REF: B11-ESP32-CAM2

ESP32 Camera with USB Connectivity

Description

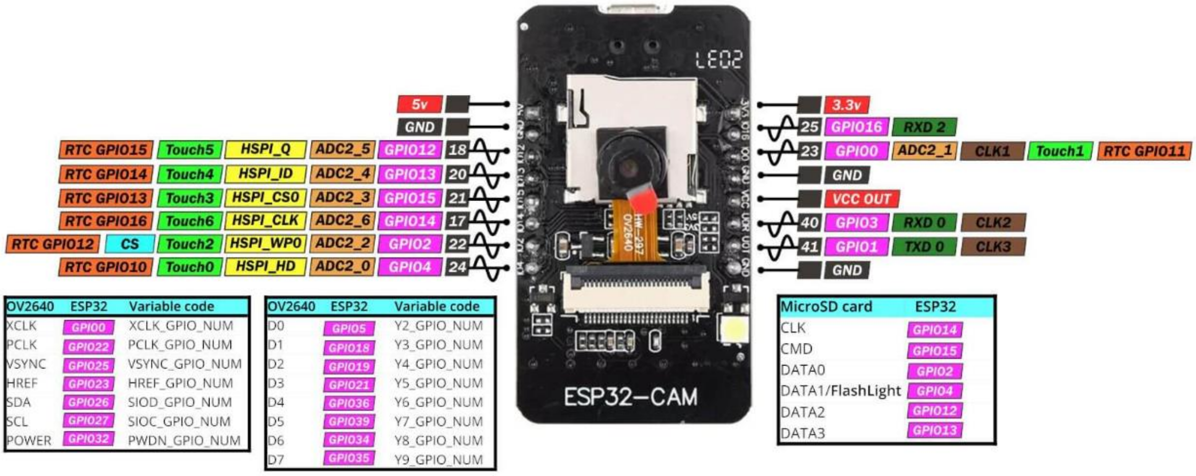
The ESP32-S contains a Tensilica Xtensa® LX6 microprocessor with two 32-bit cores operating at a staggering 240 MHz! This is what makes the ESP32-S suitable for intensive tasks like video processing, facial recognition, and even artificial intelligence. ESP32-CAM is an ESP32 Wireless IoT Vision Development Board in a tiny form factor, designed to be used in various IoT projects, such as home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, and so on. It supports deep sleep mode with power consumption as low as 6mA at 5V, making it ideal for portable IoT applications. It's suitable as a low-cost CCTV camera.

Specifications

- The smallest 802.11b/g/n/e/i Wi-Fi BT SoC Module
- Working Voltage: 5V
- Package: DIP-16
- SPI Flash: Default 32Mbit
- Built-in 520KB SRAM and 4M PSRAM
- Supports UART/SPI/PWM/I2C
- Comes with OV2640 Camera (can also support OV7670 cameras)
- Built-in Flash lamp.
- Support micro-SD card (max 4GB)
- Supports multiple sleep modes

ESP32-CAM Pinout

- Power Pins There are two power pins: 5V and 3V3. The ESP32-CAM can be powered via the 3.3V or 5V pins. Since many users have reported problems when powering the device with 3.3V, it is advised that the ESP32-CAM always be powered via the 5V pin. The VCC pin normally outputs 3.3V from the on-board voltage regulator. It can, however, be configured to output 5V by using the Zero-ohm link near the VCC pin.
- GND is the ground pin.
- GPIO Pins The ESP32-S chip has 32 GPIO pins in total, but because many of them are used internally for the camera and the PSRAM, the ESP32-CAM only has 10 GPIO pins available. These pins can be assigned a variety of peripheral duties, such as UART, SPI, ADC, and Touch.
- UART Pins The ESP32-S chip actually has two UART interfaces, UART0 and UART2. However, only the RX pin (GPIO 16) of UART2 is broken out, making UART0 the only usable UART on the ESP32-CAM (GPIO 1 and GPIO 3). Also, because the ESP32-CAM lacks a USB port, these pins must be used for flashing as well as connecting to UART-devices such as GPS, fingerprint sensors, distance sensors, and so on.
- MicroSD Card Pins are used for interfacing the microSD card. If you aren't using a microSD card, you can use these pins as regular inputs and outputs.
- ADC Pins On the ESP32-CAM, only ADC2 pins are broken out. However, because ADC2 pins are used internally by the WiFi driver, they cannot be used when Wi-Fi is enabled.
- Touch Pins The ESP32-CAM has 7 capacitive touch-sensing GPIOs. When a capacitive load (such as a human finger) is in close proximity to the GPIO, the ESP32 detects the change in capacitance.
- SPI Pins The ESP32-CAM features only one SPI (VSPI) in slave and master modes.
- PWM Pins The ESP32-CAM has 10 channels (all GPIO pins) of PWM pins controlled by a PWM controller. The PWM output can be used for driving digital motors and LEDs.


5v
GND
RTC GPIO15 Touch5 HSPI_Q ADC2_5 GPIO12 18
RTC GPIO14 Touch4 HSPI_ID ADC2_4 GPIO13 20
RTC GPIO13 Touch3 HSPI_CS0 ADC2_3 GPIO15 21
RTC GPIO16 Touch6 HSPI_CLK ADC2_6 GPIO14 17
RTC GPIO12 CS Touch2 HSPI_WP0 ADC2_2 GPIO2 22
RTC GPIO10 Touch0 HSPI_HD ADC2_0 GPIO4 24
3.3v
25 GPIO16 RXD 2
23 GPIO0 ADC2_1 CLK1 Touch1 RTC GPIO11
GND
VCC OUT
40 GPIO3 RXD 0 CLK2
41 GPIO1 TXD 0 CLK3
GND
ESP32-CAM

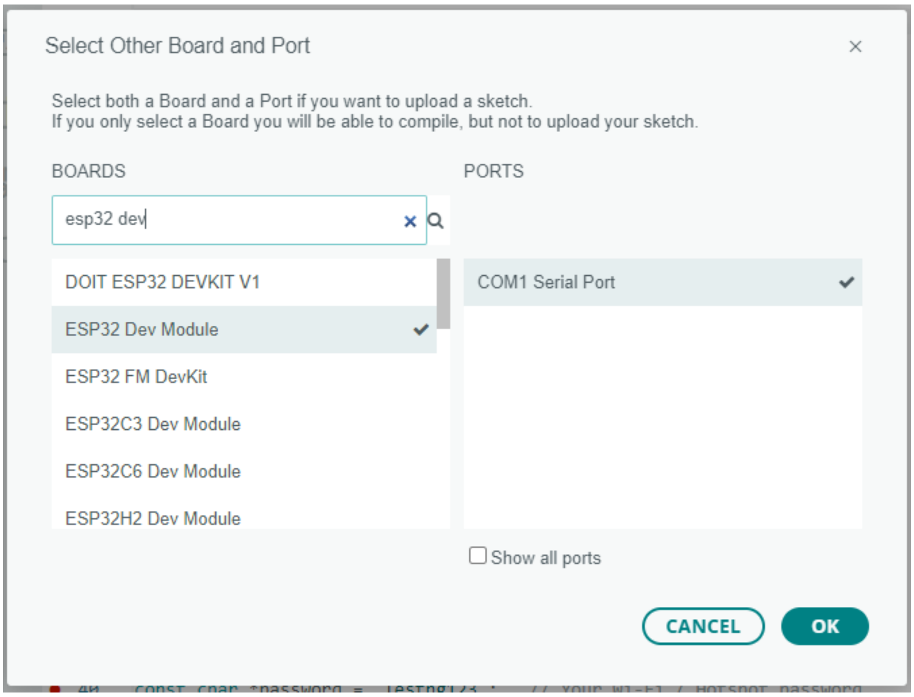
OV2640	ESP32	Variable code
XCLK	GPIO0	XCLK_GPIO_NUM
PCLK	GPIO22	PCLK_GPIO_NUM
VSYNC	GPIO25	VSYNC_GPIO_NUM
HREF	GPIO23	HREF_GPIO_NUM
SDA	GPIO26	SIOD_GPIO_NUM
SCL	GPIO27	SIOC_GPIO_NUM
POWER	GPIO32	PWDN_GPIO_NUM

OV2640	ESP32	Variable code
D0	GPIO5	Y2_GPIO_NUM
D1	GPIO18	Y3_GPIO_NUM
D2	GPIO19	Y4_GPIO_NUM
D3	GPIO21	Y5_GPIO_NUM
D4	GPIO36	Y6_GPIO_NUM
D5	GPIO39	Y7_GPIO_NUM
D6	GPIO34	Y8_GPIO_NUM
D7	GPIO35	Y9_GPIO_NUM

MicroSD card	ESP32
CLK	GPIO14
CMD	GPIO15
DATA0	GPIO2
DATA1/FlashLight	GPIO4
DATA2	GPIO12
DATA3	GPIO13

Coding

Select the ESP32 Dev Module from the board selection menu.



Select Other Board and Port

Select both a Board and a Port if you want to upload a sketch.
If you only select a Board you will be able to compile, but not to upload your sketch.

BOARDS

esp32 dev

- DOIT ESP32 DEVKIT V1
- ESP32 Dev Module ✓
- ESP32 FM DevKit
- ESP32C3 Dev Module
- ESP32C6 Dev Module
- ESP32H2 Dev Module

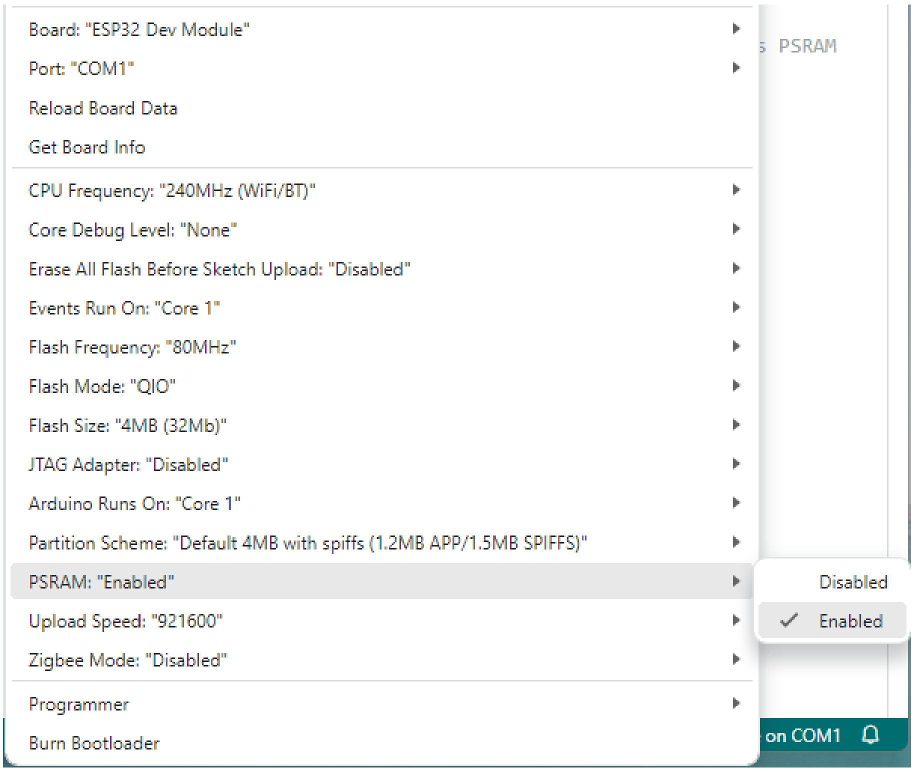
PORTS

- COM1 Serial Port ✓

□ Show all ports

CANCEL OK

Make sure to enable **PSRAM** option from board selection menu.

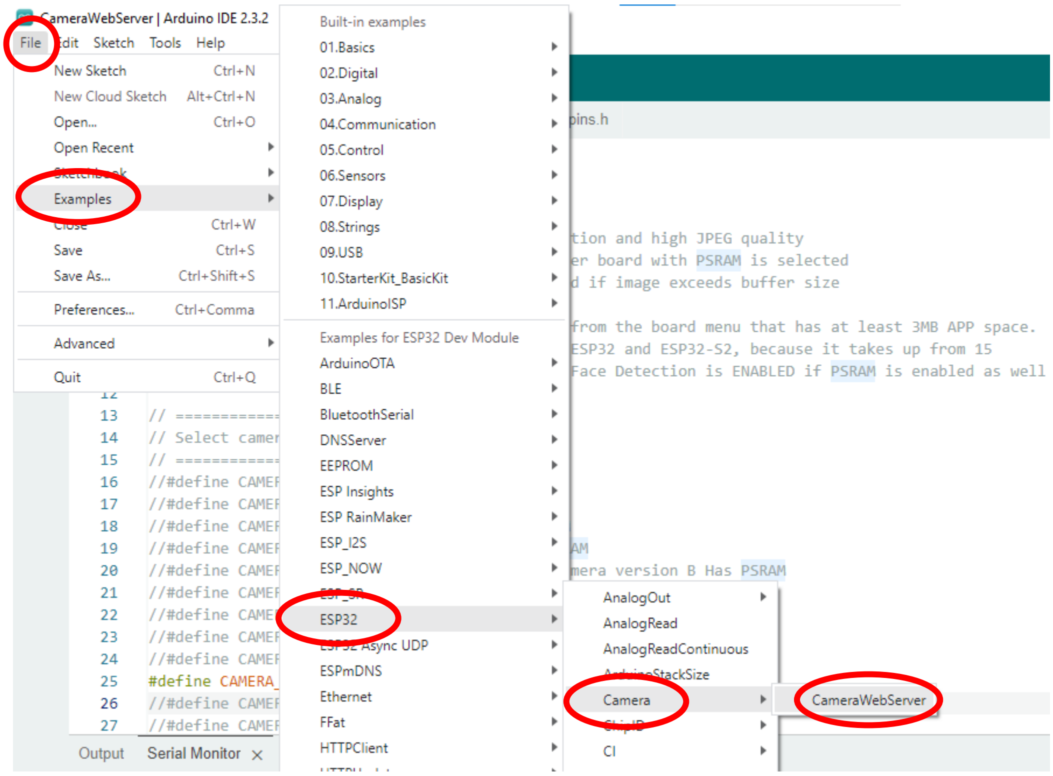


Board: "ESP32 Dev Module"
Port: "COM1"
Reload Board Data
Get Board Info
CPU Frequency: "240MHz (WiFi/BT)"
Core Debug Level: "None"
Erase All Flash Before Sketch Upload: "Disabled"
Events Run On: "Core 1"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
JTAG Adapter: "Disabled"
Arduino Runs On: "Core 1"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
PSRAM: "Enabled"
Upload Speed: "921600"
Zigbee Mode: "Disabled"
Programmer
Burn Bootloader

Disabled
✓ Enabled

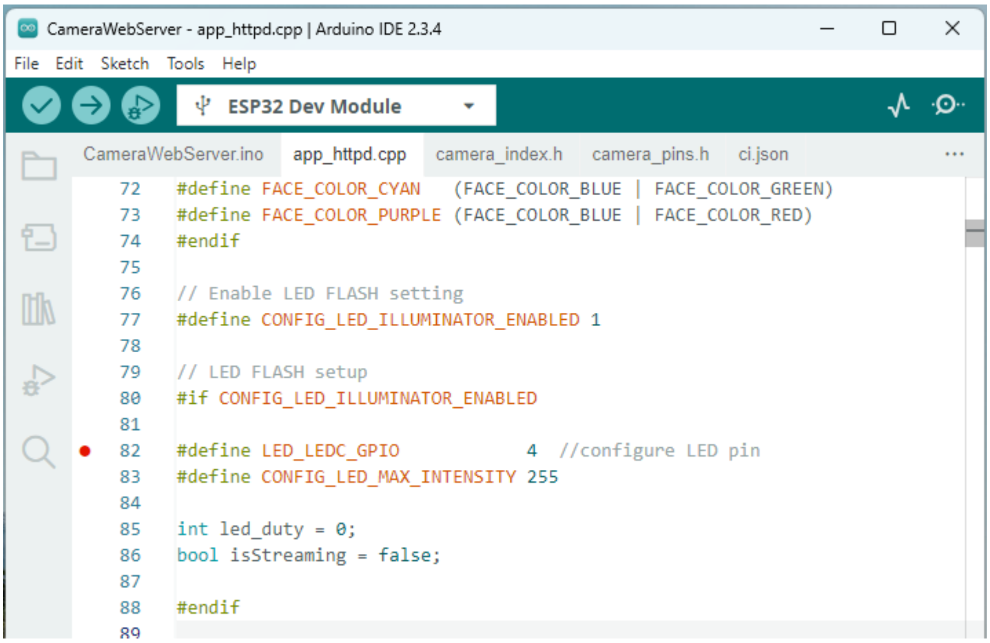
Open the sample code from Arduino IDE.

(File > Example > ESP32 > Camera > CameraWebServer)



On the **app_httpd.cpp** tab, change the LED pin to 4. (Line 82)

#define LED_LEDC_GPIO 4 //configure LED pin

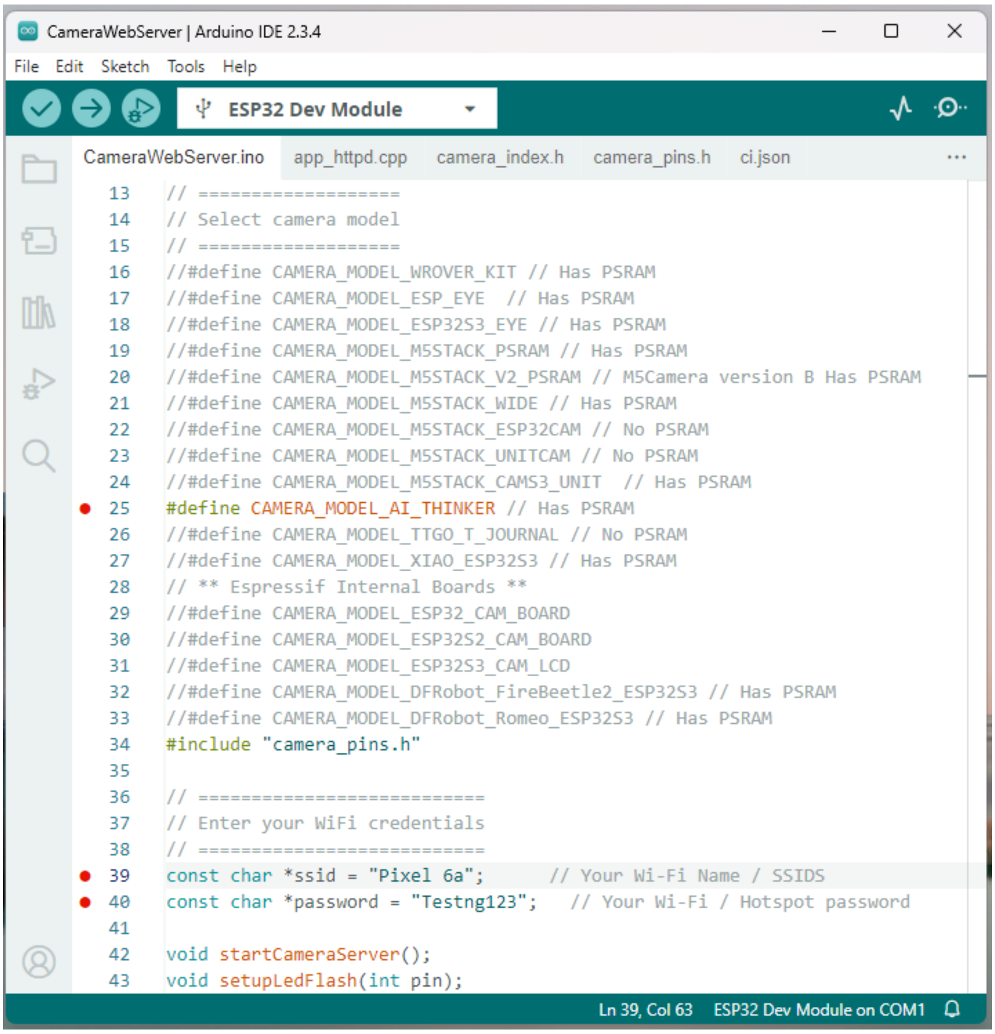


```
72  #define FACE_COLOR_CYAN   (FACE_COLOR_BLUE | FACE_COLOR_GREEN)
73  #define FACE_COLOR_PURPLE (FACE_COLOR_BLUE | FACE_COLOR_RED)
74  #endif
75
76  // Enable LED FLASH setting
77  #define CONFIG_LED_ILLUMINATOR_ENABLED 1
78
79  // LED FLASH setup
80  #if CONFIG_LED_ILLUMINATOR_ENABLED
81
82  #define LED_LEDC_GPIO            4  //configure LED pin
83  #define CONFIG_LED_MAX_INTENSITY 255
84
85  int led_duty = 0;
86  bool isStreaming = false;
87
88  #endif
89
```

On the **CameraWebServer.ino** tab, **add the "//" in front of line 17 (comment)** and **remove the "//" from line 25 (uncomment)**.

#define CAMERA_MODEL_AI_THINKER // Has PSRAM

Fill in **Wi-Fi** and **Password** details on line 39 & 40. Make sure to use the same Wi-Fi with the PC.



```
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_M5STACK_PSRAM // Has PSRAM
20  //#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
21  //#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
22  //#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
23  //#define CAMERA_MODEL_M5STACK_UNITCAM // No PSRAM
24  //#define CAMERA_MODEL_M5STACK_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  // ===========================
37  // Enter your WiFi credentials
38  // ===========================
39  const char *ssid = "Pixel 6a";      // Your Wi-Fi Name / SSIDS
40  const char *password = "Testng123";   // Your Wi-Fi / Hotspot password
41
42  void startCameraServer();
43  void setupLedFlash(int pin);
```

Ln 39, Col 63 ESP32 Dev Module on COM1

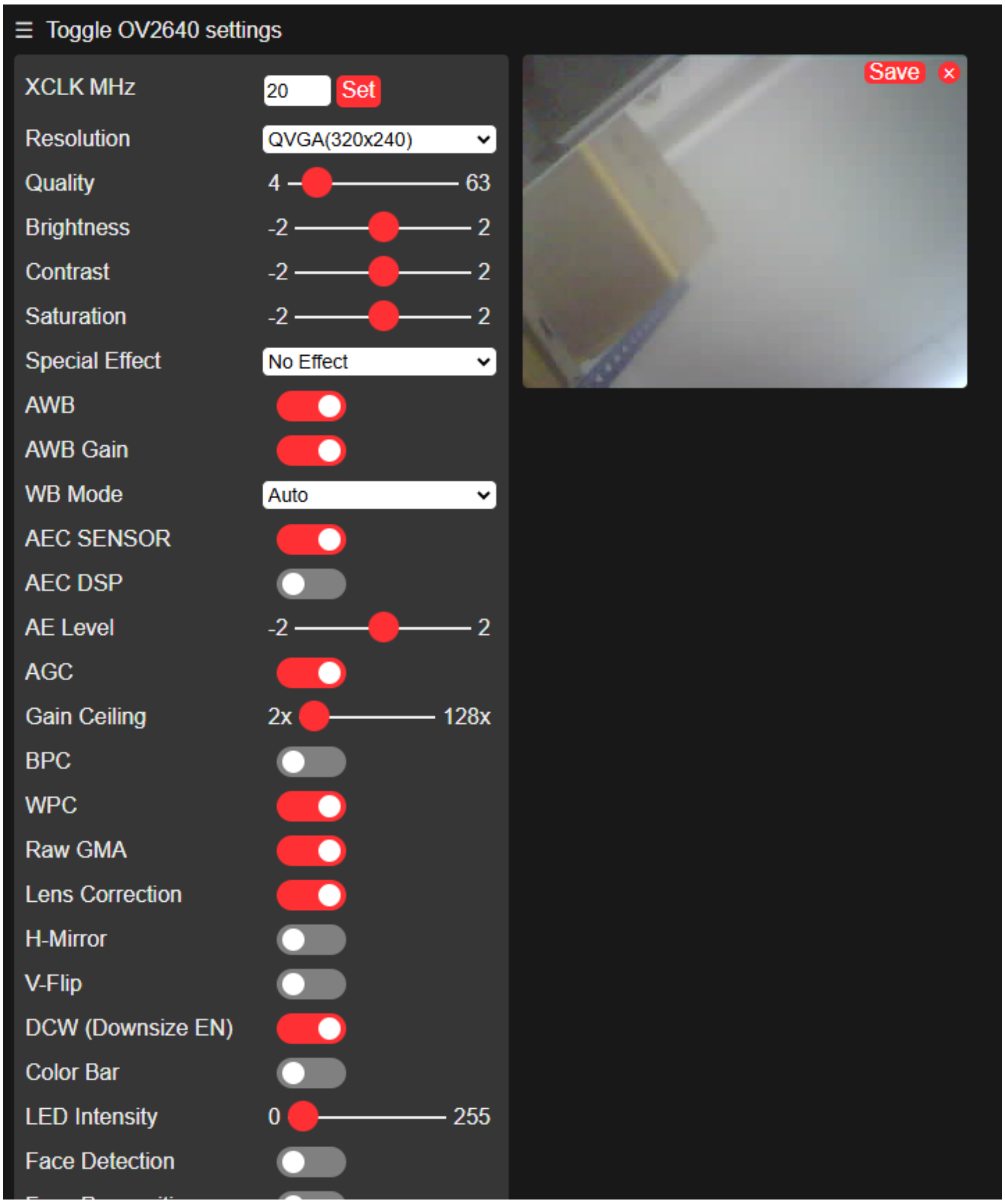
Result

Once you have uploaded the sketch, open the serial monitor at baud rate 115200 and press the Reset button on the ESP32-CAM. You should see the IP address in the Serial Monitor.

```
14:19:30.715 -> ...
14:19:31.699 -> WiFi connected
14:19:31.699 -> Camera Ready! Use 'http://192.168.1.44' to connect
```

To copy any text data from **Serial Monitor** user can **Highlight** the text & use **CTRL+C** since it **does not support mouse Right-Click**.

Launch a browser and enter the IP address shown on the serial monitor. Ensure that the web browser is on the same network that the ESP32-CAM is connected to the ESP32-CAM should display a web page. To begin video streaming, press the Start Stream button.



☰ Toggle OV2640 settings

XCLK MHz: 20 Set
Resolution: QVGA(320x240)
Quality: 4 ● 63
Brightness: -2 ● 2
Contrast: -2 ● 2
Saturation: -2 ● 2
Special Effect: No Effect
AWB: ●
AWB Gain: ●
WB Mode: Auto
AEC SENSOR: ●
AEC DSP: ○
AE Level: -2 ● 2
AGC: ●
Gain Ceiling: 2x ● 128x
BPC: ○
WPC: ●
Raw GMA: ●
Lens Correction: ●
H-Mirror: ○
V-Flip: ○
DCW (Downsize EN): ●
Color Bar: ○
LED Intensity: 0 ● 255
Face Detection: ○