

Network-Based Door Locking System

OBJECTIVE

The objective of the project, "Network-Based Door Locking System," is to design and implement a secure and efficient door locking system that operates through a SCADA networks. The system aims to enhance the security of physical access control by providing reliable and convenient door access management using network-based technology.

ABSTRACT

The project aims to develop a smart door access control system that leverages image processing, database querying, and distance sensing technology. Users can send a picture to a server, which performs analysis in real-time to verify the identity and grant access if authorized. Additionally, an ultrasonic sensor is used to detect the distance between the user and the door. When the distance is less than 10 cm, the system activates a camera to capture an image for further verification. The server compares the captured image with the database and, upon successful authentication, opens the door. This integrated solution provides efficient and secure access control by utilizing real-time image analysis and distance sensing capabilities

SYSTEM OVERVIEW & METHODOLOGY

System overview

The project incorporates a Supervisory Control and Data Acquisition (SCADA) system, employing a client-server architecture for effective control and monitoring of a door lock. The system consists of a central server, implemented on a laptop, and two remote clients based on Raspberry Pi devices. These clients are equipped with ultrasonic sensors, cameras, solenoid locks, and buzzers.

The SCADA system follows a typical architecture where the central server acts as the main control unit, responsible for data processing and system management. The remote clients serve as distributed units that collect data from the ultrasonic sensors and cameras. They then transmit this data to the server for real-time analysis and control of the door lock.

The server's role encompasses tasks such as data aggregation, monitoring the status of the door lock, and making decisions based on the received data. It can activate the solenoid lock or trigger the buzzer as necessary, providing efficient access control.

This client-server architecture enables seamless communication and coordination between the central server and remote clients, facilitating real-time monitoring and control of the door lock. By leveraging the SCADA system, the project achieves enhanced security and convenience in door access management.

Hardware components:

A. Raspberry Pi 4:

The Raspberry Pi 4 is equipped with a quad-core 64-bit Broadcom BCM2711 ARM Cortex-A72 SoC processor running at 1.5 GHz.

SPECIFICATIONS:

1. Clock frequency: 1.5 GHz
2. Chipset (SoC): Broadcom BCM2711
3. Processor: 64-bit quad-core ARM Cortex-A72
4. Graphics processor: Broadcom Dual Core VideoCore IV (OpenGL ES 2.0, H.264 Full HD @ 30 fps)
5. Memory (SDRAM): 1GB, 2GB, 4GB or 8GB LPDDR4-3200 SDRAM
6. Number of USB 2.0 ports: 2
7. Number of USB 3.0 ports: 2
8. Port extension: 40-pin GPIO
9. Video outputs: HDMI and RCA, plus 1 CSI camera connector
10. Audio outputs: 3.5 mm stereo jack or HDMI
11. Data storage: MicroSD card
12. Network connection: 10/100 Ethernet, 802.11n WiFi and Bluetooth 4.1 (BLE - Low Energy)
13. Peripherals: 17 x GPIO
14. Supply: 5V 2.5A via micro USB

B. Pi Camera Module:

The Raspberry Pi Camera Board is a custom designed add-on module for Raspberry Pi hardware. It attaches to Raspberry Pi hardware through a custom CSI interface. The sensor has a 5-megapixel native resolution in still capture mode.

C. Micro USB Cable:

The first, recommended and easiest way to power the Raspberry Pi is via the Micro USB port on the side of the unit. The recommended input voltage is 5V, and the recommended input current is 2A.

D. Relay Module:

A 5v relay is an automatic switch that is commonly used in an automatic control circuit and to control a high-current using a low-current signal. The input voltage of the relay signal ranges from 0 to 5V.

Pin COM (Common): This pin is connected to the main terminal of the Load to make it active.

Pin NC (Normally Closed): This second terminal of the load is connected to either NC/ NO pins. If this pin is connected to the load, then it will be ON before the switch.

Pin NO (Normally Open): If the second terminal of the load is allied to the NO pin, then the load will be turned off before the switch.

E. Solenoid Lock:

Raspberry Pi and Solenoid Lock are connected through the relay module. Solenoid lock requires 9 to 12V, and Raspberry pi can provide only 5V. Due to this, a 12V adapter is used to power the Solenoid Lock.

F. Ultrasonic Sensor:

The ultrasonic sensor in the project is connected to the Raspberry Pi devices. It operates on the

Principle of emitting ultrasonic waves and measuring the time it takes for the waves to bounce back after hitting an object. This allows the sensor to calculate the distance between the sensor and the object.

Methodology:

- A. **TCP Socket:** By using a TCP socket, the door lock system can facilitate secure and reliable communication between the door lock devices and the control system.
- B. **Distance Measurement:** The ultrasonic sensor measures the distance between the user and the door.
- C. **Buzzer:** it goes high when the ultrasonic sensor senses below 10 cm
- D. **Image Capture:** The client device captures an image of the user requesting access to the door and stores in a local path.
- E. **C .Data Transmission:** The client device sends the distance measurement data to the server for further processing by the user.
- F. **D.I mage Analysis:** The server accesses the image via SSH FTP and Uploads on a User-friendly interface for identification.
- G. **E. Database:** The server stores the extracted details in a database keeping track of the sensed time, Distance, Image and Socket Address of the client.
- H. **Access Decision:** Based on the server authorization decision for door access, a message is sent to the client to via TCP socket to open/not open the door.

- I. **Solenoid:** If the user is authorized and the distance measured by the ultrasonic sensor is less than 10 cm, the server activates the door opening mechanism.

Block diagram

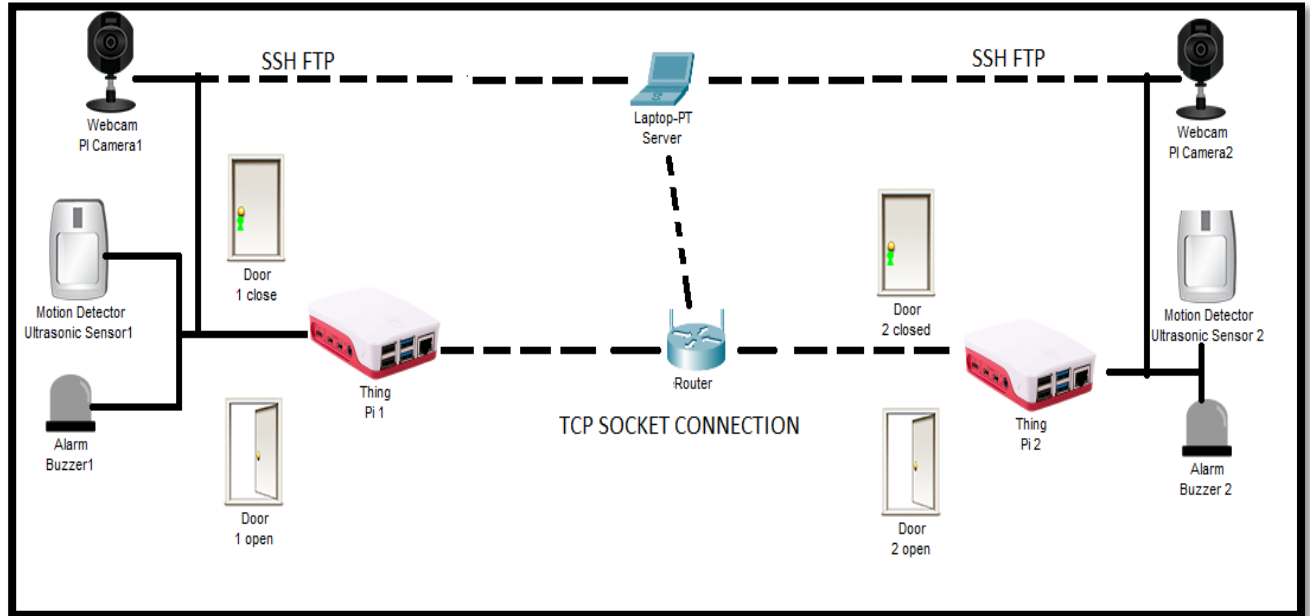


Figure 1:BLOCK DIAGRAM

IMPLEMENTATION

- Connect relay, solenoid, buzzers, cameras, and ultrasonic sensors to the Raspberry Pi clients.
- Create a server application on the laptop using socket programming.
- Create a server socket and wait for connections from clients.
- Use multithreading to handle multiple clients at once.
- Use an ultrasonic sensor to measure the distance to the Raspberry Pi client.
- Use a buzzer to provide alarms or notifications when a user is within 10 cm.
- Configure the Raspberry Pi to take a picture of the user and store it in a local directory while also sending the distance measurements to the server.
- Connecting to the server using TCP/IP is the first step.
- Set up the Raspberry Pi to keep an eye out for button press events. Access the image file stored in the client that made a connection via SSH FTP. The IP address, username, password and image file directory of the client are to be declared.
- Create a user interface on the server with button controls to send the client control signals to operate door lock mechanism. Use the client Relays to control the Solenoid based on the incoming message from the server.
- Create an Excel File to store the extracted details -sensed time, Distance and Socket Address of the client in a database. Store the images.
- Test the communication between the clients and the server.

RESULTS

```
The current date and time is 2023-06-14 09:21:46.253699  
Door monitoring server is observing at 192.168.43.115:1234
```

Figure 2:SERVER BACKEND OUTPUT

```
Distance monitoring client connected to  
192.168.43.115:1234
```

```
Distance: 467.74 cm  
Distance: 467.86 cm  
Distance: 55.34 cm  
Distance: 7.33 cm
```

Figure3:CLIENT 1 OUTPUT

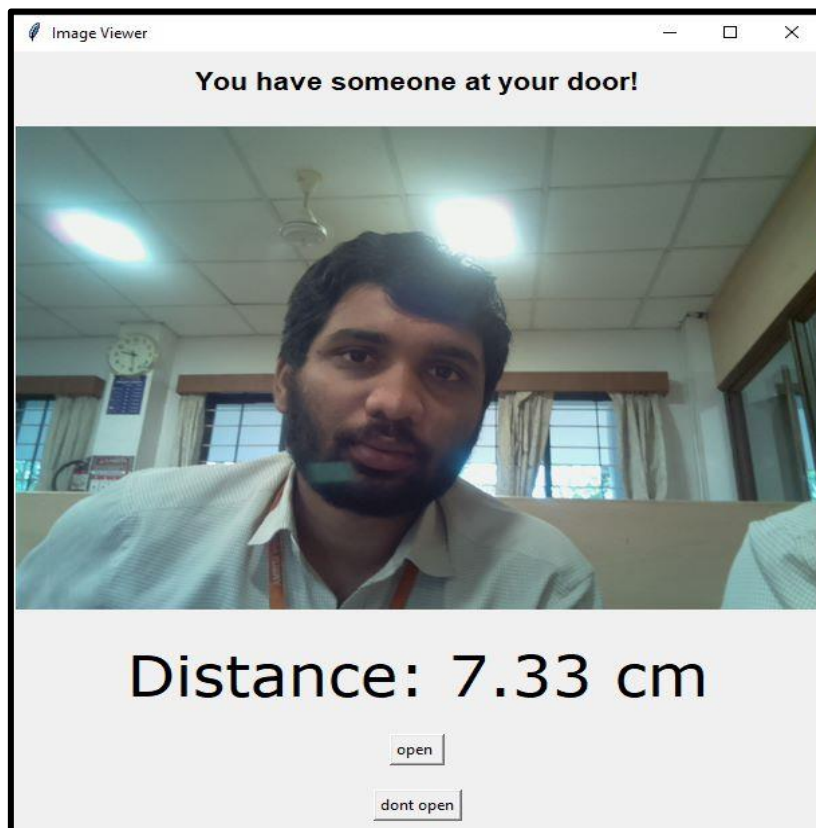


Figure 4:USER INTERFACE CLIENT 1

```
Distance monitoring client connected to  
192.168.43.115:1234
```

```
Distance: 467.74 cm
```

```
Distance: 467.86 cm
```

```
Distance: 55.34 cm
```

```
Distance: 7.33 cm
```

```
Distance: 7.33 cm
```

```
Received user message from server: 1
```

Figure5: message received from server that door is opened

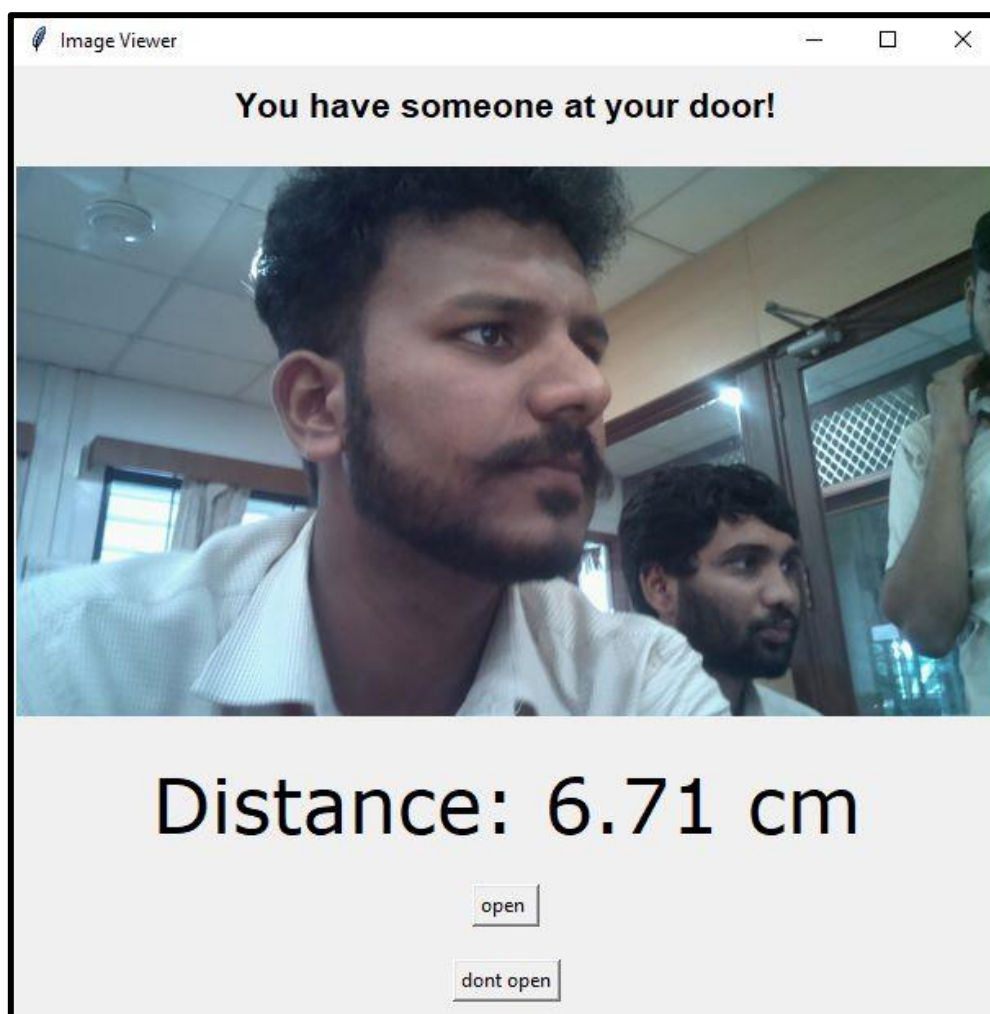


Figure 6: USER INTERFACE CLIENT 2

connection1.txt - Notepad
File Edit Format View Help
2023-06-14 10:46:56.677386
('192.168.43.152', 36998)
Distance: 7.11 cm
2023-06-14 10:46:56.677386
('192.168.43.152', 36998)

2023-06-14 10:48:55.811186
('192.168.43.44', 52752)
Distance: 4.5 cm
2023-06-14 10:48:55.811186
('192.168.43.152', 45158)
Distance: 7.08 cm
2023-06-14 10:51:12.929626
('192.168.43.152', 38440)
Distance: 6.01 cm
2023-06-14 10:51:12.929626
('192.168.43.44', 39504)
Distance: 4.75 cm

2023-06-14 10:53:05.536890
('192.168.43.152', 53330)
Distance: 6.47 cm
2023-06-14 10:53:05.536890
('192.168.43.44', 55516)
Distance: 3.71 cm
2023-06-14 10:56:56.196072
('192.168.43.44', 54552)
Distance: 8.41 cm
2023-06-14 10:56:56.196072
('192.168.43.44', 54552)

2023-06-14 10:59:07.583296
('192.168.43.44', 46168)
Distance: 6.71 cm
2023-06-14 10:59:07.583296
('192.168.43.44', 46168)

Figure7:DATABASE

CONCLUSION

The implemented network door-based system utilizes Raspberry Pi clients and a server application. By connecting various components and employing socket programming, it enables real-time monitoring and control of door access. Ultrasonic sensors measure distance, activating a buzzer when a user is within 10 cm. Raspberry Pi captures user images and sends data to the server. The server provides a user interface for control signals to operate the door lock mechanism. Excel files store important details, including sensed time and distance. This system offers an efficient solution for enhancing door security in diverse environments.

Name:
NIREK SARAVANAN
SANKRANTI SRIKAR
SHOGAN SIDHARATH