CAM SURVILLENCE SYSTEM

A Seminar report Submitted

in Partial Fulfilment of the Requirements

for the Degree of

BACHELOR OF TECHNOLOGY

In

Electronics and communication Engineering

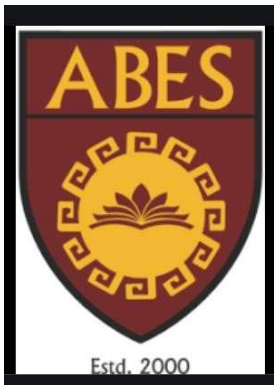
By

Kushagra Punia

1900320310080

Under the Supervision of

Dr. Manidipa Roy



ABES ENGINEERING COLLEGE

GHAZIABAD

2021-22

ACKNOWLEDGEMENT

I, express my sincere thanks to my supervisor, Dr. Manidipa Roy, ECE Department, ABES Engineering College, AKTU University for guiding us right from the inception till successful completion of the seminar. I would also like to thank our HOD Prof. (Dr.) Sanjay Kumar Singh for his valuable guidance andcooperation to decide Seminar topic and its content.

Signature:

Name:

Roll No.:

Date:

ABSTRACT

The Camera Surveillance System is made using Espressif device called ESP32 Cam Module. This system can be used in many ways like door sensor, monitoring systems etc. The use of IoT platform like Blynk used here, is very effective. Use of IoT Platform has enabled us to very effective in many cases. Bynk application enabled us to create widgets, based on the project work. Blynk IoT platform is the 3rd best platform for IoT technology. Use of Arduino IDE helped in the example codes for Camera Webserver interaction with module. Hardware Components like FTDI Programming Module helped in uploading the code from IDE to Module. Making circuit connections made us learn about circuit problems, debugging drills etc.

All along, the INIF Six Weeks Technology Entrepreneurship Development Programme on IoT Technology helped me a lot in this project. Learning all the basic concepts of IoT concepts has been main part of this programme. I thank my mentor Mr. Rakesh Gupta for Guiding me all along the programme's journey.

TABLE OF CONTENT

Chapter – 1: INTRODUCTION

The basic working of the surveillance system is that the ESP32 cam module detects the images and live video stream and displays its web link on the Serial monitor by which, we can access the live video steam and configure settings like face detection, image quality, color filters etc. along with it. The cam module intercepts the TTL transmitted by the FTDI module and configure the whole module based on the program written.

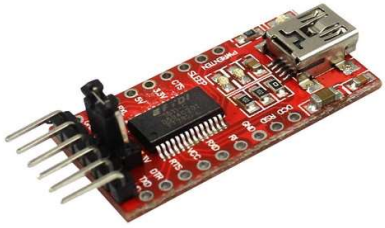
The ESP32 module comes with a camera that has to be connected to it. The FTDI Module is used here for upload the code from IDE to Cam Module. Once the code is uploaded, it gives the link and gets connected as output and displayed on Serial monitor and from there, we can find the current and precise location from anywhere.

To simplify work, we have a library called esp32 library and Camera Webserver. This library does a lot of job such as creating webserver link for camera settings and live steam.

Chapter – 2: HARDWARE OVERVIEW

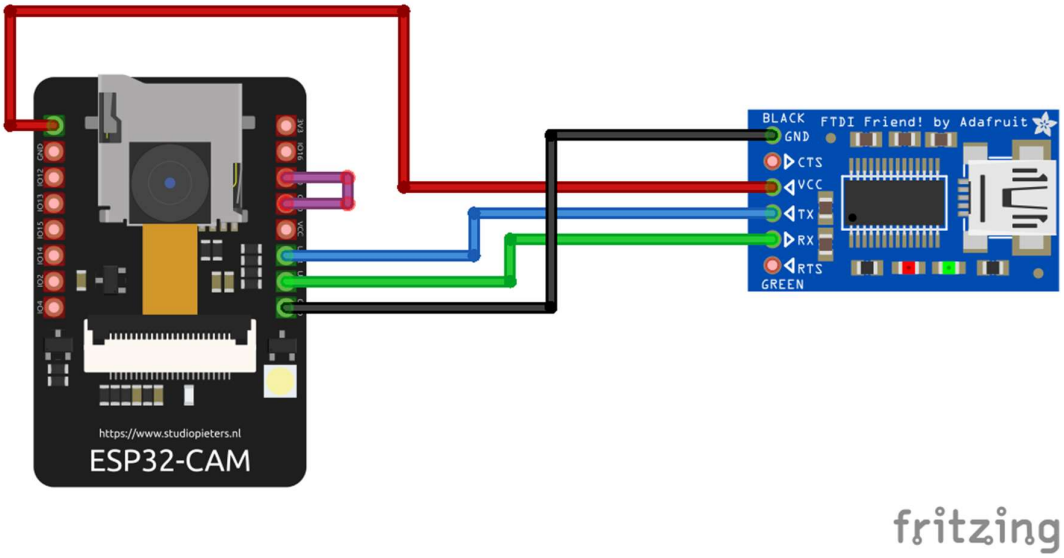
Equipment which has been used in this project are:

1. ESP32 Cam Module (AI-Thinker)
2. FTDI Programming Module (USB to TTL Converter)
3. USB Cable (5V Mini-USB Data Cable)
4. Jumper Cables
5. Mobile (for BLYNK) and Laptop (for Arduino IDE)
6. Bread Board



Chapter – 3: CIRCUITS

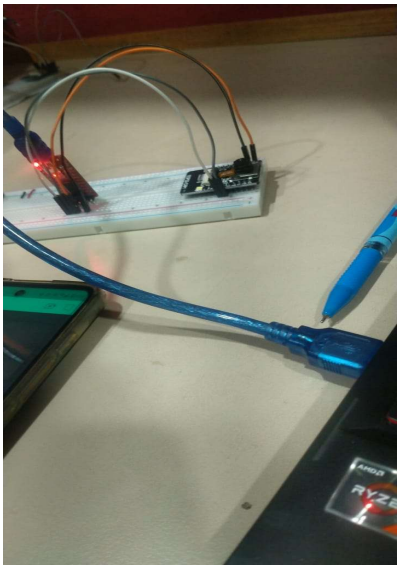
Main circuits for this project are displayed below.



ESP32-CAM	FTDI PROGRAMMER
GND	GND
5V	VCC (5V)
U0R	TX
U0T	RX
GPIO 0	GND

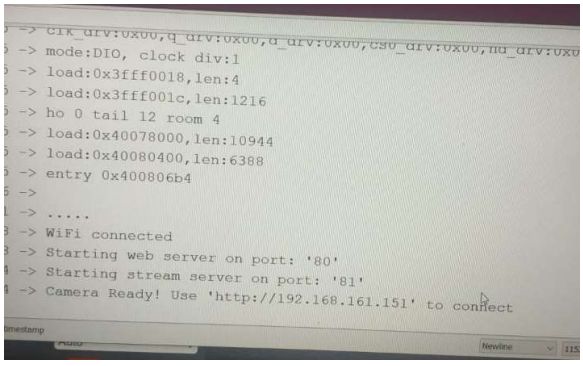
Chapter – 4: Connection and Working

1. Make connection according to the circuit diagram shown above and connect the USB data cable with FTDI module and laptop.

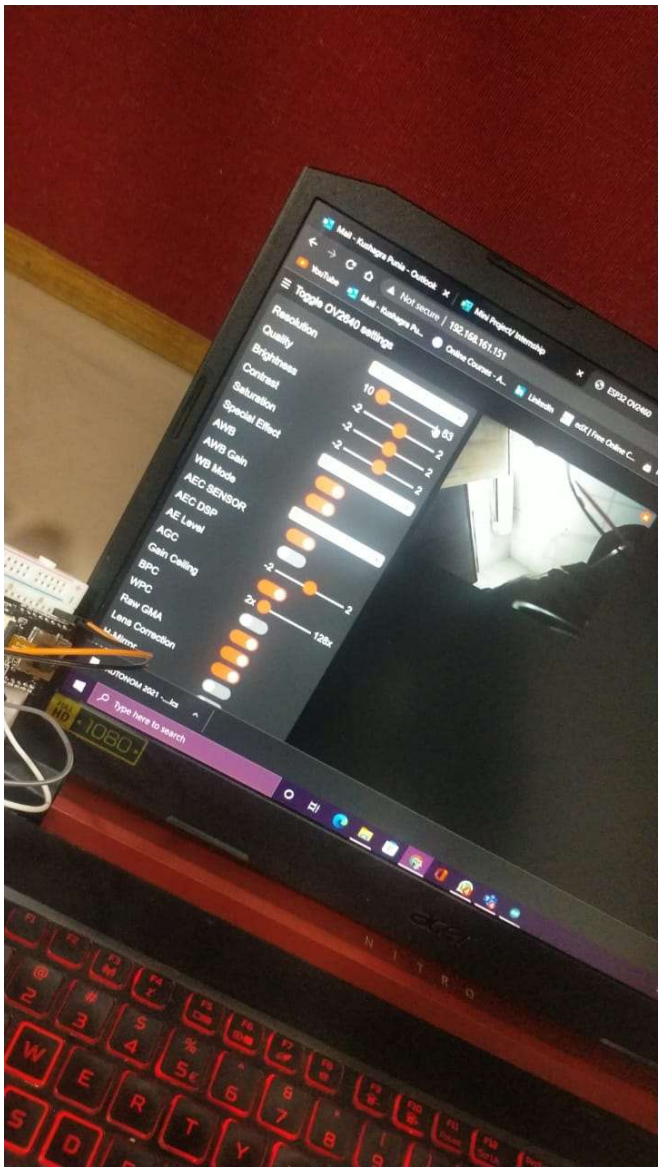


2. Write the code given below in Arduino IDE and upload the code and open Serial Monitor in Arduino IDE.

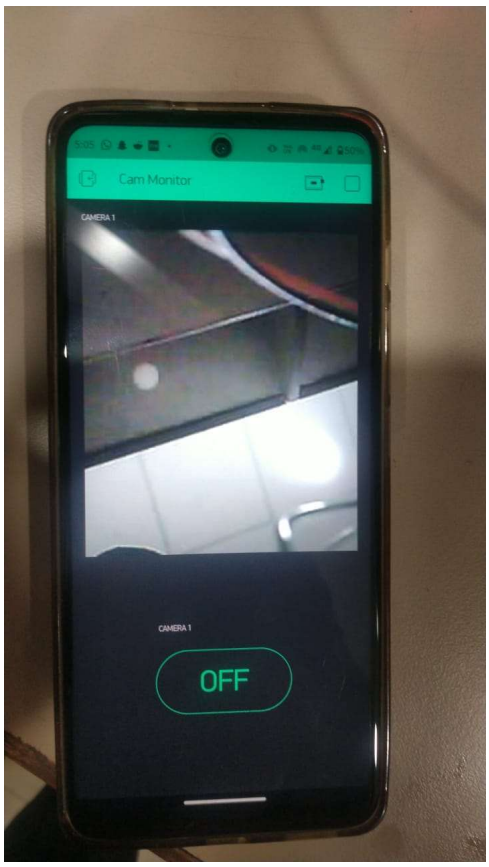
(Make sure to add your Wi-Fi id and password as well as Blynk provided Authentication code)

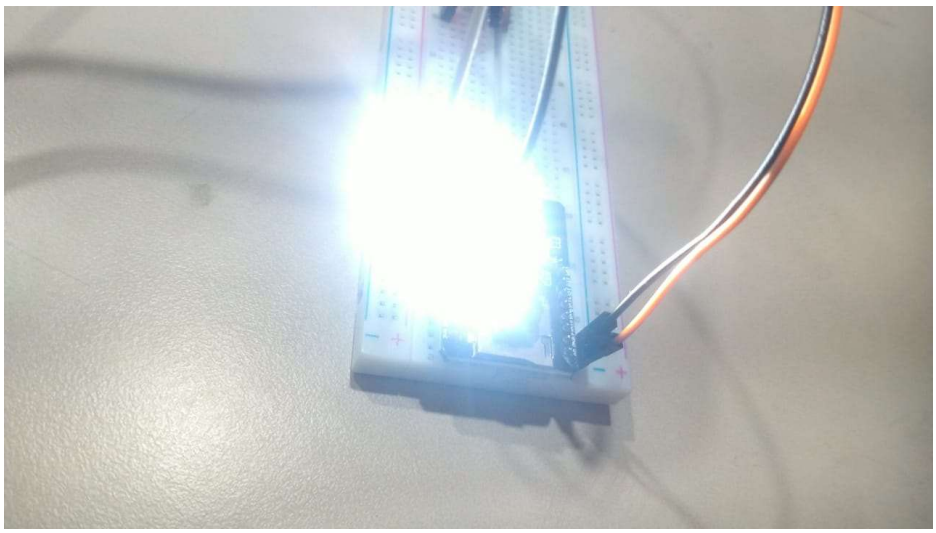


3. Copy the link provided in the Serial Monitor and paste in the browser which will get us to the Camera Webserver for live video stream. Here, we can change image quality, add face detection features etc.



4. Open the Blynk Application and click on button to get live images.





Chapter – 5: PROGRAME CODE

ESP32CAM_Blynk_TakePhoto

```
//Made by Kushagra Punia
//Blynk ESP32 CAM Simple Monitor System

#include "esp_camera.h"
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

// Select camera model
#define CAMERA_MODEL_AI_THINKER // Has PSRAM

#include "camera_pins.h"

#define PHOTO 14      //ESP32 CAM 1
#define LED 4

const char* ssid = "Kush";       //wifi name
const char* password = "28042001";       //password
char auth[] = "JsumijlkViZjFHOZXzVnwKlN4db4d8d2";          //Auth Code sent
by Blynk

String local_IP;
int count = 0;
void startCameraServer();

void takePhoto()
{
  digitalWrite(LED, HIGH);
  delay(200);
  uint32_t randomNum = random(50000);
  Serial.println("http://"+local_IP+"/capture?_cb="+ (String)randomNum);
  Blynk.setProperty(V1, "urls",
"http://"+local_IP+"/capture?_cb="+(String)randomNum);
  digitalWrite(LED, LOW);
  delay(1000);
}
```

```
void setup() {
  Serial.begin(115200);
  pinMode(LED,OUTPUT);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
  //                      for larger pre-allocated frame buffer.
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

  // 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_QVGA);

  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());
  local_IP = WiFi.localIP().toString();
  Serial.println("' to connect");
  Blynk.begin(auth, ssid, password);
}

void loop() {
  // put your main code here, to run repeatedly:
  Blynk.run();
  if(digitalRead(PHOTO) == HIGH){
    takePhoto();
    }

}
```

Chapter – 6: RESULTS

- As we click on the button, simultaneously we will get live images from it.
- The link provided in the serial monitor and be used to set image quality of the camera
- The purpose of surveillance can be fulfilled accordingly

CONCLUSION

Solution to each problem by myself and with the guidance of Mr. Rakesh Gupta sir was the most important part of the project and it provided me with hands-on experience with IoT devices which will help me in future.

We discovered solved the surveillance problem of live images using a simple yet sophisticated device called ESP32 with little bit of programming and hardware knowledge.

I got to know various libraries in this software and their functions. Some important things that I learned including designing a good program architecture and converting real life situation into an effective code, and how to write a good looking easily readable andunderstandable as well as time and memory efficient code.

And further we can also implement this project with Door Sensor which can give access to people with registered face only and can also be operated with Telegram Application.

REFERENCES

1. https://www.viralsciencecreativity.com/
2. https://www.google.com/imghp?hl=en
3. https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/