HOMI: IoT-Based Smart Home Monitoring and Control System

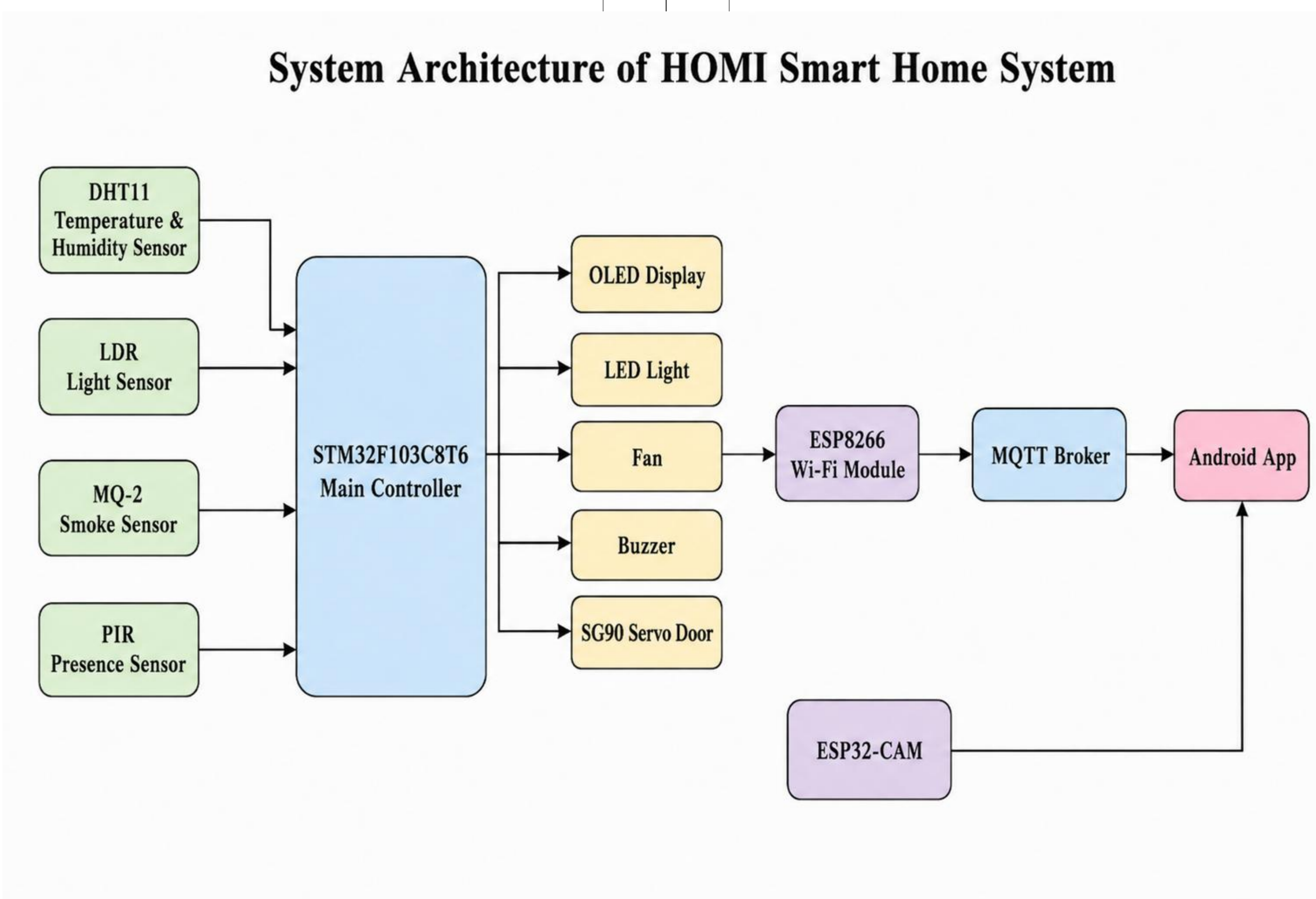
Final Report

Team Members: Kaicheng Wu kw36

Wenyu Zhou wenyu6

Yichen Zhang yichen36

This report is organized according to the final report evaluation criteria: introduction, design, cost and schedule, requirements and verification, conclusion, and ethical considerations.



1. Introduction

1.1 Purpose

The purpose of this project is to design and implement an IoT-based smart home monitoring and control system named HOMI. The system addresses the problem that traditional household devices are usually operated independently and lack centralized environmental monitoring, automatic control, remote access, and security awareness. In many indoor environments, users need to monitor temperature, humidity, light intensity, smoke concentration, human presence, and device status at the same time. Without an integrated system, these tasks require manual checking and manual device switching, which reduces convenience and may delay response to unsafe conditions such as smoke or abnormal environmental values.

The proposed solution is a smart home prototype based on an STM32F103C8T6 microcontroller, environmental sensors, local actuators, an OLED display, an ESP8266 Wi-Fi module, MQTT communication, an Android mobile application, and an ESP32-CAM module. The system collects environmental data, displays real-time values locally, uploads sensor data to the mobile application, receives remote commands from the application, and controls household devices such as a fan, LED light, humidifier, buzzer, and servo-based door mechanism.

1.2 High-Level Functionality

- Environmental monitoring: the system reads temperature, humidity, light intensity, smoke level, and human presence data.
- Local display: a 12864 OLED screen displays real-time sensor values and device states.
- Automatic control: actuator states are determined by threshold-based logic in automatic mode.
- Manual control: users can control devices through local inputs and the Android application.
- Wireless communication: the ESP8266 module connects the STM32 controller to the MQTT broker.
- Mobile interface: the Android application provides sensor display, device control, parameter setting, charts, and camera/security pages.
- Security monitoring: the ESP32-CAM and mobile application provide a demonstration of camera access and face-related functions.

1.3 Subsystem Overview

The overall system is divided into five major subsystems: the sensor subsystem, the STM32 control subsystem, the actuator subsystem, the communication subsystem, and the Android application subsystem. The STM32 controller is the core of the system. It receives sensor data, performs threshold comparison, controls actuators, updates the OLED display, and exchanges data with the ESP8266 module. The ESP8266 acts as the wireless bridge between the embedded device and the MQTT broker. The Android application acts as the user interface for remote monitoring and command transmission.

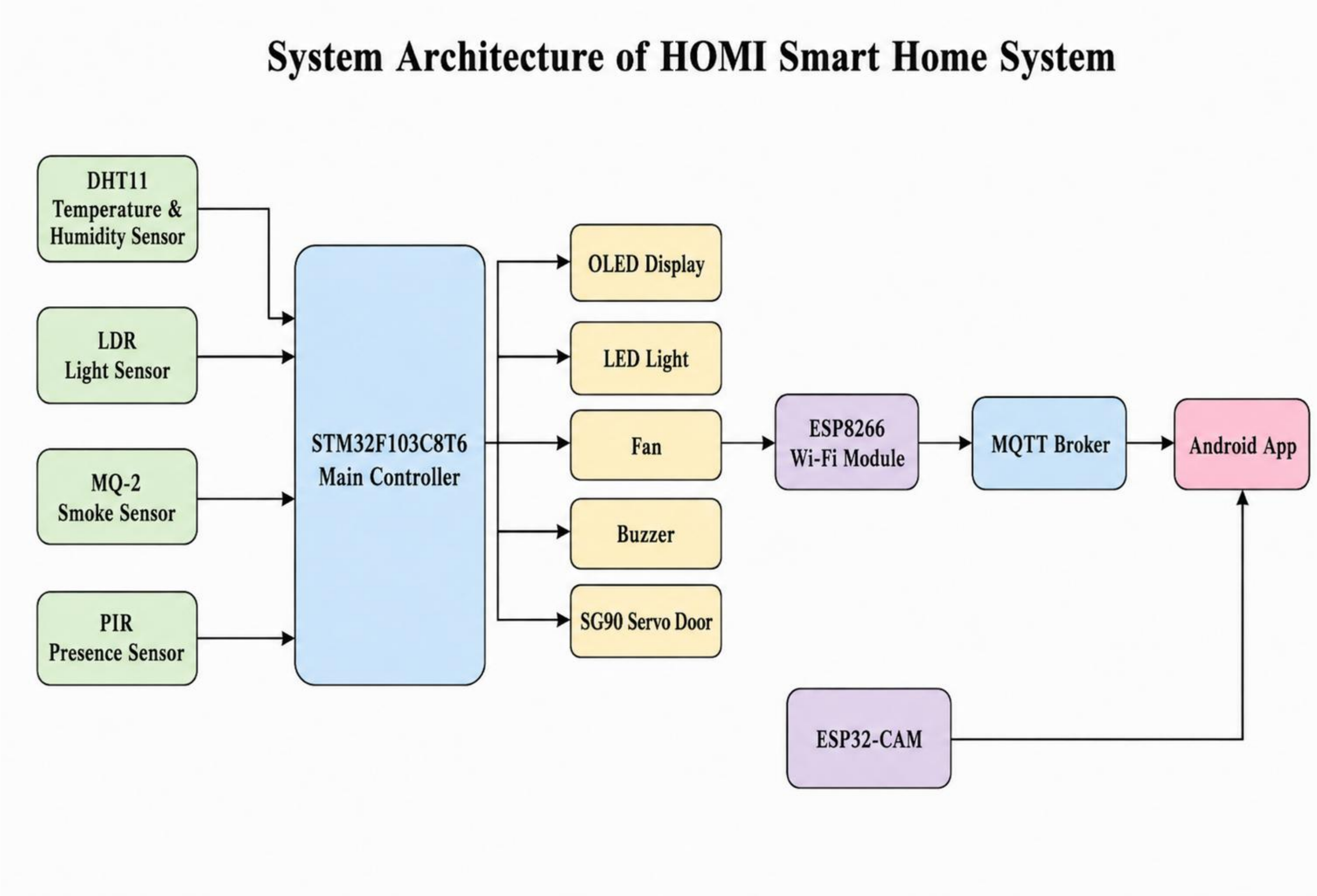


Figure 1 Overall architecture and data/control flow

2. Design

2.1 Design Equations and Calculations

The STM32 reads analog sensor signals through its ADC. The light sensor and smoke sensor produce voltage values that are converted into digital values. The input voltage can be estimated from the 12-bit ADC result using the following equation:

Vin = (ADCvalue / 4095) x Vref

where ADCvalue is the digital ADC reading, 4095 is the maximum 12-bit ADC value, and Vref is approximately 3.3 V. For display and threshold control, the normalized sensor percentage can be calculated as:

Percentage = (ADCvalue / 4095) x 100%

The SG90 servo is controlled by a PWM signal. The servo angle is determined by the high-level pulse width in a 20 ms control period. The theoretical relation is:

Thigh = 0.5 ms + (theta / 180 degrees) x 2.0 ms

This relationship allows the embedded controller to use a PWM output as the signal for a servo-based smart door or access mechanism.

2.2 Modular Design Description and Justification

The design is modular because each hardware and software part has a defined role and interface. This modularity simplifies debugging and allows individual subsystems to be tested independently before full system integration.

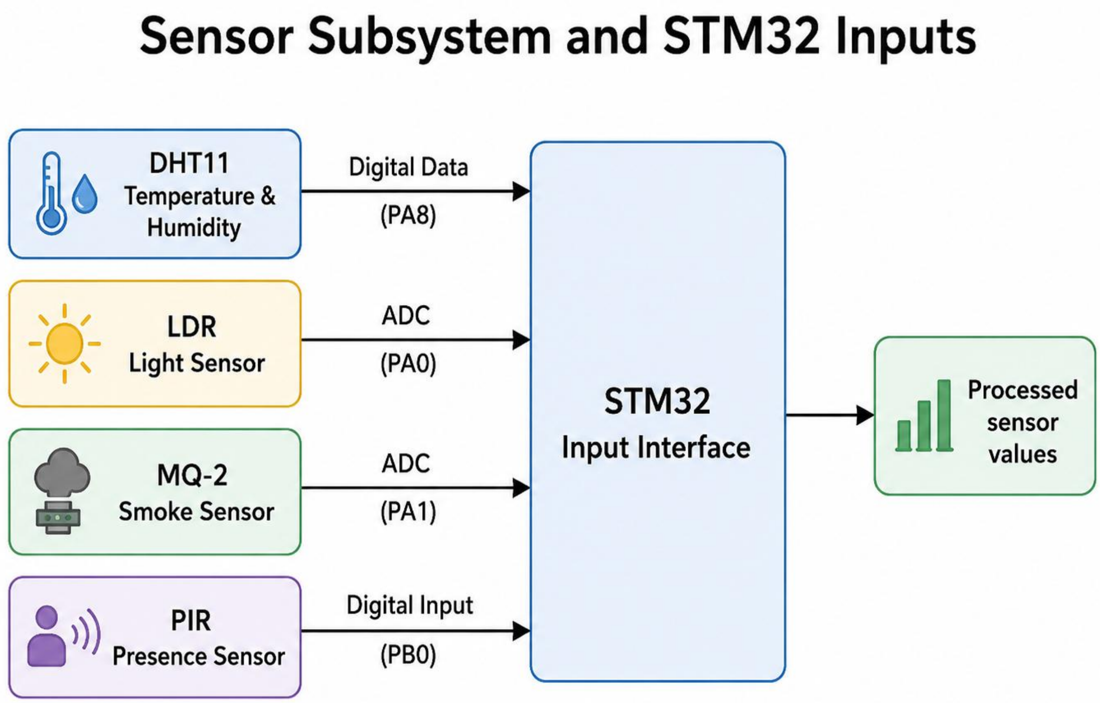


Figure 2. Sensor subsystem and input mapping.

The sensor subsystem includes the DHT11 temperature and humidity sensor, the LDR light sensor, the MQ-2 smoke sensor, and the PIR presence sensor. The DHT11 and PIR modules provide digital signals, while the LDR and MQ-2 modules provide analog signals that are suitable for ADC sampling. This combination is low-cost and sufficient for a smart home prototype.

Actuator Subsystem and STM32 Outputs

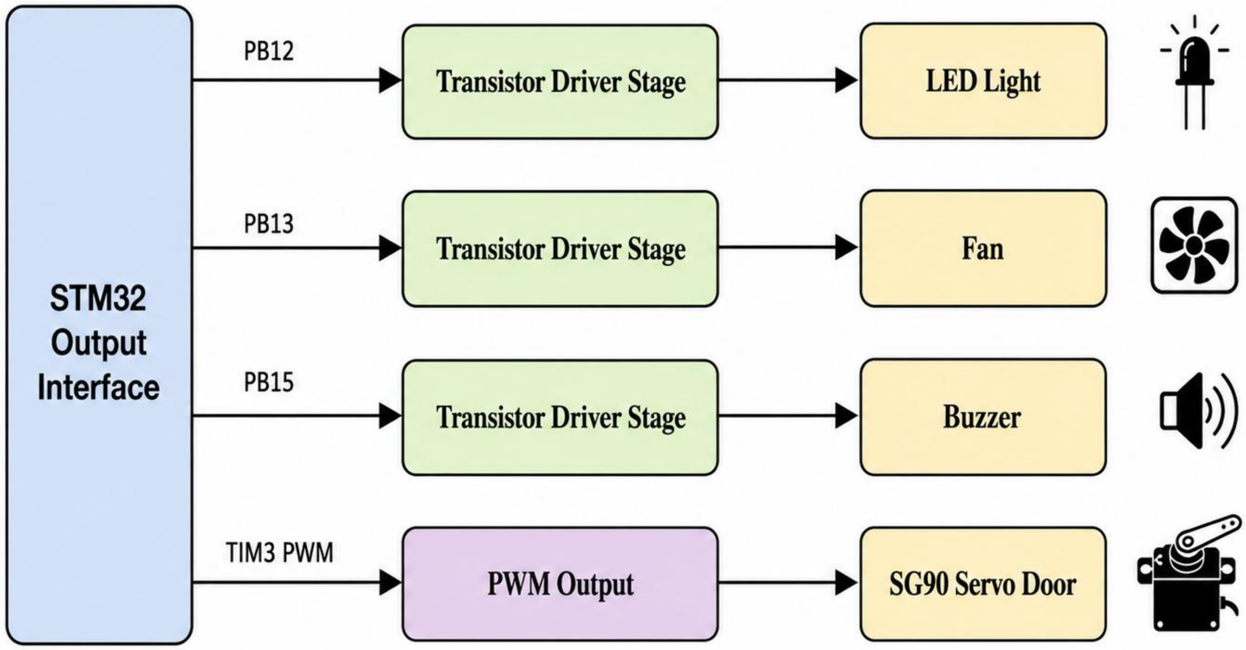


Figure 3. Actuator subsystem and output mapping.

The actuator subsystem includes the LED light, fan, humidifier, buzzer, and SG90 servo door. The LED, fan, humidifier, and buzzer are driven through transistor driver stages because STM32 GPIO pins cannot safely provide the current required by these loads. The servo is controlled by PWM from the microcontroller.

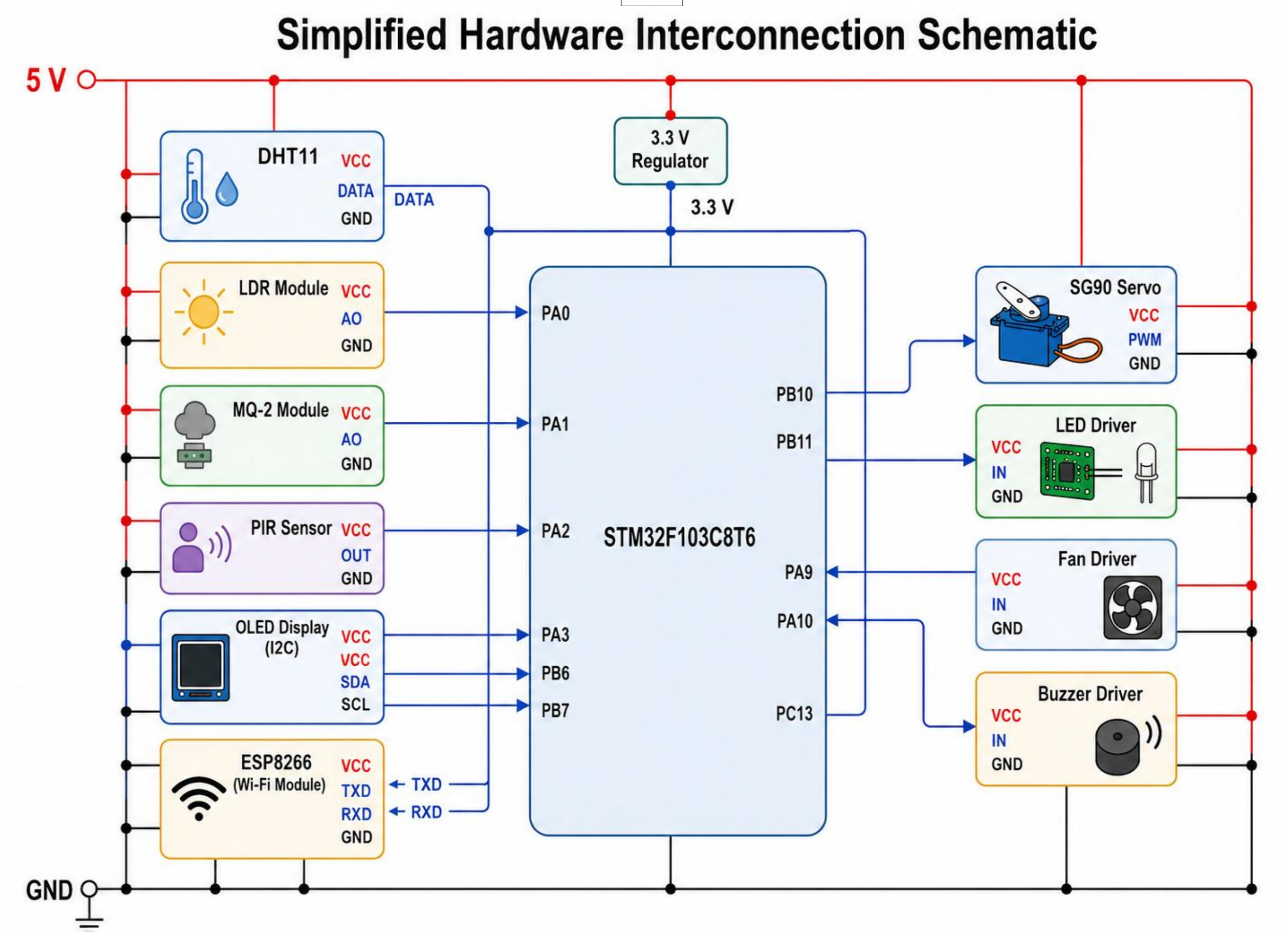


Figure 10. Simplified system wiring schematic.

The simplified wiring schematic shows the main signal and power connections. The system uses common ground, 5 V power for actuators and selected modules, and regulated 3.3 V where required by the controller and communication modules.

2.3 Communication and Software Design

MQTT was selected as the communication protocol because it is lightweight and well suited for IoT devices. The STM32 sends AT commands to the ESP8266 through UART. After Wi-Fi connection and MQTT login, the controller publishes sensor data and subscribes to the command topic. The Android application subscribes to data updates and publishes control commands.

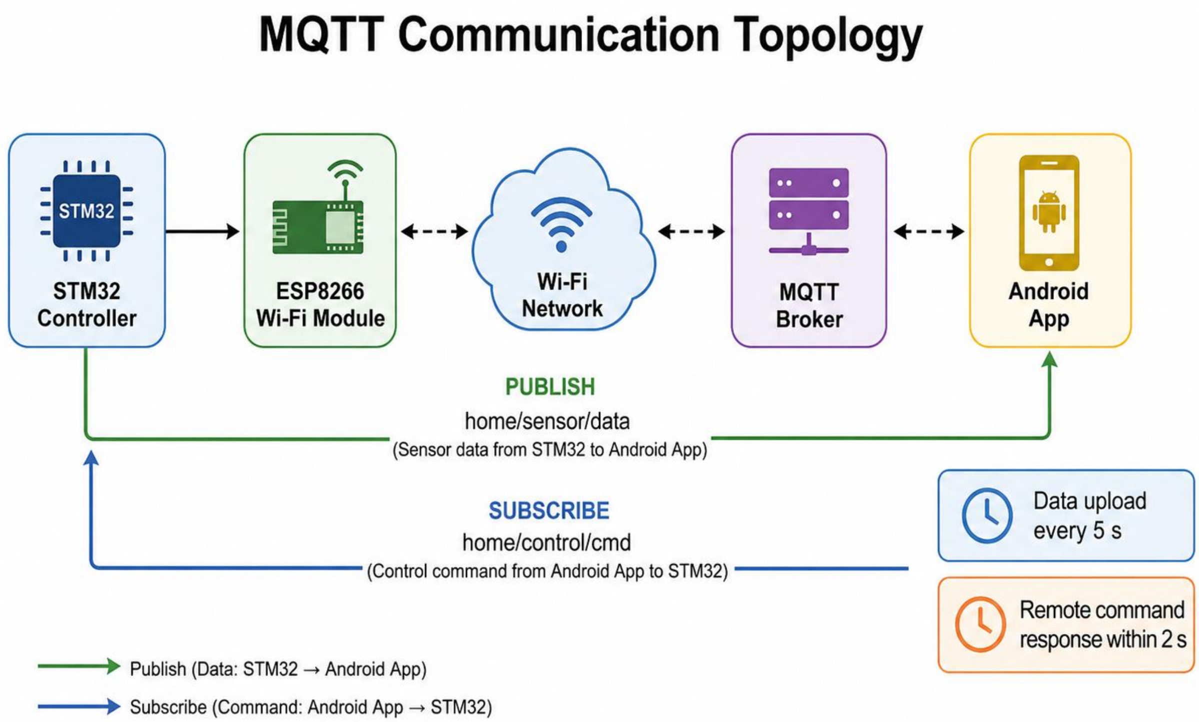


Figure 4. Wireless communication and MQTT message flow.

The embedded software is organized as a repeated control loop. The system initializes peripherals, connects to the wireless network, reads sensors, updates displays, checks the selected mode, drives actuators, publishes sensor data, receives commands, and repeats the process.

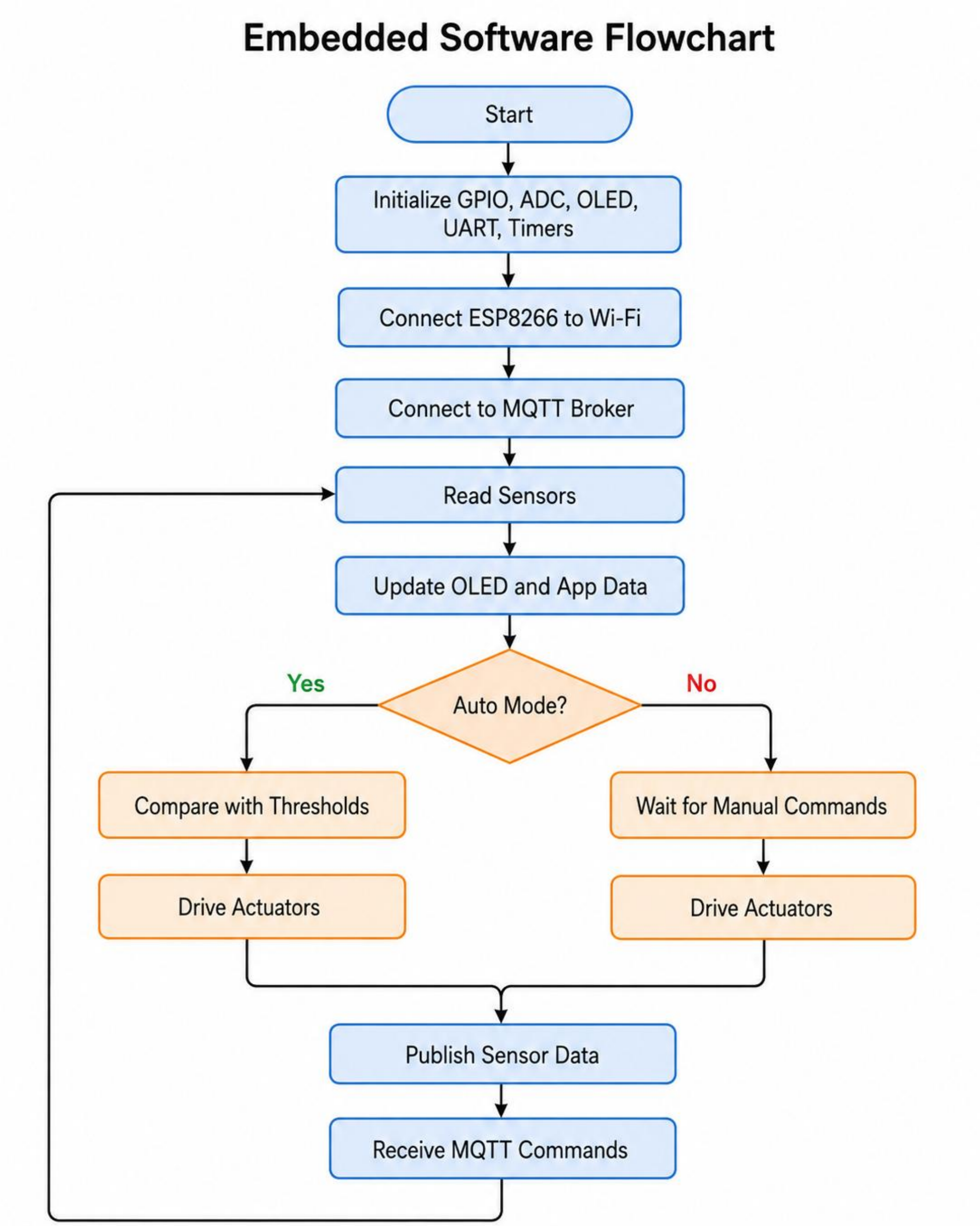
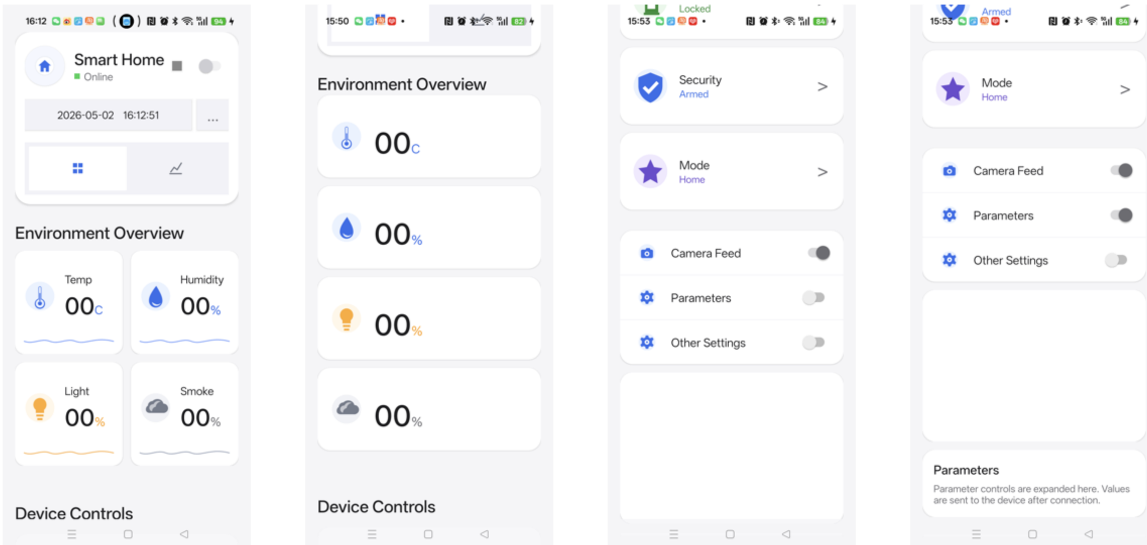


Figure 5. Main embedded control flow.

The Android application provides separate pages for the home dashboard, device control, parameter settings, and security or camera functions. This separation reduces interface complexity and makes the system easier to use.



Android app screens: overview, trend, camera, parameters

Figure 6. Android app screens used for monitoring, trend display, camera access, and parameter settings.

2.4 Design Alternatives and Corrective Actions

Several design alternatives were considered during the project. Bluetooth communication was simpler for short-range operation, but Wi-Fi and MQTT were selected because they support remote monitoring and better IoT scalability. A commercial cloud platform was considered, but a direct MQTT broker was chosen because it is lightweight, flexible, and easier to test. Higher-accuracy sensors such as DHT22 or SHT30 could replace the DHT11, but DHT11 was selected for its low cost and simple interface.

Corrective actions included adding multiple MQTT broker options to improve connection robustness, using transistor drivers instead of directly driving loads from GPIO pins, separating automatic and manual control by software flags, and saving key parameters to nonvolatile memory to reduce parameter loss after reset.

3. Cost and Schedule

3.1 Cost

The bill of materials was updated to reflect typical component prices in the China market. All prices are expressed in Chinese yuan (CNY). The total estimated hardware cost is CNY 231.

Component	Qty	Unit Price (CNY)	Subtotal (CNY)
STM32F103C8T6 board	1	18	18
ESP8266 module	1	15	15
ESP32-CAM	1	35	35
DHT11 sensor	1	6	6
MQ-2 sensor	1	12	12
LDR module	1	3	3
PIR sensor	1	4	4
OLED display	1	18	18
SG90 servo	1	14	14
Buzzer	1	2	2
LED module	1	2	2
Fan module	1	12	12
Discrete components	1	15	15
PCB and wiring	1	30	30
Power supply module	1	20	20

3.2 Schedule

The schedule was divided by week and by team role. Hardware work focused on component selection, circuit design, and driver circuits. Embedded work focused on microcontroller bring-up, sensor integration, automatic control logic, Wi-Fi communication, and MQTT implementation. Application work focused on Android interface development, data display, parameter setting, camera access, and security functions. The final weeks were used for integration, debugging, testing, and report preparation.

Week	Work Completed	Responsible Member
1	Define project topic, purpose, and major functions	All
2	Select sensors, STM32, ESP8266, OLED, and actuators	Hardware
3-4	Draw circuit design and PCB layout	Hardware
4-5	Develop GPIO, ADC, DHT11, and OLED firmware	Embedded
6-7	Develop actuator and automatic threshold control	Embedded/Hardware
8-9	Develop ESP8266 Wi-Fi and MQTT communication	Embedded
10-12	Develop Android app screens and data display	App
12	Add ESP32-CAM and face-related page	App
13-14	Integrate hardware, firmware, and app; debug response delays	All
15	Complete testing, figures, and final report	All

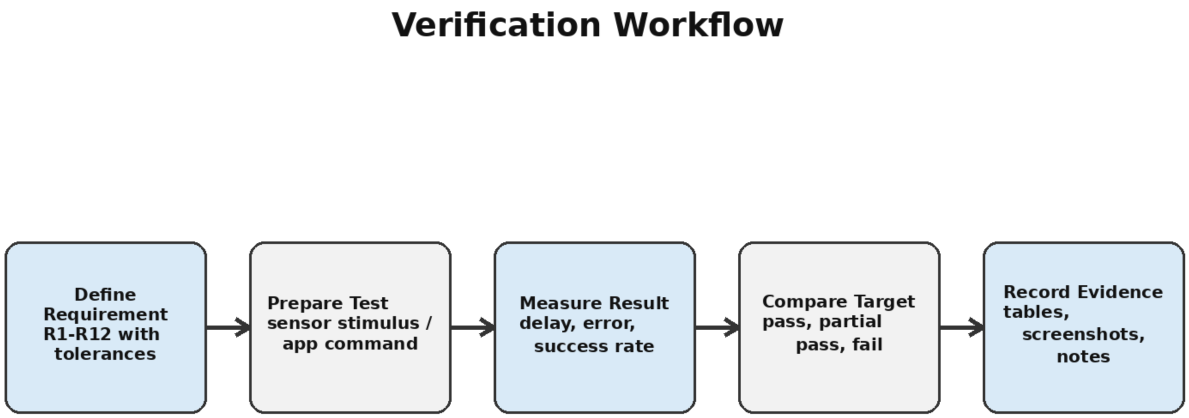
4. Requirements and Verification

4.1 Requirements

ID	Requirement	Tolerance or Target
R1	The system shall measure indoor temperature and humidity.	Temperature within +/-2 C; humidity within +/-5% RH
R2	The system shall measure light intensity and smoke level as normalized values.	0-100% normalized display
R3	The OLED shall display real-time sensor data and device status.	Refresh within 1 s
R4	The system shall control LED, fan, humidifier, buzzer, and servo.	Response within 2 s
R5	The system shall support automatic threshold-based control.	Correct action in at least 90% of test cases
R6	The system shall connect to Wi-Fi through ESP8266.	Successful connection within 30 s
R7	The system shall upload sensor data through MQTT.	At least one successful upload every 5 s
R8	The Android app shall receive and display sensor data.	Data update within 5 s
R9	The Android app shall send control commands to the STM32 system.	Device response within 2 s
R10	The camera page shall display the ESP32-CAM stream or access information.	Successful display under the same local network
R11	The face-related function shall detect or compare a face template.	Functional demonstration under stable lighting
R12	The buzzer shall activate during smoke or alarm conditions.	Alarm response within 2 s

4.2 Verification Procedures

The verification plan was designed to be reproducible and quantitative. Temperature and humidity were compared with a reference meter. Light and smoke values were checked with stimulus response tests. OLED refresh, actuator response, Wi-Fi connection, MQTT upload, and command latency were measured by repeated timing tests. Automatic control was evaluated by threshold trials. Camera and face-related functions were tested under a same-network condition and controlled lighting.



4.3 Quantitative Results

ID	Measured Result	Evaluation
R1	Temperature error approximately 1-2 C; humidity error approximately 3-5% RH	Pass
R2	Light and smoke values increased when physical stimulus increased	Pass
R3	OLED display updated within approximately 1 s	Pass
R4	Actuators responded within approximately 1-2 s	Pass
R5	18 out of 20 threshold trials were correct; success rate = 90%	Pass
R6	Wi-Fi connection completed within approximately 20-30 s	Pass
R7	MQTT sensor data uploaded approximately every 5 s	Pass
R8	Application data updated within approximately 3-5 s	Pass
R9	Remote command response occurred within approximately 1-2 s	Pass
R10	Camera stream displayed when phone and ESP32-CAM were on the same network	Pass
R11	Face function worked under stable lighting but degraded under poor lighting	Partial Pass
R12	Buzzer activated within approximately 1 s after the smoke threshold was exceeded	Pass

5. Conclusion

5.1 Accomplishments

The project successfully developed a functional IoT-based smart home monitoring and control prototype. The final system integrates an STM32 microcontroller, environmental sensors, actuators, OLED display, ESP8266 Wi-Fi module, MQTT communication, Android application, and ESP32-CAM function. It can monitor indoor environmental parameters, show values locally and remotely, control household devices, support automatic and manual operation, and demonstrate camera/security functions.

5.2 Uncertainties and Limitations

The DHT11 sensor has limited accuracy and slower response compared with higher-grade sensors. The LDR and MQ-2 values are normalized ADC percentages rather than calibrated physical units, so the system is suitable for relative monitoring and threshold control rather than precision measurement. MQTT communication depends on Wi-Fi stability and broker availability. The face-related function is affected by lighting, camera angle, network quality, and template quality. These limitations do not prevent the prototype from demonstrating the target functionality, but they should be addressed before practical deployment.

5.3 Future Work and Alternatives

Future improvements include replacing DHT11 with a more accurate sensor such as DHT22 or SHT30, calibrating the MQ-2 sensor with known gas concentrations, improving actuator drivers with MOSFETs or isolated relay modules, adding a private MQTT server, implementing secure authentication, storing historical data, improving the Android user interface, and designing a protective enclosure for the hardware.

5.4 Ethical Considerations

The system must protect user safety and privacy. Electrical loads should be driven through safe driver circuits and should not be connected directly to high-voltage household appliances without certified protection. The ESP32-CAM and face-related functions involve visual information, so users should be informed when the camera is active and face data should be handled securely. The prototype should not claim to replace certified smoke detectors or professional security systems. These limitations should be clearly communicated to avoid misleading users.

6. Final Summary

HOMI is a complete smart home prototype that demonstrates sensing, embedded control, wireless communication, mobile application development, and basic security monitoring. The system satisfies the major functional requirements and provides a practical demonstration of IoT technology in a home environment. With improved calibration, security, enclosure design, and long-term testing, the prototype can be developed into a more reliable smart home system.