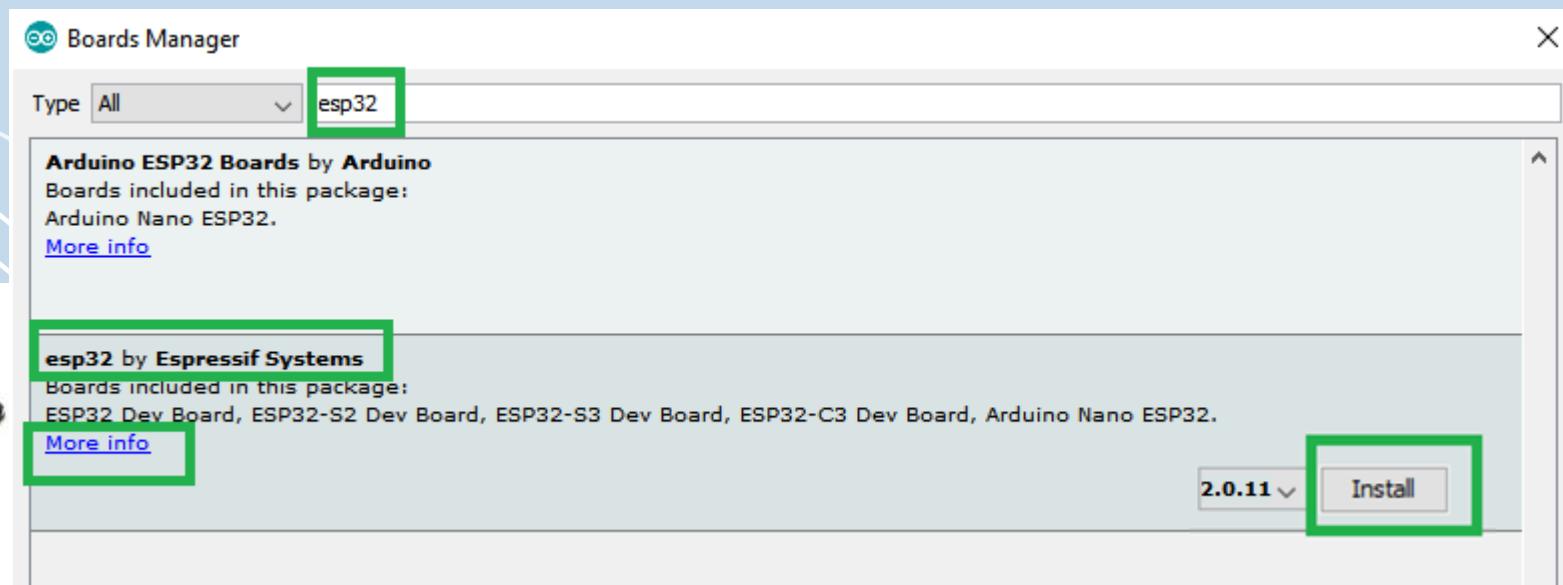
# نصب کتابخانه ESP32

## • انتخاب گزینه Boards Manager



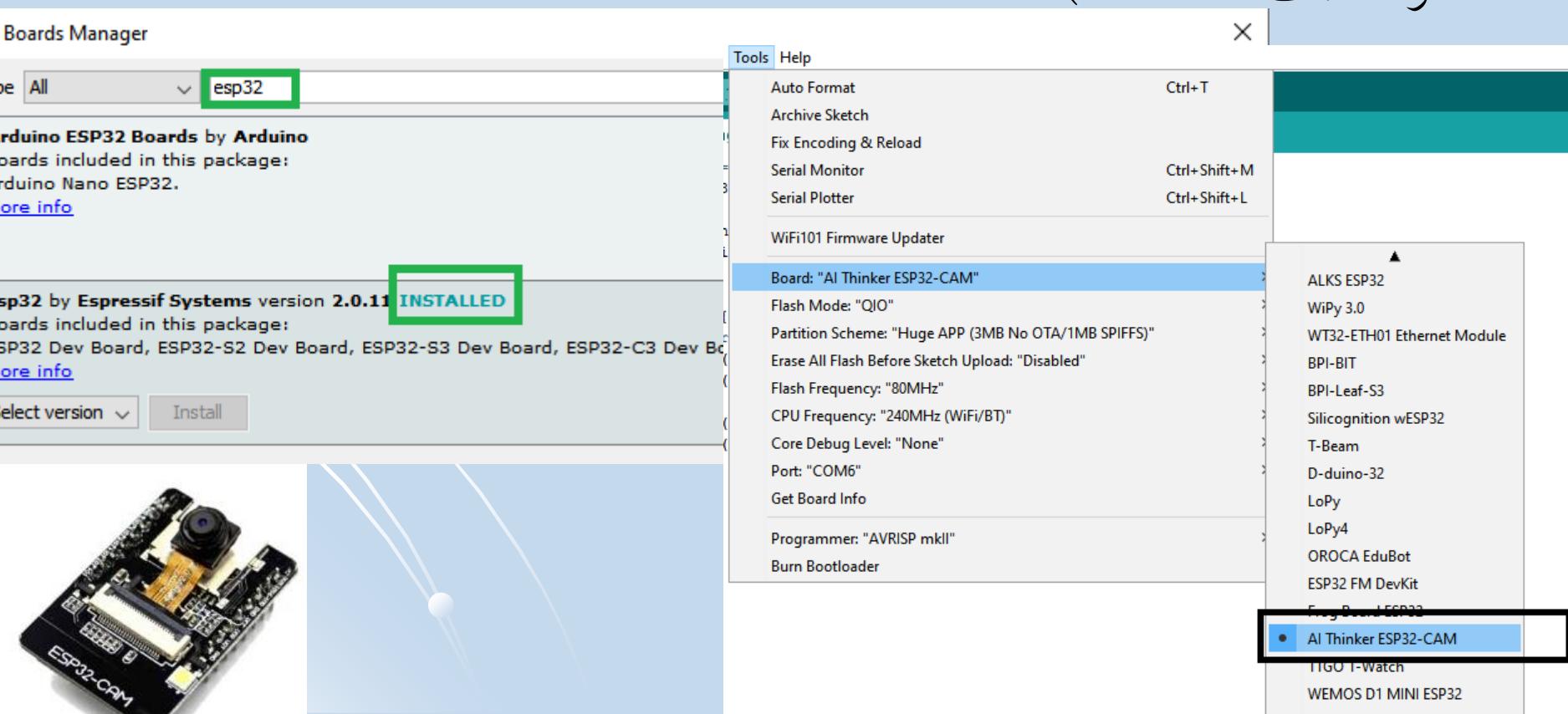
# نصب کتابخانه ESP32

- جستجوی مژول esp32
- در قسمت more Info گزینه Expressif را بزنید
- گزینه Install را انتخاب تا نصب شود



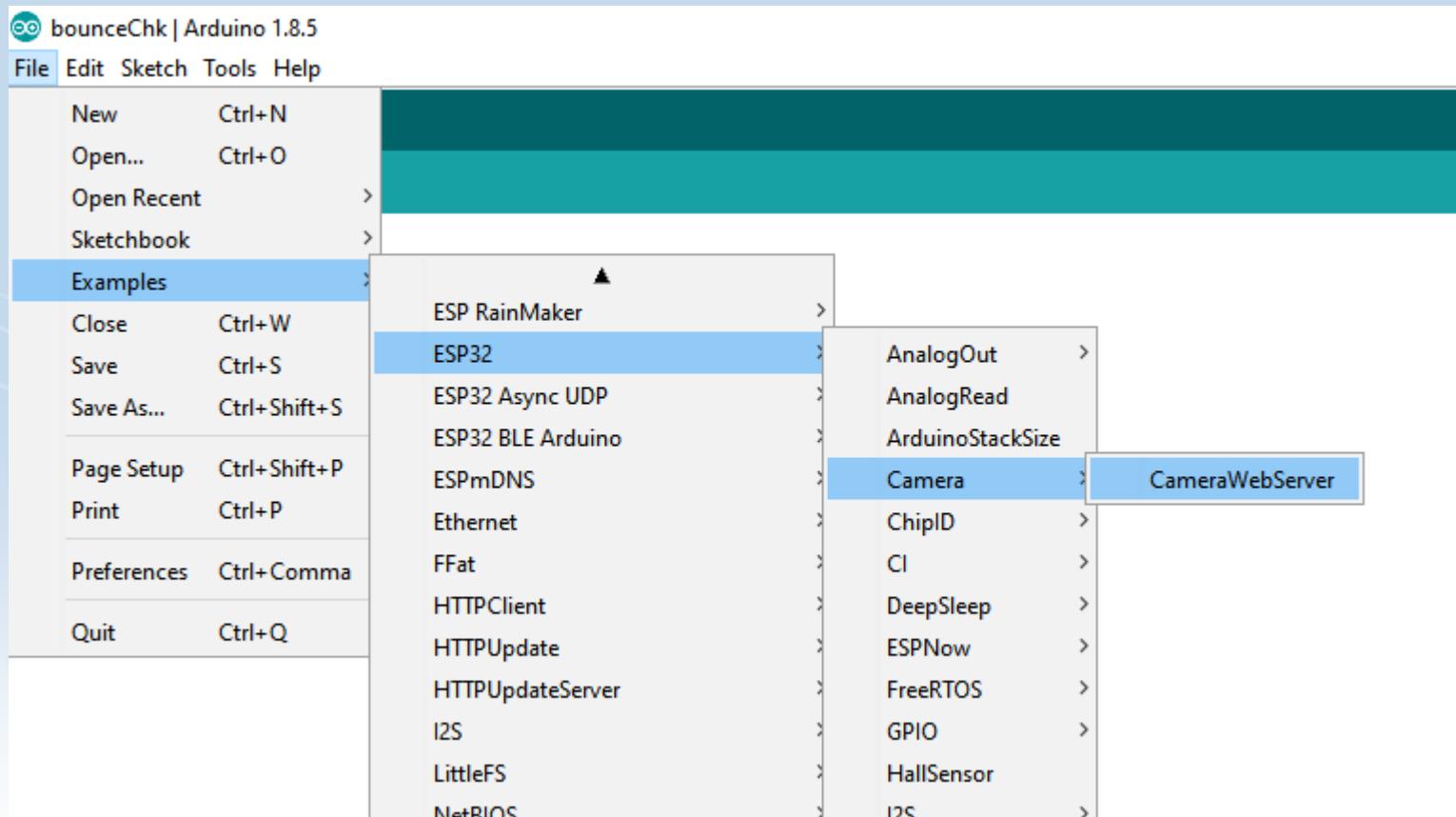
# نصب کتابخانه ESP32

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



# باز کردن پروژه آماده مربوط به دوربین

- طبق مسیر زیر، پروژه CameraWebServer را باز کنید



# انجام تغییرات روی پروژه

- کامنت کردن خط زیر (خط ۱۷ برنامه)

```
#define CAMERA_MODEL_ESP_EYE // Has PSRAM
```

- خارج کردن خط زیر از حالت کامنت(خط ۲۴ برنامه)

```
// #define CAMERA_MODEL_AI_THINKER // Has PSRAM
```

- تنظیم کردن SSID و Password مربوط به اکسس پوینت مودم یا هات اسپیات (خط ۳۸ و ۳۹) بورد در حالت STA است

# انجام تغییرات روی پروژه

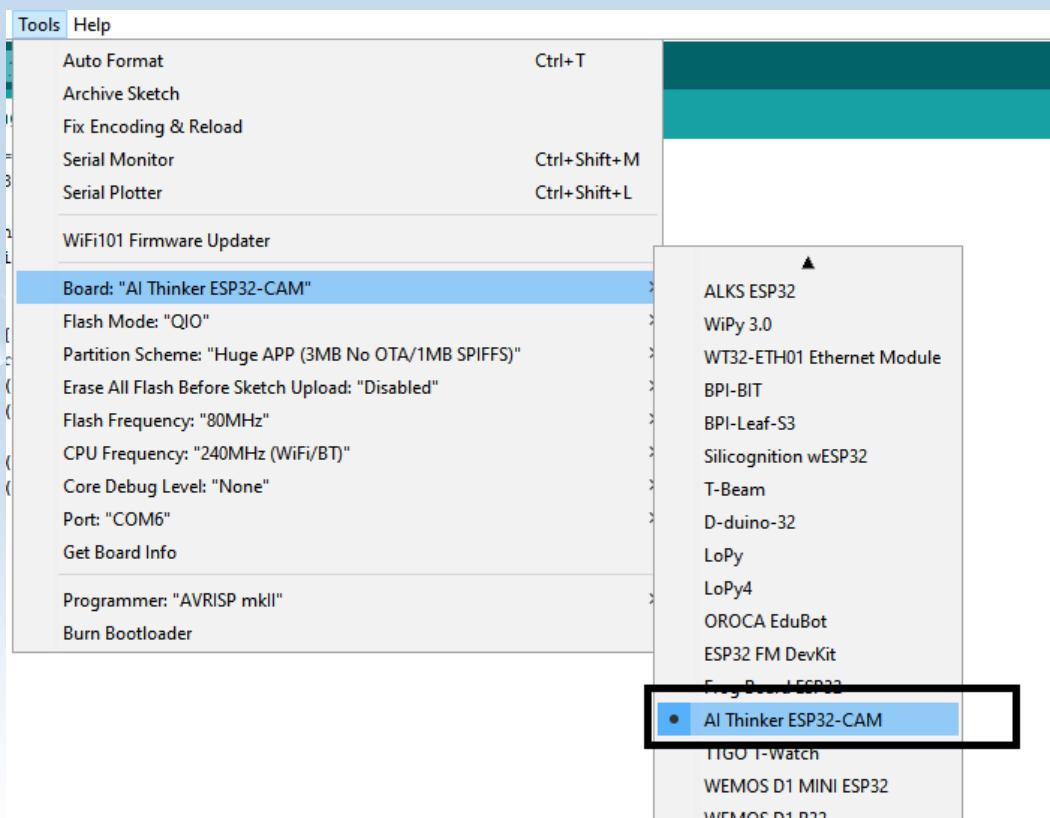
- بعد از انجام سه تغییر زیر، پروژه را حتما ذخیره کنید(در محل دیگری ذخیره خواهد شد)

```
/*
#ifndef CAMERA_MODEL_WROVER_KIT // Has PSRAM
#define CAMERA_MODEL_ESP_EYE // Has PSRAM
#endif
#define CAMERA_MODEL_ESP32S3_EYE // Has PSRAM
#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
#define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
// ** Espressif Internal Boards **
#define CAMERA_MODEL_ESP32_CAM_BOARD
#define CAMERA_MODEL_ESP32S2_CAM_BOARD
#define CAMERA_MODEL_ESP32S3_CAM_LCD
#define CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3 // Has PSRAM
#define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM
#include "camera_pins.h"

// =====
// Enter your WiFi credentials
//
const char* ssid = "*****";
const char* password = "*****";
```

# تنظیم بورد روی AI-Thinker ESP32-CAM کامپایل و آپلود آن روی بورد

- انتخاب بورد AI-Thinker ESP32-CAM
- کامپایل و آپلود برنامه روی بورد



# باز کردن ترمینال و تنظیم سرعت آن روی

## ۱۱۵۲۰۰ و زدن دکمه ریست بورد

- بعد از باز کردن ترمینال و تنظیم سرعت، دکمه ریست که در پشت بورد(سمت چپ و بالا) را بزنید، عبارات زیر در ترمینال

```
ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13964
load:0x40080400,len:3600
entry 0x400805f0

.....
WiFi connected
Camera Ready! Use 'http://172.16.41.4' to connect
```

Autoscroll      Newline      115200 baud      Clear output

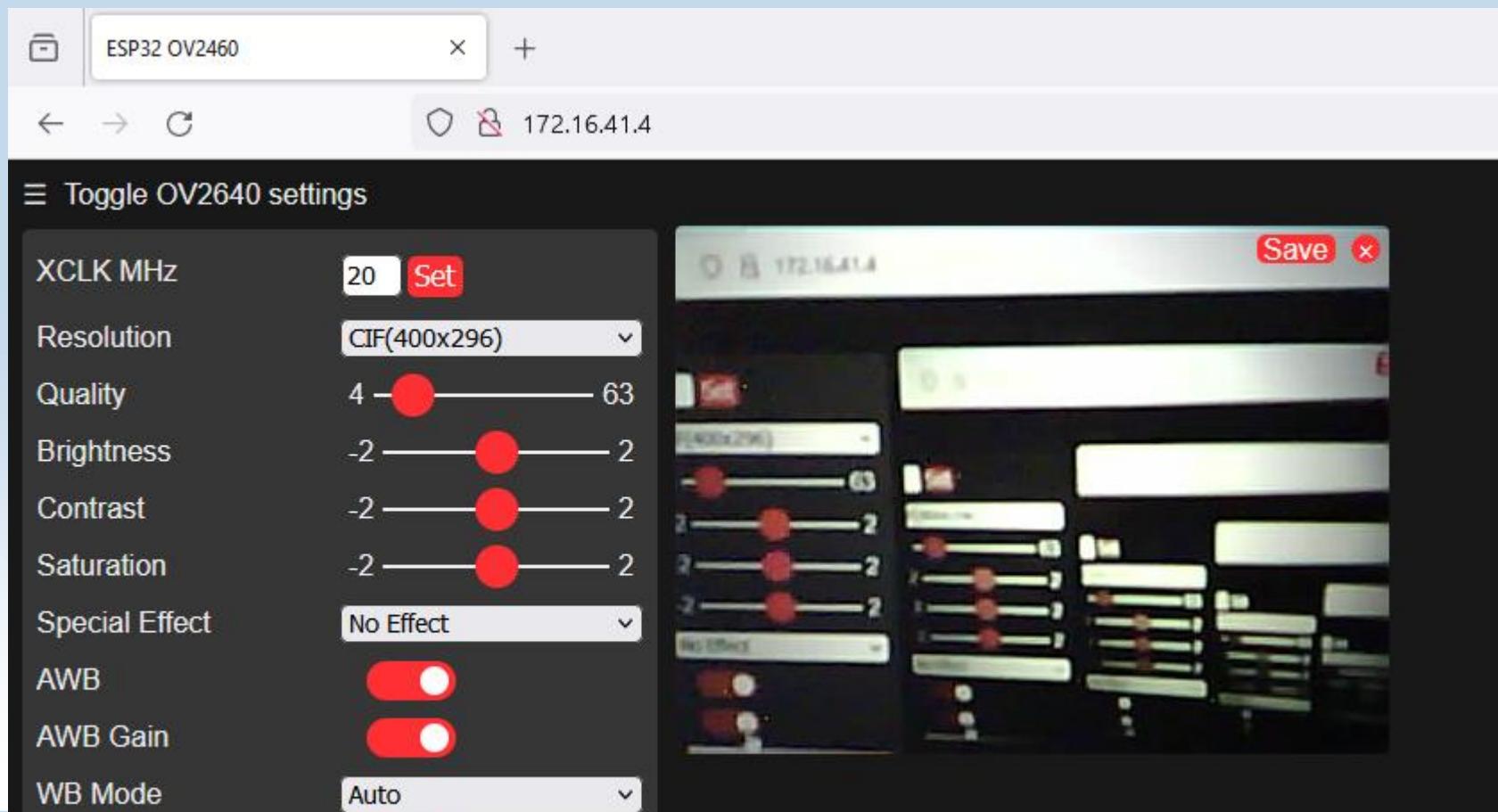
ظاهر می شود



محل دکمه ریست  
پشت بورد

# زدن آدرس IP در مرورگر و مشاهده صفحه و انتخاب گزینه Start Stream

- نکته: لپتاپ یا گوشی هم باید به همان اکسس پویتی متصل باشد که نام و پسورد آن را در برنامه وارد کرده اید





دانشکده مهندسی کامپیوتر

درس برنامه نویسی تجهیزات اینترنت اشیا

نمیسال دوم ۱۴۰۲

روشن و خاموش کردن LED از طریق مرورگر

دکتر علی بهلوانی

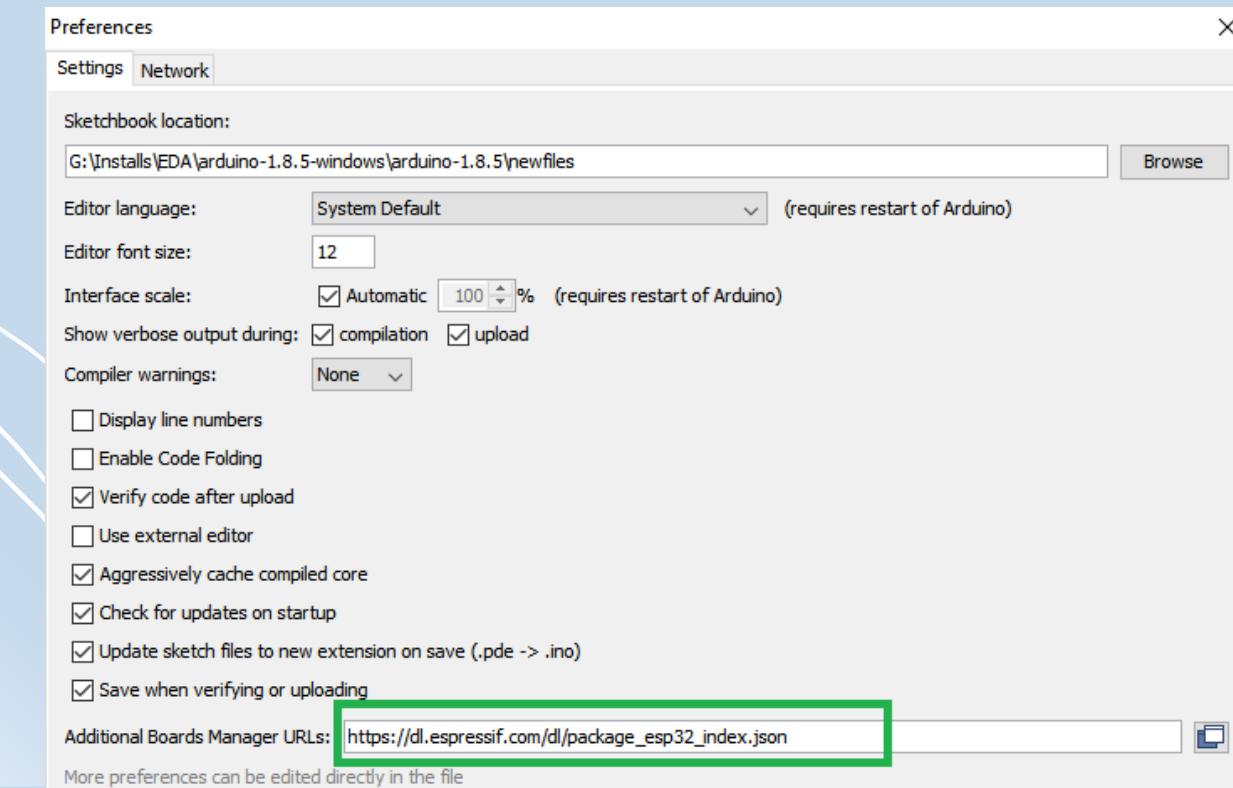


# مراحل کار

- نصب کتابخانه ESP32
- باز کردن یکی از پروژه های آماده مربوط راه اندازی و ب سرور
- انجام تغییرات روی پروژه
- تنظیم بورد روی AI-Thinker ESP32-CAM کامپایل و آپلود آن روی بورد
- باز کردن ترمینال و تنظیم سرعت آن روی 115200
- ریست کردن بورد
- مشاهده IP اختصاص داده شده در ترمینال
- زدن آدرس IP در مرورگر و مشاهده لینکها

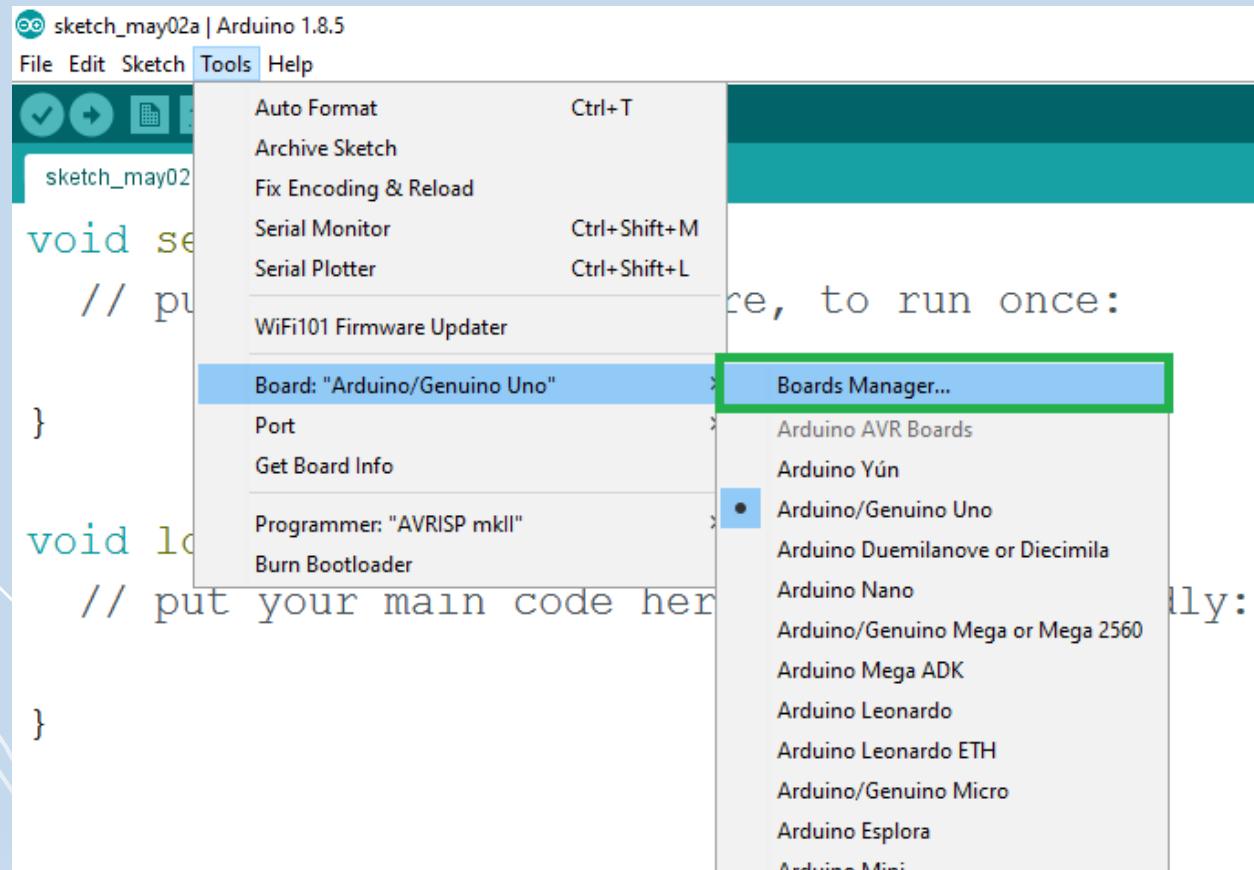
# نصب کتابخانه ESP32

- 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)



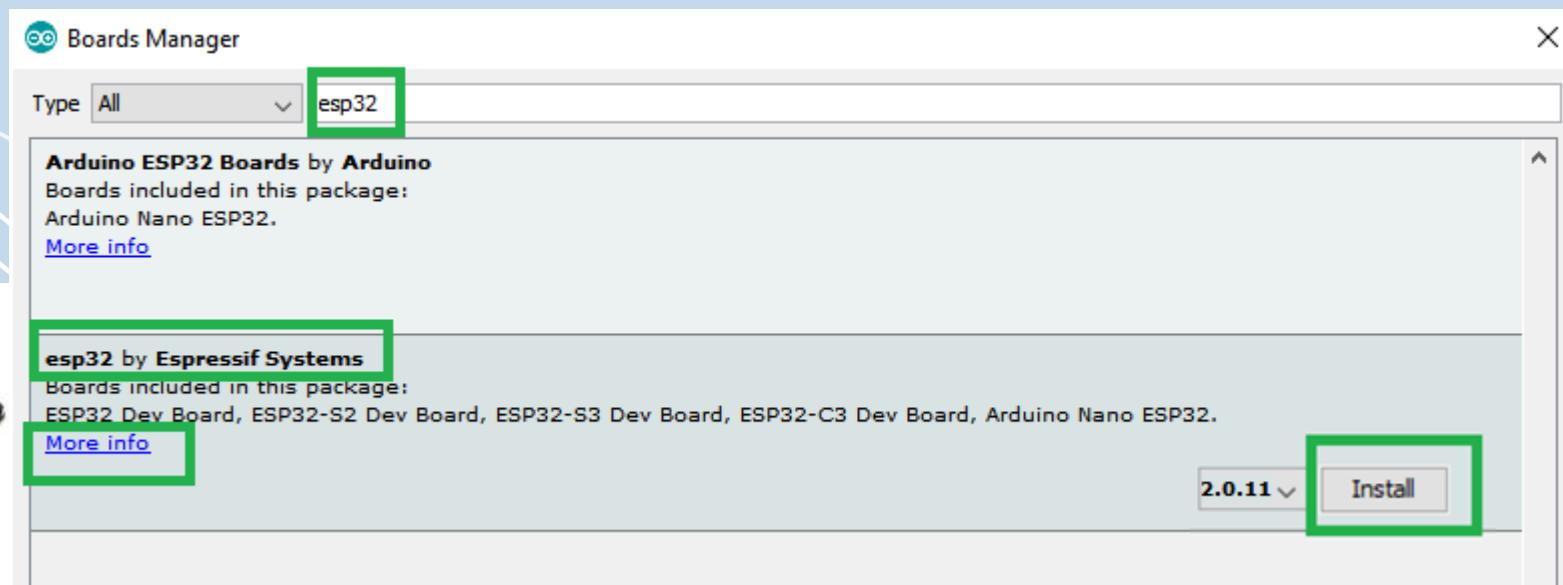
# نصب کتابخانه ESP32

## • انتخاب گزینه Boards Manager



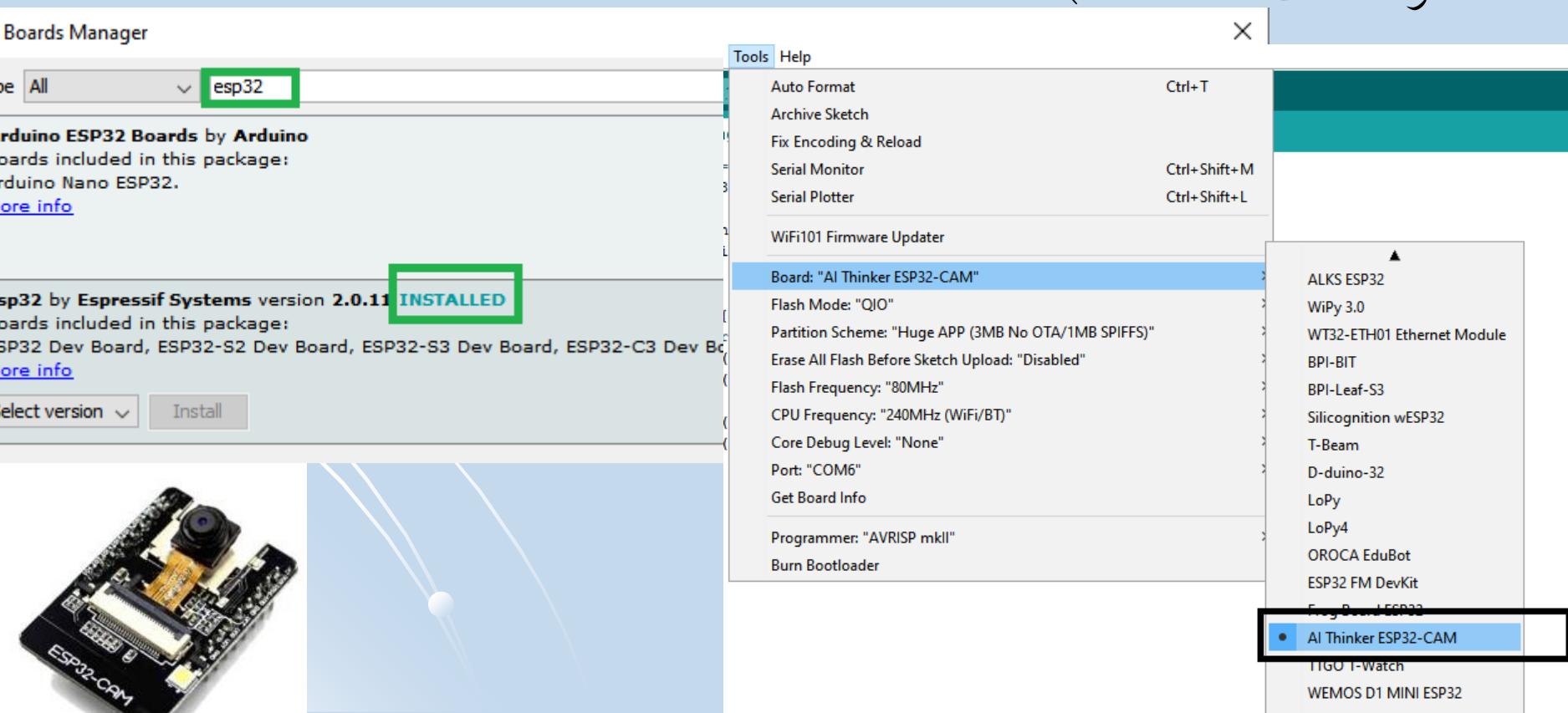
# نصب کتابخانه ESP32

- جستجوی مژول esp32
- در قسمت more Info گزینه Expressif را بزنید
- گزینه Install را انتخاب تا نصب شود

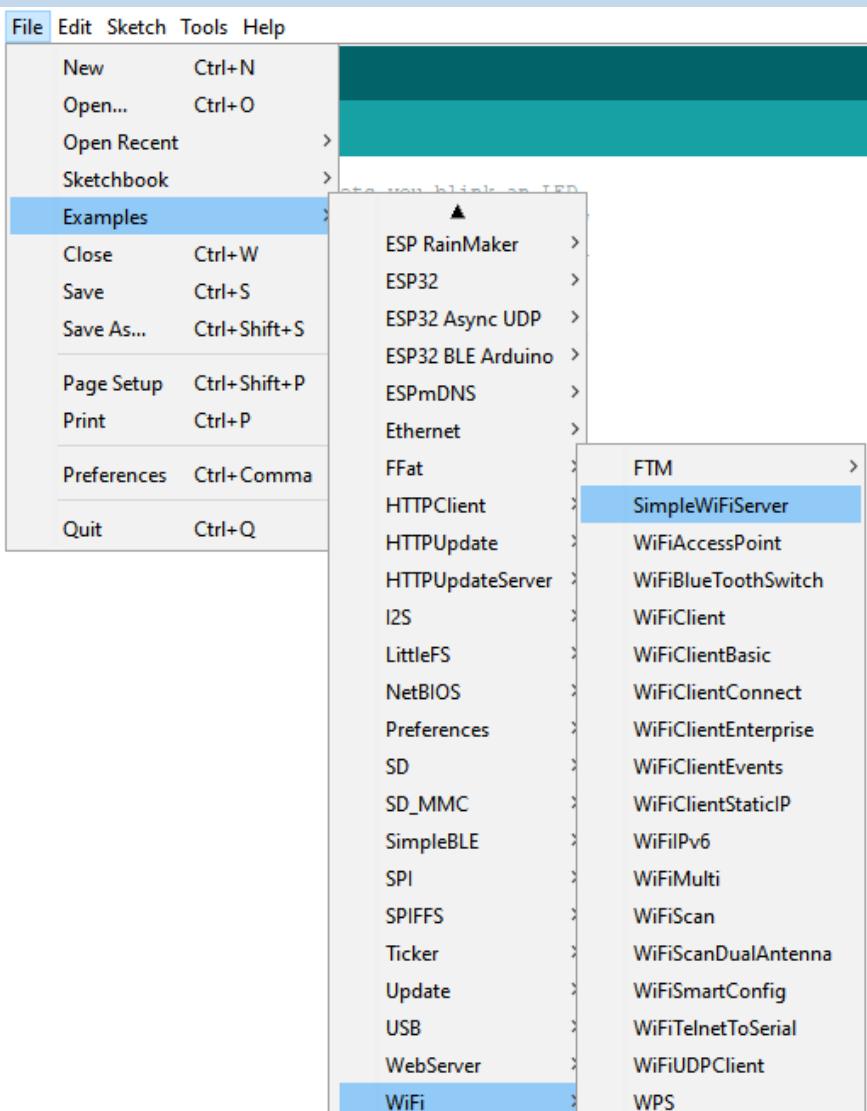


# نصب کتابخانه ESP32

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



# باز کردن پروژه آماده وب سرور ساده



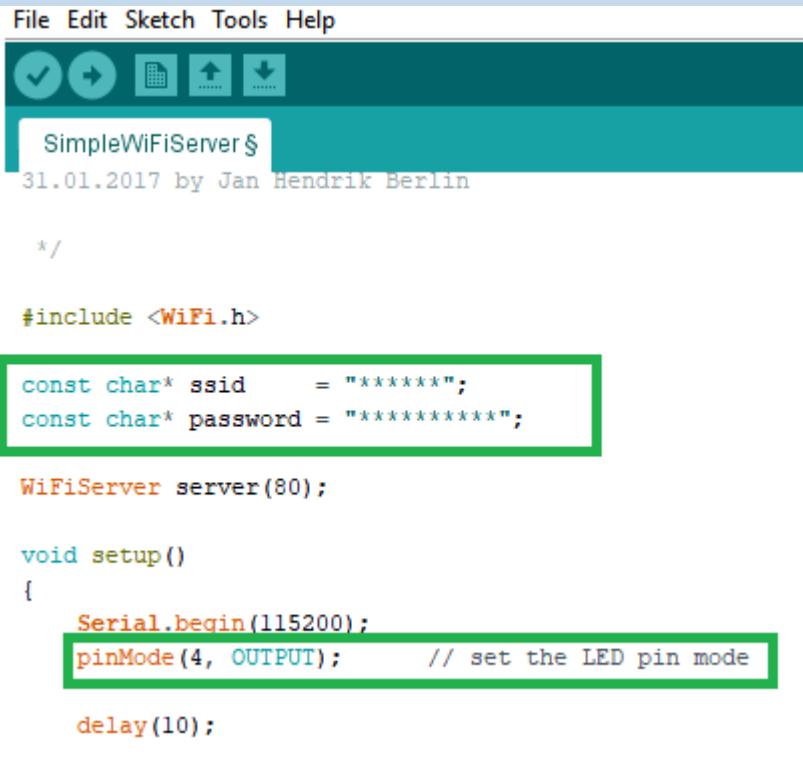
طبق مسیر رو برو، پروژه را باز **SimpleWiFiServer** کنید

## انجام تغییرات روی پروژه

- نام اکسس پوینت و رمز عبور خود را در سطر ۳۰ و ۳۱ وارد کنید.
- اصلاح شماره پایه LED : در این پروژه فرض شده است LED به پایه ۵ متصل شده است، اگر از بورد ESP32-CAM استفاده می کنید. LED به پایه ۴ متصل است و باید در کل برنامه به جای ۵، ۴ جایگزین شود.(سطرهای ۱۰۳، ۳۸ و ۱۳۶)

# انجام تغییرات روی پروژه

- بعد از انجام تغییرات، پروژه را حتما ذخیره کنید(در محل دیگری ذخیره خواهد شد)



The screenshot shows the Arduino IDE interface with the following details:

- File Bar:** File, Edit, Sketch, Tools, Help.
- Toolbar:** Contains icons for Save, Undo, Redo, Open, Upload, and Download.
- Title Bar:** SimpleWiFiServer §
- Information Bar:** 31.01.2017 by Jan Hendrik Berlin
- Code Area:** Displays C++ code for a WiFi server. A green box highlights the configuration section:

```
const char* ssid = "*****";
const char* password = "*****";
```
- Code Area:** Displays C++ code for a WiFi server. A green box highlights the setup section:

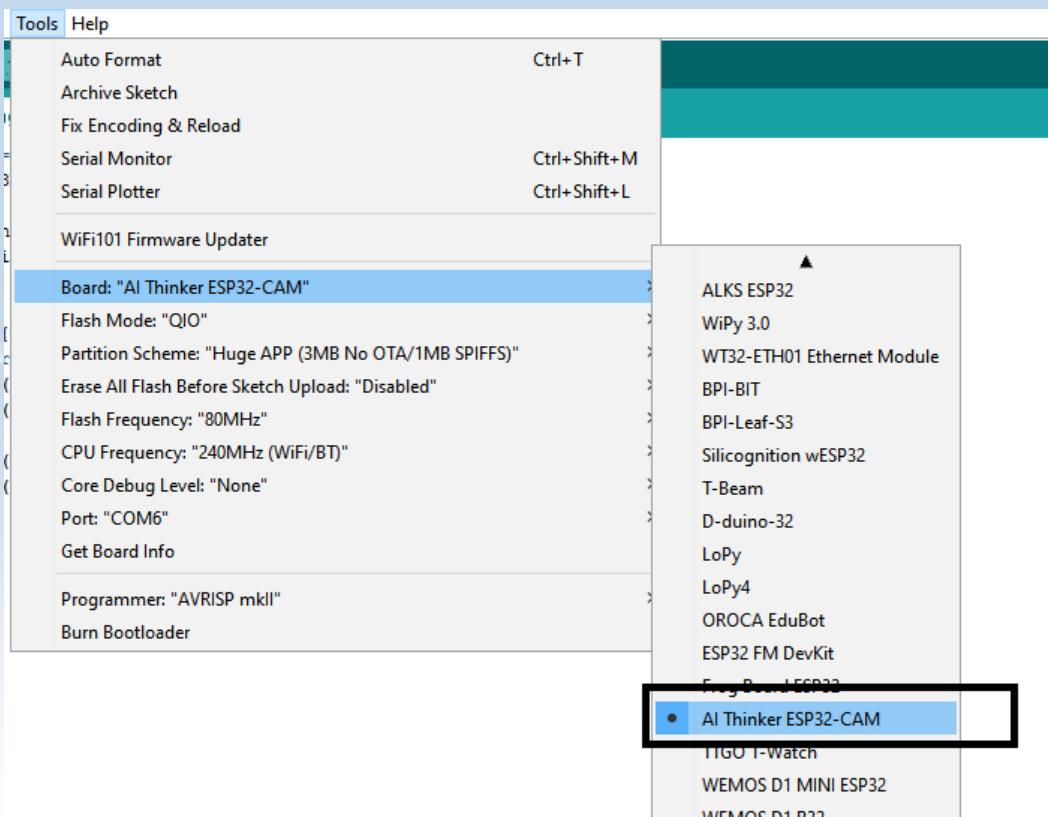
```
WiFiServer server(80);

void setup()
{
 Serial.begin(115200);
 pinMode(4, OUTPUT); // set the LED pin mode
```
- Code Area:** Displays C++ code for a WiFi server. A green box highlights the delay section:

```
delay(10);
```

# تنظیم بورد روی AI-Thinker ESP32-CAM کامپایل و آپلود آن روی بورد

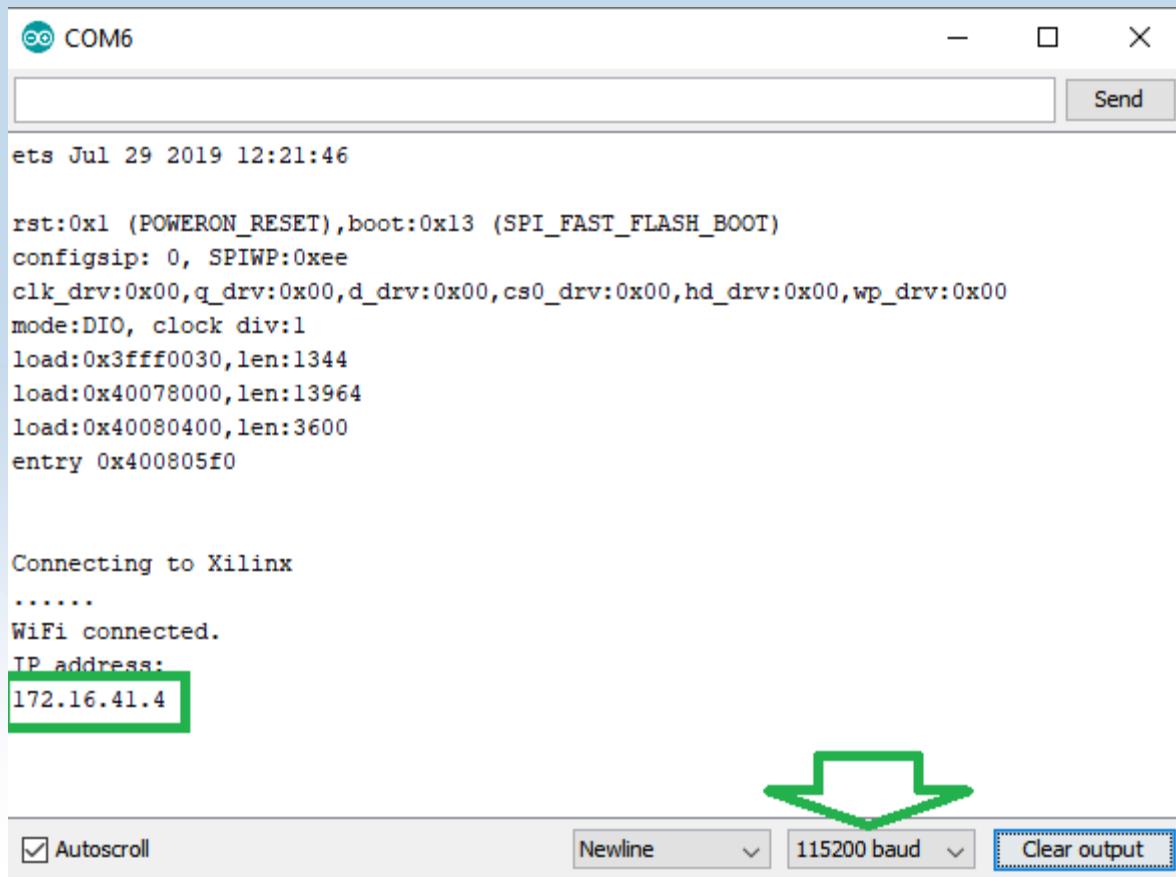
- انتخاب بورد AI-Thinker ESP32-CAM
- کامپایل و آپلود برنامه روی بورد



# باز کردن ترمینال و تنظیم سرعت آن روی

## ۱۱۵۲۰۰ و زدن دکمه ریست بورد

- بعد از باز کردن ترمینال و تنظیم سرعت، دکمه ریست که در پشت بورد(سمت چپ و بالا) را بزنید، عبارات زیر در ترمینال



```
ets Jul 29 2019 12:21:46

rst:0xl (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:1344
load:0x40078000,len:13964
load:0x40080400,len:3600
entry 0x400805f0

Connecting to Xilinx
.....
WiFi connected.
IP address:
172.16.41.4
```

115200 baud

ظاهر می شود



# زدن آدرس IP در مرورگر و مشاهده صفحه

- نکته: لپتاپ یا گوشی هم باید به همان اکسس پویتی متصل باشد که نام و پسورد آن را در برنامه وارد کرده اید

