

STM32 Based Household Electricity Monitoring and Remote Control System

Qiming Cheng, Xingyu Chen, Bowen Gao, Zhirong Song, Haowen Qu

School of Mechanical and Vehicle Engineering, Changchun University, Changchun 130022, Jilin, China

Corresponding Author: Xiaotong Zhang

ABSTRACT: To address decentralized control of household electrical appliances, the lack of centralized monitoring, and delayed fault alarms, an STM32 based household electricity monitoring and remote control system is designed and implemented. The system uses an STM32F103C8T6 microcontroller, an ESP8266-01S module for WiFi connectivity, and an HLW8032 chip for voltage and current measurement. Four relays switch electrical appliances independently, an OLED provides a local display, and a buzzer signals abnormal conditions. An Android application communicates with the device through MQTT, enabling remote status monitoring, relay control, overvoltage, undervoltage, and overcurrent threshold configuration, and power trend visualization. The system implements four independently controlled channels, remote monitoring, real-time measurement, and fault alarms. Testing indicated stable operation, timely data uploads, and prompt responses to commands. The system meets the basic requirements for centralized household electricity monitoring and remote control and offers practical application potential.

KEYWORDS: STM32; household electricity monitoring; remote control; MQTT; smart home; lightweight design

Date of Submission: 09-10-2026

Date of acceptance: 20-10-2026

I. INTRODUCTION

1.1 Research Background and Significance

The rapid development of the smart home industry has led to a continued increase in the number of household electrical appliances. Air conditioners, water heaters, routers, and other devices operate continuously or remain on standby, resulting in fragmented electricity management and limited visibility into energy consumption. In 2025, China's smart home market exceeded CNY 1.2 trillion, with a compound annual growth rate of 18.7%; the penetration of whole-home smart systems is projected to exceed 35% in 2026. At the national level, policies also promote the integration of smart homes with green and low-carbon development. Six government departments, including the Ministry of Industry and Information Technology, have explicitly called for national standards on smart home interoperability and encouraged the development of integrated solutions for intelligent and environmentally sustainable homes. Nevertheless, household electricity management still has several shortcomings. Appliance controls are distributed and lack unified management. Most households can view only total electricity consumption and cannot readily obtain the voltage, current, and power of individual appliances. Some smart sockets provide limited functionality or depend on proprietary cloud platforms, restricting compatibility. Delayed warnings of abnormal voltage or current also create safety risks. A low-cost, expandable system supporting remote electricity monitoring and control therefore has practical significance.

1.2 Related Research

Research on home energy management systems began earlier outside China and has progressed in energy forecasting and appliance scheduling. However, high system complexity and hardware costs can hinder adoption. Studies in China frequently use embedded platforms such as STM32 for remote control and electricity monitoring. One design uses an STM32 controller and NB-IoT communication to provide remote socket control and threshold alarms [1]. Another combines an STM32 controller, an energy metering chip, and a WiFi module connected to a cloud platform for energy data acquisition and remote monitoring [2]. MQTT based smart home systems are also widely used because of their lightweight operation and low bandwidth requirements [3], and numerous WiFi controlled sockets based on the ESP8266 have been implemented [4]. Existing solutions nevertheless remain limited in independent multichannel control and integrated fault alarms.

1.3 Work Presented in This Paper

To address these limitations, this paper designs and implements a household electricity monitoring and remote control system based on the STM32F103C8T6. An HLW8032 measures voltage, current, and power, while four relays independently switch electrical appliances. An ESP8266-01S module provides WiFi access, and MQTT enables data exchange with an Android application. The application supports remote status monitoring, relay control, overvoltage, undervoltage, and overcurrent threshold configuration, and power trend display. An OLED locally displays real-time measurements, and a buzzer sounds when a parameter exceeds its threshold. The modular design has a low cost and does not depend on a proprietary cloud platform. It balances measurement accuracy, control flexibility, and ease of deployment, providing a reference for similar household electricity monitoring systems.

II. OVERALL SYSTEM DESIGN

2.1 Functional Requirements

The system is required to provide four independently controlled channels, remote monitoring, real-time measurement, and fault alarms to meet practical household electricity monitoring and control needs. The specific functional and performance requirements are listed in Table 1.

Table 1. System functions and performance requirements

#	Function	Functional requirement	Performance requirement
1	Four-channel control	Independently switch four 220 V loads using relays	Relay response ≤ 100 ms; local and remote control
2	Remote monitoring	View device status and switch loads remotely over WiFi using the application	Indoor range ≥ 30 m; command response ≤ 2 s
3	Real-time measurement	Measure voltage, current, and power; display on OLED and upload to application	Voltage error $\leq 1\%$; current error $\leq 2\%$
4	Fault alarms	Sound buzzer when voltage or current crosses a threshold	Alarm response ≤ 500 ms; remotely configurable thresholds
5	Data visualization	Display real-time data and power trends in the application	Refresh period ≤ 2 s; historical trend viewing

2.2 Overall Architecture

The system adopts a layered, modular architecture comprising sensing and actuation, control, communication, and application layers. The sensing and actuation layer acquires voltage and current signals and switches electrical appliances. The control layer uses the STM32F103C8T6 to parse data, evaluate thresholds, drive relays, and update the OLED display. The communication layer connects to the home WiFi network through the ESP8266-01S and exchanges data with the Android application using MQTT. The application layer provides status display, remote control, threshold configuration, and power trend visualization. The clear division of responsibilities among the four layers facilitates debugging and subsequent expansion.

2.3 Microcontroller Selection

As the core of the system, the microcontroller must support data processing, multiple GPIO control signals, UART communication, and low-power operation. Common options are compared in Table 2.

Table 2. Comparison of microcontrollers

Item	STM32F103C8T6	AT89C51	ESP32
Core	ARM Cortex-M3	8051	Xtensa LX6
Clock frequency	72 MHz	12 MHz	240 MHz
GPIO pins	37	32	34 (some multiplexed)
UART interfaces	3	1	3
ADC	12-bit	No built-in ADC	12-bit
Development difficulty	Moderate	Low	Moderate
Cost	Approx. CNY 10	Approx. CNY 5	Approx. CNY 15
Applications	Industrial control; IoT nodes	Simple control	High-performance IoT

The STM32F103C8T6 provides a 72 MHz clock, three UART interfaces, and 37 GPIO pins. These resources meet the requirements for multichannel relay control, HLW8032 data reception, and ESP8266 communication. Its moderate cost and extensive development resources led to its selection as the main controller.

2.4 Communication Method Selection

The system must upload monitoring data to the Android application and receive control commands. Common short-range communication methods are compared in Table 3.

Table 3. Comparison of communication methods

Item	WiFi (ESP8266)	Bluetooth (HC-05)	ZigBee
Range	50–100 m	Approx. 10 m	10–100 m
Data rate	High	Moderate	Low
Internet access	Direct	Not direct	Gateway required
Direct phone connection	Supported	Supported	Usually unsupported
Power consumption	High	Low	Low
Cost	Approx. CNY 8	Approx. CNY 15	Approx. CNY 20
Applications	Remote monitoring	Short-range control	Sensor networking

Bluetooth has a short communication range and cannot meet the intended household remote monitoring requirements. ZigBee requires an additional gateway, increasing system complexity and cost. WiFi connects directly to a home router and enables communication with the application through MQTT without a dedicated gateway. The low cost of the ESP8266 and the availability of development resources further support the selection of the ESP8266-01S WiFi module.

III. HARDWARE CIRCUIT DESIGN

3.1 Power Supply Circuit

The system is powered by a 12 V lithium battery connected through a DC socket. Two voltage regulation stages are used. First, an L7805CD2T-TR reduces 12 V to 5 V to supply the relays, buzzer, and OLED. An RT9013-33GB then reduces 5 V to 3.3 V for the STM32F103C8T6 and ESP8266-01S. The power supply circuit is shown in Fig. 1.

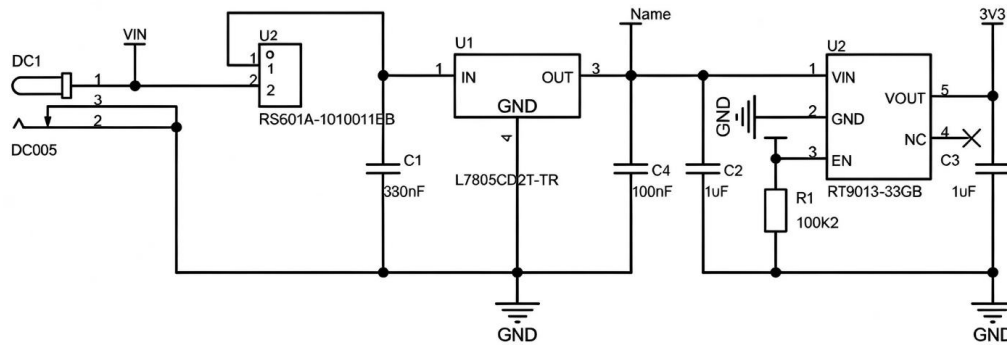


Fig. 1. Power supply circuit schematic

A 330 nF capacitor is connected across the input to filter high-frequency power supply noise. A 100 nF capacitor at the L7805 output suppresses output voltage fluctuations. The RT9013 input and output each use a 1 μF capacitor to stabilize the 3.3 V supply. A 100 kΩ resistor is connected at the EN pin to maintain the enabled state.

The power dissipation and efficiency of a linear regulator are calculated as follows:

$$P_{LDO} = (V_{in} - V_{out}) \times I_{out}$$

$$\eta = (V_{out} \times I_{out}) / (V_{in} \times I_{in}) \times 100\%$$

The cutoff frequency of the filter capacitor can be expressed as:

$$f_c = 1 / (2\pi RC)$$

where R is the equivalent load resistance and C is the filter capacitance. The voltage levels and principal loads are listed in Table 4.

Table 4. Power distribution

Voltage	Conversion device	Main loads	Filter capacitors
12 V	Direct lithium battery supply	Relay coils	330 nF
5 V	L7805CD2T-TR	Relays, buzzer, OLED	100 nF
3.3 V	RT9013-33GB	STM32, ESP8266	1 μF × 2

3.2 STM32 Minimum System Circuit

The minimum system comprises the STM32F103C8T6, a crystal oscillator, a reset circuit, BOOT configuration, and an SWD programming interface. An 8 MHz external high-speed crystal is multiplied to 72 MHz by the internal PLL to provide the system clock. The active-low reset circuit uses a 10 kΩ pull-up resistor and a 100 nF capacitor. BOOT0 is grounded through a jumper so that the controller boots from Flash by default. SWDIO and SWCLK are brought out for programming and debugging. The minimum system circuit is shown in Fig. 2.

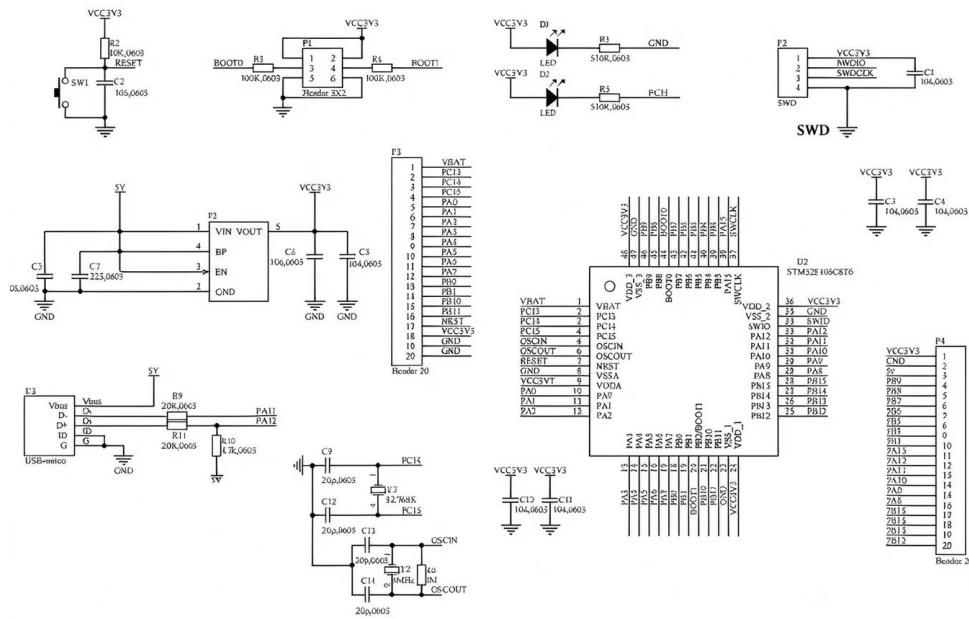


Fig. 2. STM32 minimum system circuit

The STM32 GPIO assignments are listed in Table 5.

Table 5. Main STM32 pin assignments

Pin	Function	Connected component
PA6	Relay control 1	Transistor base
PA7	Relay control 2	Transistor base
PB0	Relay control 3	Transistor base
PB1	Relay control 4	Transistor base
PA2/PA3	UART2	ESP8266-01S
PB12/PB13	I2C	OLED display
PA8	GPIO output	Buzzer driver
PB11	UART reception	HLW8032 voltage and current data

3.3 ESP8266 Communication Circuit

The ESP8266-01S communicates with the STM32 through UART2 and operates at 3.3 V. Its TX pin connects to PA3 (RX), and its RX pin connects to PA2 (TX), enabling bidirectional serial communication. A 0 Ω resistor is placed in series with RST as an isolation and configuration provision. During debugging, RST can be pulled low independently to reset the module; when reset control is unnecessary, the connection can be opened to prevent unintended triggering. EN is connected to 3.3 V to enable the module. The communication circuit is shown in Fig. 3.

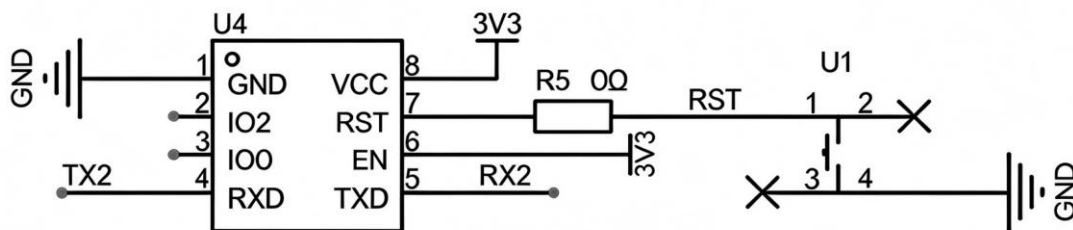


Fig. 3. ESP8266-01S communication circuit

3.4 Relay Driver Circuit

The system uses four relays to switch 220 V loads, controlled by PA6, PA7, PB0, and PB1. Because the STM32 GPIO output current is insufficient to drive a relay coil directly, an S8050 NPN transistor is used as the driver. A 1.5 k Ω current-limiting resistor is connected in series with the base, the collector connects to the relay coil, and the emitter is grounded. A high GPIO output turns the transistor on and energizes the relay; a low output turns the transistor off and releases the relay. The driver circuit is shown in Fig. 4.

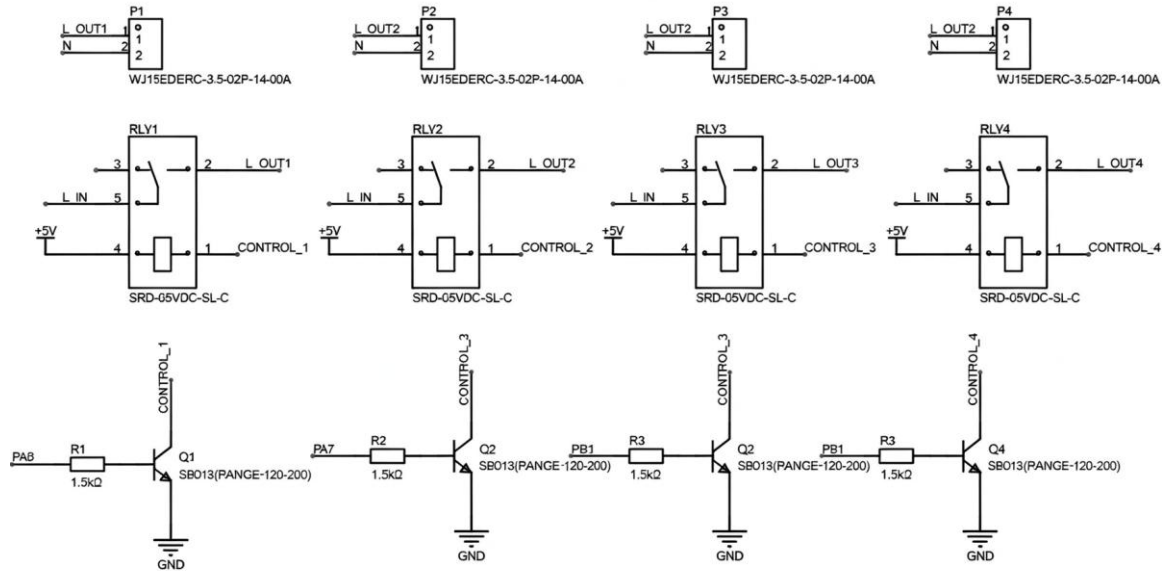


Fig. 4. Relay driver circuit

The base current is calculated as:

$$I_B = (V_{GPIO} - V_{BE}) / R_B = (3.3V - 0.7V) / 1.5k\Omega \approx 1.73mA$$

With an approximate S8050 current gain β of 120, the available collector current is:

$$I_C = \beta \times I_B = 120 \times 1.73mA \approx 208mA$$

The relay coil current is:

$$I_{coil} = V_{coil} / R_{coil}$$

The driver must satisfy:

$$I_C \geq I_{coil}$$

The upper limit of the base resistance is:

$$R_B \leq (V_{GPIO} - V_{BE}) \times \beta_{min} / I_{coil}$$

3.5 Voltage and Current Measurement Circuit

Voltage and current are measured using the HLW8032 energy metering chip, which outputs data through UART. Two integrated ADC channels acquire voltage and current signals separately. The voltage channel uses a resistor divider, while the current channel uses a manganin shunt or a current transformer. The HLW8032 data frame contains voltage, current, power, and electrical energy parameters. The STM32 receives and parses these data through the UART input on PB11. The measurement circuit is shown in Fig. 5.

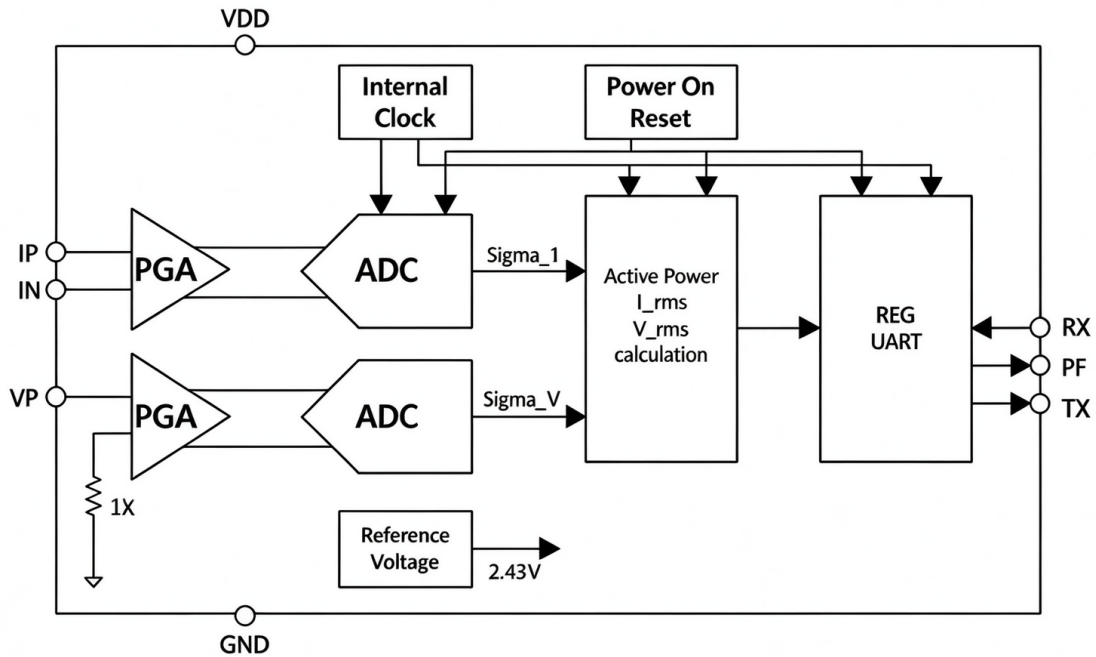


Fig. 5. HLW8032 voltage and current measurement circuit

The voltage divider relationships are:

$$V_{adc} = V_{in} \times R_2 / (R_1 + R_2)$$

$$V_{in} = V_{adc} \times (R_1 + R_2) / R_2$$

The current sensing relationship is:

$$I = V_{shunt} / R_{shunt}$$

The discrete expressions for root-mean-square voltage and current are:

$$V_{rms} = \sqrt{(1/N) \times \sum v^2[n]}$$

$$I_{rms} = \sqrt{(1/N) \times \sum i^2[n]}$$

Active power is calculated as:

$$P = (1/N) \times \sum v[n] \times i[n]$$

Apparent power and power factor are calculated as:

$$S = V_{rms} \times I_{rms}$$

$$\lambda = P / S$$

Accumulated electrical energy is calculated as:

$$E = \sum P_k \times \Delta t$$

The HLW8032 data scaling relationships are:

$$V_{rms} = K_V \times (V_{ADC} / 2^{24}) \times V_{ref}$$

$$I_{rms} = K_I \times (I_{ADC} / 2^{24}) \times V_{ref}$$

$$P = V_{rms} \times I_{rms} \times \cos\phi$$

where K_V and K_I are the calibration coefficients for the voltage and current channels, respectively, and V_{ref} is the reference voltage.

3.6 OLED Display Circuit

A 0.96-inch OLED module with an I2C interface connects to the STM32 through PB12 (SCL) and PB13 (SDA). The I2C communication rate is set to 400 kHz. The OLED locally displays voltage, current, power, and relay states. The circuit is shown in Fig. 6.

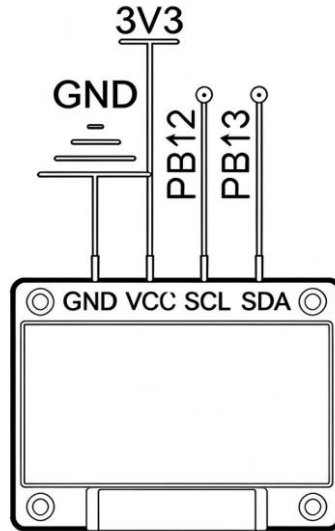


Fig. 6. OLED display circuit

3.7 Buzzer Alarm Circuit

A 5 V active buzzer is driven by an S8050 transistor with a 1.5 k Ω series base resistor and is controlled by STM32 pin PA8. When overvoltage, undervoltage, or overcurrent is detected, PA8 outputs a high level, turning on the transistor and sounding the buzzer. The circuit is shown in Fig. 7.

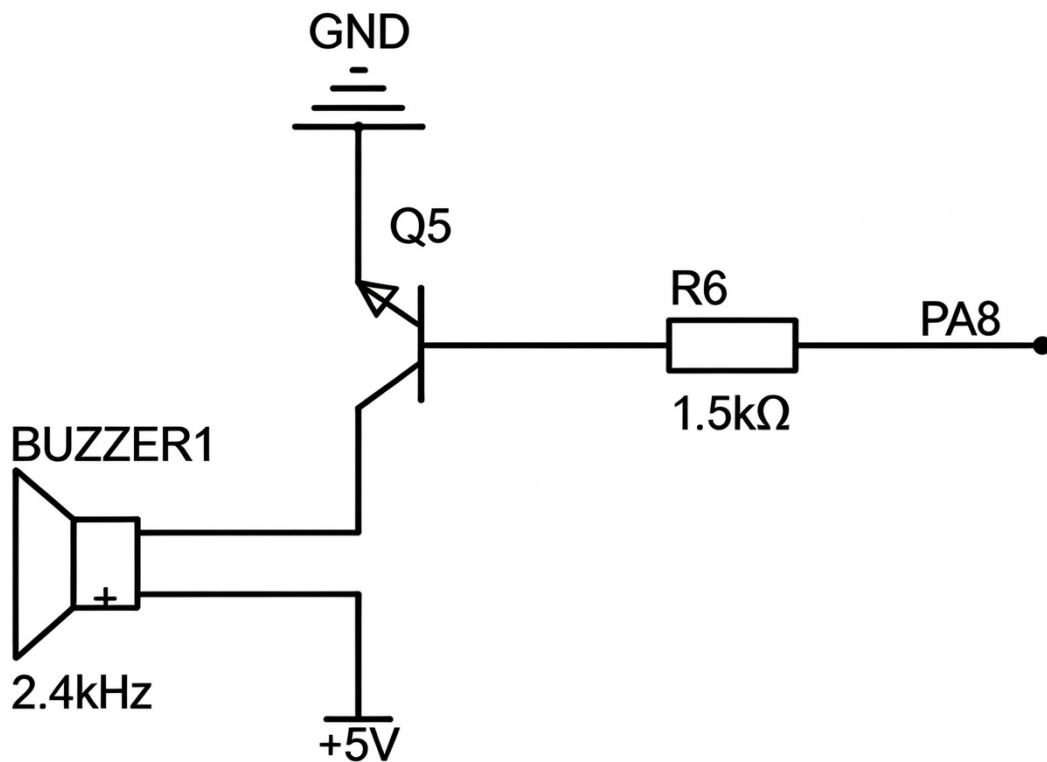


Fig. 7. Buzzer alarm circuit

The buzzer driver current satisfies:

$$I_B = (V_{GPIO} - V_{BE}) / R_B$$

$$I_C = \beta \times I_B \geq I_{buz}$$

where I_{buz} is the rated operating current of the buzzer.

IV. SOFTWARE DESIGN

4.1 Development Environment

The software consists of firmware for the STM32 and an Android application. The firmware is developed in C using Keil MDK and handles data acquisition, relay control, OLED display, and communication message transmission and reception. The Android application communicates with the device using MQTT and provides remote monitoring and control. Messages use JSON format. The development environment is summarized in Table 6.

Table 6. Software development environment

Category	Item	Description
STM32 firmware	Development tool	Keil MDK
STM32 firmware	Programming language	C
STM32 firmware	Communication protocols	UART, I2C, MQTT
Application	Platform	Android
Application	Communication protocol	MQTT
Communication	Broker	tcp://broker.emqx.io:1883
Communication	Message format	JSON

4.2 Main Program Flow

After power-up, the STM32 initializes the clock, GPIO, UART1, UART2, and I2C. It then sends AT commands to the ESP8266 through UART2 to connect to the home WiFi network, establishes a connection to the MQTT broker, and subscribes to the control and configuration topics. In the main loop, the STM32 periodically receives voltage, current, and power data from the HLW8032. After parsing the data, it updates the OLED and checks the configured thresholds. If a threshold is crossed, the buzzer sounds. Meanwhile, the STM32 publishes real-time data to the status topic and monitors the control topic. On receiving a relay command, it drives the corresponding GPIO.

4.3 WiFi Module and Communication

The ESP8266-01S connects to the STM32 through UART2, with PA2 and PA3 connected to the module's RX and TX pins, respectively. The STM32 configures WiFi using AT commands: it first selects Station mode, connects to the home router, and then establishes a connection to the MQTT broker. MQTT provides lightweight communication, low bandwidth usage, and a publish/subscribe model, making it suitable for smart home device communication [3]. The device and application exchange JSON messages containing voltage, current, power, and the states of all four relays. The application sends control commands and threshold settings, which the device parses and executes.

4.4 Data Acquisition

The HLW8032 sends data frames to STM32 pin PB11 through UART. Each frame contains voltage, current, power, and other parameters. The STM32 validates each received frame, extracts the valid data, and calculates the actual voltage, current, and power using the scaling relationships described in the hardware design section. The acquisition period is set to 1 s:

$$T_{\text{sample}} = 1 \text{ s}$$

$$f_s = 1 / T_{\text{sample}} = 1 \text{ Hz}$$

The acquired data are displayed on the OLED and compared with the thresholds configured through the application. The alarm conditions are:

$$V_{\text{over}}: V_{\text{rms}} > V_{\text{th_over}}$$

$$V_{\text{under}}: V_{\text{rms}} < V_{\text{th_under}}$$

$$I_{\text{over}}: I_{\text{rms}} > I_{\text{th_over}}$$

The alarm output can be expressed as:

$$A = 1, \text{ if } V_{\text{over}} \cup V_{\text{under}} \cup I_{\text{over}}$$

$$A = 0, \text{ otherwise}$$

When $A = 1$, PA8 outputs a high level to drive the buzzer alarm.

4.5 Application Functions

The Android application works with the STM32 firmware to provide remote monitoring and control. Users can view real-time voltage, current, and power uploaded by the device, switch each of the four relays, configure overvoltage, undervoltage, and overcurrent thresholds, and view power trends. The application communicates with the device through MQTT and displays connection status and operation results. As the application layer, it works with the STM32 to provide remote monitoring and control of household electricity use.

V. SYSTEM DEBUGGING AND RESULTS ANALYSIS

5.1 Debugging Procedure

Debugging comprised module tests followed by integrated system testing. During module testing, the power supply circuit was energized separately to confirm a normal 12 V input, a 5 V L7805 output, and a 3.3 V RT9013 output, with stable voltages at both regulation stages. A test program was then loaded into the STM32 minimum system to verify crystal oscillation, reset operation, and GPIO control. Each peripheral was tested separately. Control signals were applied to the relay module to confirm that all four relays energized and released correctly. The OLED displayed characters and numerical values correctly through I2C. After voltage and current signals were applied to the HLW8032, the STM32 successfully received serial data frames. The buzzer sounded when its control input was high. The ESP8266 connected to the home WiFi network using AT commands and established a connection to the MQTT broker.

During integrated testing, all modules were connected to the controller board, the complete firmware was programmed, and the system was powered on. Initialization and WiFi connection completed automatically. The OLED displayed real-time voltage, current, and power, and all four relays were initially off. After connecting to the MQTT broker, the application received status messages and displayed the current measurements and relay states. Operating a relay switch in the application promptly actuated the corresponding relay. Overvoltage, undervoltage, and overcurrent thresholds set in the application were received and stored by the device, and the buzzer sounded when a measured value crossed its threshold. The system remained stable during integrated testing, communication among modules was normal, and no obvious abnormalities were observed.

5.2 Functional Verification

The principal functions were verified as follows. (1) Independent four-channel control: the application individually controlled the relays associated with PA6, PA7, PB0, and PB1. All four channels switched independently without affecting one another, meeting the requirement for separate control of multiple appliances. (2) Remote monitoring: the application communicated with the device through MQTT, remotely displayed voltage, current, power, and relay states, and sent control commands. (3) Real-time measurement: the HLW8032 acquired voltage, current, and power data, which the STM32 parsed, displayed locally on the OLED, and uploaded to the application. Data refreshed normally, meeting the real-time monitoring requirement. (4) Fault alarms: when voltage or current crossed a configured threshold, the buzzer sounded and the application displayed the alarm state at the same time.

5.3 Results Analysis

The debugging and functional verification results indicated that the system broadly met its design objectives. The STM32 operated stably while handling data acquisition, relay control, OLED updates, and MQTT communication. The ESP8266 provided reliable data exchange between the device and the application. The HLW8032 voltage and current measurement module produced normal data output and met the basic accuracy needs of household electricity monitoring. The four relay driver circuits operated correctly, with the S8050 transistors reliably driving the relay coils. The OLED and buzzer responded promptly.

Measurement error can be expressed using absolute and relative error:

$$\Delta X = X_m - X_0$$

$$\delta = (\Delta X / X_0) \times 100\%$$

The mean absolute relative error is:

$$\bar{\delta} = (1/n) \times \sum |\delta_i|$$

The command response time can be expressed as:

$$t_r = t_2 - t_1$$

where t_1 is the time at which the command is sent and t_2 is the time at which the device responds.

Several limitations remain. First, communication depends on a public MQTT broker, so stability and security require improvement. Second, voltage and current measurements have not been calibrated using

professional instruments, and actual measurement errors require further testing. Third, application functions are relatively basic, with scope to improve the interface and user interaction. Overall, the system implements the intended independent four-channel control, remote monitoring, real-time measurement, and fault alarm functions, demonstrating the feasibility of the design.

VI. CONCLUSIONS AND FUTURE WORK

6.1 Summary

This paper designs and implements an STM32 based household electricity monitoring and remote control system to address fragmented appliance control, the lack of centralized monitoring, and delayed fault alarms. The STM32F103C8T6 serves as the main controller. An HLW8032 acquires voltage, current, and power data, four relays independently switch electrical appliances, and an ESP8266-01S connects to home WiFi. MQTT enables data exchange with the Android application. An OLED locally displays real-time data, while a buzzer sounds when a parameter crosses its configured threshold. Four core functions are implemented. First, PA6, PA7, PB0, and PB1 control four relays independently to switch four 220 V loads. Second, the application remotely displays device status and controls relay states through MQTT. Third, voltage, current, and power measurements from the HLW8032 are displayed on the OLED and simultaneously uploaded to the application. Fourth, abnormal voltage or current triggers the buzzer and a corresponding notification in the application. Each module was tested separately before integrated testing. After power-up, initialization and WiFi connection completed automatically, the OLED displayed real-time data, and all relays defaulted to the off state. Once connected to the MQTT broker, the application received status data and sent relay control and threshold configuration commands. Overall operation was stable, intermodule communication was normal, and the main design objectives were achieved.

6.2 Distinctive Features

The project has three main features. (1) Independent multichannel control is combined with centralized monitoring. Four relays control household appliances separately, while the HLW8032 centrally acquires voltage, current, and power data. Combining distributed control with centralized monitoring in one system provides greater practical functionality than a single-purpose smart socket. (2) MQTT enables lightweight remote communication. The ESP8266-01S connects to home WiFi and communicates with the application through MQTT without depending on a proprietary cloud platform. This reduces operating costs and data privacy risks while supporting flexibility and expansion. (3) The modular design is low-cost and suitable for wider adoption. Relatively independent functional modules simplify debugging and maintenance. Overall hardware costs are kept low, making the design suitable for ordinary households and a useful reference for similar electricity monitoring systems.

6.3 Limitations and Improvements

The system has four main limitations. (1) Communication security requires improvement. Communication currently uses a public MQTT broker, and data transmission is unencrypted, creating security risks. Future work could use a private broker with TLS encryption and user authentication. (2) Measurement accuracy has not been professionally calibrated. HLW8032 output has not been compared and calibrated against a reference electricity meter, and actual measurement errors have not been quantified. Professional instruments could be used to calibrate the voltage and current channels. (3) Application functionality is relatively basic. Current functions are limited to data viewing, relay control, threshold configuration, and other basic operations. The interface and user experience could be improved, with additional historical data queries, scheduled control, and electricity consumption statistics. (4) System expansion is limited. Only four control channels are currently supported. Additional channels or networking methods such as ZigBee could extend the control range.

6.4 Future Outlook

Continued advances in smart home and Internet of Things technologies create broad application prospects for household electricity monitoring and remote control. Future work can improve measurement accuracy through professional calibration to meet household energy metering requirements; strengthen communication security through TLS encryption and private servers to protect user data; enrich the application with electricity consumption statistics, usage analysis, and scheduled control; and expand the system to support more appliances and integrate photovoltaic generation, energy storage, and other new energy devices, progressing toward a home energy management system.

REFERENCES

- [1] Tang, H. Design and implementation of an NB-IoT smart socket overload protection system. *Household Appliances*, 2025, 67(12): 132–135. DOI: 10.3969/j.issn.1673-6079.2025.12.026. (In Chinese.)

- [2] Xiong, H. Design of a household electricity monitoring system under the smart grid. Thesis, Hunan University, Changsha, 2012. (In Chinese.)
- [3] Chen, Y., Feng, X., and Chen, S. A modular expansion system for smart homes based on MQTT. *Integrated Circuits and Embedded Systems*, 2024, 24(2): 86–90. (In Chinese.)
- [4] Dai, L., Li, H., Jin, H., et al. Design of a WiFi controlled socket based on ESP8266. *Equipment Manufacturing Technology*, 2017(4): 12–13, 23. DOI: 10.3969/j.issn.1672-545X.2017.04.004. (In Chinese.)
- [5] Wang, R. Design of a smart home control system based on STM32. *Automation Today*, 2026(6): 1–3. DOI: 10.3969/j.issn.2095-6487.2026.06.002. (In Chinese.)
- [6] Cai, X., Yuan, K., Liu, J., et al. Design of a smart home control system based on the STM32F103 microcontroller. *China Plant Engineering*, 2023(1): 28–30. DOI: 10.3969/j.issn.1671-0711.2023.01.010. (In Chinese.)
- [7] Li, Y., and Lin, D. Design of a multifunctional smart socket. *Technology Innovation and Application*, 2017(03). (In Chinese.)
- [8] Zhang, X., and Xiang, X. Research and design of a smart home system based on the MQTT communication protocol. *Practical Electronics*, 2020(3): 37–38, 5. DOI: 10.3969/j.issn.1006-5059.2020.03.013. (In Chinese.)
- [9] Yang, Z., and Qiao, H. A smart home remote control system based on STM32 and MQTT. *Electronics World*, 2019(21): 186–187. (In Chinese.)