НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:
Завідувач кафедри
_____ Сергій СТІРЕНКО
(підпис)
“___” _____2024 р.

Дипломний проект

на здобуття ступеня бакалавра

за освітньо-професійною програмою “Комп’ютерні системи та мережі”
спеціальності 123 “Комп’ютерна інженерія”

на тему: Система постійної температури і вологості у домі

Виконав (-ла): студент (-ка) 4 курсу, групи ІО-06
(шифр групи)

Юемін Лі _____
(прізвище, ім’я, по батькові) (підпис)

Керівник к.т.н., доцент Павлов В. Г. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант (нормоконтроль) к.т.н., доцент Павлов В. Г. _____
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент д.т.н., проф. Корнієнко Б. Я. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2024 р.

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти – перший (бакалавр)
Освітньо-професійна програма
“Комп’ютерні системи та мережі”
спеціальність 123 “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сергій СТІРЕНКО
(підпис) (ініціали, прізвище)
“___” _____2024 р.

ЗАВДАННЯ
на бакалаврський дипломний проект студента

Юемін Лі
(прізвище, ім’я, по батькові)

1. Тема проекту *Система постійної температури і вологості у домі*
керівник проекту Павлов В. Г., к.т.н., доцент,
(прізвище, ім’я, по батькові, науковий ступінь, вчене звання)
затверджені наказом по університету від 08.05 2024 року № 51/24-сі

2. Строк подання студентом проекту 19.06.2024

3. Вихідні дані до проекту *технічна документація. теоретичні та статистичні дані, патенти на винахід*

4. Зміст пояснювальної записки (перелік завдань, які потрібно розробити)
Опис предметної області, дослідження методики побудови електронного підручника на базі гіпертекстової технології, програма забезпечення гіпертекстових технологій

5. Перелік графічного матеріалу (з точним позначенням обов’язкових креслень) структурна схема системи, узагальнена схема роботи системи, схема алгоритму модуля редактора, схема _____ інформаційних вікон редактора.

6. Консультанти розділів проекту

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Павлов В. Г.		

7. Дата видачі завдання 01.10.2023

Календарний план

	Назва етапу дипломного проекту	Строк виконання етапу проекту	Примітка
1.	*Затвердження теми проекту*	*14.12.2023-02.04.2024*	
2.	*Вивчення та аналіз завдання*	*03.04.2024-14.04.2024*	
3.	*Розробка архітектури та загальної структури системи*	*15.04.2024-24.04.2024*	
4.	*Розробка структур окремих підсистем*	*25.04.2024-4.05.2024*	
5.	*Програмна реалізація системи*	*5.05.2024-15.05.2024*	
6.	*Оформлення пояснювальної записки*	*15.05.2024-19.05.2024*	
7.	*Захист програмного продукту*	*26.05.2024*	
8.	*Передзахист*	*11.06.2024*	
9.	*Захист*	*19.06.2024*	

Студент _____ Юемін Лі
(підпис) (ініціали, прізвище)

Керівник проекту _____ Павлов В. Г.
(підпис) (ініціали, прізвище)

Анотація

З популяризацією концепції розумних будинків і розвитком технології Інтернету речей системи постійної температури і вологості вдома стали важливим компонентом розумних будинків і отримали широку увагу. Хоча традиційні системи контролю температури і вологості вдома можуть задовольнити потреби в контролі температури і вологості, вони все ще мають недоліки в розумінні, енергозбереженні та простоті експлуатації. Система постійної температури і вологості у домі на основі мікроконтролерів привернула велику увагу завдяки своїй високій продуктивності та низькому енергоспоживанню. Він може збирати дані про навколишнє середовище в приміщенні в режимі реального часу, точно контролювати температуру і вологість та полегшувати роботу користувача через інтерфейс взаємодії між людиною та комп'ютером, покращуючи взаємодію з користувачем.

У цій статті розроблено систему постійної температури і вологості у домі на основі STM32 у поєднанні з датчиками температури та вологості для отримання даних про навколишнє середовище в приміщенні в реальному часі. Дані відображаються на OLED-екрані, а система використовує бездротовий модуль ESP8266, який може завантажувати дані в реальному часі на хмарну платформу та здійснювати віддалений моніторинг за допомогою мобільного мобільного додатку. Ця конструкція допомагає усунути недоліки традиційні системи, сприяють розвитку технології розумного дому, покращують комфорт життя та рівень здоров'я, одночасно зменшуючи споживання енергії та досягаючи концепції екологічного життя.

Annotation

With the popularization of the concept of smart homes and the development of Internet of Things technology, home constant temperature and humidity system have become an important component of smart homes and have received widespread attention. Although traditional home constant temperature and humidity system can meet the needs of temperature control, they still have shortcomings in intelligence, energy conservation, and ease of operation. Home constant temperature and humidity system based on microcontrollers has attracted much attention due to its high performance and low power consumption characteristics. It can collect real-time indoor environmental data, accurately control temperature, and facilitate user operation through human-computer interaction interface, improving user experience.

This article designs a home constant temperature and humidity system based on the STM32, combined with temperature and humidity sensors, to obtain real-time indoor environmental data. The data is displayed on an OLED screen, and the system uses the ESP8266 wireless module, which can upload real-time data to the cloud platform and achieve remote monitoring through a mobile app. This design helps to address the shortcomings of traditional systems, promote the development of smart home technology, improve living comfort and health levels, while reducing energy consumption and achieving the concept of green living.

Key Words: STM32; OLED; ESP8266; Remote monitoring.

№ сторінки	Формат	Позначення	Найменування	Кількість	Примітка
			Документація загальна		
1	А4	ІАЛЦ. 467200.001 ОА	Система постійної температури і вологості у домі Опис альбому	1	
2	А4	ІАЛЦ. 467200.002 ТЗ	Система постійної температури і вологості у домі Технічне завдання	3	
3	А4	ІАЛЦ. 467200.003 ПЗ	Система постійної температури і вологості у домі Пояснювальна записка	48	
4	А4	ІАЛЦ. 467200.004 Д1	Система постійної температури і вологості у домі Блок-схема алгоритму дій	1	
5	А4	ІАЛЦ. 467200.005 Д2	Система постійної температури і вологості у домі Діаграма функціональна	1	
6	А4	ІАЛЦ. 467200.006 Д3	Система постійної температури і вологості у домі Діаграма взаємодії	1	
7	А4	ІАЛЦ. 467200.007 Д4	Система постійної температури і вологості у домі Програмний код	23	

Зм.	Арк	№ докум.	Підпис	Дата	ІАЛЦ 467100.001 ОА		
Розроб.		Лі Юемін			Система постійної температурі і вологості у домі	Літ.	Аркуш. 1 / Аркушів 1
Перевір.		Павлов В. Г.					
Реценз.		Корнієнко Б.Я.				КПІ ім. Ігоря Сікорського, ФІОТ, ІО-06	
Н. Контр.		Павлов В. Г.					
Затверд.		Стіренко С. Г.			Опис альбому		

CATALOG

1. NAME AND SCOPE OF USE

The scope of application of the developed system is a tool designed to monitor the temperature and humidity in the House. The tool allows users to get up-to-date information about the temperature and humidity in the house, as well as manage it for optimal results.

2. REASONS FOR DEVELOPMENT

The basis for the development is the task for the implementation of The Bachelor's thesis project, approved by the Department of computer engineering of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".

3. PURPOSES OF DEVELOPMENT

The main goal of developing the tool is to create a reliable, accurate and convenient tool for controlling the temperature and humidity in the House. This includes providing access to in-house data online, temperature and humidity metrics that can help users make decisions..

4. DEVELOPMENT SOURCES

Sources of development are scientific and technical literature, technical documentation, publications in periodicals and the Internet.

5. TECHNICAL REQUIREMENTS

5.1 Requirements for the software product being developed.

The system under development must meet the requirements set out below:

- Real-time data updates
- Data analysis.
- Managing the tool.

5.2 Requriments for the tool software.

To develop a project on a PC, we need the following software:

- Windows.
- Installed VSCord for writing code in the C programming language.

6. DEVELOPMENT STAGES

Назва етапу виконання:	Термін виконання:
Approval of the project topic	*14.12.2023-02.04.2024*
Study and analysis of the task	*03.04.2024-14.04.2024*
Development of the architecture and overall strcture of the system	*15.04.2024-24.04.2024*
Development of individual subsystem structurt	*25.04.2024-4.05.2024*
Software implementaiton of the system	*05.05.2024-15.05.2024*
Making an explanatory note	*15.05.2024-19.05.2024*
Software product passing	*25.05.2024*
Pre-passing	*11.06.2024*
Passing	*19.06.2024*

CATALOG

INTRODUCTION

Authenticity of the topic.In order to keep the temperature and humidity of the home within a reasonable range, reasonable steps are the main way to achieve automation. It is necessary to develop a scientific and practical intelligent system monitoring method and an intelligent system control method to ensure that each component can operate reasonably and efficiently. The function can be automatically executed by the device or manually turned on by the user to ensure that the temperature and humidity remain within a reasonable range.

At present, there is a lack of products that can control temperature and humidity at the same time on the market, and temperature and humidity are important indicators for measuring human comfort. An effective way to solve this problem is to design and make a temperature and humidity control system.

The research objectives of this paper are:

1. Use the temperature and humidity sensor DHT11 to obtain data and control components such as servos, fans, and relays to design a temperature and humidity control system.

2. Use WiFi technology to enable users to monitor data or perform operations through mobile terminals

3. Provide a system design plan, including schematics and circuit flow charts

4. Master the system debugging of the temperature and humidity detection system and realize the required detection methods

5. Debug the system to meet the detection requirements

6. Improve the ability to consult literature and write papers.

The research object is: indoor temperature and humidity measurement, and the process of the system making corresponding decisions.

The research topic is: Home constant temperature and humidity system.

Research method: This paper adopts theoretical methods and experimental research to analyze whether the threshold is reached by detecting the ambient temperature and humidity. The basis of the paper research is: the method of determining parameters and the development of monitoring and control systems.

Solution: When the indoor temperature and humidity change, the temperature and humidity sensor DHT11 feeds back the data to the control system, and the control system performs the corresponding operation control according to the set solution. By controlling the working status of the fan, air conditioner, humidifier, and servo, the indoor temperature can be quickly restored to the set comfort range.

Scientific novelty is as follows: Taking into account the common trend of future science and technology, and the gradual popularization and improvement of the Internet of Things. Developing a home temperature and humidity control system can control the constant temperature and humidity in the room by setting and adjusting, which can be used for practical life applications and has certain expansion capabilities.

Taking into account the common trend of monitoring and automation, a fairly simple wireless temperature and humidity sensor model was developed. Through setting and programming, a system for monitoring changes in these parameters can be realized. The result is a real model of the device and system, which can be used for practical applications and has the ability to be further expanded.

CHAPTER 1. PROLEGOMENON

1.1 Research background

With the concept of smart home becoming more and more popular and the rapid development of Internet of Things technology, the design and research of home constant temperature and humidity system, as an important part of smart home, have received widespread attention. Although traditional home constant temperature and humidity systems can meet people's needs for temperature control to a certain extent, they still have shortcomings in terms of intelligence, energy saving, and ease of operation. Therefore, the design of home constant temperature and humidity system based on single-chip microcomputer came into being and became a hot spot in current research [1].

In daily life, common devices for controlling temperature and humidity include air conditioners, water heating, floor heating, heaters, humidifiers and other products. But they only have one of the functions of a single item. The functionality is not perfect. Doesn't give a good experience. For example, when we turn on the air conditioner in the summer, we often feel dry. For example, when we turn on the water heating in winter, some people may experience chest tightness due to inappropriate humidity. There are many products on the market that can change temperature, and there are also many devices that can change humidity. However, there is always a lack of a device that can combine temperature and humidity well and control temperature and humidity at the same time through the same controller. In an automatic way, the indoor temperature and humidity are always kept within a comfortable range. So I researched a device that could control temperature and humidity at the same time.

1.2 Research significance

As a high-performance, low-power microcontroller, the STM32 microcontroller has powerful computing capabilities and rich peripheral interfaces, providing the possibility for intelligent control of home constant temperature and humidity systems. The home constant temperature and humidity system can collect indoor environment data in real time, perform temperature control, and realize constant temperature adjustment of the indoor environment. At the same time, the system can also facilitate users to perform personalized settings and operations through the human-computer interaction interface, improving user experience. It can solve the shortcomings of traditional constant temperature and humidity systems in terms of intelligence and energy saving, and can also promote the development and application of smart home technology [2]. Through the home constant temperature and humidity system based on STM32 microcontroller, people can control the indoor environment more conveniently and improve the comfort and health of living. At the same time, the intelligent and energy-saving features of the system also help reduce household energy consumption and environmentally friendly life concept.

The significance of this research is to innovate the interaction of various hardware and make up for the shortcomings of temperature and humidity control in the home environment. A highly integrated system meets the conditions for achieving daily functions. While improving the degree of integration, it is easier to observe and operate. Intelligent control of human-computer interaction complies with the needs of the new era and lays the foundation for the home Internet of Things. While improving the experience, it also improves user comfort, saves energy and reduces emissions, achieves low-carbon environmental protection, and promotes users' healthy life and improve the quality of life.

1.3 Research status at home and abroad

Developed countries such as Europe and the United States started research in the field of smart home earlier, and have formed a relatively complete technical system and market structure. Foreign researchers focus on innovation and breakthroughs, and constantly introduce new algorithms and technical means to achieve indoor environment control. For more precise and intelligent control, some well-known smart home companies have launched a variety of home thermostatic system products based on microcontrollers and have gained a good reputation in the market. These products are not only highly intelligent and personalized, but also focus on user experience, energy conservation and environmental protection, and meet the needs of modern families [3].

In recent years, with the continuous expansion of the domestic smart home market and the increasingly mature technology, home thermostatic systems have received widespread attention in the country. Many scientific research institutions and universities have begun to invest in research in this field and have achieved a series of important results. Domestic researchers have achieved precise control of the indoor environment by optimizing algorithms and improving hardware performance. Due to its high performance and low power consumption, microcontrollers have become the preferred solution for home thermostatic system design. At the same time, domestic researchers have also actively explored linkage control with other smart home devices to achieve a more intelligent and convenient living environment [4]. Some smart home companies have begun to launch home thermostatic system products and gradually occupy a place in the market. These products not only have excellent temperature control performance, but also have high cost performance and easy operation, and are favored by consumers.

1.4 Review existing solutions

1.4.1 Household air conditioner

Let’s take a look at common solutions and an analysis of their feasibility.

As one of the most common temperature control solutions, air conditioning has the advantage of being convenient and fast. The air conditioner passes through a compressor, which condenses the air. It can quickly lower the room temperature in summer. In winter, you can use the principle of heating resistance wire to heat the air and blow out hot air to increase the indoor temperature. But at the same time, the shortcomings are also very obvious, and it is impossible to effectively control the air humidity in the room. Any time you use an air conditioner, it removes moisture from the room. This reduces the air humidity and makes people feel dry. While we gain rapid cooling, it comes at the cost of losing moisture.

At the same time, the air conditioning control system consists of a microcontroller, compressor, resistance wire, blower, etc. The microcontroller controls the operation of the compressor blower to blow out cold air. By controlling the resistance wire and the hairdryer, hot air is blown out. However, this system cannot control the temperature and humidity of the air at the same time. While getting temperature control, there is no way to control humidity. Household air conditioner is shown in Figure 1.1. Air-conditioning compressor is shown in Figure 1.2.

Figure 1.1 – Household air conditioner

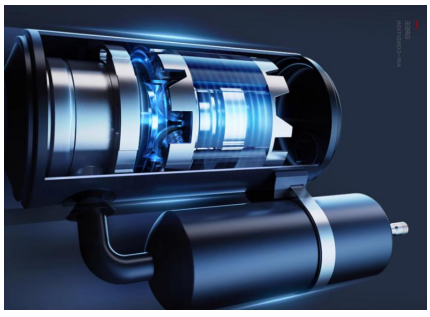


Figure 1.2 – Air-conditioning compressor

1.4.2 Household air heater

Figure 1.3 – Household air heater

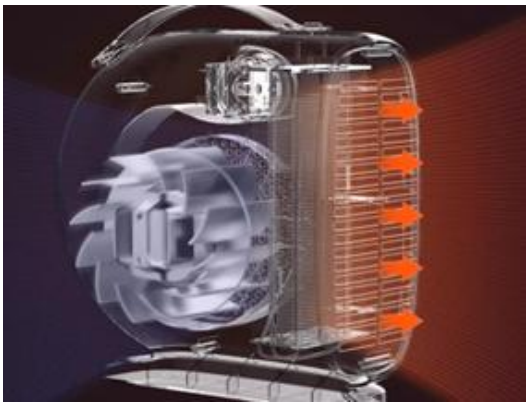


Figure 1.4 – Heater heating principle

As one of the common solutions to heat energy problems in winter, heaters can reach working status in a very short time and continuously output warm air. The advantages are that they are convenient, fast and easy to move. They can be used on desktops or floors to provide hot air where needed. Its shortcomings are also very obvious. For large spaces, the heating effect on the entire room is very slight, and blowing heat onto people's skin for a long time can easily cause local dryness of the skin and even cause diseases. Its working principle is also to heat the air at the expense of losing moisture. But while it heats, it also makes the room drier in the winter.

Its system structure is simpler, and the controller only needs to control the power of the fan and resistance wire, but while it is simple, it also loses functionality. Still unable to control humidity. Household air heater is shown in Figure 1.3. Heater heating principle is shown in Figure 1.4.

1.4.3 Floor heating

Figure 1.5 – Floor heating Figure

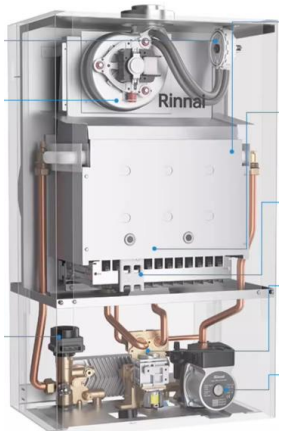


Figure 1.6 – Floor heating schematic diagram

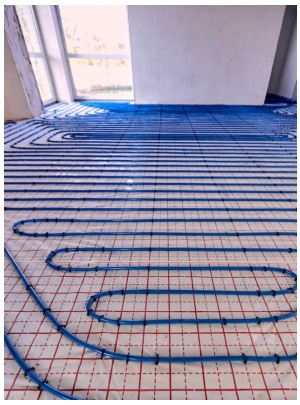


Figure 1.7 – Floor heating pipes

As one of the common heating solutions in northern winter, floor heating can greatly improve indoor comfort. It can spread heat over a wide range inside the floor, keeping the entire room in a stable state of uniform heat. This bottom-up heating method can reduce people's discomfort in the room to a certain extent. Its shortcomings are also obvious, including high energy consumption, slow heating speed, and the need for manual adjustment.

You can clearly see its system structure in the picture. The controller heats the heat pipe and the pump controls the water flow. The hot water flows through the water pipe to bring heat into the room. Its shortcomings are also very obvious. It cannot detect the real-time temperature of the room and adjust the humidity. This system cannot meet the automation needs of today's Internet of Things. Floor heating Figure is shown in Figure 1.5. Floor heating schematic diagram is shown in Figure 1.6. Floor heating pipes is shown in Figure 1.7.

1.4.4 Air humidifier

Figure 1.8 – Air humidifier

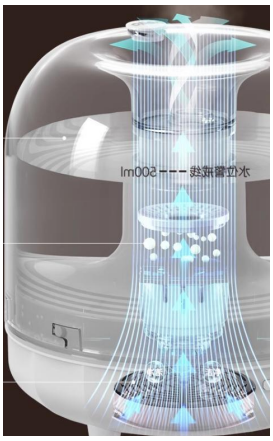


Figure 1.9 – Air humidifier working principle diagram

Air humidifier is a product that is used frequently throughout the year. Its advantage is that it can replenish moisture in the air conveniently and quickly. It uses the principle of atomization to atomize pure water into small water droplets, thereby increasing air humidity. Its shortcoming is also obvious, it requires manual control. And due to design reasons, it cannot accurately calculate the average humidity of the room. Whenever we think about using it, often the air is already excessively dry.

The air humidifier system consists of a simple atomizer, which is turned on or off through switch control. At the same time, it cannot detect humidity or change working status based on humidity. Just human control. The structure of this system is very simple. Air humidifier is shown in Figure 1.8. Air humidifier working principle diagram is shown in Figure 1.9.

Conclusion about 1st CHAPTER

I analyze the various ways of controlling temperature and humidity on the market, and they have their own advantages. However, there is a lack of a comprehensive product on the market that can automatically and reasonably control room temperature and humidity at the same time. With the changes of the times, we have ushered in the era of the Internet of Things. Perhaps we should adapt to the development of the times. In the future development trend, smart home and the Internet of Everything are an area with a huge market. We should adapt to the development of the market. Make products more competitive.

After analyzing the advantages and disadvantages of each, I decided to combine their advantages and develop a controller with networking capabilities to control the temperature and humidity of the room and always maintain the range that is suitable for the human body.

Task: Develop a controller with a network function, select the appropriate single -chip machine, and choose the appropriate component. And assemble, debug, and program on them.

Objective: Develop a controller with networking capabilities, so that the controller can control the temperature and humidity of the room, and the network can upload the data to the cloud. Users can experiment with a single -chip board control or the mobile terminal for remote control. By adjusting the parameters, the temperature and humidity in the room can reach the ideal state of the user.

CHAPTER 2. OVERALL DESIGN SOLUTION

2.1 Overall system scheme

This design uses a single-chip microcomputer as the core control device to accurately obtain the temperature and humidity data of the home environment through external sensors and display it on the LCD screen in real time, making it convenient for users to check the temperature and humidity status of the current environment at any time. The system is also equipped with a simple button operation function, allowing users to easily set temperature and humidity thresholds. When the ambient temperature and humidity exceed the set threshold range, the system will automatically activate the adjustment mechanism to ensure that the indoor temperature and humidity are maintained within the user's desired comfort range. Real-time temperature and humidity readings can be synchronized to the cloud platform through wireless communication, and the system's setting parameters can be easily adjusted using a mobile phone.

Temperature control scheme: When the user sets the temperature threshold on the interface or cloud platform, the system receives the instruction to set the value. If the temperature sensor detects that the temperature is lower than the set value, the steering gear works to open the water valve, and the flow rate sensor monitors the flow rate of the liquid in the heating to increase the indoor temperature. If the temperature sensor detects that the temperature is higher than the set value, the air conditioner will be turned on to lower the indoor temperature.

Humidity control scheme: When the user sets the humidity threshold on the interface or cloud platform, the system receives the instruction to set the value. If the humidity sensor detects that the humidity is lower than the set value, the humidifier will be turned on to increase the indoor humidity. If the humidity sensor detects that the humidity is higher than the set value, the fan will be turned on to exhaust air and reduce the indoor humidity. Block diagram of hardware system design scheme is shown in Figure 2.1.

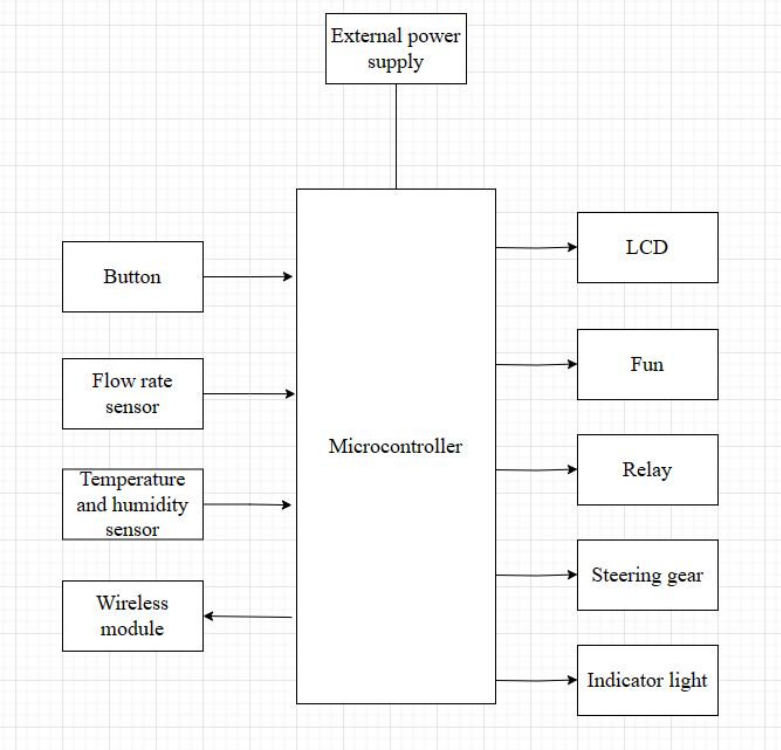


Figure 2.1 – Block diagram of hardware system design scheme

(1) Microcontroller: collects and analyzes data and controls system work;

(2) Temperature and humidity sensor: collects indoor environment temperature and humidity data;

(3) Liquid crystal display module: displays temperature, humidity, clock, etc.;

(4) Wireless module: Upload temperature and humidity data to the cloud platform for remote real-time monitoring of data;

(5) Fan: The fan works to discharge the humid air indoors and reduce the humidity;

(6) Flow rate sensor: detects the flow rate of liquid in the heater;

(7) Steering gear: controls the flow of liquid in the heater and increases the indoor temperature;

(8) Indicator light: Indicates the working status of servos, relays, etc;

(9) Button(key): used to set time, temperature threshold, and humidity threshold in the microcontroller control panel;

(10) Relay: controls the power of external equipment such as air conditioners.

2.2 Argument and comparison of plans

2.2.1 Microcontroller selection and demonstration

Option 1: 51 microcontroller has the advantages of low operating voltage, low power consumption, high operating frequency, low price, etc. It can be widely used in various low-power, high-precision, miniaturized electronic equipment [5]. Due to its larger size, lower computing speed, and fewer pins, the 51 microcontroller may gradually be replaced by more advanced microprocessors. 51 microcontroller is shown in Figure 2.2.

Option 2: STM32 microcontroller, which has the characteristics of powerful functions, numerous models, and wide use. It integrates many functions internally, such as analog signal conversion, serial peripheral interface, serial communication, etc. Due to its high main frequency, it has high-performance processing capabilities and low power consumption, which is very suitable for the needs of wireless product design [6]. STM32 microcontroller is shown in Figure 2.3.

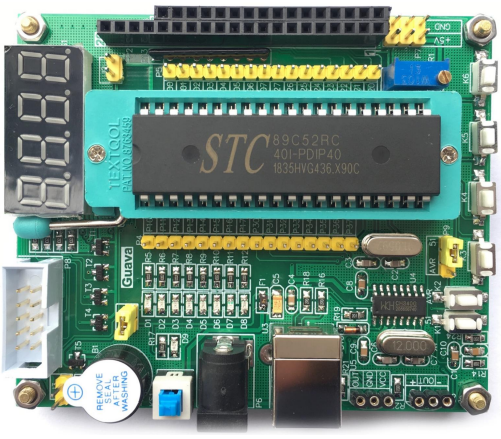


Figure 2.2 – microcontroller

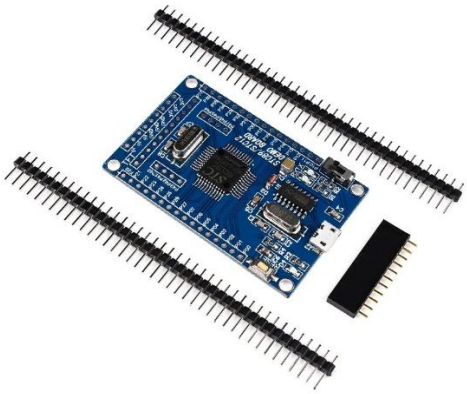


Figure 2.3 – STM32 microcontroller

Option 3: Arduino microcontroller, which uses the Atmel AVR series microcontroller, which is based on the Hanard structure RISC processor and has the characteristics of low power consumption, low cost, and easy programming. It uses standardized pins and interfaces (such as GPIO) and various expansion boards (Shields), making it more scalable to meet application needs. Typically used in small embedded systems. Arduino microcontroller is shown in Figure 2.4.

Option 4: Rasberry Pi development board, which is based on ARM chip and has more powerful processing performance and larger memory for similar products. It can run complex applications and multi-task, and has network connectivity and graphics capabilities. But essentially, it's a powerful single-board computer with full computing power and running a full operating system. The Rasberry Pi development board is shown in Figure 2.5.

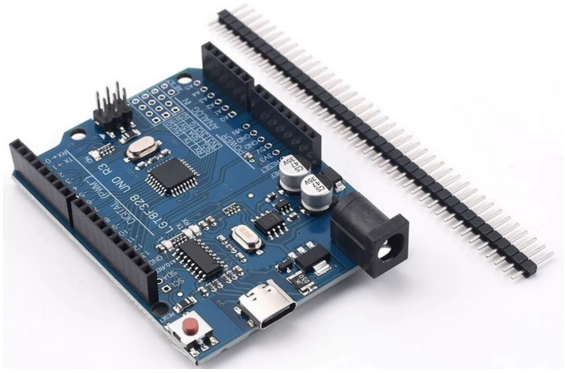


Figure 2.4 – Arduino microcontroller

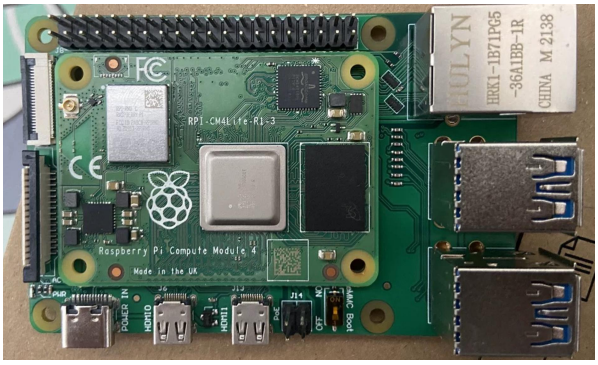


Figure 2.5 – Rasberry Pi development board

The theme of this design is: Home constant temperature and humidity system. The Home constant temperature and humidity system designed this time uses the STM32 microcontroller with higher integration and smaller size as the control device, which can meet the design requirements and adapt to future development trends.

Compared with option 1: Although the I/O pin of the 51 microcontroller is simple to use, it has no output capability when the level is high. When parsing double data pointers, it runs too slowly. Moreover, the self-protection ability of the 51 microcontroller is very poor and it is easy to burn out the chip. Even though the 51 microcontroller is one of the most widely used 8-bit microcontrollers, considering future scalability, the 32-bit core of STM32 is more in line with the mainstream of future development.

Compared with option three: Arduino usually uses a low-power microcontroller with weak processing capabilities and is suitable for processing simple control tasks and sensor data. It has lower clock speed and memory and cannot handle complex computing tasks. In terms of scalability, STM32 has richer supporting equipment and more flexible programming methods, which is more suitable for the production of this theme.

Compared with option 4: First of all, the Raspberry Pi is a complete single-board computer with strong processing capabilities and rich interfaces. Secondly, the Raspberry Pi supports a large number of expandable modules and sensors, and can be expanded through USB, GPIO and other interfaces. It can run Linux operating system and various application software. But behind its powerful performance is its high price. Its performance completely exceeds the performance required for this theme. In terms of economy, there is a lack.

To sum up, the Home constant temperature and humidity system designed this time uses the STM32 microcontroller with higher integration and smaller size as the control device, which can meet the design requirements and adapt to future development trends.

2.2.2 Wireless module selection and demonstration

Option 1: HC-05 Bluetooth module is a classic Bluetooth module, using the standard Bluetooth V2.0 protocol, working in the 2.4GHz frequency band, supporting serial communication and transparent transmission mode, and the price is relatively low. The maximum transmission rate is 2.1Mbps, using serial communication, which can be used to connect wireless products to communicate with each other. HC-05 Bluetooth module is shown in Figure 2.6.

Option 2: The ESP8266 wireless module is a low-cost wireless product. It integrates WIFI function and can be widely used in smart homes, Internet of Things products and other fields. It has a built-in processor and can run custom programs, making it suitable for development - IoT devices and embedded systems. Users can realize functions such as sensor data collection, remote control and data transmission through programming [7]. ESP8266 wireless module is shown in Figure 2.7.

This design is applied to indoor temperature and humidity monitoring in homes. The WIFI module can be connected to the router for relay to realize the networking function. Bluetooth is a near-field communication device with point-to-point connection, which requires additional transfer equipment to achieve remote control. Therefore, this design The WIFI module was selected for this design.

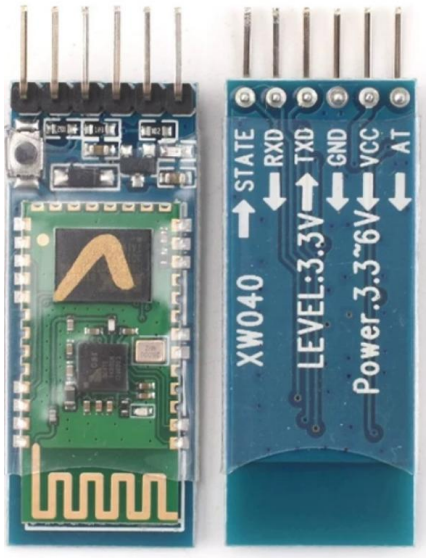


Figure 2.6 – HC-05 Bluetooth module

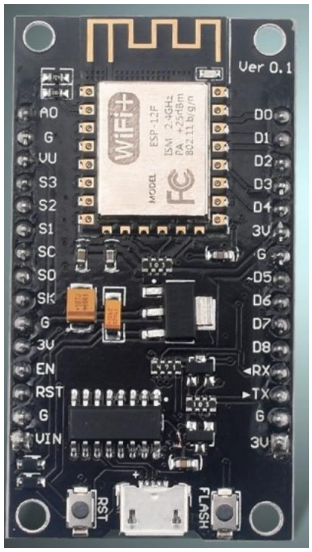


Figure 2.7 – ESP8266 wireless module

2.2.3 Selection and justification of display screen

Option 1: The digital tube is a simple character display device that displays numbers by lighting up different segment codes. It is often used for timing, counting and other occasions that require digital display, and has the advantage of strong reliability. The digital tube is shown in Figure 2.8.

Option 2: OLED LCD display is small in size, has many pixels, and has relatively rich display content. It has its own backlight and the display effect is clear and vivid. It can be used in industrial and medical fields. OLED LCD display is shown in Figure 2.9.

This design needs to display information such as the temperature, humidity of the indoor environment, and the set clock. The display content is relatively diverse, so option 2 is selected as the display screen for this design.

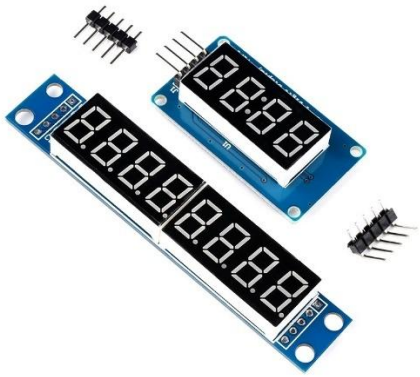


Figure 2.8 – The digital tube

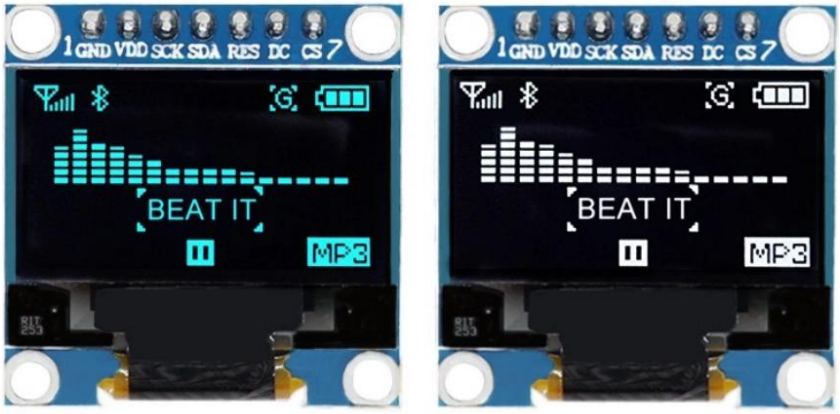


Figure 2.9 – OLED LCD display

2.2.4 Selection and demonstration of temperature and humidity modules

Option 1: PT100 thermal resistance sensor, which has a wide collection temperature range, fast response speed, small size, high accuracy, three-proof appearance and structure, and can be used in water tanks, bathtubs and other occasions. PT100 thermal resistance sensor is shown in Figure 2.10.

Option 2: DHT11 sensor module can collect temperature and humidity at the same time, and realize data transmission and control through a single signal line. It is simple and easy to use and can be widely used in the field of temperature and humidity monitoring, such as smart home, warehousing environment monitoring, etc. DHT11 sensor module is shown in Figure 2.11.

Its compact package design makes it easy to install in various devices, and it is stable and reliable. It can provide real-time and accurate temperature and humidity information, providing users with valuable environmental data support [8].

This design uses the DHT11 sensor module as the sensor to collect the temperature and humidity of the indoor environment.

Figure 2.10 – PT100 thermal resistance sensor

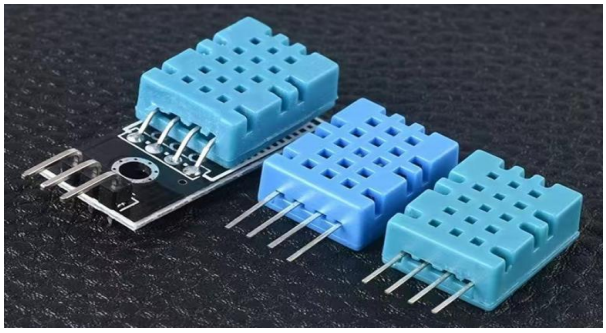


Figure 2.11 – DHT11 sensor module

Conclusion about 2nd CHAPTER

I analyzed the overall system design plan, as well as the temperature design plan and humidity design plan. Among the four mainstream microcomputers on the market, they have their own advantages. After weighing the functionality and economy of the microcomputer, I chose the STM32 as the microcomputer used in this design and configured it with various sensors.

As the times change, various types of components are constantly updated on the market. While their performance continues to improve, the prices are also very affordable. We not only follow the development of the times, we also enjoy the development of the times and enjoy the changes in life through technology. Make it easier for us to turn the ideas in our heads into reality.

STM32 has a 32-bit core, which brings higher computing power, more complete interfaces and accessories, and better economy, which is the core of this design. When designing, we need to have a full understanding and analysis of the design goals, analyze the advantages and disadvantages of each component from a debate perspective, and find the most suitable hardware. This is not only the first step to make excellent products, but also improves the product's economy and cost-effectiveness. The hardware used is shown in Figure 2.12.

Name	Model
Microcontroller	STM32F103
OLED	12864 OLED IIC
WIFI module	ESP8266 wireless module
Temperature and humidity sensor	DHT11
Flow rate sensor	YF-S201
Button（key）	C1-4
Steering gear	SG90
FUN	FUN-STM32
Relay	relay-stm32

Figure 2.12 – Use hardware

CHAPTER 3. DESIGN AND IMPLEMENTATION OF HARDWARE AND SOFTWARE

3.1 Design and implementation of hardware circuit structure

3.1.1 Introduction to the core board of microcontroller

This design uses STM32F103C8T6 microcontroller. The microcontroller is based on the ARM core and adopts Harvard structure, which is configured with a higher main frequency. The chip has 49 general-purpose I/O ports and a series of rich peripheral resources, including AD/DA modules, timers, communication interfaces, PWM output, etc., which can meet the needs of a variety of applications. The STM32 microcontroller system clock can reach up to 72 MB through configuration. It has an internal RTC clock function, an external 32.768KHz crystal oscillator, and can be powered by a CR1220 battery [10]. The core board of the microcontroller is shown in Figure 3.1.

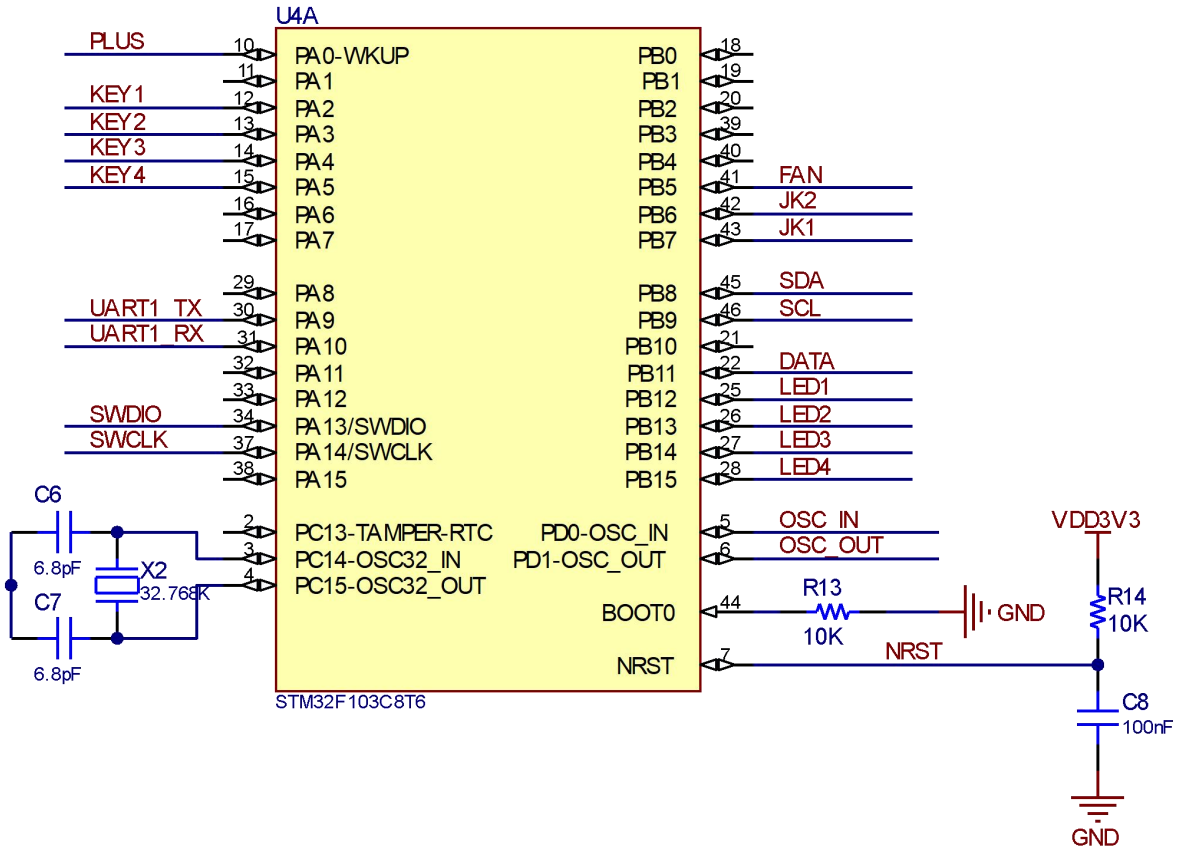


Figure 3.1 – Pin diagram of the microcontroller core board

3.1.2 Partial circuit design of OLED liquid crystal display module

This design uses a monochrome LCD display with 128*64 LCD display points. It is manufactured using organic light-emitting diode technology and has the characteristics of high contrast, fast response, wide viewing angle and low power consumption. It uses I2C or SPI interface for serial communication. , can be easily connected to various main control devices [11]. This design uses the IIC method for connection, which has the characteristics of fewer pins and fast refresh speed. The OLED screen is externally powered by 5V. The hardware interface circuit diagram is shown in 3.2.

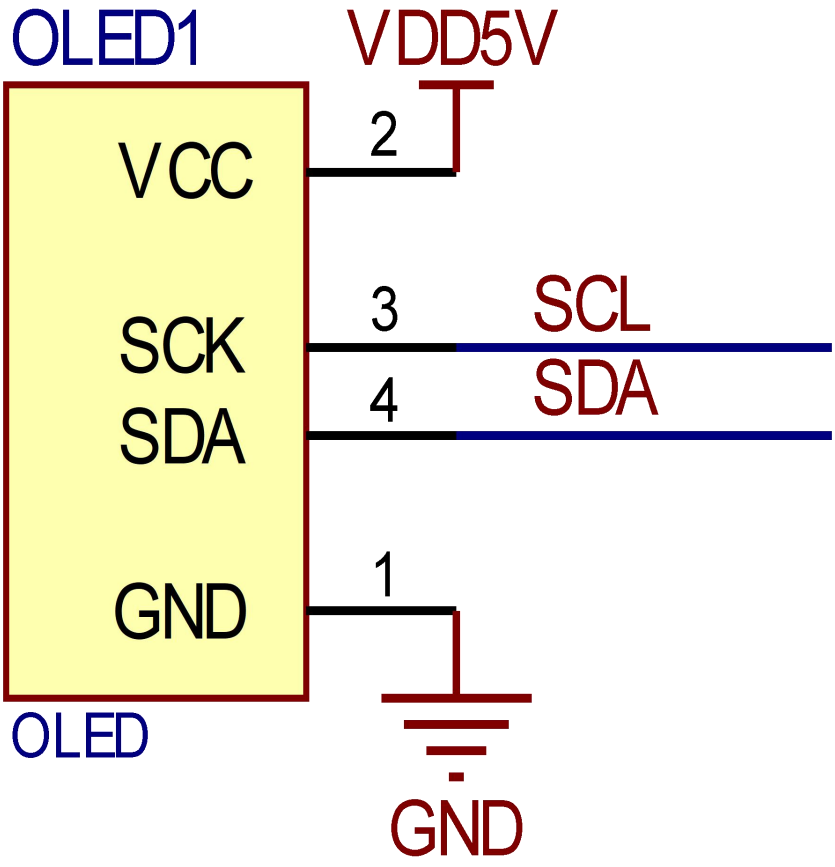


Figure 3.2 – LCD screen interface circuit

3.1.3 Partial circuit design of OLED liquid crystal display module

DHT11 sensor module has two built-in sensitive elements, which can collect temperature and humidity at the same time. It uses a single wire for communication. It can be used to send and receive data. It can obtain secondary data of the integer and decimal parts of temperature and humidity, followed by the smallest the working voltage' can reach 3V[12]. The hardware circuit interface is shown in Figure 3.3.

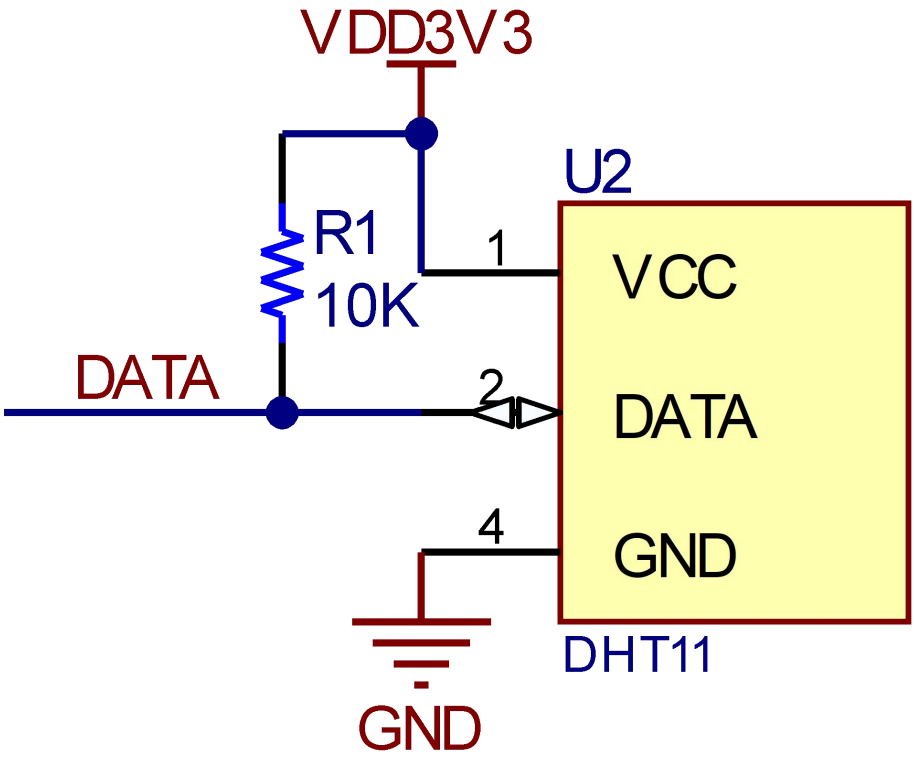


Figure 3.3 – DHT11 temperature and humidity sensor pin connection diagram

3.1.4 WiFi module circuit interface design

This design uses the ESP8266 wireless module, which can transmit data and communicate with other devices through wireless connections. Its working principle includes integrating wireless modules and processors inside the hardware, firmware providing wireless communication protocol implementation, communicating with other devices through wireless connections, and using TCP/IP protocol stack for data transmission [13[14]]. The hardware circuit connection between the microcontroller and the wireless module is shown in Figure 3.4.

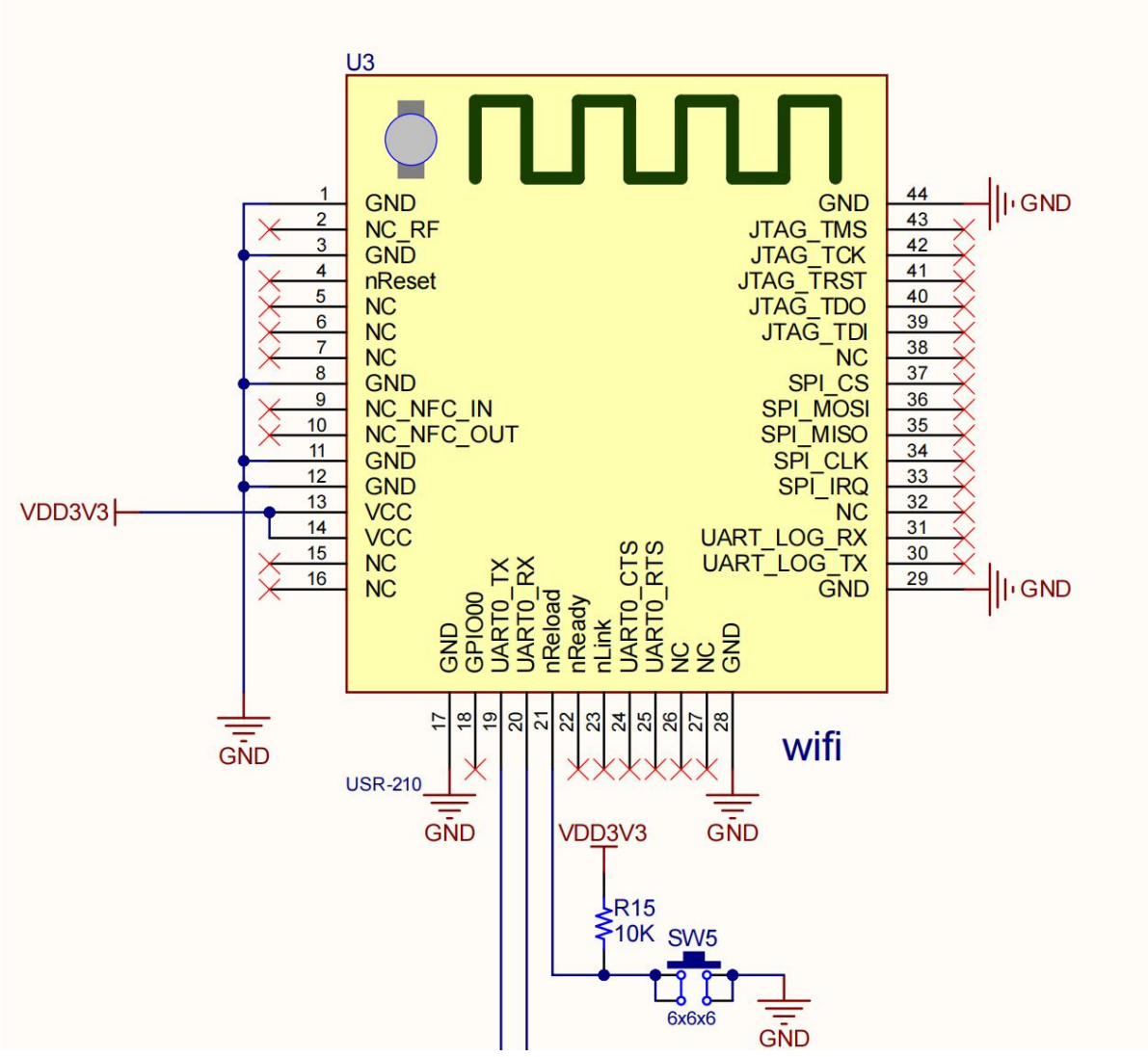


Figure 3.4 – ESP8266 hardware connection circuit diagram

3.1.5 Flow velocity sensor interface design

This design uses the S201 flow rate sensor, which contains a Hall element inside. It is designed based on the principle of magnetic field and current conversion. When the external magnetic field changes, the voltage changes. The internal amplifier circuit contains an amplification circuit to amplify the changing voltage signal. Contains a comparator output pulse signal internally. The flow rate sensor interface circuit is shown in 3.5.

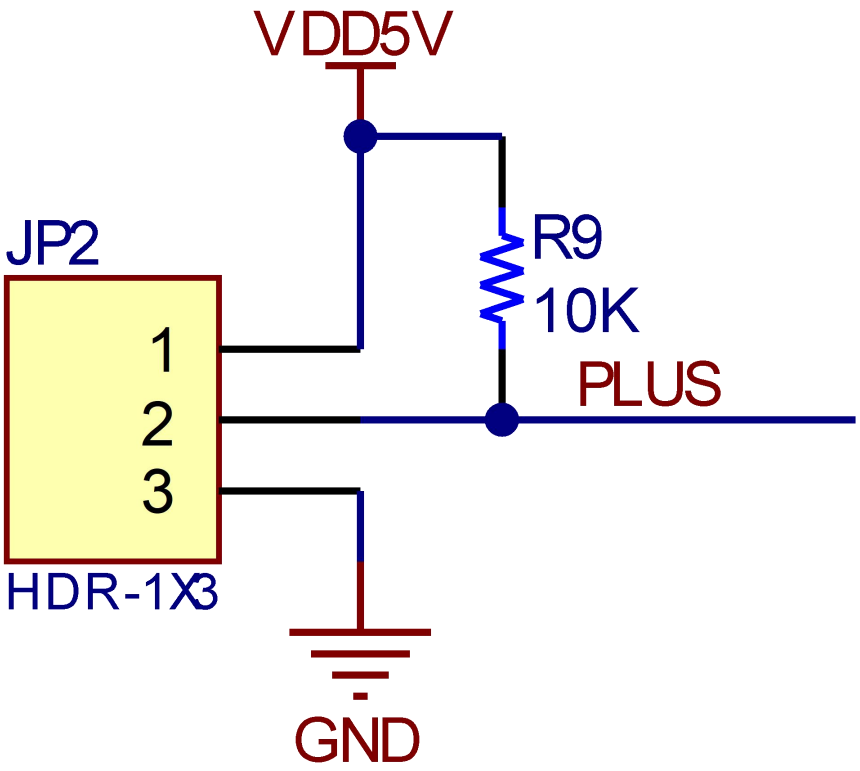


Figure 3.5 – Flow velocity sensor circuit schematic diagram

3.1.6 Relay control circuit design

The increase in humidity and cooling of the indoor environment is controlled by two relays as switches. The relay works based on the principle of magnetic induction. When the base level of the triode is pulled low, the triode is turned on, and the coil generates a magnetic field to attract the shrapnel. Make the relay conductive. At this time, the humidifier or air conditioner is powered on and starts working. The hardware circuit design is shown in Figure 3.6.

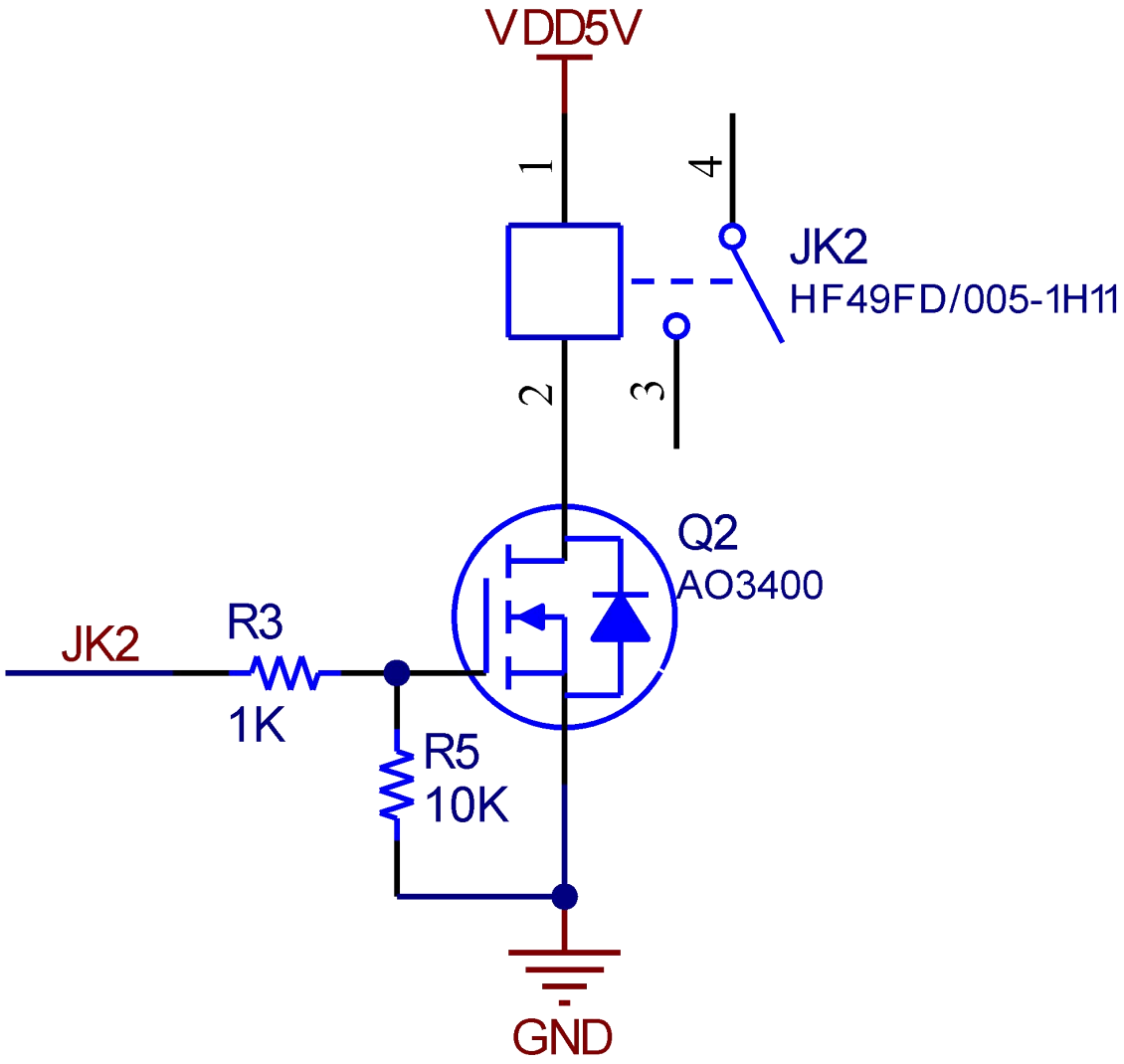


Figure 3.6 – Relay control principle diagram

3.1.7 Fan control circuit design

In this design, the fan acts as a dehumidifier. When the humidity threshold is reached, the exhaust fan is started. The fan is powered by 5V and controlled by a triode as a switch. When the control signal is high level, the transistor is turned on, thereby driving the fan to start working. The hardware circuit is shown in Figure 3.7.

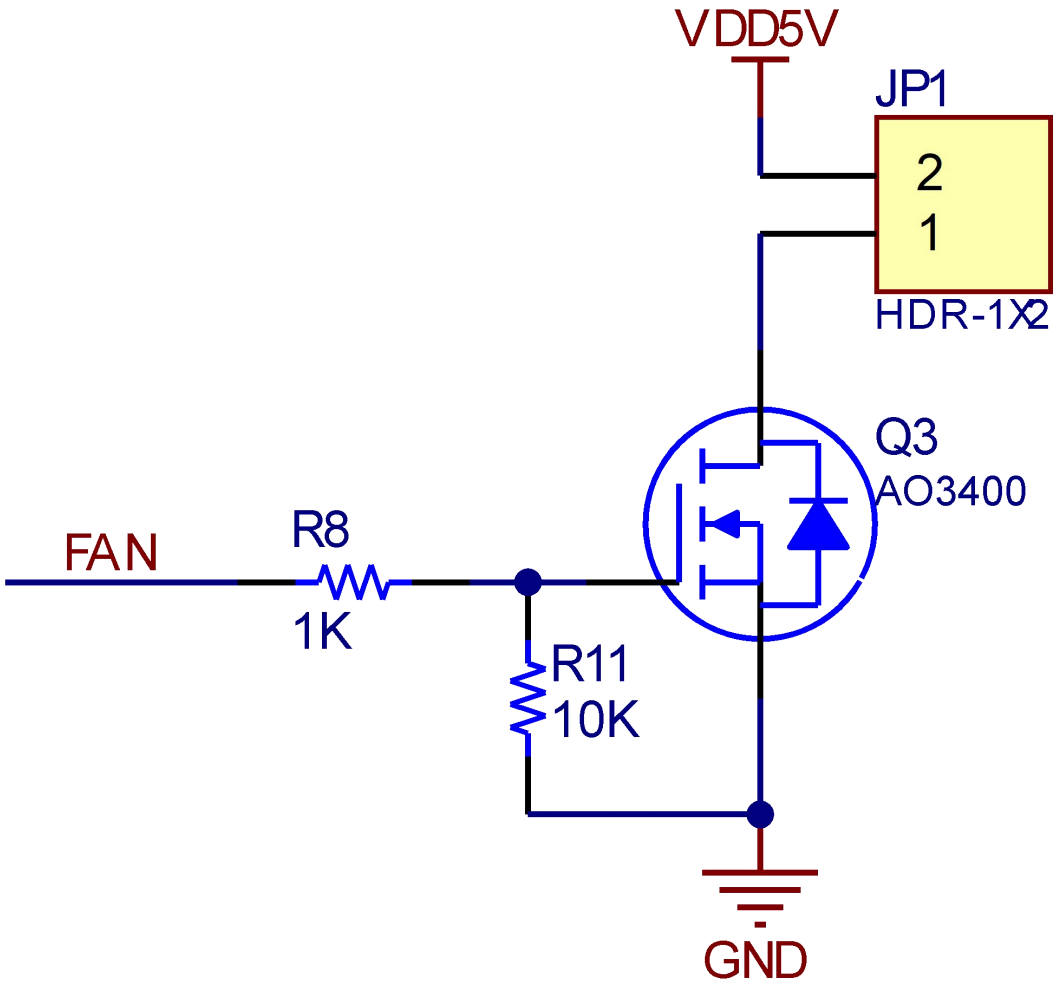


Figure 3.7 – Fan control circuit schematic diagram

3.1.8 Key circuit design

The key circuit design uses four separate key controls. The key detection port is connected to the I/O pin of the microcontroller. The microcontroller detects the low level state. In this design, it is used as addition, subtraction, threshold setting and clock setting functions. As shown in Figure 3.8.

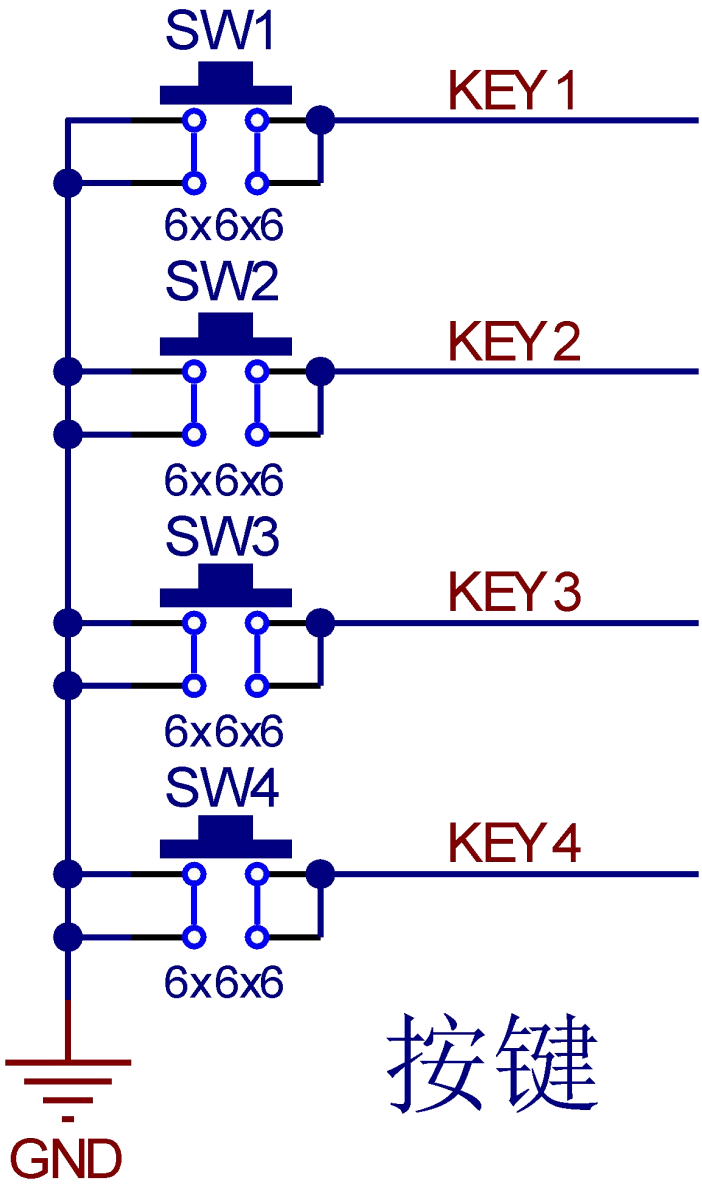


Figure 3.8 – Circuit diagram of the button part

3.1.9 Steering gear control circuit design

The SG90 servo has only three leads, the power line, the ground line and the control signal line. The control signal line is a pulse signal with a frequency of 50Hz. The angle of the rotating shaft can be adjusted by changing the pulse width [15]. The hardware interface is shown in Figure 3.9.

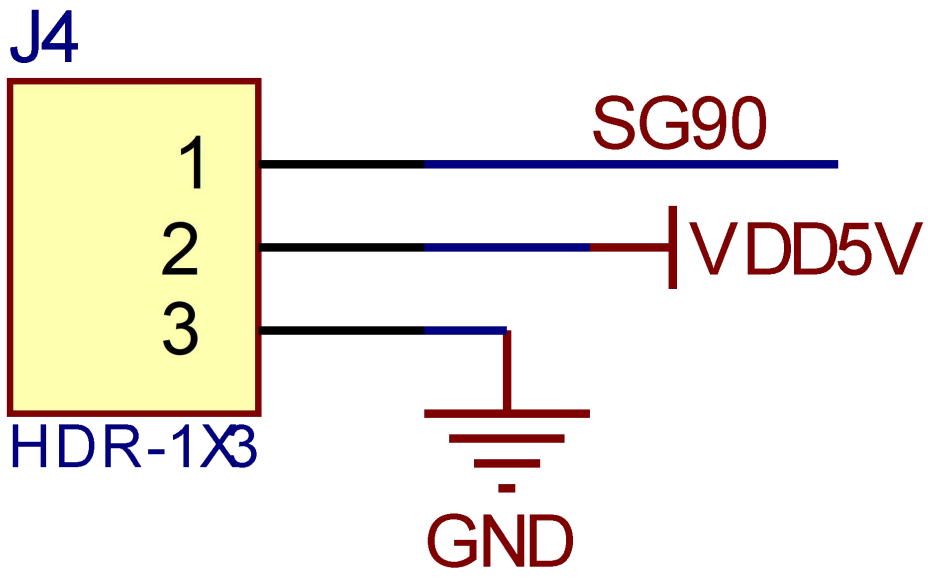


Figure 3.9 – Servo hardware interface

3.2 Software system design and implementation

3.2.1 Software system program flow char

The microcontroller program part of this design is carried out using the KEIL software compilation environment, and the software is configured according to the functions of the DHT11 temperature and humidity module, WIFI module and other peripherals. After the system is powered on, the main program starts to run, first initializing the system clock and peripheral ports. Use a microcontroller to control the temperature and humidity sensor and flow rate sensor, obtain the current data, and display it in real time through the OLED display. At the same time, the obtained data is sent to the WIFI module through the serial port and uploaded to the cloud. Users can obtain the data through the mobile APP. The main program flow chart is shown in Figure 3.10.

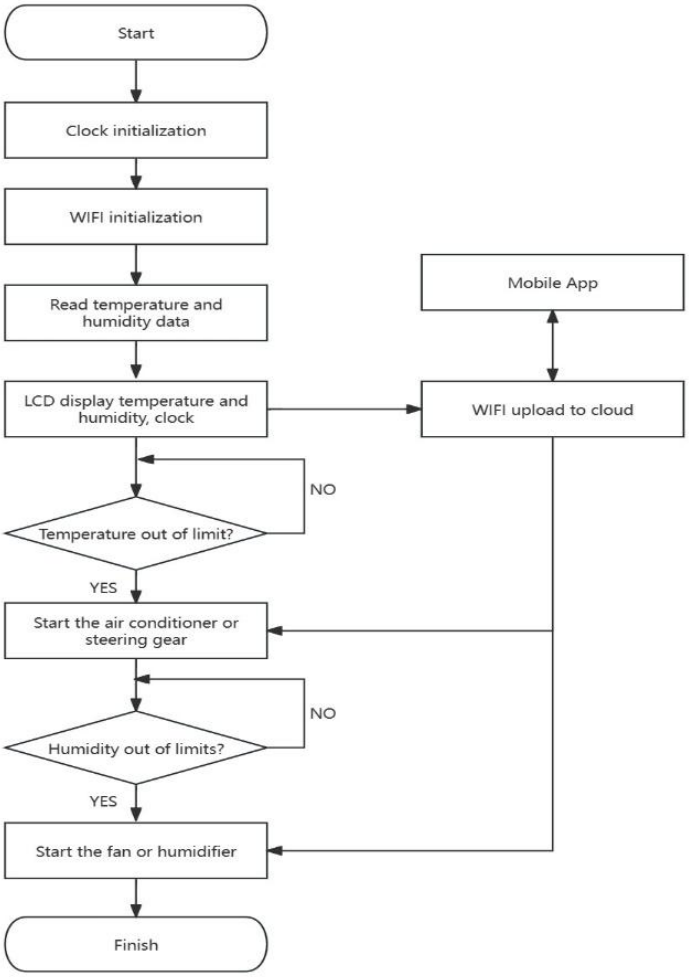


Figure 3.10 – main program flow chart

3.2.2 OLED LCD screen programming

Based on the hardware design OLED LCD screen pin definition and the timing diagram provided in the specification book, the corresponding code was written. In the process of writing code, two commonly used sub-functions are mainly used, including the write command function and the write data function. The LCD screen display process must first initialize the SSD1306 driver controller and set its working parameters and display mode. Next, turn on the display function of the LCD screen and prepare to receive and process image data. After clearing the video memory, the data to be displayed is written into the video memory and sent to the controller. The controller displays the image on the LCD screen based on the data. The OLED LCD screen control process is shown in Figure 3.11.

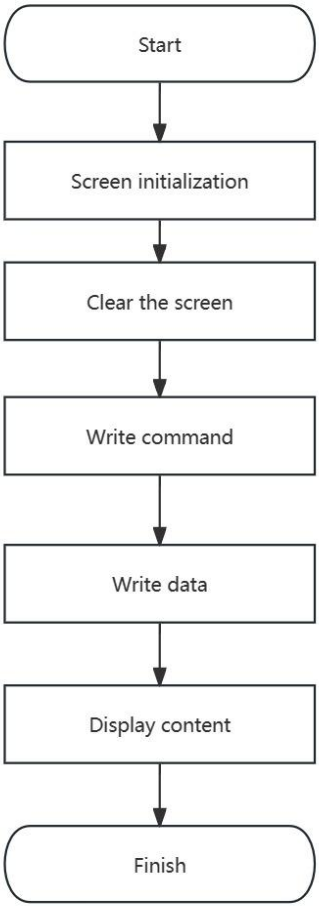


Figure 3.11 – OLED LCD screen control process

3.2.3 Programming of ESP8266 module

This design controls the ESP8266 wireless module through the microcontroller serial port. According to the module's data manual, AT instructions are used to configure the server port to upload the data to the server. The user can view the data in real time through the mobile phone App. The data sent by the microcontroller is sent according to a certain protocol format. The ESP8266 module sends the data to the human cloud platform in transparent transmission mode, and the mobile APP receives the information and displays it. The specific flow chart design is shown in Figure 3.12.

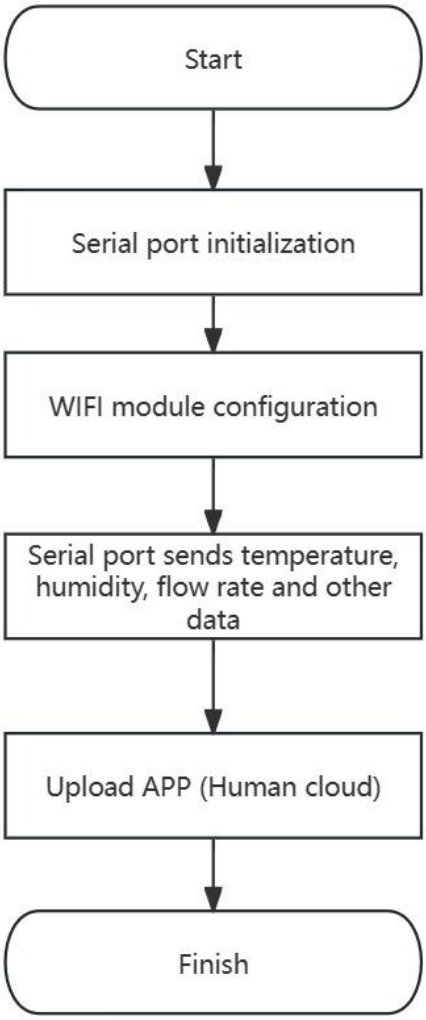


Figure 3.12 – specific flow chart design

3.2.4 Temperature and humidity detection program design

The DHT11 temperature and humidity sensor is a single-bus design. When reading the DHT11, you can directly read the entire package of data from the temperature sensor by writing a program, and process the data at the same time to achieve more accurate measurement results. The detection flow chart is shown in Figure 3.13.

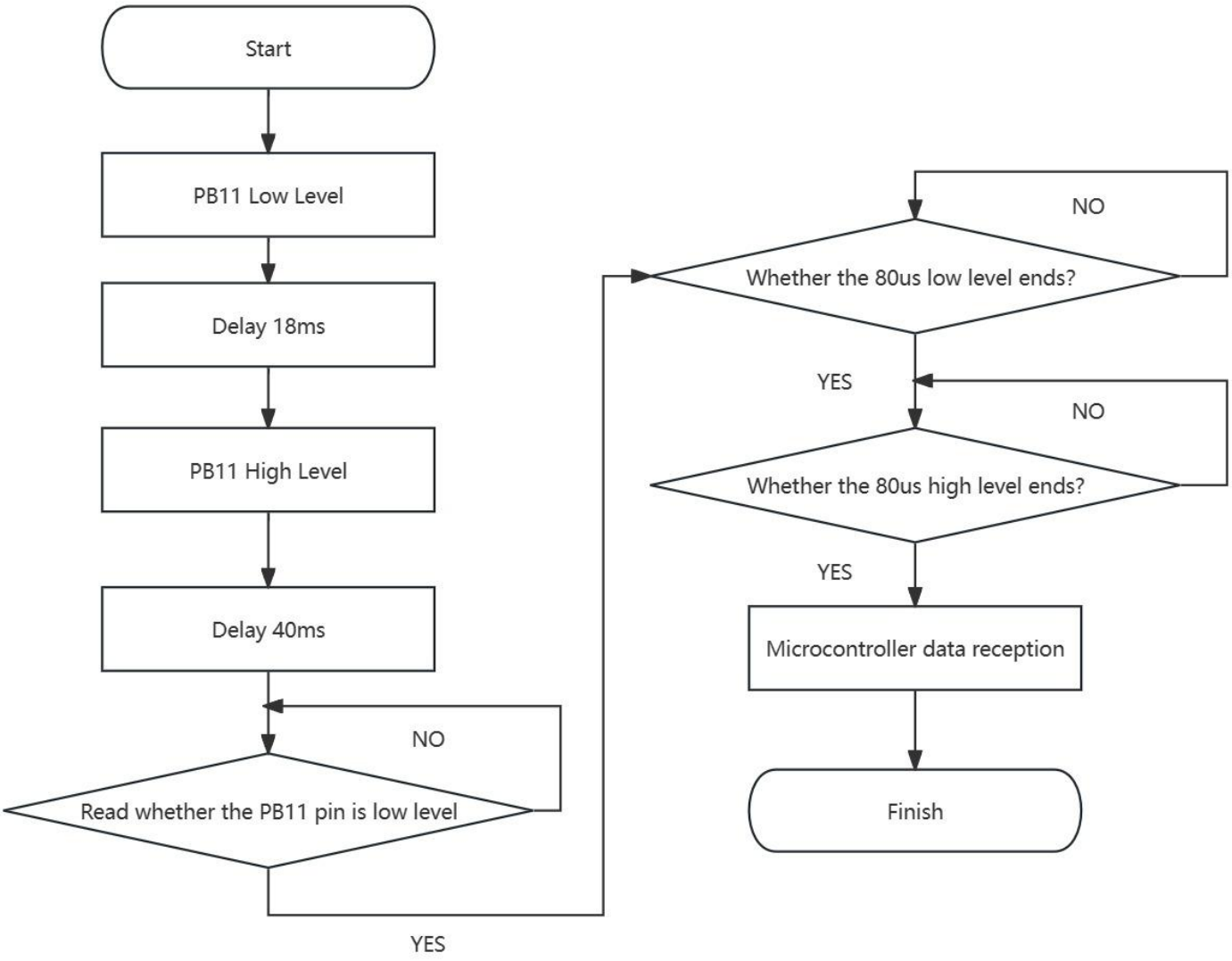


Figure 3.13 – detection flow chart

3.2.5 Steering gear control program design

Since this time to control the opening and closing of the heating water pipe liquid, the input signal pulse width period is 20ms, which can be controlled by configuring PWM. It only needs to configure the two states of the switch. The program flow is shown in Figure 3.14.

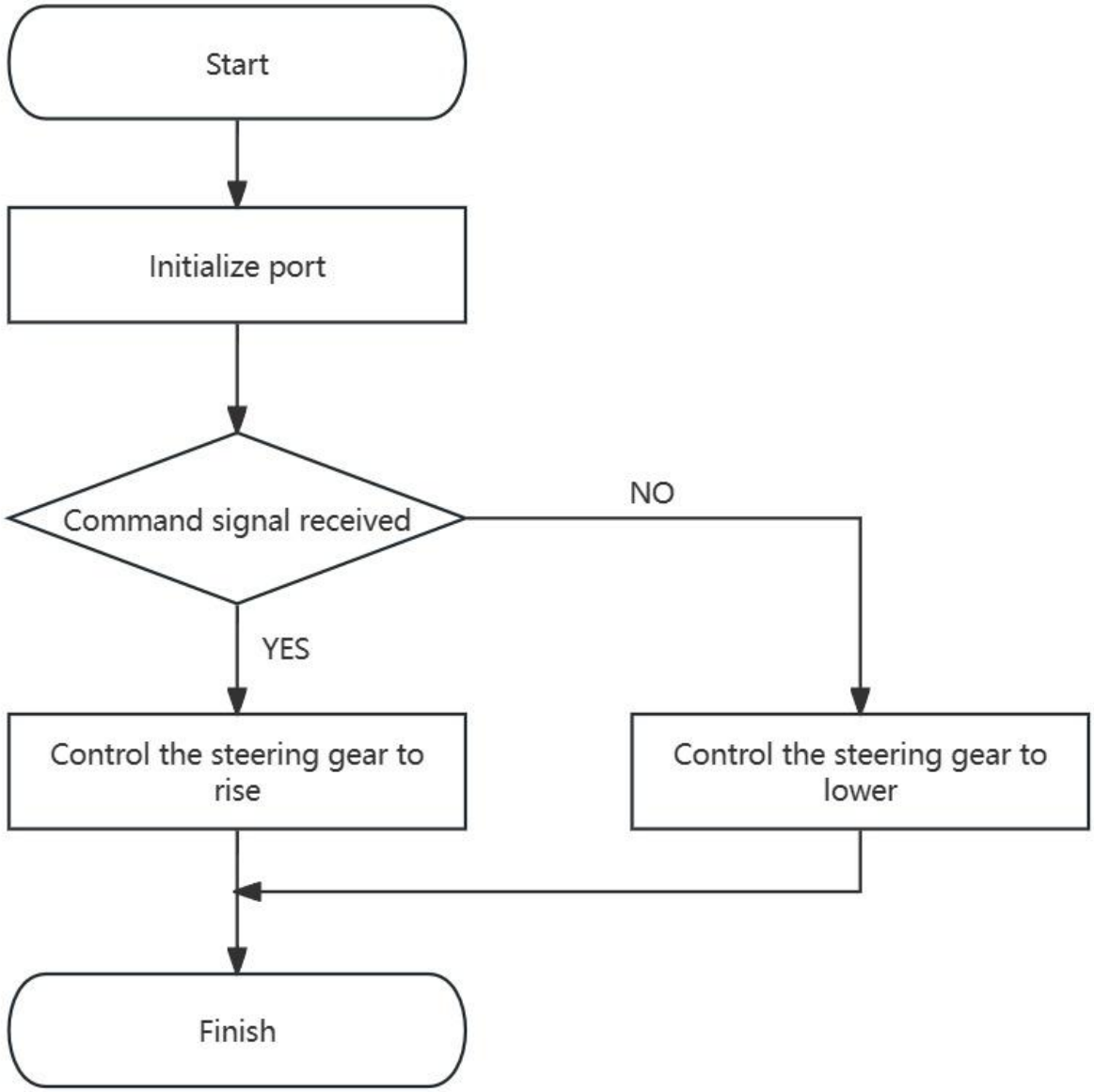


Figure 3.14 – program flow

3.2.6 Programming of button module

The button module uses four physical buttons. The button module selects falling edge trigger. When the button is pressed and the microcontroller detects that the I/O level is low, it starts button processing. The key triggering process has a steady-state process. Delayed sampling is used in the program for key detection to avoid false triggering of the key. This design contains four keys: plus and minus keys, temperature and humidity threshold setting and time setting keys. Through button adjustment, the flow chart of button recognition is shown in Figure 3.15.

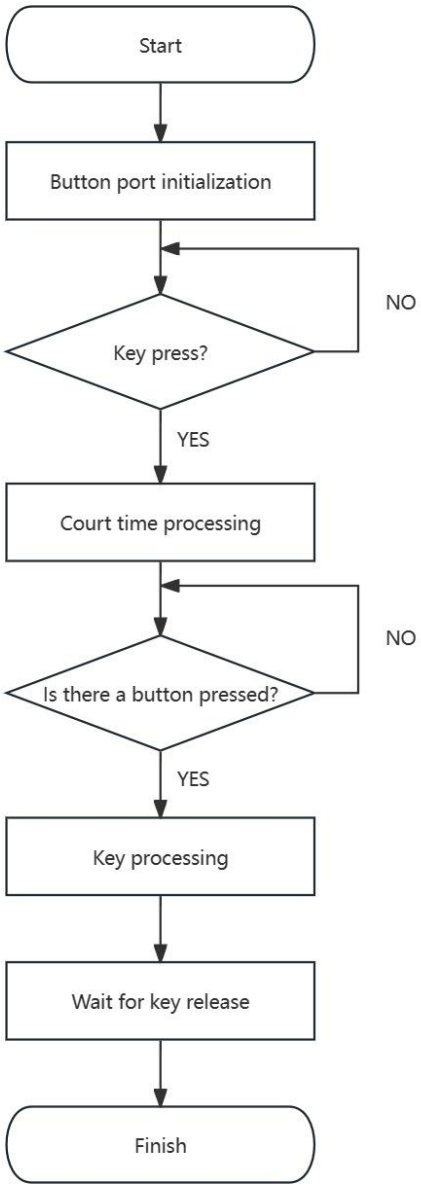


Figure 3.15 – flow chart of button recognition

3.2.7 Programming of status indicator light

Two sets of two-color indicator lights are used to determine humidity and temperature. The green light indicates operation within the threshold, and the red light indicates exceeding the limit. The indicator light control flow chart is basically as shown in Figure 3.16

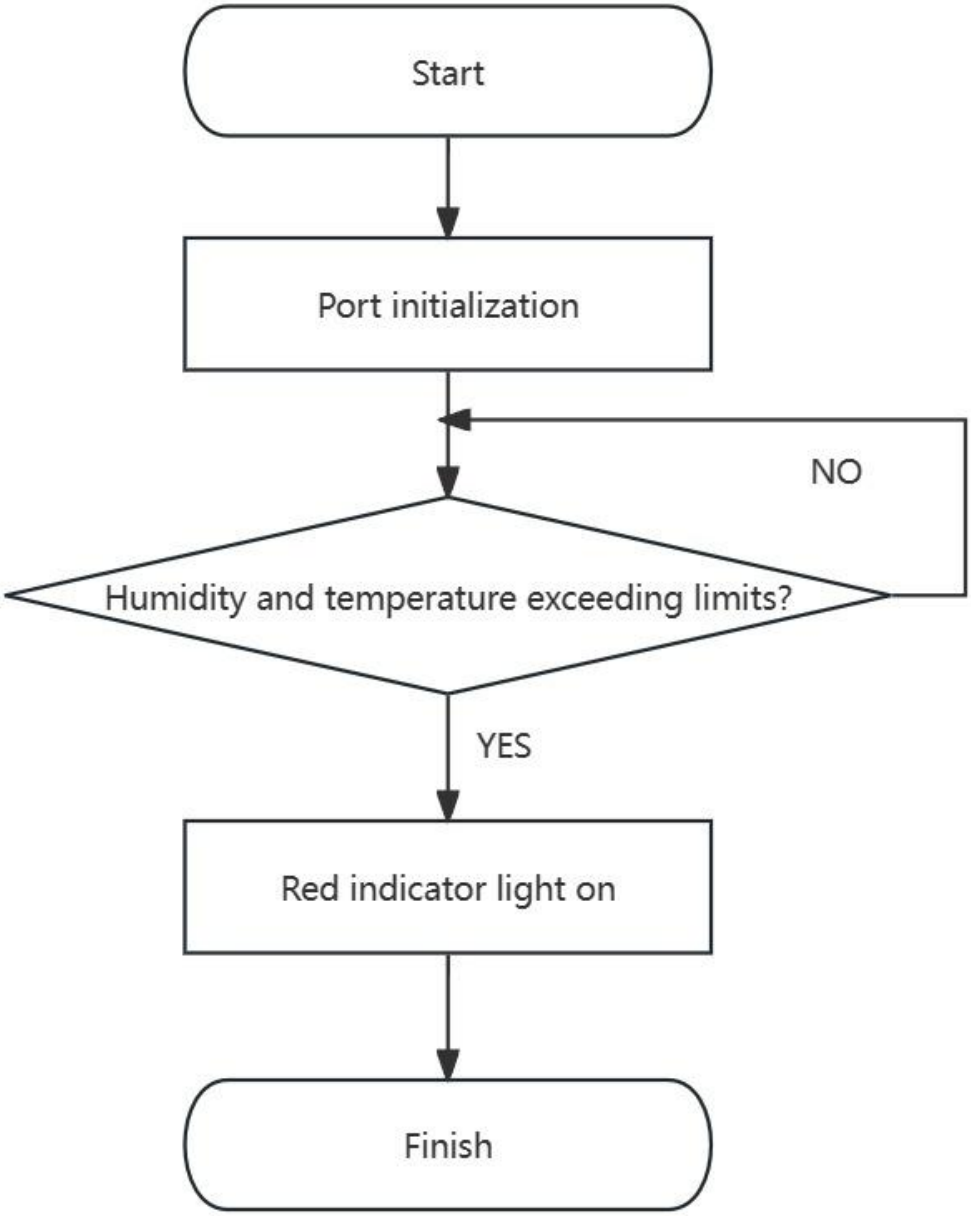


Figure 3.16 – indicator light control flow chart

Conclusion about 3rd CHAPTER

In this chapter, I designed the circuit interface for the core board of microcontroller, OLED liquid crystal display module, DHT11 temperature and humidity module, etc. Hardware circuit structure design is an essential part of completing the entire project. We have clarified the usage specifications of the components used to avoid possible functional deviations in the later stage.

Circuit design is a very important part of the whole thing. I considered the reasonable connection of all components and planned a reasonable signal path. Make the circuit have sufficient reliability through reasonable layout.

I used Altium Designer software to draw the circuit schematic diagram of each component to ensure that all components are connected correctly and reasonably. And consider signal integrity, anti-interference, heat dissipation and other issues. Timely correction of these problems can greatly shorten the time and cost of later debugging.

In general, I carefully thought about every link and verified it one by one. I believe that only through rigorous design and extensive testing can we obtain excellent, reliable, and high-quality works.The overall circuit connection diagram is shown in Figure 3.17.

In the complete software development life cycle, software system design and implementation is a very important part. Its process includes a series of processes from requirements analysis to actual coding. The core requirements of the task can be converted into specific software structures and codes to achieve our goals.

In this chapter, I drew the software system program flow chart in detail, and drew the design flow chart of each component according to the working principle, so that we can understand the system structure and functions more intuitively. Through detailed design of each component, the implementation path of each module is clarified, thereby reducing uncertainty and rework risks when writing programs later. This process plays a vital role throughout the project, ensuring that the overall functionality is as expected.

All in all, software system design and implementation is a systematic and detailed process. Only through rigorous analysis and design can we lay the foundation for development and meet the ease of maintenance and scalability of the entire project.

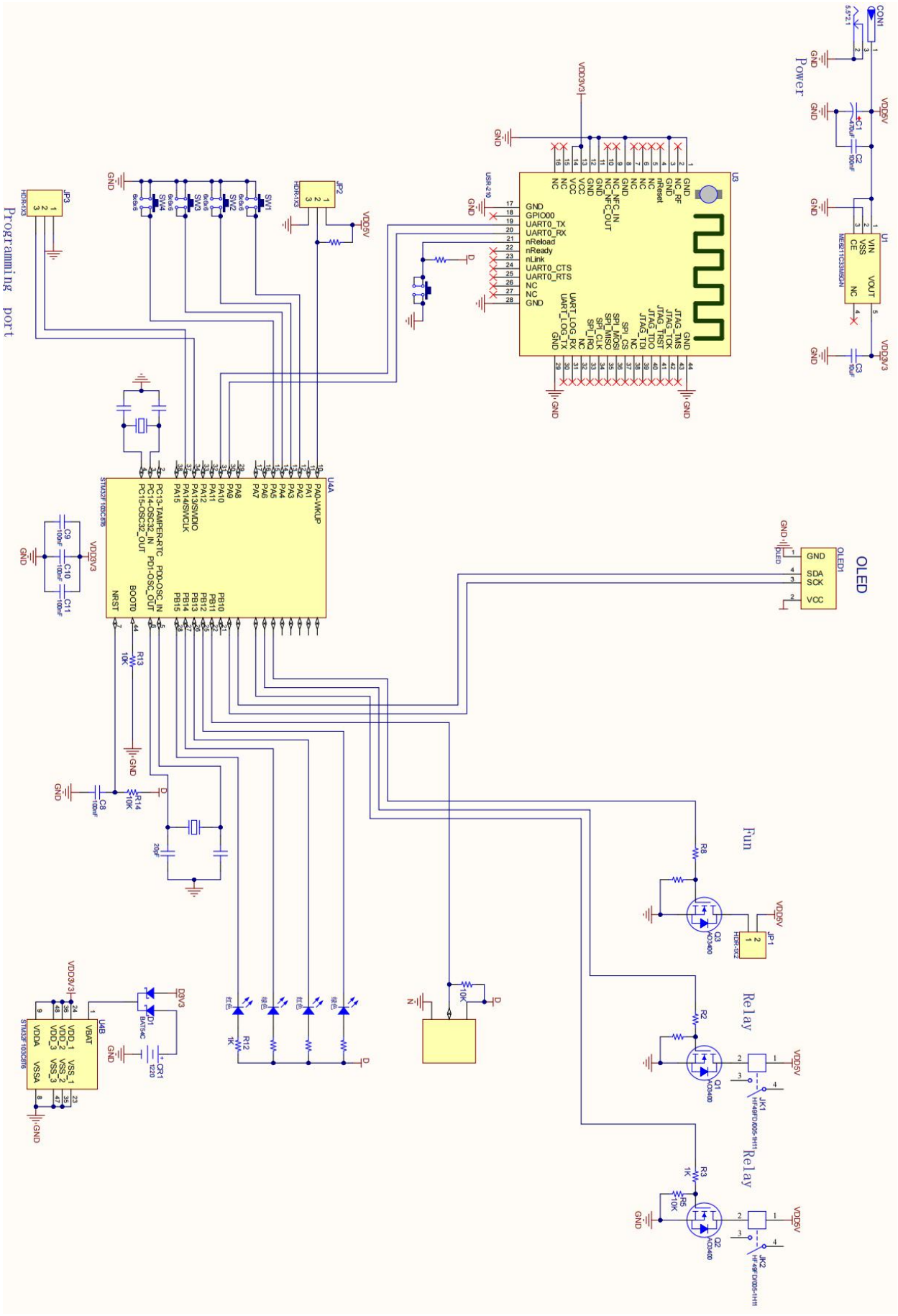


Figure 3.17 – The overall circuit connection diagram

CHAPTER 4. SYSTEM CODE DESIGN AND DEBUGGING

4.1 System debugging

4.1.1 Software and hardware debugging

This design uses circuit perforated boards, and each module is reasonably laid out according to the schematic diagram, and then welded. After completing the production of the experimental board, first use a multimeter to detect short circuits on the capacitors in the circuit to ensure that there is no short circuit. Refer to the schematic and measure the voltage output at each key point to ensure it is as expected. If it matches the design value, it means that the debugging work of the hardware circuit has been completed. When writing a microcontroller program, debug external modules such as DHT11 temperature and humidity reading data, ESP8266 wireless module serial port data, and check whether the current register status is correct.

4.1.2 Display content debuggin

First, debug the interface and button functions. The text will be displayed. According to the row and column scanning method, the fonts will be taken out and displayed first. After repeated verification, the LCD screen display content is correct, and the LCD screen will display the temperature and humidity values on the main interface. , date and clock, etc. The main interface display of the OLED LCD screen is shown in Figure 4.1.

Figure 4.1 – The main interface display of the OLED LCD screen

4.1.3 Temperature and humidity sensor debugging

Place the DHT11 temperature and humidity sensor indoors and compare it with the instrument to verify whether the data currently received is correct. If it is observed that the displayed value is close to the value in the meter, it means that the temperature and humidity data packet has been successfully and correctly parsed. At this time, the sensor measurement accuracy must be taken into consideration. Warm and humidify the area around the sensor. At this time, the value should change significantly. Temperature and humidity test is shown in Figure 4.2.

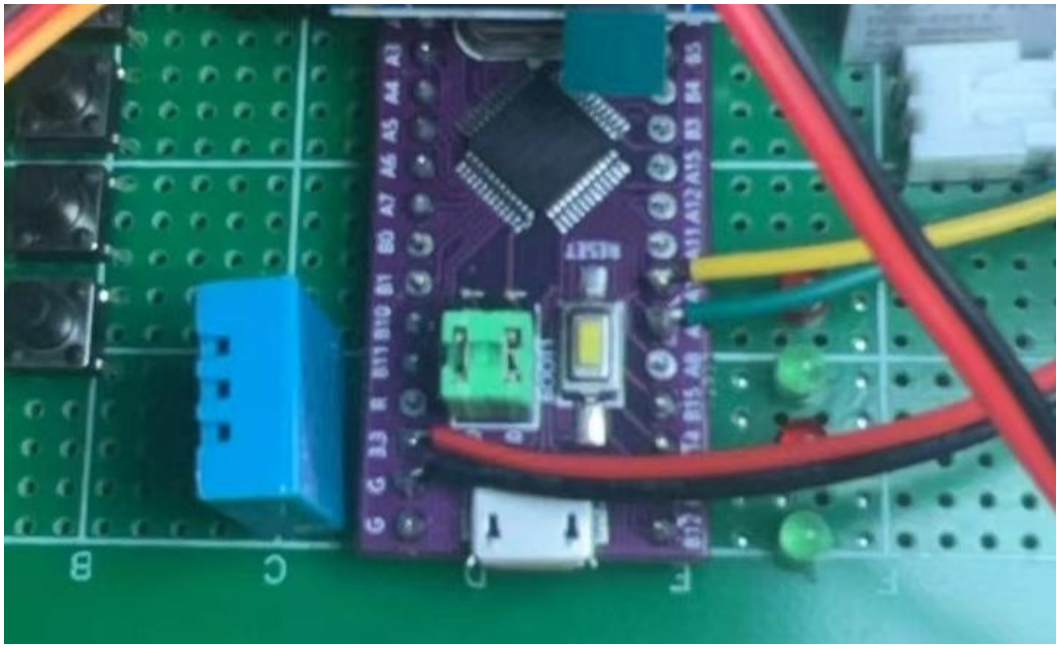


Figure 4.2 – Temperature and humidity test

4.1.4 WIFI wireless module debugging

Use the serial communication module to connect the ESP8266 module to the computer. According to the data manual, enter the AT command through the serial port tool and verify whether the reply content is consistent with expectations. Open the network debugging assistant, set the protocol to TCP server, then enter the server's IP address and port number, and click Create Server. Initiate a TCP connection for data reception and transmission. The data is first sent from the server to the ESP8266 module, and then transmitted through the serial port to the serial port debugging software for reception and display. The network debugging assistant and serial port tool interface are shown in Figure 4.3.

Figure 4.3 – The network debugging assistant and serial port tool interface

4.1.5 USR cloud platform introduction

USR cloud platform is an IoT technology company in Jinan, China. It focuses on the research and development, production and sales of communication products and is a leading industrial IoT software and hardware solution service provider in the industry. It has its own factory and has the ability to provide services for the entire industry chain from chips to the cloud. It is committed to providing customers with communication terminals, IoT modules, cloud platform services and IoT solutions.

For individual users, you can use the USR cloud platform, confirm the hardware and software equipment, add the equipment to the cloud, and then build IoT applications, allowing users to highly customize functions to achieve user goals. And it is adaptable to multiple devices, using WeChat applets or web pages to facilitate viewing of various data and operation control. USR cloud platform is shown in Figure 4.4.

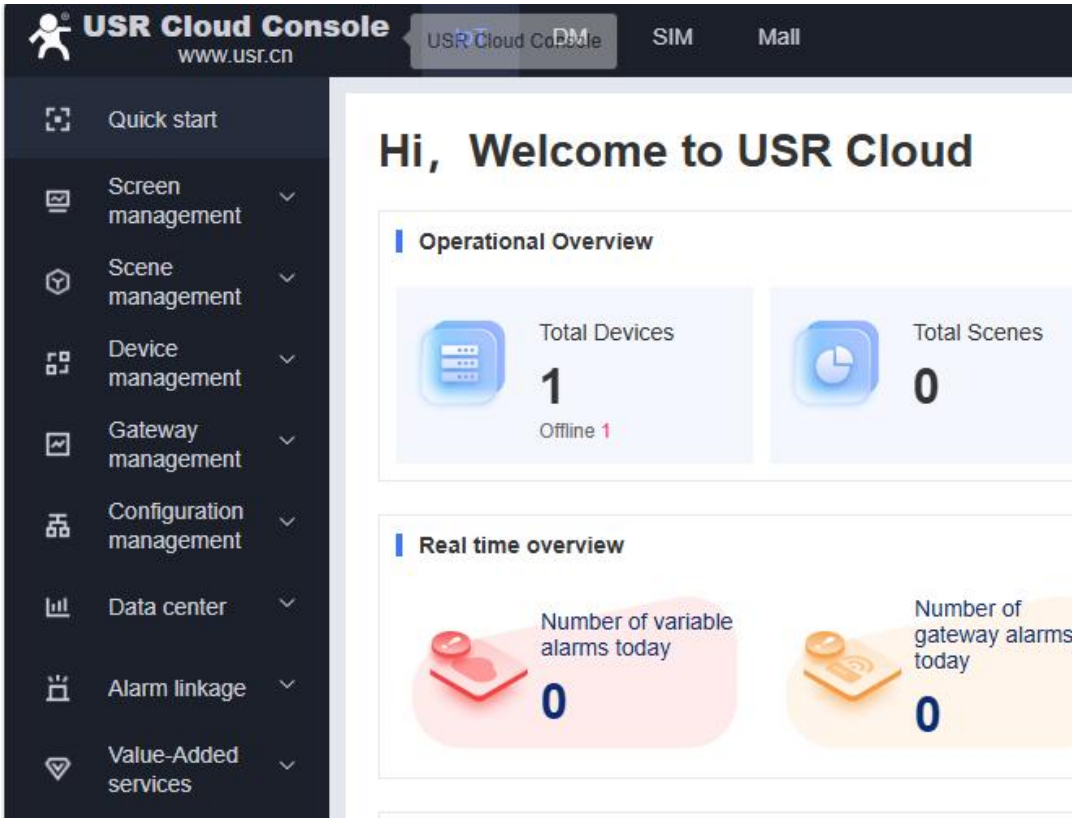
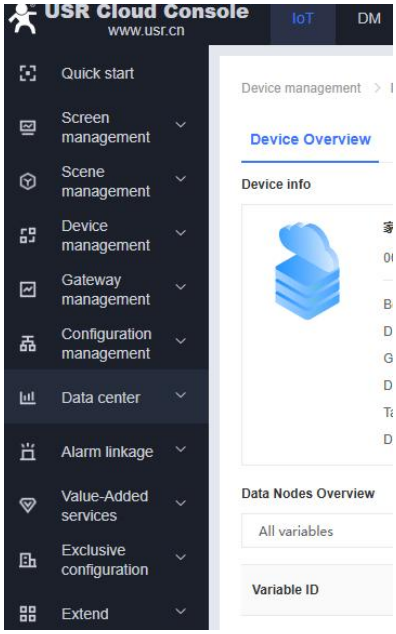


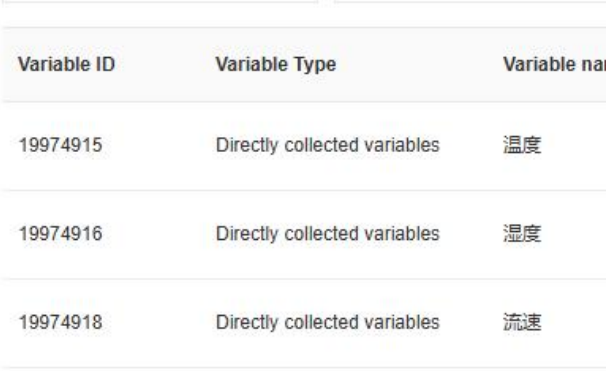
Figure 4.4 – USR cloud platform

4.1.6 USR cloud console introduction

The USR cloud console has a wide range of functions. It can monitor the online status of each device in real time, form an overview architecture diagram, manage device usage scenarios, and manage the device's gateway and configuration. It also includes alarm linkage and other functions. The USR console is shown in Figure 4.6.

This design configures various variables on the USR cloud platform, and uses the platform function to directly sample data such as temperature, humidity, and flow rate. In addition, the device can be controlled online through the console, and each component can be turned on separately, and the temperature and humidity thresholds can also be changed online. The import of various variables is shown in Figure 4.5.





Variable ID	Variable Type	Variable na
19974915	Directly collected variables	温度
19974916	Directly collected variables	湿度
19974918	Directly collected variables	流速

Figure 4.5 – The USR console Figure 4.6 – The import of various variables

4.1.7 USR cloud platform debugging

Enter the USR cloud development platform free cloud development APP, set up the interface to configure the temperature, humidity and other environmental variables, and input the variable byte information according to the serial port command. After the APP is completed, you can choose to use the automatically generated domain name or bind your own domain name. The USR cloud platform APP interface is shown in Figure 4.7.

Figure 4.7 – The USR cloud platform APP interface

Conclusion about 4th CHAPTER

In the design of the Home constant temperature and humidity system, I comprehensively considered multiple factors such as performance, power consumption, feasibility, reliability, etc., and rationally designed the system architecture and effective interrupt processing to make the system efficient and stable. By using the powerful processing performance of STM32, the design requirements were met, and the control from simple sensors to replicating IoT devices was completed.

I defined the functions of the four controllable physical buttons, which can control the time and change the thresholds of temperature and humidity. The microcontroller receives the sensor signal, determines whether the threshold range is reached, and gives corresponding feedback. Use the Wi-Fi module to upload data to the USR cloud platform, or receive control instructions from the USR cloud platform.

In order to ensure that the functions and performance of the Home constant temperature and humidity system meet expectations after development, I performed hardware debugging, software debugging, and functional verification. This directly affects reliability, stability, and user experience.

In order to ensure that the product meets design requirements and quality standards, I tested the connection reliability of each interface, ensured that each module can work stably, and ensured the correctness of data transmission and function calls. And tested the response time to ensure timeliness. I also tested the environmental test of the product to ensure its stable operation at high and low temperatures, high humidity, and low humidity.

Passing the above system debugging and testing can effectively guarantee the quality and reliability of the developed products and provide users with high-performance and reliable products.

FULL TEXT SUMMARY

The design of a home thermostat system based on the STM32 microcontroller is an integrated and intelligent solution. It combines peripherals such as temperature sensors, relay modules and display modules to build a home thermostat system that can monitor and adjust indoor temperature in real time. In the home thermostat system, the microcontroller can collect temperature and humidity sensor data in real time, and perform preliminary processing and analysis to control the hot water pipes, air conditioners, and humidifier switches. In terms of remote monitoring and control, remote mobile phone APP monitoring is achieved through the WIFI module and cloud platform, so that the indoor environment can be remotely monitored and intelligently managed.

There are many imperfections in this design, such as how to better control the relationship between the flow control valve angle and temperature. In addition, the stability and security of the wireless network are also an important part of the design that cannot be ignored. Effective measures must be taken to ensure the real-time transmission of data and the safe operation of the system. In order to solve these problems, it is necessary to continue in-depth research, optimize the design scheme, and achieve better results in practical applications.

REFERENCES

1. Zhu Mingshuo. Development trend and application analysis of smart home system [J]. Intelligent Building Electrical Technology, 2024, 18(01): 143-147.
2. Peng Yong, Tao Zengjie, Lin Zhen, et al. Design of smart home system based on STM32 and OneNET [J]. Internet of Things Technology, 2024, 14(02): 86-89.
3. Guan Jiacheng, Li Xiaofeng, Huang Zhifang, Wu Peiwei. Internet of Things smart home based on ESP8266 and Gizwits Cloud [J]. Internet of Things Technology, 2023, 13(03): 140-142.
4. Liu Shibing, Liu Chengyin. Water quality online monitoring system based on human cloud [J]. Internet of Things Technology, 2021, 11(10): 30-32.
5. Wang Hongyu. Design and implementation of smart home monitoring system based on WiFi under OneNET cloud platform [D]. Inner Mongolia University, 2021.
6. Chen Yacong, Feng Xingle, Chen Shupeng. Smart home modular expansion system based on MQTT protocol [J]. Integrated Circuits and Embedded Systems, 2024, 24(02): 86-90.
7. Yang Lulu. Design of home environment monitoring and control system based on STM32 [D]. Tianjin University of Science and Technology, 2019.
8. Hou Changhua, Bao Jinghai, Liang Yankui, et al. Design of remote home environment monitoring system based on Cortex M4 and Android APP [J]. Internet of Things Technology, 2019, 9(10): 104-106+109.
9. Zhao Junfeng, Liu Xueliang, Zhi Yichao. Design of home health care constant temperature heater based on STM8 [J]. Journal of Nanyang Institute of Technology, 2017, 9(06): 44-49.
10. Cheng Xuemin. Design of constant temperature intelligent control system based on STM32F103C8T6[J]. Electronic Production, 2023, 31(19):26-29.
11. Hu Shuai, Jiang Yafeng, Du Mingquan, et al. Intelligent energy-saving constant temperature control device based on STM32[J]. Mechanical Manufacturing and Automation, 2020, 49(04):184-187.

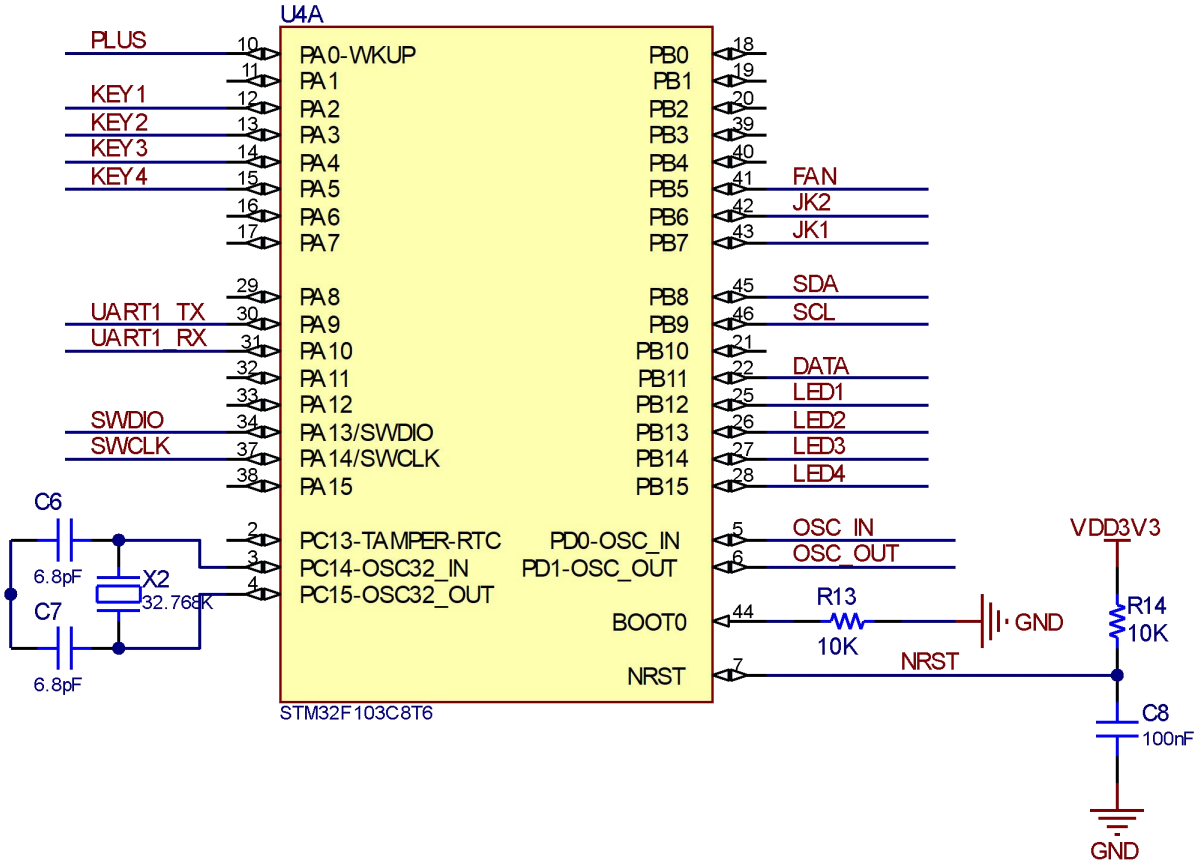
12. Wu Yangbo, Yan Jinmeng, Hu Haifeng. Design and implementation of smart home environment monitoring system based on ESP8266[J]. Communications and Information Technology, 2024, (02):10-14.
13. Wang Ziliang. Research on the design of temperature and humidity monitoring system based on single chip microcomputer[J]. Modern Salt Chemical Industry, 2019, 46(06):72-73.
14. Ma Qiang. Design and implementation of smart home system based on Internet of Things technology[J]. Electronic Technology, 2024, 53(02): 252-253.
15. Jie Xiao, Jing Tao Li. Design and Implementation of Intelligent Temperature and Humidity Monitoring System Based on ZigBee and WiFi[J]. Procedia Computer Science, 2020, 166.

Appendix №1

To the diploma project

On the topic: «Home constant temperature and humidity system»

Kyiv – 2024



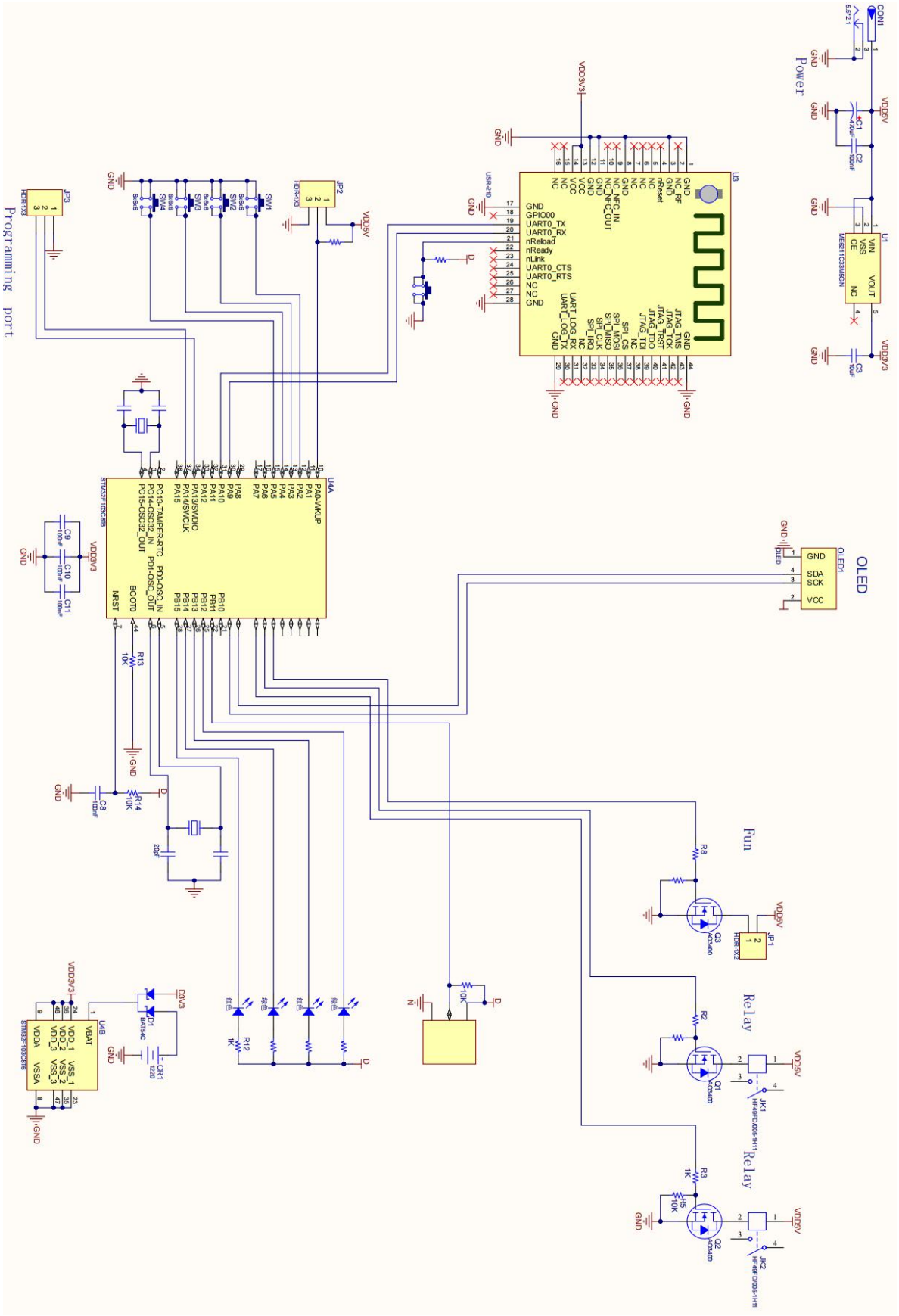
					ІАЛЦ.467100.004 Д1		
Зм.	*Арк.*	*№ докум.*	*Підп.*	*Дата*			
Розробив		Лі Юемін			*Схема структурна*	*Літ.*	*Аркуш*	*Аркушів*
Перевірив		Павлов В. Г.					*1*	*1*
Реценз.		Корнієнко Б. Я.						
Н.контр.		Павлов В. Г.			*Додаток 1*	*НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ*		
Затв.		Стіренко С. Г.						

Appendix №2

To the diploma project

On the topic: «Home constant temperature and humidity system»

Kyiv – 2024



					ІАЛЦ.467100.005 Д2
Зм.	*Арк.*	*№ докум.*	*Підп.*	*Дата*	
Розробив		Лі Юемін			*Схема функціональна*
Перевірив		Павлов В. Г.			*Додаток 2*
Реценз.		Корнієнко Б. Я.			
Н.контр.		Павлов В. Г.			
Затв.		Стіренко С. Г.			

Літ.	*Аркуш*	*Аркушів*
	1	*1*

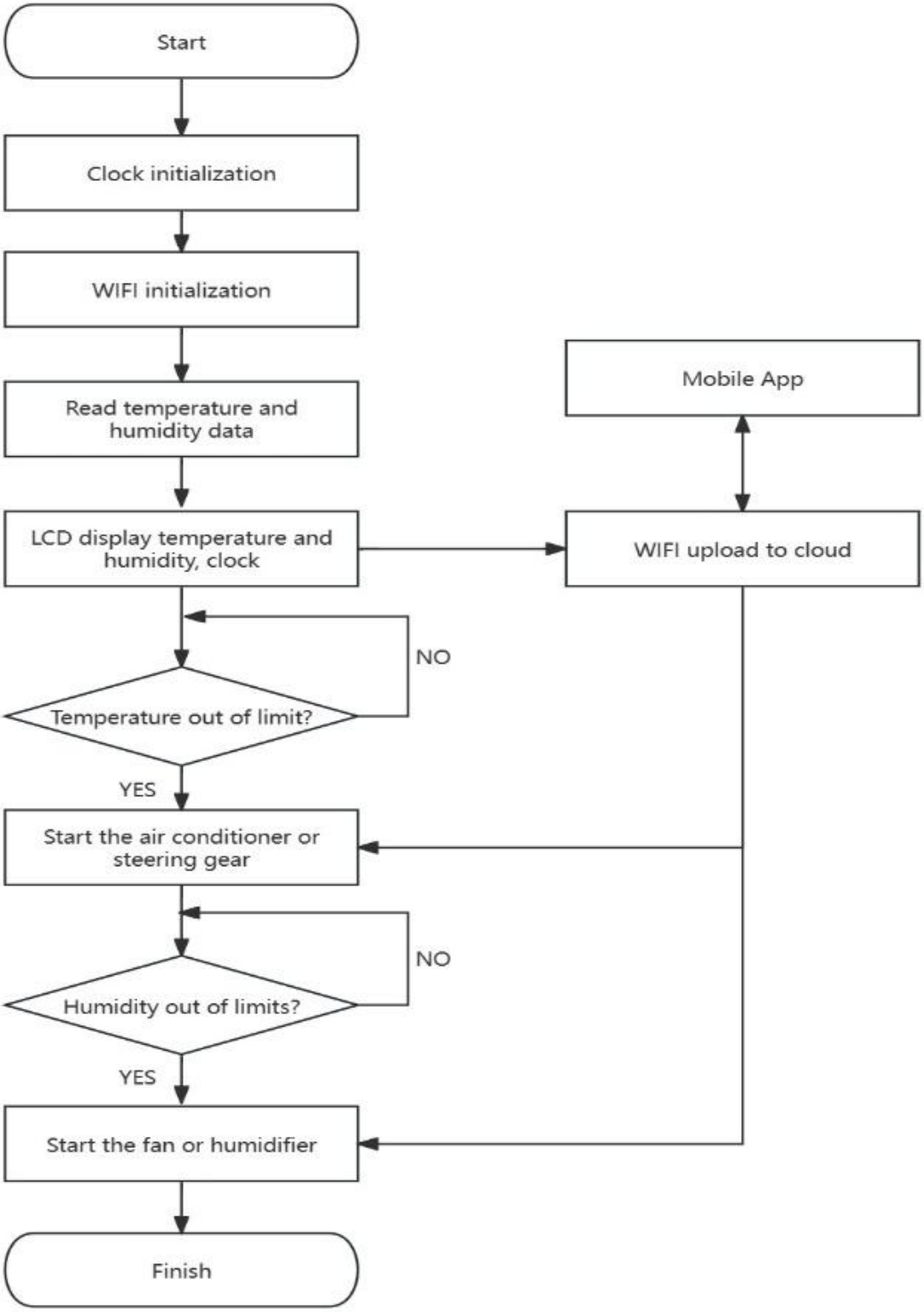
НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ

Appendix №3

To the diploma project

On the topic: «Home constant temperature and humidity system»

Kyiv – 2024



					ІАЛЦ.467100.006 Д3
Зм.	*Арк.*	*№ докум.*	*Підп.*	*Дата*	
Розробив		Лі Юемін			*Схема принципова*
Перевірив		Павлов В. Г.			*Додаток 3*
Реценз.		Корнієнко Б. Я.			
Н.контр.		Павлов В. Г.			
Затв.		Стіренко С. Г.			

Літ.	*Аркуш*	*Аркушів*
	1	*1*

НТУУ «КПІ ім. Ігоря Сікорського» ФІОТ

Appendix №4

To the diploma project

On the topic: «Home constant temperature and humidity system»

Kyiv – 2024

```
void task_key(void)
{
    if(IsKeyDown3() != 0)                /* Time setting button */
    {
        bsp_DelayMS(10);
        if(IsKeyDown3() != 0)
        {
            set_u_type = 0;
            set_time_type ++;
            if(set_time_type == 1)
            {
                set_year = g_tRTC.Year;
                set_mon = g_tRTC.Mon;
            set_day = g_tRTC.Day;
          set_hour = g_tRTC.Hour;
          set_min = g_tRTC.Min;
          set_sec = g_tRTC.Sec;
          OLED_ClrScr(0);          /* Clear screen */
            }
```

```
            else if(set_time_type == 7)
            {
                set_time_type = 0;
                RTC_WriteClock(set_year, set_mon, set_day, set_hour, set_min,
set_sec);
            }
        }while(IsKeyDown3() != 0);
    }
    if(IsKeyDown4() != 0)                /* Threshold setting button */
    {
        bsp_DelayMS(10);
        if(IsKeyDown4() != 0)
        {
            set_time_type = 0;
            set_u_type ++;
            if(set_u_type == 1)
            {
                OLED_ClrScr(0);         /* Clear screen */
            }
            else if(set_u_type == 5)            /* Threshold setting completed */
            {
                set_u_type = 0;
            }
```

```
        }while(IsKeyDown4() != 0);
    }

    if(IsKeyDown1() != 0)                   /* Time-key */
    {
        bsp_DelayMS(10);
        if(IsKeyDown1() != 0)
        {
            switch(set_time_type)
            {
                case 1:                     /* Set year */
                    set_year --;
                    if(set_year < 2024)
                    {
                        set_year = 2024;
                    }
                    break;
                case 2:                     /* Set month */
                    set_mon --;
                    if(set_mon < 1)
                    {
                        set_mon = 12;
                    }
                    break;
                case 3:                     /* Set day */
                    set_day --;
                    if(set_day < 1)
                    {
                        set_day = 31;
                    }
```

```
            break;
        case 4:                 /* Set hour */
            set_hour --;
            if(set_hour < 0)
            {
                set_hour = 23;
            }
            break;
        case 5:                 /* Set minutes */
            set_min --;
            if(set_min < 0)
            {
                set_min = 59;
            }
            break;
        case 6:                 /* Set seconds */
            set_sec --;
            if(set_sec < 0)
            {
                set_sec = 59;
            }
            break;
    }

    switch(set_u_type)
    {
        case 1:
            temp_ul --;
            break;
```

```
                case 2:
                    temp_uh --;
                    break;
                case 3:
                    hum_ul --;
                    break;
                case 4:
                    hum_uh --;
                    break;
            }
        }while(IsKeyDown1() != 0);
    }

    if(IsKeyDown2() != 0)                    /* time+key */
    {
        bsp_DelayMS(10);
        if(IsKeyDown2() != 0)
        {
            switch(set_time_type)
            {
                case 1:                      /* Set year */
                    set_year ++;
                    if(set_year > 2099)
                    {
                        set_year = 2024;
                    }
                    break;
                case 2:                      /* Set month */
                    set_mon ++;
```

```
        if(set_mon > 12)
        {
            set_mon = 1;
        }
        break;
    case 3:                 /* Set day */
        set_day ++;
        if(set_day > 31)
        {
            set_day = 1;
        }
        break;
    case 4:                 /* Set hour */
        set_hour ++;
        if(set_hour > 23)
        {
            set_hour = 0;
        }
        break;
    case 5:                 /* Set minutes */
        set_min ++;
        if(set_min > 59)
        {
            set_min = 0;
        }
        break;
    case 6:                 /* Set seconds */
        set_sec ++;
        if(set_sec > 59)
        {
```

```
                    set_sec = 0;
                }
                break;
            }

            switch(set_u_type)
            {
                case 1:
                    temp_ul ++;
                    break;
                case 2:
                    temp_uh ++;
                    break;
                case 3:
                    hum_ul ++;
                    break;
                case 4:
                    hum_uh ++;
                    break;
            }
        }while(IsKeyDown2() != 0);
    }

}
```

```
int main(void)
{
	char buf[16];
	uint16_t tmp_co2;

	/* Set font parameters */
	{
		tFont16.FontCode = FC_ST_16;		/* Font code 16 dot matrix */
		tFont16.FrontColor = 1;			/* Font color 0 or 1 */
		tFont16.BackColor = 0;			/* Text background color 0 or 1 */
		tFont16.Space = 0;			/* Text spacing, unit = pixels */

		tFont12.FontCode = FC_ST_12;		/* Font code 12 dot matrix */
		tFont12.FrontColor = 1;			/* Font color 0 or 1 */
		tFont12.BackColor = 0;			/* Text background color 0 or 1 */
		tFont12.Space = 0;			/* Text spacing, unit = pixels */
	}


	bsp_Init();
	OLED_DispStr(0, 0, "pwr on", &tFont12);
	bsp_StartAutoTimer(0, 200);
	bsp_StartAutoTimer(1, 1000);
```

```
    EXTI_Config();
        while(1)
    {
        task_wifi_get();
        task_auto();
        task_key();

        if(bsp_CheckTimer(0))
        {
            if(DHT11_ReadData(&dht11))
            {
                temp = dht11.Temp;
                hum = dht11.Hum;
            }
            OLED_StartDraw();

            RTC_ReadClock();
            OLED_StartDraw();

            if(set_time_type == 0 && set_u_type == 0)                /* Display real-
time time when not set time */
            {
                RTC_ReadClock();
                sprintf(buf, "%4d/%02d/%02d %02d:%02d:%02d",
                            g_tRTC.Year, g_tRTC.Mon, g_tRTC.Day,
                            g_tRTC.Hour, g_tRTC.Min, g_tRTC.Sec);
                OLED_DispStr(0, 0, buf, &tFont12);

                sprintf(buf, "温度:%3d ℃", temp);
                OLED_DispStr(0, 16, buf, &tFont16);
```

```
                sprintf(buf, "湿度:%3d %%", hum);
                OLED_DispStr(0, 32, buf, &tFont16);
                sprintf(buf, "%4d", speed);
                OLED_DispStr(0, 48, buf, &tFont16);
            }
            else                                    /* Display the set time when setting
the time */
            {
                sprintf(buf, "%4d/%02d/%02d %02d:%02d:%02d",
                        set_year, set_mon, set_day,
                        set_hour, set_min, set_sec);
                OLED_DispStr(0, 0, buf, &tFont12);

                sprintf(buf, "%3d     %3d", temp_ul, temp_uh);
                OLED_DispStr(0, 16, buf, &tFont16);

                sprintf(buf, "%3d     %3d", hum_ul, hum_uh);
                OLED_DispStr(0, 32, buf, &tFont16);
            }

            OLED_EndDraw();
        }
        if(bsp_CheckTimer(1))
        {
            speed = plus;
            plus = 0;
            task_wifi_send();
        }
    }
}
```

```
void Analyze(uint8_t *buf, uint8_t len)
{
    uint8_t tmp;

    if(buf[3] == 3) /* temperature threshold */
    {
        temp_ul = buf[4] << 8 | buf[5];
    }
    else if(buf[3] == 4)      /* temperature threshold */
    {
        temp_uh = buf[4] << 8 | buf[5];
    }
    else if(buf[3] == 5)      /* humidity threshold */
    {
        hum_ul = buf[4] << 8 | buf[5];
    }
    else if(buf[3] == 6)      /* humidity threshold */
    {
        hum_uh = buf[4] << 8 | buf[5];
    }
    else if(buf[3] == 7)      /* Humidifier switch */
    {
        if(buf[5] == 1)
```

```
        {
            comSendBuf(COM1, (uint8_t *)ack_7_1, 8);
            bsp_jk_on(1);
            jk1_sw = 1;
        }
        else
        {
            comSendBuf(COM1, (uint8_t *)ack_7_0, 8);
            bsp_jk_off(1);
            jk1_sw = 0;
        }
    }
    else if(buf[3] == 8)      /* air conditioner switch */
    {
        if(buf[5] == 1)
        {
            comSendBuf(COM1, (uint8_t *)ack_8_1, 8);
            bsp_jk_on(2);
            jk2_sw = 1;
        }
        else
        {
            comSendBuf(COM1, (uint8_t *)ack_8_0, 8);
            bsp_jk_off(2);
            jk2_sw = 0;
        }
    }
    else if(buf[3] == 9)      /* Humidifier switch */
    {
        if(buf[5] == 1)
```

```
        {
            comSendBuf(COM1, (uint8_t *)ack_9_1, 8);
            bsp_fan_on();
            fan_sw = 1;
        }
        else
        {
            comSendBuf(COM1, (uint8_t *)ack_9_0, 8);
            bsp_fan_off();
            fan_sw = 0;
        }
    }
    else if(buf[3] == 10)     /* Plumbing switch */
    {
        if(buf[5] == 1)
        {
            comSendBuf(COM1, (uint8_t *)ack_10_1, 8);
            bsp_open_water();
            step_motor_sw = 1;
        }
        else
        {
            comSendBuf(COM1, (uint8_t *)ack_10_0, 8);
            bsp_close_water();
            step_motor_sw = 0;
        }
    }

}
```

About each component working code

```
void task_auto(void)
{
    if(temp <= temp_ul)            /* The temperature is below the threshold, turn on the
water heating */
    {
        step_motor_off_flag = 0;
        if(step_motor_on_flag == 0)
        {
            step_motor_on_flag = 1;
            step_motor_sw = 1;
            bsp_open_water();
        }

    }
    else
    {
        step_motor_on_flag = 0;
        if(step_motor_off_flag == 0)
        {
            step_motor_off_flag = 1;
            step_motor_sw = 0;
            bsp_close_water();
        }
    }

    if(temp >= temp_uh)        /* The temperature is higher than the threshold, turn on the
air conditioner */
    {
```

```
        jk2_off_flag = 0;

        if(jk2_on_flag == 0)
        {
            jk2_on_flag = 1;
            jk2_sw = 1;
            bsp_jk_on(2);
        }

    }
    else
    {
        jk2_on_flag = 0;
        if(jk2_off_flag == 0)
        {
            jk2_off_flag = 1;
            jk2_sw = 0;
            bsp_jk_off(2);
        }
    }

    if(hum <= hum_ul)          /* Turn on the humidifier when the humidity is less than the threshold */
    {
        jk1_off_flag = 0;

        if(jk1_on_flag == 0)
        {
            jk1_on_flag = 1;
            jk1_sw = 1;
```

```
            bsp_jk_on(1);
        }

    }
    else
    {
        jk1_on_flag = 0;
        if(jk1_off_flag == 0)
        {
            jk1_off_flag = 1;
            jk1_sw = 0;
            bsp_jk_off(1);
        }
    }

    if(hum >= hum_uh)
    {
        fan_off_flag = 0;

        if(fan_on_flag == 0)
        {
            fan_on_flag = 1;
            fan_sw = 1;
            bsp_fan_on();
        }

    }
    else
        {
            fan_on_flag = 0;
```

```
        if(fan_off_flag == 0)
        {
            fan_off_flag = 1;
            fan_sw = 0;
            bsp_fan_off();
        }
    }

    if(step_motor_on_flag == 0 && jk2_on_flag == 0)
    {
        bsp_led_off(2);
        bsp_led_on(1);
    }
    else
    {
        bsp_led_off(1);
        bsp_led_on(2);
    }
    if(jk1_on_flag == 0 && fan_on_flag == 0)
    {
        bsp_led_off(4);
        bsp_led_on(3);
    }
    else
    {
        bsp_led_off(3);
        bsp_led_on(4);
    }
}
```

3.1.3 About wifi related settings and working logic

Wifi send to USR cloud

```
void task_wifi_get(void);
static void EXTI_Config(void);

void task_wifi_send(void)
{
    uint8_t tx_buf[32];
    uint16_t crc;

    tx_buf[0] = 0x01;    // Slave Address
    tx_buf[1] = 0x46;    // function code
    tx_buf[2] = 0x00;    // Register Address
    tx_buf[3] = 0x00;
    tx_buf[4] = 0x00;    // Number of registers
    tx_buf[5] = 11;
    tx_buf[6] = 22;      // Number of data bytes

    tx_buf[7] = 0;
    tx_buf[8] = temp;
    tx_buf[9] = 0;
    tx_buf[10] = hum;
    tx_buf[11] = speed >> 8;
    tx_buf[12] = speed;
    tx_buf[13] = 0;
    tx_buf[14] = temp_ul;
    tx_buf[15] = 0;
    tx_buf[16] = temp_uh;
```

```
    tx_buf[17] = 0;
    tx_buf[18] = hum_ul;
    tx_buf[19] = 0;
    tx_buf[20] = hum_uh;
    tx_buf[21] = 0;
    tx_buf[22] = jk1_sw;
    tx_buf[23] = 0;
    tx_buf[24] = jk2_sw;
    tx_buf[25] = 0;
    tx_buf[26] = fan_sw;
    tx_buf[27] = 0;
    tx_buf[28] = step_motor_sw;

    crc = CRC16_Modbus(tx_buf, 29);
    tx_buf[29] = crc >> 8;
    tx_buf[30] = crc;

    comSendBuf(COM1, tx_buf, 31);
}

void bsp_open_water(void)
{
    bsp_SetTIMOutPWM(GPIOA, GPIO_Pin_1, TIM2, 2, 50, 300);
}
void bsp_close_water(void)
{
    bsp_SetTIMOutPWM(GPIOA, GPIO_Pin_1, TIM2, 2, 50, 1000);
}
```

Wifi receives USR cloud control signal

```
void task_wifi_get(void)
{
    uint8_t ucData;
    static uint8_t ucHead = 0;
    static uint8_t ucaBuf[64];
    static uint8_t ucPos = 0;

    while(1)
    {
        if (comGetChar(COM1, &ucData))
        {
            if(ucHead == 0)
            {
                if(ucData == 0x01)   /* data pack */
                {
                    ucHead = 1;
                    ucPos = 0;
                    ucaBuf[ucPos++] = ucData;
                }
            }
            else if(ucHead == 1)
            {
                if(ucData == 0x06)   /* data pack */
                {
                    ucHead = 2;
                    ucaBuf[ucPos++] = ucData;
                }
                else
```

```
            {
                if (ucPos < sizeof(ucaBuf))
                {
                    ucaBuf[ucPos++] = ucData;

                    if(ucPos == 8)
                    {
                        Analyze(ucaBuf, ucPos);
                        ucHead = 0;
                    }
                }
                else
                {
                    ucHead = 0;
                }
            }
            continue;     /* There may be more data, continue to analyze */
        }
        break;
    }
}
```

```
static void EXTI_Config(void)
{
    /* Configuration PA8 */

    GPIO_InitTypeDef GPIO_InitStructure;
    EXTI_InitTypeDef    EXTI_InitStructure;
    NVIC_InitTypeDef    NVIC_InitStructure;

    /* GPIO Clock */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);

    /* Configure PA7 to input floating mode */
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    /* AFIO Clock */
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO, ENABLE);

    /* Connect EXTI0 Line to PA.00 pin */
    GPIO_EXTILineConfig(GPIO_PortSourceGPIOA, GPIO_PinSource0);

    /* Configure EXTI0 line */
    EXTI_InitStructure.EXTI_Line = EXTI_Line0;
```

```
    EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
    EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Rising;        /* Rising edge trigger
*/
    EXTI_InitStructure.EXTI_LineCmd = ENABLE;
    EXTI_Init(&EXTI_InitStructure);

    /* Enable and set EXTI0 Interrupt to the lowest priority */
    NVIC_InitStructure.NVIC_IRQChannel = EXTI0_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0x0F;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0x0F;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);

}
```