## 1. Executive Summary

The Mega Security System firmware transforms an Arduino Mega 2560 with Ethernet Shield into a fully functional, smart-home integrated alarm system. It supports traditional wired sensors, an alarm panel relay output, entry/exit delays with audible buzzer feedback, EEPROM-persisted zone configuration, and a web-based UI for control. It integrates seamlessly with Home Assistant via MQTT for state monitoring and remote control.

### 2. System Architecture

The system is modular for ease of maintenance:

- Zones.cpp/h: Reads zone sensors, manages EEPROM storage for zone descriptions/bypass, triggers alarm logic.
- MQTTHandler.cpp/h: Manages MQTT connection, subscriptions, publishes states and events.
- Countdown.cpp/h: Handles entry/exit delay countdowns with buzzer feedback.
- WebEndpoints.cpp/h: Implements password-protected HTTP endpoints for configuration and control.
- WebServer.h: Stores minified HTML/JS/CSS for the /zones configuration page.
- AlarmStates.h: Defines constants for alarm states and global alarm state variable.
- credentials.h: Stores network, MQTT, and login settings.

The firmware loop continuously polls zones, processes MQTT messages, updates countdown timers, and serves HTTP requests.

### 3. Hardware Requirements & Wiring

Required Components:

- Arduino Mega 2560
- Ethernet Shield (W5100 or W5500)
- Magnetic contact sensors or motion detectors
- Piezo buzzer (Pin 6)
- Relay module (Pin 5)

Pin Mapping for 16 zones (expandable):

Zone 1: 22 | Zone 9: 30

Zone 2: 23 | Zone 10: 31

Zone 3: 24 | Zone 11: 32

Zone 4: 25 | Zone 12: 33

Zone 5: 26 | Zone 13: 34

Zone 6: 27 | Zone 14: 35

Zone 7: 28 | Zone 15: 36

Zone 8: 29 | Zone 16: 37

Buzzer: Pin 6

Alarm Relay Output: Pin 5 (active LOW)

Ensure pull-up resistors or INPUT\_PULLUP mode is used for normally closed sensors. Connect Ethernet Shield to router/switch with CAT5/6 cable.

### 4. Software Requirements

- Arduino IDE 1.8.x or newer
- Ethernet3 or UIPEthernet library (depending on shield)
- PubSubClient library

- ArduinoJson library

Compile mega-security-eth\_FINAL\_PRO.ino with all module files in the same project folder.

Update credentials.h before compiling to set IP address, MQTT broker info, and login credentials.

### 5. Setup Instructions

- 1. Wire all sensors, buzzer, and relay as per Section 3.
- 2. Edit credentials.h with correct IP, network, MQTT broker, and credentials.
- 3. Compile and upload the firmware to Arduino Mega.
- 4. Access web UI via http://<device-ip>/zones and log in.
- 5. Set zone descriptions, bypass where needed, click Save All.
- 6. Integrate with Home Assistant using home\_assistant\_config.yaml.

### 6. Web Interface Guide

Login: Use WEB\_ADMIN\_ID and WEB\_ADMIN\_PASSWORD from credentials.h

#### Zone Table:

- Description: Editable text name for each zone.
- Bypass: Checkbox to exclude a zone from arming logic.

#### Color Codes:

- Red: Zone triggered
- Yellow: Zone bypassed
- White: Zone clear

#### **Buttons:**

- Save All: Write all changes to EEPROM
- Toggle Test Mode: Enable/disable alarm triggering (logs only)
- Reboot: Restart the controller
- Toggle Theme: Switch between light and dark mode

The table auto-refreshes every 2 seconds with live status updates.

### 7. Home Assistant Integration

Import home\_assistant\_config.yaml into HA configuration.yaml.

#### Entities:

- alarm\_control\_panel.mega\_security: Full arming/disarming control
- sensor.alarm\_countdown: Shows seconds remaining
- binary\_sensor.alarm\_test\_mode: Test Mode status
- binary\_sensor.zone\_<n>: Per-zone sensors
- sensor.last\_triggered\_zone: Last zone to trigger alarm

Add Lovelace alarm-panel card for user-friendly control.

Automations can use MQTT topics for advanced behavior.

### 8. Alarm Operation Logic

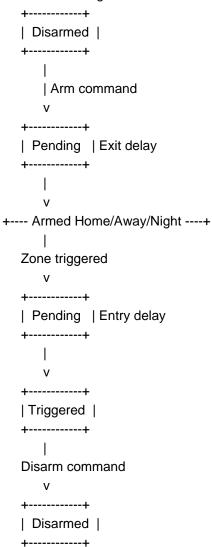
### States:

- Disarmed
- Pending (exit delay)
- Armed (Home/Away/Night)
- Pending (entry delay)
- Triggered

```
Flow:
```

Disarmed -> (arm command) -> Pending (exit delay) -> Armed mode -> (zone trip) -> Pending (entry delay) -> Triggered -> (disarm) -> Disarmed

State Machine Diagram:



### 9. MQTT API Specification

```
Topic | Direction | Payload Example | Description home/alarm | Pub | armed_home | Current alarm state home/alarm/set | Sub | arm_away | Command to change alarm state home/alarm/countdown | Pub | 25 | Seconds remaining in countdown home/alarm/test_mode | Pub | on | Test mode status home/alarm/zones | Pub | {"total_zones":16,"zones":[...]} | JSON array of all zones home/alarm/zone/<n> | Pub | triggered | State of individual zone home/alarm/last_trigger | Pub | {"zone":3,"desc":"Garage","time":"14:35:22"} | Last zone triggered
```

### 10. EEPROM Layout

Each zone:

- 20 bytes: ASCII description (null-padded)

- 1 byte: Bypass flag (0/1) Offset = zone index \* 21

## 11. Maintenance & Troubleshooting

- No MQTT updates: Check broker IP/port and network cables

- No Web UI access: Confirm IP address, LAN connectivity
- Zones always triggered: Check sensor wiring and pull-up configuration
- Settings not saved: Check EEPROM write logic and power cycle

### 12. Security Best Practices

- Change default WEB\_ADMIN\_ID and WEB\_ADMIN\_PASSWORD
- Use unique MQTT username/password
- Restrict device to LAN or VLAN segment
- Consider TLS-secured MQTT broker

### 13. Commissioning Checklist

- All zones wired and detected in UI
- Zone descriptions set and bypass flags configured
- MQTT connection confirmed
- Alarm arms/disarms from both UI and Home Assistant
- Exit delay buzzer works
- Entry delay triggers alarm
- Test Mode disables triggering
- Relay output activates alarm panel
- Settings persist after power cycle
- Reboot button functions correctly