SEPTEMBER 2022

IoT Based Face Recognition Door Lock System

USEFUL SMART HOME PROJECT USING THE ESP32 CAMERA MODULE

Prepared by

INURI GUNATHILAKA
AR-96541
AF/18/14575

Lecturer, Dept. of ICT

DR. SENAKA AMARAKEERTHI,
INTERNET OF THINGS

Abstract

The Project reported here in and titled as “IoT Based Face Recognition Smart Door Lock System for Remote Monitoring and Control using ESP32- CAM and a Smartphone". When inside or outside of their houses, people seek security wherever it is possible. More than just the numbers, digital marketers must also create a concise yet effective social media report that is meaningful for all stakeholders involved. Automatic facial recognition in a smart security system is the most difficult computer vision challenge of the last ten years. Applications for face recognition and computer vision are expanding daily, taking on new dimensions and having good effects on society. In this study, provide a system that can control and recognize faces from a variety of positions. To determine the system's suitability under diverse circumstances, it saved a variety of datasets of well-known individuals.

wireless camera and sensors which ensured remote monitoring and control of doorways. The system empowered the user to monitor the doorway by capturing images using a high-performance wireless camera i.e. ESP32-CAM connecting other devices and sensors in an IoT network. The development of this dynamic system with zero error, real-time responsiveness, smooth performance, and viability, intelligence, and feasibility was a significant task.

Table of Content

enroll multiple faces, detect and the door will unlock automatically...

01. Introduction

The project work carried out here provided an insight into the development of IoT systems. IoT refers to the interconnection of physical objects such as machinery, vehicles, buildings, and other things with electronics, software, sensors, actuators, and network connectivity that allow them to gather and share data. The Internet of Things (IoT) is made up of the conventional domains of embedded systems, wireless sensor networks, control systems, and automation systems. Because of this, the internet of things builds on the revolutionary success of mobile and internet networks.

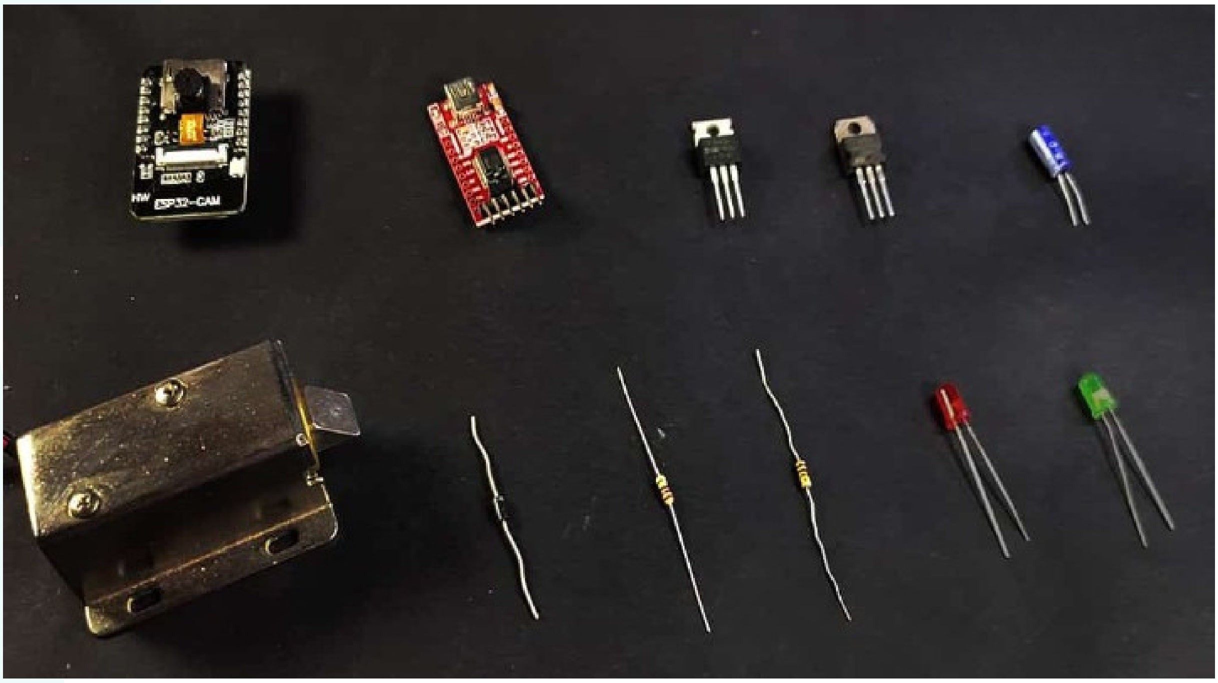
In this ESP32CAM project, I have made the ESP32CAM Face Recognition Door Lock System. You can enroll multiple faces in the esp32cam face detection automatic door lock. If it detects any enrolled face, the door will unlock automatically.

02. Objective

The objective was to develop a dynamic wireless doorway security system which empowers the user to acquire visitor's photo identity and take an informed decision for giving that person access into his/ her premises. The developed system should be user friendly and feasible.

a versatile, viable, low power security solution with real-time response was what this project aimed to achieve.

03. Required Components for Face Detection Door Lock

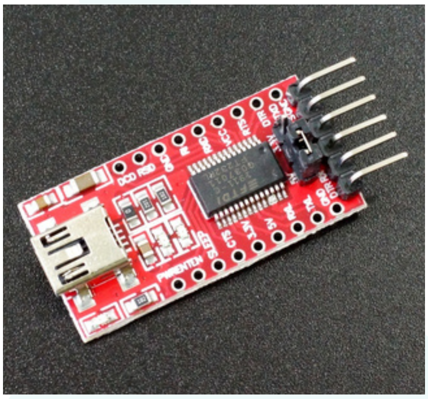


1. ESP32-CAM board
2. Electronic door lock 12v
3. 7805 voltage Regulator (5v)
4. TIP122 NPN Transistor
5. 10k Resistor (1no)
6. 220-ohm Resistors (2no)
7. Capacitor 220uF
8. Diode 1N4007 (1no)
9. LEDs 5-mm (2no)
10. 12V DC adaptor
11. FTDI232 USB to TTL converter (for programming the esp32cam)

ESP32-CAM board

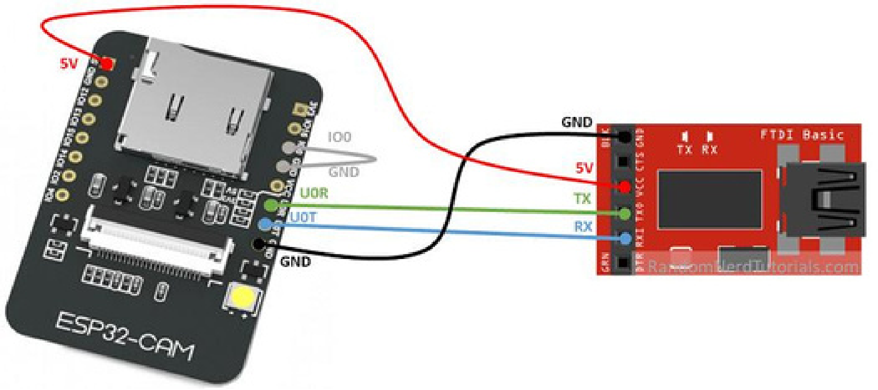
A fully functional microcontroller with an inbuilt video camera and microSD card slot is called the ESP32-CAM. It's affordable, simple to use, and ideal for Internet of Things (IoT) devices that need a camera with sophisticated features like image tracking and identification. The sample software distributed by Espressif includes a sketch that allows you to build a web-based camera with a sophisticated control panel. After you get the hang of programming the device, you'll find that it is very easy to use.

FTDI232 USB to TTL converter (for programming the esp32cam)



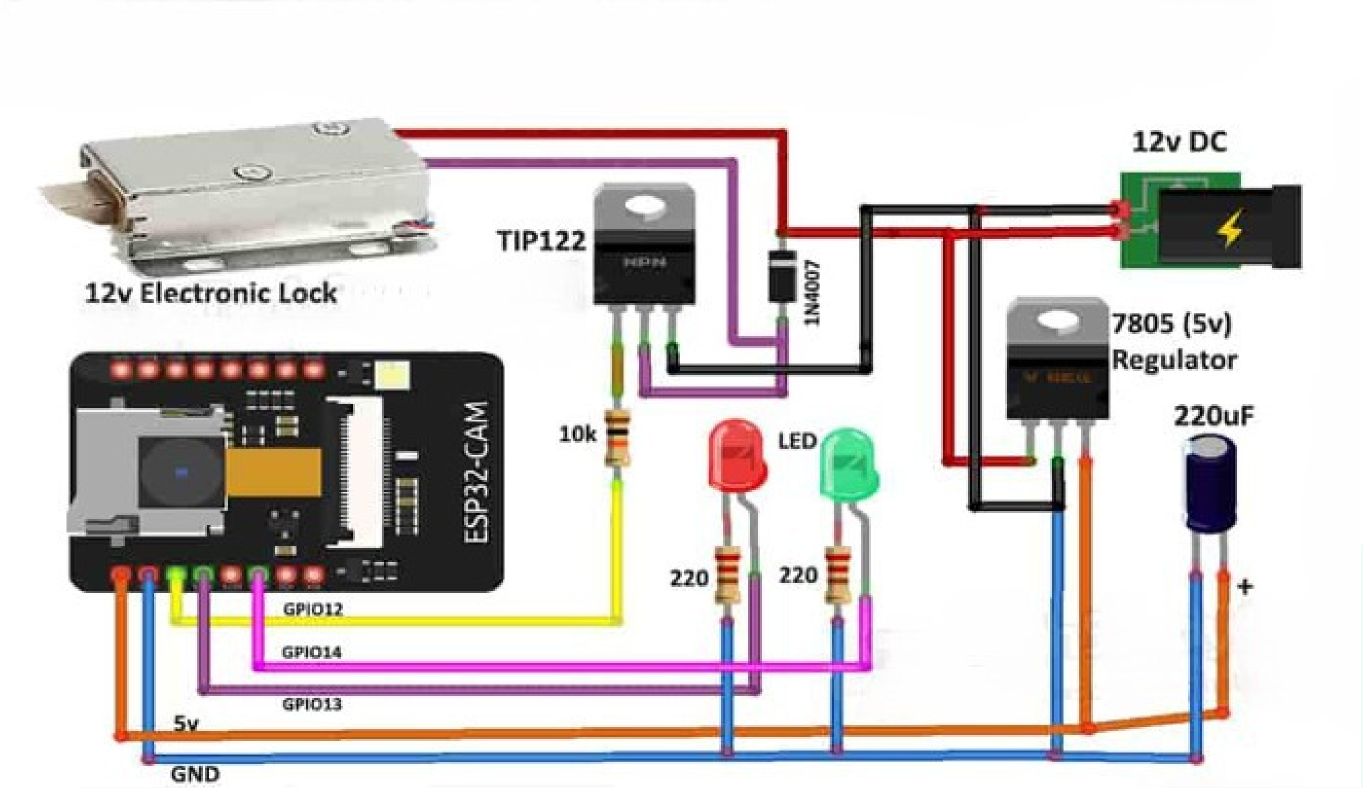
The ESP32-CAM module has fewer I/O pins than the previous ESP-32 module we looked at. Many of the GPIO pins are used internally for the camera and the microSD card port. Another thing missing from the ESP32-CAM module is a USB port. In order to program this device, you'll need to make use of an FTDI adapter.

Here is the hook up diagram for connecting the FTDI adapter to the ESP32-CAM module:



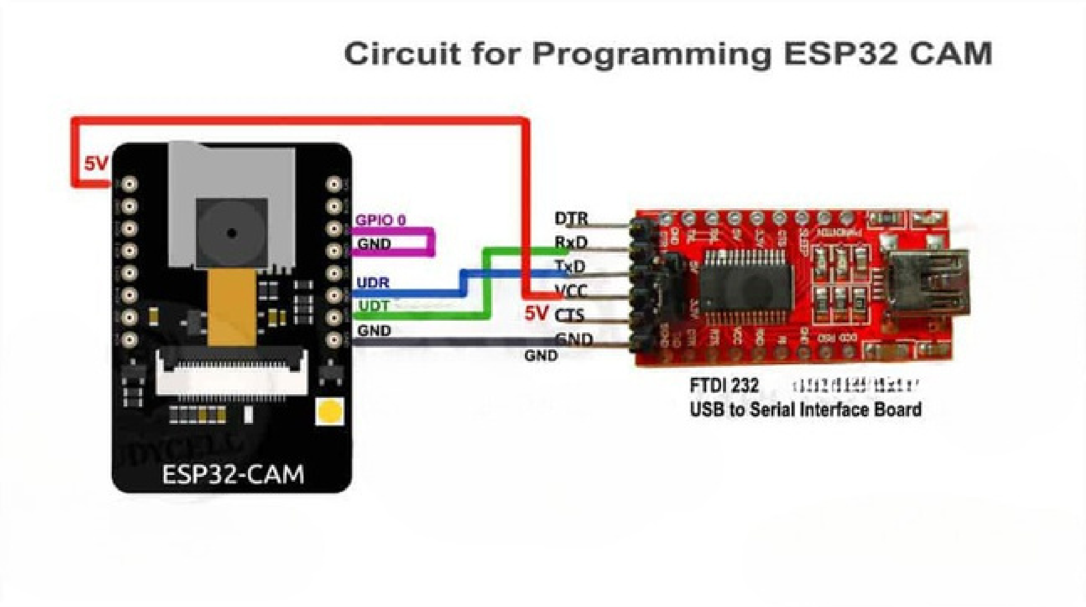
04. Circuit of the ESP32CAM Face Recognition Lock

Face Recognition Door Lock using ESP32-CAM



You can easily design this smart door lock with the camera using a 12v electronic lock, ESP32 CAM module, and some basic electronics components.

05. Programming ESP32-CAM board


Circuit for Programming ESP32 CAM
5V
GPIO 0
GND
UDR
UDT
GND
DTR
RxD
TxD
VCC
5V
CTS
GND
GND
ESP32-CAM
FTDI 232
USB to Serial Interface Board

To program the ESP32CAM, I have used FTDI232 USB to Serial interface board. I have connected the FTDI232 with ESP32CAM as per the above circuit.
While uploading the code we have to connect GPIO 0 with the GND pin of ESP32CAM.

Once you have finished programming the module you can power it down and remove this connection.

06. Implementation

```
/*********************************************************************************************
 *  PREFERENCES--> ADITIONAL BOARDS MANAGER URLS :
HTTPS://DL.ESPRESSIF.COM/DL/PACKAGE_ESP32_INDEX.JSON,
HTTP://ARDUINO.ESP8266.COM/STABLE/PACKAGE_ESP8266COM_INDEX.JSON
 *  BOARD SETTINGS:
 *  BOARD: "ESP32 WROVER MODULE"
 *  UPLOAD SPEED: "921600"
 *  FLASH FREQUENCY: "80MHZ"
 *  FLASH MODE: "QIO"
 *  PARTITION SCHEME: "HUE APP (3MB NO OTA/1MB SPIFFS)"
 *  CORE DEBUG LEVEL: "NONE"
 *  COM PORT: DEPENDS *ON YOUR SYSTEM*
 *
 *  GPIO 0 MUST BE CONNECTED TO GND PIN WHILE UPLOADING THE SKETCH
 *  AFTER CONNECTING GPIO 0 TO GND PIN, PRESS THE ESP32 CAM ON-BOARD RESET BUTTON TO PUT THE BOARD
IN FLASHING MODE
 ***************************************************************************************************/




#INCLUDE "ESP_CAMERA.H"
#INCLUDE <WIFI.H>
//
// WARNING!!! PSRAM IC REQUIRED FOR UXGA RESOLUTION AND HIGH JPEG QUALITY
//         ENSURE ESP32 WROVER MODULE OR OTHER BOARD WITH PSRAM IS SELECTED
//         PARTIAL IMAGES WILL BE TRANSMITTED IF IMAGE EXCEEDS BUFFER SIZE
//

// SELECT CAMERA MODEL
//#DEFINE CAMERA_MODEL_WROVER_KIT
//#DEFINE CAMERA_MODEL_ESP_EYE
//#DEFINE CAMERA_MODEL_M5STACK_PSRAM
//#DEFINE CAMERA_MODEL_M5STACK_WIDE
#DEFINE CAMERA_MODEL_AI_THINKER

#INCLUDE "CAMERA_PINS.H"

#DEFINE RED 13
#DEFINE GREEN 14
#DEFINE LOCK 12
```

06. Implementation

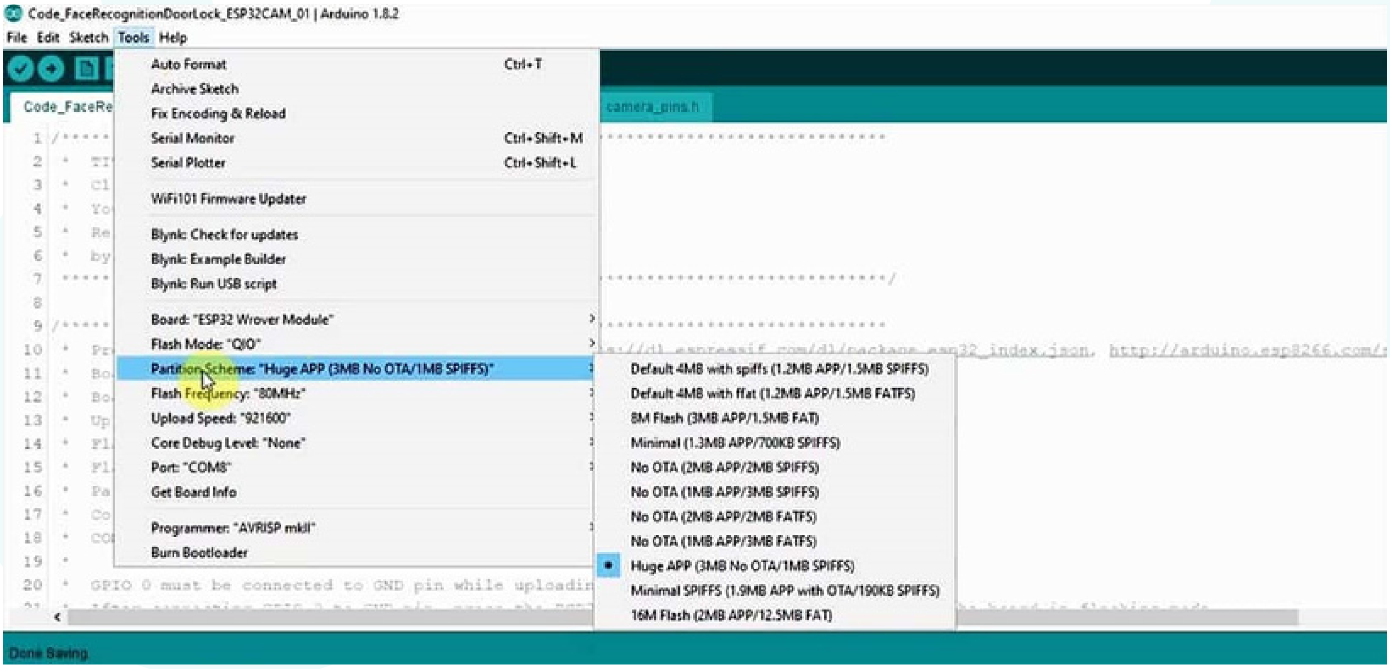
```
CONST CHAR* SSID = "WIFI NAME"; //WIFI SSID
CONST CHAR* PASSWORD = "WIFI PASSWORD"; //WIFI PASSWORD

VOID STARTCAMERASERVER();

BOOLEAN MATCHFACE = FALSE;
BOOLEAN OPENLOCK = FALSE;
LONG PREVMILLIS=0;
INT INTERVAL = 6000;  //DELAY

VOID SETUP() {
 PINMODE(LOCK,OUTPUT);
 PINMODE(RED,OUTPUT);
 PINMODE(GREEN,OUTPUT);
 DIGITALWRITE(LOCK,LOW);
 DIGITALWRITE(RED,HIGH);
 DIGITALWRITE(GREEN,LOW);

 SERIAL.BEGIN(115200);
 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;
 //INIT WITH HIGH SPECS TO PRE-ALLOCATE LARGER BUFFERS
 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;
 }

#IF DEFINED(CAMERA_MODEL_ESP_EYE)
 PINMODE(13, INPUT_PULLUP);
 PINMODE(14, INPUT_PULLUP);
#ENDIF
```

06. Implementation

```
// 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 BLIGHTNESS 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);

#IF DEFINED(CAMERA_MODEL_M5STACK_WIDE)
  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() {
  IF(MATCHFACE==TRUE && OPENLOCK==FALSE)
  {
    OPENLOCK=TRUE;
    DIGITALWRITE(LOCK,HIGH);
    DIGITALWRITE(GREEN,HIGH);
    DIGITALWRITE(RED,LOW);
    PREVMILLIS=MILLIS();
    SERIAL.PRINT("UNLOCK DOOR");
   }
   IF (OPENLOCK == TRUE && MILLIS()-PREVMILLIS > INTERVAL)
   {
    OPENLOCK=FALSE;
    MATCHFACE=FALSE;
    DIGITALWRITE(LOCK,LOW);
    DIGITALWRITE(GREEN,LOW);
    DIGITALWRITE(RED,HIGH);
    SERIAL.PRINT("LOCK DOOR");
    }
}
```

Before uploading the code to ESP32CAM, please check the following setting:

- Update the **Preferences** -> Aditional boards Manager URLs: https://dl.espressif.com/dl/package_esp32_index.json, http://arduino.esp8266.com/stable/package_esp8266com_index.json
- **Board Settings:**
  - Board: “ESP32 Wrover Module”
  - Upload Speed: “921600”
  - Flash Frequency: “80MHz”
  - Flash Mode: “QIO”
  - Partition Scheme: “Hue APP (3MB No OTA/1MB SPIFFS)”
  - Core Debug Level: “None”
- COM Port: Depends *On Your System*
- GPIO 0 must be connected to GND pin while uploading the sketch
- After connecting GPIO 0 to GND pin, press the ESP32 CAM on-board RESET button to put the board in flashing mode



Code_FaceRecognitionDoorLock_ESP32CAM_01 | Arduino 1.8.2

Board: "ESP32 Wrover Module"
Flash Mode: "QIO"
Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)"
Flash Frequency: "80MHz"
Upload Speed: "921600"
Core Debug Level: "None"
Port: "COM8"
Programmer: "AVRISP mkII"

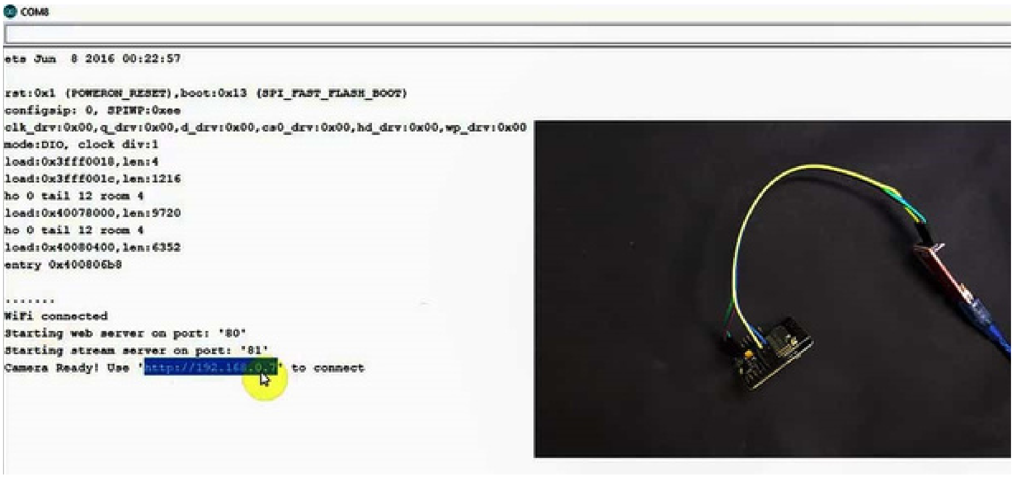
● Huge APP (3MB No OTA/1MB SPIFFS)

```
* GPIO 0 must be connected to GND pin while uploadin
```

Done Saving

File
Edit
Sketch
Tools
Help
Code_FaceRe
camera_pins.h
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
WiFi101 Firmware Updater
Blynk: Check for updates
Blynk: Example Builder
Blynk: Run USB script
Get Board Info
Burn Bootloader
Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)
Default 4MB with ffat (1.2MB APP/1.5MB FATFS)
8M Flash (3MB APP/1.5MB FAT)
Minimal (1.3MB APP/700KB SPIFFS)
No OTA (2MB APP/2MB SPIFFS)
No OTA (1MB APP/3MB SPIFFS)
No OTA (2MB APP/2MB FATFS)
No OTA (1MB APP/3MB FATFS)
Minimal SPIFFS (1.9MB APP with OTA/190KB SPIFFS)
16M Flash (2MB APP/12.5MB FAT)

07. Get the IP from ESP32-CAM



```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_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:9720
ho 0 tail 12 room 4
load:0x40080400,len:6352
entry 0x400806b8

.......
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.0.7' to connect
```

How to get the IP from ESP32CAM:

1. After uploading the code disconnect GPIO 0 from GND pin.

2. Open Serial Monitor with Baud rate 115200.

3. Press the RESET button on the ESP32CAM board

4. Copy the IP Address from the last line.

Now connect all the components as per the circuit diagram. And give the 12v DC supply to circuit.

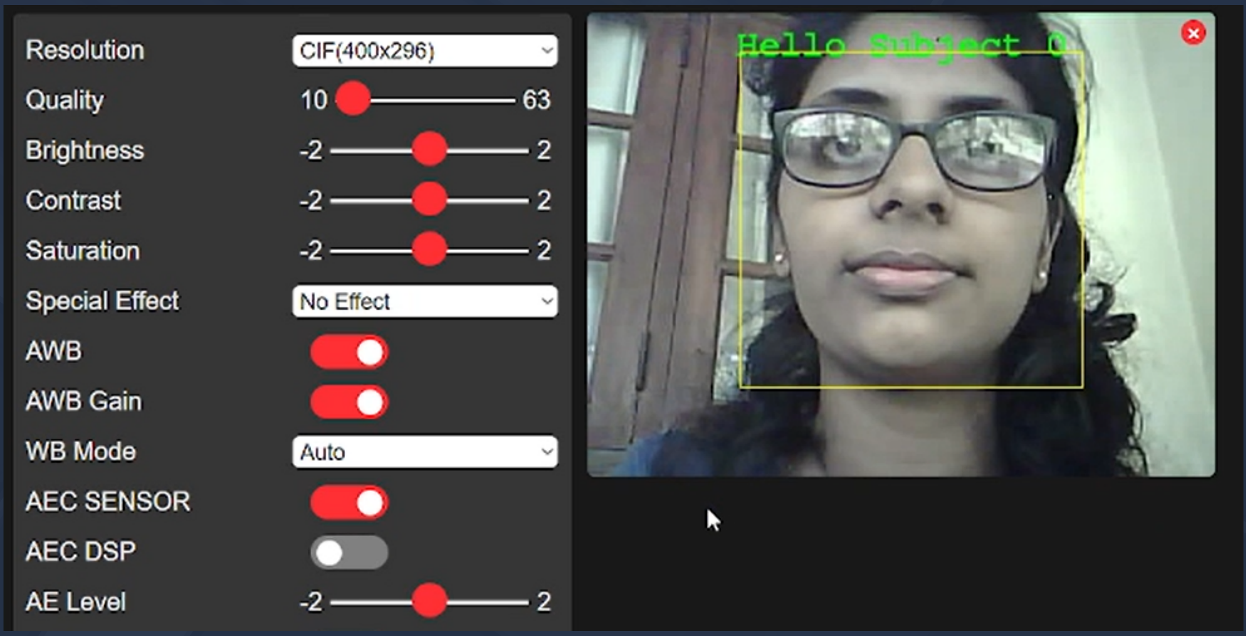
After that open any browser, then type the IP address to start the stream.
Now, we have to enroll faces.

1) Click on the Start Stream.
2) Turn on Face Detection and Face Recognition.
3) Click on Enroll Face.
4) Then ESP32CAM will take some sample pictures of the face.
5) At last, a green box will appear around the face.

In this way we can enroll multiple faces.

08. Finally, ESP32-CAM Face Recognition Lock is ready

Now, if the ESP32 CAM detects any enrolled face, it will unlock the door and Green LED will turn on for 6 seconds.



Resolution CIF(400x296)

Quality 10 — 63

Brightness -2 — 2

Contrast -2 — 2

Saturation -2 — 2

Special Effect No Effect

AWB ●

AWB Gain ●

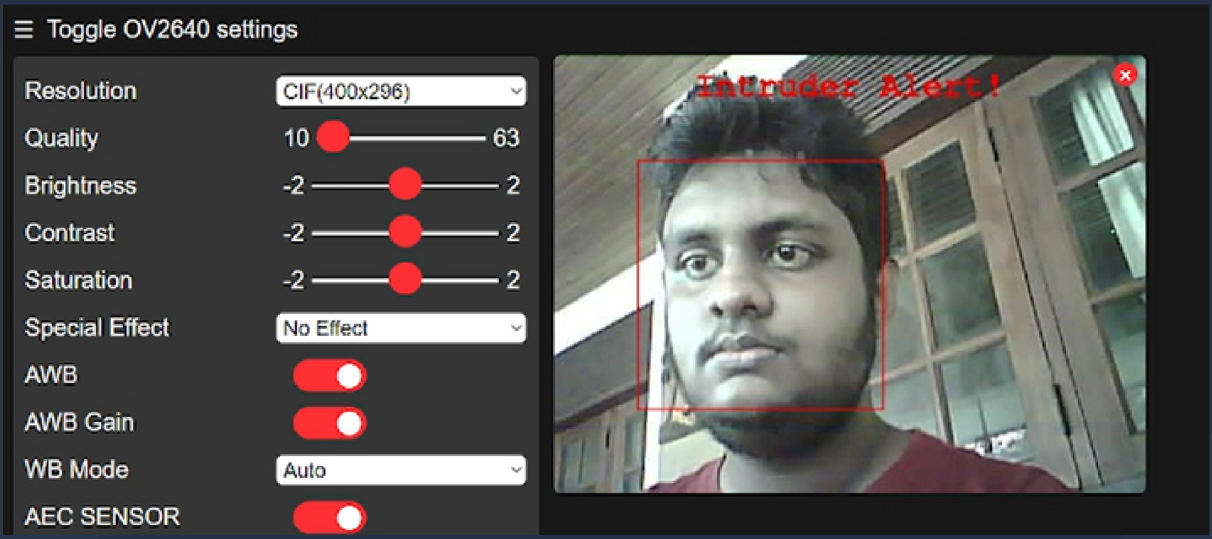
WB Mode Auto

AEC SENSOR ●

AEC DSP ○

AE Level -2 — 2

08. Finally, ESP32-CAM Face Recognition Lock is ready



☰ Toggle OV2640 settings

Resolution CIF(400x296)

Quality 10 63

Brightness -2 2

Contrast -2 2

Saturation -2 2

Special Effect No Effect

AWB ●

AWB Gain ●

WB Mode Auto

AEC SENSOR ●

If the face does not match with the enrolled face, the door will remain lock as you can see in the above picture.

09. Tutorial video of ESP32CAM Face Recognition

URL- https://www.youtube.com/watch?v=ig2Z77t43uA

In this tutorial video, I have shown how to make this DIY Face Recognition Door Lock System using esp32 cam step by step. For better understanding please watch the complete video.

10. Uses

10.1 Security

There are many flaws in the traditional lock and key framework. The fact that the key is easily duplicable is the most important of these. This could at any time lead to an unwanted intrusion by a stranger. Each person's face identity is distinct, making it nearly hard for someone to break into your private property when using a biometric lock.

10.2 Cost-Effective

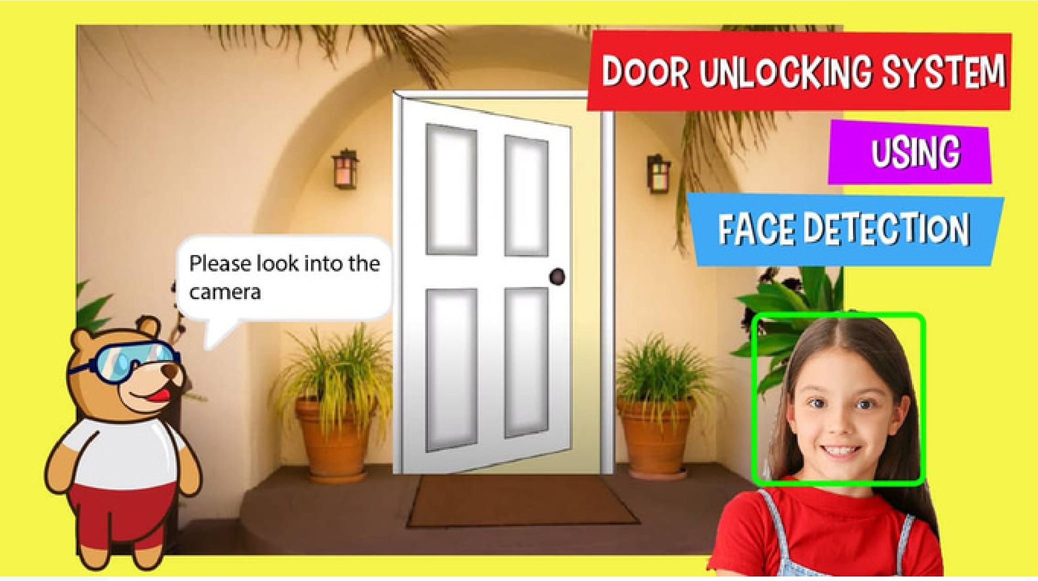
There are many flaws in the traditional lock and key framework. The fact that the key is easily duplicable is the most important of these. This could at any time lead to an unwanted intrusion by a stranger. Each person's face identity is distinct, making it nearly hard for someone to break into your private property when using a biometric lock.

10.3 In-accessible

A typical lock can be picked or forced open. A card-based entrance system is vulnerable to hacking attempts and card loss. These are not entirely fool-proof. Since detection of each person's face is unique, which is the only criteria required to get access, the security provided here is quite reassuring.

11. Conclusion

Face recognition door lock have gained tremendous benefits compared to conventional key door locks, combination door locks, keyless keypad lock or card reader door locks. Thus, face detect door locks surpass security protection, convenience, and speed. This system is very affordable, simple to setup, and beneficial because it may be used in several modes. The potential of the smart locking system is enormous. Users will be able to gain entry to the required location using simply their mobile device instead of their traditional key. Future development of the system will focus on making it as mobile and extensible as possible.
The arrangement of a facial recognition system using Arduino IDE that can make the system littler, lighter and work successfully utilizing lower control use, so it is more convenient than the PC-Windows based face recognition system. Also, it triggers the security alarm for unauthorized persons whose faces data doesn't match with the stored data inside its database.



12. References

[1]
“ESP32-CAM Video Streaming and Face Recognition with Arduino IDE,” Random Nerd Tutorials, Dec. 10, 2019. https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/

[2]
“How to upload the code to ESP32 CAM using Arduino or programmer,” Circuit Schools. https://www.circuitschools.com/how-to-program-upload-the-code-to-esp32-cam-using-arduino-or-programmer/ (accessed Sep. 03, 2022).

[3]
“ESP32-CAM Face Recognition Door Lock System,” circuitdigest.com. https://circuitdigest.com/microcontroller-projects/esp32-cam-face-recognition-door-lock-system (accessed Sep. 03, 2022).