Election Voting Machine Using Fingerprint Scanner, ESP32, Arduino, and Real-Time React Web Page

Team Leader: Brijesh Nishad Contact :9506563150

September 25, 2024

Contents

Chapter 1

Introduction

This document outlines the design, implementation, and setup of an election voting machine that includes fingerprint authentication, candidate selection, and real-time vote monitoring through a web interface built with React. The hardware components consist of a fingerprint scanner, ESP32 microcontroller, and Arduino, while the software architecture allows for real-time updates.

Chapter 2

System Overview

2.1 Hardware Components

The election voting machine utilizes the following components:

- **Fingerprint Scanner** - Used for voter authentication.
- **Arduino** - Handles the fingerprint scanner input and candidate selection.
- **ESP32** - Sends data to a real-time web server for vote tracking.
- **React Web Page** - Displays voting results in real-time.

2.2 Software Components

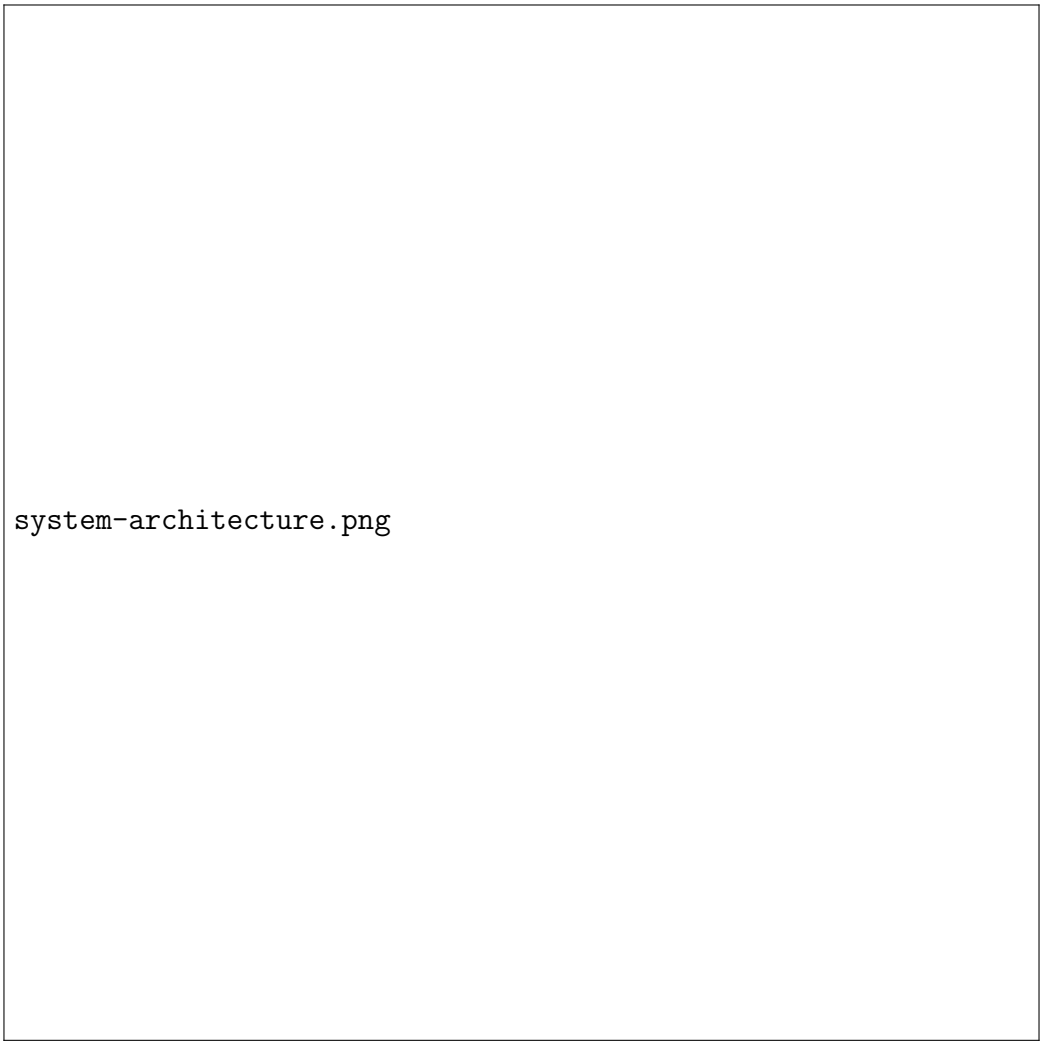
The software stack includes:

- Arduino IDE for programming the fingerprint scanner and candidate selection.
- WebSocket communication to transfer votes between ESP32 and the server.
- React.js for real-time display of voting results.

Chapter 3

System Architecture

The system is divided into three major parts: hardware, communication layer (ESP32), and the web interface. Below is a basic overview of the architecture.

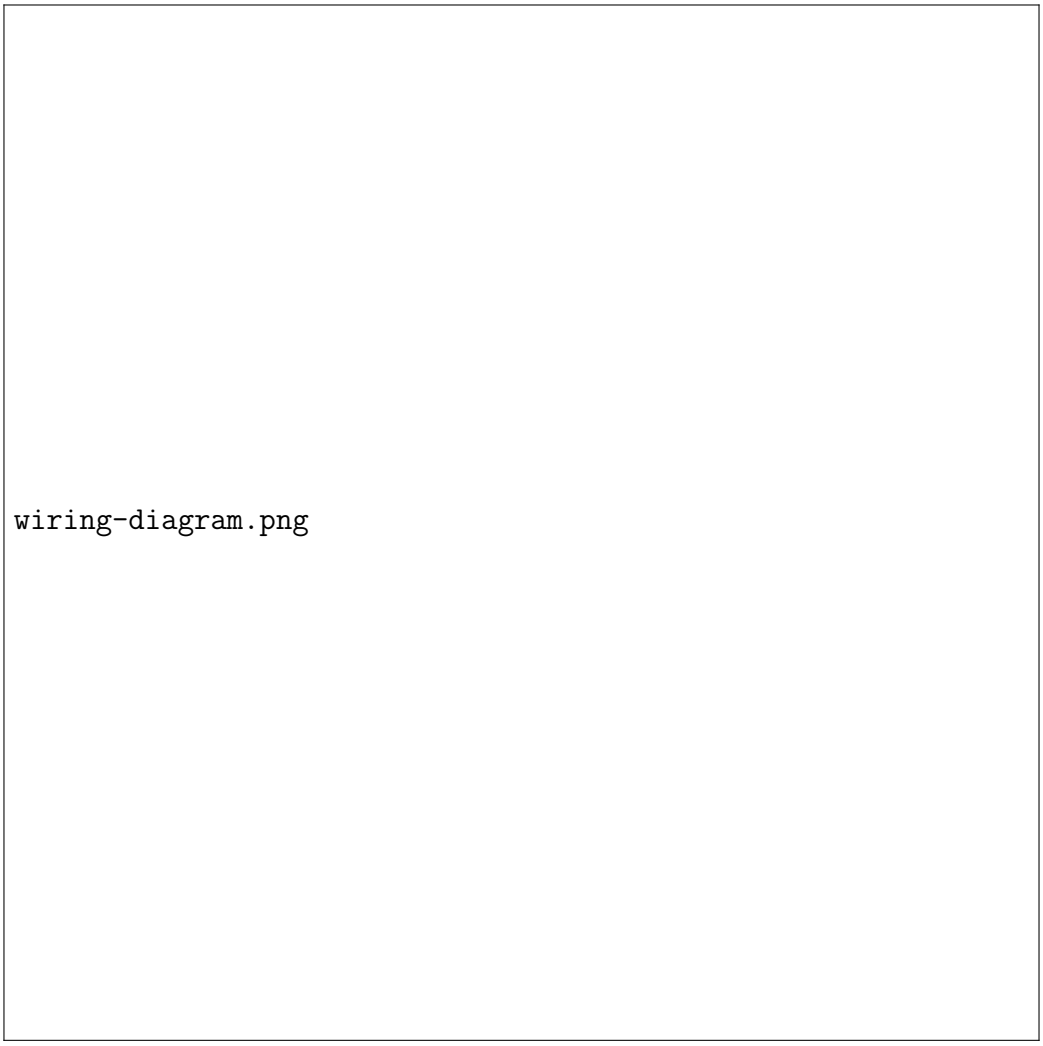

system-architecture.png

Chapter 4

Hardware Setup

4.1 Wiring Diagram

The following diagram shows how to connect the fingerprint scanner to Arduino and how ESP32 communicates with the web server.



4.2 Component Connections

- Connect the **Fingerprint Scanner** to the Arduino RX and TX pins.
- Power the ESP32 and connect it to the Arduino using Serial.
- Connect push buttons (for candidate selection) to Arduino digital pins.

Chapter 5

Software Setup

5.1 Arduino Code

Here is the full Arduino code for handling fingerprint authentication and candidate selection:

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>

// Define the software serial pins for fingerprint scanner
SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);

void setup() {
  Serial.begin(9600);
  mySerial.begin(57600);

  // Initialize the fingerprint scanner
  if (finger.verifyPassword()) {
    Serial.println("Fingerprint scanner ready.");
  } else {
    Serial.println("Fingerprint scanner not detected.");
  }
}

void loop() {
  // Capture fingerprint and process vote
  int id = getFingerprintID();
  if (id > 0) {
    Serial.println("Fingerprint recognized. Casting vote.");
```

```
    sendVoteToESP32(id); // Communicate with ESP32
  }
}

int getFingerprintID() {
  // Read the fingerprint data and return the voter ID
  if (finger.getImage()) {
    finger.image2Tz();
    return finger.fingerID;
  } else {
    return -1;
  }
}

void sendVoteToESP32(int voterID) {
  // Code to send vote to ESP32 via Serial
  Serial.print("Voter ID: ");
  Serial.println(voterID);
}
```

5.2 ESP32 Code for Communication

The ESP32 sends voting data to the server and manages real-time communication. Here's the ESP32 code:

```
#include <WiFi.h>
#include <WebSocketsClient.h>

WebSocketsClient webSocket;

const char* ssid = "your_SSID";
const char* password = "your_PASSWORD";

void setup() {
  Serial.begin(115200);

  // Connect to Wi-Fi
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  Serial.println("Connected to WiFi.");
```

```
  // Initialize WebSocket connection
  webSocket.begin("server_address", 8080);
  webSocket.onEvent(webSocketEvent);
}

void loop() {
  webSocket.loop();
}

void webSocketEvent(WStype_t type, uint8_t * payload, size_t length) {
  switch (type) {
    case WStype_TEXT:
      Serial.printf("Message from server: %s\n", payload);
      break;
    case WStype_DISCONNECTED:
      Serial.println("Disconnected from server.");
      break;
    case WStype_CONNECTED:
      Serial.println("Connected to server.");
      break;
  }
}

void sendVote(int candidateID) {
  // Send vote data to the server
  webSocket.sendTXT(String(candidateID));
}
```

Chapter 6

React Web Page Setup

6.1 React Code

Below is the code for a React web page that receives and displays real-time voting data.

```
import React, { useState, useEffect } from 'react';

const VotingResults = () => {
  const [votes, setVotes] = useState([]);

  useEffect(() => {
    const ws = new WebSocket('ws://server_address:8080');
    ws.onmessage = (event) => {
      const data = JSON.parse(event.data);
      setVotes(data.votes);
    };
  }, []);

  return (
    <div>
      <h1>Voting Results</h1>
      <ul>
        {votes.map((candidate, index) => (
          <li key={index}>
            {candidate.name}: {candidate.votes}
          </li>
        ))}
      </ul>
    </div>
```

```
  );
};

export default VotingResults;
```

Chapter 7

Voting Process

The voting process begins with voter authentication using the fingerprint scanner. Once authenticated, the voter selects a candidate, and their vote is registered in the system. The ESP32 sends the voting data to the web server, which updates the React web page in real-time.

Chapter 8

Security Features

8.1 Fingerprint Authentication

The system uses fingerprint-based authentication to ensure that only registered voters can cast their votes. The fingerprint data is checked against a stored database of voter records.

8.2 Data Encryption

All communication between the ESP32 and the web server is encrypted to protect against unauthorized access and ensure the integrity of the voting data.

Chapter 9

Troubleshooting

9.1 Fingerprint Scanner Not Recognized

- Verify wiring connections to the Arduino. - Check the serial communication settings and baud rate.

9.2 ESP32 Not Connecting to WiFi

- Ensure correct SSID and password. - Check for Wi-Fi signal strength.

9.3 Real-Time Web Page Not Updating

- Verify WebSocket connection between ESP32 and the server. - Check server logs for errors.

Chapter 10

Conclusion

The election voting machine with fingerprint authentication provides a secure, reliable way to conduct elections. The real-time display of votes via a React web page ensures transparency and allows election officials to monitor results as they come in.