



GUJARAT TECHNOLOGICAL UNIVERSITY

Face Detection Using ESP32 CAM Sensor

A Detailed Report to be submitted

for

Summer Internship (3170001) Semester VII

in

Bachelor of Engineering (Computer Engineering)

Submitted By

Yash Makwana

180110107023

Internal Guide: Dr.Priyang Bhatt



**DEPARTMENT OF
COMPUTER ENGINEERING**

G H PATEL COLLEGE OF ENGINEERING & TECHNOLOGY

BAKROL ROAD, VALLABH VIDYANAGAR – 388 120

A.Y. 2021 – 2022

DISCLOSURE

*This is to certify that the Summer Internship report which is being submitted by **Yash Makwana** for Summer Internship (3170001) Semester VII in **Bachelor of Engineering (Computer Engineering)** to **Gujarat Technological University** is a record of the candidates' own work, carried out by him/her under my supervision and guidance, during the period of semester VII. I have read the report and that the information enclosed is correct and contains no confidential information.*

Internal Guide: Dr.Priyang Bhatt

INDEX

Topic	Page No
ACKNOWLEDGEMENT	i
PREFACE	ii

Sr. No.	Title	Page No.
1	INTRODUCTION	1
2	PROJECT DEFINITION	3
3	PROJECT	4
4	Module	5
5	Hardware Connection	7
6	Software	9
7	Code	12
8	CONCLUSION	14
9	BIBLIOGRAPHY	15

ACKNOWLEDGEMENT

My special gratitude to my project guide Dr. Priyang Bhatt for him inspiration, adroit guidance, constant supervision in the successful completion of the project.

I am very grateful to express my deep sense of gratitude to Dr. Maulika Patel, professor and Head of Computer Department, G. H. Patel College of Engineering & Technology, V. V. Nagar for her constant encouragement throughout the project.

I express my gratitude to all the professors of our department for their cooperation and keen interest throughout this project.

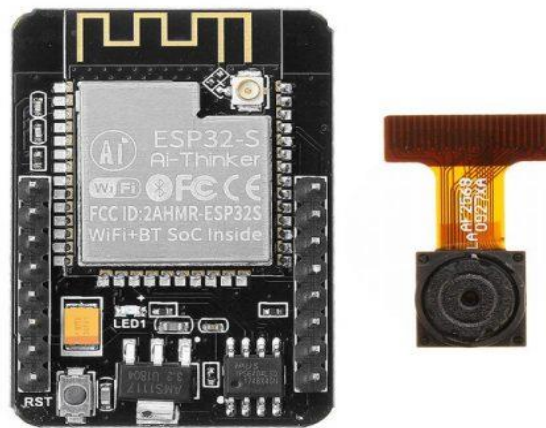
My sincere thanks to all my friends who helped me during this project.

PREFACE

Nowadays we are connecting online networks so we use wireless communication for our daily life. ESP32-CAM sensor is work with camara module to recognize your face. ESP32-CAM sensor is use for door security.

INTRODUCTION

ESP32-CAM SENSOR



The ESP32-CAM is a full-featured microcontroller that also has an integrated video camera and microSD card socket. It's inexpensive and easy to use, and is perfect for IoT devices requiring a camera with advanced functions like image tracking and recognition.

Summer Internship Project A.Y. 2021-22

PROJECT DEFINITION

ESP32-CAM sensor can be used to detect face.using sensor,ESP32 detect the face of the person and displays “HELLO SUBJECT” if recognised and “INTRUDER ALERT” if not recognised it can be use for security purpose like door alaram.

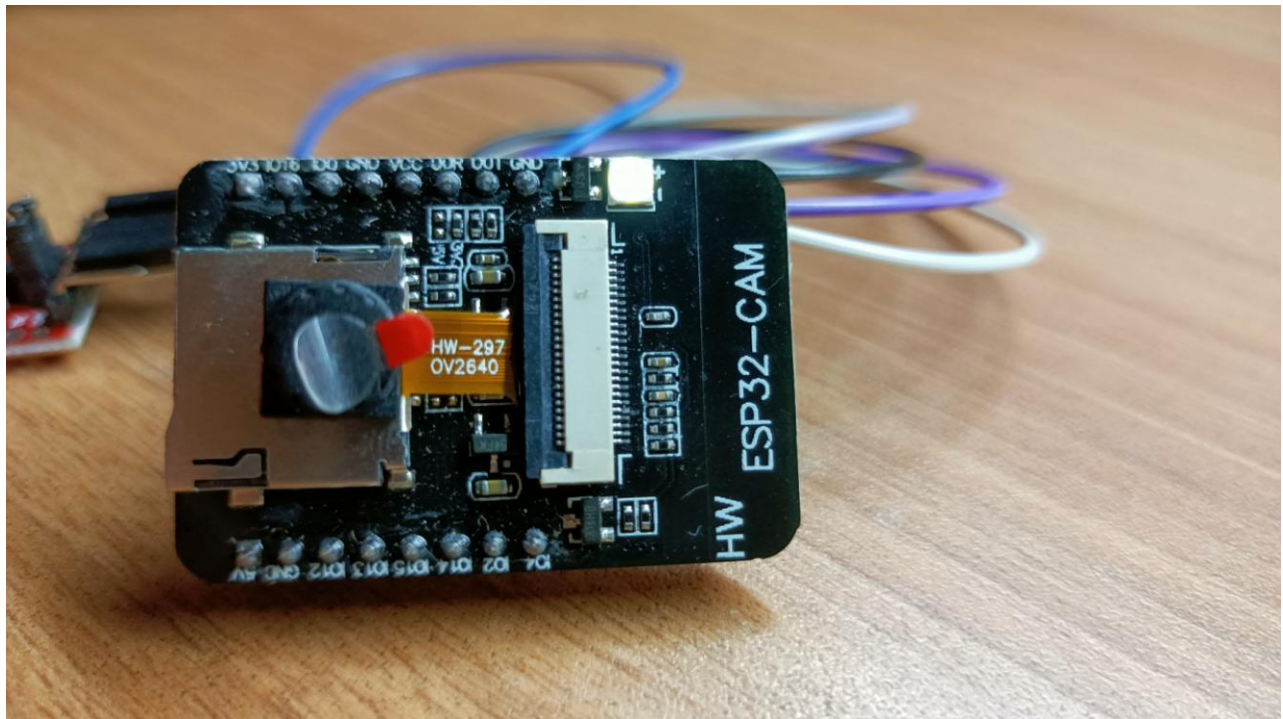
PROJECT

INTRODUCTION

ESP32-CAM is a low-cost ESP32-based development board with onboard camera, small in size. It is an ideal solution for IoT application, prototypes constructions and DIY projects. The board integrates WiFi, traditional Bluetooth and low power BLE , with 2 highperformance 32-bit LX6 CPUs. It adopts 7-stage pipeline architecture, on-chip sensor, Hall sensor, temperature sensor and so on, and its main frequency adjustment ranges from 80MHz to 240MHz. Fully compliant with WiFi 802.11b/g/n/e/i and Bluetooth 4.2 standards, it can be used as a master mode to build an independent network controller, or as a slave to other host MCUs to add networking capabilities to existing devices ESP32-CAM can be widely used in various IoT applications. It is suitable for home smart devices, industrial wireless control, wireless monitoring, QR wireless identification, wireless positioning system signals and other IoT applications. It is an ideal solution for IoT applications.

Module

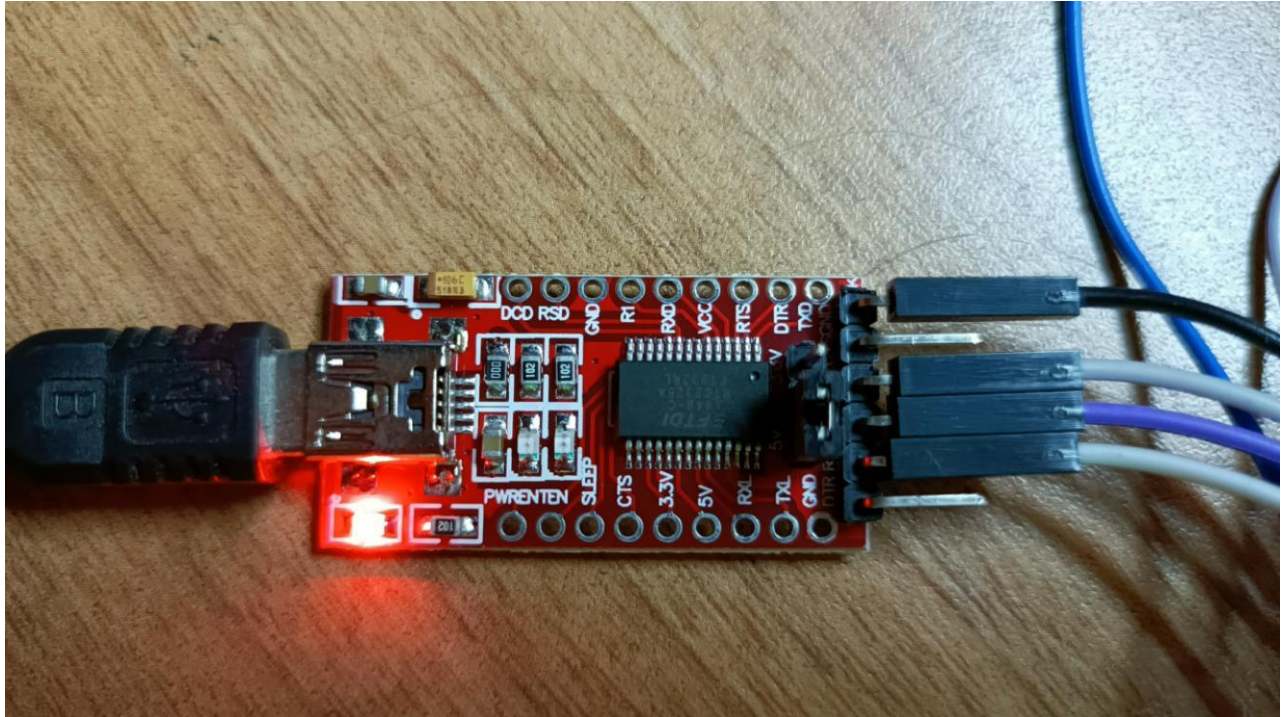
ESP32-CAM :-



The ESP32-CAM is a small size, low power consumption camera module based on ESP32. It comes with an OV2640 camera and provides onboard TF card slot.

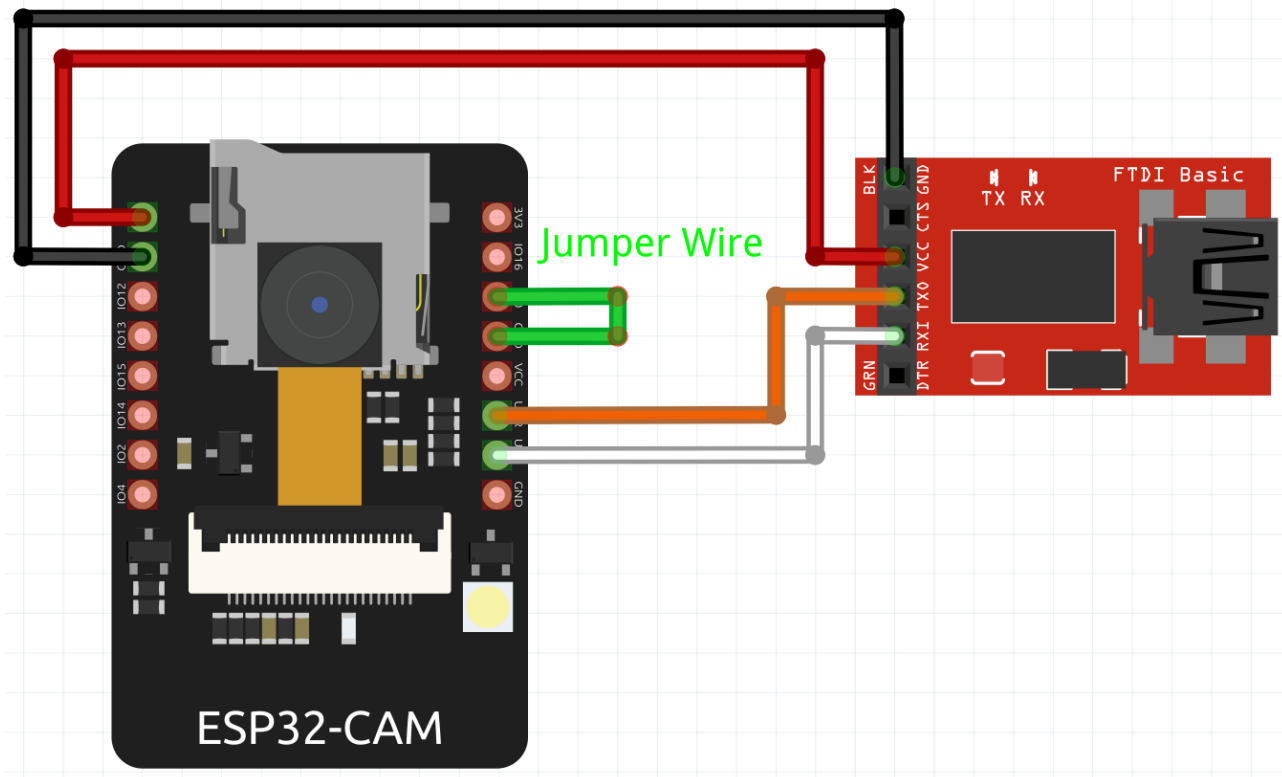
The ESP32-CAM can be widely used in intelligent IoT applications such as wireless video monitoring, WiFi image upload, QR identification, and so on.

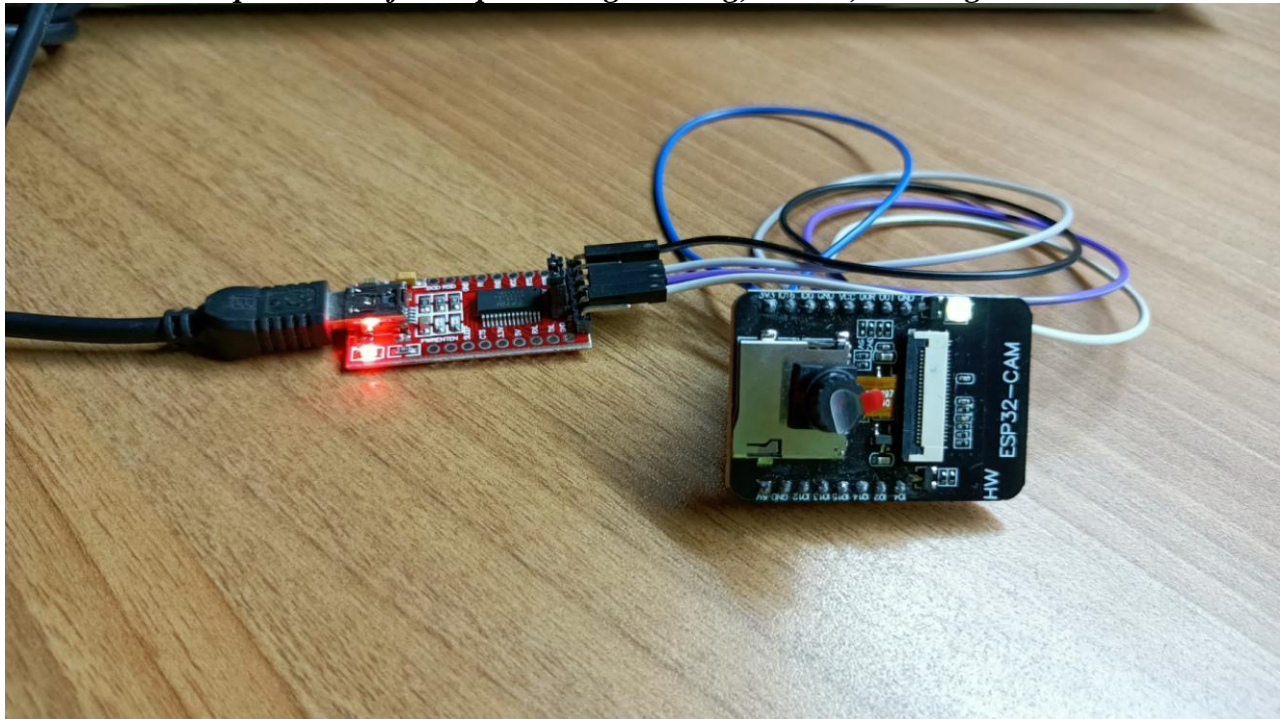
FTDI PROGRAMMER :-



FTDI USB to TTL serial converter modules are used for general serial applications. It is popularly used for communication to and from microcontroller development boards such as ESP32-CAM and Arduino micros.

Hardware Connection





Programming languages, frameworks, platforms, and environments used for ESP32 programming:

- Arduino IDE with the ESP32 Arduino Core
- Espressif IoT Development Framework - Official Espressif development framework for ESP32.
- Lua RTOS for ESP32
- Mongoose OS
- PlatformIO Ecosystem and IDE
- Pymakr
- Simba Embedded Programming Platform
- Whitecat Ecosystem Blockly Based Web IDE

In this tutorial we just focus on Arduino IDE with ESP32 Arduino Core because it is easy for beginner (certainly user must know about Arduino). And we will look a little on Espressif IoT Development Framework.

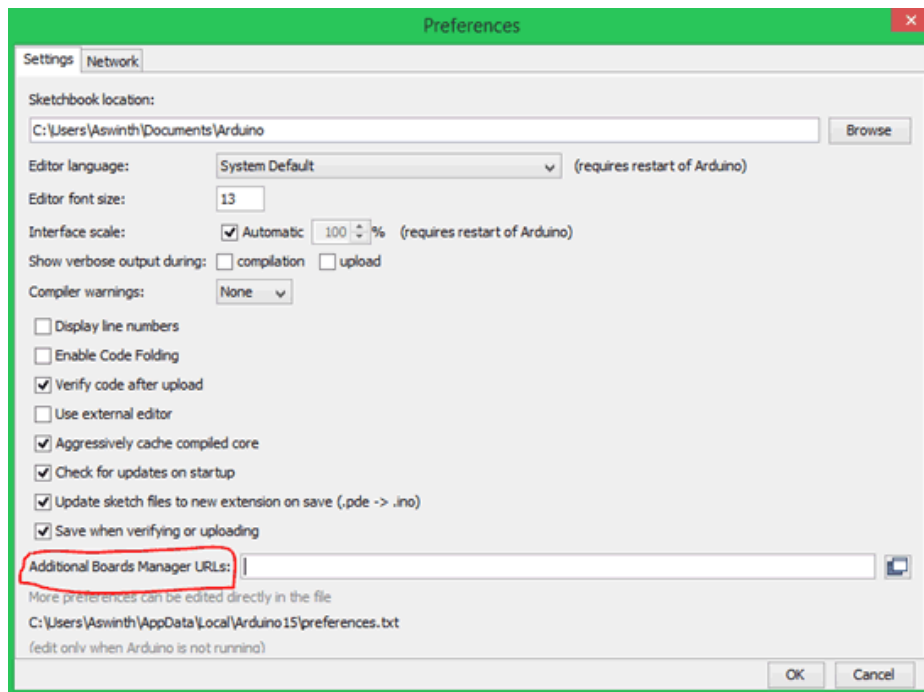
Software

Arduino ide :-



STEP 1: Now, let's get started. The first step would be to **download and install the Arduino IDE**.

STEP 2: Once installed, open the Arduino IDE and go to Files -> Preferences to open the preferences window and locate the "Additional Boards Manager URLs:" as shown below

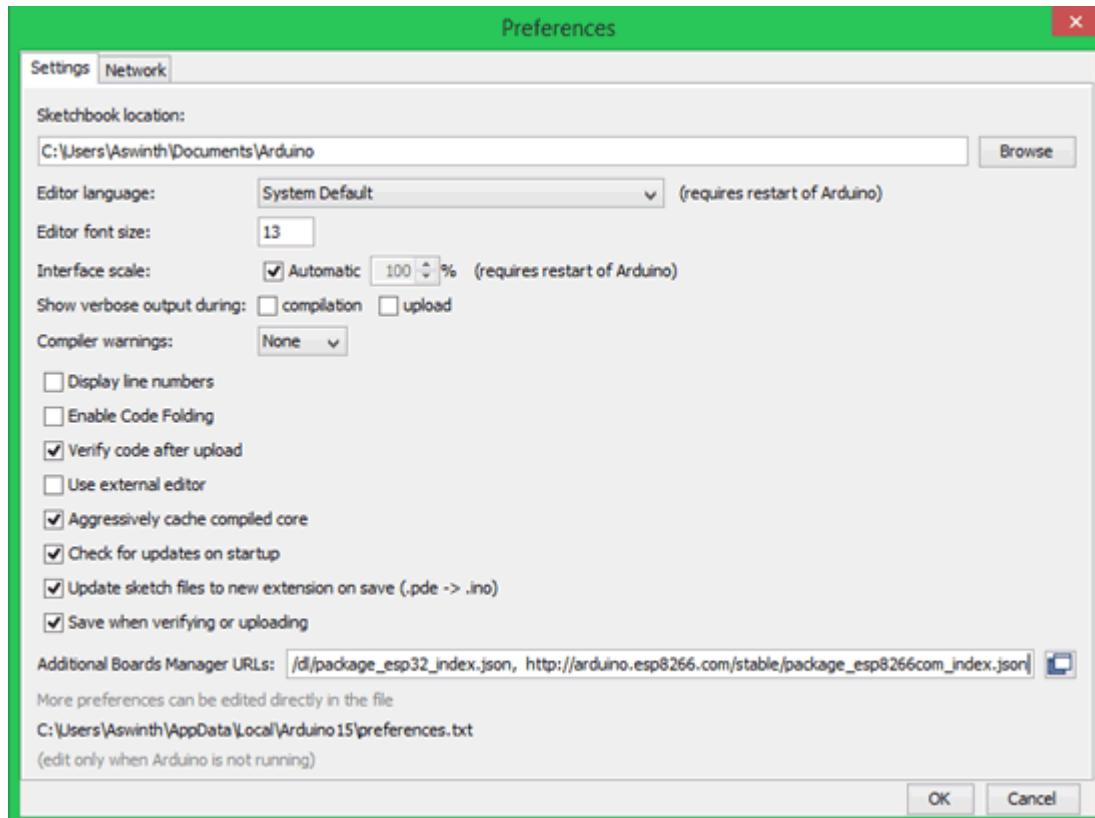


Department of Computer Engineering, GCET, VV Nagar

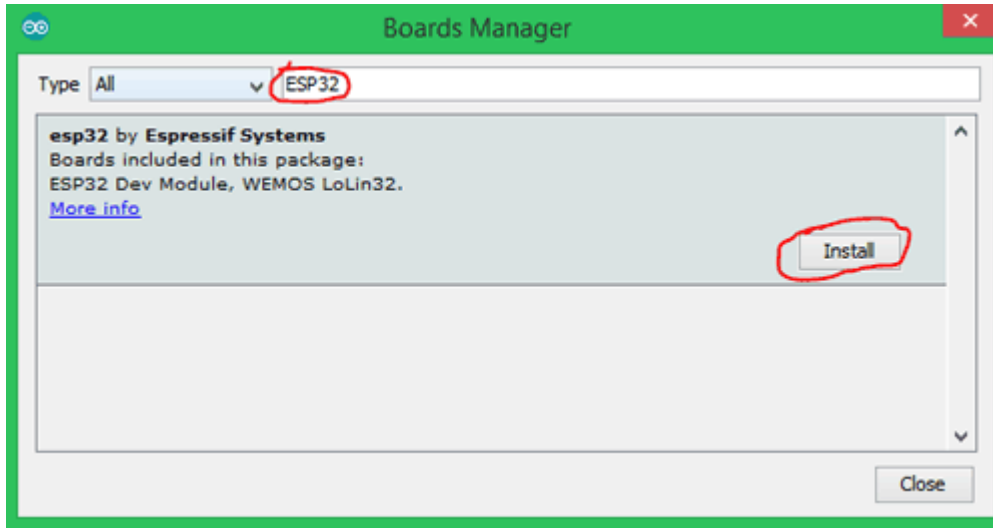
STEP 3: This text box might be empty or might also contain some other URL. If it is empty simply paste the below URL into the text box

```
https://dl.espressif.com/dl/package_esp32_index.json,
```

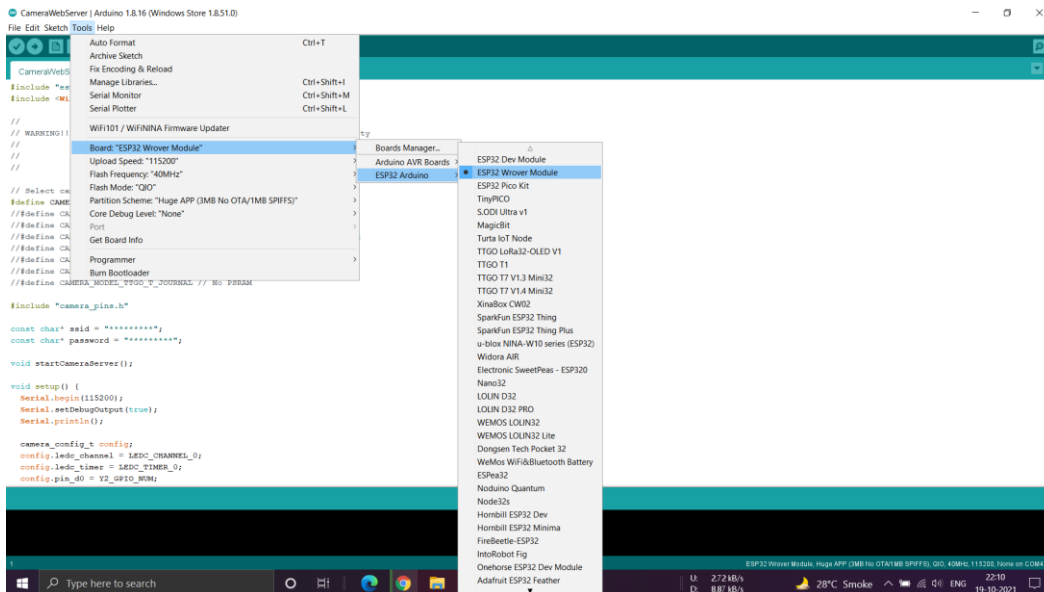
```
http://arduino.esp8266.com/stable/package_esp8266com_index.json
```

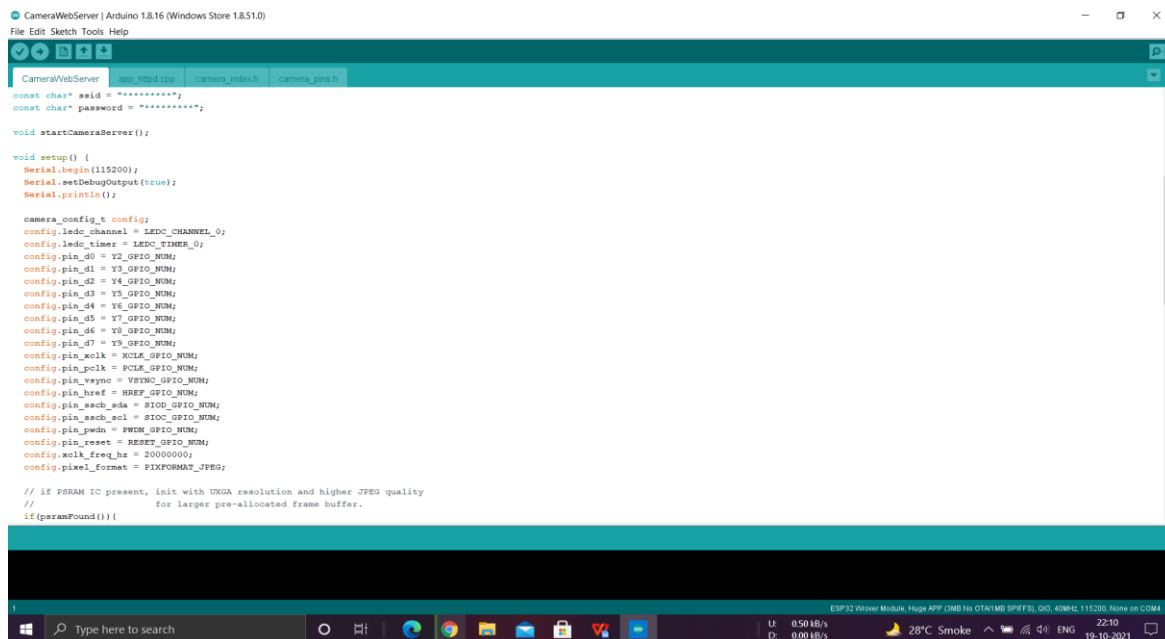


STEP 4: Now go to *Tools -> Boards -> Board Managers* to open the Board manager window and search for ESP32. If the URL was pasted correctly your window should find the below screen with *Install* button, just click on the Install button and your board should get installed.

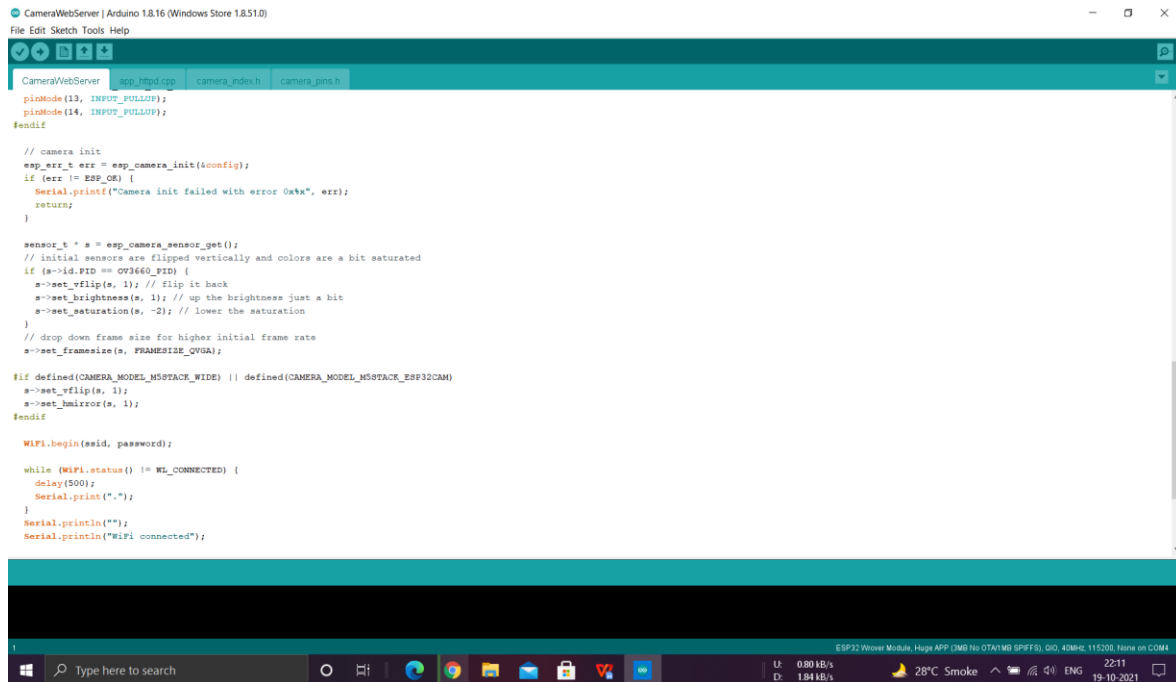


STEP 5: *Tools -> Board -> ESP32 Arduino -> ESP32 Wrover Module*.





Department of Computer Engineering, GCET, VV Nagar



```
CameraWebServer | Arduino 1.8.16 (Windows Store 1.8.51.0)
File Edit Sketch Tools Help

CameraWebServer app_http.cpp camera_index.h camera_pins.h

pinMode(13, INPUT_PULLUP);
pinMode(14, INPUT_PULLUP);
#endif

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

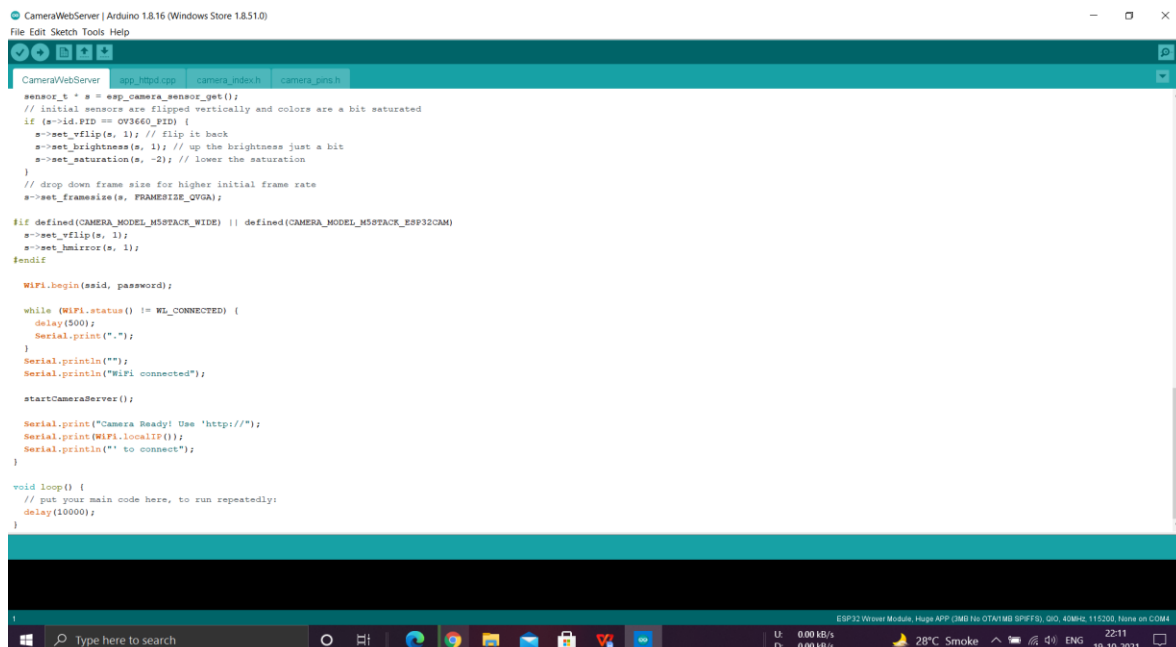
sensor_t * s = esp_camera_sensor_get();
// Initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
}
// drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_OVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

1
```



```
CameraWebServer | Arduino 1.8.16 (Windows Store 1.8.51.0)
File Edit Sketch Tools Help

CameraWebServer app_http.cpp camera_index.h camera_pins.h

sensor_t * s = esp_camera_sensor_get();
// Initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
}
// drop down frame size for higher initial frame rate
s->set_framesize(s, FRAMESIZE_OVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM)
s->set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://'");
Serial.print(WiFi.localIP());
Serial.println(" to connect");
}

void loop() {
    // put your main code here, to run repeatedly:
    delay(10000);
}

1
```

CONCLUSION

We can conclude that ESP32 with camera has enough computing power for simple security tasks and for camera picture taking and preparing for other more powerful processors.

BIBLIOGRAPHY

1. <https://www.google.com/>
2. <https://electronicsinnovation.com/esp32-cam-face-recognition-and-video-streaming-with-arduinoide/>
3. <http://www.iotsharing.com/2017/05/introduction-to-esp32.html>
4. <https://circuitdigest.com/microcontroller-projects/programming-esp32-with-arduino-ide>