Secure Home Calling Bell

Sanjeeva Polepaka[1*], *R. P.* Ram Kumar[1], *P.* Likith Kumar[2], *Karthik* Vajja[2]

[1]Department of AIMLE, GRIET, Hyderabad, Telangana, India
[2]UG Student, Department of CSBS, GRIET, Hyderabad, Telangana, India

Abstract. The Secure Home Calling Bell focuses on home security, a very beneficial IoT application that they are leveraging to develop a low-cost security system for residential and commercial applications. An ESP32 cam, ESP32 board, a flame sensor, and a PIR sensor are the components of the project. When a visitor approaches the door, the secure house project's initial phase sends the user a notification that reads, "Someone's At Your Door." An ESp32 camera that would be connected to a calling bell for surveillance whenever the user's notification was received makes up the project's next step. An additional component of the IoT project is a flame sensor-based fire alarm system.

1 Introduction

The Internet of things, commonly known as IoT, is a technology bombardment in the fields of electronics and computing. The term IoT or Internet of Things refers to the collective network of interconnected devices and the technology that enables the communication between the devices and the cloud, and between the devices themselves.

1.1 History

When a group of college students planned to alter a Coca-Cola vending machine to allow for remote monitoring of its contents, the idea of embedding sensors and intelligence into physical things was first brought up. However, development was slow and the technology was cumbersome. Computer scientist Kevin Ashton first used the phrase "Internet of Things" in 1999. Ashton advocated putting radio-frequency identification (RFID) chips on items to follow them through a supply chain while he was employed with Procter & Gamble. He allegedly used the then-trending word "internet" in his presentation to catch the executives' eyes. And the saying persisted. As more connected devices entered the market over the following decade, public interest in IoT technology started to soar. As more connected devices entered the market over the following decade, public interest in IoT technology started to soar. The first smart refrigerator was unveiled by LG in 2000, the first iPhone was released in 2007, and by 2008, there were more linked gadgets than there were people on the earth. Google began testing autonomous cars in 2009, and in 2011, the company's Nest smart thermostat, which allowed for remote management of the heating system, was released.

[*] Corresponding author: sanjeeva1690@grietcollege.com

1.2 Working principle

The World Wide Web (Internet) connected us, changing the way we work and communicate, and the Internet of Things (IoT) is doing the same by simultaneously linking many things to the Internet. intends to advance this link. encourage interaction between machines and people. The guy who had this concept also realised that his IoT ecosystem had business uses in areas like home automation, automotive automation, industrial automation, medical, retail, healthcare, etc.

1.3 Challenges in building IoT- based systems

- As the number of connected devices increases and more information is exchanged between them, the potential for hackers to steal sensitive information also increases.
- An organization may have to deal with a huge number of(possibly millions) IoT devices, and it is difficult to collect and manage data from all these devices.
- Any device connected to the can be damaged if the system fails.
- The lack of an international standard for IoT compatibility makes it difficult for devices from different manufacturers to communicate with each other.
- The Internet of Things (IoT) has rapidly become a big part of people's lives, communications and businesses. Across the globe, Internet-enabled devices are transforming our global entitlement into a larger, empowered realm. IoT faces many kinds of challenges
- Encryption is a great way to prevent hackers from accessing your data, but it is also one of the major IoT security challenges.
- These drives are similar to the storage and processing capabilities found in traditional computers.
- As a result, there is an increasing number of attacks where hackers can easily manipulate algorithms designed to protect them.
- As IoT (Internet of Things) devices become more prevalent, IoT manufacturers strive to manufacture and ship the devices as quickly as possible but offer too little security. there is no. Most of these devices and IoT products have not received sufficient testing and updates and are vulnerable to hacking and other security issues.
- Weak credentials and credentials make almost all IoT devices vulnerable to password hacking and brute force. An organization that uses factory-default credentials on devices puts both its business and his assets, and its customers and their valuable information, at risk of being vulnerable to brute force attacks.
- Ransomware uses encryption to effectively lock users out of various devices and platforms but uses users' valuable data and information.
- Example – A Hacker can hijack your computer's camera and take pictures.
- Malware Access Points allow hackers to demand a ransom to unlock your device and return your data.
- IoT botnet operatives can compromise privacy and can pose significant risks to the open cryptocurrency market. The exact value and creation of the cryptocurrency code are compromised by malicious hackers. The blockchain company is looking to increase the security of its Blockchain technology itself is not particularly vulnerable, but the process of developing an app is.

2 Existing methods

A project proposal “IoT smart doorbell surveillance” that was released in August 2018 is one of the current approaches. The project was chosen to upgrade the current manual calling bell because, in most cases, these bells serve only to notify the resident that a visitor is present. When a visitor presses the calling bell, their design aims to be an addition to the current doorbell that assists in informing the users about the guests. The doorbell, the smart doorbell system, the wireless transmitter-receiver module, and the LCD response are the major components of the system architectural diagram.. The system's back end is depicted in the data flow diagram. When the doorbell is pressed, it demonstrates the data flow and procedure that occurs, as well as how the visitor receives an LCD response. We created a doorbell with an Arduino and a webcam that includes a visitor's photo in an SMS notice sent to the owner. The programming portion is delegated via the Pushing Box service. The Arduino makes an HTTP GET request to the Pushing Box API whenever the doorbell is pressed. Information is retrieved from a server via an HTTP GET request using the supplied URL.Data is retrieved through GET requests without being changed in any way. When Pushing Box receives a request, it starts the user's scenario and launches the web camera to take a picture, which is then transmitted back to the API. A SMS Notification is delivered to the owner's phone along with the visitor's image as soon as the image is obtained. The Pushing Box API is used for communication between the visitor and the owner. An HTTP request is issued in order to start a notification scenario. The device ID is the only argument needed with the request. A LCD Response is delivered to the visitor as the owner's response. Its drawback includes,

- The only additional feature of this concept is the buzzer sound that is activated when the calling bell switch is pressed. It lacks a sophisticated IoT circuit.
- Due to the lack of a camera in this project, we won't be able to see who is at the door. There aren't any additional features in this project, such as a voice recognition feature or a fire detection method.

2.1 Proposed solutions

- This standard calling bell is transformed into a high-tech security calling bell with the aid of IoT sensors and a Esp32 camera.
- A visitor’s notification is sent to the user's mobile device when the ESP8266 and Blynk app are used to take images of visitors at the door. Temperature sensors are used to monitor any fire occurrences outside the home. These sensors are attached to the Esp32 board and sound an alarm in the event of a fire incident outside the door.

3 Proposed method

Everyone now worries more about security, and protecting our homes has become more difficult lately. Security of the family and the home has grown to be more of a worry whether the owner is around or not. This specific issue must be resolved, at the very least.

3.1 Objectives of project

1. Leveraging IoT to create a trustworthy home security system
2. Delivering affordable protection for every home
3. Providing both personal and residence safety

4. The smart doorbell has an integrated high-definition infrared camera that allows the homeowner to utilise a smartphone to see who is at the door.
5. Smart doorbells also enable to detect any fire incidents around the house.

3.2 Architecture diagram

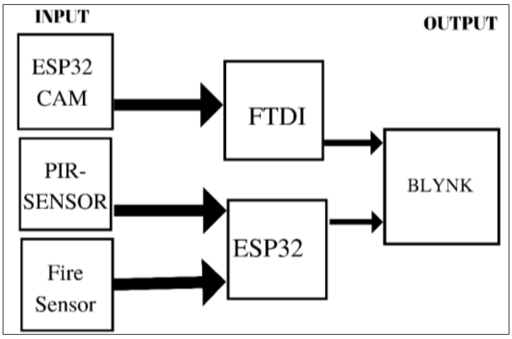


Fig. 1. Architecture diagram.

The esp32 microcontroller will be wired with the components solenoid lock and flame sensor whereas the esp32 cam is connected to FTDI before being connected via Wifi to the Blynk app. The Blynk app may be used to regulate the solenoid lock as well as outputs like alerting of any fire occurrence and showing images of the guests. The projects' working flow operates as shown in the previously mentioned.

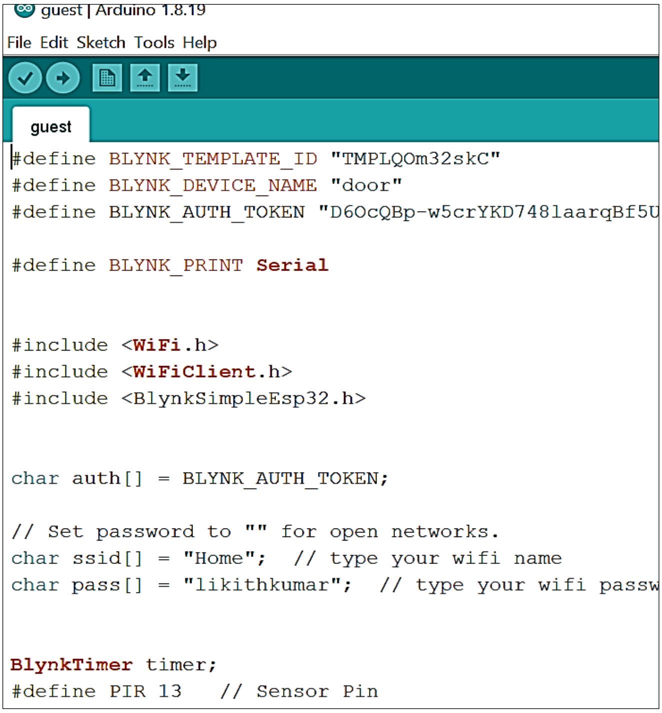
- An object's adjacent infrared light is recognized by a passive infrared (PIR) sensor. It's possible to presume that "passive" IR sensors are simpler than their active counterparts. In order to detect heat energy in the immediate vicinity, passive infrared (PIR) sensors use a pair of pyroelectric sensors. If the signal difference between these two sensors changes when they are put next to one another, the sensor will turn on. That might imply that it activates an alert, calls the police, or maybe turns on a floodlight.
- The ESP32-CAM is a fully operational microcontroller that has a built-in video camera and a microSD card slot. For Internet of Things (IoT) devices that require a camera with advanced functions like image tracking and identification, it is inexpensive, easy to use, and excellent. You may build a web-based camera with an elaborate control panel using the sketch included with Espressif's sample program. You'll discover that using the device is very simple once you master its programming.
- It is possible to detect flames thanks to a flame detector, which is a sensor designed to recognise and respond to the presence of a flame or fire. According to the installation, possible reactions to a flame detection include sounding an alarm or cutting off a gasoline supply and turning on a fire suppression system. They are used in industrial furnaces to verify that the furnace is operating as intended. They can also be used to cut off the ignition system, though they frequently do nothing more than alert the operator or control system.
- FTDI stands for Future Technology Devices International Limited, a privately held semiconductor device firm based in Scotland that specializes in Universal Serial Bus (USB) technology. In order to support old devices with contemporary computers, it develops, produces, and supports devices, as well as the connections and software drivers that go with them, for converting RS-232 or TTL serial communications to and from USB signals additionally, the business offers consultancy services for product

design, particularly in the field of electrical devices, as well as services for application-specific integrated circuit (ASIC) design.

- The ESP32 family of system on a chip micro controller is reasonably priced, has low power consumption, and has integrated Wi-Fi and dual-mode Bluetooth. The ESP32 series uses Tensilica Xtensa LX6 dual-core or single-core, Tensilica Xtensa LX7 dual-core, or a single-core RISC-V microprocessor, together with integrated antenna switches, RF baluns, power amplifiers, low-noise receive amplifiers, filters, and power-management modules . Chinese business Espressif Systems, with headquarters in Shanghai, invented and constructed the ESP32, which is produced by TSMC using their 40 nm technology. [2] It is the ESP8266 microcontroller's replacement.
- Blynk allows you to create smartphone applications that facilitate communication with microcontrollers or even whole computers like the Raspberry Pi. The primary objective of the Blynk platform is to make developing mobile phone applications exceedingly simple. This course will teach you how to easily create a mobile application that can communicate with your Arduino by dragging a widget and setting up a pin. With Blynk, you can operate an LED or a motor with essentially no programming from your smartphone. Actually, the first experiment I'll show you in this course is this one.

4 Results and discussions

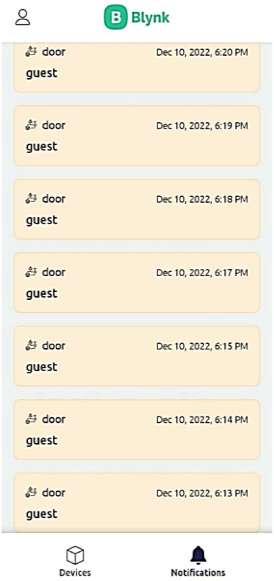
4.1 PIR sensor module results



```
#define BLYNK_TEMPLATE_ID "TMPLQOm32skC"
#define BLYNK_DEVICE_NAME "door"
#define BLYNK_AUTH_TOKEN "D6OcQBp-w5crYKD748laarqBf5U

#define BLYNK_PRINT Serial


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


char auth[] = BLYNK_AUTH_TOKEN;

// Set password to "" for open networks.
char ssid[] = "Home";  // type your wifi name
char pass[] = "likithkumar";  // type your wifi passw


BlynkTimer timer;
#define PIR 13   // Sensor Pin
```

Fig. 2. PIR Sensor Arduino code.

When the ESP32 executes the Arduino code of the PIR Sensor, it produces the result described in Fig. 2 PIR Sensor Arduino code.



Blynk

door Dec 10, 2022, 6:20 PM
guest

door Dec 10, 2022, 6:19 PM
guest

door Dec 10, 2022, 6:18 PM
guest

door Dec 10, 2022, 6:17 PM
guest

door Dec 10, 2022, 6:15 PM
guest

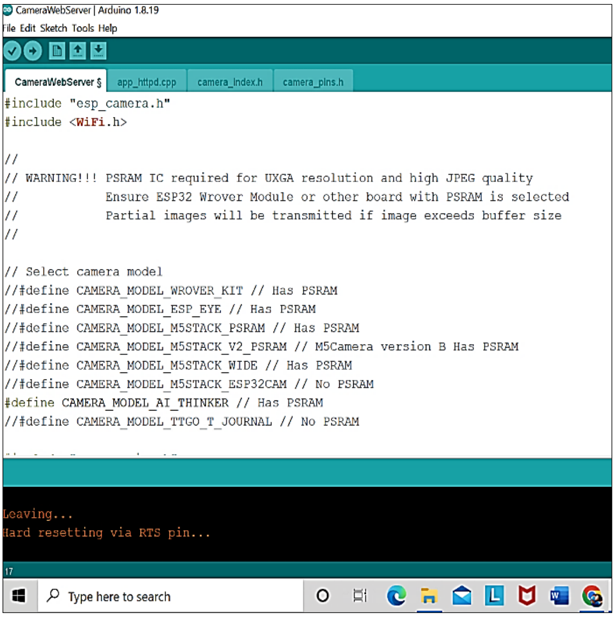
door Dec 10, 2022, 6:14 PM
guest

door Dec 10, 2022, 6:13 PM
guest

Fig. 3. PIR Sensor output on Blynk app.

When a PIR Sensor detects a visitor at the door, it notifies the user using the Blynk app as mentioned above Fig 3 PIR Sensor Output on Blynk app, which is already installed on the user's mobile device. To learn more about the guest at the door, the user is instructed to use the ESP32 CAM model by way of the notification.

4.2 ESP32 cam module results

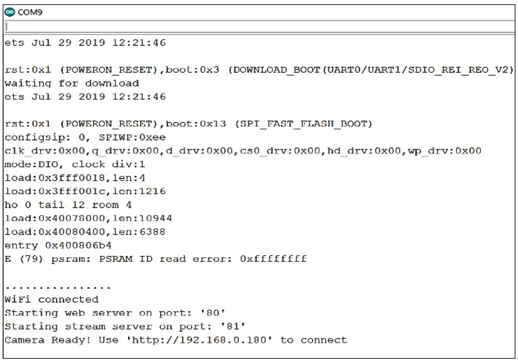


```
#include "esp_camera.h"
#include <WiFi.h>

//
// 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 // Has PSRAM
//#define CAMERA_MODEL_ESP_EYE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_PSRAM // Has PSRAM
//#define CAMERA_MODEL_M5STACK_V2_PSRAM // M5Camera version B Has PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE // Has PSRAM
//#define CAMERA_MODEL_M5STACK_ESP32CAM // No PSRAM
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
//#define CAMERA_MODEL_TTGO_T_JOURNAL // No PSRAM
```

```
Leaving...
Hard resetting via RTS pin...
```

Fig. 4. ESP32 cam Arduino code.

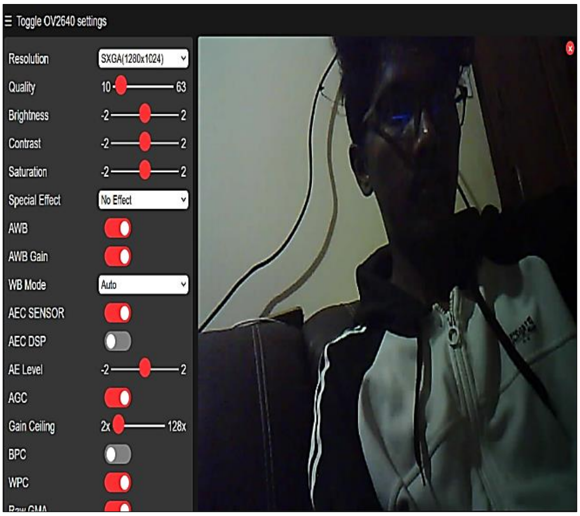
When the ESP32 executes the Arduino code of the ESP32 Cam Code, it produces the result “Hard resetting Via RTS pin” as described in Fig. 4 ESP32 Cam Arduino code.



```
ets Jul 29 2019 12:21:46

rst:0x1 (POWERON_RESET),boot:0x3 (DOWNLOAD_BOOT(UART0/UART1/SDIO_REI_REO_V2))
waiting for download
ets Jul 29 2019 12:21:46

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:10944
load:0x40080400,len:6388
entry 0x400806b4
E (79) psram: PSRAM ID read error: 0xffffffff

................
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.0.180' to connect
```

Fig. 5. ESP32 cam COM results.

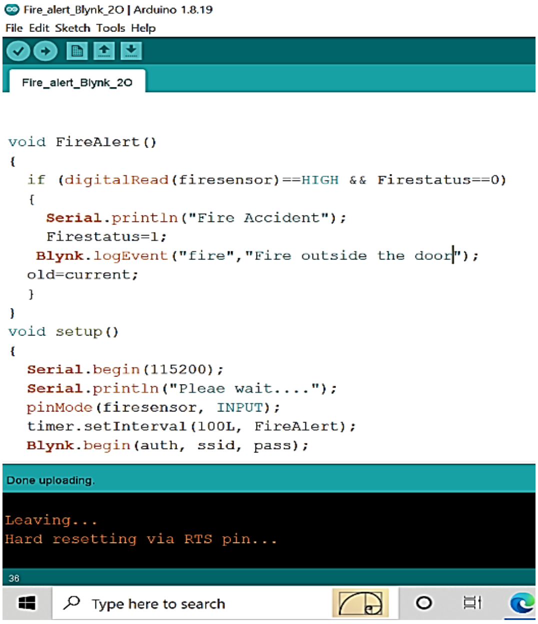
After the project is successfully completed, the COM 9 results of the Esp32 cam are provided, as illustrated in Fig. 5 above. COM ESP32 Cam Results for ESP32 Cam web server connection to any device. When the code is successfully uploaded and the port (COM 9) in the Arduino displays the results as seen in the above Fig 6 ESP32 Cam Working Results, the results of the cam module can be seen. We will receive the ESP32 CAM's HTTP link from the port, and we may use it anywhere by pasting the link into a browser or using the instructions from the Blynk app. When the ESP32 executes the Arduino code of the Flame Sensor, it produces the result “Hard resetting Via RTS pin” as described in Fig. 7 Flame Sensor Arduino Code. Once the connection is made successfully, the red light on the flame sensor will illuminate, signaling the connections are made precisely. Every time a flame appears close, the flame sensor detects it and sends a notice stating "fire accident" to the user's installed Blynk app over wi-fi. According to Fig. 8 Flame Sensor Com findings, the COM12 outputs "Fire Accident" after the code is successfully executed.



Resolution SXGA(1280x1024)
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
AGC ●
Gain Ceiling 2x – 128x
BPC ○
WPC ●
Raw GMA ●

Fig. 6. ESP32 Cam Working Results

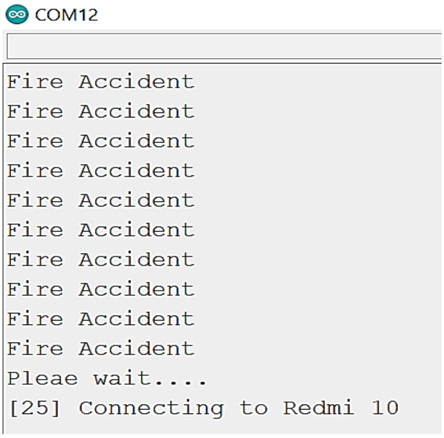
4.3 Flame sensor result


Fire_alert_Blynk_20 | Arduino 1.8.19
File Edit Sketch Tools Help
Fire_alert_Blynk_20
Type here to search

```
void FireAlert()
{
  if (digitalRead(firesensor)==HIGH && Firestatus==0)
  {
    Serial.println("Fire Accident");
    Firestatus=1;
   Blynk.logEvent("fire","Fire outside the door");
  old=current;
  }
}
void setup()
{
  Serial.begin(115200);
  Serial.println("Pleae wait....");
  pinMode(firesensor, INPUT);
  timer.setInterval(100L, FireAlert);
  Blynk.begin(auth, ssid, pass);
```

```
Done uploading.

Leaving...
Hard resetting via RTS pin...
```

Fig 7. Flame Sensor Arduino code.



```
Fire Accident
Fire Accident
Fire Accident
Fire Accident
Fire Accident
Fire Accident
Fire Accident
Fire Accident
Fire Accident
Fire Accident
Pleae wait....
[25] Connecting to Redmi 10
```

Fig 8. Flame Sensor Com results.

4.4 Significance of proposed method

Everyone is now more concerned about security, and recently, protecting our homes has gotten harder. Whether the owner is home or not, security of the family and home has become a bigger concern. With the help of an esp32 camera and an automated solenoid lock, which can be operated with a single click on a smartphone using the Blynk app, the current module enables users to recognize guests at the door. This module's flame detection capability enables users to identify flames, whether they are outside or inside the door, and alerts them via a notification sent to the registered phone number via the Blynk application.

5 Conclusion

The security of our homes is crucial to our daily life. These days, break-ins are incredibly common. No one ought to leave their house undefended. So it's essential to take safety precautions and strengthen your home security system. The Internet of Things can help with home security. Many various IoT gadgets are useful for keeping an eye on questionable behaviour. When you are away, there is a good probability that a thief may break into your home. Most thefts take place while home owners are away. You can quickly and effectively monitor your home using connected smart home security equipment. The Secure Home Calling Bell uses the Internet of Things to build an affordable security system for residential and commercial use that focuses on home security, which is a particularly valuable application of IoT. An ESP32 cam, an ESP32 board, and a Flame sensor make up the project. When a visitor rings the doorbell, the secure home project's initial phase of operation sends a snapshot of the visitor to the user's registered mobile number along with the message "Someone's At Your Door." An automatic door lock open system is part of the project's next phase. When a user recognizes a visitor, they can use their mobile devices to quickly and easily open the door lock.

References

1. IoT and Cloud Computing, https://www.geeksforgeeks.org/iot-and-cloud-computing/
2. What is the Internet of Things, https://www.techtarget.com/iotagenda/definition/Internet-of-Things-IoT
3. Types of Sensors, https://www.variohm.com/news-media/technical-blog-archive/types-of-sensors
4. Different Types of Sensors with Applications, https://www.electricaltechnology.org/2018/11/types-sensors-applications.html
5. ESP32 Cam Smart IoT bell, https://www.electroniclinic.com/esp32-cam-smart-iot-bell-circuit-diagram-and programming/ESP32_CAM_IoT_Bell_Complete_Code
6. ESP32 CAM Smart IoT Bell Circuit diagram, https://www.electroniclinic.com/esp32-cam-smart-iot-bell-circuit-diagram-and-programming/ #ESP32_CAM_IoT_Bell_Complete_Code
7. C. Fiorenza, S. Bompelli, M. Sampath, M. Ghogre, A. Ajithkumar, Int. J. of Adv. Res. and Innov. Ideas in Edu. **4**, 2, (2018)
8. Flame or Fire Sensor Working, https://microcontrollerslab.com/iot-fire-detection-esp32-flame-sensor-email-alert/
9. How to Detect a Fire with a Flame Sensor for Microcontroller, https://diyi0t.com/flame-sensor-arduino-esp8266-esp32/
10. ESP32-CAMVideoStreaming, https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-ardui no-ide/
11. Intelligent Flame Detection, https://www.robotique.tech/robotics/intelligent-flame-detection-system-with-esp32/
12. V. S. Lalitha, Kayiram Kavitha, N. V. Krishna Rao, G. Rama Mounika, V. Sandhya, Int. J. of Innov. Tech. and Expl. Eng. **8**, 8, (2019)