复杂机电产品程序开发实例

# 基础测试

## 安装arduino

## 添加其它库

文件->首选项

<https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json>

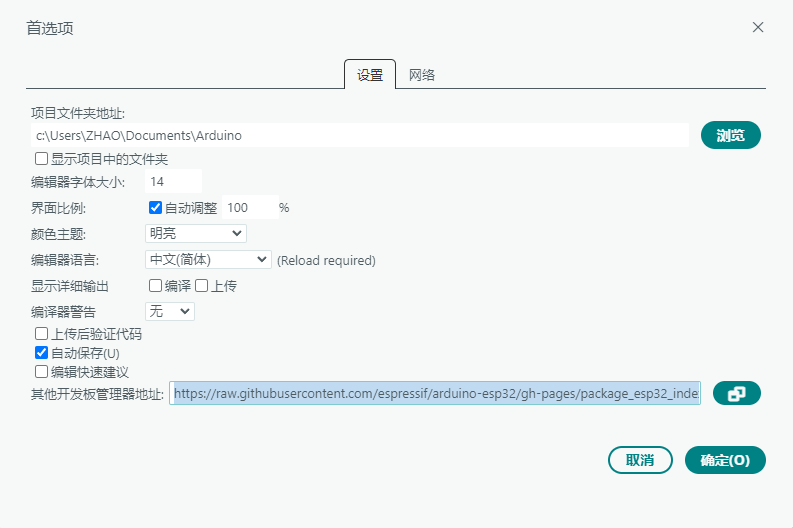


图 1 添加esp32的资源链接

## 添加开发板

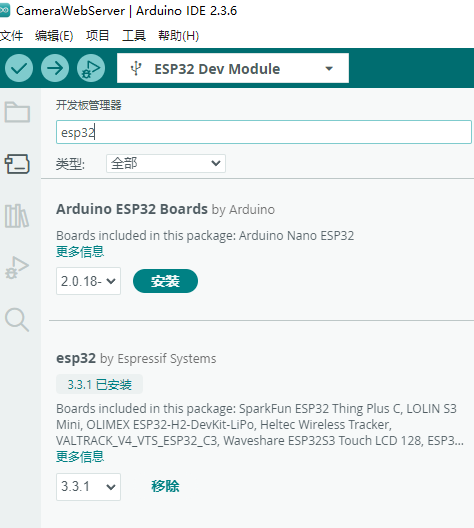


图 2 添加esp32开发板支持

# 添加设备配置

将esp32开发板的usb查到电脑上，同时打开电脑的设备管理器，查看对应的串口号，一般在：计算机->管理->设备管理器，里面的端口里，可以看到有个USB之类的转字样的，对应的“COM4”就是串口号：

图 3 查找设备管理器，查看ESP32的串口号

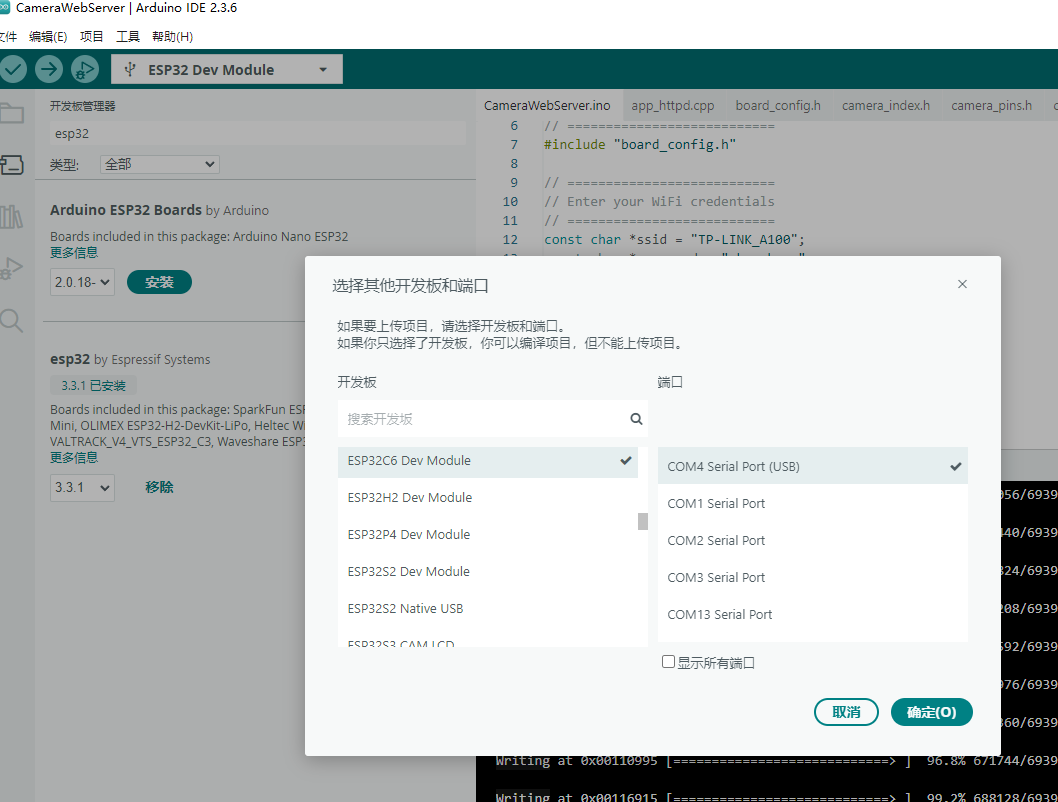
 

图 4 配置Arduino连接ESP32开发板

# 视频流开发

## 创建工程

可以从零开发，也可以基于官方的案例来做，流程如下：

打开 Arduino IDE, 选择 选择 ESP32 Dev Module 开发板, 选择 File->Examples->ESP32->Camera->CameraWebServer 例程驱动 OV2640 在局域网内的设备访问, 需要填写当地 WiFi 账号密码

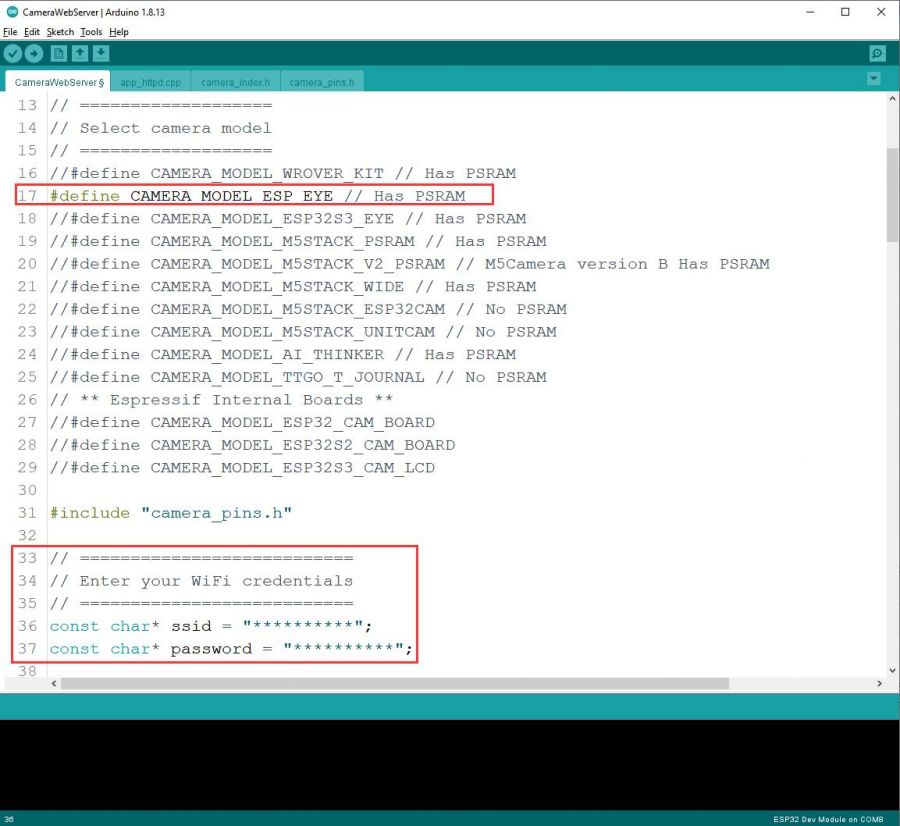


图 修改配置，并保存工程到某个位置

## 烧录配置

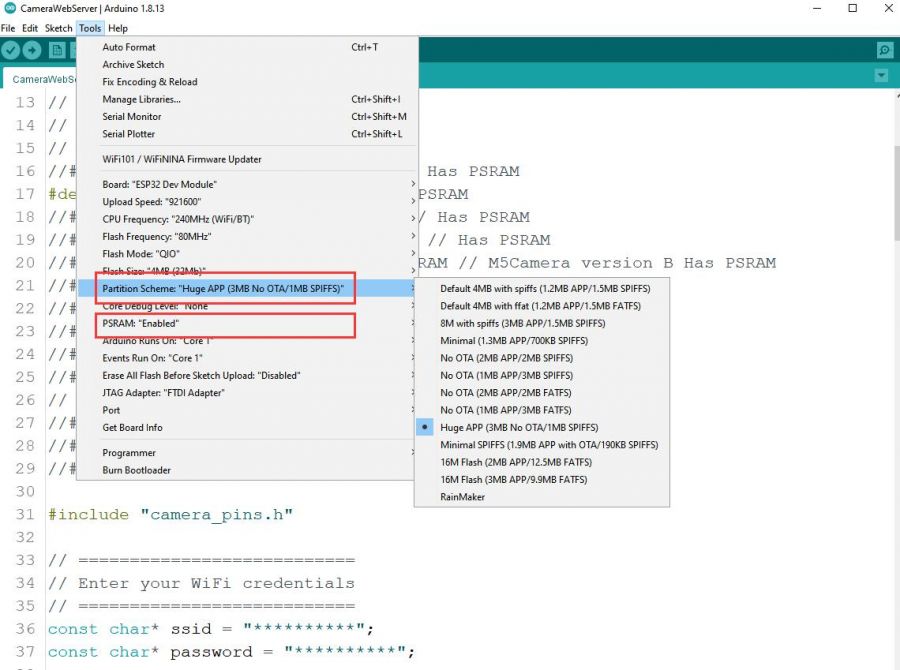


图 6 烧录配置

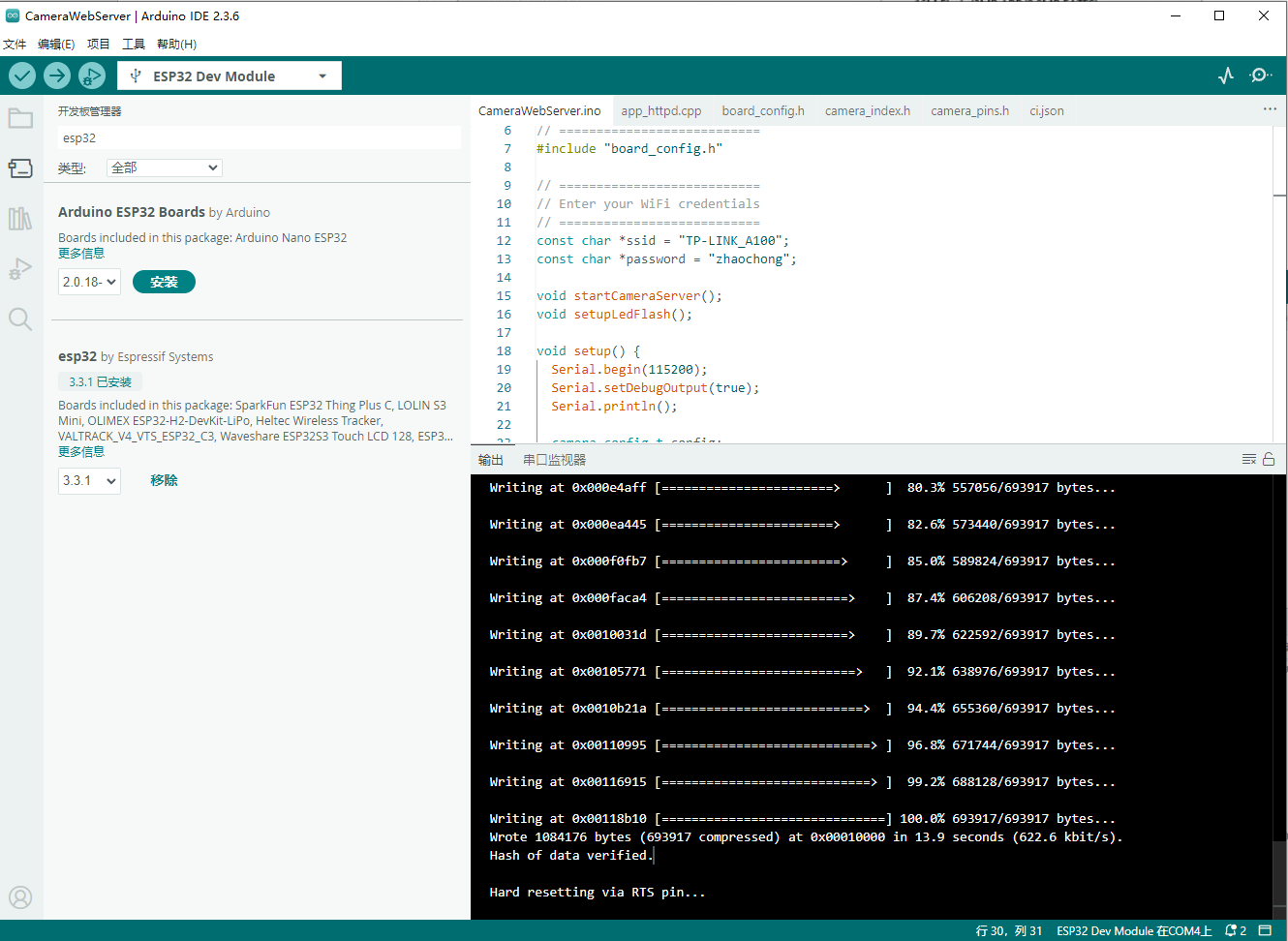


图 7 烧录上传：点击右上角的➡

在“输出”窗口看到烧录完成即可，如果提示设备连接不到ESP32，可以在“connecting”的时候，按一下开发板的“rest”按键。

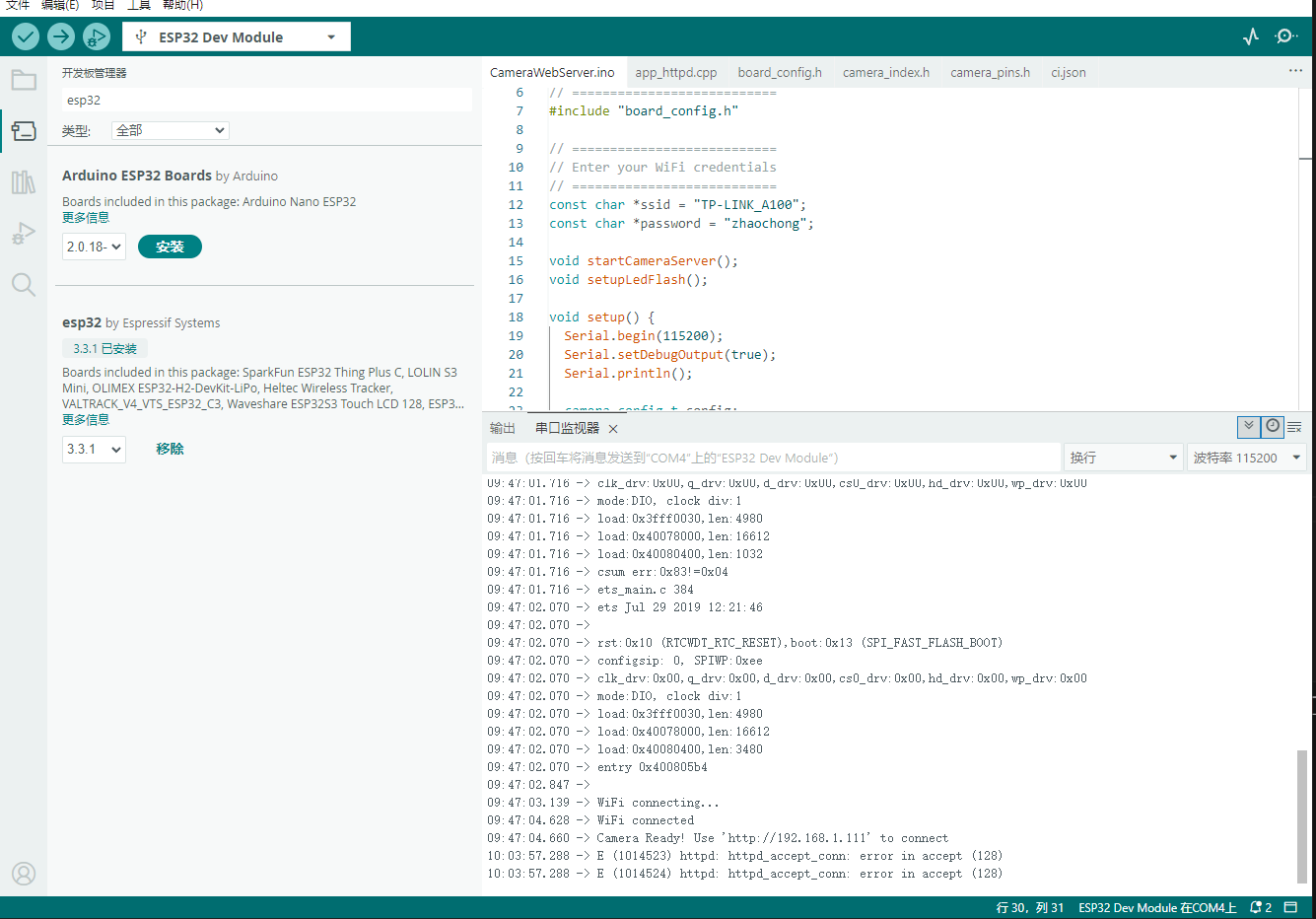


图 8 查看wifi地址：点击右上角的圆圈标识，可以看到“串口监视器”，里面输出“Camera Ready！”即可

如果配置的wifi热点账号、密码正确，按开发板“rest”，可以在串口监视器里看到输出的ip地址字样，即完成软件烧录和配置。

## 测试视频流

测试流，打开浏览器，输入上述输出的ip地址，如：

09:47:03.139 -> WiFi connecting...

09:47:04.628 -> WiFi connected

09:47:04.660 -> Camera Ready! Use 'http://192.168.1.111' to connect

则浏览器输入：<http://192.168.1.111>，打开网页，点击getstream，能看到视频即可：

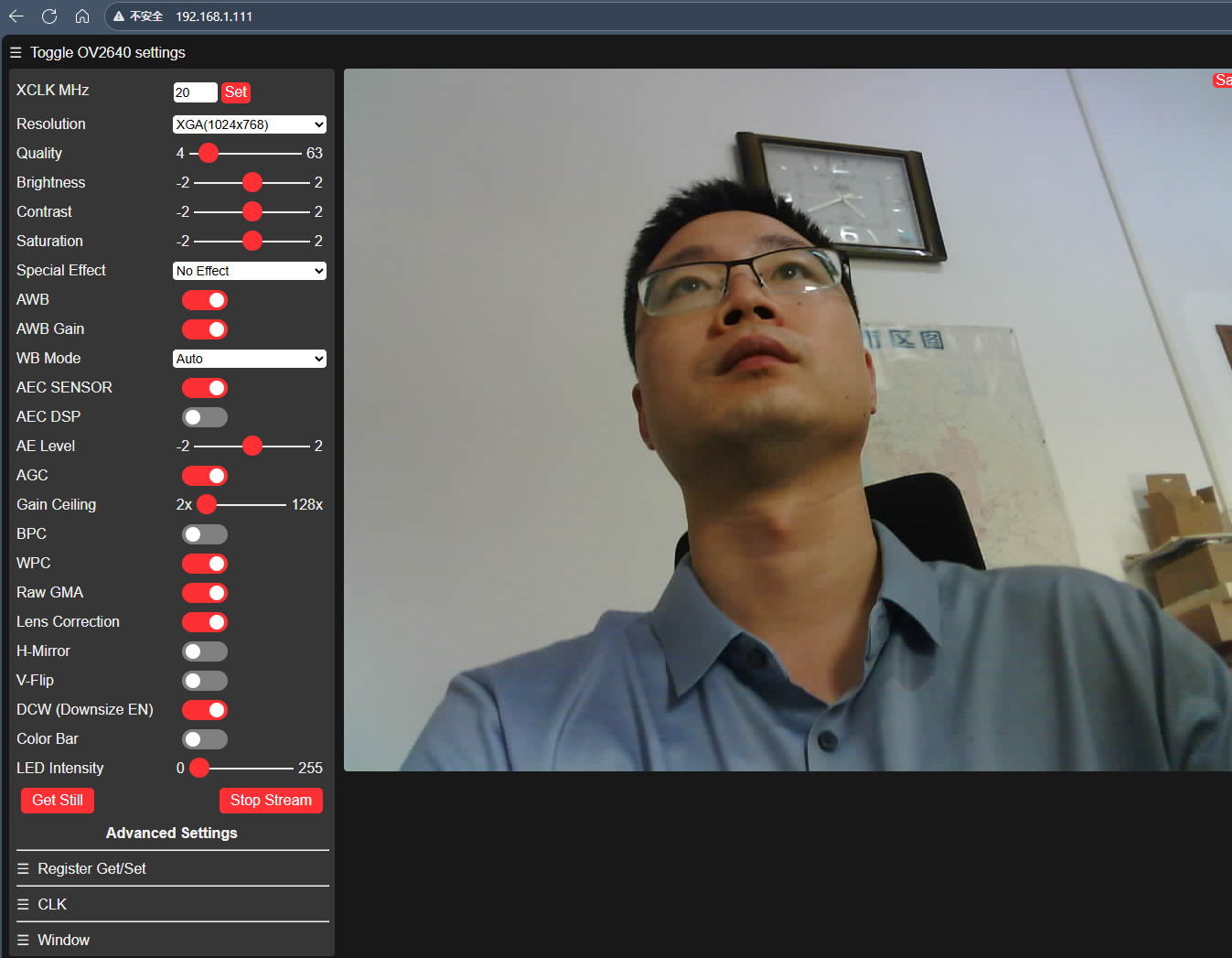


图 9 打开视频流

# 深度学习测试

前提：按照之前的环境配置文档，已完成python环境的配置，如已安装cv2、yolo方可进行下步实验。

*注：后续我这里环境用的y8，已安装cv2、yolo。如果需要安装这两个库，教程在之前的环境配置文档里。*

## 编写python程序

代码如下，复制代码保存为video.py即可（文件名可任意）：同时可提前下载yolo11n.pt模型放到与video.py代码同路径下，否则网络不好不一定能下载下来【附件已提供】。

*注：代码中的ip地址需要替换成你自己的，我这里是http://192.168.1.111:81/stream*

import cv2

from ultralytics import YOLO

# Load a pretrained YOLO11n model

model = YOLO("yolo11n.pt")

cap = cv2.VideoCapture("http://192.168.1.111:81/stream")

while True:

    success, img = cap.read()

    results=model(img)

    for r in results:

        boxes=r.boxes

        for box in boxes:

            x1,y1,x2,y2=box.xyxy[0]

            x1,y1,x2,y2=int(x1),int(y1),int(x2),int(y2)

            cv2.rectangle(img,(x1,y1),(x2,y2),(255,0,255),3)

            conf=box.conf[0]

            cls=int(box.cls[0])

            cv2.putText(img,f'{model.names[cls]} {conf:.2f}',(x1,y1-10),cv2.FONT\_HERSHEY\_SIMPLEX,0.9,(255,0,255),2)

    cv2.imshow("Video", img)

    if cv2.waitKey(1) & 0xFF == ord('q'):

        break

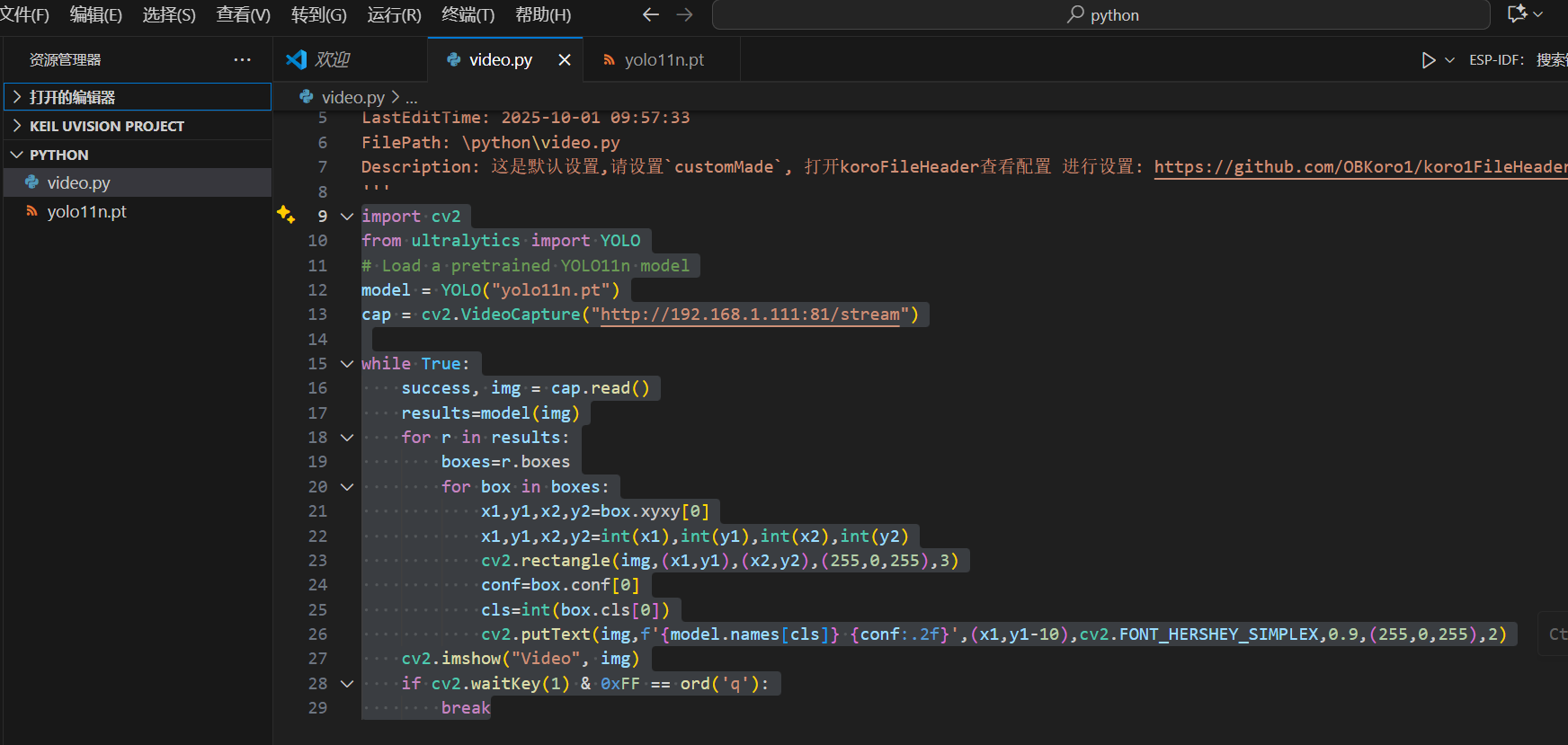
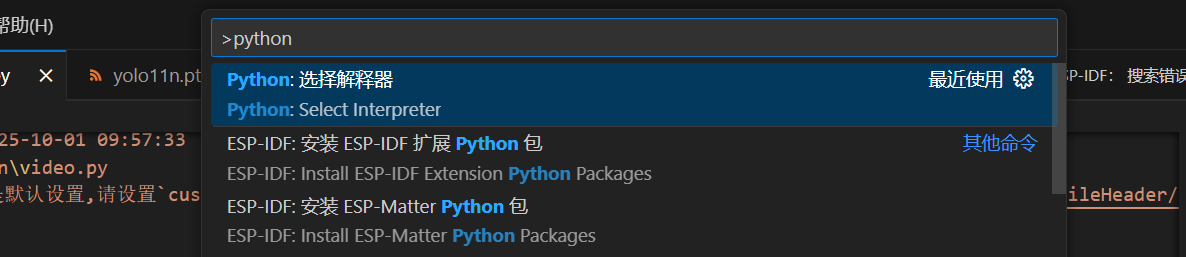


图 10 vscode代码界面

## 配置python启动

Vscode的python虚拟环境需要预先选择：ctrl+p，输入>python，选择python:选择解释器，选择你创建的虚拟空间，我这里用的y8作为名称，选择y8：



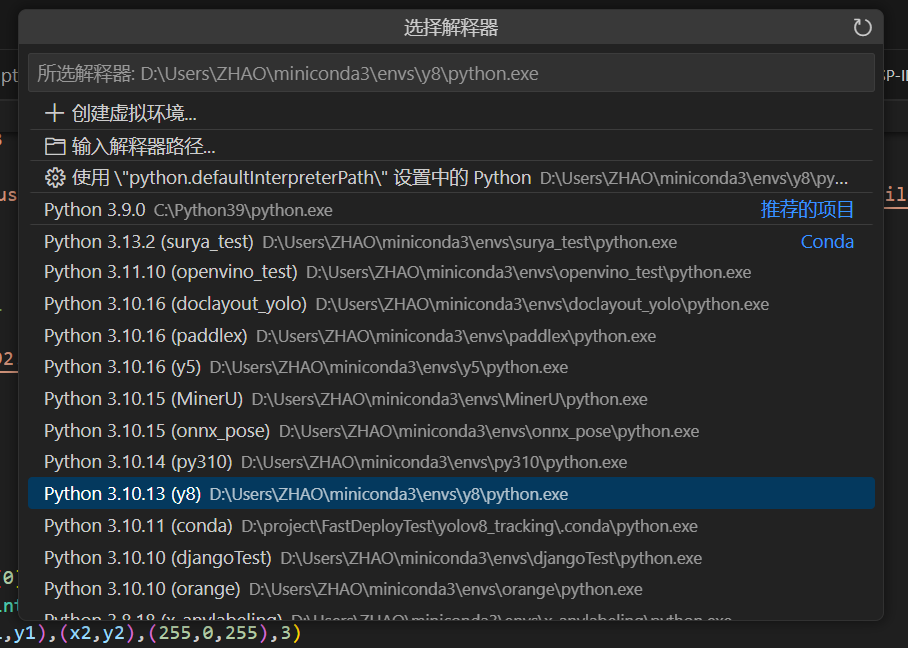


图 11 配置选择vscode的python环境

## 执行测试

Vscode菜单“终端”，打开控制台：

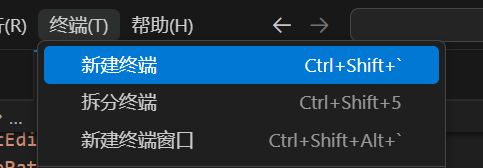
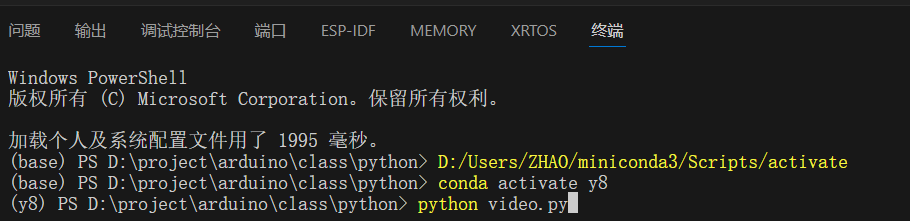
 

图 11 启动命令行，执行命令

命令行中输入：python video.py，回车即可，即可看到抓取视频流，并推理：按‘q’可退出程序。

*注：受限于ESP32的算力和程序单线程的约束，python程序和浏览器访问摄像头不可同时访问摄像头。如果发现python程序卡住，可查看浏览器或控制台，确保网页页面已关闭、python程序全部退出。*

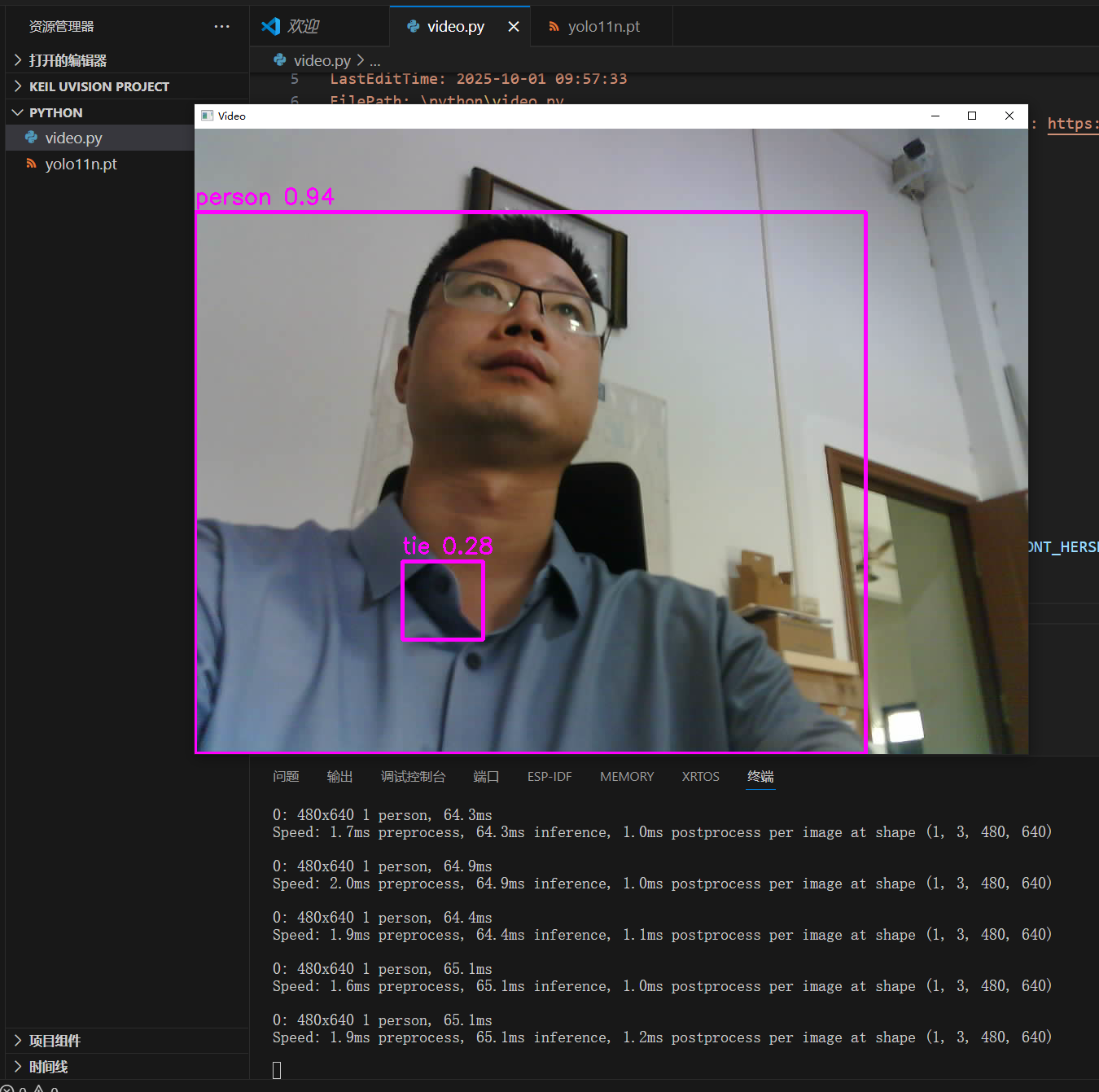


图 11 yolo推理