



ESP32 WiFi Network IP Camera Arduino Source Code Manual



M. Mahdi K. Kanan

Rev 1.1

With this source code, and the suggested circuit, you can make a WiFi IP Camera to record the video with an “ESP32-CAM” module.

Also this project has a WebApp (Web UI) which lets the user to see the streams, explore the memory and simply configure the system and WiFi connections.

Compilation and programming of this source code is compatible with “AI Thinker ESP32-CAM” and “Arduino ESP32 board” rev 3.2.0

WiCardTech Engineering Group

<https://wicard.net>

mvtdesign@gmail.com

1 / 5 / 2023

Table of Contents

Table of Figures.....2

Other Revisions.....3

Introduction4

Uploading the project.....5

Preparing The ESP32 IPCam.....6

The ESP32 IP Camera Web Application7

 The Camera Configuration Section:.....8

 Video Streaming Page.....9

 Memory page.....10

Programming Manual10

Revision History12

Project Update.....13

Table of Figures

Figure 1-Arduino settings 5

Figure 2-ESP32-Cam inside of an enclosure6

Figure 3-ESP32 WiFi Camera Microphone6

Figure 4-WiFi config.....7

Figure 5-Camera config.....8

Figure 6-Video stream9

Figure 7-Memory manager10

Other Revisions

* ESP32 WiFi Microphone (12bit@8KHz - Free Revision) -->

<https://wicard.net/projects/Arduino/ESP32-ESP8266/ESP32%20WiFi%20Microphone%20free%20arduino%20source%20code>

* ESP8266 WiFi Microphone (10bit@8KHz - Free Revision) -->

<https://wicard.net/projects/Arduino/ESP32-ESP8266/ESP8266%20WiFi%20Microphone%20free%20arduino%20source%20code>

* ESP32 WiFi Microphone and Recorder --> <https://wicard.net/projects/Arduino/ESP32-ESP8266/esp32-voice-recorder>

* ESP8266 WiFi Microphone and Recorder --> <https://wicard.net/projects/Arduino/ESP32-ESP8266/nodemcu-voice-recorder-esp8266>

* ESP32-Cam IP-Camera (Free Revision) --> <https://wicard.net/projects/Arduino/ESP32-ESP8266/esp32cam-ipcamera-bin>

* ESP32-Cam HD DashCam --> <https://wicard.net/projects/Arduino/ESP32-ESP8266/esp32-cam-DashCam>

* ESP32-Cam CCTV --> <https://wicard.net/projects/Arduino/ESP32-ESP8266/ESP32-CAM%20CCTV%20IP%20Camera>

* ESP32-Cam Camera and Microphone --> <https://wicard.net/projects/Arduino/ESP32-ESP8266/spycam-video-audio-esp32cam>

Introduction

ESP32-CAM module contains a 2MP Camera, some I/O pins and one MicroSD slot to store the images and videos. With its tiny size and 240MHz CPU is able to provide a fair quality images. This project is a source code which lets you to handle the camera and use the module's special features.

This project contains two folders ESP32CamProgram (the Arduino source code) and PHP Script (For server use option).

The source code contains 7 files:

- 1- ESP32CamProgram.ino – the main program
- 2- AC.ino (camera and autoconnect configuration)
- 3- AC.h (settings and configuration headerfile)
- 4- CamSettings.h (Camera settings)
- 5- CamCapture.ino (Capturing and storing the images)
- 6- CamMemory.ino (Video player/images gallery)
- 7- CamPages.ino (Internal pages)

Uploading the project

First open one of the files with Arduino program, then set the settings as the following image (Board and CPU Frequency):

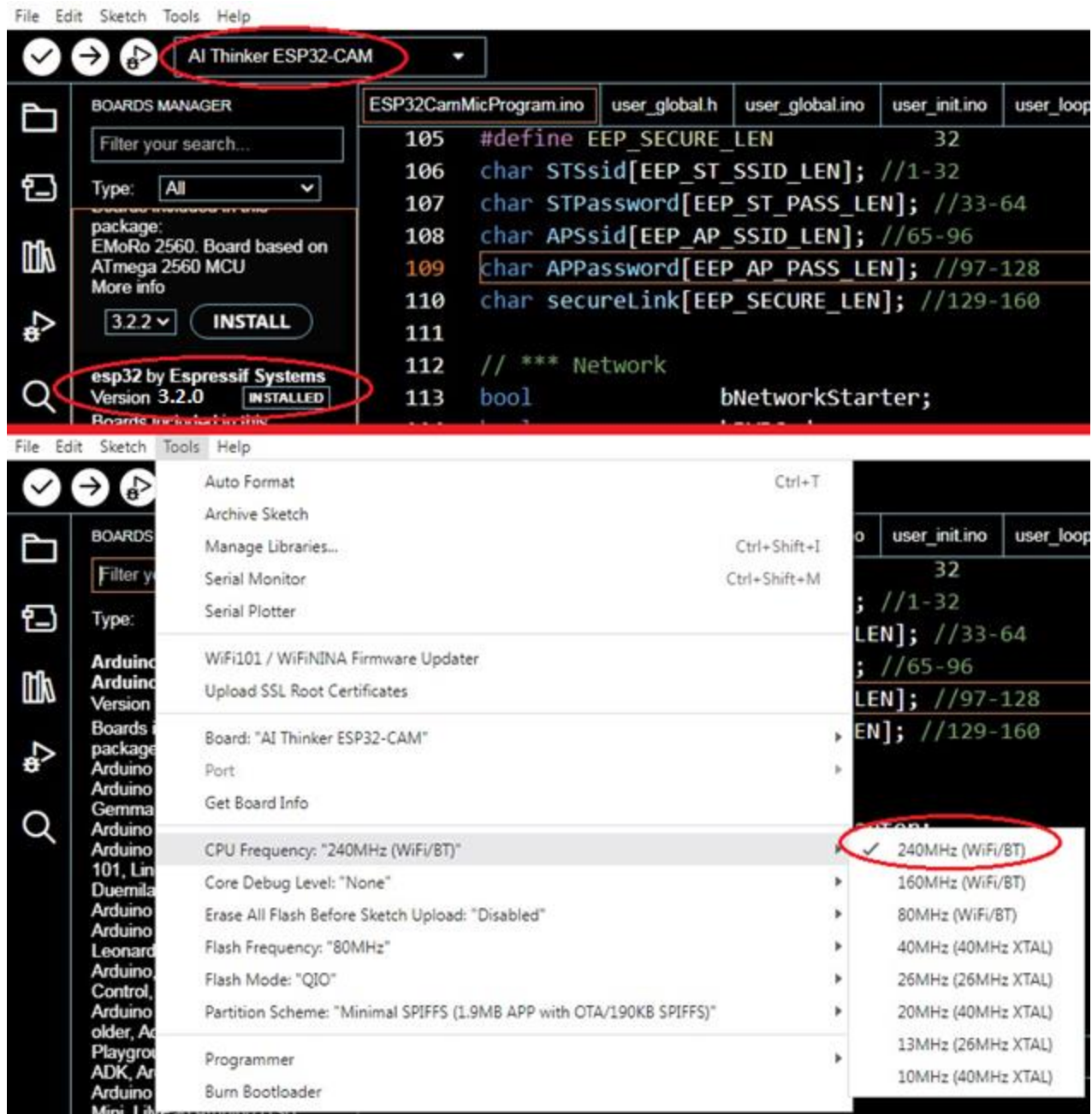


Figure 1-Arduino settings

Then turn on the ESP32 module in the “download mode” (GPIO0→GND) and upload the program with using an USB2Serial module.

- It's suggested to install (or upgrade/downgrade to) “ESP32 Boards rev 3.2.0”. This revision is more compatible with the project.

- The “Board” must be set on the compatible ESP32 Modules (Ai Thinker ESP32-Cam) before doing anything.
- The start-up CPU speed has been set to 240MHz.
- For the “download mode”, use a jumper for connecting “IO 0” to the GND.
- It’s suggested to remove the whole data in the ESP flash memory before the first upload.

Preparing The ESP32 IPCam

The “Ai Thinker ESP32-Cam” module is completed. You only need a standard 5V power source.

Also you can stick a heat sink to the module’s regulator. It’s not necessary but recommended.

- The onboard ESP32 module must only receive 3.3v. Higher voltage will hurt the module.
- Do not use higher voltage for the 5V and 3.3V input pins.

This is the assembled circuit with a 5V power source inside of an enclosure:

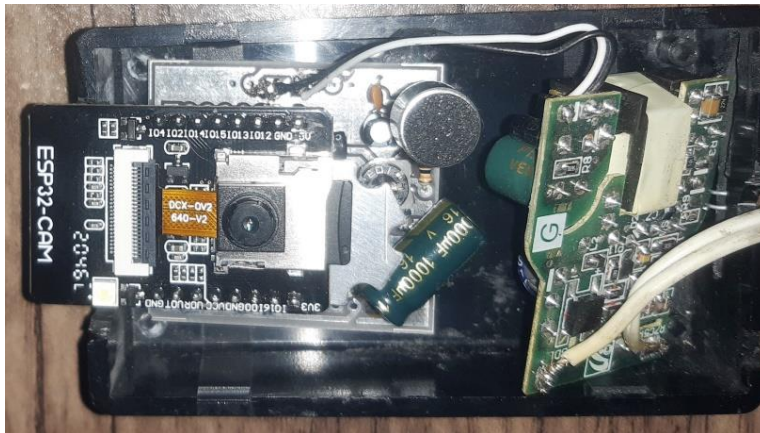


Figure 2-ESP32-Cam inside of an enclosure



Figure 3-ESP32 WiFi Camera Microphone

The ESP32 IP Camera Web Application

After upload, run the program in “normal mode”.

Then in case you are using ESP32-Cam, the red LED on the board would be turned on for 1 second then turns off. Then you'll be able to see the module's hot spot ssid via the WiFi networks in your PC or smart phone.

The default SSID is WiCardIPCam and the default password is 12345678. Connect and go to 192.168.4.1/config with a web browser.

The WiFi configuration page:

ESP32-AC Configuration

192.168.0.7/config

WiCard

Modem Configuration

Connected to: A8 [-76]
IP: 192.168.0.7

Available APs

-74
D-Link -85
JACK DANIEL'S -90
Mahya -91

MAC: f4:cf:a2:98:98:3d
SSID:
A8
PASSWORD:

Save

ESP32-AC Configuration

192.168.0.7/config

PASSWORD:

Save

Device HotSpot Configuration

SSID:
WiCardCam

PASSWORD:
12345678

SECURE LINK: 192.168.x.x/

☒ Hidden HotSpot
☒ Disable HotSpot when is connected to the modem
Save

Figure 4-WiFi config

In the “Modem Configuration” section, you’re able to view/edit the SSID and password of WiFi modem/router and then click on “Save” after inserting. The module would connect to the modem after about 30 seconds in case of validity of ssid and password.

🔗 This page is also available via the DHCP IP of the module.

In the “Device Hot Spot Configuration” section, you can set the module hotspot’s SSID and password. Also you can set a password for the internal page with Secure Link section. (Example: the secure link is ABCD and the root page would be at 192.168.4.1/ABCD/ address) The “Hidden HotSpot” button will set the module’s hot spot as hidden hotspot and the “Disable HotSpot when is connected to the modem” would disable the module’s hotspot, when is connected to the modem.

The Camera Configuration Section:

☒ Hidden HotSpot

☒ Disable HotSpot when is connected to the modem

Save

Video Streaming Settings

Size

Quality

Video Recording Settings

☐ Enable Recording

Record Time

Motion Sensivity

Save

Figure 5-Camera config

With “Enable Recording” button you can enable or disable the motion sensor. After the configuration click on save button and re-start the module.

Video Streaming Page

This page is at the root address (192.168.4.1 or the router’s given IP) and works as a CCTV camera and you can see the live video.

In case of placing a MicroSD card, with click or tapping on the stream video, the module takes a picture (the stream freezes for 2 seconds).

If you click or tap twice, the white LED of the ESP32-CAM turns on or toggles off.

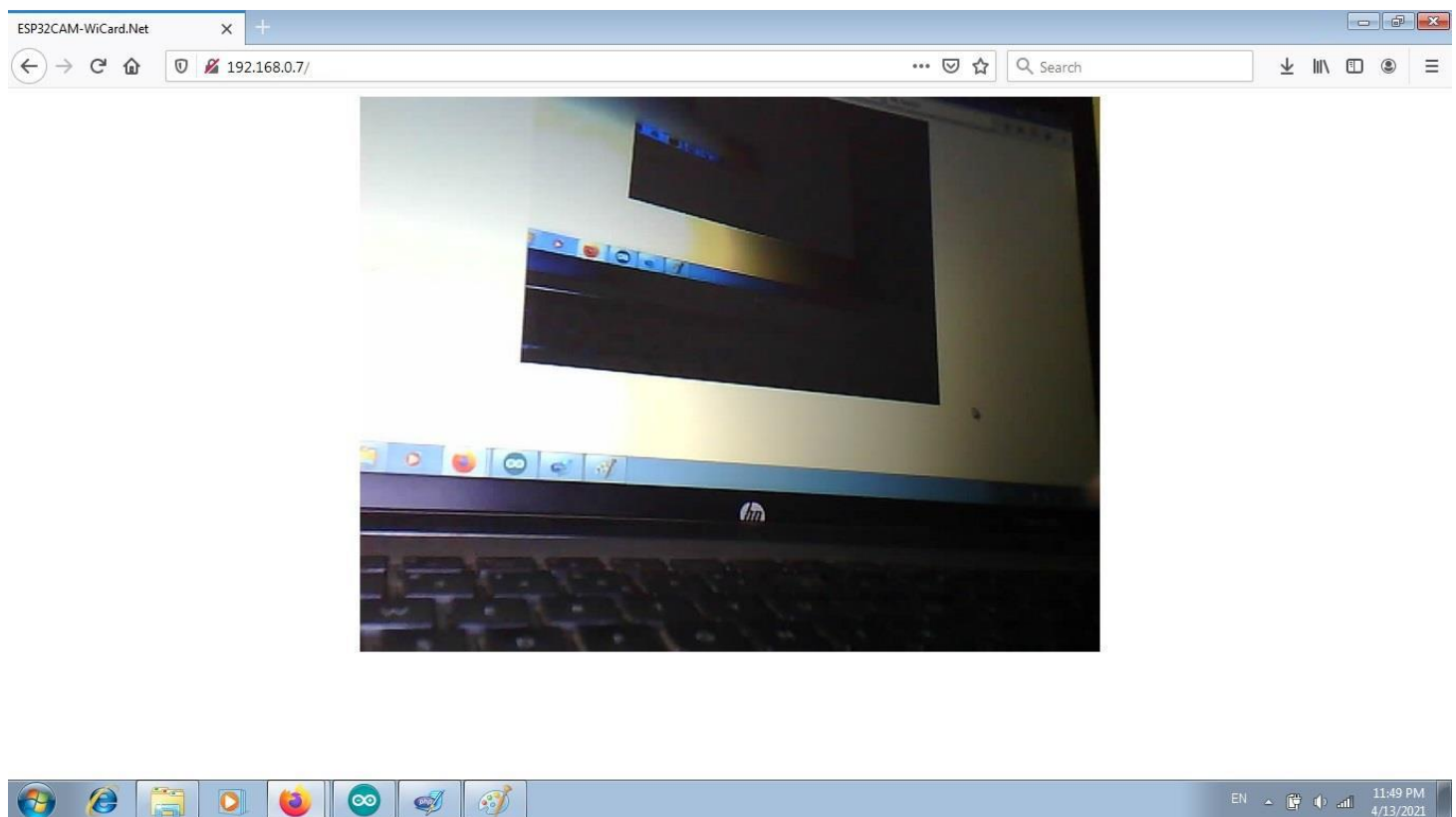


Figure 6-Video stream

- The light of scene increases images quality.
- Do not keep the white LED on for a long time.
- Better to use a suitable heatsink for the module’s regulator.
- Better to use a 5V-2A power source.
- The WiFi signal strength takes effect on the module capability and the video streams.

Memory page

The memory page is located at 192.168.4.1/memory (or with the DHCP IP) and shows the memory's images and videos.

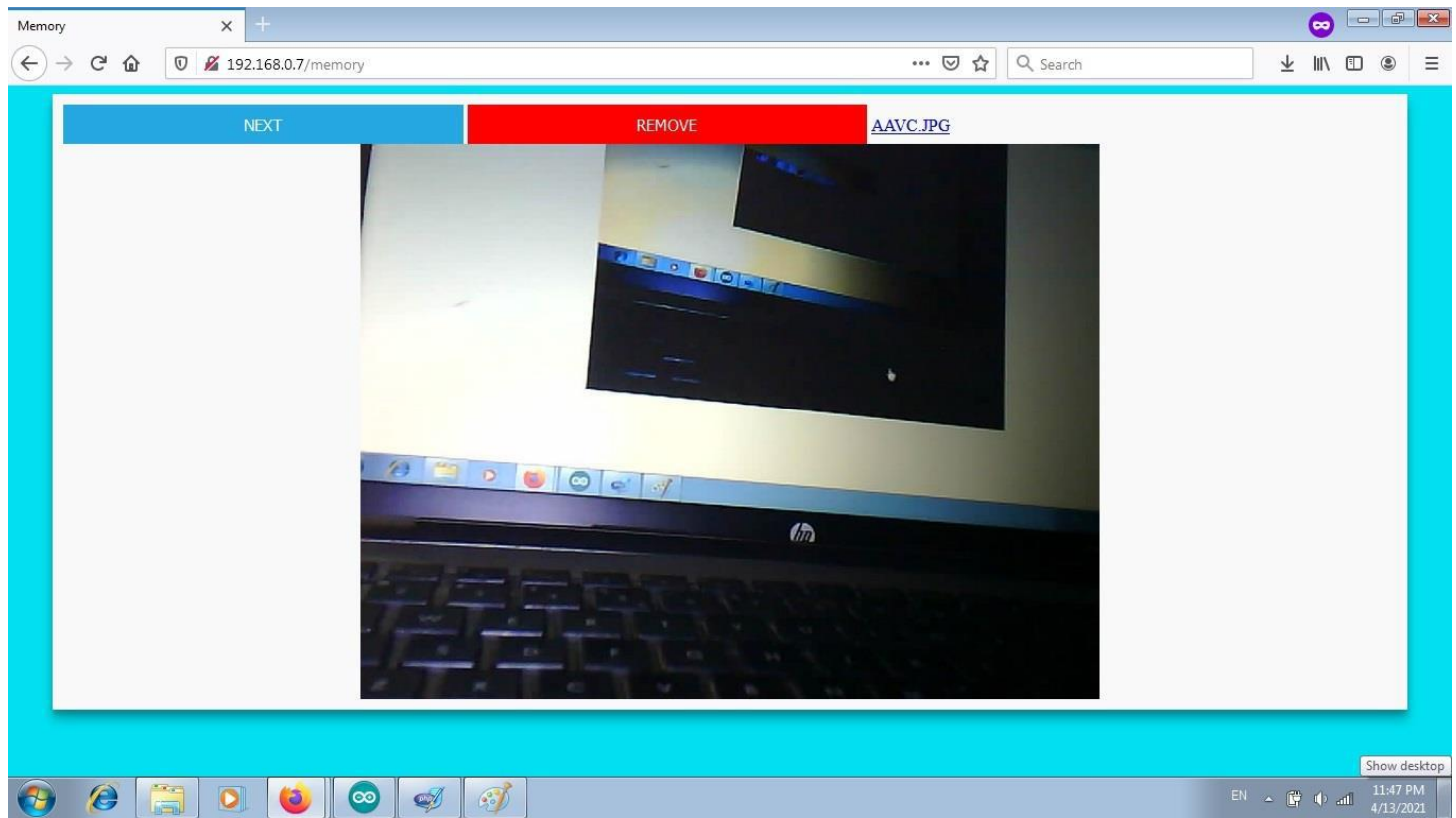


Figure 7-Memory manager

The video's format is .JP4 and only just the module can play them. With tap on the image of video, the video would play and pause.

Programming Manual

In case of some changes such as increasing the CPU frequency, it's necessary to make some changes in the program.

In the AC.h file, if you remove `"#define LOG_ENABLE"`, the logs will not be shown and the log memory would be released.

With add or remove `"#define UPDATE_IP"` you can enable or disable the server connection option.

The “LOOP_10_S” value set the printing “-” In the log time (in standby mode). The best time is 10 seconds and set according to your module CPU frequency.

The “LOOP_10_S_CAPTURING” value set the printing “-” In the log time (in streaming and recording mode). The best time is 10 seconds and set according to your module CPU frequency.

- Increasing the CPU frequency will increase the heat of module.
- This module has been tested with the original configuration not with more CPU frequency.
- Changing the configuration wrongly may hurt the module.

The code to changing the default image size in “ESP32CamProgram.ino”:

```
/* Size and Quality
FRAMESIZE_96x96, // 96x96
FRAMESIZE_QQVGA, // 160x120
FRAMESIZE_QQVGA2, // 128x160
FRAMESIZE_QCIF, // 176x144
FRAMESIZE_HQVGA, // 240x176
FRAMESIZE_240x240, // 240x240
FRAMESIZE_QVGA, // 320x240
FRAMESIZE_CIF, // 400x296
FRAMESIZE_VGA, // 640x480
FRAMESIZE_SVGA, // 800x600
FRAMESIZE_XGA, // 1024x768
FRAMESIZE_SXGA, // 1280x1024
FRAMESIZE_UXGA, // 1600x1200
FRAMESIZE_QXGA, // 2048*1536
*/
if(ucVidSize == 6)
config.frame_size = FRAMESIZE_QVGA;
else if(ucVidSize == 7)
config.frame_size = FRAMESIZE_CIF;
else if(ucVidSize == 8)
config.frame_size = FRAMESIZE_VGA;
else //if(ucVidSize == 9)
config.frame_size = FRAMESIZE_SVGA;
```

- The process of smaller images is faster.
- If you have not enough programming and electronics knowledge, do not make changes

Revision History

1.1

- Motion Sensor
- Recoding videos
- Playing videos and images
- Camera configuration
- Server option
- Internal pages secure link
- Server script
- Shop MAC and IP in config page

1.0

- Video stream
- Saving the images
- LED toggle on/off
- Set module hotspot and modem's ssid and password
- Automatically connection to the modem
- Config page
- Hidden hotspot button
- Disable hotspot button

Project Update

To receive the latest update for this project, please refer to our website:

[ESP32-CAM Arduino Source Code \(CCTV Video Recorder and IP Camera\)](#)

<https://wocard.net>