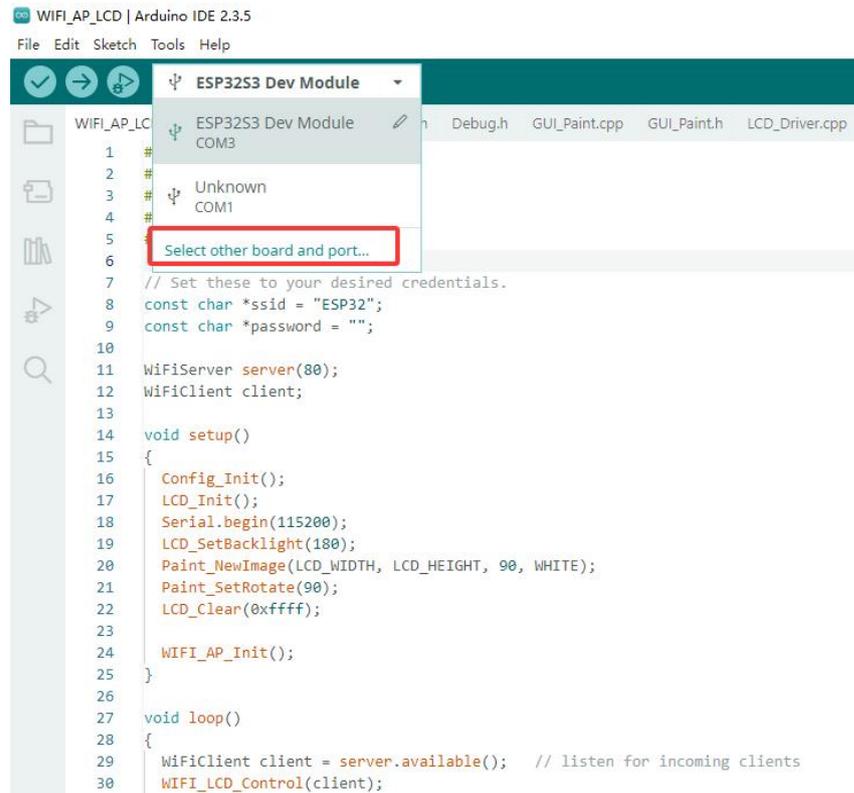


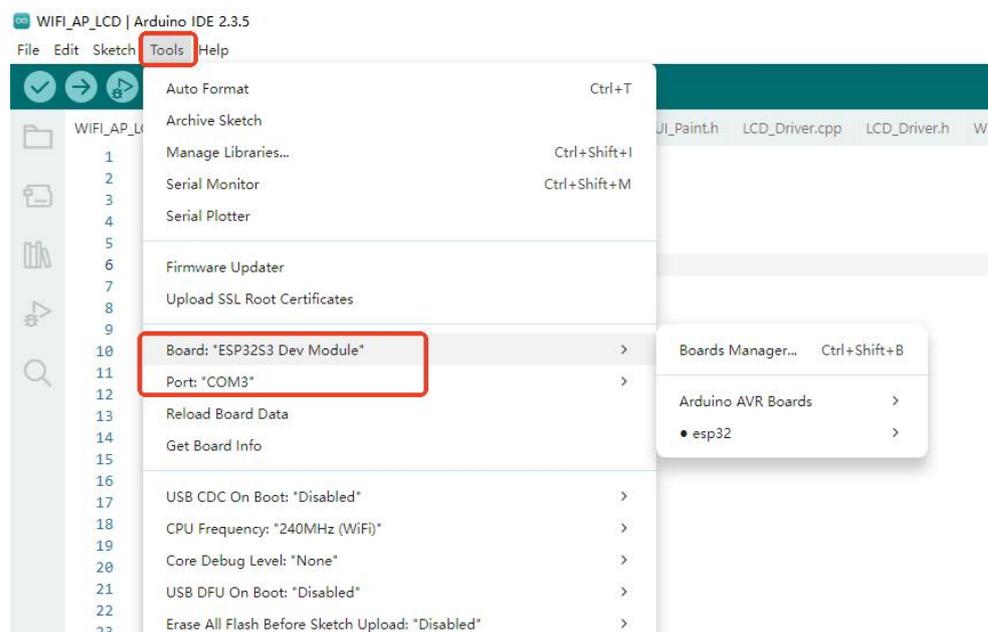
Arduino IDE 2.3.5 + ESP32-S3-CAM + OV2640 Camera

After setting up the Arduino IDE development environment and installing the serial port driver CH343, perform the following steps:

1. Development board selection: “ESP32S3 Dev Module”
2. Port number selection: COM3 (each computer port number is different, choose by yourself)



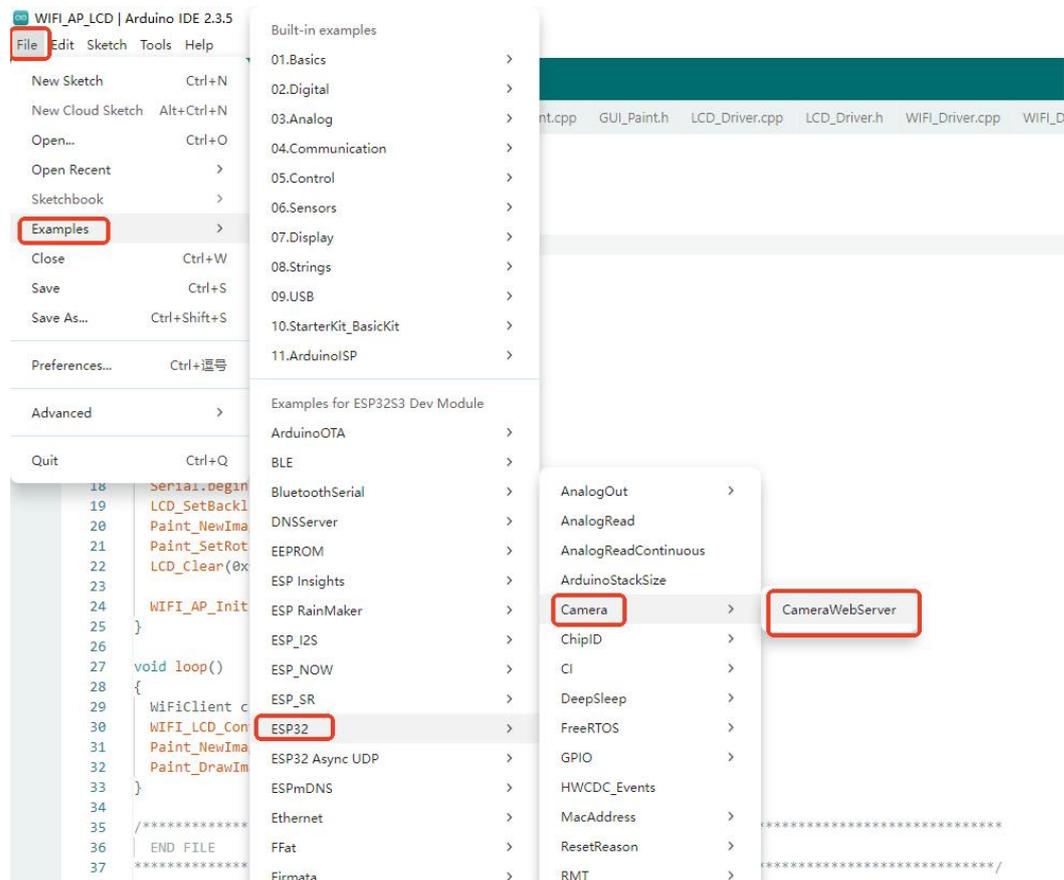
Alternatively, select the development board and port number in the Tools field:



3. Test Example

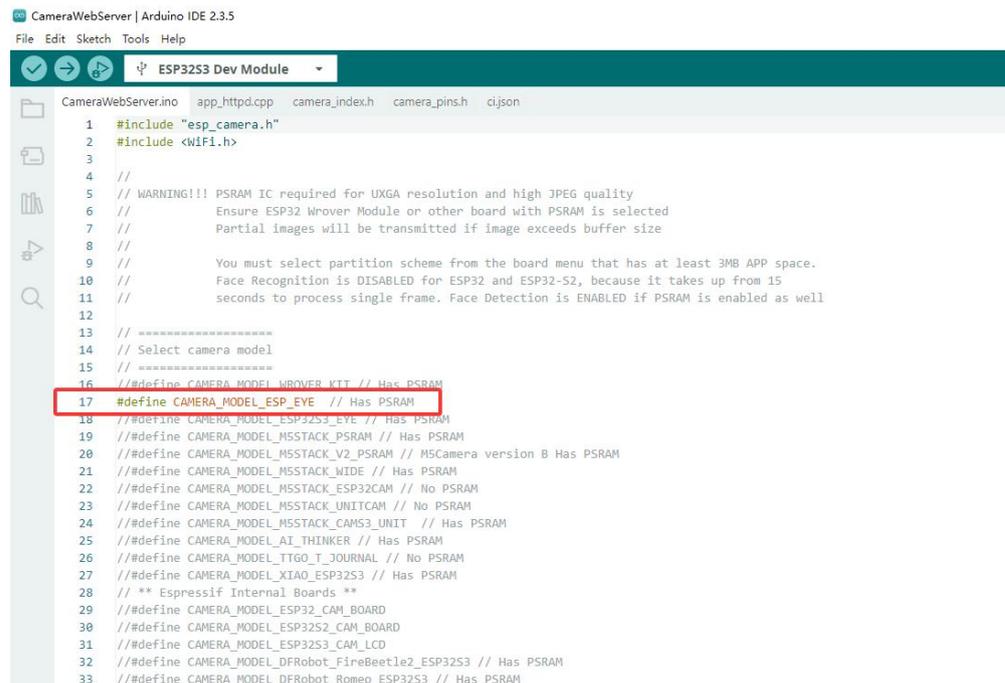
Use the esp32-cam example that comes with the Arduino IDE for testing:

(1) Select File->Examples->ESP32->Camera->CameraWebServer

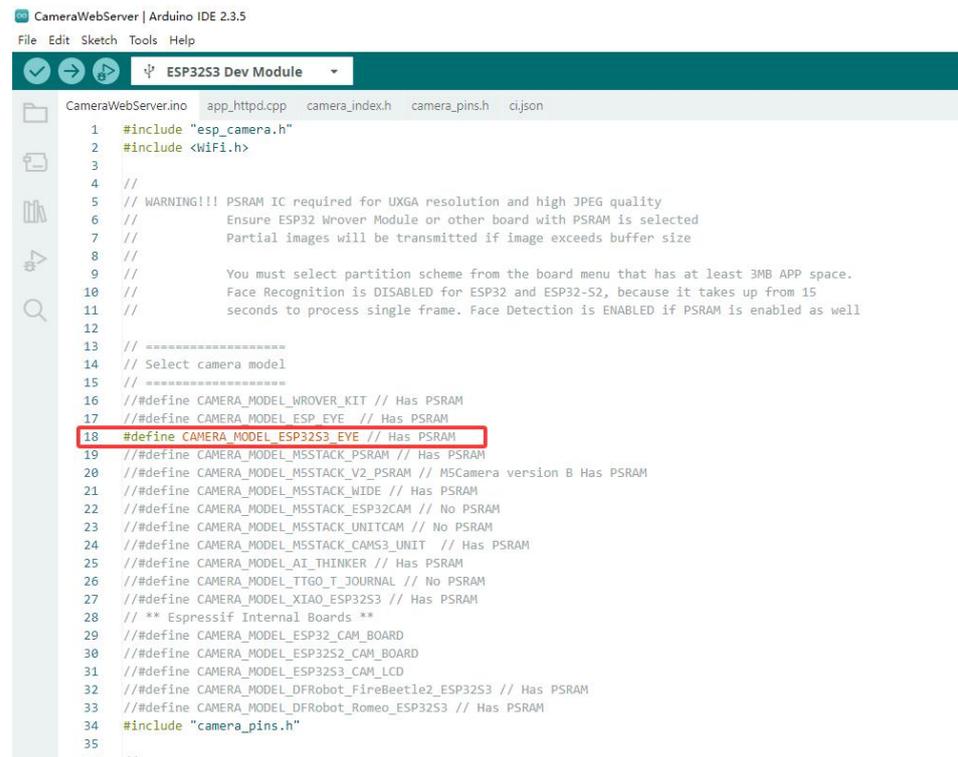


(2) Select development board model,

Before modification:



After modification, select: CAMERA_MODEL_ESP32S3_EYE



```
CameraWebServer.ino  app_httpd.cpp  camera_index.h  camera_pins.h  cijson
1  #include "esp_camera.h"
2  #include <WiFi.h>
3
4  //
5  // WARNING!!! PSRAM IC required for UXGA resolution and high JPEG quality
6  // Ensure ESP32 Wrover Module or other board with PSRAM is selected
7  // Partial images will be transmitted if image exceeds buffer size
8  //
9  // You must select partition scheme from the board menu that has at least 3MB APP space.
10 // Face Recognition is DISABLED for ESP32 and ESP32-S2, because it takes up from 15
11 // seconds to process single frame. Face Detection is ENABLED if PSRAM is enabled as well
12
13 // =====
14 // Select camera model
15 // =====
16 // #define CAMERA_MODEL_WROVER_KIT // Has PSRAM
17 // #define CAMERA_MODEL_ESP_EYE // Has PSRAM
18 #define CAMERA_MODEL_ESP32S3_EYE // Has PSRAM
19 // #define CAMERA_MODEL_MSSTACK_PSRAM // Has PSRAM
20 // #define CAMERA_MODEL_MSSTACK_V2_PSRAM // M5Camera version B Has PSRAM
21 // #define CAMERA_MODEL_MSSTACK_WIDE // Has PSRAM
22 // #define CAMERA_MODEL_MSSTACK_ESP32CAM // No PSRAM
23 // #define CAMERA_MODEL_MSSTACK_UNITCAM // No PSRAM
24 // #define CAMERA_MODEL_MSSTACK_CAMS3_UNIT // Has PSRAM
25 // #define CAMERA_MODEL_AI_THINKER // Has PSRAM
26 // #define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
27 // #define CAMERA_MODEL_XIAO_ESP32S3 // Has PSRAM
28 // ** Espressif Internal Boards **
29 // #define CAMERA_MODEL_ESP32_CAM_BOARD
30 // #define CAMERA_MODEL_ESP32S2_CAM_BOARD
31 // #define CAMERA_MODEL_ESP32S3_CAM_LCD
32 // #define CAMERA_MODEL_DFRobot_FireBeetle2_ESP32S3 // Has PSRAM
33 // #define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM
34 #include "camera_pins.h"
35
36 //
```

(3) Add the AP's Wi-Fi account number and password

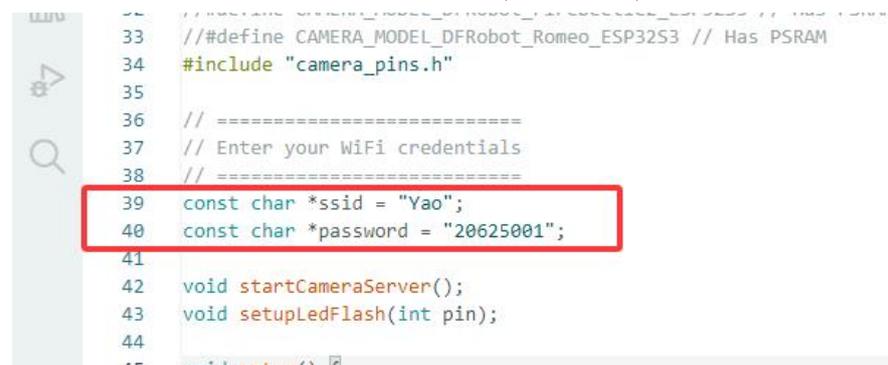
Computer or cell phone open hotspot as AP, the account password is the hotspot account and password of the computer or cell phone as hotspot. **Note: The connected WiFi needs to be in the 2.4GHz band.**

Before modification:



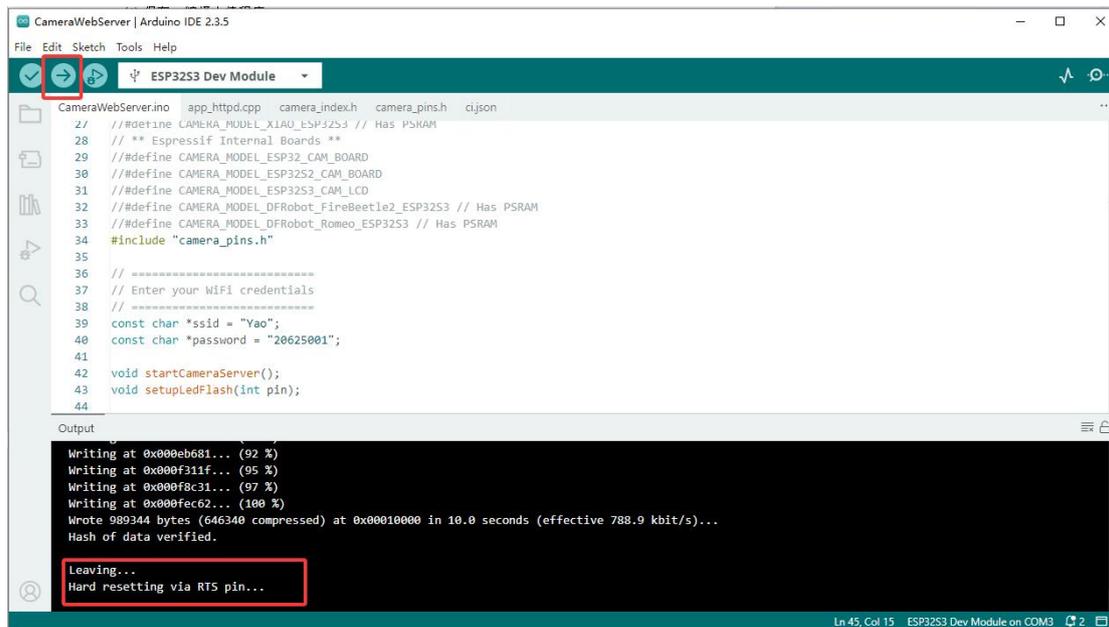
```
33 // #define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM
34 #include "camera_pins.h"
35
36 // =====
37 // Enter your WiFi credentials
38 // =====
39 const char *ssid = "*****";
40 const char *password = "*****";
41
42 void startCameraServer();
43 void setupLedFlash(int pin);
44
45 void setup() {
```

After modification: WiFi to be connected (customized)

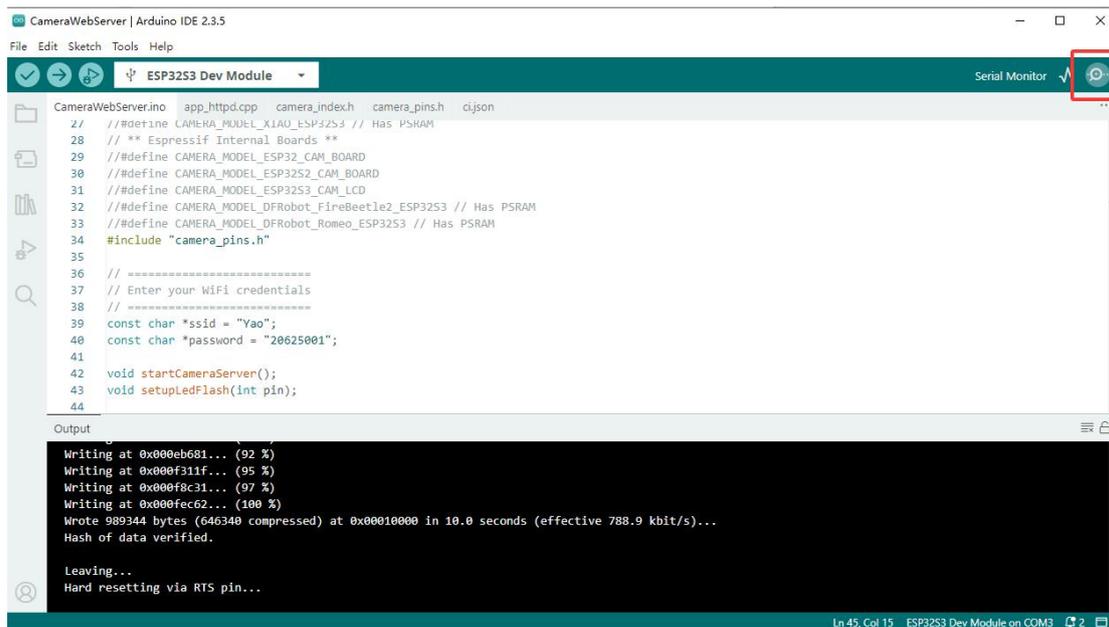


```
33 // #define CAMERA_MODEL_DFRobot_Romeo_ESP32S3 // Has PSRAM
34 #include "camera_pins.h"
35
36 // =====
37 // Enter your WiFi credentials
38 // =====
39 const char *ssid = "Yao";
40 const char *password = "20625001";
41
42 void startCameraServer();
43 void setupLedFlash(int pin);
44
45 void setup() {
```

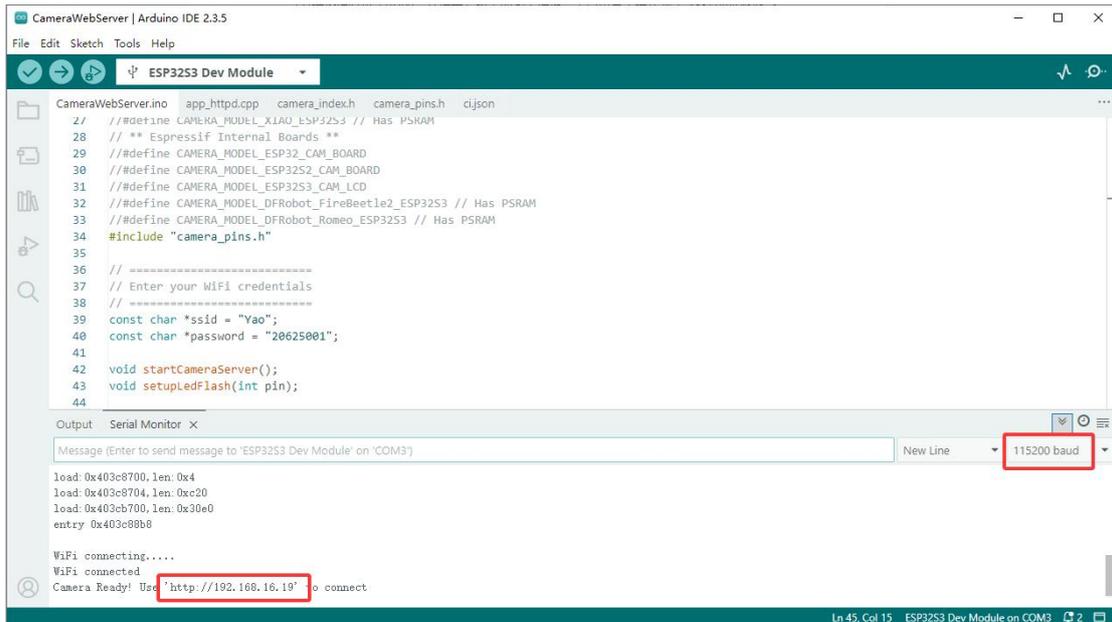
(4) Save. Compile the upload program



(5) Open the serial port monitor



(6) Select the baud rate 115200 and press the RST button on the development board, the serial port shows that it is connected to WiFi and is assigned the network address: <http://192.168.16.19>



(7) Copy the display IP address (<http://192.168.16.19>) in the serial monitor to the browser and open it, then you can use ESP32-S3-CAM through LAN, click Start Stream to see the real-time screen transmitted from ESP32-S3-CAM.

Rendering:

