Smart WIFI Door lock using ESP32 Cam and Telegram

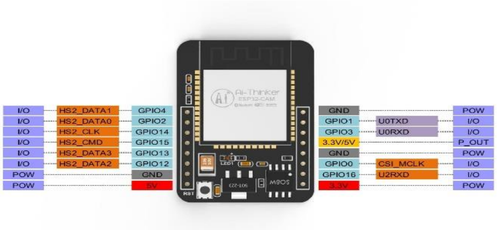
Objective:

My project is about opening and closing the door, if someone knocks the doorbell then the ESP32 CAM sends the user can capture the photo and photo is send to the telegram bot from esp32cam. Accordingly, the user can open or close the door with the help of the Telegram bot.

Components Required for this Project are:

1)ESP 32 CAM AI THINKER MODEL:

ESP 32 cam is a small size, low power consumption camera module based on ESP32. It comes with an OV2640 camera provides onboard TF Card slot. ESP32 Cam can be widely used in intelligent to IOT applications such as wireless video monitoring.



OV2640 camera: this is a 2MP camera attached to the microcontroller using FPC connector. This camera supports 8-bit compressed data with an image transfer rate of 15 to 60 fps. Outputs images in formats UXGA, SXGA, SVGA and any scaling down from SXGA to 40×30.

FTDI232:

It is used for uploading the code from Arduino IDE to the ESP32 CAM.

2)12 V Electric lock:

It used in my project as a locking and unlocking of door which works with 12 V supply.

3)TIP122 Transistor:

It is a Darlington braces NPN transistor. It works like an ordinary NPN transistor but it has a decent collector current assessment of nearby 5 amperes and its gain is around 1000.

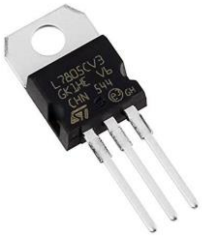
Pin	Type	
Pin#1	Emitter	Current comes out by the emitter it is usually linked to ground
Pin #2	Base	It governs biasing of the transistor and works to turn ON or OFF the Transistor
Pin #3	Collector	Current movements in

		over collector, usually linked to load

4)7805 5V Regulator:

7805 Voltage Regulator IC is a commonly used voltage regulator it provides a constant +5V output voltage for a variable input voltage Supply.

Pin Number	Pin Name	Description
1	Input(V+)	Unregulated Input Voltage
2	Ground(GND)	Connected to ground
3	Output(Vo)	Outputs Regulated +5V



5)1N4007 diode:

1N4007 is a rectifier diode, designed specifically for circuits that needs to convert alternating current to direct current. It can pass currents of up to 1 A , and have a peak inverse voltage(PIV) rating of 1000V.

6)1k and 10k ohm 0.25-watt Resistor:

A resistor is a passive electronic component, used with other electronic components to prevent or limit the flow of electrons through them. It works on the principle of Ohm's law which prevent overflow of voltage

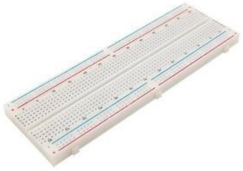
9) Push Button:

Electric Push Button are used with electric circuits, they contain the electronics required to make or break circuits, depending on the requirements of the circuit.

10)12 V DC supply Adapter and Female jack adapter:

For Connecting 12V DC power supply to the Breadborad we used Female jack . Female jack is having positive and negative terminal from this we can Connect To the bread board.

11) Breadboard:



A **breadboard** is an electronic tool which can be used to test electrical circuits. Instead of using solidering to connect wires and components together they can be stuck into the holes of the breadboard. It has metal strips inside that will connect them, and it lets them be removed easily or moved around when testing a circuit.is used to make the connection in the circuit.

12) Jumper wires:

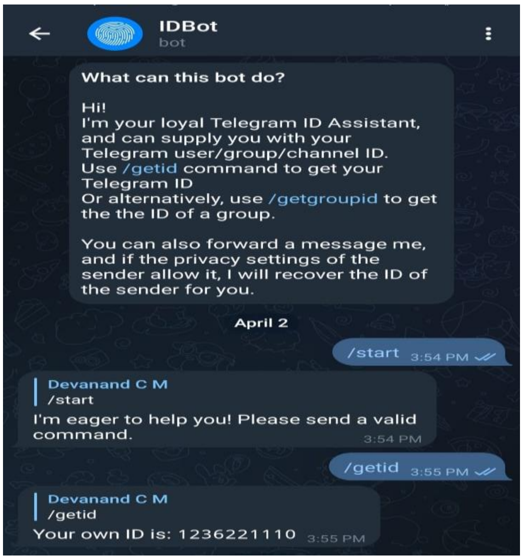
Jumper wires are used to make connections like TIP22, regulator etc., in the breadboard.

Procedure:

Step1:

Create Telegram Bot for the Project:

1. Search for Bot Father in the Telegram.
2. Tap on Start and then type **/newbot** and then enter.
3. Type the name for the bot. And then type a username ending with _bot
4. We will get token for our bot
5. Go to the IDBOT and get the ID of the telegram.



IDBot
bot

What can this bot do?

Hi!
I'm your loyal Telegram ID Assistant, and can supply you with your Telegram user/group/channel ID.
Use /getid command to get your Telegram ID
Or alternatively, use /getgroupid to get the the ID of a group.

You can also forward a message me, and if the privacy settings of the sender allow it, I will recover the ID of the sender for you.

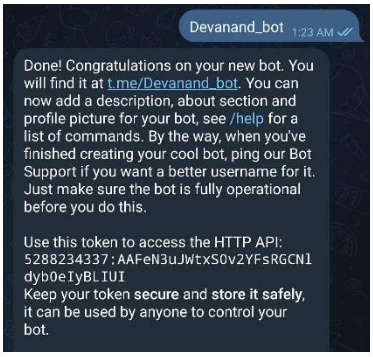
April 2

/start 3:54 PM

Devanand C M
/start
I'm eager to help you! Please send a valid command. 3:54 PM

/getid 3:55 PM

Devanand C M
/getid
Your own ID is: 1236221110 3:55 PM



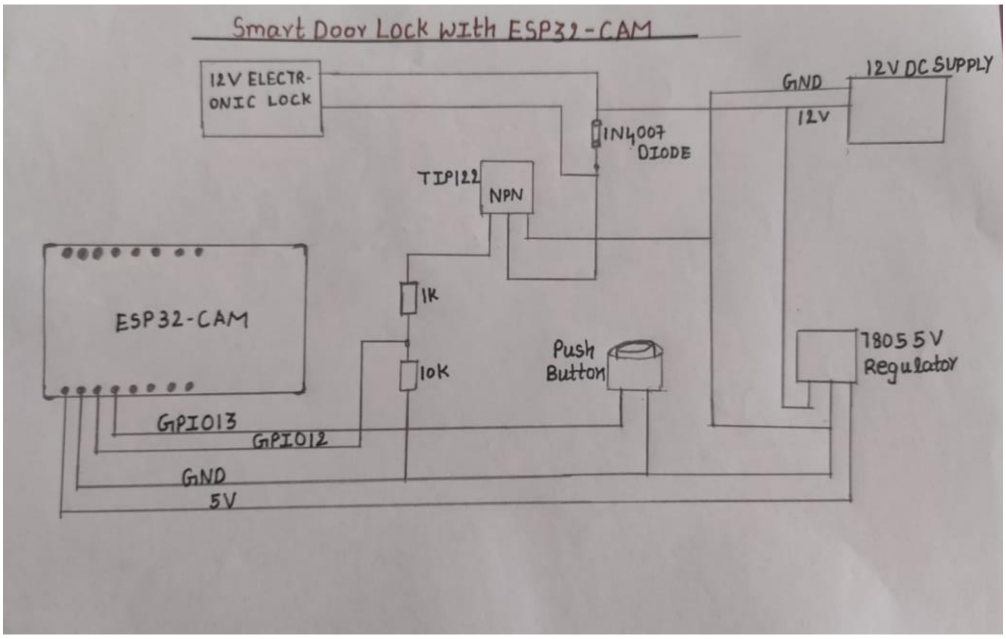
Devanand_bot 1:23 AM

Done! Congratulations on your new bot. You will find it at t.me/Devanand_bot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
5288234337:AAFeN3uJWtxSOv2YFsRGCNl
dyb0eIyBLIUI
Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

Step2:

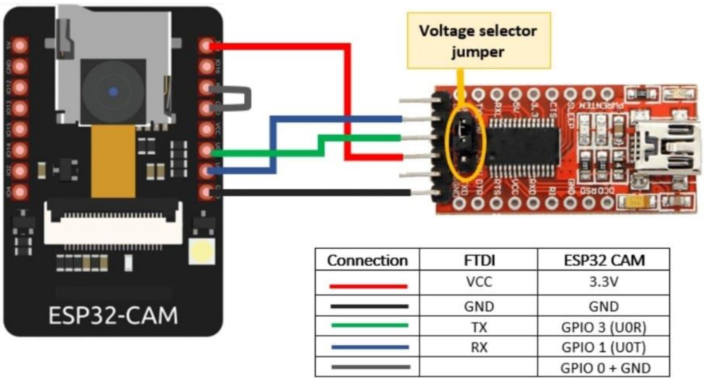
Circuit Diagram:



In the Circuit I have given 12 V volts as a Input with the help of the 7805 5V regulator converted into the 5volts and then Inputted this to the ESP 32 CAM.
I have used TIP 122 transistor to control 12 Volts electric lock from the GPIO12 and the push button is connected with the GPIO13 of ESP 32 cam.

STEP-3:
Upload the code to ESP 32 CAM and start the Telegram Bot:

We have to use the FTDI232 to program the ESP 32 cam.
We have to connect the ESP32 TO FTDI232 RxD to UDT , TxD to UDR, VCC to 5V , GND to GND and also we have to connect the GPIO 0 of ESP32CAM to GND of ESP32CAM.

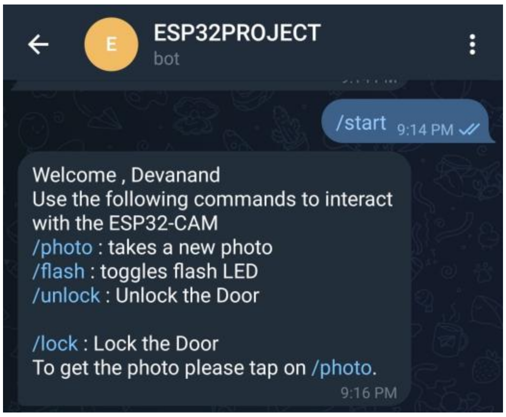


Connection	FTDI	ESP32 CAM
🟥	VCC	3.3V
⬛	GND	GND
🟩	TX	GPIO 3 (U0R)
🟦	RX	GPIO 1 (U0T)
▬		GPIO 0 + GND

For this code we require the telegram Library and the Arduino json library.
Token used in my Code is : 5288234337:AAFeN3uJWtxS0v2YFsRGCNldyb0eIyBLlUI
Telegram ID is : 1236221110

After all connection in the ESP32CAM. In the Arduino IDE we have to choose the ESP32CAM WROVER MODULE.

And we have to upload the code and after uploading we have to remove the GPIO 0 from GND connection. We have to open the serial monitor and set the braud rate 115220 and then it will connect to the wifi which were used in code and then we can the **start our bot in the telegram** it will show the **menu of our Circuit which includes Photo, Flash, Lock, Unlock options.**

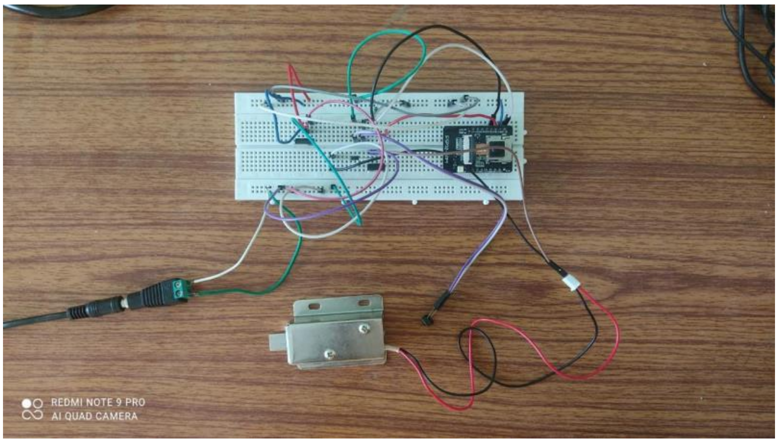


/start 9:14 PM

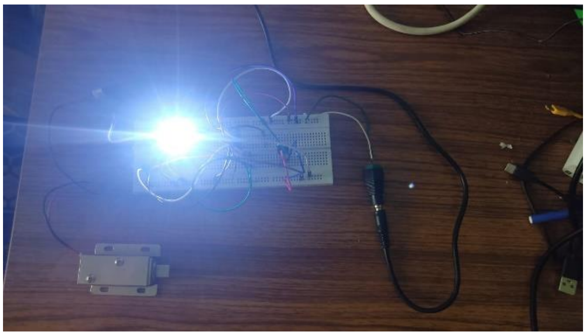
Welcome , Devanand
Use the following commands to interact with the ESP32-CAM
/photo : takes a new photo
/flash : toggles flash LED
/unlock : Unlock the Door

/lock : Lock the Door
To get the photo please tap on /photo.
9:16 PM

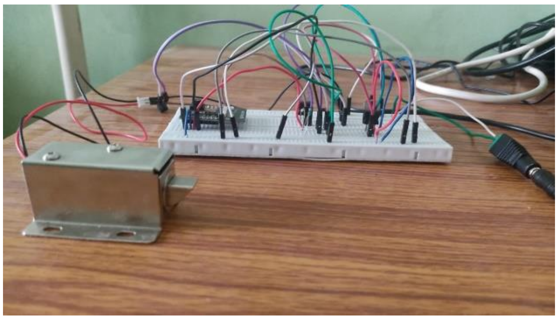
Working model:



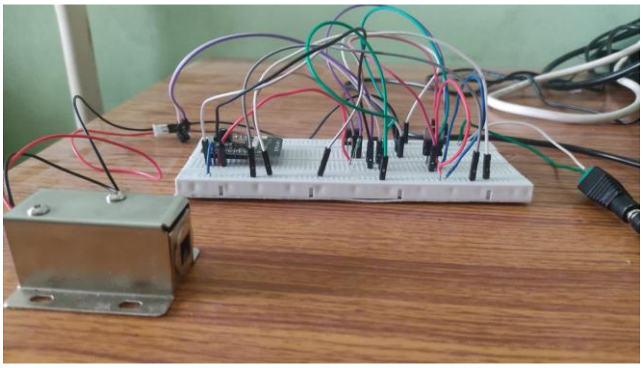
Circuit Image



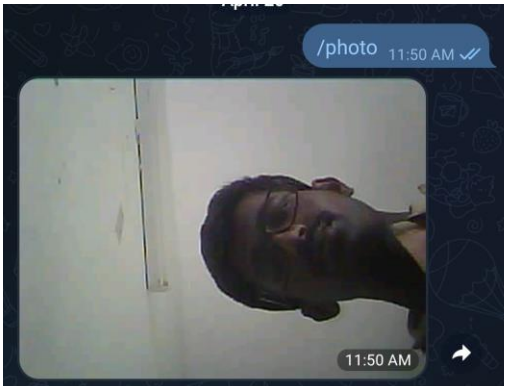
Flash mode of the Circuit



Lock State of the Circuit



Unlock state of the Circuit

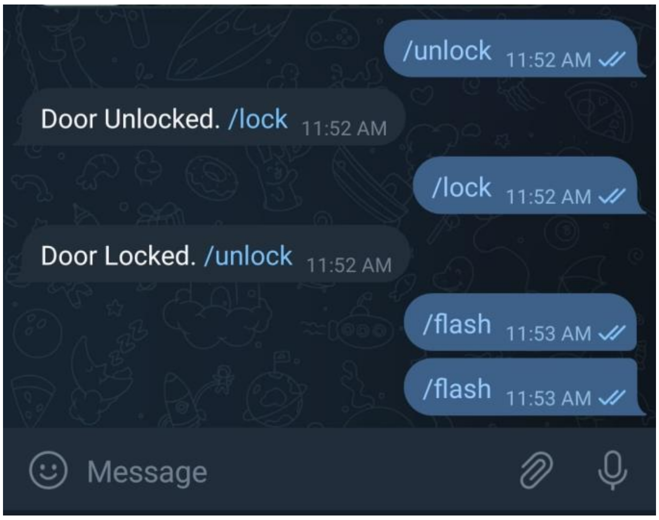


/photo 11:50 AM

11:50 AM

ESP32 CAM captured photo

and send it to Telegram.



/unlock 11:52 AM

Door Unlocked. /lock 11:52 AM

/lock 11:52 AM

Door Locked. /unlock 11:52 AM

/flash 11:53 AM

/flash 11:53 AM

Operating Circuit in Telegram

Working Flow Chart of the Circuit:

Power Supply ⇨ Regulator converts 12 V to 5V ⇨ ESP32CAM ⇨ Transistor ⇨ DIODE

⇩

Electric lock

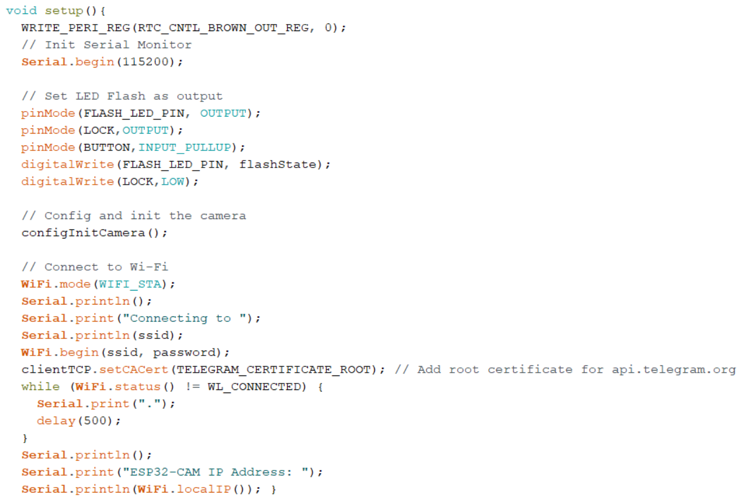
1.12V DC power supply Adapter with the help of jack Adapter connector.

and it is connected to breadboard from this power supply is done.

2. Since we require 5V for the ESP32 CAM we used Regulator for that which converts 12V to 5 V.

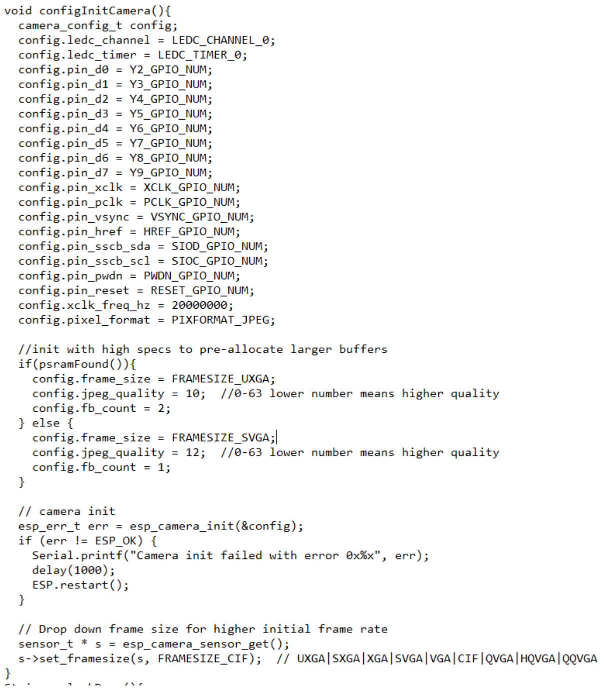
3. Since ESP32 CAM is connected to telegram bot according to the operation from the user ESP32 CAM sends the lock state to the TIP122 transistor and transistor insists the electric lock whether to close or open and also it can take photos and send it to bot. Diode is for converting AC current to DC current in the Process.

Code for the project:



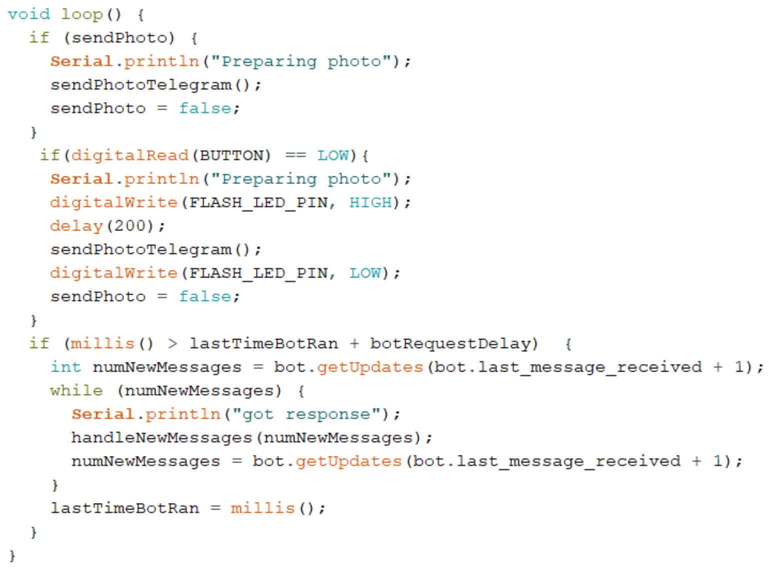
```
void setup(){
  WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0);
  // Init Serial Monitor
  Serial.begin(115200);

  // Set LED Flash as output
  pinMode(FLASH_LED_PIN, OUTPUT);
  pinMode(LOCK,OUTPUT);
  pinMode(BUTTON,INPUT_PULLUP);
  digitalWrite(FLASH_LED_PIN, flashState);
  digitalWrite(LOCK,LOW);

  // Config and init the camera
  configInitCamera();

  // Connect to Wi-Fi
  WiFi.mode(WIFI_STA);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  clientTCP.setCACert(TELEGRAM_CERTIFICATE_ROOT); // Add root certificate for api.telegram.org
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("ESP32-CAM IP Address: ");
  Serial.println(WiFi.localIP()); }
```

In setup we are configuring the camera pins and connecting to the wifi and Telegram



```
void configInitCamera(){
  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;  //0-63 lower number means higher quality
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;  //0-63 lower number means higher quality
    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);
    delay(1000);
    ESP.restart();
  }

  // Drop down frame size for higher initial frame rate
  sensor_t * s = esp_camera_sensor_get();
  s->set_framesize(s, FRAMESIZE_CIF);  // UXGA|SXGA|XGA|SVGA|VGA|CIF|QVGA|HQVGA|QQVGA
}
```

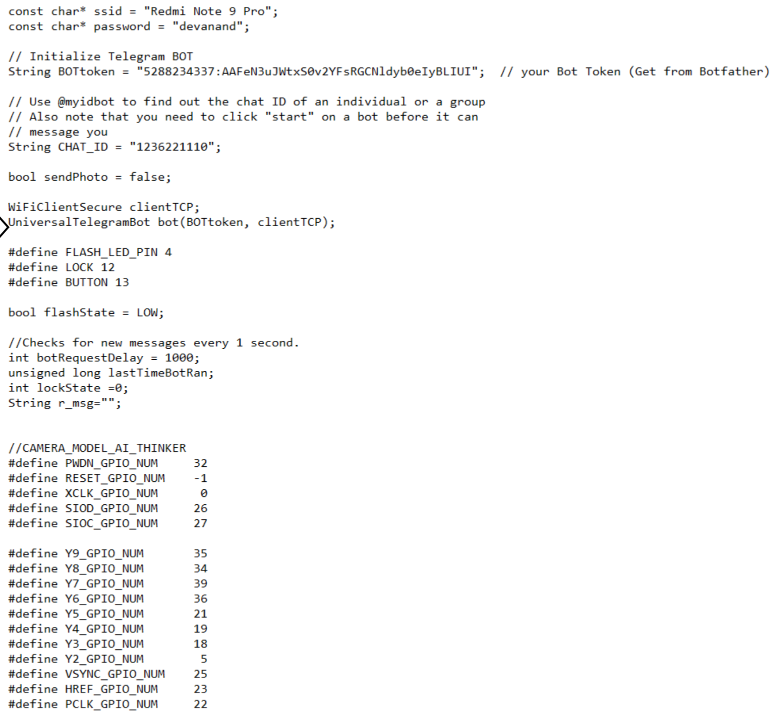
This is the function for configuring the camera Pins to the ESP32



```
void loop() {
  if (sendPhoto) {
    Serial.println("Preparing photo");
    sendPhotoTelegram();
    sendPhoto = false;
  }
   if(digitalRead(BUTTON) == LOW){
    Serial.println("Preparing photo");
    digitalWrite(FLASH_LED_PIN, HIGH);
    delay(200);
    sendPhotoTelegram();
    digitalWrite(FLASH_LED_PIN, LOW);
    sendPhoto = false;
  }
  if (millis() > lastTimeBotRan + botRequestDelay)  {
    int numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    while (numNewMessages) {
      Serial.println("got response");
      handleNewMessages(numNewMessages);
      numNewMessages = bot.getUpdates(bot.last_message_received + 1);
    }
    lastTimeBotRan = millis();
  }
}
```

This the code for loop for every message from telegram capture and send it to the handlenewmessage function in the code

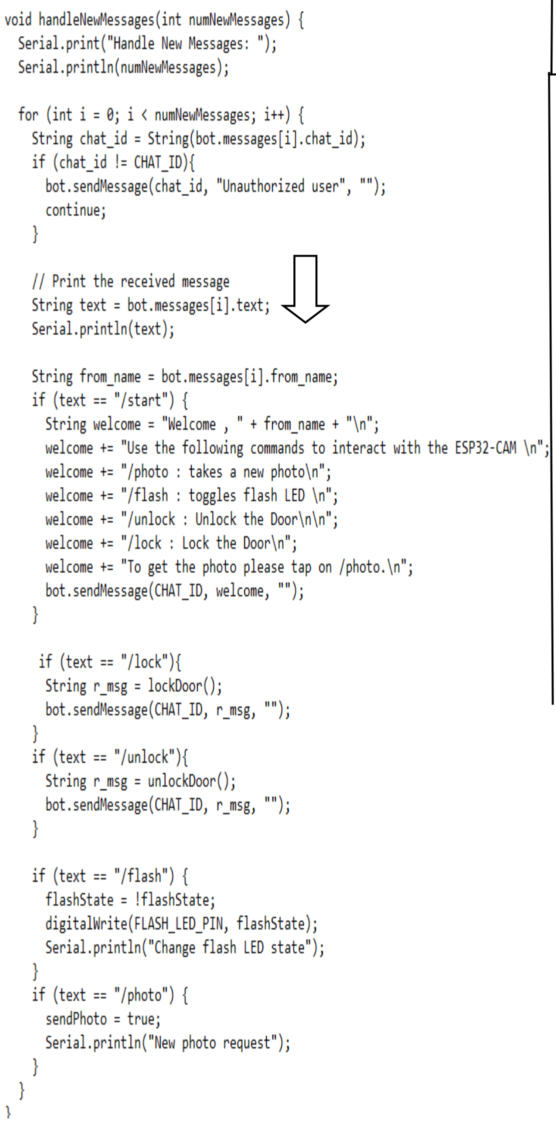
Code for connecting to wifi and defining the pins and writing the telegram Credentials and defining lock, button and flash certain number



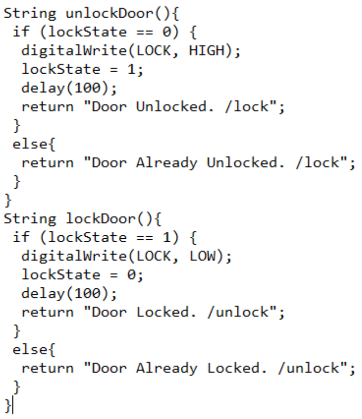
```
const char* ssid = "Redmi Note 9 Pro";
const char* password = "devanand";

// Initialize Telegram BOT
String BOTtoken = "5288234337:AAFeN3uJWtxS0v2YFsRGCNldyb0eIyBLIUI";  // your Bot Token (Get from Botfather)

// Use @myidbot to find out the chat ID of an individual or a group
// Also note that you need to click "start" on a bot before it can
// message you
String CHAT_ID = "1236221110";

bool sendPhoto = false;

WiFiClientSecure clientTCP;
UniversalTelegramBot bot(BOTtoken, clientTCP);

#define FLASH_LED_PIN 4
#define LOCK 12
#define BUTTON 13

bool flashState = LOW;

//Checks for new messages every 1 second.
int botRequestDelay = 1000;
unsigned long lastTimeBotRan;
int lockState =0;
String r_msg="";


//CAMERA_MODEL_AI_THINKER
#define PWDN_GPIO_NUM     32
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22
```

handleNewMessage:

In handlenewmessage it includes the function for every text which will come from the Telegram.



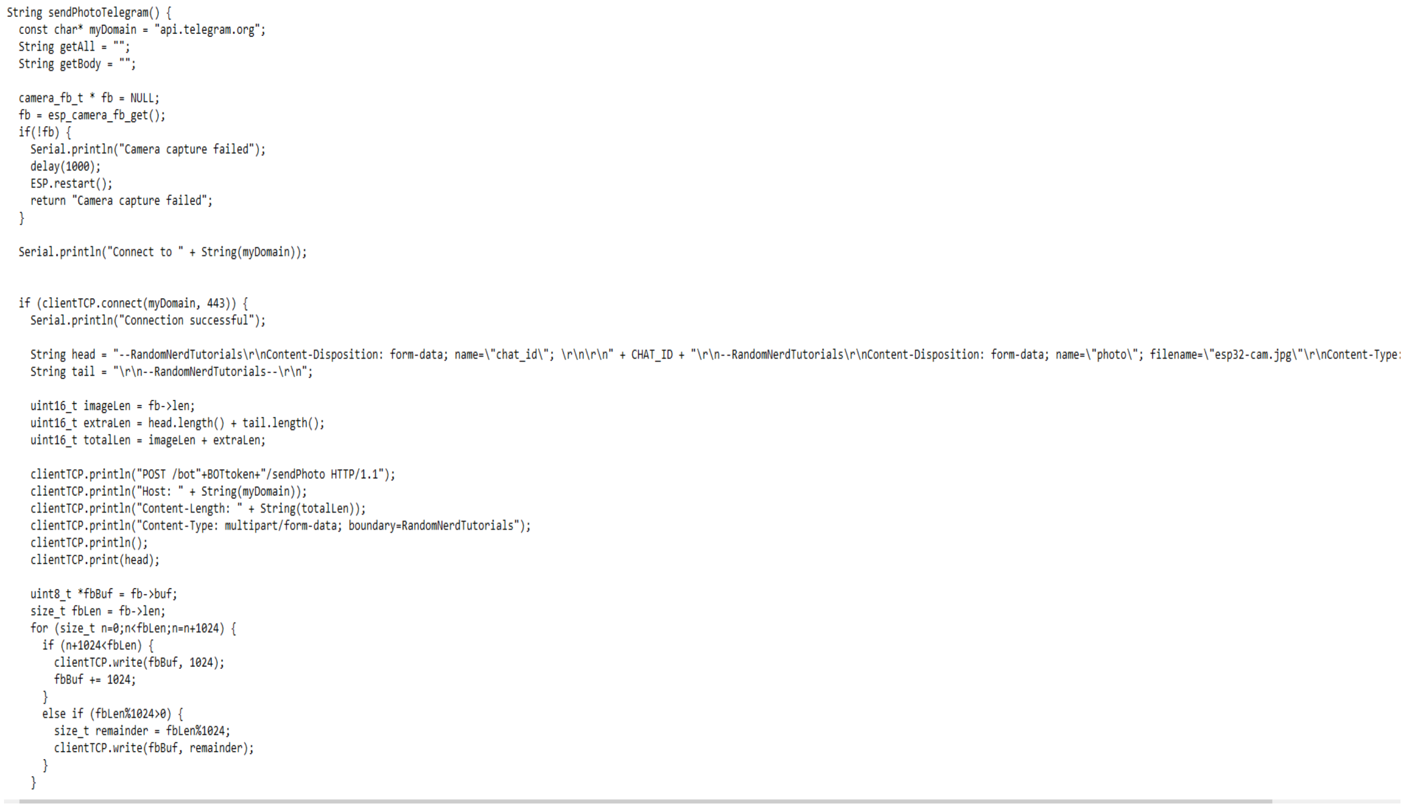
```
void handleNewMessages(int numNewMessages) {
  Serial.print("Handle New Messages: ");
  Serial.println(numNewMessages);

  for (int i = 0; i < numNewMessages; i++) {
    String chat_id = String(bot.messages[i].chat_id);
    if (chat_id != CHAT_ID){
      bot.sendMessage(chat_id, "Unauthorized user", "");
      continue;
    }

    // Print the received message
    String text = bot.messages[i].text;
    Serial.println(text);

    String from_name = bot.messages[i].from_name;
    if (text == "/start") {
      String welcome = "Welcome , " + from_name + "\n";
      welcome += "Use the following commands to interact with the ESP32-CAM \n";
      welcome += "/photo : takes a new photo\n";
      welcome += "/flash : toggles flash LED \n";
      welcome += "/unlock : Unlock the Door\n\n";
      welcome += "/lock : Lock the Door\n";
      welcome += "To get the photo please tap on /photo.\n";
      bot.sendMessage(CHAT_ID, welcome, "");
    }

     if (text == "/lock"){
      String r_msg = lockDoor();
      bot.sendMessage(CHAT_ID, r_msg, "");
    }
    if (text == "/unlock"){
      String r_msg = unlockDoor();
      bot.sendMessage(CHAT_ID, r_msg, "");
    }

    if (text == "/flash") {
      flashState = !flashState;
      digitalWrite(FLASH_LED_PIN, flashState);
      Serial.println("Change flash LED state");
    }
    if (text == "/photo") {
      sendPhoto = true;
      Serial.println("New photo request");
    }
  }
}
```



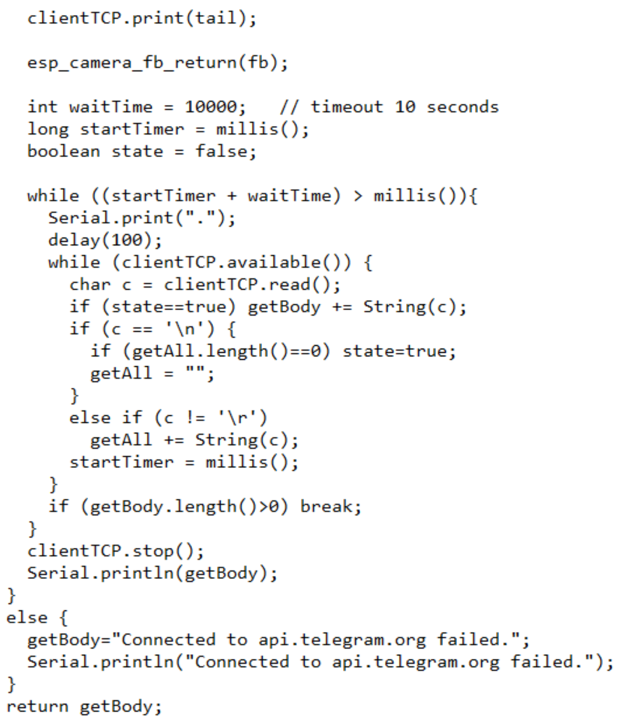
```
String unlockDoor(){
 if (lockState == 0) {
  digitalWrite(LOCK, HIGH);
  lockState = 1;
  delay(100);
  return "Door Unlocked. /lock";
 }
 else{
  return "Door Already Unlocked. /lock";
 }
}
String lockDoor(){
 if (lockState == 1) {
  digitalWrite(LOCK, LOW);
  lockState = 0;
  delay(100);
  return "Door Locked. /unlock";
 }
 else{
  return "Door Already Locked. /unlock";
 }
}
```

These two functions are for used to change lock state in the circuit which Is a part of handlenewmessage

Code for capturing the photo and send it to the Telegram Bot



```
String sendPhotoTelegram() {
  const char* myDomain = "api.telegram.org";
  String getAll = "";
  String getBody = "";

  camera_fb_t * fb = NULL;
  fb = esp_camera_fb_get();
  if(!fb) {
    Serial.println("Camera capture failed");
    delay(1000);
    ESP.restart();
    return "Camera capture failed";
  }

  Serial.println("Connect to " + String(myDomain));


  if (clientTCP.connect(myDomain, 443)) {
    Serial.println("Connection successful");

    String head = "--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"chat_id\"; \r\n\r\n" + CHAT_ID + "\r\n--RandomNerdTutorials\r\nContent-Disposition: form-data; name=\"photo\"; filename=\"esp32-cam.jpg\"\r\nContent-Type:
    String tail = "\r\n--RandomNerdTutorials--\r\n";

    uint16_t imageLen = fb->len;
    uint16_t extraLen = head.length() + tail.length();
    uint16_t totalLen = imageLen + extraLen;

    clientTCP.println("POST /bot"+BOTtoken+"/sendPhoto HTTP/1.1");
    clientTCP.println("Host: " + String(myDomain));
    clientTCP.println("Content-Length: " + String(totalLen));
    clientTCP.println("Content-Type: multipart/form-data; boundary=RandomNerdTutorials");
    clientTCP.println();
    clientTCP.print(head);

    uint8_t *fbBuf = fb->buf;
    size_t fbLen = fb->len;
    for (size_t n=0;n<fbLen;n=n+1024) {
      if (n+1024<fbLen) {
        clientTCP.write(fbBuf, 1024);
        fbBuf += 1024;
      }
      else if (fbLen%1024>0) {
        size_t remainder = fbLen%1024;
        clientTCP.write(fbBuf, remainder);
      }
    }
```



```
  clientTCP.print(tail);

  esp_camera_fb_return(fb);

  int waitTime = 10000;   // timeout 10 seconds
  long startTimer = millis();
  boolean state = false;

  while ((startTimer + waitTime) > millis()){
    Serial.print(".");
    delay(100);
    while (clientTCP.available()) {
      char c = clientTCP.read();
      if (state==true) getBody += String(c);
      if (c == '\n') {
        if (getAll.length()==0) state=true;
        getAll = "";
      }
      else if (c != '\r')
        getAll += String(c);
      startTimer = millis();
    }
    if (getBody.length()>0) break;
  }
  clientTCP.stop();
  Serial.println(getBody);
}
else {
  getBody="Connected to api.telegram.org failed.";
  Serial.println("Connected to api.telegram.org failed.");
}
return getBody;
```

Conclusion:

My project will help the people to know the person who is knocking before opening the door and also every time the doorbell rings person doesn’t need to manually open or close the door with the help of the WIFI it can be operated from Telegram for opening or Closing the Door Especially it is useful in Night Times.

Everyone can’t afford the Camera in-front of their door anyone can make this device and used in place of camera for their safety needs.