

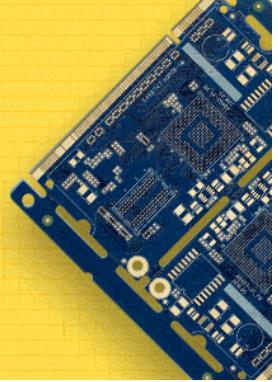
**NextPCB**

## Try Free PCB Prototype

**0\$ for 1-4 layer PCB Prototype**

- 1-4 layers, green mask, Fr-4
- Fast lead time as 24 hours

**Order Now**



ESP32-CAM Projects   IoT Projects

# ESP32 CAM Based Object Detection & Identification with OpenCV

Object Detection & Identification with ESP32 Camera & OpenCV



Admin   Last Updated: October 5, 2021   18   28,231   5 minutes read



Table of Contents [[hide](#)]

- 1 Overview: ESP32 CAM Based Object Detection & Identification
- 2 Bill of Materials
- 3 ESP32 CAM Module
- 4 ESP32-CAM FTDI Connection
- 5 Project PCB Gerber File & PCB Ordering Online
- 6 Installing ESP32CAM Library
- 7 Source Code/Program for ESP32 CAM Module
- 8 Python Library Installation
- 9 Python Code for ESP32 CAM Object Detection/Identification
- 10 Applications
- 11 Video Tutorial & Guide

## Overview: ESP32 CAM Based Object Detection & Identification

This tutorial introduces the topic of **ESP32 CAM Based Object Detection & Identification** with OpenCV. OpenCV is an open-sourced image processing library that is very widely used not just in industry but also in the field of research and development.

Here for object detection, we have used the [cvlib Library](#). The library uses a pre-trained AI model on the COCO dataset to detect objects. The name of the pre-trained model is YOLOv3.

In this tutorial, we will go through its features, pins description and the method to program ESP32 Camera Module using FTDI Module. We will also set up the Arduino IDE for the ESP32 Camera Module. We will also upload the **firmware** and then work on the **object detection & identification** part. The script for object detection is written in the python programming language, thus we will also have to **install Python** and its required **Libraries**.

In an earlier ESP32 CAM Based project we learned about **Face Detection System** & also **Color Detection System** using Python & OpenCV. This project also requires the use of **OpenCV** for Object Detection & Identification.

## O melhor da tecnologia

Experimente uma tecnologia que pode transformar o seu negócio

Google Cloud

## Bill of Materials

The following is the list of Bill of Materials for building an **ESP32 CAM Based Object Detection & Identification System**. The ESP32 CAM when combined with other **hardware & firmware** track and identify the object. You can purchase all these components from Amazon.

S.N.	COMPONENTS	DESCRIPTION	QUANTITY	
1	ESP32-CAM Board	AI-Thinker ESP32 Camera Module	1	<a href="https://amzn.to/3fJLWzD">https://amzn.to/3fJLWzD</a>
2	FTDI Module	USB-to-TTL Converter Module	1	<a href="https://amzn.to/3fJLWzD">https://amzn.to/3fJLWzD</a>
3	USB Cable	5V Mini-USB Data Cable	1	<a href="https://amzn.to/3fJLWzD">https://amzn.to/3fJLWzD</a>

S.N.	COMPONENTS	DESCRIPTION	QUANTITY	<a href="https://amzn.to/3LcOOGI">https://amzn.to/3LcOOGI</a>
4	Jumper Wires	Female to Female	10	
		Connectors		

## ESP32 CAM Module

The ESP32 Based Camera Module developed by **AI-Thinker**. The controller is based on a **32-bit CPU** & has a combined **Wi-Fi + Bluetooth/BLE Chip**. It has a built-in **520 KB SRAM** with an external **4M PSRAM**. Its **GPIO Pins** have support like **UART, SPI, I2C, PWM, ADC, and DAC**.



Front Side



Back Side

The module combines with the **OV2640 Camera Module** which has the highest Camera Resolution up to **1600 × 1200**. The camera connects to the ESP32 CAM Board using a 24 pins gold plated connector. The board supports an **SD Card** of up to **4GB**. The SD Card stores capture images.

To learn in detail about the ESP32 Camera Module you can refer to our previous [Getting Started Tutorial](#).

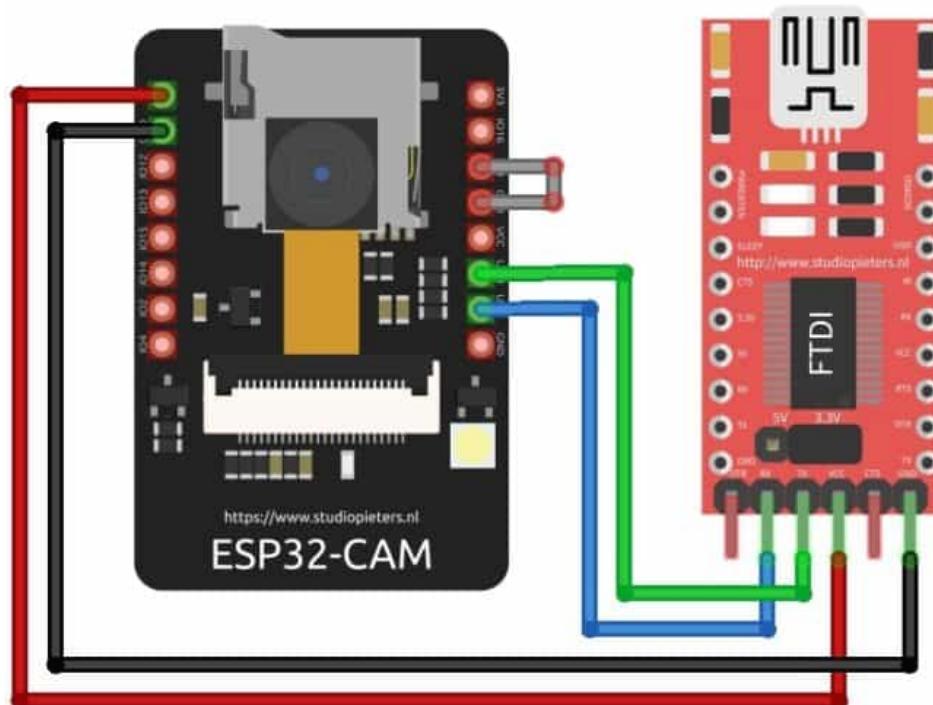
# O melhor da tecnologia

Google Cloud

## ESP32-CAM FTDI Connection

The board doesn't have a programmer chip. So In order to program this board, you can use any type of **USB-to-TTL Module**. There are so many **FTDI Module** available based on **CP2102** or **CP2104** Chip or any other chip.

Make a following connection between FTDI Module and ESP32 CAM module.



ESP32-CAM

FTDI Programmer

GND	GND
5V	VCC
U0R	TX
U0T	RX
GPIO0	GND

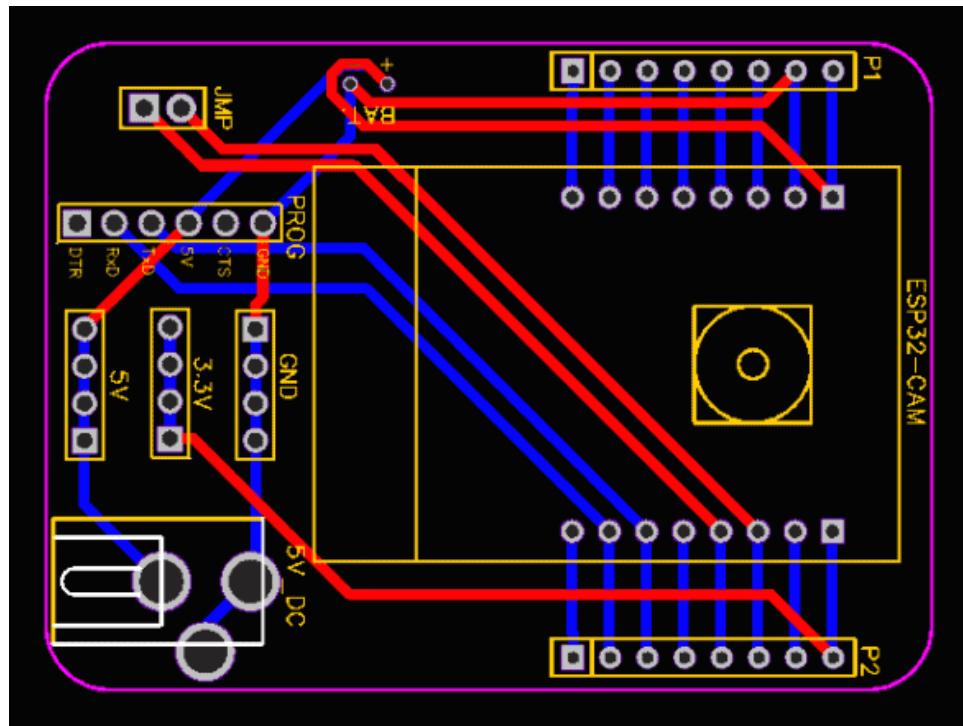
Connect the **5V** & **GND** Pin of ESP32 to 5V & GND of FTDI Module.

Similarly, connect the **Rx** to **UOT** and **Tx** to **UOR** Pin. And the most important thing, you need to short the **IO0** and **GND** Pin together.

This is to put the device in **programming mode**. Once programming is done you can remove it.

## Project PCB Gerber File & PCB Ordering Online

If you don't want to assemble the circuit on a breadboard and you want PCB for the project, then here is the PCB for you. The PCB Board for ESP32 CAM Board is designed using [EasyEDA](#) online Circuit Schematics & PCB designing tool. The PCB looks something like below.



The Gerber File for the PCB is given below. You can simply download the Gerber File and order the PCB from  
<https://www.nextpcb.com/>

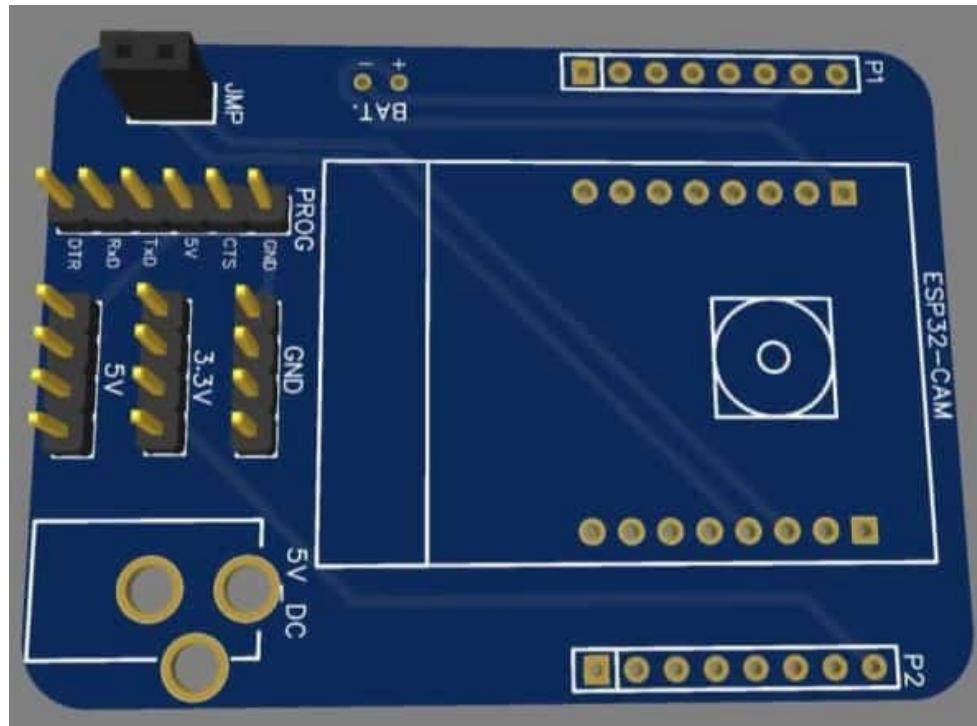
**Download Gerber File: [ESP32-CAM Multipurpose PCB](#)**

Now you can visit the NextPCB official website by clicking here:  
<https://www.nextpcb.com/>. So you will be directed to **NextPCB website.**



You can now upload the Gerber File to the Website and place an order. The PCB quality is superb & high. That is why most of the people trust NextPCB for **PCB & PCBA Services**.

You can assemble the components on the PCB Board.



## Installing ESP32CAM Library

Here we will not use the general **ESP webserver example** rather another streaming process. Therefore we need to add another **ESPCAM library**. The **esp32cam** library provides an object oriented API to use **OV2640 camera** on **ESP32 microcontroller**. It is a wrapper of **esp32-camera library**.

Go to the following [Github Link](#) and download the zip library as in the image

Once downloaded add this zip library to Arduino Libray Folder. To do so follow the following steps:

*Open Arduino -> Sketch -> Include Library -> Add .ZIP Library... -> Navigate to downloaded zip file -> add*

-25%

Milhões de Ofertas  
AliExpress

## Source Code/Program for ESP32 CAM Module

Here is a [source code](#) for Object Detection & Identification with ESP32 Camera & OpenCV. Copy the code and paste it in the

## Arduino IDE.

```
#include <WebServer.h>
#include <WiFi.h>
#include <esp32cam.h>

const char* WIFI_SSID = "ssid";
const char* WIFI_PASS = "password";

WebServer server(80);

static auto loRes = esp32cam::Resolution::find(320, 240);
static auto midRes = esp32cam::Resolution::find(350, 530);
static auto hiRes = esp32cam::Resolution::find(800, 600);
void serveJpg()
{
    auto frame = esp32cam::capture();
    if (frame == nullptr) {
        Serial.println("CAPTURE FAIL");
        server.send(503, "", "");
        return;
    }
    Serial.printf("CAPTURE OK %dx%d %db\n", frame->getWidth(),
                  static_cast<int>(frame->size()));

    server.setContentLength(frame->size());
    server.send(200, "image/jpeg");
    WiFiClient client = server.client();
    frame->writeTo(client);
}

void handleJpgLo()
{
    if (!esp32cam::Camera.changeResolution(loRes)) {
        Serial.println("SET-LO-RES FAIL");
    }
    serveJpg();
}

void handleJpgHi()
{
    if (!esp32cam::Camera.changeResolution(hiRes)) {
        Serial.println("SET-HI-RES FAIL");
    }
    serveJpg();
}
```

```
}

void handleJpgMid()
{
    if (!esp32cam::Camera.changeResolution(midRes)) {
        Serial.println("SET-MID-RES FAIL");
    }
    serveJpg();
}

void setup(){
    Serial.begin(115200);
    Serial.println();
    {
        using namespace esp32cam;
        Config cfg;
        cfg.setPins(pins::AiThinker);
        cfg.setResolution(hiRes);
        cfg.setBufferCount(2);
        cfg.setJpeg(80);

        bool ok = Camera.begin(cfg);
        Serial.println(ok ? "CAMERA OK" : "CAMERA FAIL");
    }
    WiFi.persistent(false);
    WiFi.mode(WIFI_STA);
    WiFi.begin(WIFI_SSID, WIFI_PASS);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
    }
    Serial.print("http://");
    Serial.println(WiFi.localIP());
    Serial.println(" /cam-lo.jpg");
    Serial.println(" /cam-hi.jpg");
    Serial.println(" /cam-mid.jpg");

    server.on("/cam-lo.jpg", handleJpgLo);
    server.on("/cam-hi.jpg", handleJpgHi);
    server.on("/cam-mid.jpg", handleJpgMid);

    server.begin();
}

void loop()
{
    server.handleClient();
}
```



Before Uploading the code you have to make a small change to the code. Change the **SSID** and **password** variable and in accordance with your WiFi network.

Now **compile** and **upload** it to the ESP32 CAM Board. But during uploading, you have to follow few steps every time.

- Make sure the **IO0 pin** is shorted with the ground when you have pressed the upload button.
- If you see the dots and dashes while uploading press the **reset button** immediately
- Once the code is uploaded, remove the IO1 pin shorting with Ground and press the reset button once again.
- If the output is the **Serial monitor** is still not there then press the reset button again.

Now you can see a similar output as in the image below.

```

ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
flash read err, 1000
ets_main.c 371
ets Jun  8 2016 00:22:57

rst:0x10 (RTCWDT_RTC_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:10944
load:0x40080400,len:6388
entry 0x400806b4

CAMERA OK
http://192.168.1.61
  /cam-lo.jpg
  /cam-hi.jpg
  /cam-mid.jpg

```

Autoscroll  Show timestamp      Newline      115200 baud      Clear output

Here, copy the IP address visible, we will be using it to edit the URL in python code

## Python Library Installation

For the live stream of video to be visible on our computer we need to write a **Python script** that will enable us to retrieve the **frames of the video**. The first step is to install Python. Go to [python.org](https://www.python.org) and download Python.

Once download, install Python.

Then Go to the command prompt and install **NumPy**, **OpenCV** and **cvglib** libraries.

- type: **pip install numpy** and press enter. After the installation is done.
- type: **pip install opencv-python** and press enter.
- type: **pip install cvlib** and press enter, close the command prompt.

In our python code we have used **urllib.request** to retrieve the frames from the URL and the library for **image processing** is OpenCV. For Object detection, we have used the **Cvlib library** that uses an **AI model** for detecting objects. Since the whole process requires a good amount of processing power, thus we have used **multiprocessing** which utilizes multiple cores of our CPU.

## Python Code for ESP32 CAM Object Detection/Identification

Now open **Idle code editor** or any other **python code editor**.

Copy and paste the code from below and do the replacements as mentioned below.

```
import cv2
import matplotlib.pyplot as plt
import cvlib as cv
import urllib.request
import numpy as np
from cvlib.object_detection import draw_bbox
import concurrent.futures

url='http://192.168.10.162/cam-hi.jpg'
im=None

def run1():
    cv2.namedWindow("live transmission", cv2.WINDOW_AUTOSIZE)
    while True:
        img_resp=urllib.request.urlopen(url)
        imgnp=np.array(bytarray(img_resp.read()), dtype=np.u
        im = cv2.imdecode(imgnp, -1)

        cv2.imshow('live transmission',im)
        key=cv2.waitKey(5)
        if key==ord('q'):
            break

    cv2.destroyAllWindows()

def run2():
```

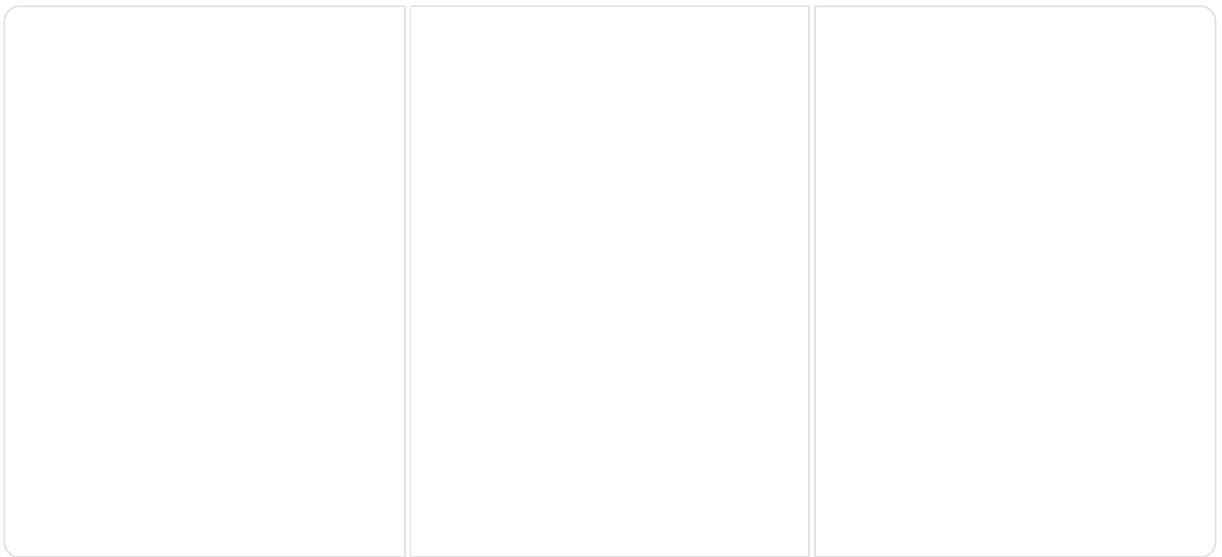
```
cv2.namedWindow("detection", cv2.WINDOW_AUTOSIZE)
while True:
    img_resp=urllib.request.urlopen(url)
    imgnp=np.array(bytarray(img_resp.read()),dtype=np.u
    im = cv2.imdecode(imgnp, -1)

    bbox, label, conf = cv.detect_common_objects(im)
    im = draw_bbox(im, bbox, label, conf)

    cv2.imshow('detection',im)
    key=cv2.waitKey(5)
    if key==ord('q'):
        break

cv2.destroyAllWindows()

if __name__ == '__main__':
    print("started")
    with concurrent.futures.ProcessPoolExecutor() as execute
        f1= execute.submit(run1)
        f2= execute.submit(run2)
```



Here we have to replace the **IP address** with the IP on Arduino Serial Monitor. For the first time, it will install a few files if they are not existing.

Once we have done that we can see two windows named **live transmission** and **detected** is visible.

Now in the detected window, one can view different detected objects as around them different **colored boxes** are visible.

## Applications

Object detection is having uses in almost all sorts of industries. It is used for tracking objects, people counting, automated CCTV surveillance, vehicle detection, etc.

These are just some basic examples but in reality, the potential is tremendous.

## Video Tutorial & Guide

Object Detection & Identification using ESP32 CAM Module & OpenCV

Watch this video [on YouTube](#).

## 18 Comments

---

## Markus Togi Fedrian Rivaldi Sinaga

November 8, 2021 at 9:52 PM

Excuse mr / ms

Sorry for my bad English

I'm confusing with teh FTDI? Is it used as the programmable microprocessor?

I'm making school projecs, and I want to use your project as refference.

I don't have the FTDI module, but I have ESP32 Development Board (the one without the camera) separated with the ESP32 Cam, can my ESP32 Development Board substitute the FTDI?

I'll be so grateful if you can help me by replying via my email.

Thanks mr/ms.

Loading...

Reply

---

## Akash

November 10, 2021 at 8:40 PM

Hello. So basically the FTDI allows you to connect the ESP32(the programmable board) to your PC so that you can upload you board. You use it to transfer your code onto your ESP32. There are also other devices you can use to connect you microprocessor onto the board. For example, a TTL UART

Loading...

Reply

---

## some\_guy

December 4, 2021 at 8:07 AM

it runs, but the object detection window isn't responding or showing anything. i've done everything in the vid and this guide, even downloaded cudart64\_110, what's happening? what did i do wrong?

Loading...

Reply

---



## Fernando Carvalho

December 19, 2021 at 3:16 AM

I have the same problem

Loading...

Reply

---



## Amir\_Eshaqy

December 24, 2021 at 4:35 PM

hi,

I have this problem in executing Python code, please help  
Traceback (most recent call last):

File "C:\Users\Amir\_Eshaqy\Desktop\sss.py", line 3, in  
import cvlib as cv

File

"C:\Users\Amir\_Eshaqy\AppData\Local\Programs\Python\Python310\lib  
\site-packages\cvlib\\_init\_.py", line 8, in  
from .gender\_detection import detect\_gender

File

"C:\Users\Amir\_Eshaqy\AppData\Local\Programs\Python\Python310\lib  
\site-packages\cvlib\gender\_detection.py", line 3, in  
from tensorflow.keras.utils import get\_file

ModuleNotFoundError: No module named 'tensorflow'

Thank you

Loading...

Reply

---



## Cemil

January 20, 2022 at 6:49 PM

It about the version of Python. Install the 3.9 and install the tensorflow library.

Loading...

Reply

---

 **Chris Ellingson**

January 29, 2022 at 2:40 PM

Thank you for the informative write-up.

I'm waiting for my ESP32 CAM modules to ship.

Can this detection system distinguish between different animals of the same species? I am working on a remotely-operated tick treatment sprayer to dose the bush buck living on the smallholding. There are several of them and they are individually distinct enough in coloration and size.

I would like to automate the process as the bush buck tend to sporadically pass through a gate in the early morning and evenings when I am not around to control the sprayer. Having a system that can identify each animal and dose appropriate amounts of tick treatment would help massively to reduce the tick load in the bush. Having it avoid spraying people and young bush buck automatically would be a benefit too.

Loading...

---

Reply

**al06825AL**

February 19, 2022 at 12:01 PM

Hello... I upgraded ESP32 to 1.0.6 (latest). Keeps getting this error from your esp32cam.cpp.

```
\esp32cam-main\src\esp32cam.cpp: In member function  
'esp32cam::ResolutionList esp32cam::CameraClass::listResolutions()  
const':  
\esp32cam-main\src\esp32cam.cpp:30:3: error: 'camera_sensor_info_t'  
was not declared in this scope  
camera_sensor_info_t* info = esp_camera_sensor_get_info(&sensor-  
>id);  
^  
\esp32cam-main\src\esp32cam.cpp:30:25: error: 'info' was not declared  
in this scope  
camera_sensor_info_t* info = esp_camera_sensor_get_info(&sensor-  
>id);  
^  
\esp32cam-main\src\esp32cam.cpp:30:70: error:  
'esp_camera_sensor_get_info' was not declared in this scope  
camera_sensor_info_t* info = esp_camera_sensor_get_info(&sensor-  
>id);
```

Loading...

Reply

---

**skeerthan98gmailcom**

February 22, 2022 at 2:35 PM

I too am getting the same error. Does anyone know the solution to this problem?

Loading...

Reply

**Prithvi**

March 1, 2022 at 8:51 PM

Iam also getting the same error. There is some problem in the installed esp32cam library.....anyone pls help

Loading...

Reply

---

**ToddC**

March 11, 2022 at 11:26 PM

looks like camera\_sensor\_info\_t has changed in the new version, raise an issue in the git repo

Loading...

Reply

---

**Fattymcnugget**

March 16, 2022 at 2:36 AM

Updating the esp module to 2.02 fixes the issue link bellow on how to install  
<https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html>

Loading...

Reply

---

## Sashi Kiran

April 5, 2022 at 10:39 PM

hello iam getting this error while executing the code can any one  
please help me to solve this

```
C:\Users\Bala Krishna\Documents\Arduino\libraries\esp32cam-
main\src\esp32cam.cpp: In member function 'esp32cam::ResolutionList
esp32cam::CameraClass::listResolutions() const':
```

```
C:\Users\Bala Krishna\Documents\Arduino\libraries\esp32cam-
main\src\esp32cam.cpp:30:3: error: 'camera_sensor_info_t' was not
declared in this scope
```

```
camera_sensor_info_t* info = esp_camera_sensor_get_info(&sensor-
>id);
```

^

```
C:\Users\Bala Krishna\Documents\Arduino\libraries\esp32cam-
main\src\esp32cam.cpp:30:25: error: 'info' was not declared in this
scope
```

```
camera_sensor_info_t* info = esp_camera_sensor_get_info(&sensor-
>id);
```

^

```
C:\Users\Bala Krishna\Documents\Arduino\libraries\esp32cam-
main\src\esp32cam.cpp:30:70: error: 'esp_camera_sensor_get_info' was
not declared in this scope
```

```
camera_sensor_info_t* info = esp_camera_sensor_get_info(&sensor-
>id);
```

^

Multiple libraries were found for "WiFi.h"

Used: C:\Users\Bala

Krishna\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.
0.6\libraries\WiFi

Not used: C:\Program Files (x86)\Arduino\libraries\WiFi

exit status 1

Error compiling for board ESP32 Wrover Module.

Loading...

Reply

**JL**

April 10, 2022 at 4:51 PM

You might be using an older version of the esp32 library. You can use this link for additional boards manager. You should be using 2.0.2 version.[https://github.com/espressif/arduino-esp32/releases/download/2.0.2/package\\_esp32\\_index.json](https://github.com/espressif/arduino-esp32/releases/download/2.0.2/package_esp32_index.json)

Loading...

Reply

---

**wahyu**

April 17, 2022 at 10:52 PM

hello, sorry to ask why when my python program is running it says unterminated string literal (detected at line 9)

Loading...

Reply

---

**wahyu**

April 17, 2022 at 11:24 PM

hello, I want to ask why when my python program runs it says unterminated string literal (detected on line 9)

Loading...

Reply

---

**Ke\_S**

April 21, 2022 at 1:35 AM

Hello,  
did you find a solution, i have the same problem

Loading...

Reply

---

**Indian Guy**

April 23, 2022 at 10:51 AM

you can upload code to esp32 with arduno also

Loading...

[Reply](#)