

Discussion Items

1. PDS
2. Assign roles/timeline for breadboard design and schematic design
3. Assign roles/timeline for coding design
4. Assign time frame for week 6 projects
5. Git page
6. Set recurring meeting times

Discussion Notes

1. PDS
 - a. Used L0 and L1 examples to fix ours
 - b. Added several design specs
 - i. Multiple User Profiles
 - ii. Logging and Audit Trail
 1. Generates log whenever lock is accessed or user is denied access
 - iii. Tamper Detection
 1. David has integration
 - iv. User Interface Enhancements
 - v. Backup Power System
 - vi. RFID Integration
2. Assign roles/timeline for breadboard design and schematic design
 - a. Davyd and felix will work on beadboard prototype schematic
 - b. Due date is Tuesday, 11/5
3. Assign roles/timeline for coding design
 - a. Alex and Anthony will code two different versions of the code and see which one works best
 - b. Also will collab together
 - c. Due date is Tuesday, 11/5
4. Assign time frame for week 6 projects
 - a. Breadboard/coding due 11/5
 - b. Review and fix things wednesday night during meeting
5. Git page
 - a. Drag and drop this document into Git
 - b. Put schematics and eventually code versions
6. Set recurring meeting times
 - a. 6pm Wednesday recurring meeting

7. Other

ADDED FEATURES 11/1/24

1. Multiple User Profiles

- **Implementation:** Use a fingerprint sensor (like the R307) that supports multiple fingerprints. Store user data in the ESP32's flash memory or SPIFFS.
- **Menu Interface:** Utilize an OLED display (e.g., SSD1306) to create a simple menu for adding/deleting users. Use buttons or a rotary encoder for navigation.

2. Logging and Audit Trail

- **Implementation:** Create a logging system that records timestamps for each access attempt. Use the ESP32's built-in flash memory or SPIFFS for storage.
- **Data Format:** Store logs in a simple text format (e.g., CSV) for easy readability.

3. Tamper Detection

- **Implementation:** Integrate a vibration sensor (like SW-420). If the sensor detects tampering, log the event and display a warning on the OLED.
- **Action:** Trigger an alert or store the event in the logs for later review.

4. User Interface Enhancements

- **Implementation:** Use an OLED display to create a user-friendly interface. Show messages for adding fingerprints, logging attempts, and navigating the menu.
- **Navigation:** Implement intuitive menu navigation using buttons or a rotary encoder.

5. Customizable Alerts

- **Implementation:** Use the ESP32's Wi-Fi capabilities to send notifications. You can leverage IFTTT or Pushbullet APIs for alerting users via email or mobile notifications.
- **Event Triggers:** Set alerts for unauthorized access attempts and tampering events.

6. Backup Power System

- **Implementation:** Integrate a rechargeable lithium battery (e.g., 18650) with a charging module (like TP4056). Ensure that the system can seamlessly switch between AC power and battery.
- **Power Management:** Implement low-power modes to extend battery life.