ESP32 CAM Based Face & Eyes Recognition System

In this project, we will build an ESP32 CAM Based Face & Eyes Recognition System.

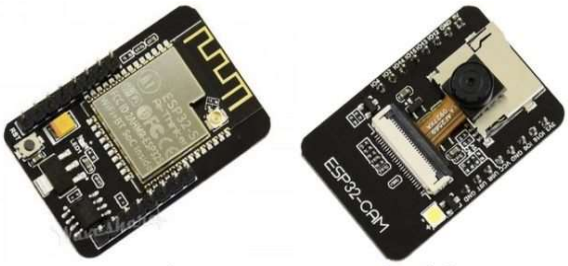
we have used the ESP32-CAM module, which is a small camera module with the ESP32-S chip.

Besides the OV2640 camera and several GPIOs to connect peripherals, it also features a microSD card slot that can be useful to store images taken with the camera.

components we used in this project :-

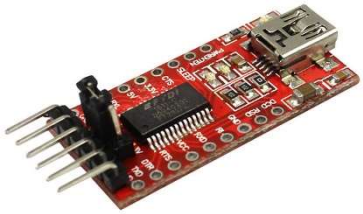
ESP32 CAM -

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.



Front Side Back Side

2)OV2640 camera: The board doesn't have a programmer chip.



3)**FTDI Programmer:** It is used for uploading the code from Arduino IDE to the ESP32 CAM.

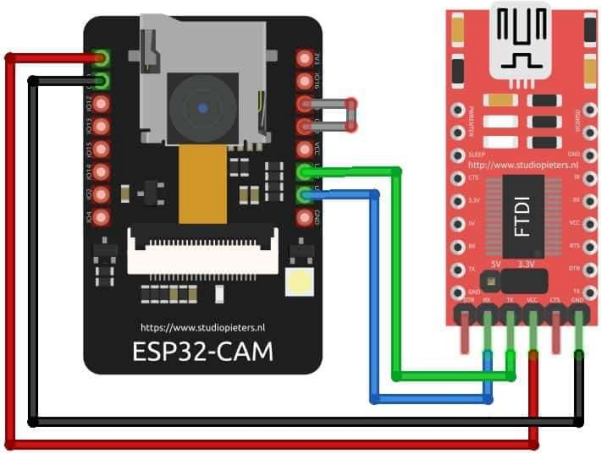
4)Cable:

With the help of this we can connect the usb of system to ftdi Programmer

5)Jumper Wire:

It is used to connect the FTDI programmer to the ESP32 CAM.

ESP32 CAM connection --

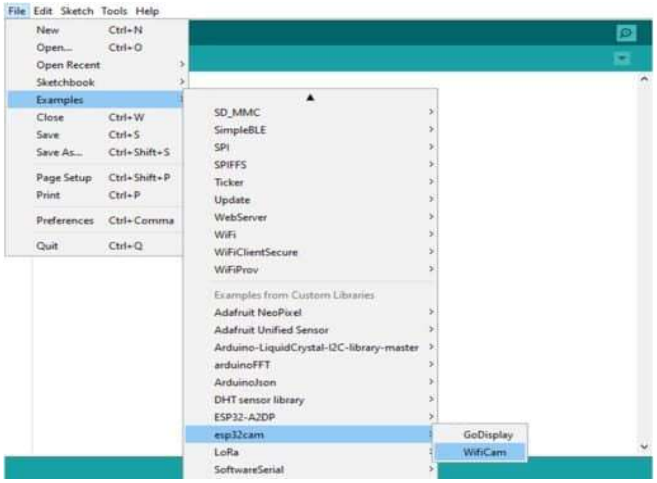


Procedure -

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**.

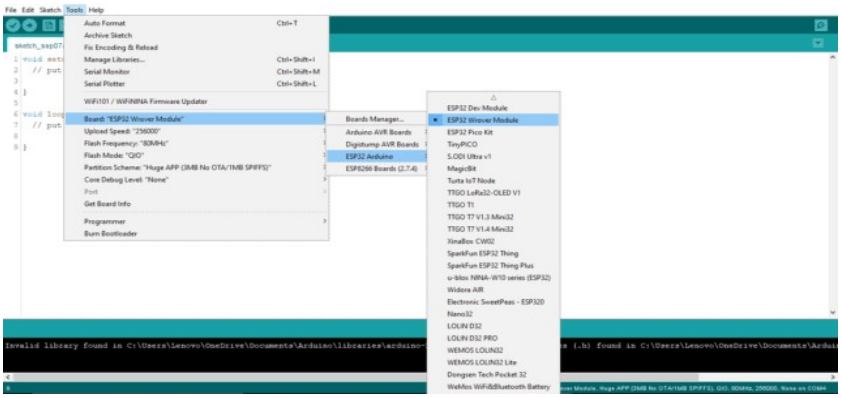
First to get the ip address of esp32 cam we need to run the code in Arduino ide.

Examples -> esp32cam -> wificam



Now we'll change wifi name and password to which we are connecting.

Now go to **tools**. Then select the **ESP32** Board. And from the list select **ESP32Wroover Module**.



Then connect the FTDI Module to your Computer and select the **COM Port.**

Now you can upload the code. But while you upload the code the IO0 pin should be shorted with the ground. Once the code is uploaded, remove the shorting Jumper and press the RESET pin.

If everything works fine then in serial monitor we get “camera ok” along with that we get an IP address.

The first step is to install Python.

Install Python once downloading is completed.
Then Go to the command prompt and install **NumPy** and **OpenCV** libraries.

Before copying this URL in python code we need to install all the libraries required like numpy library and cv2 for video capturing.

In Arduino ide we’ll have 3 options like low resolution, medium resolution and high resolution.

We can select any one depending on the situation.

Coming to python code we’ll train the model using “cascadeclassifier”.

After you run the code, a pop-up window with the name “**Live Transmission**” appears, showing the live video.

red-colored rectangular box covering the whole face and within that if it recognizes eyes.
Then **green colored boxes** appear around the eyes.

Code explanation :-

We have trained the model using cascadeclassifier and then we ran a while loop until we press “q” also we mentioned the dimensions of the rectangular boxes that are forming around face and eyes and also the color that should be formed around our face and eyes.

```python
import cv2
import urllib.request
import numpy as np
```

```python
f_cas= cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
eye_cascade=cv2.CascadeClassifier(cv2.data.haarcascades +'haarcascade_eye.xml')
url='http://192.168.97.151/cam-lo.jpg'
##'''cam.bmp / cam-lo.jpg /cam-hi.jpg / cam.mjpeg '''
cv2.namedWindow("Live Transmission", cv2.WINDOW_AUTOSIZE)

while True:
    img_resp=urllib.request.urlopen(url)
    imgnp=np.array(bytearray(img_resp.read()),dtype=np.uint8)
    img=cv2.imdecode(imgnp,-1)
    gray=cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    face=f_cas.detectMultiScale(gray,scaleFactor=1.1,minNeighbors=5)
    for x,y,w,h in face:
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),3)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        eyes = eye_cascade.detectMultiScale(roi_gray)
        for (ex,ey,ew,eh) in eyes:
            cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)


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

cv2.destroyAllWindows()
```

The first picture are the libraries we have used and the second image is the python code

working stills :

this are some stills of our working model where red color rectangular box will be formed around our face and green color rectangular box will be around our eyes if it detects face(to know what is face and eye we have trained the model using “cascadeclassifier” model) if no face is there in front of the camera nothing will be detected.

Team members

Gnaneshwar 2001CS15

Devanand 2001CS19