PROJECT BASED LEARNING REPORT

on

EVM(Electronics Voting Machine)

Submitted by

Nikunj Gajipara (1032212451)

Under the Guidance of

Prof. Harshali Zodpe

Subject : MiniESR

School of Electronics & Communication Engineering

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY,
PUNE. [2023-2024]

Title: EVM(Electronics Voting Machine)

Brief Overview: To create an EVM using ESP32 and integrate ESP32 for wireless communication and UI, ESP32 for core functions like vote counting, utilize push buttons for vote casting, LED displays for result visualization, ensuring secure power supply, and implementing enclosures for physical protection, adhering to legal standards and ethical considerations for fair and transparent elections.

Problem Statement:

I. Design and develop an electronic voting system using ESP32 processor.

II. Ensure the system is secure, reliable, and user-friendly.

III. Implement features such as candidate selection, vote casting, result display, and data storage.

IV. Incorporate security measures to prevent tampering and ensure the integrity of the voting process.

V. Provide a user interface that is intuitive and accessible to all voters.

Overview of Features

1. **Wireless Connectivity:** Utilizes ESP32's Wi-Fi capabilities for seamless communication, enabling remote monitoring and data transmission.

2. **Secure Voting Process:** Implements encryption protocols to ensure the confidentiality and integrity of voting data, enhancing the security of the electoral process.

3. **User-Friendly Interface:** Employs intuitive interfaces on ESP32 for voters, displaying candidate options and voting results, enhancing accessibility and usability.

4. **Real-time Vote Tallying:** ESP32 manages real-time vote counting and tallying, providing instant feedback on voting outcomes and promoting transparency.

5. **Robust Hardware Integration:** Integrates push buttons for vote casting, LED displays for information dissemination, and reliable power management for uninterrupted operation.

6. **Scalable and Adaptable:** Designed for scalability to accommodate varying voting scenarios and adaptable for deployment in diverse electoral environments.

Interface Overview

1. Wireless Communication: ESP32 facilitates wireless communication, enabling connectivity for remote monitoring and result transmission.

2. User Interface: ESP32 manages the user interface, displaying candidate options and voting results on an LCD screen or LED display.

3. Input Handling: ESP32 interfaces with input devices such as push buttons, registering and processing votes securely.

4. Data Exchange: ESP32 and Arduino IDE communicate seamlessly, exchanging voting data and instructions for real-time processing.

5. Central Control: ESP32 serves as the central controller, orchestrating the voting process, tallying votes, and ensuring data integrity.

6. Integration: The ESP32 integrate harmoniously, combining wireless capabilities with robust hardware processing for an efficient and reliable EVM solution.

❖ Components Needed:

1. ESP32 Development Board: Handles wireless communication with Google Cloud and acts as a user interface.
2. 16x2 LCD Display: Displays candidate options, voting instructions, and results.
3. Push Buttons: Input devices for voters to cast their votes.
4. Fingerprint Sensor: Biometric authentication for voter identification.
5. Jumper Wires, Breadboard: For wiring and connections.
6. Google Cloud Account: For cloud connectivity and data storage.
7. Google Sheets: Used for storing and updating voting data.

❖ Hardware Setup:

1. Connect the Arduino IDE, ESP32, LCD, Buttons, and Fingerprint Sensor using jumper wires and a breadboard.
2. Wire the LCD to the ESP32 using its digital pins.
3. Connect the push buttons to the Arduino UNO's digital pins.
4. Interface the Fingerprint Sensor with the Arduino UNO according to its datasheet.

❖ Software Setup:

1. Install Arduino IDE for programming ESP32.
2. Set up the ESP32 development environment in Arduino IDE.
3. Install necessary libraries for interfacing with the LCD, Buttons, and Fingerprint Sensor.

❖ Arduino UNO Programming:

1. Write code to initialize the LCD and display voting instructions and candidate options.
2. Implement functions to handle button presses for casting votes.
3. Integrate the Fingerprint Sensor for voter authentication.
4. Develop logic to tally votes and update results.

❖ ESP32 Programming:

1. Configure ESP32 to connect to Wi-Fi and Google Cloud IoT Core.
2. Develop user interface functionalities for displaying voting status and results on the LCD.
3. Implement MQTT protocol for communication with Google Cloud.
4. Integrate fingerprint sensor data with voting process and send it to Google Cloud.

❖ Google Cloud Integration:

1. Set up a Google Cloud IoT Core project and register your ESP32 device.
2. Create a Google Sheets document to store voting data.
3. Develop a Google Cloud Function to receive data from ESP32 and update Google Sheets.
4. Configure MQTT communication between ESP32 and Google Cloud IoT Core.

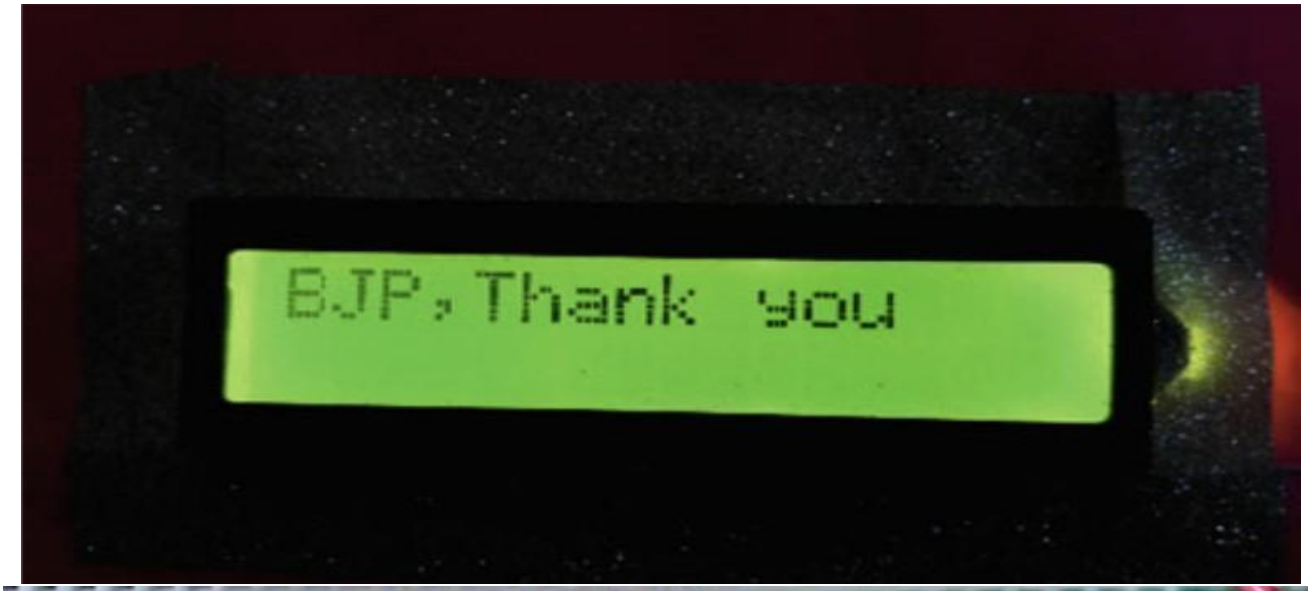
❖ Testing and Deployment:

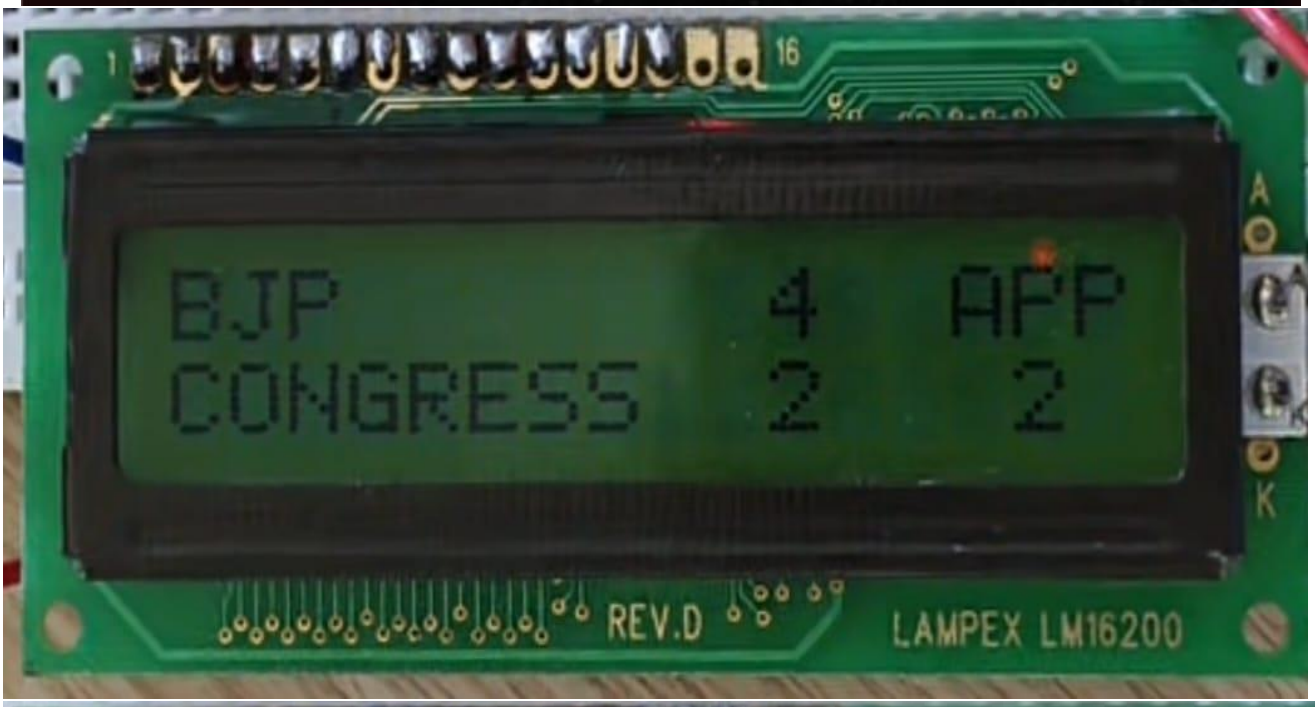
1. Test the EVM system thoroughly, ensuring all components work as expected.
2. Deploy the system in a controlled environment, such as a mock election scenario.
3. Monitor the system during operation and troubleshoot any issues that arise.
4. Gather feedback from users and stakeholders for further improvements.

❖ Security Considerations:

1. Implement encryption protocols for securing data transmission between ESP32 and Google Cloud.
2. Ensure proper access control measures to prevent unauthorized access to the EVM system.
3. Protect voter privacy and confidentiality throughout the voting process.
4. Regularly update and monitor the system for security vulnerabilities.

❖ Result


BJP, Thank you


BJP 4 AFP
CONGRESS 2 2
REV.D
LAMPEX LM16200


BJP is the
Winner
REV.D
LAMPEX LM16200

❖ Conclusion

In conclusion, leveraging the ESP32 for wireless communication and user interface, along with the ESP32 for core functionality management, provides a robust foundation for an **Electronic Voting Machine (EVM).** With reliable input devices, clear output displays, and secure enclosures, this **EVM** ensures efficient, transparent, and secure voting processes. Through careful component selection and meticulous design, the **EVM** facilitates democratic participation while upholding integrity and trust in the electoral system.

References

[1].https://www.researchgate.net/deref/https%3A%2F%2Fdatasheet.octopart.com%2FA000066-Arduino-datasheet38879526.pdf?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19

[2].https://www.researchgate.net/deref/https%3A%2F%2Fwww.sunrom.com%2Fp%2Ffinger-print-sensor-r305?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19

[3].https://www.researchgate.net/deref/https%3A%2F%2Fcdn.sparkfun.com%2Fdatasheets%2FIoT%2Fesp32_datasheet_en.pdf?_tp=eyJjb250ZXh0Ijp7ImZpcnN0UGFnZSI6InB1YmxpY2F0aW9uIiwicGFnZSI6InB1YmxpY2F0aW9uIn19