Internetof Things (IoT) Based Video Streaming

Khin Thida Aye
University of Computer Studies Yangon
khinthidaaye@ucsy.edu.mm

Khine Myint Zu
University of Computer Studies Yangon
khinemyintzu@ucsy.edu.mm

Abstract

Web server camera is a digital camera that sending the streaming imageusing the local WiFi access. They are compact and cost effective so that they are used in home security.No wire connection between camera module and monitor is needed but they require only WiFi access. In this research, the camera is made with the ESP32-CAM board which is included WiFi. ESP32-S chip, the OV2640 camera, a microSD card are combined in ESP32 –cam module. All of GPIO pins of ESP32 are not used in ESP32-cam but some GPIO pins are used for connection to OV2640 camera and microSD card. The ESP32 camera transmits a video streaming to web server that share videoto any users in local WiFiaccess .

1. Introduction

The video streaming is the continuous transmission of video file via a network or the internet to a remote user. The user can visualize video online without downloading them onto a host computer or smart phone. During the video streaming process, video file data is compressed and sent to the requesting device such as PC or phone via a nternet connection. The user's device decompresses the streamed video data and displays on the screen without waiting for the entire file to download.

The construction of an Internet Protocol camera (IP camera) is based on internet access.OV2640 camera is attached to the ESP32 module which composes of wifi and Bluetooth for internet connection. This combination is named as ESP32_CAM. There is no FDTI in the ESP32_ CAMso that it has no USB. ESP32 _CAM is programmed by using the external FDTI board but Arduinouno is used to program uploading in this research. There are many camera models used in ESP32_CAM module. Among them, AIthinker model is used.The camera can be used as photo shooting and video streaming.

2. Materials and Method

2.1.ESP32_CAM

The ESP32 is the combination of microcontroller and wireless system such as WiFi or blue tooth. It consists of a 32-bit microprocessor in dual-core ,GPIO pins, several kinds of memory, peripheral connectivity with many interface facilities. Sinceits clock speed is operating up to 240 MHz, the video streaming is quite smooth. There are 520 KB SRAM in memory and Wi-Fi ,Bluetooth v4.2 and bluetooth low energy (BLE) for wireless connectivity.[1]

The architecture of ESP32 is illustrated in figure . There are two CPU cores that can be individually controlled, and the clock frequency is adjustable from 80 MHz to 240 MHz. The user may also power off the CPU and make use of the low-power co-processor to constantly monitor the peripherals for changes or crossing of thresholds. ESP32 integrates a set of peripherals, ranging from capacitive touch sensors, Hall sensors, SD card interface, Ethernet, high-speed serial peripheral interface (SPI), universal asynchronous receiver and transmitter (UART), inter-integrated sound (I2S) and inter- integrated circuit (I2C). Using Wi-Fi , a large physical range and direct connection to the internet through a Wi-Fi router can be implemented, while using Bluetooth allows the user to conveniently connect to the phone or broadcast low energy beacons for its detection. The sleep current of the ESP32 chip is less than 5 µA. ESP32 supports a data rate of up to 150 Mbps, and 20.5 dBm output power at the antenna to ensure the widest physical range. ESP 32 consists of eighteen ADC channels, two DAC channels , two SPI , one I2C and two UART.

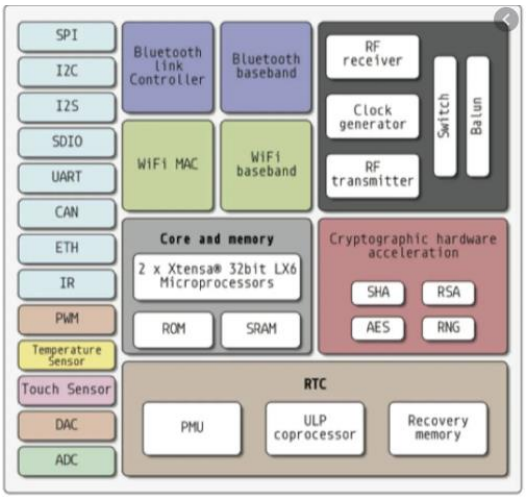


Figure. 1 Architecture of ESP32

The internal connection between ESP32 and AI thinker model Camera is as follows;

OV2640 CAMERA	ESP32
D0	GPIO 5
D1	GPIO 18
D2	GPIO 19
D3	GPIO 21
D4	GPIO 36
D5	GPIO 39
D6	GPIO 34
D7	GPIO 35
XCLK	GPIO 0
PCLK	GPIO 22
VSYNC	GPIO 25
HREF	GPIO 23
SDA	GPIO 26
SCL	GPIO 27
POWER PIN	GPIO 32

ESP32_CAM shown in Figure 2 is an essentia component in this research.Arduinouno is functione as the programmer.

Figure 2. ESP32_CAM and its pin diagram

2.2. Software preparation

ESP32_CAM can be operated by using Arduino IDE but ESP32 board is not included in default Arduino IDE.[2] ESP32 board is included in Arduino IDE as follows;

1. In Arduino IDE, open “file” and then “ Preferences”
2. Fill” https://dl.espressif.com/dl/package_esp32_index.json” into“Additional Board Manager URLs”as shown in Figure 3. And press OK.

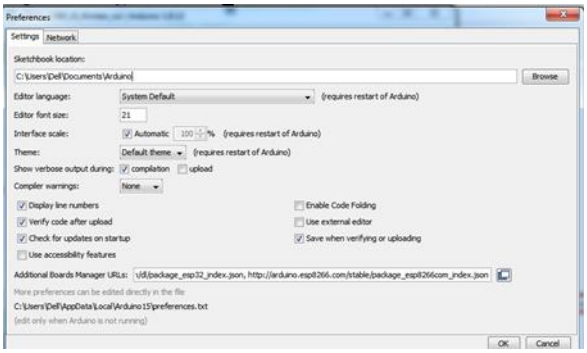


Figure 3. Installing ESP32 in Arduino IDE

2.3.Programming to ESP32_CAM Using Arduino

EP32_CAM is programmed via Arduino which is used as programmer.The connection of ESP32_CAM and Arduinounois completed as follows;

- Connect Arduino 5volt to ESP 32 _CAM 5 volt.
- Arduino GND to GND
- Arduino RX to RX of ESP32_CAM
- Arduino TX to TX of ESP32_CAM
- Arduino reset pin to GND
- ESP 32cam D0 to Gnd

The circuit diagram to program ESP32 _CAM is as shown in Figure 4.

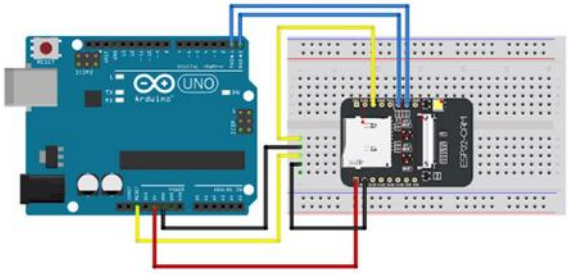


Figure 4. Programming setting by Arduino

The librariesof "esp_camera”,”esp_timer.h", “,”,“WiFi”,""img_converters", "Arduino","fb_gfx. "soc/soc.h" "soc/rtc_cntl_reg.h" and "esp_http_server.h" are required for programming of ESP32 _CAM video streaming

The network credentials of SSID and password are filled up in program properly. If it is not correct, the web server cannot be connected.

There are four camera models in ESP32_cam. They are AI_THINKER model, M5STACK_PSRAM model, M5STACK_WITHOUT_PSRAM model and WROVER_KIT model. Each model has their own different circuit connection. In this research , AI_ THINKER model is used , the circuit configuration is included in the program.

3. System Operation

The block diagram of IoT based video streaming system is as shown in Figure 5.

Figure 5. Block diagram of System operation

After connection of ESP32_CAM to Arduinouno,program is uploaded as following procedures;

1) Tools > Board andselect AI-Thinker ESP32-CAM.in figure 6.
2) Open again Tools , then Port and choose the right COM port which the ESP32 is connected to.
3) start the uploading
4) press the ESP32-CAM on-board RST button after seeing dots on debugging window
5) disconnect GPIO 0 from ground after uploading. Then, serial monitor is opened and IP address can be obtained as in Figure. 7.

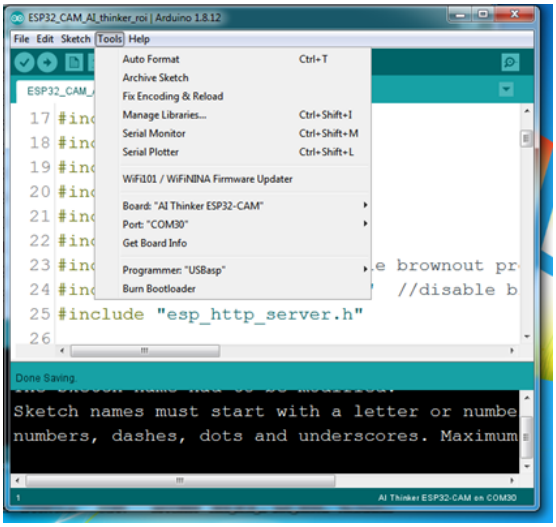


Figure 6. Selection board

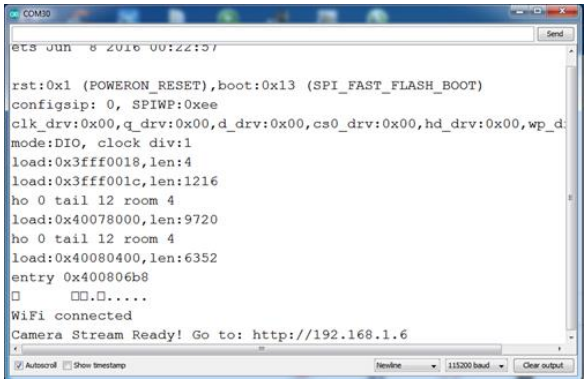


Figure 7. Obtaining IP address

4. Results

For photo capture and video stream, a browser is opened and the ESP32-CAM IP address of http://192.168.1.6 is typed. The video streaming on the PC is illustrated in Figure. 8. The video streaming on smart phone is shown in Figure 9 and Figure 10.

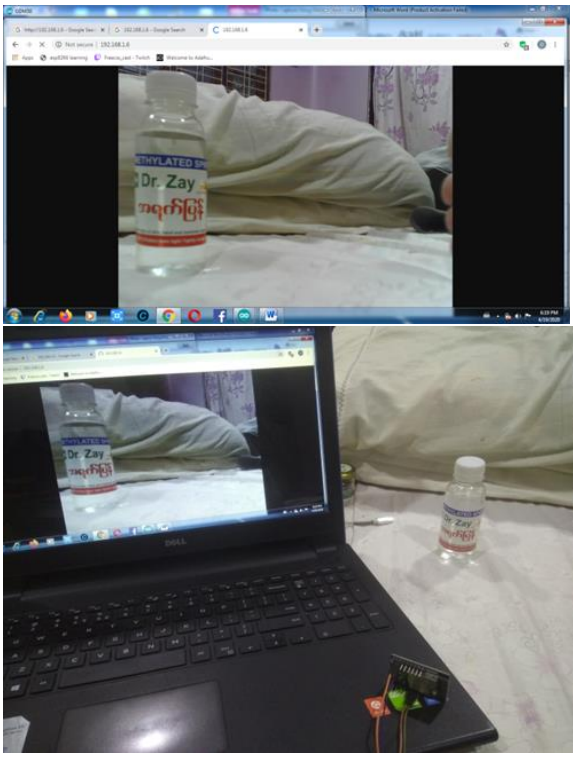
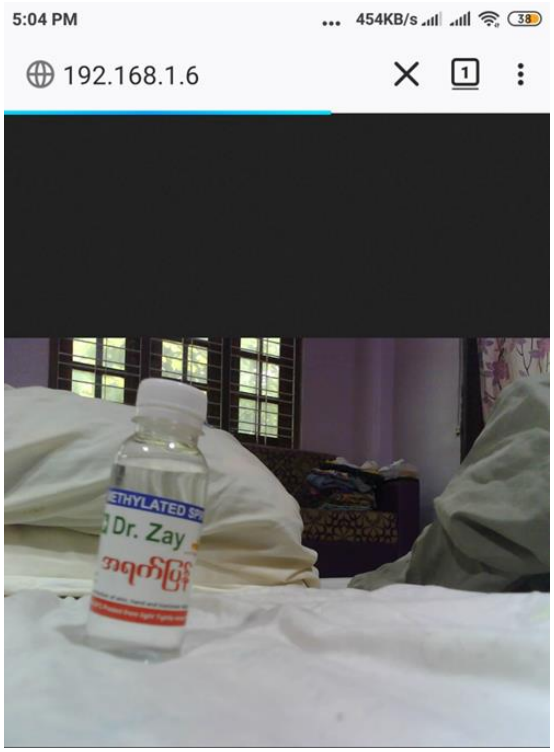


Figure 8. Video streaming the near object on PC



Figur 9. Video streaming the near object on Phone

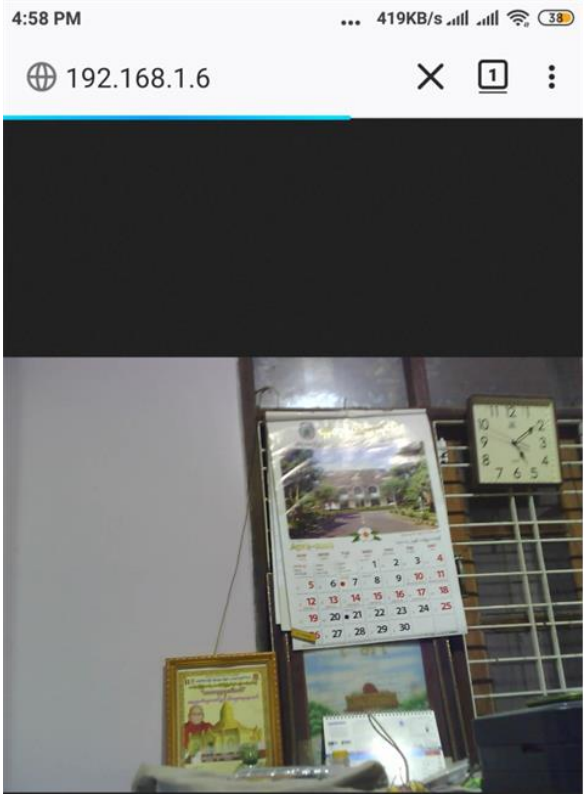


Figure 10. Video streaming the distant object on Phone

5. Discussion

There are many troubles shooting in this implementation. The "Timed out waiting for packet header" is occurred as shown in Figure 9.

This error is occurred when the circuit connection between ESP32_CAM and Arduinouno is not contacted properly.

Figure 10. "Timed out waiting"error

"Camera init failed with error 0x20001" error is displayed on Arduino window as shown in Figure 11. It is because the camera has tiny connector to contact ESP32 connector and they are not connected properly. Another is wrong camera model.

Sometimes, there is no IP address in Arduino IDE when the WiFi strength is weak or ESP32-CAM cannot connect a Wi-Fi router.

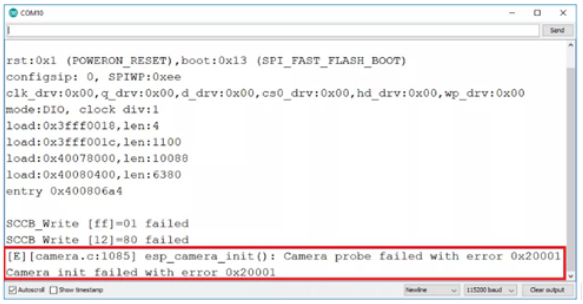


Figure. 11 "Camera init fail" error

ESP32 consists of 3.3V regulator so that the 5V power supply is select. Before the program is uploaded, GPIO 0 is connected to ground to be able to upload. After the program is already uploaded , GPIO 0 is disconnected from ground and reset button is pressed to obtain the IP address on serial monitor.[3]

6. Conclusion

Camera setting such as resolution, quality, brightness, contrast, saturation and etc.can be adjustable. The quality number can be varied 0 to 63. Lower number is best quality. When the quality number and resolution (frame size) are set 1 and UXGA(1600 × 1200) respectively, the images are cut

in half with the strange color. If the quality number and theresolution are set 10 and SVGA(800 × 600), the good photo frames and video streaming are visualized. It is found that the video streaming is trouble with the lower quality number and resolution higher than SVGA (800 × 600). The frame rate is 30fps for VGA and SVGA frame size. As PSRAM is used for buffering images from the camera, the video streaming can be used without crashing the ESP32. Using optimum quality number and resolution ,the video streaming yields the reliable quality of service(QoS). ESP32 _CAM support the video streaming to only one client at a time.

Web server camera can be integratedtohome securitysystem.ESP32_CAM can be used in capturing photo, video streaming, face recognition, trap camera with motion sensor and home security. Its cost is lower than the CCTV camera. Since it works with internet access, it has internet expenditure. But ,it is cost effective surveillance camera and it produces the reliable sharp images.

Acknowledgements

I would like to thank Rector Dr Mie Mie Thet Thwin, University of Computer Studies, Yangon, for her kind permission and encouragement to do this research.

References

[1] RuiSantos,"LearnESP32 withArduino IDE" Random Nerd Tutorial Lab,2020
[2] Purdun J, "Beginning C for Arduino" Apress, NY, 2012
[3] RuiSantos,"ESP 32 troublesshootingguide"Random Nerd Tutorial Lab,2020