ESP32-CAM-MB development board

Catalogue

Product Introduction

The ESP32-CAM development board is a multifunctional board that integrates the ESP32 chip and camera module, suitable for IoT projects, especially those applications that require image capture and transmission.

The ESP32-CAM development board features an ESP32-S chip, an OV2640 camera, a micro SD card slot, and several GPIO ports for connecting peripherals. This module is a small-sized camera module that can work independently as the smallest system. A new WiFi+Bluetooth dual-mode development board designed based on ESP32, featuring onboard PCB antennas and 2 high-performance 32-bit LX6CPUs. It adopts a 7-stage pipeline architecture and has a frequency adjustment range of 80MHz to 240MHz. ESP32-CAM is an 802.11b/g/n Wi Fi+BT/BLE SoC module with ultra-low power consumption and deep sleep current as low as 6mA, making it suitable for IoT applications with high power consumption requirements. Easy to use, ESP32-CAM is a miniature module with camera function, equipped with OV2640 camera, GPIO for connecting peripherals, and micro SD card for storing captured images. It can be directly plugged into the motherboard for use.

ESP32-CAM-MB, as an IoT camera module based on ESP32 chip, integrates the functions of microcontroller unit (MCU) and image sensor, and is suitable for various application scenarios that require image capture and wireless transmission. It can be widely used in various IoT scenarios, including home smart devices, industrial wireless control, wireless monitoring, QR wireless recognition, wireless positioning system signals, and other IoT applications. It is an ideal solution for IoT applications.

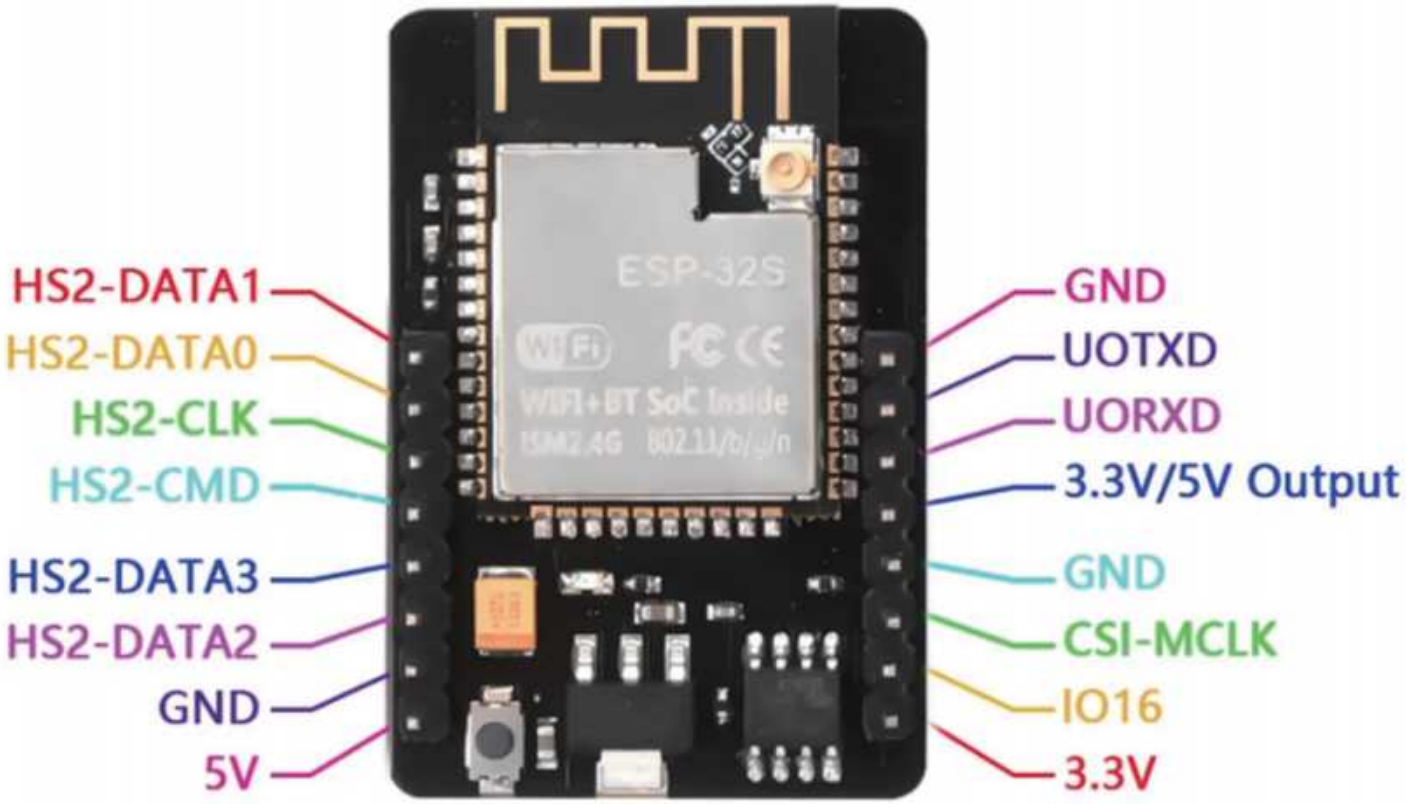
Performance Introduction:
Processor: Dual core 32-bit LX6 microprocessor
Main frequency: up to 240 MHz
Computing capability: up to 600 DMIPS
SPI Flash： Default 32mbit
Built in SRAM: 520 KB
External PSRAM: 4 MB/8 MB
Wi-Fi： 802.11b / G / n / e / i
Bluetooth: Bluetooth 4.2BR/EDR and BLE standards
Supports interfaces (2Mbps): UART, SPI, I2C, PWM
Support TF card: Supports up to 4G
IO ports: 9
Serial port speed: default 115200bps
Spectrum range: 2400 ~ 2483.5 MHz
Antenna shape: Integrated PCB, 2dBi gain
Working temperature:- 20℃～70℃
Storage environment:- 40℃～125℃， < 90% RH

OV2640 Image Sensor:
OV2640 is a widely used camera module, mainly used in embedded systems and IoT devices, especially in conjunction with development boards such as Arduino, ESP32, and Raspberry Pi. The performance parameters are as follows:
1. Image sensor: OV2640 adopts a 1/4 inch CMOS sensor, which can provide good image quality.
2. Resolution: Supports multiple resolutions, 2 million pixels, easy to choose according to application scenarios.

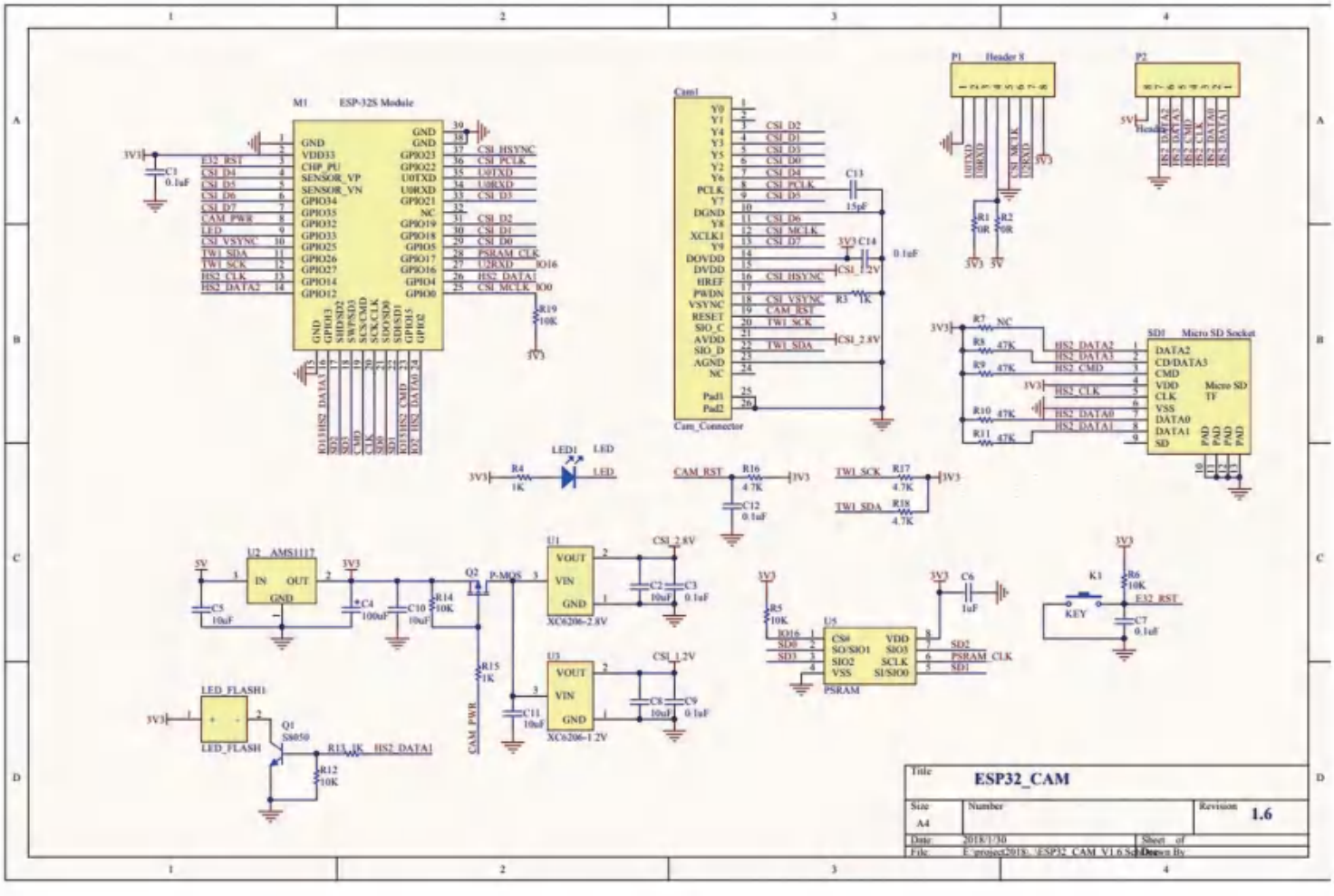
3. Frame rate: At maximum resolution, the OV2640 can achieve a shooting speed of approximately 15 frames per second (fps). Reducing the resolution can increase the frame rate, for example, it can reach over 30fps in VGA mode.

4. Image output format: JPEG (only supports OV2640) BMP、GRAYSCALE

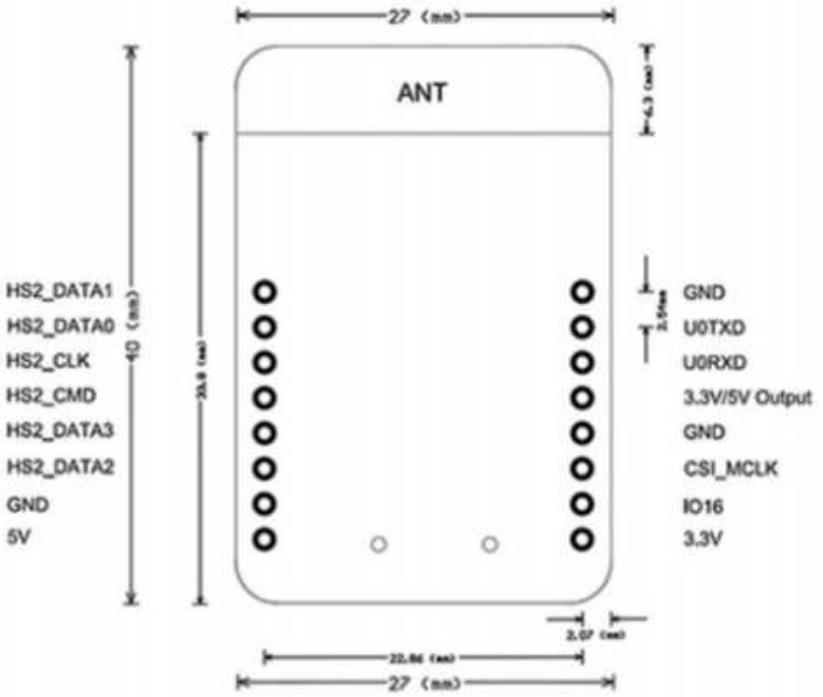
Product Pin Description



CAM	ESP32	SD	ESP32
D0	PIN5	CLK	PIN14
D1	PIN18	CMD	PIN15
D2	PIN19	DATA0	PIN2
D3	PIN21	DATA1/flash	PIN4
D4	PIN36	DATA2	PIN12
D5	PIN39	DATA3	PIN13
D6	PIN34		
D7	PIN35		
XCLK	PIN0		
PCLK	PIN22		
VSYNC	PIN25		
HREF	PIN23		
SDA	PIN26		
SCL	PIN27		
POWER PIN	PIN32		



Product size



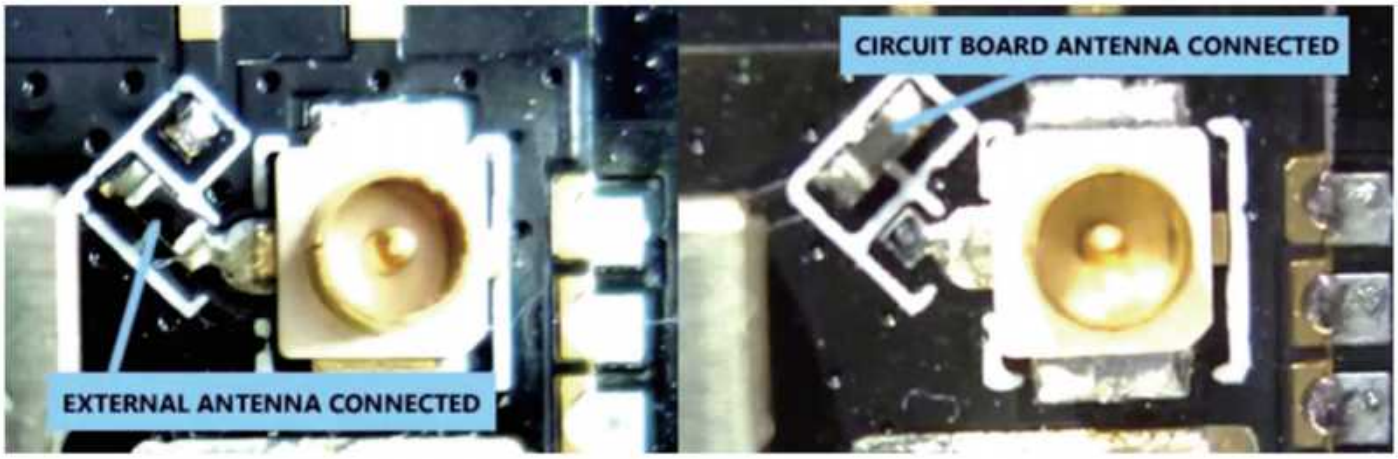
Instructions for use

Preparation stage

1. Hardware preparation: Ensure that you have the ESP32-CAM module, USB to TTL module (for data transfer), power adapter or battery, and necessary connection cables such as DuPont cables.

2. Software preparation: Install Arduino IDE. If your Arduino IDE version does not include development support for ESP32, you will need to install an additional ESP32 development board package. You can find the option to manage IDE packages in the preferences of Arduino IDE, and then search for and install the "ESP32" package.

The default antenna for ESP32 cam is an external antenna, and the signal may not be very good without an external antenna. Check if the jumper 0K resistor on the antenna connector is in the correct position for the desired antenna.



Introduction Tutorial

1、 Materials

Arduino

ESP32-CAM

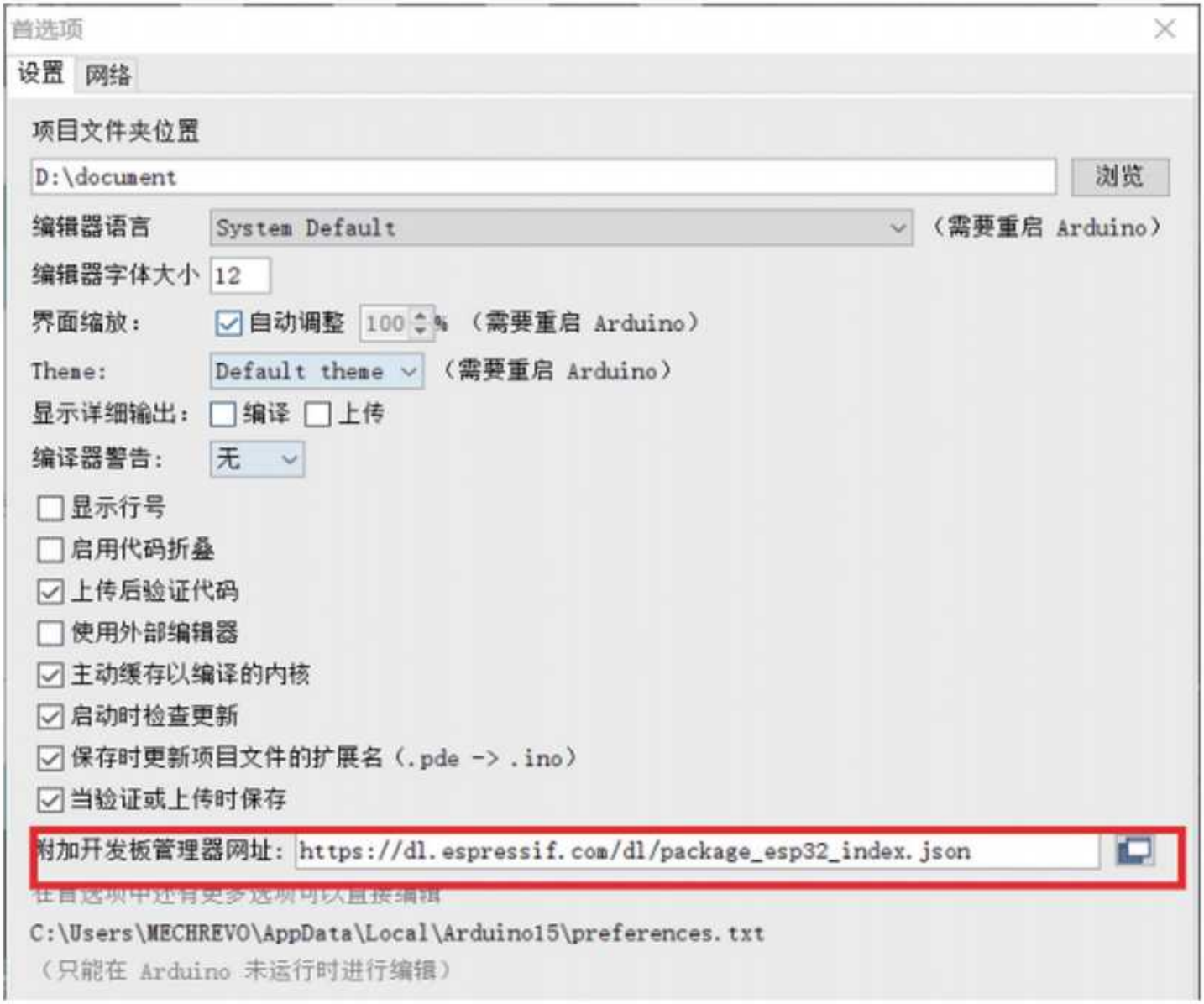
Download module (can be used by connecting USB, very convenient, if not purchased, only funds can be used with USB-TTL) as shown in the following picture

Two together as shown in the following picture

2、 Arduino+ESP32 development environment setup

Download Arduino from the official website https://www.arduino.cc/en/-software (It is recommended not to download the beta version as there may be issues) If the official website is too slow, you can download Arduino Chinese Community - Powered by Discuz! Here

Install ESP32 development package, file ->preferences ->add development board manager input https://dl.espressif.com/dl/package_ esp32_-index.json



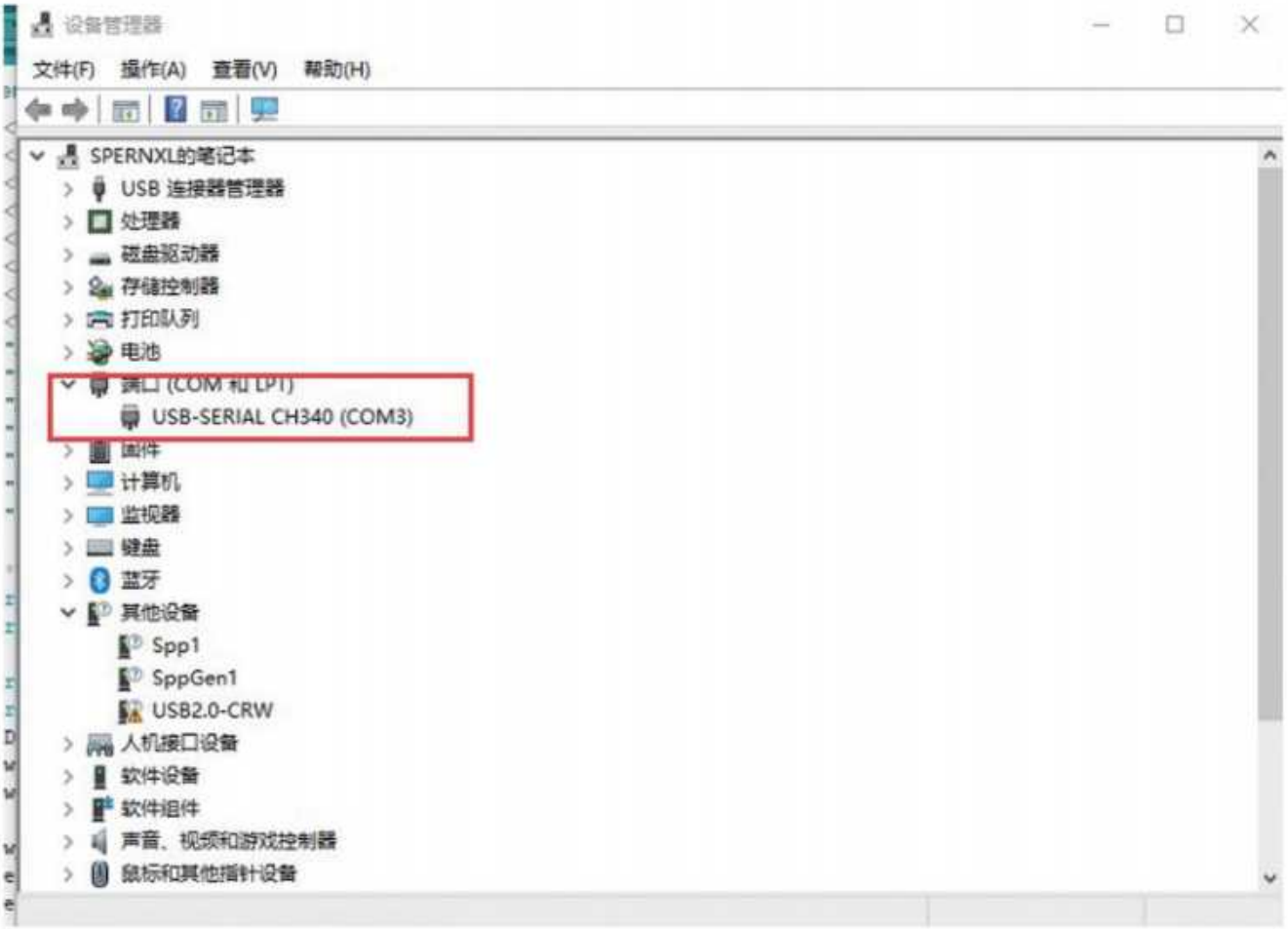
Then restart Arduino, go to Tools ->Development Board ->Development Board Manager, search for ESP32 download (as shown below)

The environment has been configured up here

3、 Burn program

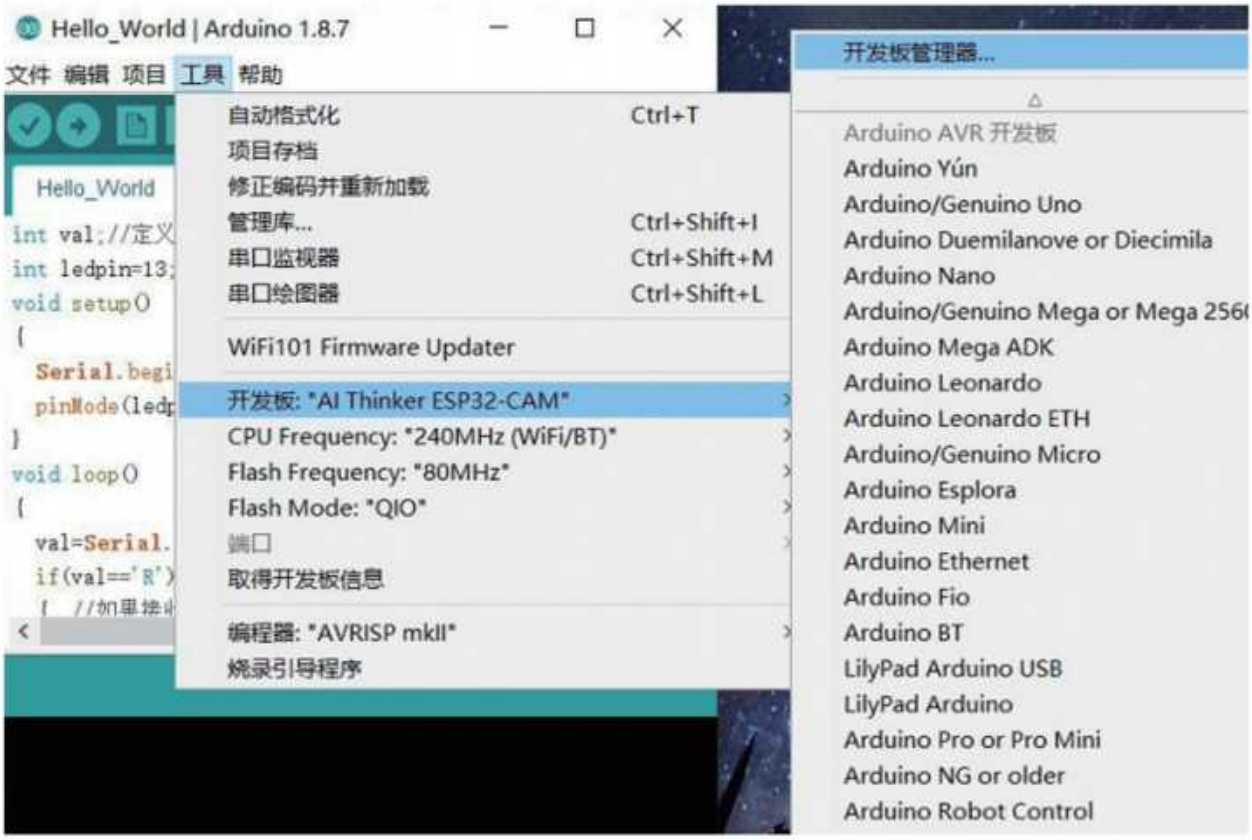
1. Preparation

1.1 Connect the esp32 cam to the computer. If the connection is successful, it will be as shown in the following figure



If not connected, it may be because the USB cable does not have data transfer function

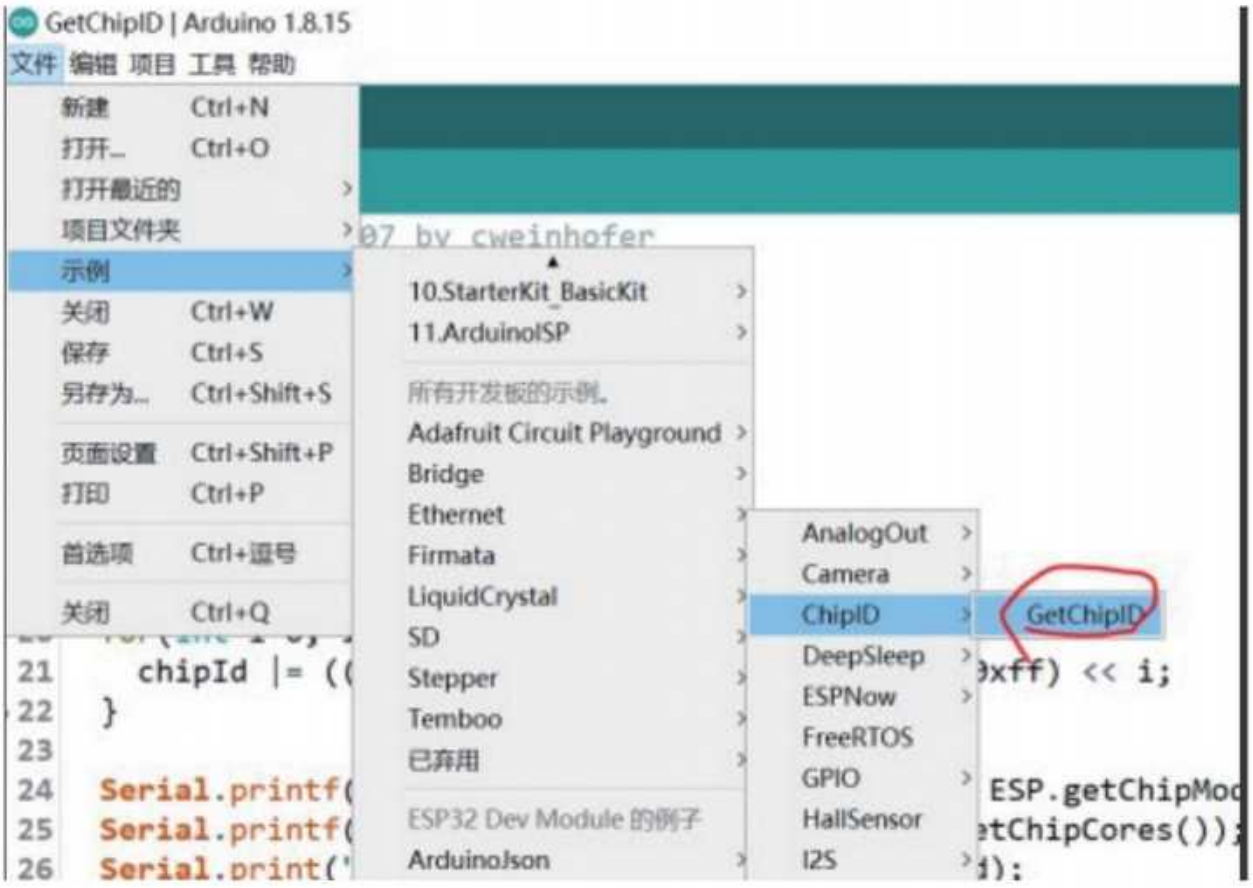
1.2 Tools ->Development Board Select AI Thinker ESP32-CAM Development Board



1.3 Check the port of your device

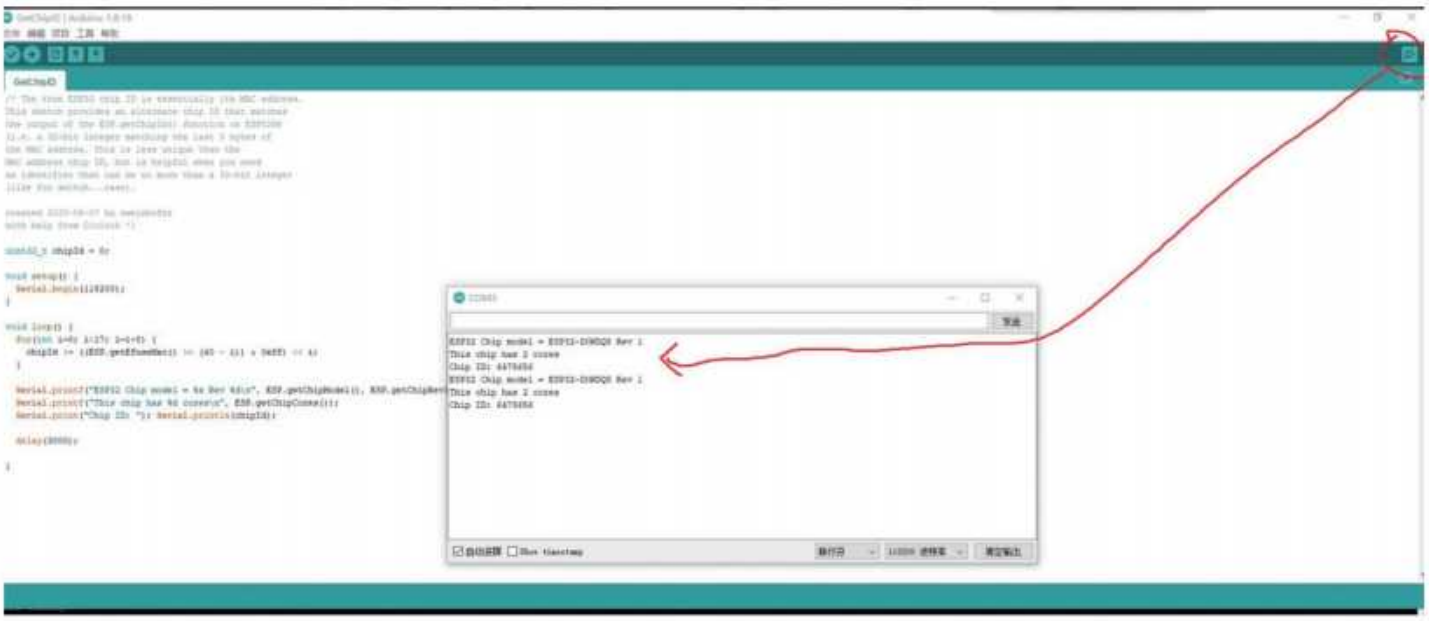
4、 Verification

1. Open the official file example: ChipID， Obtain the development board chip ID number.



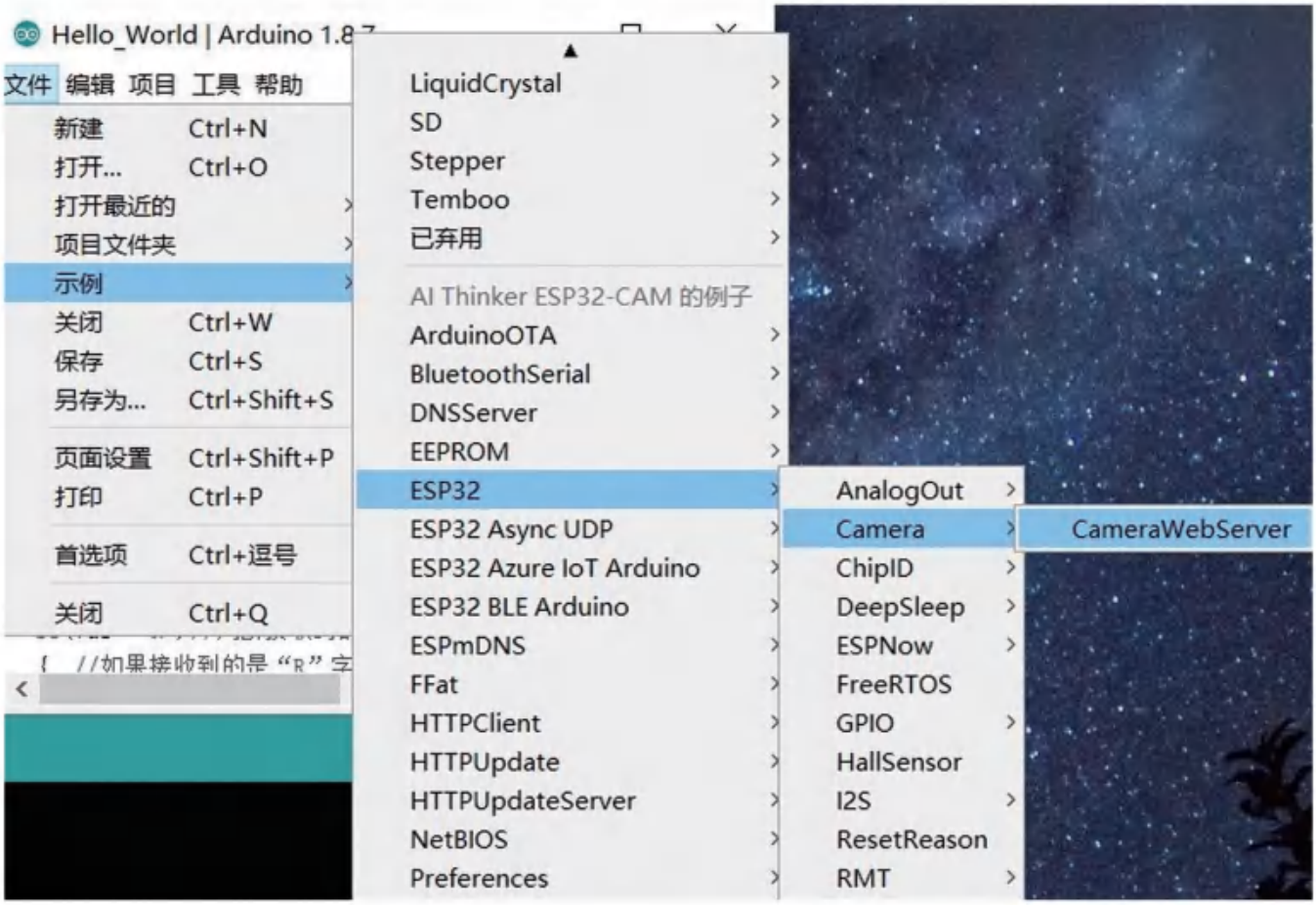
2. Click the burn button

3.After successful burning, click on the serial port monitor in the upper right corner to see the ID number of the ESP32 chip. Prove successful environment setup!



5、 Example of using ESP32 CAM

1. Open the file in Arduino ->Example ->ESP32->Camera ->CameraWebServer



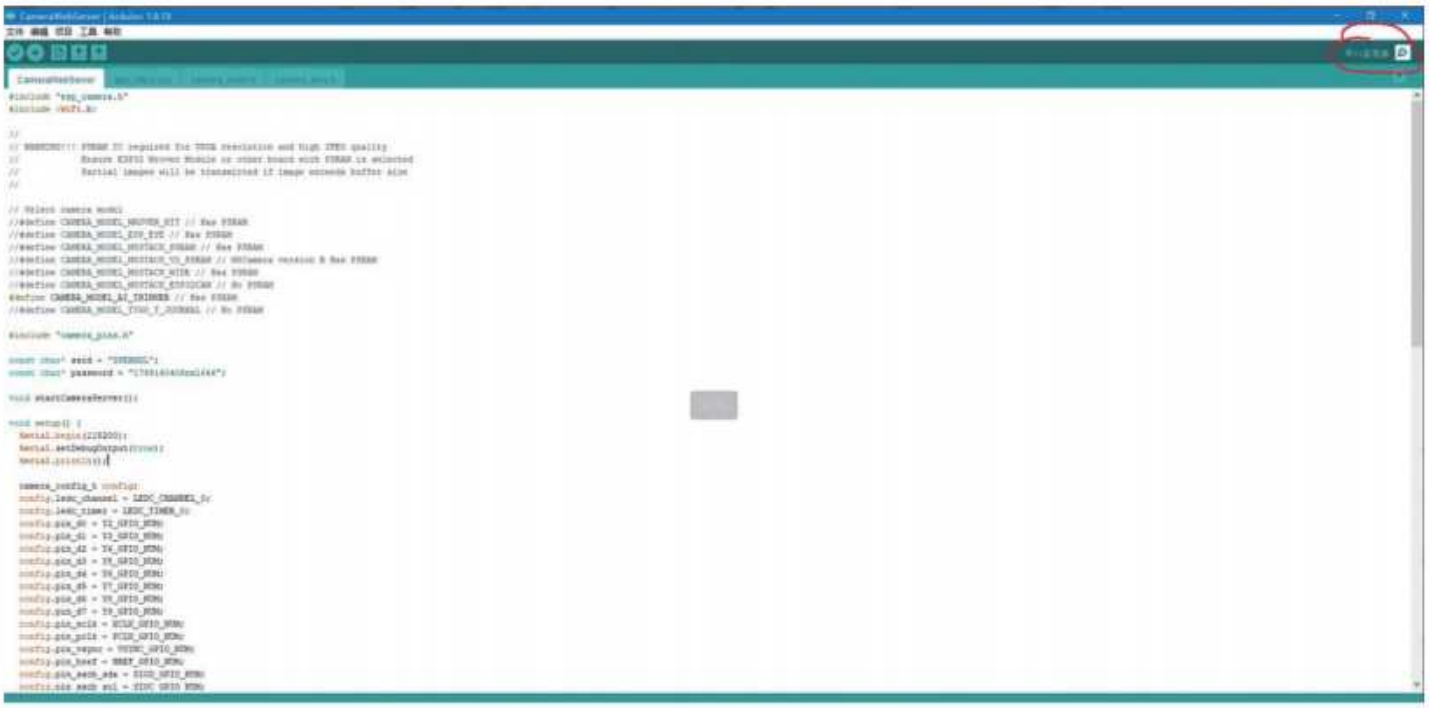
(1) Unsubscribe the models you need to use, and annotate all the ones you don't need (what I have chosen here is...)
CAMERA_MODEL_AI_THINKER)

(2) Set the WiFi name and password you want to connect to in SSID and password.

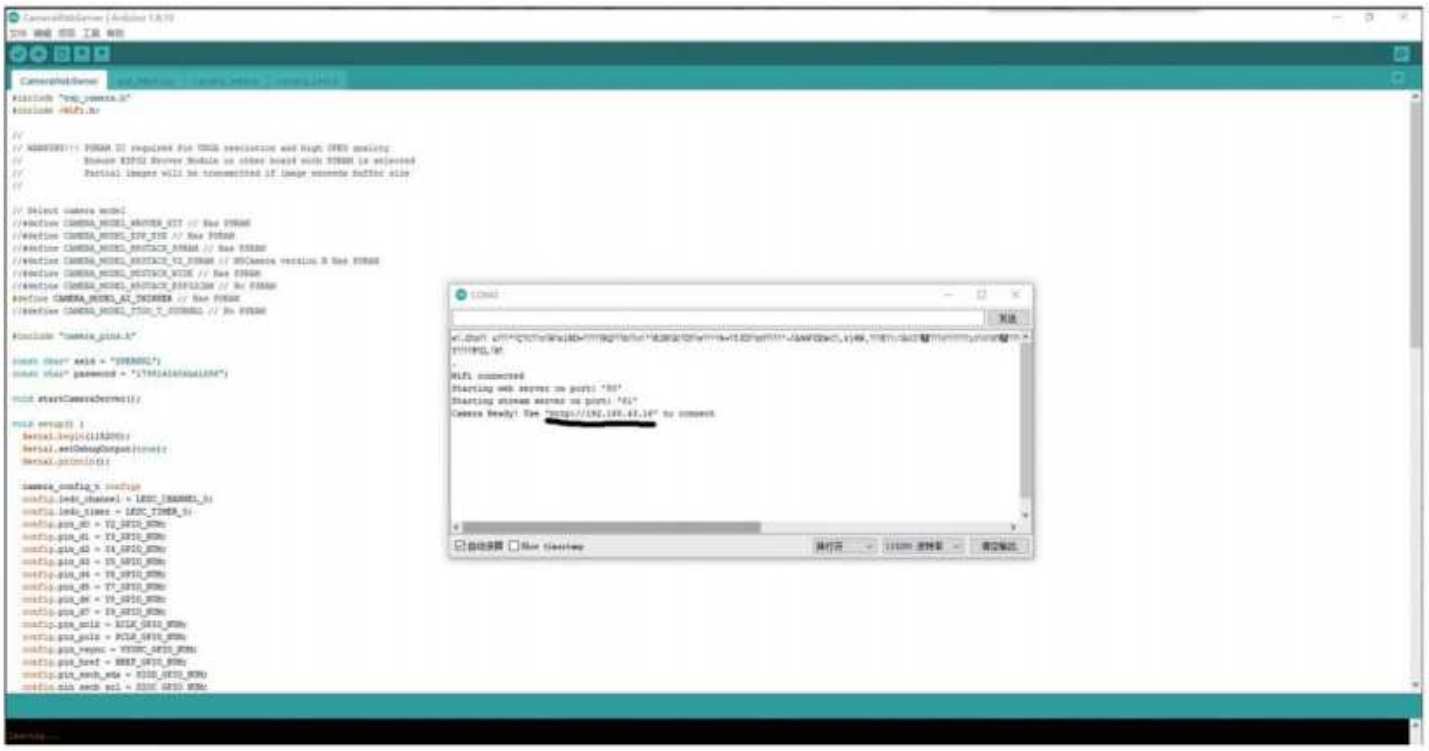
```
// Select camera model
//#define CAMERA_MODEL_WROVER_KIT // Has PSRAM
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM

#include "camera_pins.h"

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

After setting it up, click Burn

After successful burning, click on the serial port listener in the upper right corner

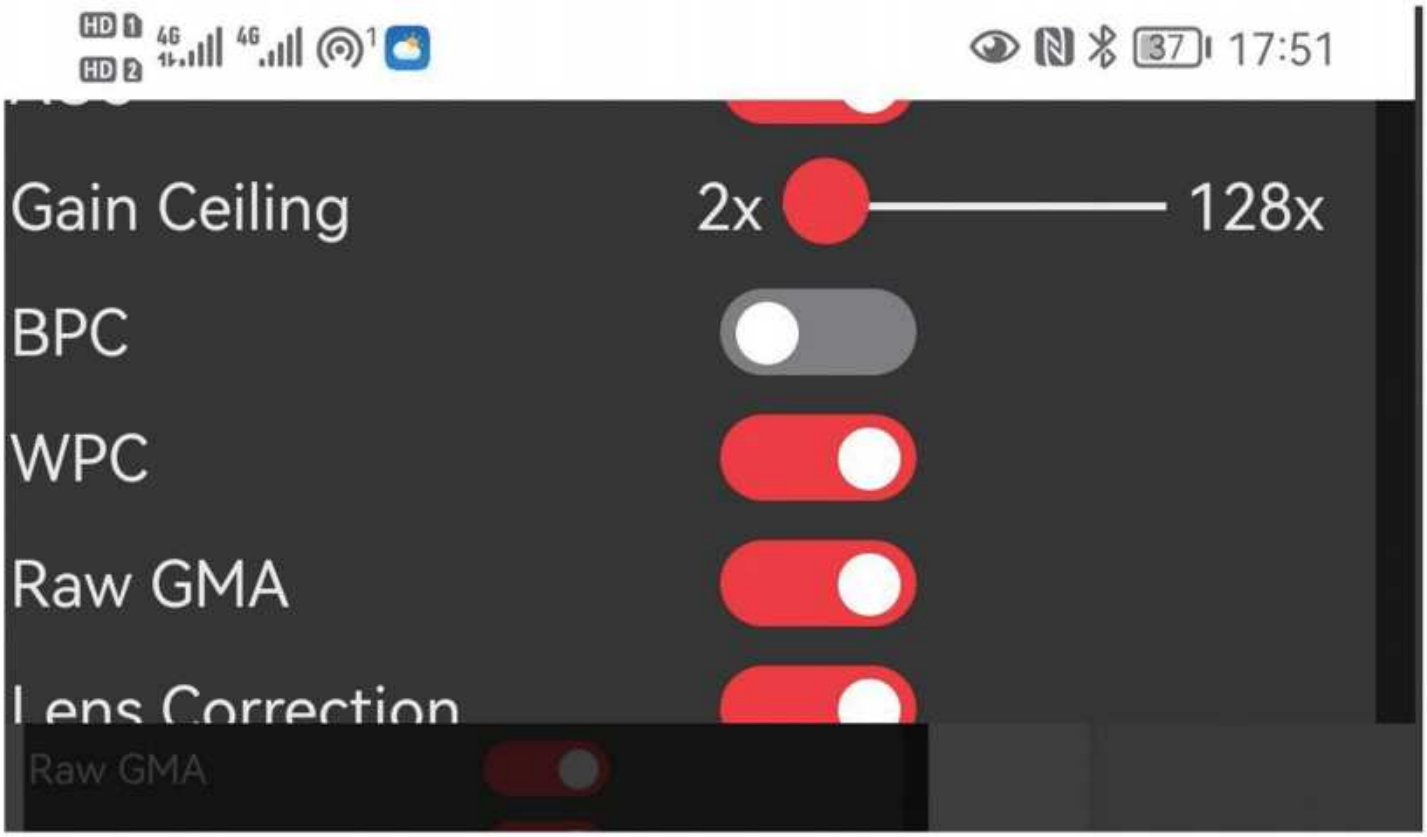


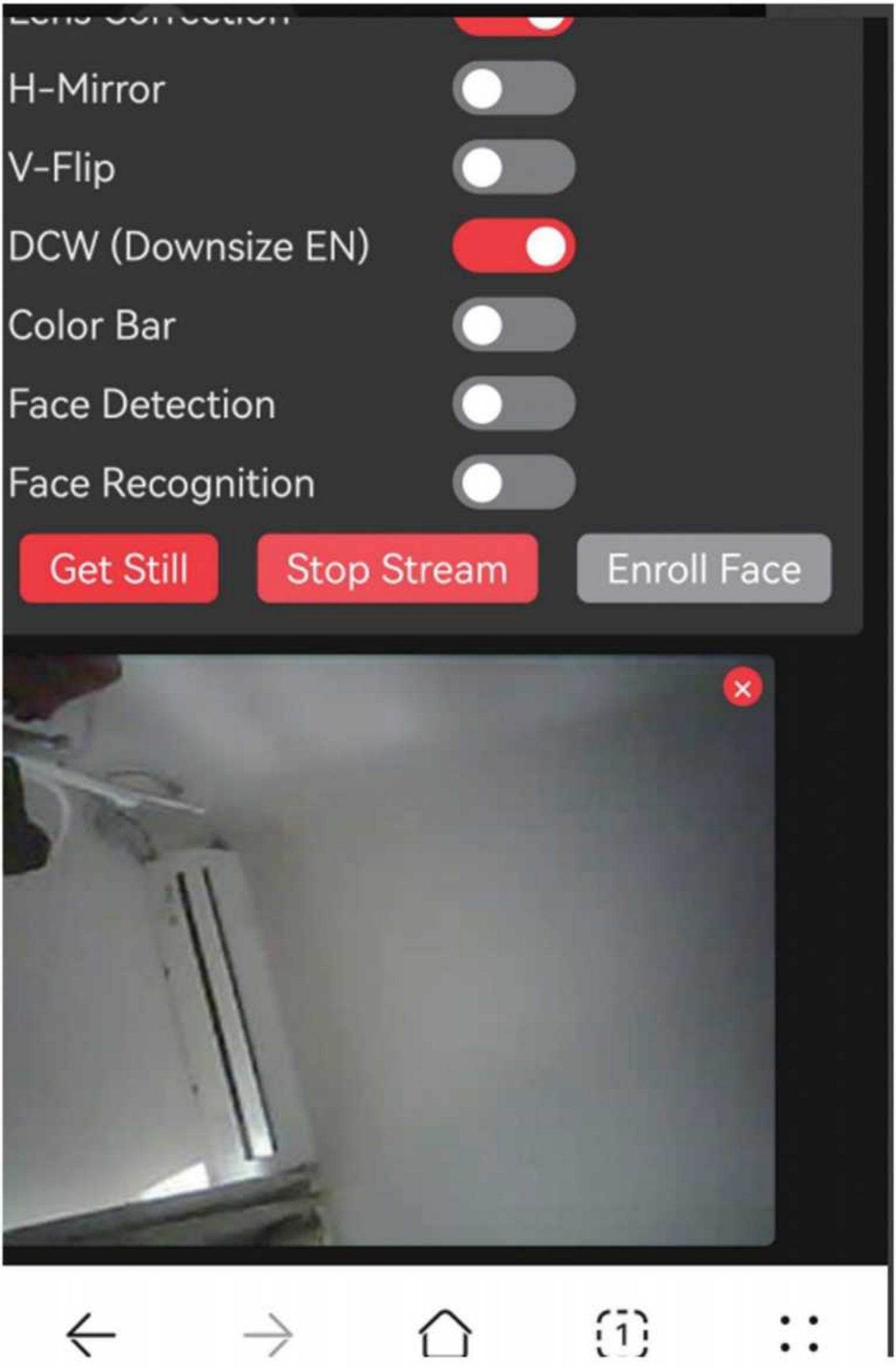
Open the serial monitor at a baud rate of 115200 and press the reset button on the ESP32-CAM board. The following window will pop up. (There are some garbled text on my page, but it doesn't affect its usability)



We can copy the display IP address from the monitor to the browser and open it to use ESP32-CAM through the local area network (which is where I drew the horizontal line)

Click on Start Stream to see the real-time image transmitted by ESP32-CAM.




H-Mirror
V-Flip
DCW (Downsize EN)
Color Bar
Face Detection
Face Recognition
Get Still
Stop Stream
Enroll Face