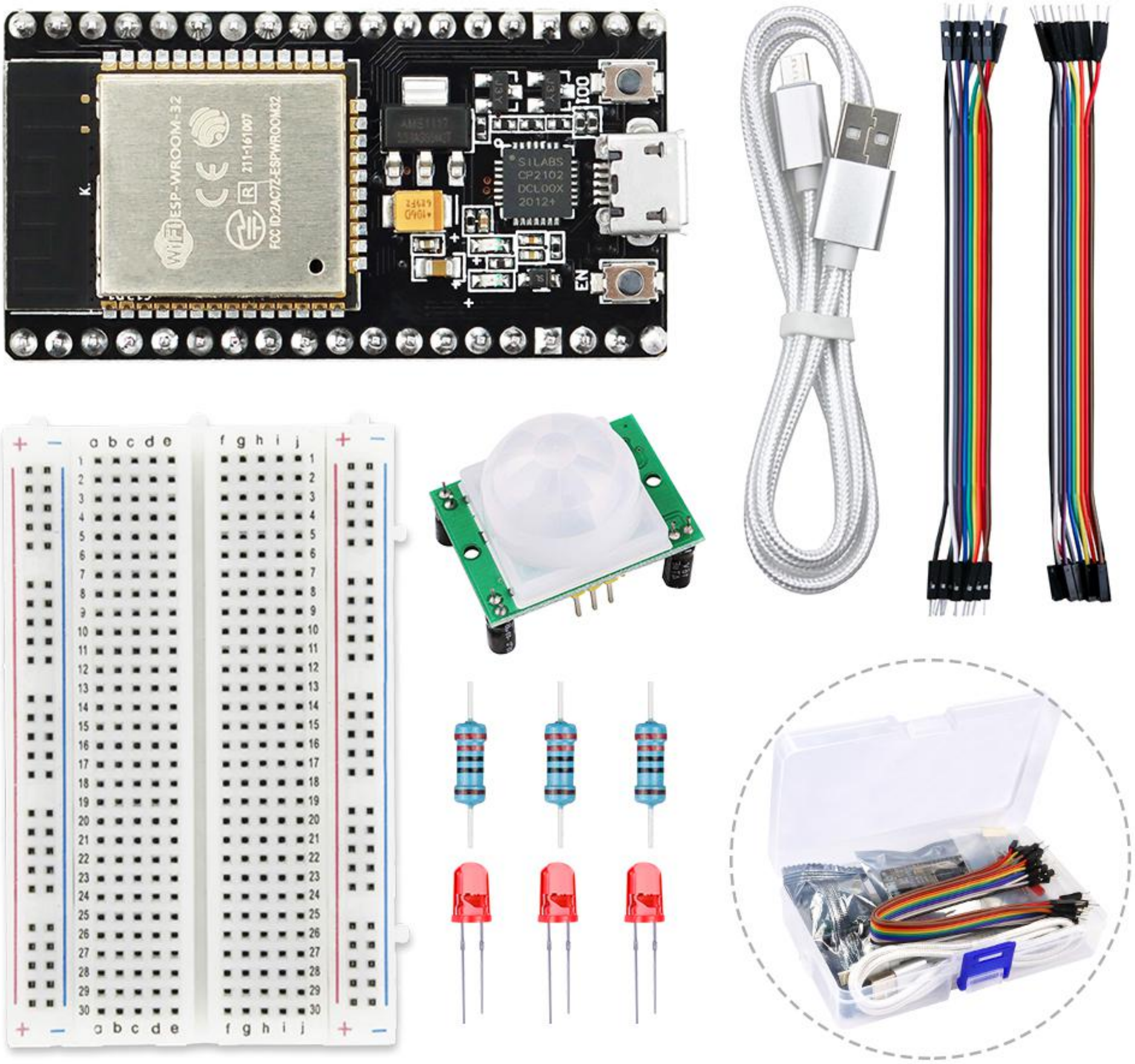
Preface

About Our Company

WayinTop, Your Top Way to Inspiration, is a professional manufacturer over 2,000 open source motherboards, modules, and components. From designing PCBs, printing, soldering, testing, debugging, and offering online tutorials, WayinTop has been committed to explore and demystify the wonderful world of embedded electronics, including but not limited to Arduino and Raspberry Pi. We aim to make the best designed products for makers of all ages and skill levels. No matter your vision or skill level, our products and resources are designed to make electronics more accessible. Founded in 2013, WayinTop has grown to over 100+ employees and a 50,000+ sq ft. factory in China by now. With our unremitting efforts, we also have expanded offerings to include tools, equipments, connector kits and various DIY products that we have carefully selected and tested.

US Amazon Store Homepage:
https://www.amazon.com/shops/A22PZZC3JNHS9L
CA Amazon Store Homepage:
https://www.amazon.ca/shops/A22PZZC3JNHS9L
UK Amazon Store Homepage:
https://www.amazon.co.uk/shops/A3F8F97TMOROPI
DE Amazon Store Homepage:
https://www.amazon.de/shops/A3F8F97TMOROPI
FR Amazon Store Homepage:
https://www.amazon.fr/shops/A3F8F97TMOROPI
IT Amazon Store Homepage:
https://www.amazon.it/shops/A3F8F97TMOROPI
ES Amazon Store Homepage:
https://www.amazon.es/shops/A3F8F97TMOROPI
JP Amazon Store Homepage:
https://www.amazon.co.jp/shops/A1F5OUAXY2TP0K

Based on ESP32 PIR Infrared Motion Sensor Kit



This tutorial uses ESP32 as the main control board and the PIR infrared motion sensor as the detector. When motion is detected (an interrupt is triggered), the LED light will light up and the detection information will be displayed on the Web. You can replace the LED light with a relay to make a practical intelligent control system.

Pin Connection

ESP32<-------->PIR Motion Sensor

3.3V<------->VCC

GND<--------->GND

G13<--------->OUT

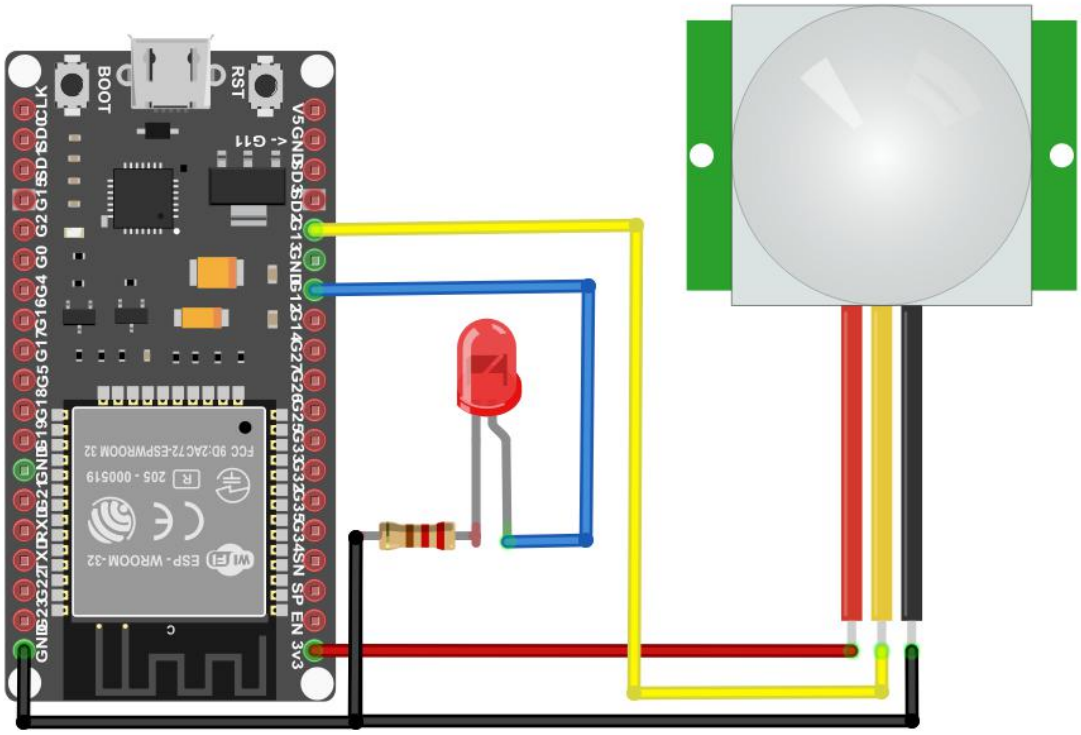
ESP32<-------->LED

G12<--------->LED Anode

LED Negative<--------->Resistance

The other end of the Resistance<--------->GND

Schematic Diagram



Environment Setup

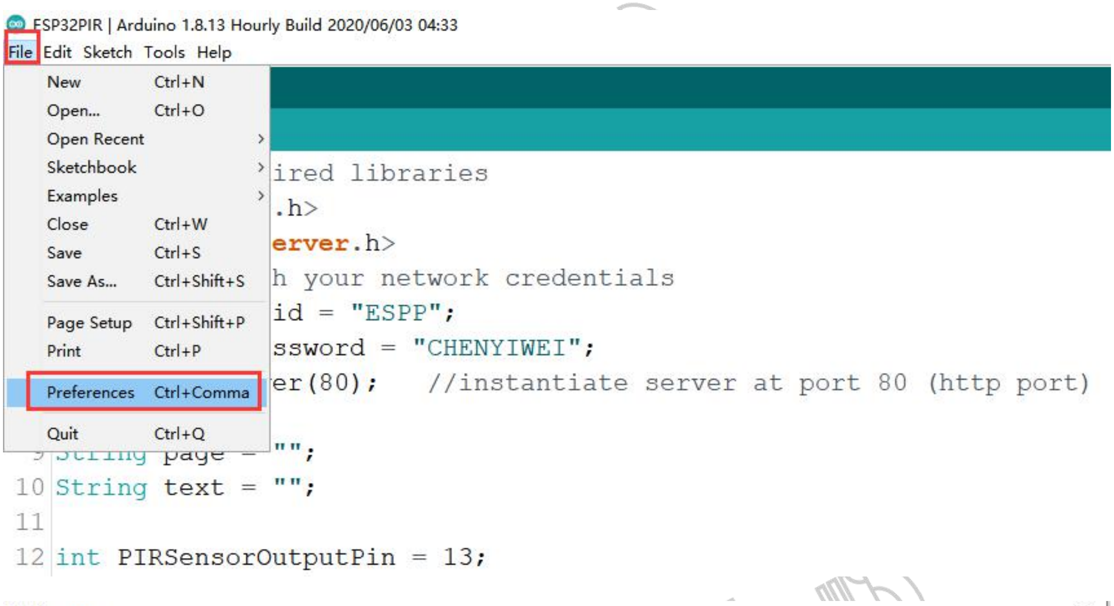
Step 1: Download arduino IDE

Windows：

https://www.arduino.cc/en/Main/Software

Step 2: Add ESP32 development board

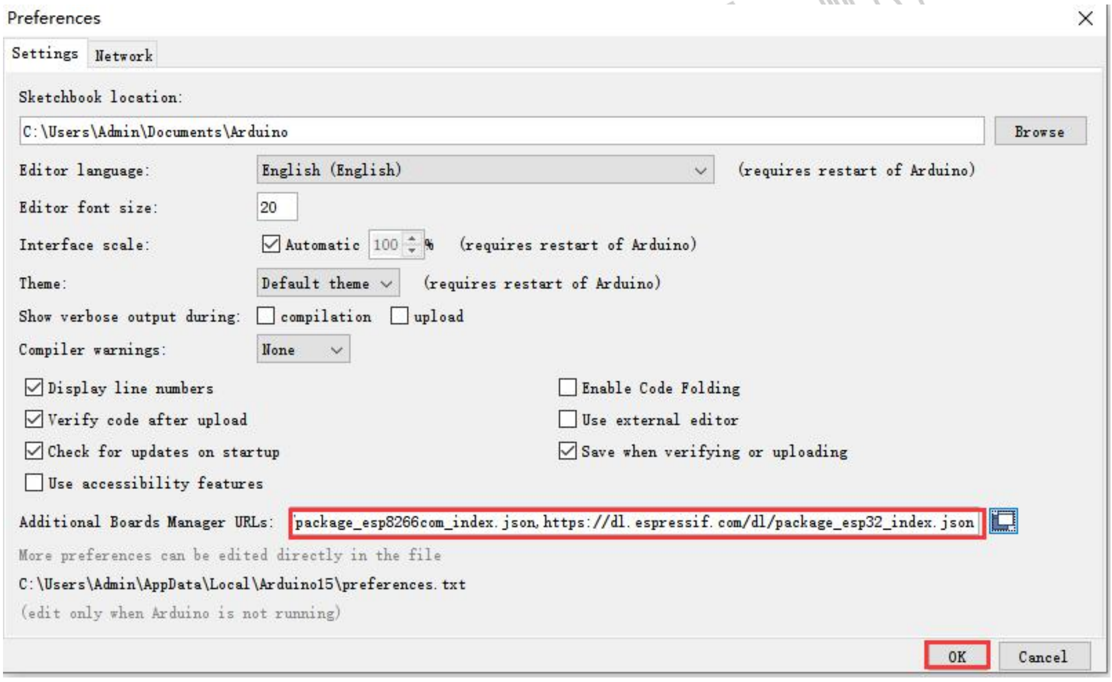
Open the arduino IDE, click **file->Preferences** in turn, and enter in the window **Additional Boards Manager URLs**: https://dl.espressif.com/dl/package_esp32_index.json, and then click OK, as shown in the figure below.



```
ired libraries
.h>
erver.h>
h your network credentials
id = "ESPP";
ssword = "CHENYIWEI";
er(80);    //instantiate server at port 80 (http port)

String page = "";
String text = "";

int PIRSensorOutputPin = 13;
```



Sketchbook location: C:\Users\Admin\Documents\Arduino

Editor language: English (English) (requires restart of Arduino)

Editor font size: 20

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

Compiler warnings: None

☑ Display line numbers

☑ Verify code after upload

☑ Check for updates on startup

□ Use accessibility features

□ Enable Code Folding

□ Use external editor

☑ Save when verifying or uploading

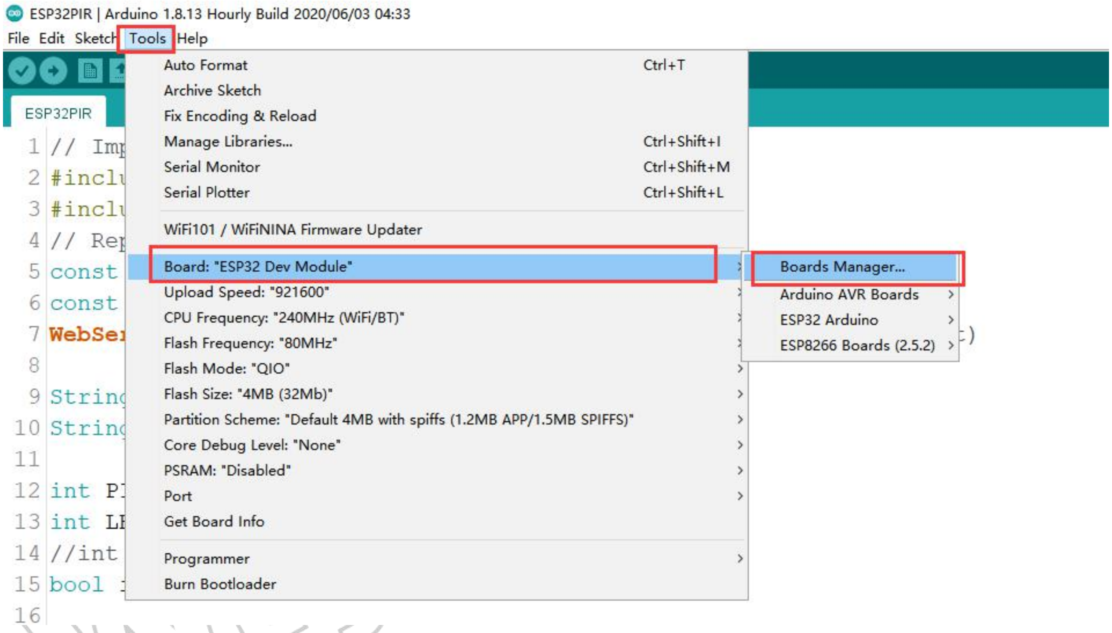
Additional Boards Manager URLs: package_esp8266com_index.json,https://dl.espressif.com/dl/package_esp32_index.json

More preferences can be edited directly in the file

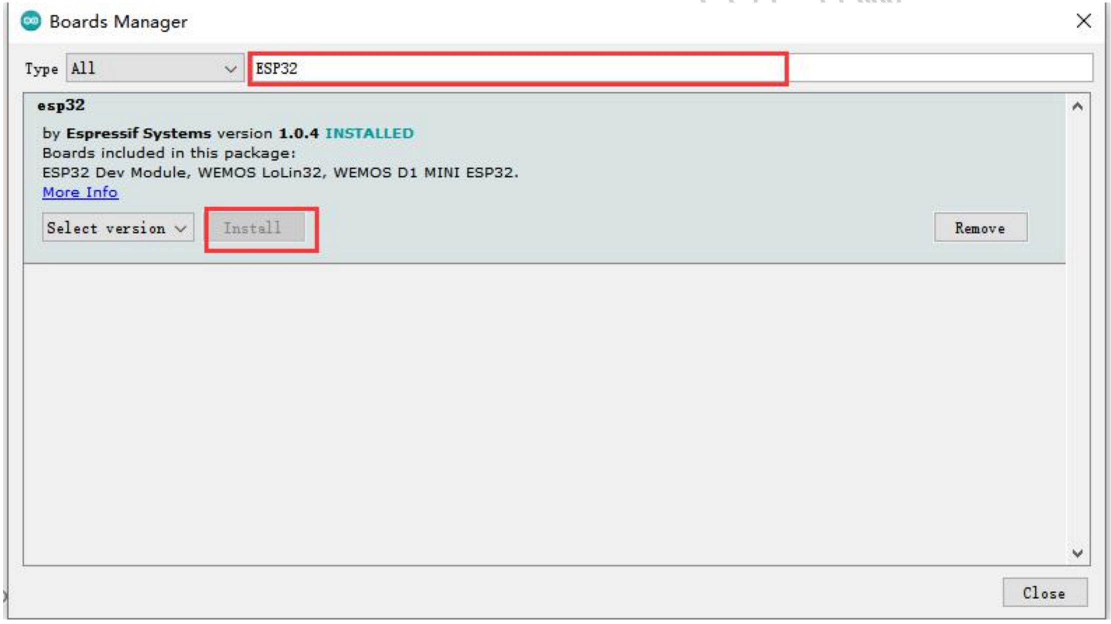
C:\Users\Admin\AppData\Local\Arduino15\preferences.txt

(edit only when Arduino is not running)

Step 3: Click **Tools->Board:->Boards Manager...**, as shown in the figure below:


ESP32PIR | Arduino 1.8.13 Hourly Build 2020/06/03 04:33
File Edit Sketch Tools Help
ESP32PIR
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
Board: "ESP32 Dev Module"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
Core Debug Level: "None"
PSRAM: "Disabled"
Port
Get Board Info
Programmer
Burn Bootloader
Boards Manager...
Arduino AVR Boards
ESP32 Arduino
ESP8266 Boards (2.5.2)

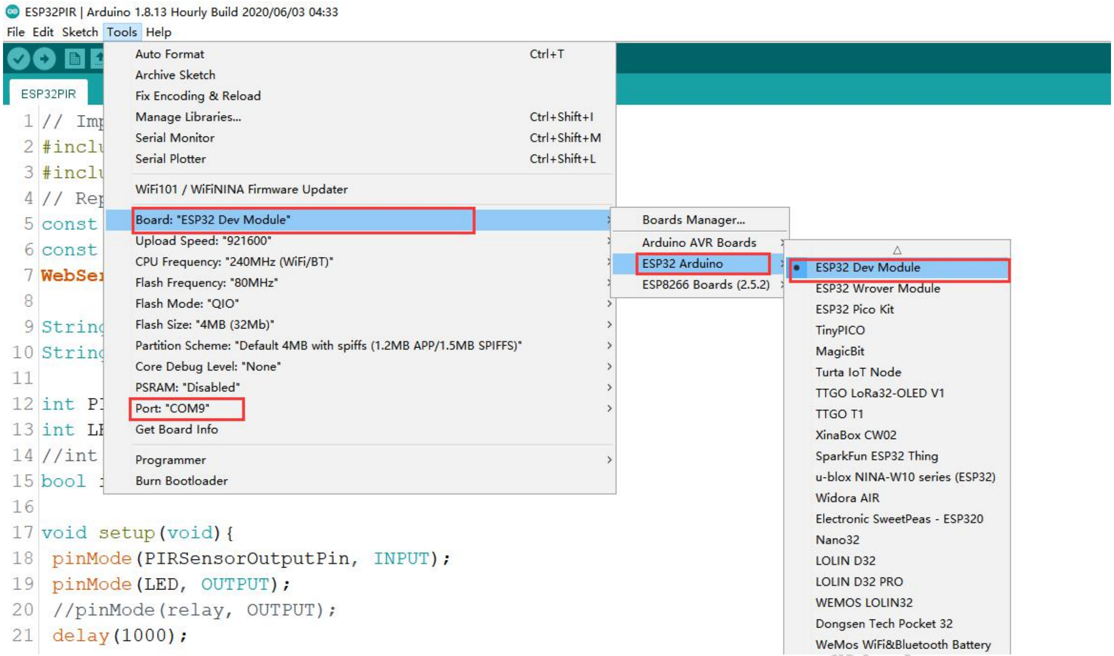
Step 4: Enter ESP32 in the pop-up window, and then install the ESP32 development board library that appears below.



esp32

by **Espressif Systems** version **1.0.4** INSTALLED
Boards included in this package:
ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32.
More Info

Step 5: After the installation is complete, connect the connected ESP32 module to the computer, and select the correct ESP32 module and port in the Tools menu, as shown in the figure below (the port number is subject to the identification of the computer).



ESP32PIR | Arduino 1.8.13 Hourly Build 2020/06/03 04:33

Board: "ESP32 Dev Module"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
Core Debug Level: "None"
PSRAM: "Disabled"
Port: "COM9"

```
void setup(void){
 pinMode(PIRSensorOutputPin, INPUT);
 pinMode(LED, OUTPUT);
 //pinMode(relay, OUTPUT);
 delay(1000);
```

Step 6: Create a new program file, copy the following code into the program file (you can also directly open the code file we provide)

// Import required libraries

#include <WiFi.h>

#include <WebServer.h>

```
// Replace with your network credentials

const char* ssid = "********";

const char* password = "**********";

WebServer server(80);    //instantiate server at port 80 (http port)


String page = "";

String text = "";


int PIRSensorOutputPin = 13;

int LED = 12;

//int relay=14;

bool isTriggered = false;


void setup(void){

  pinMode(PIRSensorOutputPin, INPUT);

  pinMode(LED, OUTPUT);

  //pinMode(relay, OUTPUT);

  delay(1000);

  Serial.begin(115200);

  WiFi.begin(ssid, password); //begin WiFi connection

  Serial.println("");
```

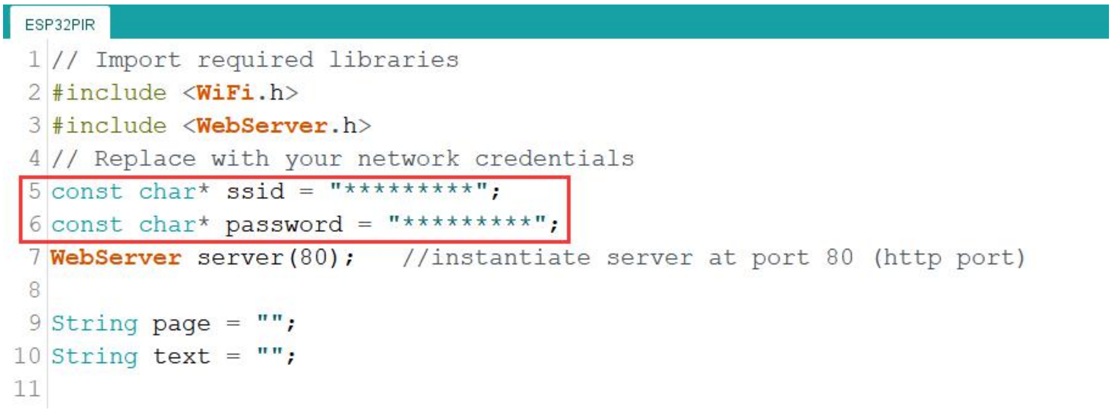
```
 // Wait for connection
 while (WiFi.status() != WL_CONNECTED) {
 delay(500);
 Serial.print(".");
}

 Serial.println("");
 Serial.print("Connected to ");
 Serial.println(ssid);
 Serial.print("IP address: ");
 Serial.println(WiFi.localIP());
 server.on("/data.txt", [](){
   if(isTriggered){
     text = "Person Detected!";
     digitalWrite(LED, HIGH);
   }else{
     text = "";
     digitalWrite(LED, LOW);
   }
   server.send(200, "text/html", text);
 });
 server.on("/", [](){
   page = "<h1>PIR Sensor to NodeMCU/h1><h1>Data:</h1> <h1 id=\"data\">""</h1>\r\n";
   page += "<script>\r\n";
   page += "var x = setInterval(function() {loadData(\"data.txt\",updateData)}, 1000);\r\n";
   page += "function loadData(url, callback){\r\n";
   page += "var xhttp = new XMLHttpRequest();\r\n";
   page += "xhttp.onreadystatechange = function(){\r\n";
   page += " if(this.readyState == 4 && this.status == 200){\r\n";
   page += " callback.apply(xhttp);\r\n";
   page += " }\r\n";
   page += "};\r\n";
   page += "xhttp.open(\"GET\", url, true);\r\n";
   page += "xhttp.send();\r\n";
   page += "}\r\n";
   page += "function updateData(){\r\n";
   page += " document.getElementById(\"data\").innerHTML = this.responseText;\r\n";
   page += "}\r\n";
```

```
      page += "</script>\r\n";
      server.send(200, "text/html", page);
});

  server.begin();
  Serial.println("Web server started!");
}
void loop(void){
    long state=digitalRead(PIRSensorOutputPin);
  if(digitalRead(PIRSensorOutputPin) == HIGH){
        isTriggered = true;
  }else{
        isTriggered = false;
  }
  server.handleClient();
}
```

Step 7: Change the WIFI name and password in the program to your own WIFI.



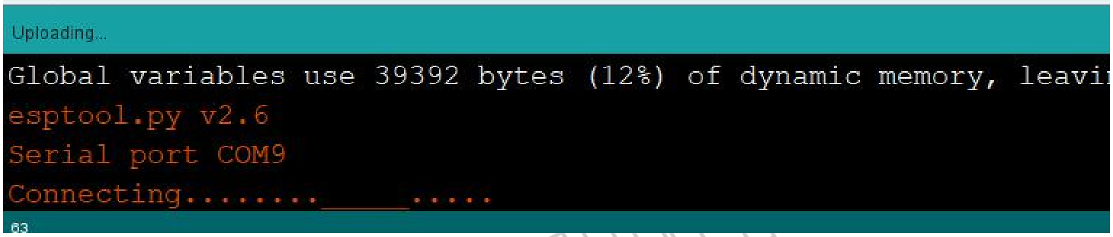
ESP32PIR

```
// Import required libraries
#include <WiFi.h>
#include <WebServer.h>
// Replace with your network credentials
const char* ssid = "*********";
const char* password = "*********";
WebServer server(80);   //instantiate server at port 80 (http port)

String page = "";
String text = "";

```

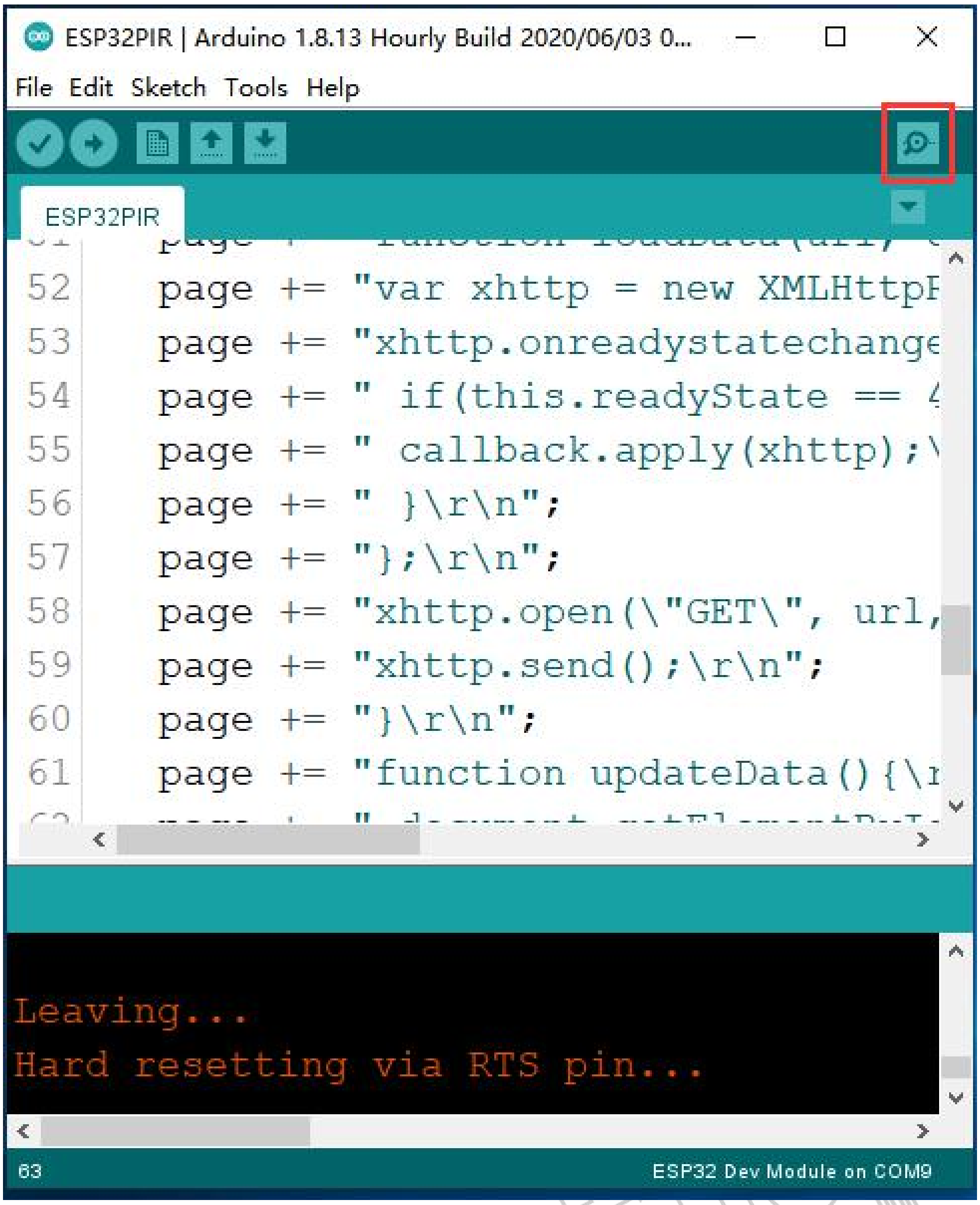
Step 8: Upload the code to ESP32, when the waiting symbol "......____......." appears in the arduinoIDE during the upload process (as shown in the figure below), please press the IO0 button on the ESP32 module for about 1 second, so that the code can be uploaded successfully.



Uploading...

```
Global variables use 39392 bytes (12%) of dynamic memory, leavi
esptool.py v2.6
Serial port COM9
Connecting........_____.....
```

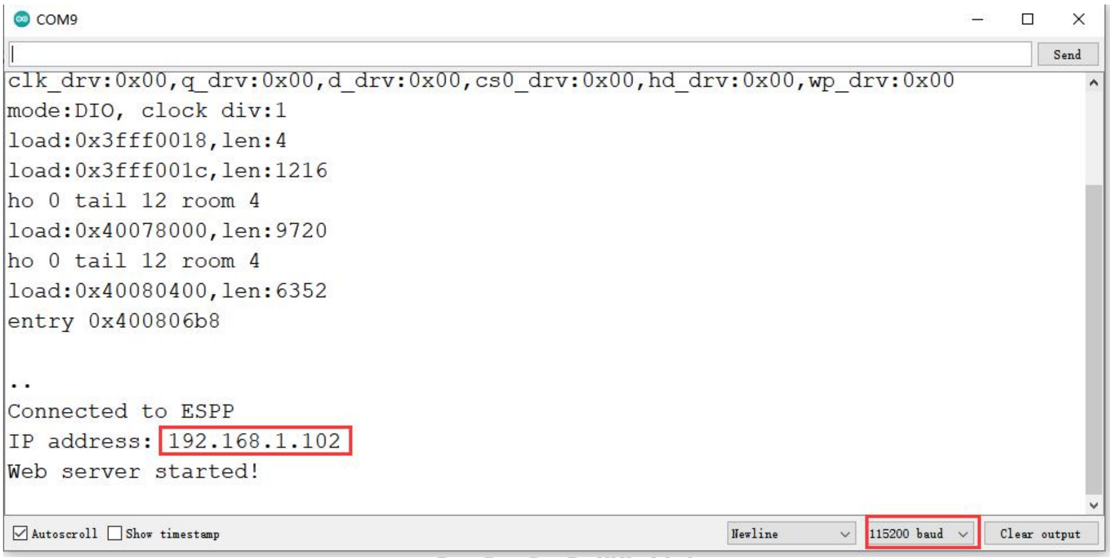
Step 9: After the code is uploaded successfully, click the button in the upper right corner of arduinoIDE to open the serial monitor, as shown in the figure below.


ESP32PIR | Arduino 1.8.13 Hourly Build 2020/06/03 0...
File Edit Sketch Tools Help
ESP32PIR
63
ESP32 Dev Module on COM9

```
    page += "var xhttp = new XMLHttpR
    page += "xhttp.onreadystatechange
    page += "  if(this.readyState == 4
    page += "  callback.apply(xhttp);\
    page += "  }\r\n";
    page += "};\r\n";
    page += "xhttp.open(\"GET\", url,
    page += "xhttp.send();\r\n";
    page += "}\r\n";
    page += "function updateData(){\r
```

```
Leaving...
Hard resetting via RTS pin...
```

Step 10: As shown in the figure below, set the serial port baud rate to **115200**, and then press the EN button on the ESP32 module. If the WIFI setting is correct, the IP address will be printed out, as shown in the figure below.

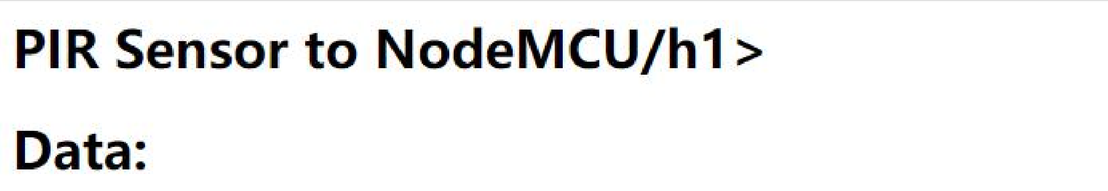


```
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:9720
ho 0 tail 12 room 4
load:0x40080400,len:6352
entry 0x400806b8

..
Connected to ESPP
IP address: 192.168.1.102
Web server started!
```

☑ Autoscroll □ Show timestamp Newline 115200 baud Clear output

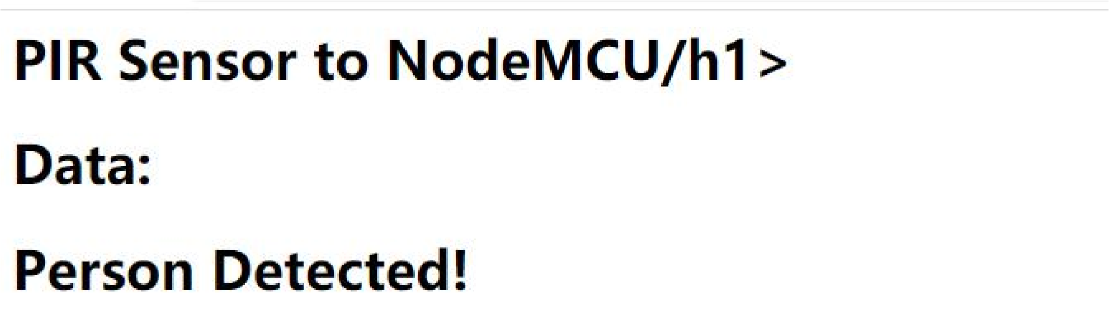
Step 11: Copy the IP address to the browser and open it (the computer must be in the same local area network as the wifi), and the browser will display the page as shown in the figure below



PIR Sensor to NodeMCU/h1>

Data:

Step 12: When the sensor detects human movement, the LED light will light up, and the web page will show the prompt message as shown in the figure below.



PIR Sensor to NodeMCU/h1>

Data:

Person Detected!

Warm Tips: When the two potentiometer of the PIR sensor are rotated counterclockwise to the maximum and the jumper cap is connected to H, the project achieves the best effect.