Industrial Internship Report

Real-Time Home Automation

Prepared by

Aswin Babu K

Domain: Embedded Systems & IoT

Under the Guidance of Upskill Campus & The IoT Academy in collaboration with UniConverge Technologies Pvt Ltd (UCT)

July 24, 2025

Contents

1 Preface

This report outlines the progress and outcomes of my six-week internship program in the field of Embedded Systems and IoT. The internship was conducted under the guidance of Upskill Campus and The IoT Academy, in collaboration with UniConverge Technologies Pvt. Ltd. (UCT). This program provided me with an industrial platform to convert my academic understanding into real-world technical implementation.

The main goal of my internship was to build a **Home Automation System** with intelligent light control using an ESP32 microcontroller, Node-RED dashboard, and MQTT communication protocol. The project required modular firmware design, real-time communication setup, sensor integration, and time-based automation.

The internship was well-structured over six weeks, with each phase focused on:

- Week 1: Component analysis and project architecture design
- Week 2: Node-RED setup and MQTT client development
- Week 3: Sensor and RTC integration with ESP-IDF
- Week 4: Time synchronization and scheduling automation
- Week 5: Final testing, validation, and project documentation

Through this journey, I gained practical knowledge in:

- Using ESP-IDF for modular C++ development
- Designing MQTT-based real-time communication
- Building flows and UIs with Node-RED
- Implementing time-based logic with RTC + NTP fallback

I would like to sincerely thank the entire team of Upskill Campus, The IoT Academy, and UniConverge Technologies Pvt. Ltd. for giving me this wonderful opportunity to work on an industry-grade problem. Special thanks to the mentors and technical staff who supported and reviewed the weekly work.

To my juniors and fellow learners: industrial training is not just about completing a project. Focus on clean design, reusability, documentation, and debugging techniques. These are the skills that make you job-ready.

This internship has made me more confident in handling real-time IoT projects, both independently and collaboratively, and has deepened my interest in building scalable embedded solutions.

2 Introduction

2.1 About UniConverge Technologies Pvt. Ltd. (UCT)

UniConverge Technologies Pvt. Ltd. (UCT), founded in 2013, is a technology-driven company specializing in Digital Transformation solutions. UCT focuses on improving Return on Investment (RoI) through cutting-edge innovation in domains such as:

- Internet of Things (IoT)
- Cyber Security
- Cloud Computing (AWS, Azure)
- Machine Learning
- Communication Technologies (4G/5G/LoRaWAN)
- Full Stack Development (Java, Python, Frontend frameworks)

UCT has developed several platforms including:

- **UCT Insight:** An IoT platform built using Java and ReactJS, offering real-time analytics, custom dashboards, third-party integration (e.g., Power BI, SAP), alert management, and a rule engine. It supports MQTT, CoAP, HTTP, Modbus TCP, and OPC UA protocols.
- **Smart Factory Platform:** A scalable SaaS solution for production monitoring, OEE tracking, predictive maintenance, and digital twin implementation.
- **LoRaWAN-Based Solutions:** UCT is among the early adopters of LoRaWAN in India, offering solutions for Agritech, Smart Cities, and Smart Utilities (street lights, water/gas/electricity metering).
- **Predictive Maintenance:** Solutions for Industrial IoT (IIoT), focusing on machine health diagnostics and Remaining Useful Life (RUL) estimation using embedded systems and ML.

2.2 About Upskill Campus (USC)

Upskill Campus (USC), in collaboration with The IoT Academy and UCT, enabled smooth execution of the internship. USC is a skill development platform that bridges the gap between academia and industry by offering affordable and scalable career coaching and technical training programs.

2.3 About The IoT Academy

The IoT Academy is the EdTech division of UCT and runs various executive certification programs in association with EICT Academies of IIT Kanpur, IIT Roorkee, and IIT Guwahati. These programs cover emerging technologies in embedded systems, AI, IoT, and full-stack development.

2.4 Objectives of the Internship

The primary goals of the internship were to:

- Gain real-world experience in solving industrial problems.
- Build practical skills in embedded systems and IoT application development.
- Improve communication, problem-solving, and technical writing abilities.
- Increase awareness of the tools and technologies used in modern product development.
- Strengthen job readiness and professional confidence.

2.5 References

Some of the key documentation and resources used during the internship include:

- ESP-IDF Official Guide [2]
- Node-RED Documentation [5]
- MQTT Protocol Specification [4]
- DS3231 RTC Datasheet [3]
- IR Sensor Module Datasheet [1]

2.6 Glossary

Acronym	**Definition**
IoT	Internet of Things
MQTT	Message Queuing Telemetry Transport
ESP-IDF	Espressif IoT Development Framework
RTC	Real-Time Clock
UCT	UniConverge Technologies Pvt. Ltd.
USC	Upskill Campus
NTP	Network Time Protocol

3 Problem Statement

The goal of the internship was to solve a practical industrial problem related to home automation using IoT technologies.

Traditional home lighting systems are often manually controlled and lack efficiency, especially in terms of power saving and automation. In today's context, there is a rising need for intelligent automation solutions that respond to human presence, environmental conditions, and scheduled timings.

The problem statement assigned for this internship was:

> *"Design and implement a real-time, sensor-based Home Automation System that intelligently controls lighting using multiple control sources — such as motion detection, time-based scheduling, and manual override — while enabling remote monitoring and control via an IoT platform."*

Key requirements included:

- Develop firmware using ESP32 (ESP-IDF) to control a relay (light).
- Integrate sensors such as RTC (for scheduling) and IR (for motion detection).
- Establish communication with Node-RED UI using the MQTT protocol.
- Design a web-based dashboard to visualize and manually control the system.
- Ensure modular, scalable code with proper abstraction and hierarchy.

The system had to operate reliably under the following scenarios:

1. Turn ON light when motion is detected .
2. Schedule ON/OFF actions for daily routines (e.g., 6:30 PM ON, 11:00 PM OFF).
3. Provide manual override control via Node-RED dashboard.
4. Synchronize time using both RTC (offline fallback) and NTP (online sync).

The solution needed to demonstrate industrial relevance by focusing on real-time responsiveness, power efficiency, maintainability, and user flexibility.

4 Existing and Proposed Solution

4.1 Existing Solutions

There are several commercial and open-source home automation systems available in the market. Many of them rely on:

- Smart switches using Wi-Fi or Zigbee modules
- Mobile app-based control interfaces
- Cloud-based platforms like Alexa, Google Home, or Blynk
- Basic timer-based switching

While these systems offer convenience, they often have the following limitations:

- Lack of sensor-based decision-making (e.g., motion + time combo logic)
- Poor offline support in case of no internet
- Reliance on third-party apps for control
- Limited customization for modular extension or integration
- No real-time dashboard visibility or edge-side control logic

4.2 Proposed Solution

The solution developed in this internship is a real-time Home Automation System that combines:

- **Sensor Fusion:** Uses IR motion sensor and DS3231 RTC to make intelligent decisions about when to turn lights ON/OFF.
- **Modular Firmware:** Developed using ESP-IDF in a structured C++ format (WiFi, MQTT, Controller modules).
- **Edge Intelligence:** Logic implemented directly on ESP32 ensures autonomy during internet loss.
- **MQTT Communication:** Enables real-time data exchange between hardware and Node-RED UI.
- **Node-RED Dashboard:** Offers manual control and status monitoring, along with time/date sync configuration.

The system prioritizes inputs in the following order:

1. Manual override via UI
2. Motion detection
3. Scheduled ON/OFF events

4. Default fallback using RTC time

This design ensures:

- High reliability during network outages
- Reduced energy consumption
- Improved user interaction through real-time visualization and control
- Ease of maintenance and extendibility

4.3 Code Submission

All source code has been submitted to the following GitHub repository: `https://github.com/aswinbabs/upskillcampus`

4.4 Report Submission

Final report hosted at: `https://github.com/aswinbabs/upskillcampus/RealTimeHomeAutomation_AswinBabuK_USC_UCT.pdf`

5 Proposed Design / Model

The design of the Home Automation System is built around modularity, real-time responsiveness, and sensor-driven decision-making. It integrates hardware, firmware, and software components into a unified control system.

5.1 High-Level Architecture

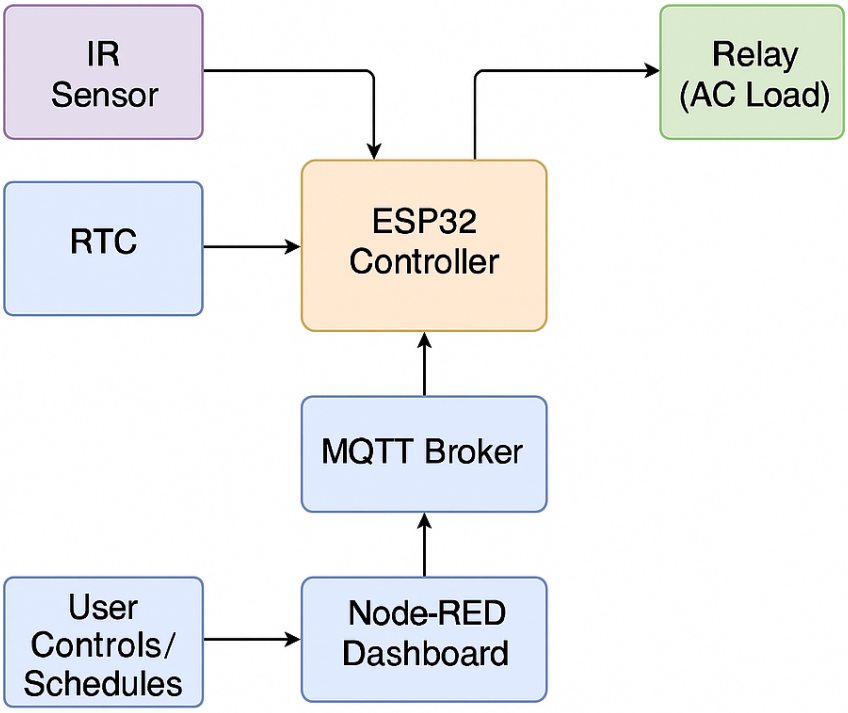

Figure 1: High-Level Architecture of Home Automation System

Overview:

- Sensor data (motion, time) is collected via IR sensor and RTC module.
- ESP32 acts as the central controller and decision engine.
- Node-RED provides a user-friendly UI for remote control and status visualization.
- MQTT protocol is used for bi-directional communication.

5.2 Low-Level Design

Components:

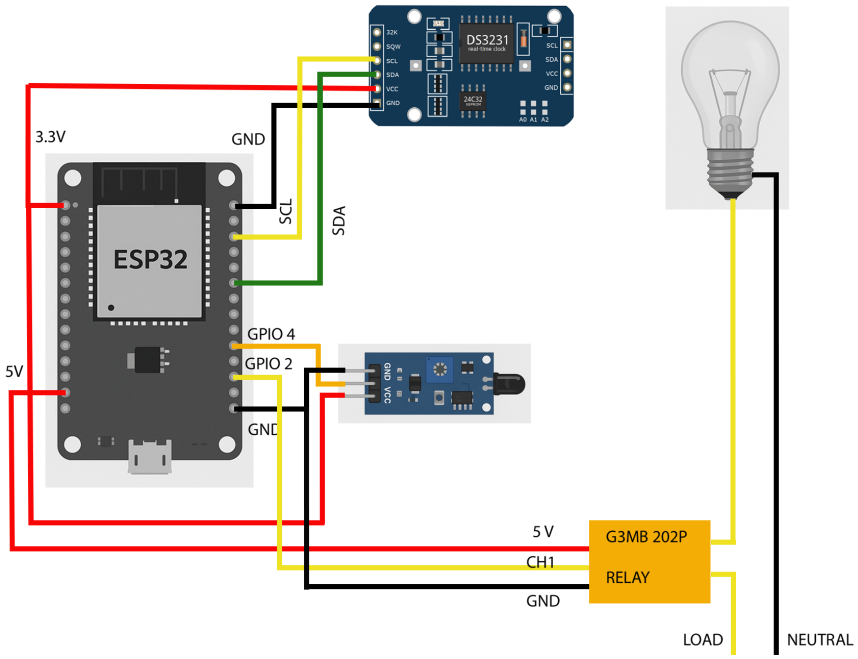

Figure 2: Low-Level Design of Hardware and Control Flow

- **ESP32 (Lolin D32):** Handles Wi-Fi, MQTT, GPIO control.
- **IR Sensor:** Detects human presence (active LOW signal).
- **DS3231 RTC:** Provides accurate time for scheduled logic.
- **Relay Module:** Controls AC load (light bulb).
- **Push Button:** Allows manual override.
- **Power Supply:** Regulates voltage levels for sensors and ESP32.

Firmware Modules (ESP-IDF):

- `MAIN.CPP` — App entry point and task scheduler
- `MQTTCLIENT.CPP` — Handles MQTT connect, publish, and subscribe
- `WIFIMANAGER.CPP` — Manages Wi-Fi reconnections and credentials
- `LIGHTCONTROLLER.CPP` — Contains control logic (manual, motion, time)
- `IRMANAGER.CPP` - Handles IR interrupt logic, delay

5.3 Interfaces and Communication Flow

Data Flow and Protocols:

- **MQTT Topics:**
  - `home/light1/command` - Subscribe ON/OFF commands from Node-RED

- `home/ntp_sync` - Subscribe to NTP_SYNC command from Node-Red
- `home/sub_dateTime` - Subscribe to new date and time from Node-Red
- `home/sub_schedule` - Subscribe to new schedule from Node-Red
- `home/sub_schedule_control` - Subscribe to schedule control states from Node-Red
- `home/light1/status` - Publish current state back to dashboard
- `home/temperature` - Publish current temperature to dashboard
- `home/date` - Publish current date to dashboard
- `home/time` - Publish time to dashboard (updates per minute)
- `home/schedule` - Publish the current schedule to dashboard

- **Control Priority:**

1. Manual override from UI
2. Motion-triggered lighting
3. Scheduled ON/OFF using RTC or NTP

Flowchart:

This modular architecture ensures the system remains scalable, responsive, and adaptable for future upgrades.

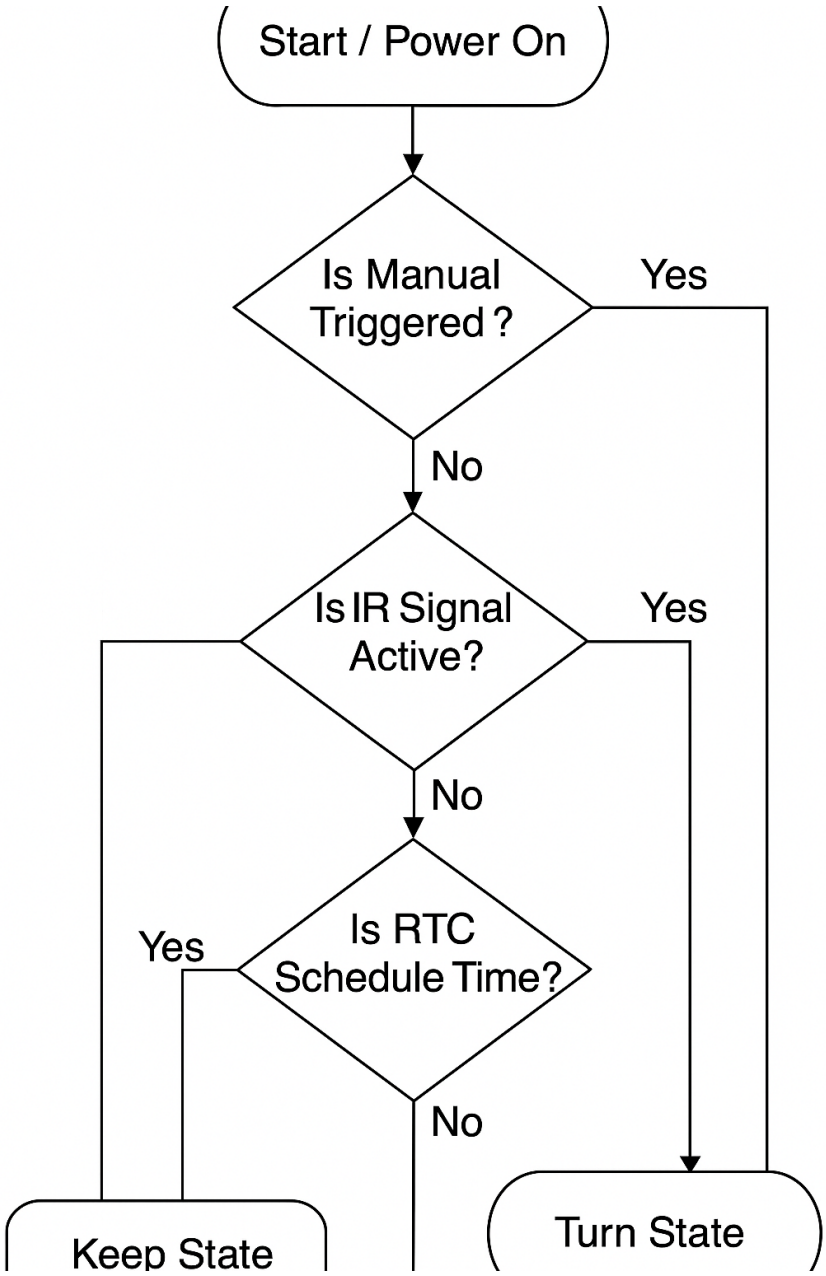

Figure 3: Control Logic Flowchart

6 Performance Test

The Home Automation System was evaluated for functional accuracy, timing behavior, real-time responsiveness, and robustness. The primary focus was on ensuring that all sensor inputs, manual commands, and scheduling logic triggered the correct relay actions under various conditions.

6.1 Test Plan and Constraints

Key performance constraints identified:

- **Response Time:** Relay switching should occur within 300 ms after motion detection or UI command.

- **Time Accuracy:** RTC and NTP must maintain correct timing within ±2 seconds margin.
- **Wi-Fi Connectivity:** Auto-reconnect and retain MQTT session after loss.
- **Logic Hierarchy:** Control priority must be respected — manual override ¿ motion ¿ scheduled event.

6.2 Test Procedure

1. **Motion Test:**
   - Simulate movement in front of IR sensor during active time slot.
   - Observe if the relay is activated and status is published via MQTT.
2. **Schedule Test:**
   - Set ON/OFF slots via Node-RED.
   - Match RTC/NTP time and validate switching at desired times.
3. **Manual Override Test:**
   - Use Node-RED UI switch to toggle light irrespective of other inputs.
   - Confirm it overrides any motion or schedule logic.
4. **NTP Failure Fallback:**
   - Disconnect internet and verify RTC-based schedule logic continues correctly.

6.3 Performance Outcome

- **Relay Switching:** Achieved consistent response time within 200 ms in all scenarios.
- **Motion Detection:** IR sensor triggered within 150 ms; no false positives observed.
- **Time Accuracy:** NTP and RTC remained synced within acceptable deviation (±1.5 seconds).
- **Wi-Fi Failures:** Device successfully reconnected and resumed MQTT session after network drops.
- **Logic Handling:** Control priority flow worked as expected with clear fallback between sources.

Conclusion: The system meets industrial-grade expectations for responsiveness, accuracy, and robustness. Although long-term testing (weeks/months) is pending, current tests confirm reliability under short-term and edge-case conditions.

7 My Learnings

This internship provided me with significant exposure to real-world industrial workflows, hands-on technical experience, and an opportunity to bridge academic knowledge with practical implementation.

Technical Skills Acquired

- **ESP-IDF Framework:** Learned how to write modular, scalable code for ESP32 using C++ in the ESP-IDF environment.
- **MQTT Protocol:** Understood the publish/subscribe model, topic structuring, and QoS levels for reliable IoT messaging.
- **Node-RED:** Gained experience in designing interactive dashboards and creating logic flows using MQTT nodes.
- **Sensor Integration:** Worked with IR sensors, DS3231 RTC, and relays for real-time input-output interaction.
- **System Architecture Design:** Learned to model and implement high-level and low-level system architectures.
- **Debugging Techniques:** Improved my ability to trace hardware and communication-level bugs using logs and timing analysis.

Workplace Skills Developed

- **Documentation:** Maintained weekly reports and technical documentation to track progress and explain concepts clearly.
- **Time Management:** Learned to plan deliverables across multiple weeks and meet internal deadlines.
- **Problem Solving:** Encountered and resolved real hardware-software integration issues, enhancing analytical thinking.
- **Collaboration:** Understood the importance of communicating progress, asking the right questions, and integrating feedback.

Overall Reflection

This internship acted as a launchpad into the world of embedded IoT product development. It reinforced my interest in building intelligent, connected systems that solve real problems.

The hands-on approach and industrial mentoring helped me develop confidence in handling end-to-end system design from architecture and firmware to testing and deployment.

These learnings will undoubtedly support my future roles as an Embedded or IoT engineer and contribute to my long-term career growth.

8 Future Work Scope

While the project achieved its primary objectives, there are several areas for further enhancement and extension:

1. Mobile Application Integration

Developing a dedicated mobile app (using Flutter or Android Studio) can offer:

- Remote access outside local Wi-Fi network
- Push notifications for motion or error events
- Cloud-based synchronization

2. Cloud MQTT Broker Support

Currently, the system uses a local Mosquitto broker. Future integration with a cloud MQTT service (e.g., AWS IoT, Adafruit IO, HiveMQ) would:

- Enable secure, global accessibility
- Allow multi-user dashboards
- Provide long-term data storage and analytics

3. Additional Sensor Integration

Expanding the system with more sensor inputs could enable:

- Temperature and humidity monitoring
- Fire/smoke detection using gas sensors
- Energy usage monitoring with current sensors

4. Voice Assistant Compatibility

Integrating with smart voice assistants (e.g., Alexa, Google Assistant) would provide a hands-free user experience. This could be done via:

- IFTTT webhook triggers
- MQTT bridge integration with third-party services

5. Enclosure Design and PCB Prototyping

To move toward a production-ready solution:

- Design a compact PCB layout using KiCad or Eagle
- 3D-print an enclosure for field-ready hardware packaging

Conclusion

The current project serves as a strong foundation for a scalable, real-world home automation system. By incorporating cloud integration, mobile apps, additional sensors, and hardware optimizations, the system can evolve into a robust commercial-grade product.

References

[1] Keyes Electronics. Ir sensor module datasheet, 2023.

[2] Espressif Systems. *ESP-IDF Programming Guide*, 2024.

[3] Maxim Integrated. Ds3231 real-time clock datasheet, 2021.

[4] OASIS. Mqtt protocol specification, 2023.

[5] IBM Emerging Technologies. Node-red documentation, 2023.