Graphic Era Deemed to be University

Department Of Computer Science and Engineering

Project work

On

Smart Door Locking System With Face Recognition

Submitted by –   **Under supervision of –**

Name   : Shubham Chaudhary   **Mr. Piyush Agarwal Sir**
Class   :Btech CSE
Section   : B
University Roll : 2017049
Class ROLL   : 10

Acknowledgement

Place :Graphic Era (Deemed To be) University, Dehradun

Date : December 21ST ,2021

I would like to express my sincere thanks to **Mr. Piyush Aggarwal sir,** for his valuable guidance and support in completing my project.

I would also like to express my gratitude towards our **HOD sir Mr.Devesh Pratap** for giving me this great opportunity to do a project on **Smart Door Locking System With Face Recognition .** Without their support and suggestions, this project would not have been completed.

Shubham Chaudhary

Smart Door Locking System With Face Recognition

ABOUT THIS PROJECT

In the light of the current scenario the problems associated with security in buildings of smart cities,home and offices,there should be a way so that it can be prevented from any theft activity.There is a great solution for it in which the doors will be opened by face recognition only when the registered users will try to enter.

Working

This system has two LEDs one green coloured and the other red coloured.When the door is opened the green LED will glow and when the door is closed red LED will glow.It will be done in accordance with the recognition done by the ESP-32 camera.
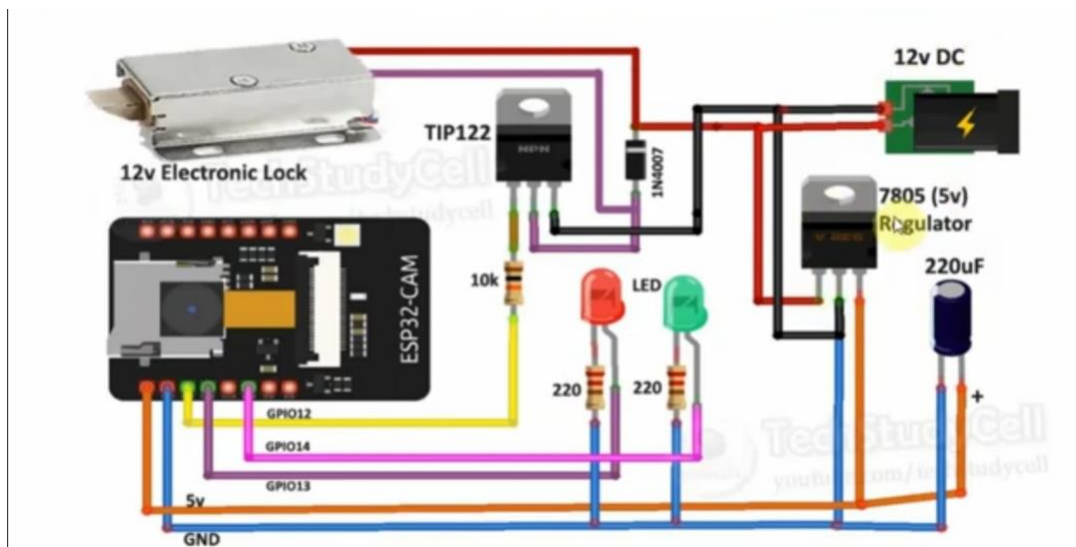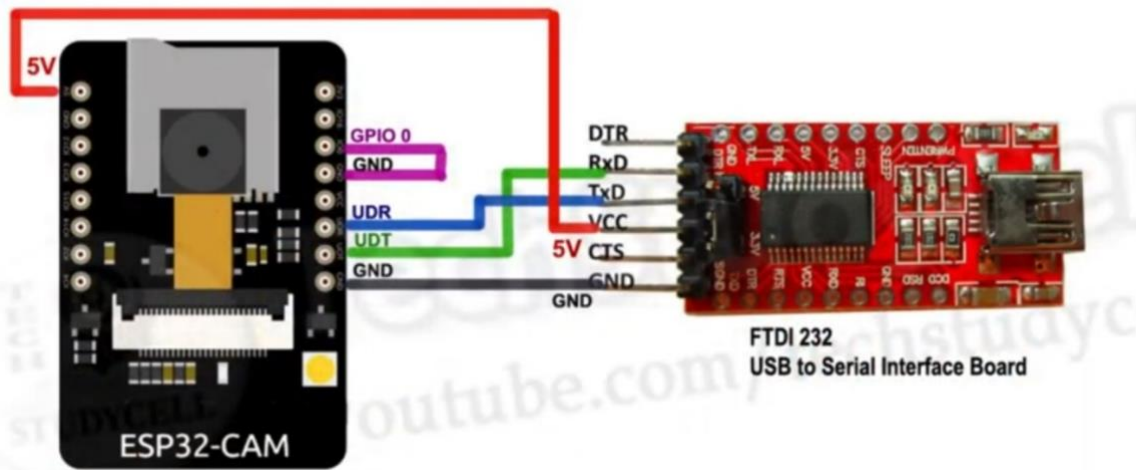
Components Required

1.)HARDWARE:

* ESP 32 Camera board

*Breadboard

*Jumper Wires(Male-Male,Female-Female,Male-Female)

*LEDs(Red,Green)

*Electronic Door Lock 12V

*TIP 122 NPN Transistor

*7805 voltage regulator(5V)

*Two 10K resistors

*12V DC Adaptor

*Diode 1N4007

*FTDI 232 USB to Serial Interface Board

*USB Power Cable

2.)SOFTWARE:

*Arduino IDE

Circuit Diagram

CODE :
```
#include "esp_camera.h"
```

```cpp
#include <WiFi.h>
#define CAMERA_MODEL_AI_THINKER

#include "camera_pins.h"
#define RED 13
#define GREEN 14
#define LOCK 12

const char* ssid = "Shubham"; //WiFi SSID
const char* password = "shubham18"; //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;
```

```cpp
  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

  // 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");
  }
}
```

OUTPUT: