LOW-POWER AND LOW-COST MODULAR SMART SENSOR PLATFORM WITH AUTOMATIC NODE NETWORKING AND INFORMATION MANAGEMENT

Chen Ling

A thesis
in
The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Science (Computer Science) at
Concordia University
Montreal, Quebec, Canada

December 2020

© Chen Ling, 2021

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Chen Ling**

Entitled: **Low-Power and Low-Cost Modular Smart Sensor Platform with Automatic Node Networking and Information Management**

and submitted in partial fulfillment of the requirements for the degree of

Master of Science (Computer Science)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. Tse-Hsun (Peter) Chen

______________________________ Examiner
Dr. Jinqiu Yang

______________________________ Examiner
Dr. Hua Ge

______________________________ Supervisor
Dr. Weiyi (Ian) Shang

______________________________ Supervisor
Dr. Liangzhu (Leon) Wang

Approved by ______________________________
Leila Kosseim
Chair of Department or Graduate Program Director

__________ 2020 _____ ______________________________
Dr. Amir Asif, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Low-Power and Low-Cost Modular Smart Sensor Platform with Automatic Node Networking and Information Management

Chen Ling

The development of industry and computer networks and the requirements of automation in daily life have led to the need for greater networking capability between smarter computer programs, services, and environments. In the past few decades, because of the rapid development of integrated circuits, the computing ability of chips increased rapidly, and the size of peripheral components and devices decreased rapidly. Nowadays, computer systems must handle far more than single calculations, and people expect them to be increasingly smarter to make their work easier. Systems are based on a modular design to respond rapidly to changing requirements while consuming less energy. Companies have developed many kinds of devices and software to satisfy various requirements, including environmental sensing, data processing, and automatic controlling. These products provide hardware and software to help researchers collect various kinds of environmental data. However, while not all products facilitate remote data collection, those that do often entail increased communication costs and power consumption, especially when deploying many devices. Many public places have been closed in 2020 due to COVID-19, which poses new challenges to data collection, including gathering environmental data when access to the site is prohibited and how to transfer data via cellular networks while minimizing costs and power consumption. This thesis proposes a design that encompasses hardware and software that will solve the problems related to remote data collection, edge computing, remote system control, low power data collection, the modular system, data security, and backup. The ultimate goals include reducing the costs of system maintenance, minimizing the difficulties of data collection and processing, simplifying system configuration, and controlling and ensuring data integrity.

Acknowledgments

I was fortunate to have met many people while conducting this research, and they offered their generous help and advice. I would like to sincerely thank my supervisors: Dr. Weiyi Shang and Dr. Liangzhu Wang. Their professional guidance helped me gradually overcome the difficulties. Also, many thanks to my lab mates and friends: Chang Shu, Danlin Hou, Senwen Yang, Lili Ji, Yuhao Mao, Jianing Yao, among others, for their help and kindness. Special thanks to Dr. Yong Zeng for his thoughts on design methods, and sincere thanks to my family and my parents, Chunyan Zhao and Shiping Ling, for their support and understanding. Thank you to my university professors during my undergraduate studies: Haifeng Sun, Wenxin Yu, and Lili Song. Even though I have graduated, we still keep in touch. Lastly, thanks to all those who have helped me along the way. Even if I have not mentioned you here, know that you are all important to me.
Thank you all.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Recently, there is an increasing need for quality field measurements to be collected to study urban environments and microclimate (including air temperature, speed, humidity, and pollution level) and their impacts on buildings. Especially during extreme weather events, such as heatwaves, this field information will evaluate the impacts of the duration, intensity, and frequency of heatwaves on building indoor environments and the occupants inside. Such impacts can be quantified by installing indoor sensors, including air temperatures, humidities, and carbon dioxide levels. As a result, indoor and outdoor data at multiple locations in an urban context will need to be collected. Recently, this is particularly important for an ongoing research project in our research group, which is funded by the National Sciences and Engineering Research Council (NSERC) of Canada through the "Advancing Climate Change Science in Canada" program: "Assessment and Mitigation of Summertime Overheating Conditions in Vulnerable Buildings of Urban Agglomerations." The project aims to make the buildings more resilient to extreme heat events such as urban heatwaves and study the impacts on the building's interior based on collecting and modeling the relevant data. Therefore, we need to collect as much environmental data as possible. The larger the amount of data we collected, the more accurate the modeling and analysis effect will be. During the project, we deployed many sensors that are supported by commercial solutions. However, these sensors have the adverse price and functionality factors, so a solution needs to be designed to help reduce costs and expand functionality to collect more data. Moreover, the COVID-19 pandemic has highlighted the shortcomings of some commercial solutions, which hinder information collection. Therefore, in the design process of this thesis, we need to identify the existing functions in a low-cost design and, more importantly, we need to incorporate more functions, collect more data and simplify the information collection process.

At the same time, it is also necessary to consider the connectivity with the systems involved in other projects in the laboratory. Therefore, in addition to the hardware design, it is also necessary to complete the design of the related software system.

1.1.1 Commercial Solutions

Two main commercial solutions are currently in use in our laboratory. The first is the Niagara series from Tridium Inc., which is a complete commercial solution that includes hardware and software while providing a development platform. Niagara allows developers to extend a large number of hardware modules by developing drivers through the Java language. It can also be paired with other industrial hardware and has control capabilities with the hardware. The strong compatibility of the Niagara platform makes it the preferred control platform for many industrial platforms. It has also been successfully applied to many environmental monitoring projects, such as water monitoring. However, the Niagara platform is very expensive. The price of a single hardware node ranges from several thousand to tens of thousands USD. The server of the Niagara platform also requires software licenses to develop programs and drivers, as Niagara users need professional training by Tridium, so the cost of labor, hardware, and software on the Niagara platform is expensive. Onset Inc. also provides commercial solutions for the laboratory in the form of the indoor and outdoor environmental sensor series. The indoor sensor series includes sensors that collect ambient temperature and humidity, and the price for a single node is 89 USD. OnSet Inc. provides a dedicated mobile phone application to download data via Bluetooth.

This requires a maintenance group to regularly visit the installation place to download data to a mobile phone. One phone can download data from one node at a time. Due to Covid-19, the site is closed and it is difficult to download the data. OnSet Inc. also provides sensors with cellular communication capabilities. However, this type of sensor is only available in the outdoor sensor series and consumes a significant amount of energy. It requires continuous power and a large lithium battery pack to ensure power supply. OnSet Inc. also provides supporting software for data analysis and processing.

1.1.2 Open-source Platform

In addition to the commercial solutions, many other open-source platforms on the market can fulfill the related needs. Raspberry Pi can be used on Linux-based embedded platforms, and Arduino is available for IoT development. Due to the low price of Raspberry Pi, there is an extensive supporting software library and development documentation and various peripheral hardware customized for Raspberry Pi. Raspberry Pi can build a hardware platform rapidly, which meets the functional requirements. The Linux-based software and hardware architecture allows it to execute any modern computer language, and it is easy to update the software because it runs a complete modern operating system. The Arduino platform also has comprehensive development documents, and Arduino provides a supporting IDE and a framework based on C++ as its development language. Most of the hardware modules designed for Raspberry Pi can also be used in Arduino, so Arduino can build the hardware platform quickly that has the required functions, but the computing performance is limited compared to Raspberry Pi. Although Raspberry Pi and Arduino have these advantages, they still have certain shortcomings. Raspberry Pi and Arduino are primarily for education, so they have added as many functions and interfaces as possible. These additional functions and interfaces occupy substantial space on the circuit board. The development based on these types of platforms leads to excessive product size, and the unnecessary additional hardware is not energy efficient. For example, Ethernet PHY does not reduce platform power consumption.

As more additional interfaces are provided on these platforms, the circuit integration rate is reduced. Too many cables lead to uncertainty in hardware communication. For example, poor quality module cables, poor interface connectivity caused by repeated plugging and unplugging cables will affect system stability. Moreover, the connector design of some modules may also occupy communication lines that should not be occupied, which will lead to decreased platform expansion ability. The Arduino platform may also face performance problems. For example, Arduino UNO's microcontroller, the ATmega328P, uses an 8-bit controller with only 2KB RAM. This kind of low-performance controller can reduce system power consumption. However, when faced with complex programs or if the calculations require handling large amounts of data, the Arduino will not able to meet the computing ability requirements. At the same time, the price of the platform is also a problem that needs to be considered. In mid-2020, the retail price of Raspberry Pi 3 was 35 USD, while Arduino UNO is 23 USD, which is low enough compared to commercial platforms. However, considering additional hardware modules and the price of battery and design cases, the price is still not ideal.

1.1.3 Requirements

Based on the price and functional issues of the above commercial and open-source platforms, we hope to create a new platform that meets the following needs:

- We aim to reduce the pressure of cloud services and the delayed response of the system so that some lightweight calculations can be completed in the sensor itself, which will increase the size of the program code and running memory requirements, so a more powerful platform design is needed to perform more complicated calculations.

- Data downloading on commercial platforms has always been a problem, and manual data downloads may be interrupted due to the inaccessibility of the installation site. These include insurmountable challenges, such as the closure of buildings caused by the Covid-19 crisis. We hope that the platform can automatically transfer data through networks, especially cellular networks, to save human resource costs and improve our ability to handle such crises.

- Due to the high power consumption of cellular network modules and to reduce the costs of cellular network plans, the hardware platform should construct an inter-device network between the cellular node and data collecting nodes through other protocols and transmit data to the server through a single node that has the cellular network communication ability to save power and costs.
- If a single cellular network communication node can cover a larger communication range and collect data from as many data nodes as possible, the platform may provide a forwarding network between each data node, turning every sensor node into a network transit station to reduce the number of cellular communication nodes required.
- As the laboratory may have other needs for data collection and analysis in the future, the platform might have a certain reconfiguration ability to adapt to new application scenarios quickly.
- The project participants do not all have professional computer backgrounds. Therefore, the platform must have a simple management client that is user-friendly.
- Since the data may be used in other projects, we hope that the platform interface will allow the export of data and interact with the software systems of other projects to enhance collaboration capabilities between laboratory projects.

1.2 Focus of Research

Based on the needs and problems encountered by the existing commercial solutions in the laboratory, as well as related research on open-source platforms, the costs involved, and the performance and functions of various platforms, Table 1 provides a comparison of the various factors:

Table 1: Comparisons between existed platforms

Platform function	Niagara	OnSet indoor series	Raspberry Pi	Arduino UNO
Information management software	Y	Y		
Database	Y			
Cellular communication	Y		Purchase additional modules and develop related functions	Purchase additional modules and develop related functions
Battery-powered	Y	Y	Purchase additional modules	Purchase additional modules
Low power consumption		Y		Y
Data storage	Y	Y	Purchase additional modules	Purchase additional modules and develop related functions
Cost (2020)	Thousands to tens of thousands USD	85 USD per sensor	35 USD per platform without modules	23 USD per platform without modules
Modular design	Y		Y	Y
Size	Extra large	Small	Medium	Medium
Data backup	Y		Purchase additional modules and develop related functions	Purchase additional modules and develop related functions

Platform function	Niagara	OnSet indoor series	Raspberry Pi	Arduino UNO
Wireless communication	Y	Y	Purchase additional modules and develop related functions	Purchase additional modules and develop related functions
Multi-node communication	Y		Purchase additional modules and develop related functions	Purchase additional modules and develop related functions
Hardware resources	High		High	Low
Secondary development	Limited		Y	Y

In the research process, we found that the hardware platform provided by the commercial solution, which offers all the functions, is large, expensive, and requires substantial energy. The hardware platform provided by the less expensive commercial solution has limited functions in its communication ability and computing performance. The software system provided by the commercial solution is usually limited to the specific company's hardware equipment, the data analysis function also needs to be executed on the specific company's platform, and the docking between different commercial solutions is problematic. The commercial solutions also offer limited support for platform customization, and cheaper commercial solutions do not even support the secondary development of the platform; specialized technicians are required to provide technical support. Moreover, the open-source platform offers user-friendly platform customization and software development. However, the integration rate of the platform is low, and the size of the platform is slightly larger. The implementation of various functions depends on the expansion of external modules and the software needs to be implemented by the developer. The cost and the product size of the open-source platform are inferior to the platforms provided by commercial solutions.

1.3 Targets

Based on comparative research and requirements analysis, we decided to create a new platform that would solve the specific problems that people and the environment may encounter now and in the future. We will directly start with circuit design, electronic component selection, and factory production. This will significantly reduce the cost of the hardware platform, improve the platform integration rate, and reduce the size of the circuit. Designing the circuit in-house allows us to omit irrelevant functions on the hardware, which improves energy performance and costs. To simplify the user's use process, we will also provide information management software and database, as commercial solutions do, but our management system provides external interfaces to achieve better compatibility.

1.4 Application Scenarios

In this thesis, we primarily discuss the application of collecting environmental information at a fixed indoor location. Sensors will be installed at the location, and multiple sensors will form a private network. The center of the network is the control node, which has cellular network communication ability. The communication process can be affected by the blocking of the wireless signal caused by people, walls, and doors. In these instances, the network strength can be increased by deploying more sensors. The sensor platform can also be used for human tracking and movement recording. Because of the modular design, motion sensors can replace environmental sensors, which will provide accurate data for movement recording and motion tracking, and it offers convenience in motion capture, film and television and game production, and human-motion research. The sensor platform can be used for edge computing to collect and process data directly on the platform itself. The off-line communication function and the built-in computing function enable the platform to be used for environmental information collection, such as handheld measurement. This is vital for platforms in environments where the network and personal computers cannot be used.

1.5 Thesis Outline

This chapter introduces the advantages and disadvantages of the laboratory's existing commercial solutions and the open-source platforms on the market, introduced the related problems that the laboratory faces on the project, and proposed corresponding requirements for the platform based on the related problems. Based on the platform requirements, the research goals are formulated and the scenarios applicable to the platform are summarized. In the following chapters, this thesis introduces the following contents:

- The second chapter details related research on the relevant parts of the platform design process, which will improve our solutions.
- In Chapter 3, we offer solutions to the hardware design of the node in the platform. The hardware foundation of the functions that the platform can provide and the basic hardware architecture are shown in this chapter.
- Chapter 4 presents a solution to the software design of the node in the platform, focusing on how to process data and the composition of a Wi-Fi-based sensor network.
- In Chapter 5, we introduce a solution to the implementation of the platform's management system, including the functional components of the server and the management client of the system, as well as the external compatibility interface, which is discussed here.
- Chapter 6 describes our testing of the platform functions. An experimental environment is created. The platform will work under ideal conditions and show how to solve the problems encountered in the laboratory project.

- In Chapter 7, we summarize the overall advantages of this program and some limitations that it faces, and we discuss the future direction of the work.

Chapter 2

Related Work

This chapter examines background knowledge related to the topic. The relevant content is divided into three parts for discussion. The first part addresses the general system design, the second part examines two related documents, and the third part surveys the relevant software, hardware, and libraries.

2.1 Environmental Sensors

After years of development, environmental sensors have gradually penetrated daily use. The environmental sensors have gradually developed from a single device to a sensor network, the network composition has gradually changed from wired to wireless, and the functions of sensors have gradually increased with the progress of semiconductor technology.

Figure 1[22]shows the design scheme of a conventional environmental sensor from 1997. The center of the system is an 80C31 8-bit embedded microcontroller. The system contains a sensing unit and a control unit. The additional keyboard and LCD display form a complete system design. The single-node sensor of this architecture is still widely used in industrial equipment and household appliances. For example, the speed sensor and temperature sensor of the drum washing machine belong to the sensor unit, while the speed control of the motor and the switch control of the heater belong to the control unit. Moreover, the buttons of the washing machine and the LED display device correspond to the keyboard and LCD screen, and the entire washing machine operation process is completed by the embedded microcontroller. Regardless of how the sensor evolves, the design of such a microcontroller with a sensor unit is still the core structure of the sensor node.

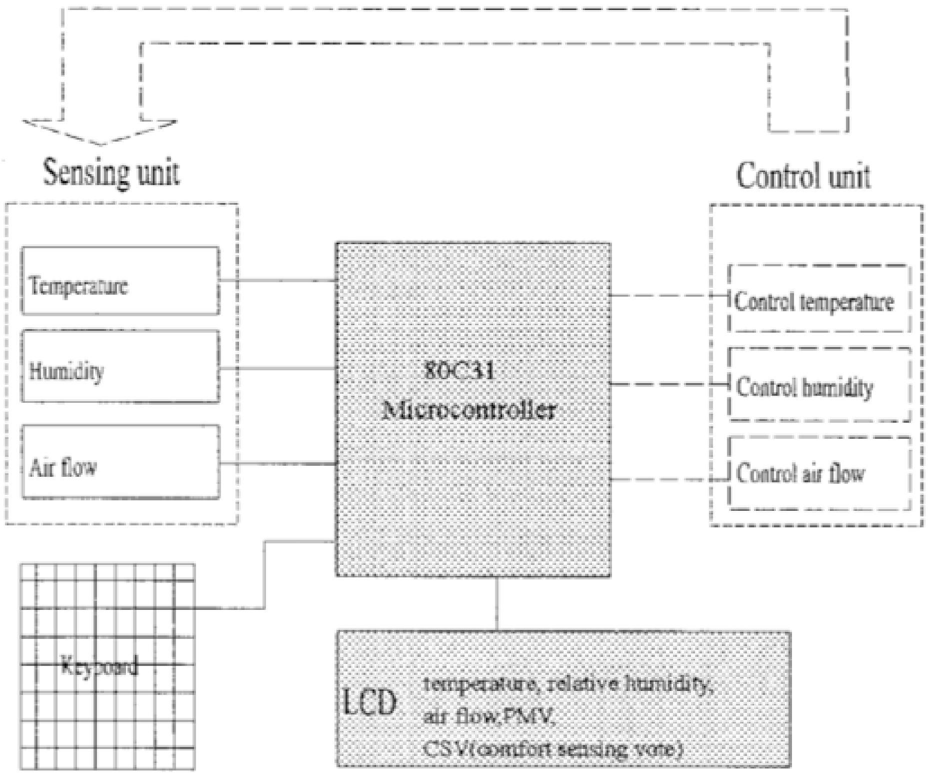


Figure 1: A block diagram of the PMV-indicator

Thereafter, with the development of computer networks, the network part was added to part of the sensor system design. Early sensors with computer network support relied on wired connections and could not communicate directly with the computer network, mainly relying on industrial buses and the conversion card on the computer to connect the computer network with the sensor network. This system is connected to the simulator network and is included as a client of the simulator. This is done by using the network formatter bridge [7] as Figure 2[7] shows:

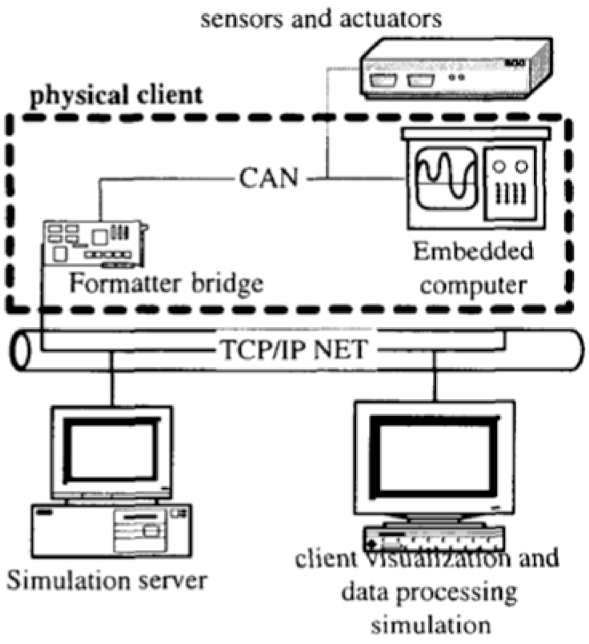


Figure 2: Physical client and the simulator

With the development of wireless technology and the rapid development of portable devices, wireless sensor networks (WSNs) play a key role in extending the smart grid implementation to residential premises and energy management applications [1]. Wireless sensor networks are the key elements in applications such as smart building, environment monitoring [17]. A wireless sensor that has a dedicated wireless module for wireless communication was created, as shown in Figure 3[41] and Figure 4[27]. The function of the wireless sensor node is to acquire existing sensor measurements and send them to the data storage receiver [41]. Because of the design of this sensor, it can only be used as a transmitter, and the storage can only be used as a receiver. In the application scenario where only data collection is performed, the wireless sensor based on this architecture can complete the work effectively. However, with the unified management of wireless resources, the use of such wireless transceivers is likely to face legal problems because of transmission power and frequency.

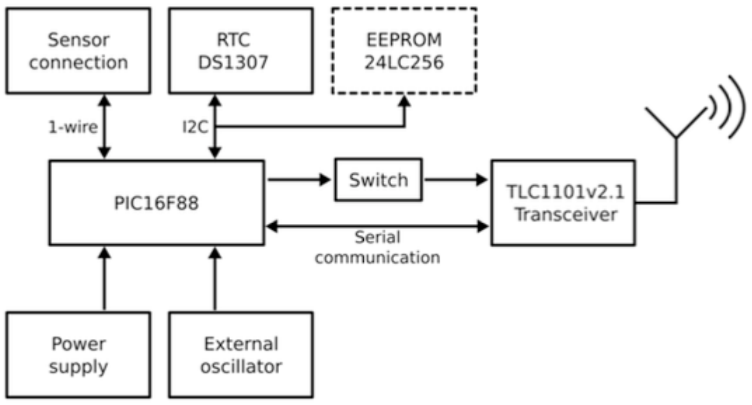


Figure 3: General structure of the wireless node

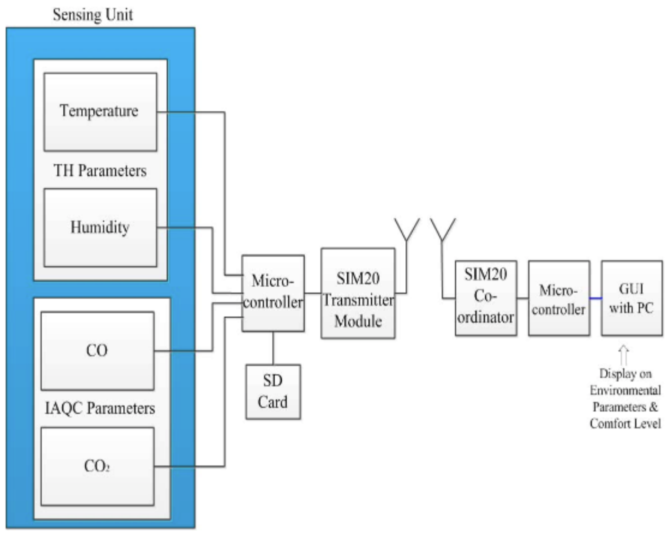


Figure 4: Block diagram of the developed smart comfort sensing system

With the invention of Bluetooth technology, a management system for human body tracking sensors is proposed. In the proposed system, the sensor nodes are connected to a central control unit with Bluetooth technology [11]. In this system, Bluetooth connects various sensors worn on the human body, and the data is sent via the wireless network, as shown in Figure 5[11]. This kind of design, when applied to design human body sensors, has clear advantages. The most prominent advantage of Bluetooth technology is low power consumption. At the same time, using mobile phones as data transmission devices can simplify the hardware structure. However, such systems cannot be applied to long-term environmental measurement independently due to the power limitation of Bluetooth communication and the instability of using mobile phones as central nodes. The most significant problem is that such system control devices cannot be interconnected, which will cause the control nodes to become independent of each other and increase management costs.

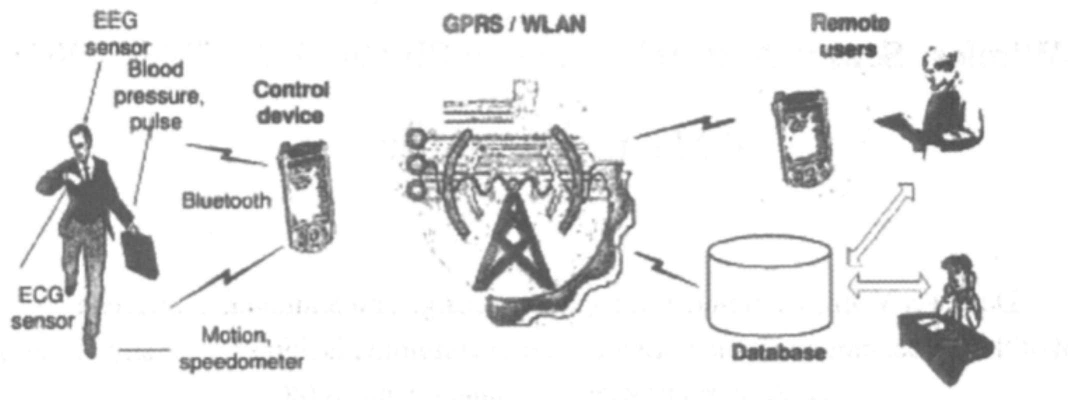


Figure 5: System architecture

Wireless sensor network nodes usually use batteries that cannot be replaced by fixed power supply [14]. With the popularity of RFID technology, a low-power sensor was designed that can obtain system power through a coil that meets the RFID standard and then measure environment factors. The most significant feature of this type of sensor is that it does not require an additional power supply, and it can be powered wirelessly by RFID scanning equipment. It can also complete a one-time measurement with limited power, as shown in Figure 6[60]. While this type of sensor cannot form a sensor network due to the lack of continuous power supply, the design idea of this kind of low-power sensor is still worth investigating.

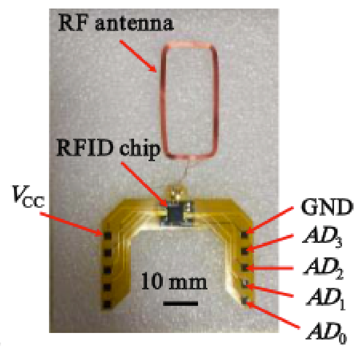


Figure 6: RFID chip being assembled on an FPCB with an antenna

Wi-Fi is another important part of the field of computer wireless communication, as shown in Figure 7[40], and it is gradually entering the design of sensor systems, especially when large amounts of data transmission are needed in the system. Furthermore, Wi-Fi has a longer communication distance but it requires more power consumption.

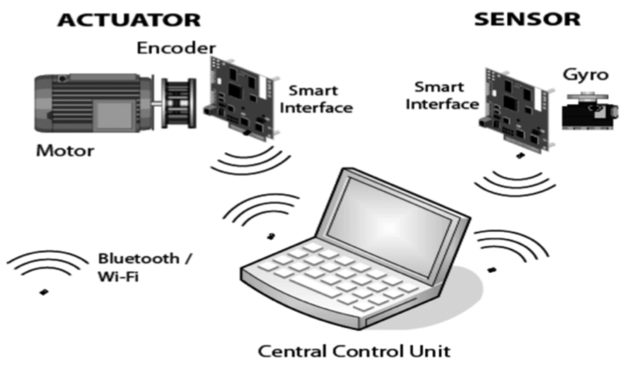


Figure 7: General application scenario of wireless sensor

In recent years, ZigBee technology has been widely used in wireless sensors. The complete ZigBee protocol stack includes the physical layer; media control layer, security layer, and a high-level application specification [14]. ZigBee has inherent advantages when used to build multiple sensor networks. However, ZigBee, Bluetooth, and Wi-Fi communication are all based on the 2.4 GHz frequency band. However, the 2.4 GHz frequency is very congested, and a large number of Wi-Fi signals and Bluetooth signals have almost run out of frequency bandwidth in the 2.4 GHz frequency. As a result, the interference between different signals has become a serious problem. Texas Instruments has introduced a BT/WLAN coexistence solution that is being used in a variety of industry products [35]. However, as there is no coexistence solution between Wi-Fi and ZigBee signals, ZigBee may face signal interference problems. Moreover, ZigBee can only carry very limited data, which complicates sending and receiving large amounts of data on the network.

As shown in Figure Figure 8[31][32] and Figure 9[30], to make the sensor network more widely applicable, it is necessary to split the sensor network from the physical configuration. A sensor network structure based on computer network protocols is proposed. This kind of connection method makes it possible to synchronize and manage data between nodes. The sensor network becomes a part of the standard computer network. With the current rapid development of computer networks, connecting nodes to the computer network will enhance the versatility of the system and reduce the difficulty of software development.

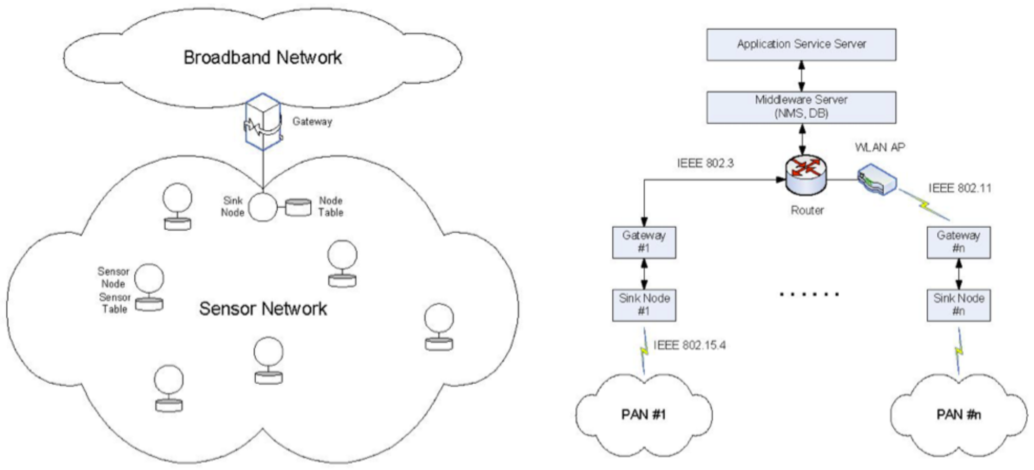


Figure 8: Sensor network

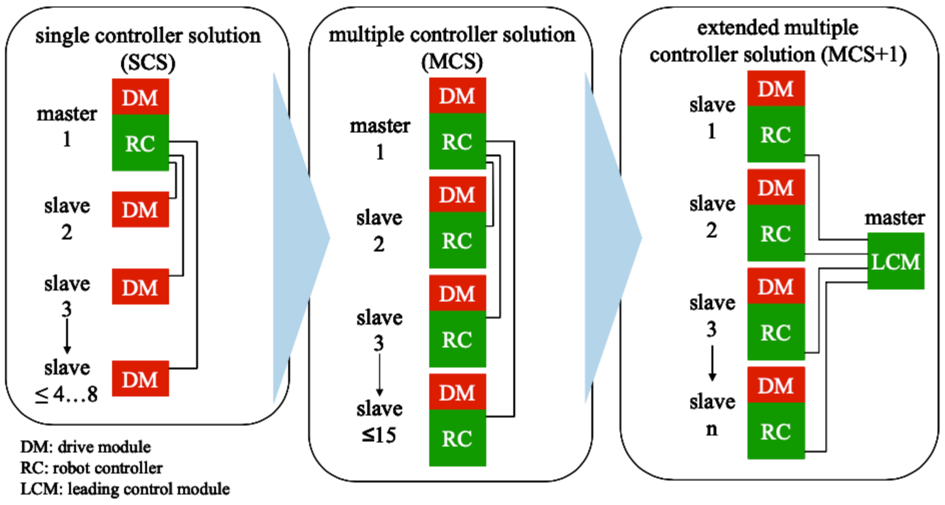


Figure 9: Hardware concept for a modular control system

Much work has been undertaken to develop standards for smart sensor communication, such as IEEE 1451 [44], as shown in Figure 10[44]. The IEEE 1451 standard is formulated to standardize and universalize smart sensor interfaces [13], which standardizes the integration of different sensors and computer networks. IEEE1451 does not stipulate the physical interface of communication. Therefore, a wired sensor that is compatible with the IEEE1451 protocol is connected to the computer system via USB in addition to being connected to the traditional computer network, as shown in Figure 11[6]. While some applications still need to use a wired connection, the USB has gradually become the preferred modern sensor besides serial connection due to its convenience and ease of connectivity [55].

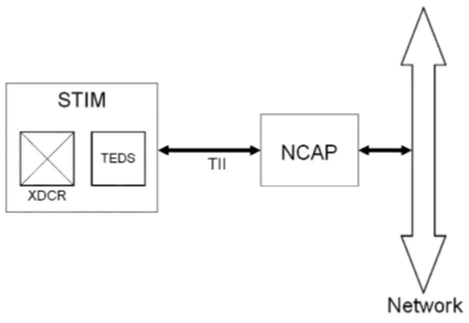


Figure 10: Block diagram of IEEE 1451 smart sensor

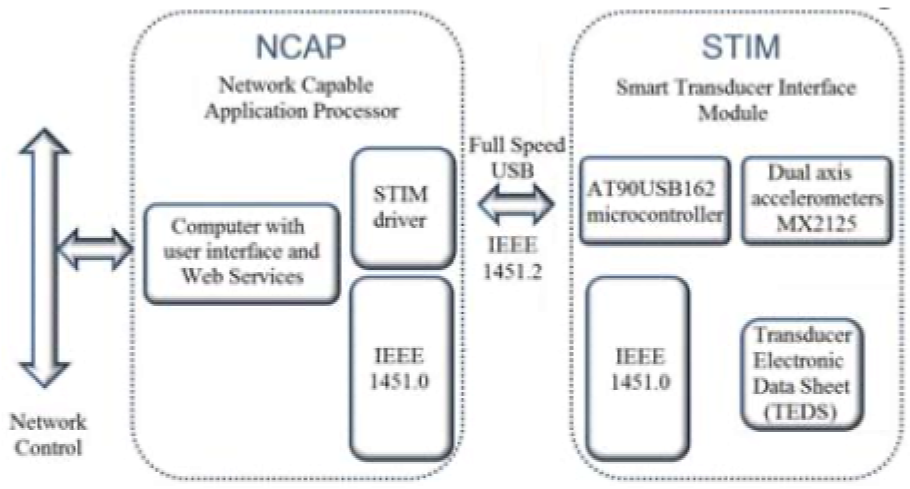

NCAP
Network Capable Application Processor
STIM driver
Computer with user interface and Web Services
IEEE 1451.0
Network Control
Full Speed USB
IEEE 1451.2
STIM
Smart Transducer Interface Module
AT90USB162 microcontroller
Dual axis accelerometers MX2125
IEEE 1451.0
Transducer Electronic Data Sheet (TEDS)

Figure 11: Block diagram with an electronic system and IEEE1451 architecture

New sensors are equipping major systems around us with unparalleled intelligence as in the case of smart grids, smart homes, and driverless vehicles [3]. Sensors need to collect data for extended periods and the demand for miniaturized sensors is increasing. However, because the power supply is limited, the question arises of how to reduce the power consumption of sensors, and a normally-off architecture sensor system is designed. A normally-off architecture for a low-power multisensor system is shown in Figure 12[18]. In this case, the sensor modules and microcontroller are normally-off and/or intermittent [18], as shown in Figure 12[18]. During the information collection interval, the system will cut off irrelevant components and reduce the system speed and then reactivate the system when needed, which will greatly extend the battery life of the system.

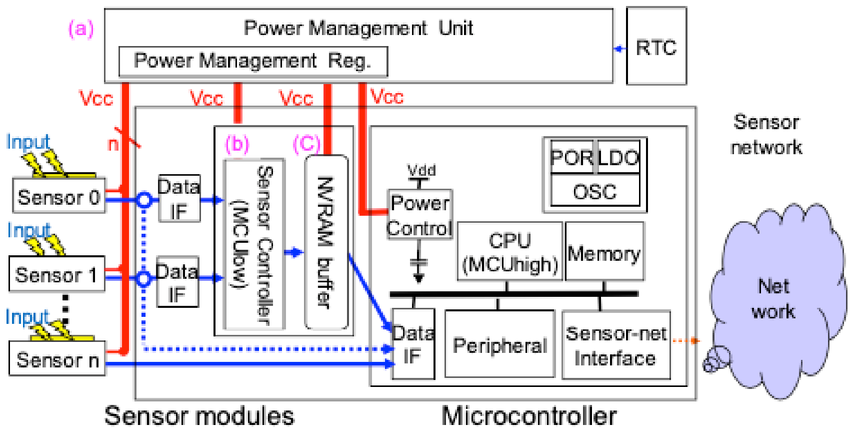


Figure 12: Normally-off architecture

In modern systems, the ability to reuse elements across multiple product offerings yields a competitive advantage in cost, configurability, capability, interoperability, and producibility to vendors who design for reuse [10]. Therefore, modular design is becoming increasingly popular because it allows one to reuse most of the software design and hardware design in the system and only modify the changed component parts, which reduces the cost, accelerates the development process, and shortens the development cycle.

2.2 Information Management System

In the daily work nowadays, a large number of data are produced, which is crucial to guide the safe production and management of coal [46]. The architecture of the information management system can generally be divided into front-end, back-end, and database, as shown in Figure 13[29]. The database is connected to the back-end, the user is connected to the front-end, and the front-end and back-end communicate to complete the information transfer, as shown in the figure.

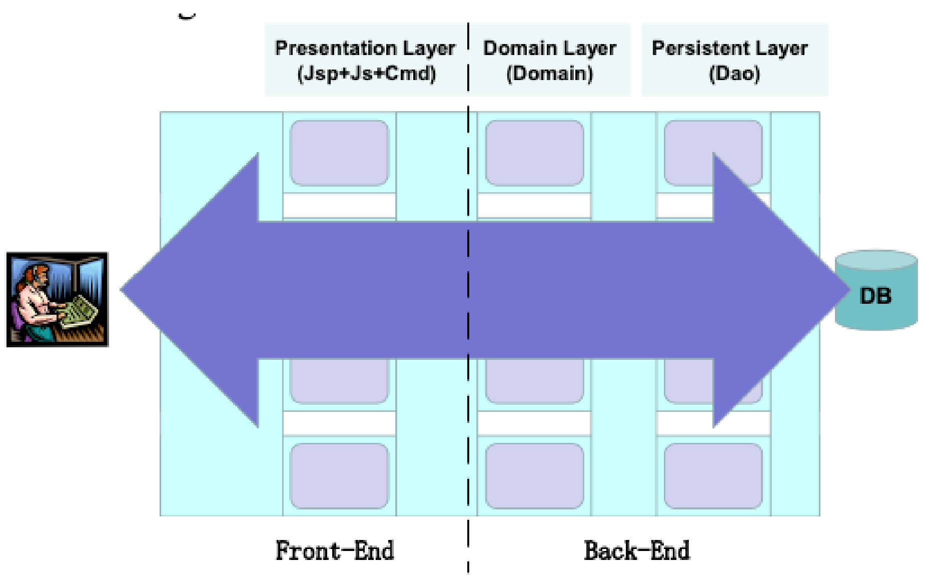


Figure 13: System Architecture of Front-end and Back-end Separation

In this architecture, it can be divided into the client/server mode design and the browser/server mode design. Because the C/S and B/S mode designs share the same server and database structure, only the front-end implementation is different. Therefore, an increasing number of systems tend to use a mixed-mode design, as shown in Figure 14[53]. The basic functions can be provided by the web page and the standalone client at the same time, and the advanced functions that the web page cannot complete can be completed by the standalone client. The service of the web page can be provided by the general server and coexist with the management system server.

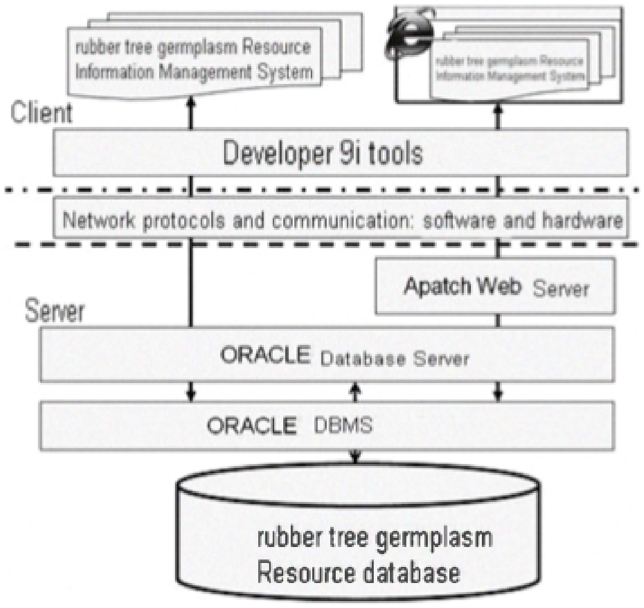

rubber tree germplasm Resource Information Management System
rubber tree germplasm Resource Information Management System
Client
Developer 9i tools
Network protocols and communication: software and hardware
Apatch Web Server
Server
ORACLE Database Server
ORACLE DBMS
rubber tree germplasm Resource database

Figure 14: B/S and C/S integrated framework diagram of information management system

The most important purpose of this type of client/web page + server design is to realize remote communication on the network and provide services to multiple clients with one server. The system includes a database server, WEB server, service logic server, and firewall [39]. The network structure is shown in Figure 15[39]. The client can be located in any network location, whether it is connecting over the Internet or LAN. Provided the connection can be established with the server, the communication can be completed. In actual applications, if the number of client requests is not substantial or the performance of the single server is somewhat high, the web services, databases, and logical services can be combined on the same server. Thus, there is only one server physically, but there are still three servers logically, and this configuration mode can simplify the complexity of physical device configuration and network configuration.

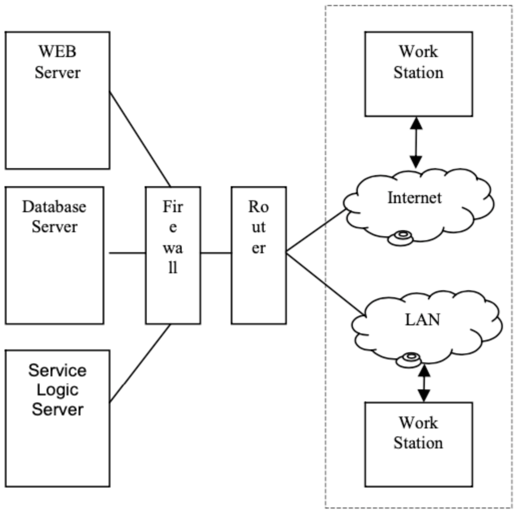


Figure 15: Hardware structure diagram of archives management information system

After the software architecture and server design have mature solutions, some information management systems combined with hardware are designed, one of which is GIS. GIS is a system that analyses and displays geographic information [33]. As shown in Figure 16[33][21], the hardware part of the GIS is the bus station and bus, and the software part is the information system, database, and client. Through the GPS signal system, information can be drawn and calculated on the satellite map. GIS enables a substantial amount of information to be retrieved from it by calculating the emergency response times, in the case of natural disasters, to the locating of suitable business locations [33]. A management system such as GIS simplifies measuring the environment on site. There is also an extended GIS based on the previous GIS, in which not only information is collected from the hardware, but the system can also control the hardware, and more system functions can be extended through the Internet. A type of information system also combines with the natural environment, such as disaster management systems, as shown in Figure 17[47]. As the complexity of the system increases, an increasing number of parts need to be authenticated when a user attempts to connect to the system. Therefore, the user's identity authentication process can be simplified by providing a single sign-on server. It certainly improves the quality of the system, and users can login to the system and pass the firewall as well as the proxy server to the single sign-on server [20].

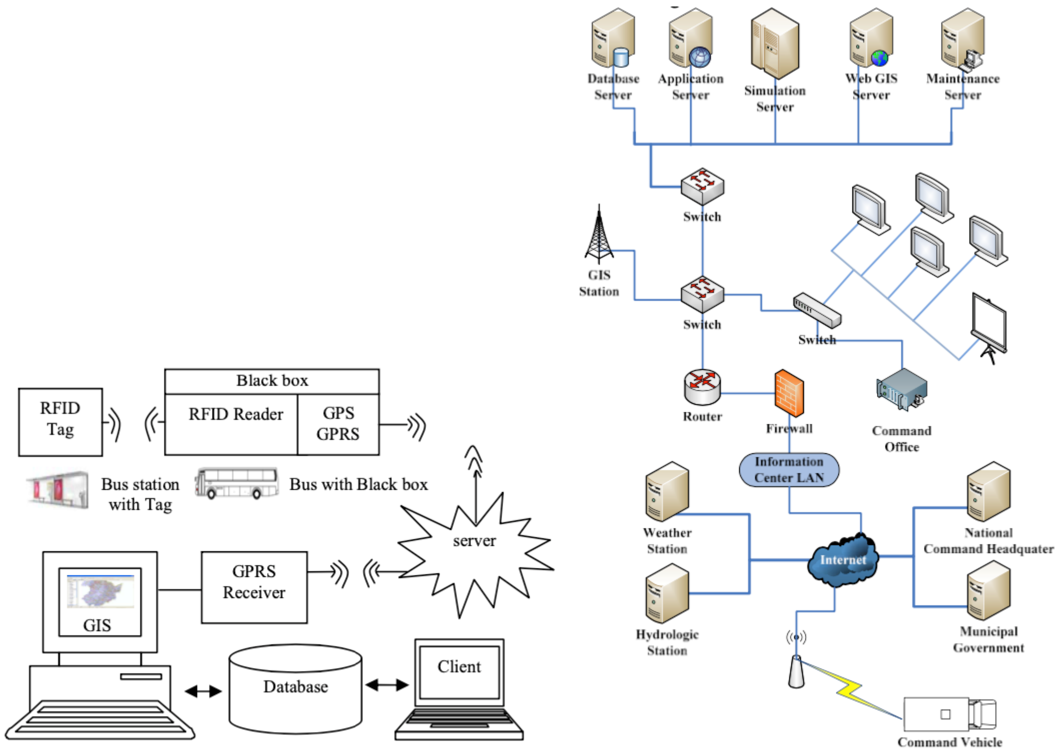


Figure 16: Architecture of an information system including hardware

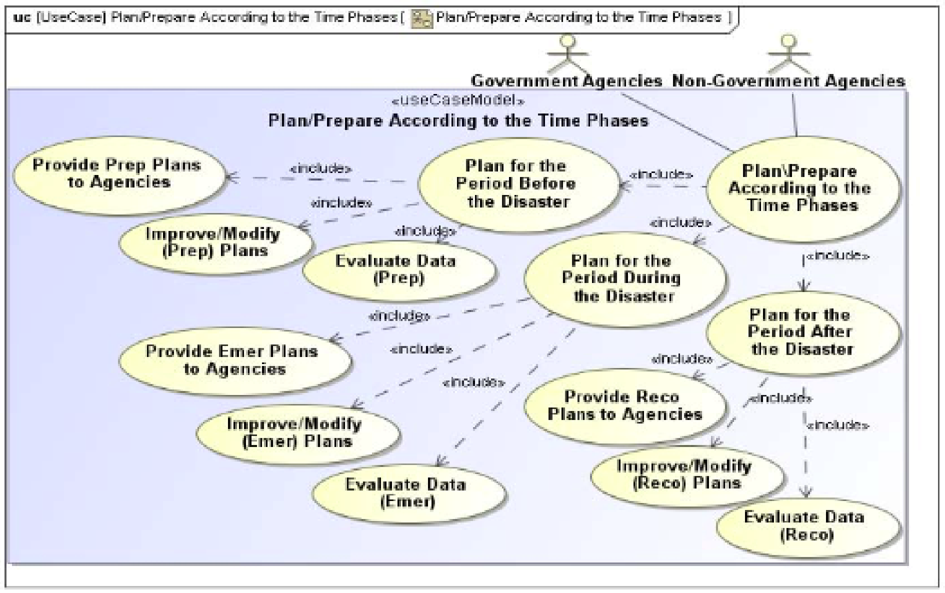

uc [UseCase] Plan/Prepare According to the Time Phases [Plan/Prepare According to the Time Phases]
Government Agencies
Non-Government Agencies
«useCaseModel»
Plan/Prepare According to the Time Phases
Provide Prep Plans to Agencies
Plan for the Period Before the Disaster
Plan\Prepare According to the Time Phases
Improve/Modify (Prep) Plans
Evaluate Data (Prep)
Plan for the Period During the Disaster
Plan for the Period After the Disaster
Provide Emer Plans to Agencies
Improve/Modify (Emer) Plans
Evaluate Data (Emer)
Provide Reco Plans to Agencies
Improve/Modify (Reco) Plans
Evaluate Data (Reco)
«include»

Figure 17: Lower Level Use Case Diagram of the DMS

A general belief across many engineering fields holds that the complexity of most social and technological systems has significantly increased in the last decade and the trend will continue in the coming years [19]. As the complexity of the system increases, the question of how to maintain the availability of the system must be solved. In such a server-centric information management system, once the server becomes unavailable, for example, due to a problem in the network, storage, and power, the entire system will stop responding and the data will not be able to upload, download, or process. The easiest way to avoid this situation is to increase the number of servers, as shown in Figure 18[45]. This kind of multi-server system design can enhance the robustness of the system, and all servers will check status between each other through the heartbeat response and other methods. The independent design of the database can prevent problems between data synchronization and multiple writes.

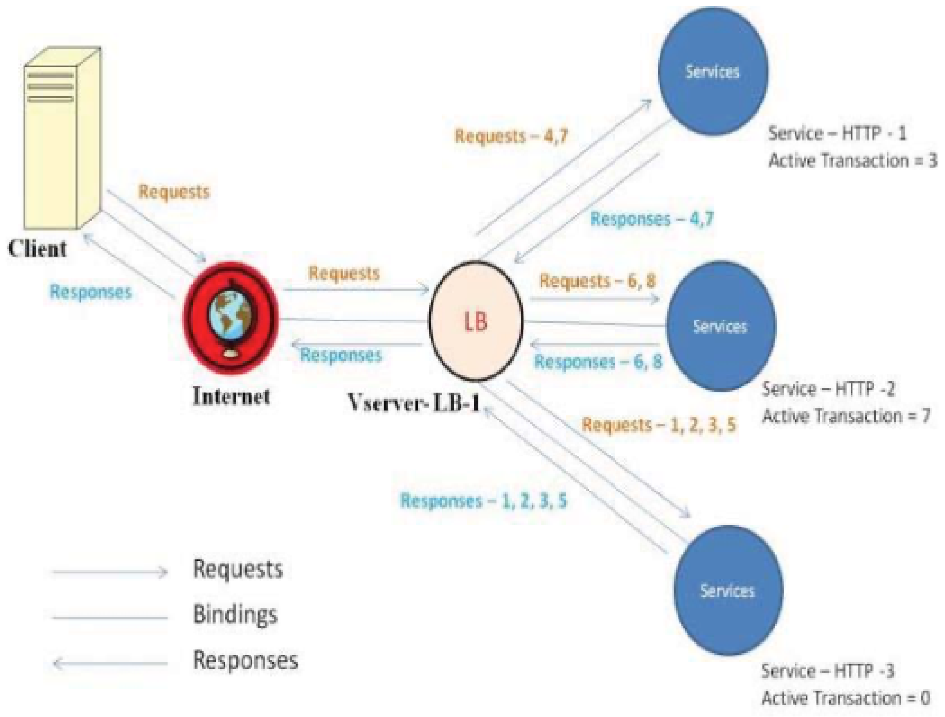


Figure 18: Architecture diagram of redundant service

As a key component of the information management system, the database is also very important. For data storage, the traditional SQL database or SQL database can be selected. Because the traditional SQL database is based on relationships, it has advantages when used to solve business logic problems. However, because of the need to resolve the relationship, SQL performance will encounter a bottleneck when dealing with a large number of records. As a result, the Not Only SQL database appeared. NOSQL records data based on files and the metadata of index files. Therefore, it has inherent advantages when large amounts of information are processed, as shown in Figure 19[42]. However, NOSQL faces a memory usage problem. When system power is lost, the volatility of memory may cause data loss on a NOSQL-based database. It can be observed in a performance comparison of MySQL and MongoDB, two mainstream databases for SQL and NOSQL. In some scenarios, MongoDB required less response time compared to MySQL.However, MySQL responses were stable compared to MongoDB. Therefore, choosing a better database for IoT depends on which query is used most often and the requirements of the application [42].

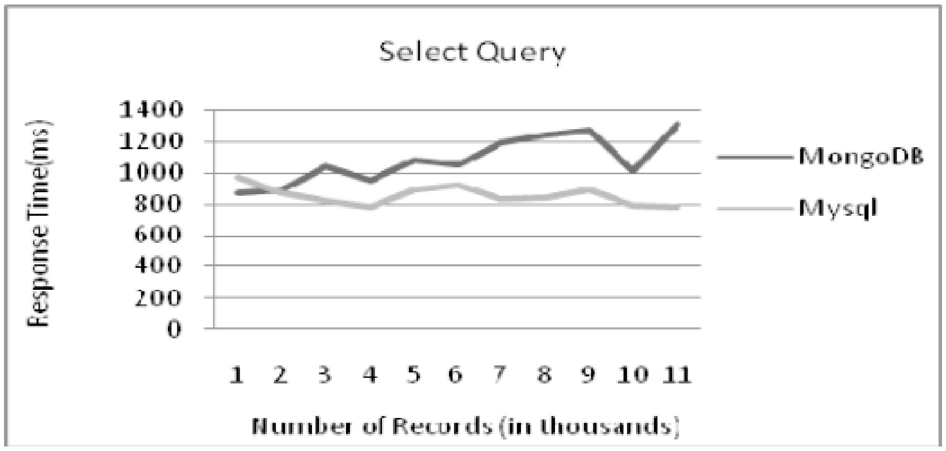


Figure 19: Select query vs number of records of two databases

2.3 System Security

The importance of system security has become increasingly prominent. System security mainly includes the following: resilience, security assurance, identity management, privacy data, and communication security [37], as shown in Figure 20[37]. Security is an important part of information system design and development [50].

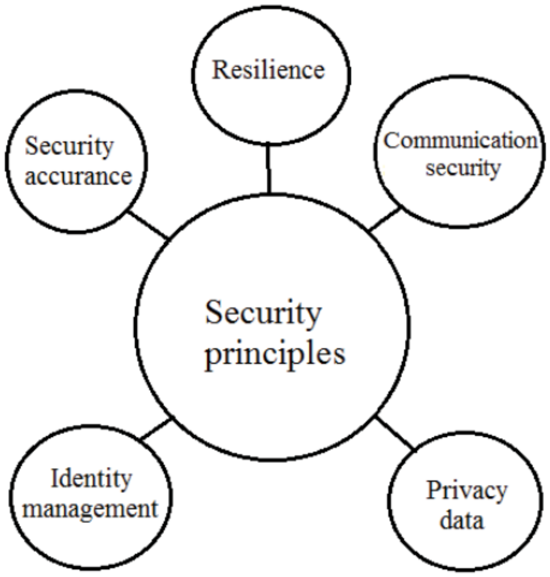
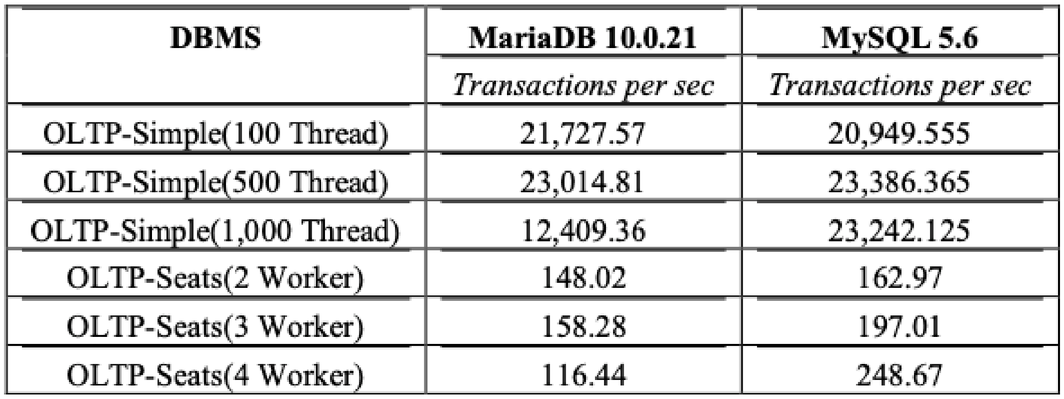


Figure 20: The key security principles

In SQL databases, MySQL and its branch, MariaDB, have been widely used. In a specific version, it can be seen that the performance of MariaDB is inferior to that of MySQL, as shown in Table 2[49]. However, MariaDB has relatively high security. It has full table encryption, which is not available in MySQL encryption. The database file can prevent the database from being read in plaintext when the database file is leaked. Moreover, MariaDB is fully compatible with MySQL syntax, so it is worth sacrificing some performance to enhance system security.

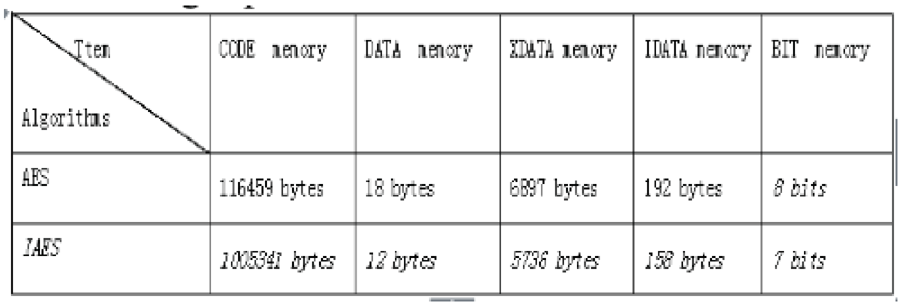
Table 2: Performance of two databases



DBMS	MariaDB 10.0.21	MySQL 5.6
	Transactions per sec	*Transactions per sec*
OLTP-Simple(100 Thread)	21,727.57	20,949.555
OLTP-Simple(500 Thread)	23,014.81	23,386.365
OLTP-Simple(1,000 Thread)	12,409.36	23,242.125
OLTP-Seats(2 Worker)	148.02	162.97
OLTP-Seats(3 Worker)	158.28	197.01
OLTP-Seats(4 Worker)	116.44	248.67

Because the information system needs to transmit data on the network, and because the data packets on the computer network are sent in the form of broadcast, to prevent the information from being intercepted and read by a third party client, it is necessary to encrypt the communication between the client and the server. The AES encryption algorithm has been widely applied. AES is a symmetric block cipher that can encrypt and decrypt information [2]. It is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits [2]. The AES algorithm places the data in a square matrix for transformation. Since each block has only 128 bits for transformation, both encryption and decryption will not cause heavy system load, and it has considerable advantages in CPU and memory consumption. In a comparison between AES and Improved AES, the AES calculation process only required an additional 18 bytes of data memory, as shown in Table 3[14], and the calculation process only requires an average of 400 CPU cycles on the X86 platform, as shown in Figure 21[14]. Thus, IAES can save more memory on software-based platforms. However, modern embedded controllers usually carry hardware-accelerated AES calculation modules. Therefore, choosing the AES algorithm on platforms that support hardware acceleration can save more CPU cycles and reduce calculation time and power consumption.

Table 3: Storage space overhead data tables



Item / Algorithms	CODE memory	DATA memory	XDATA memory	IDATA memory	BIT memory
AES	116459 bytes	18 bytes	6897 bytes	192 bytes	8 bits
IAES	*1005341 bytes*	*12 bytes*	*5736 bytes*	*158 bytes*	*7 bits*

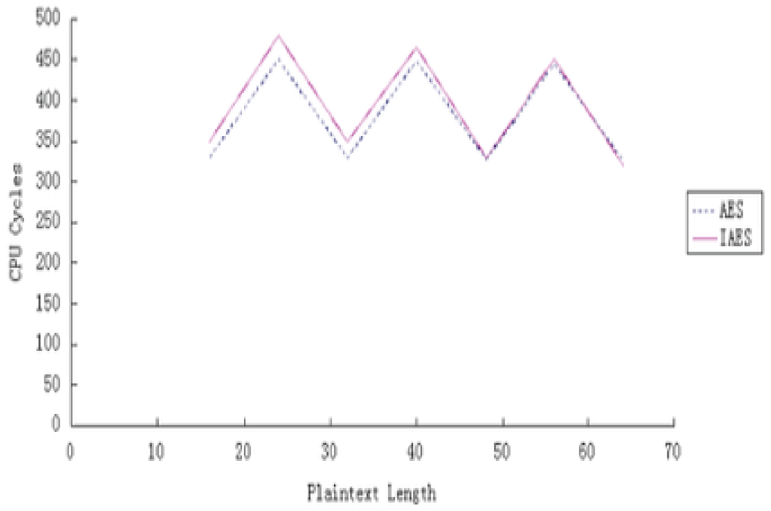
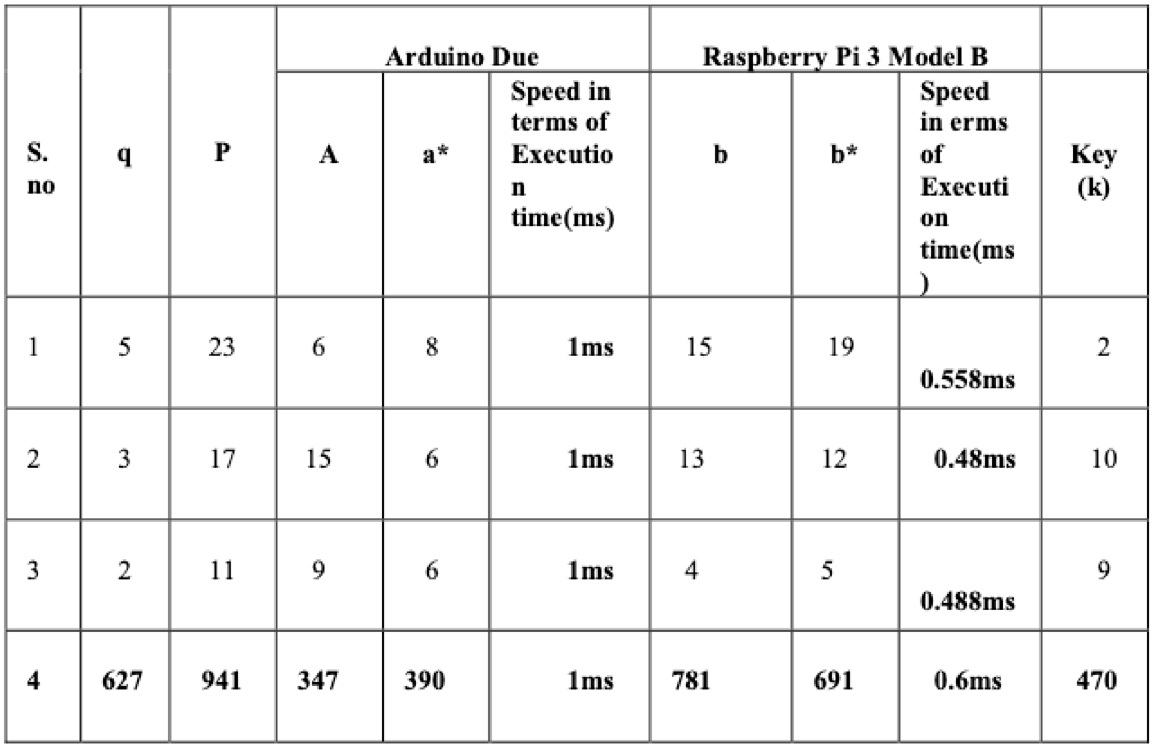


Figure 21: Comparison of CPU cycles by length

Symmetric encryption of data can only guarantee information security for a period, but as time passes and computing power increases, AES can eventually be cracked after a certain number of data packets are collected. When the AES key is cracked, all the data sent and received can be decoded in real time, and the encryption becomes ineffective. The best way to reduce information leakage is to change the AES password regularly. This requires the communication parties to have the ability to change keys and generate the same key so that the Diffie-Hellman (DH) protocol can be applied. The purpose of the DH protocol is to exchange a secret key between two users for subsequent encryption of messages [34], even if both parties do not have a predefined key. A performance report based on a test platform built on the ARM platform demonstrates that the exchange time can be as low as 1ms, as shown in Table 4[12]. Compared to the data transmission time on the Internet, the calculation time of 1ms is acceptable. Furthermore, the load of the system is also low.

Table 4: DH algorithm and its speed calculation



S. no	q	P	Arduino Due			Raspberry Pi 3 Model B			
			A	a*	Speed in terms of Execution time(ms)	b	b*	Speed in erms of Execution time(ms)	Key (k)
1	5	23	6	8	**1ms**	15	19	**0.558ms**	2
2	3	17	15	6	**1ms**	13	12	**0.48ms**	10
3	2	11	9	6	**1ms**	4	5	**0.488ms**	9
4	**627**	**941**	**347**	**390**	**1ms**	**781**	**691**	**0.6ms**	**470**

The dynamically changing key can reduce the probability of cracking of the key and reduce the amount of information leakage after the key is cracked. However, the key agreement process is defective as the key agreement process can be hijacked by a third party who can obtain information by establishing communication with the client and server at the same time. The client thinks that he has established communication with the server but is actually communicating with a third person, while the server thinks that it is communicating with the client and is actually communicating with the third person. Therefore, it is necessary to perform identity authentication on both parties before the key is negotiated, and the RSA algorithm can be used as such. The RSA public-key cryptosystem is widely used to provide security protocols and services in the network communication [51]. The RSA algorithm is a type of non-symmetric cryptographic algorithm depending on the factorization of large numbers proposed by three professors of the Massachusetts Institute of Technology: Ron Rivest, Adi Shamir, and Len Adleman [54].

After solving the communication security problem mathematically, to solve the problem of secure Web communication, Netscape proposed the SSL (Secure Socket Layer) Protocol in 1994, which ensures secure and confidential information transmission over the internet [15]. SSL integrates a variety of algorithms. As time progresses, some algorithms can no longer meet current security requirements. The SSL standard has also been revised many times to discard outdated algorithms and add new algorithm support.

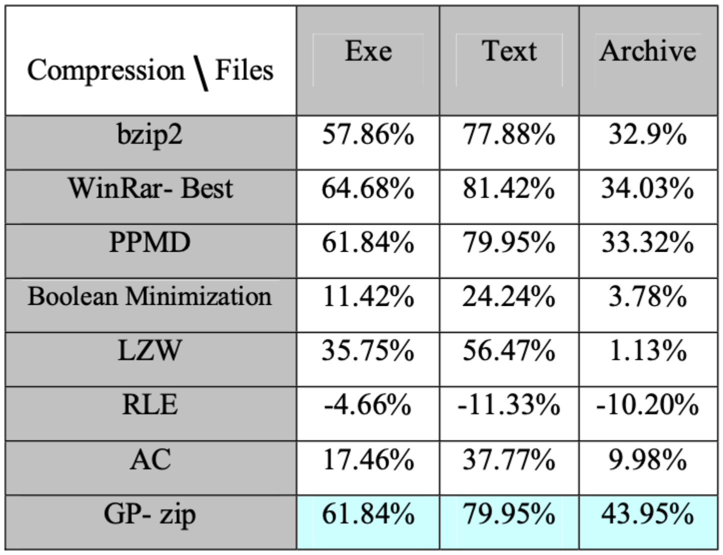
Users are a key part of information systems since the operation process inevitably requires user participation, for example, in the process management system. The workflow management system refers to the software systems that define, implement, and manage the operation of workflows. Interacting with the workflow executors, it facilitates the workflow operation and monitors the working status [52]. Users need to make different responses at different stages of the workflow in the system until they complete their work tasks. Different users can participate in different stages, and different users can query different work content. The information management system has a similar working mechanism, and it can be seen as having multiple tasks, while the content of the work that each user can perform is different. Therefore, the system needs to have different roles to isolate the users and the corresponding work in the system. After setting up different roles, access control needs to be introduced to enhance system security, associating entities (e.g., users or objects) in a system with permissions and allowing an entity to perform a certain action on another entity only if it has the necessary permissions [23].

Security is about deliberate, targeted, well-resourced threats, and it is no longer about opportunistic threats [26]. System security is defined as the ability of the interconnected system to provide electricity with the appropriate quality under normal and contingency conditions [16]. In addition to the system itself, it is also necessary to address attacks from outside the system. The most common form of threat is to attack vulnerable hosts through the network. Firewalls are crucial elements in network security and have been widely deployed in most businesses and institutions to secure private networks. The function of a firewall is to examine each incoming and outgoing packet and decide whether to accept or to discard the packet based on its policy [28]. Modern operating systems already include firewall applications, such as iptables with kernel support on Linux systems. Closing unused ports and blocking untrusted connections can enhance system security.

2.4 Data Processing

Whether it is a sensor node or an information management system, data needs to be processed. In the era of Big Data and the Internet of Things, devices ubiquitously sense and gather data at a rapid pace. Various methods have been proposed to speed up the analysis of the data and mining it for information [9]. With the development of computer software, many compression algorithms and software have been created, and different compression algorithms and software have different efficiencies, as shown in Table 5[25]. However, some compression algorithms cannot be applied to embedded platforms due to high memory consumption or heavy system load.

Table 5: Comparison of algorithms



Compression \ Files	Exe	Text	Archive
bzip2	57.86%	77.88%	32.9%
WinRar- Best	64.68%	81.42%	34.03%
PPMD	61.84%	79.95%	33.32%
Boolean Minimization	11.42%	24.24%	3.78%
LZW	35.75%	56.47%	1.13%
RLE	-4.66%	-11.33%	-10.20%
AC	17.46%	37.77%	9.98%
GP- zip	61.84%	79.95%	43.95%

Since the data may be damaged during transmission, it is necessary to perform digest processing on the data. By transmitting the checksum and the data together, the receiver can verify the integrity of the received data through the same digest algorithm after receiving the data. This prevents data damage or malicious tampering with data by a third party. Algorithms, which include MD5, SHA, and CRC, are widely applied in many systems today. Message digest 5 (MD5) is one of the message-digest algorithms developed by Ron Rivest in 1991. The MD5 algorithm takes an input data in any of length and produces an result in the form of a digest with a length of 128 bits [38]. SHA was developed by the National Institute of Standards and Technology (NIST) and published as a Federal Information Processing Standards (FIPS 180) in 1993 [38]. SHA-1 input can accept a maximum of less than 64 bits and 160 bits long output [38]. CRCs are based on polynomial arithmetic, base 2. CRC-32 is a 32-bit polynomial with several useful error-detection properties. It will detect all errors that span less than 32 contiguous bits within a packet and all 2-bit errors less than 2048 bits apart [48]. Many digest algorithms can successfully run on embedded systems. The Hash MD 5 algorithm can be implemented on an 8-bit microcontroller, with the result of 100%. It still has limitations, which include that the data can be processed to a maximum of 15 (fifteen) characters [4]. Although the MD5 algorithm has certain limitations on 8-bit systems, the core of the computing node of today's embedded systems is already based on 32-bit system architecture. Moreover, the core of the computing node of some embedded systems is even based on 64-bit architecture. As a result, the existing limitations on 8-bit systems are no longer a problem for modern embedded systems.

For a complete information system, how to protect data from physical hardware damage and data loss is also a problem to be addressed. This is especially in the case of embedded systems since no dedicated hardware provides data backup. Therefore, developers need to implement the backup function themselves, which gives developers the freedom to use different backup strategies, but it also increases the workload of development. Redundant data and multiple storages are always the core of the data backup process, so it is useful to refer to the redundancy strategy of computer disks, such as the most basic RAID-1 design, which is also called mirroring, as shown in Figure 22[24], and operations on Disk 1 will be executed on Disk 2 simultaneously.

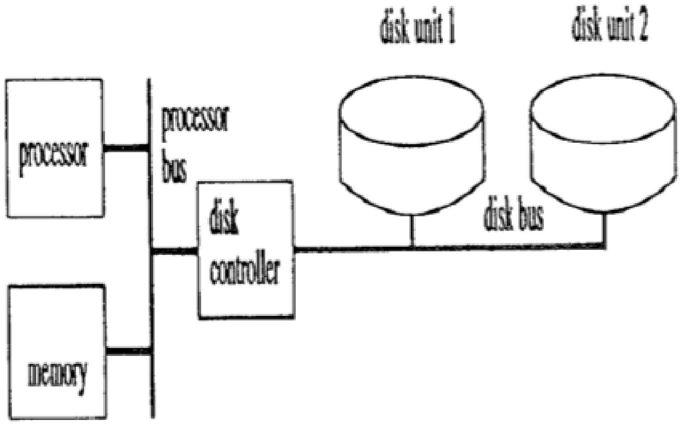


Figure 22: Model of a mirrored-disk system

In addition to the redundancy on the hardware, the backup and restoration of the software system is also an important part. A dedicated subsystem in the information system could perform data storage and backup, which is oriented to the application's database, business system, and core server and implements functions such as data storage, data backup and recovery, system backup and recovery, and application backup and recovery [59]. The system structure is shown in Figure 23[59].

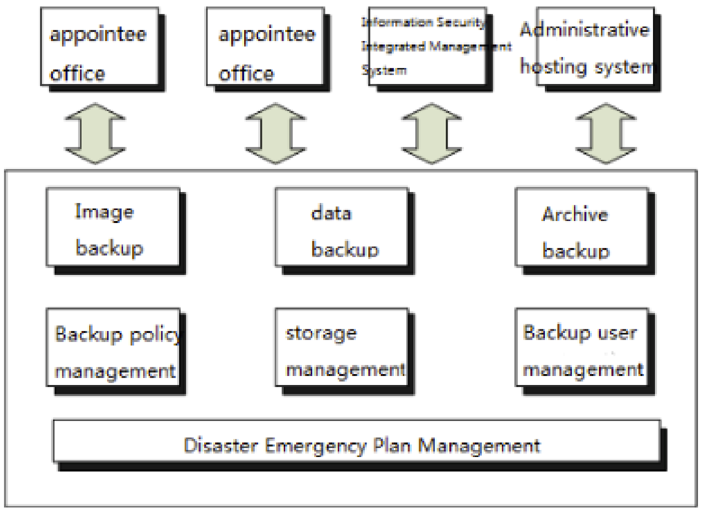


Figure 23: Data storage backup system function

2.5 Design Method

The design of the system and the logic behind it are different, which often determines system efficiency and user experience. To carry out the design process, it is necessary to study the essence of design. Design is an intelligent activity that begins with design requirements and ends with a product description [58]. The design process consists of proposing solutions to problems. In terms of logic, all problems have corresponding logical solutions. The purpose of design is to make an artifact that can perform the expected function [57].

An engineering system comprises the product structure, the environment structure, and the mutual relations between the product and the environment [56]. Therefore, in the case of system design, the system can be divided into elements. The attributes of the system can also exist as elements. The sensor information system is divided into two parts: hardware and software. The hardware includes controllers, memories, sensors, and peripherals. Each component has its specific properties, such as calculation performance, capacity, accuracy, and withstand voltage. One must select the appropriate components to achieve a balance between low cost and performance, which requires research on system application scenarios, such as indoors or outdoors, and the range of data changes, such as the upper and lower temperature limits, and the performance of the operating environment of the system deployment, such as CPU computing performance, and memory size. The success of industrial design and new product development rests on the effective management of the design-user interface [43]. Since a product will eventually be created for human use, it is necessary to simplify user operations and design simple and easy-to-use functions and interfaces.

2.6 Related Hardware, Software and Libraries

In this part, we introduce the related software, hardware, and libraries involved in the system design.

2.6.1 CISC and RISC

The 8086 CPU launched by Intel in 1978 opened a new era of X86 architecture, which has become the mainstream architecture in the personal computer field. It is the representative of CISC. Since CISC contains a large number of microinstructions, it is easier to generate assembly language code for the CISC platform. On the other hand, RISC provides limited microinstructions, which makes generating assembly language code more difficult but the execution speed of the processor can be greatly improved. ARM architecture is a typical RISC-based architecture. The RISC Project aims to explore alternatives to the general trend toward architectural complexity [36]. RISC makes it possible to use less chip area, increase the number of registers, reduce design cost, reduce system power consumption, and simplify hardware structure when implementing architecture through VLSI. Because RISC transfers complex instructions to simple instructions by software design, it can improve operating efficiency through optimized software code, while CISC integrates complex instructions into hardware, and it can only improve the execution efficiency of corresponding instructions by changing the hardware design.

2.6.2 ARM

Because of the excellent power performance ratio of the ARM architecture, it has gained a strong market in applications that require low power consumption, especially embedded systems. In the field of industrial control, ARM has launched an MCU dedicated to embedded industrial control. This kind of MCU provides multiple functions and control abilities, low power consumption, high-efficiency signal processing functionality, low cost, and easy-to-use advantages [5].

2.6.3 USB

The Universal Serial Bus(USB) is widely used in personal computers. Compared with traditional serial communication and parallel communication interfaces, USB has better anti-interference, a smaller physical interface, larger bandwidth, and other characteristics that are in line with modern communication. As personal computers increasingly abandon serial and parallel communication interfaces, a growing number of peripheral devices and scientific instruments use the USB interface as the main communication interface. In the case of the USB3.0 standard, the power supply capacity of the USB interface can reach 900 mA, and a computer with a charging port can provide a power output of 1.5 A, which is even higher. A single USB cable can meet the power supply and communication needs of most devices. The layers of USB are shown in Figure 24[8].

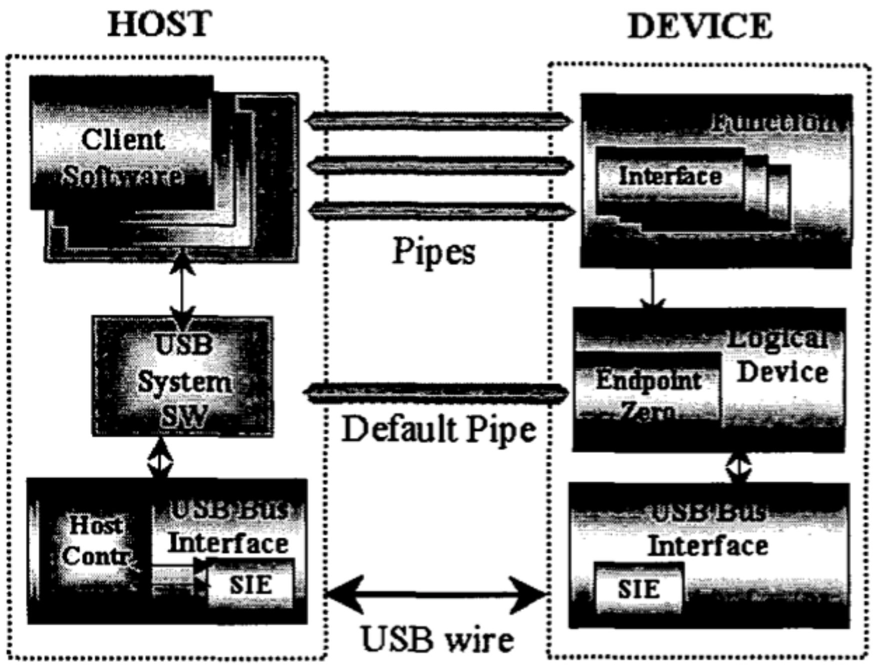


Figure 24: USB abstraction layers

2.6.4 pyDHE

pyDHE is an open-source, complete DH algorithm library based on Python implementation. By using this library, the security level can be enhanced by combined pyDHE with other algorithm libraries to meet the security requirements of the client under the C/S design model. This algorithm library reduces dependence on system modules in the development process and enhances the migration ability of the system.

2.6.5 QT

QT is a cross-platform GUI development framework based on the C++ language. On desktop operating systems such as Windows and MacOS, it usually has its own GUI development framework, such as .NET and UIKit. However, if the graphics-supporting environment on embedded Linux is provided by the traditional X Window environment, it will consume many system resources, and X Window relies heavily on graphics acceleration hardware. However, many embedded Linux systems do not include graphics acceleration hardware, and they do not even have standard display interfaces, such as VGA and HDMI. In such an environment, QT has become a good choice. QT includes a complete software renderer and a double buffering function. It can write to the display directly through the system kernel, which prevents reduced performance caused by intermediate layers, such as X Window and high memory usage.

2.6.6 MariaDB

MariaDB is a branch of MySQL. It is completely open-source and is maintained by the community. It can replace MySQL in most cases. In CentOS 7, MariaDB has replaced MySQL by default. MariaDB supports more storage engines. Some of these new engines can enhance compression capabilities, and some provide better caching. At the same time, MariaDB has a full table encryption function that MySQL does not have.

2.6.7 OpenSSL

It is very common to use OpenSSL to provide HTTPS services in B/S systems. Various applications can also provide security service for their own application communications by using the OpenSSL library. Self-signed certificates can also be generated through OpenSSL to simplify the process of system deployment. OpenSSL provides complete data encryption, data integrity check, security verification, and other functions. OpenSSL is an open-source and cross-platform library, which makes it applicable to all major operating systems.

2.7 Summary

In this chapter, we briefly discussed the software and hardware contents related to the system design. Furthermore, we referred to various sensor designs including single sensors and multi-node sensors. Sensors may be connected by wire or wireless.

Some wireless sensors have a dedicated wireless communication transceiver, while others are based on general wireless transceiver equipment, such as Bluetooth and Wi-Fi. To build a complete sensor system, we also address the design of the information management system. The purpose of the sensor is to collect data, and the purpose of the information management system is to deal with the data for processing and statistical filing, for example. Most modern management systems are no longer operated by a single person, which reflects the evolution of computers, from the initial single-person operation to multi-terminal sharing operation to multi-user operation to system sharing based on computer networks. The structure has become increasingly complex and the connection with people has become closer. After setting up the sensor network and information management system, it becomes increasingly important to protect the security of the system, and the data transmitted on the network during the communication process becomes increasingly important. Therefore, we refer to mathematical algorithms and attempt to solve problems, for example, how to combine different mathematical algorithms to prevent information and communication from being intercepted and other security issues, and then develop a plan to protect communications by referring to the existing security protocol standards. To solve the problem in the system design, we studied design theory, solved the design problem through logic-based thinking, divided all the components and attributes in the system into elements, analyzed the relationship between the elements, and derived a suitable design based on these works. Finally, we introduced the main related hardware, software, and libraries involved in the system. These components provide key support for the realization of the solution.

Chapter 3

Node System Hardware Design

3.1 Overall Architecture of Data Nodes

The overall system architecture takes the embedded system controller as the core and expands around the hardware resources provided by the controller. The overall system architecture design is shown in Figure 25.

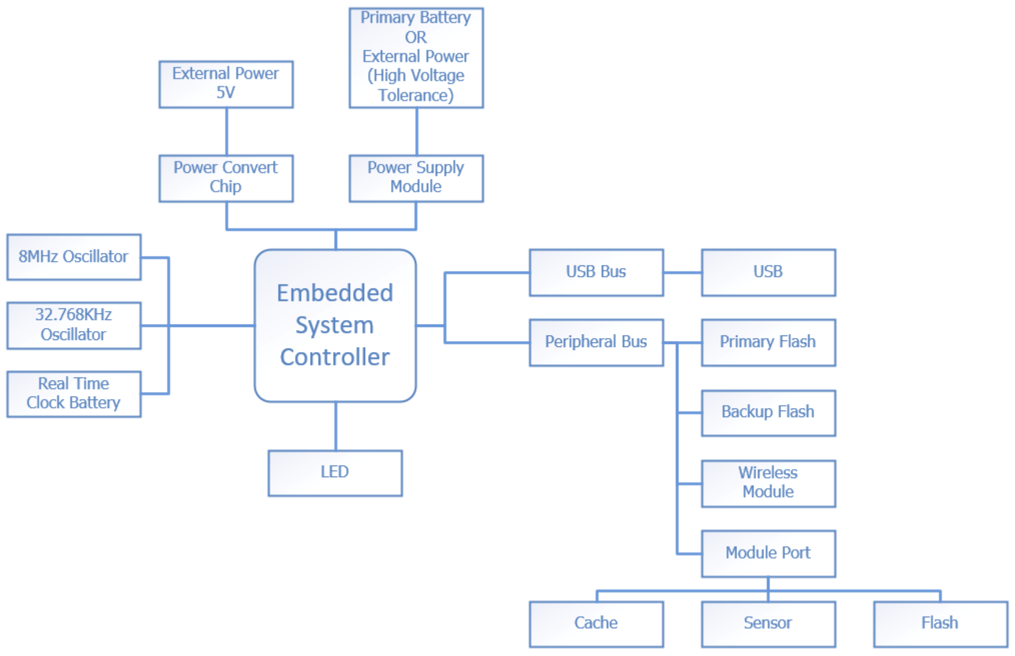


Figure 25: Overall system architecture design

3.2 Data Node Board Design

To meet the sensor size requirements in different scenarios, the overall design is divided into a standard version, mini version, and micro version. The standard version board size is 2135 mil * 1355 mil, the mini version board size is 1425 mil * 1355 mil, and the micro size board version is 1255 mil * 1355 mil.

The standard version is a full-featured design version, including the following: 1. External high-speed clock supporting circuit 2. External low-speed clock supporting circuit 3. USB circuit 4. External real-time clock battery 5. Primary flash memory 6. Backup flash memory 7. Wireless communication module 8. LED indication light 9. Power module interface 10. 5* Expansion module interfaces 11. External power supply/battery power supply switching circuit 12. Microcontroller and 13. Firmware upgrade interface.

Based on the standard version, the Mini version has removed the following parts: 1. USB circuit; 2. External real-time clock battery; 3. Backup flash memory; and 4. 2* Expansion module interfaces. Moreover, to ensure offline data collection and emergency data dumping, the following parts are added: 1. External Power interface 2. Serial communication interface.

The Micro version has simplified the following parts based on the Mini version: 1. Primary flash memory; 2. External power interface; 3. External power/battery power switching circuit; and 4.1* Expansion module interface. Comparisons are shown in Table 6.

Table 6: Comparisons between board types

	Standard version	Mini version	Micro version
External high-speed clock supporting circuit	Y	Y	Y
External low-speed clock supporting circuit	Y	Y	Y
Wireless communication module	Y	Y	Y
LED indication light	Y	Y	Y
Power module interface	Y	Y	Y
Battery monitor	Y	Y	Y
External / battery power switching circuit	Y	Y	
Micro controller	Y	Y	Y
Firmware upgrade interface	Y	Y	Y
Primary flash memory	Y	Y	
USB circuit	Y		
External real-time clock battery	Y		
Backup flash memory	Y		
Expansion module interfaces	5	3	2
Serial communication interface		Y	Y
External power interface		Y	

3.2.1 Board Type Functions Comparison

The standard version has all optional system functions. The on-board backup flash memory can ensure data backup requirements during offline collection. At the same time, it provides a large number of external interfaces for carrying various modules to meet complex functional requirements.

Although the mini version does not include the USB circuit, it has a serial communication interface and a standalone external power interface that can be used as a replacement for the USB interface to a certain extent. Normally, three expansion module interfaces are enough to satisfy the common usage scenarios. Because the

mini version lacks backup flash memory, data security cannot be guaranteed without an external flash module when primary flash memory fails.
The micro version removes all hardware designed for offline use, so the micro version is not suitable for long-term offline use.

3.2.2 Application Scenarios

The standard version can communicate directly with the computer at a high speed because of the complete USB circuit. The independent power supply circuit based on USB enables the standard version design to work with a computer for on-site measurement without the battery. Multiflash memory design satisfies the data security requirements during long-time recording, so it is suitable for application scenarios that have strict requirements for data storage. Because a large number of module interfaces are provided, stand version design can be equipped with multiple external modules at the same time, so it is suitable for situations that require multiple sets of data collection for fixed points. Real-time clock battery can provide continuous time recording support when the network cannot be accessed and the main power supply may be temporarily cut off, so it is suitable for scenarios that require time records under bad running conditions.
The mini version has a balance between function and cost. It has removed some functions prepared for running in bad conditions, but still providing an acceptable number of external interfaces. Therefore, it is more suitable for applications in conventional environments. The saved circuit space can be reserved for a Lithium battery to improve battery life or provide to the external module to expand system functions.
The micro version simplifies the functions prepared for offline use to save substantial board space. Therefore, the micro version can be equipped with a larger battery or reduce the product size to save costs. It is suitable for multiple fixed-point collection requirements and application scenarios that require limited types of information to collect.

3.3 External High-Speed Clock Supporting Circuit

According to the acceptable input of the microcontroller, the selectable frequency of the external high-speed clock source is between 4 MHz to 16 MHz. The microcontroller has an internal 8 MHz oscillator, but it can only guarantee a 1% difference precision at 25°C, so the accuracy is poor. The solution is to choose an external oscillator with the same frequency as the internal oscillator to provide a high-precision clock signal. For the oscillator to work normally, matching capacitors are needed so that the equivalent capacitance at both ends of the oscillator is equal to or close to the load capacitance, according to the calculation formula of parallel capacitance:

$$C = \frac{C_1 * C_2}{C_1 + C_2} \tag{1}$$

As the selected oscillator's internal capacitance is 10 pF, it can be calculated that C_1 and C_2 should each be 20 pF, as shown in Figure 26.
A parallel resistor on the oscillator side inverts the input by 180 degrees. The phase of the entire loop is 360 degrees to meet the oscillating phase condition. The signal is reversed through the resistor to drive the oscillator to work and limit the circuit current to avoid damaging the oscillator.

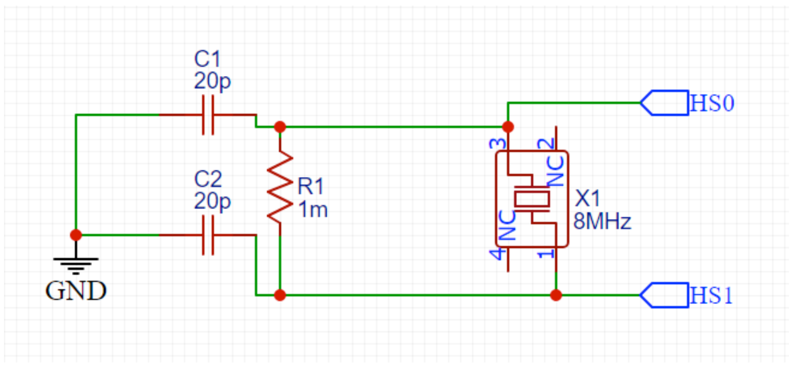


Figure 26: External high-speed clock supporting circuit

3.4 External Low-Speed Clock Supporting Circuit

According to the input of the microcontroller, the frequency of the external low-speed clock source must be 32.768 kHz. The microcontroller has an internal oscillator with a frequency of about 40 kHz. The difference in frequency will cause deviations in the timing of the real-time clock and the timing function. The solution is to choose the standard 32.768 kHz oscillator as an external oscillator to provide a high-precision low-speed clock signal. The external matching capacitor value should be 12 pF, according to the calculation formula of the parallel capacitor and the 6 pF internal capacitor of the selected oscillator, as shown in Figure 27.

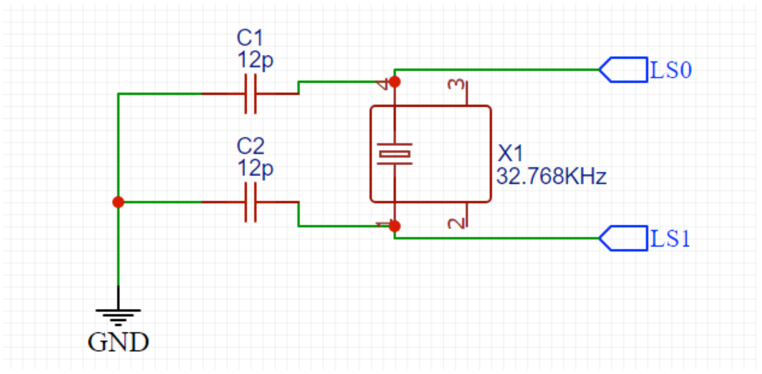


Figure 27: External low-speed clock supporting circuit

3.5 Clock Configuration

After having accurate high-speed and low-speed clock sources, the system clock needs to be configured to meet the operation requirements of various external devices. Different external devices work at different frequencies, so a frequency multiplier/divider device is needed to distribute the frequency to ensure the peripheral bus and devices can obtain the correct clock signal. The signal provided by the 8 MHz oscillator becomes the 72 MHz system clock after passing through the x9 frequency multiplier. After passing through the /1.5 frequency divider, it provides the 48 MHz clock signal for USB communication. The system clock passes through the /1 frequency divider and the /2 frequency divider, thus providing the clock signal to the peripheral bus 1. The system clock passes through the /1 frequency divider and the /1 frequency divider, providing the clock signal to the peripheral bus 2, as shown in Figure 28.

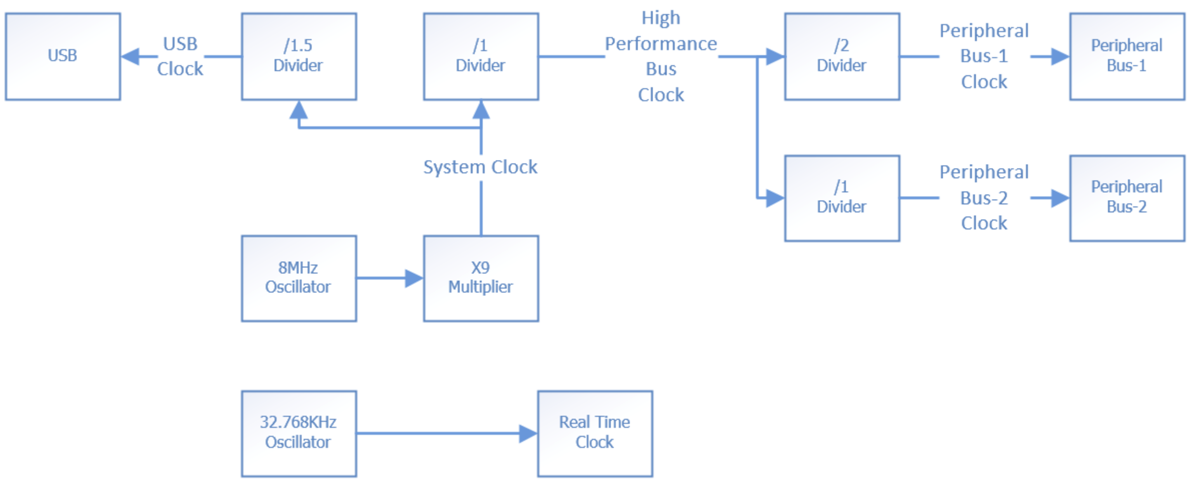


Figure 28: System clock configuration

3.6 Primary Flash Memory

The primary flash memory is used to save a certain amount of system data. The larger the capacity of the flash memory, the more information can be saved and the flash memory has a longer running life. As shown in Figure 29, the FLSCK, FLOI, and FLIO signals are required to operate the flash memory and are all provided by the main controller, but the FL1CS signal can be provided by a dedicated signal selection chip to save the interface resources of the microcontroller. The FL1CS signal is used to activate the flash memory chip. When the FL1CS signal is at a low level, the flash memory chip is activated. Note that there can only be one flash memory chip is active at the same time, as shown in Figure 30.

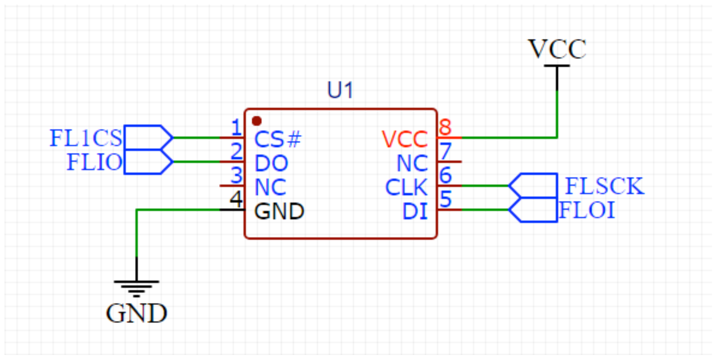


Figure 29: Primary flash memory

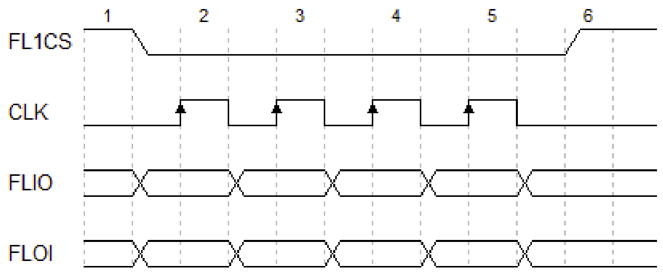


Figure 30: Timing of primary flash memory

3.7 Backup Flash Memory

The backup flash memory is used to save the backup data outside the primary flash memory to avoid data loss when the primary flash memory is accidentally damaged. As shown in Figure 31 and Figure 32, similar to the primary flash memory, the backup flash memory will be activated when the FL2CS signal is low and cannot be activated at the same time as FL1CS. It should be noted that although the main purpose of the backup flash memory is to ensure data security through redundant data, the backup flash memory can still be used as an additional memory without ensuring data security to increase storage capacity.

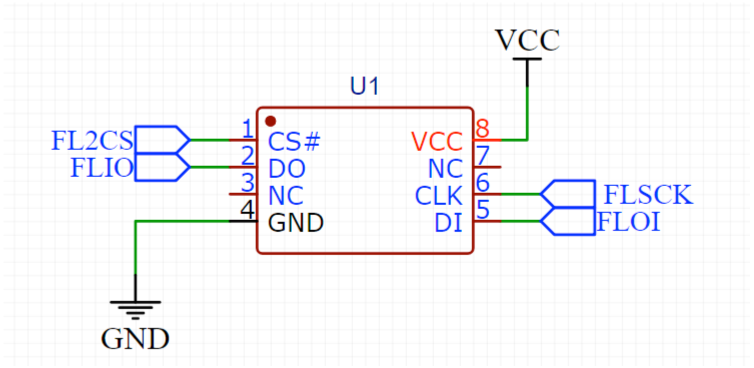


Figure 31: Backup flash memory

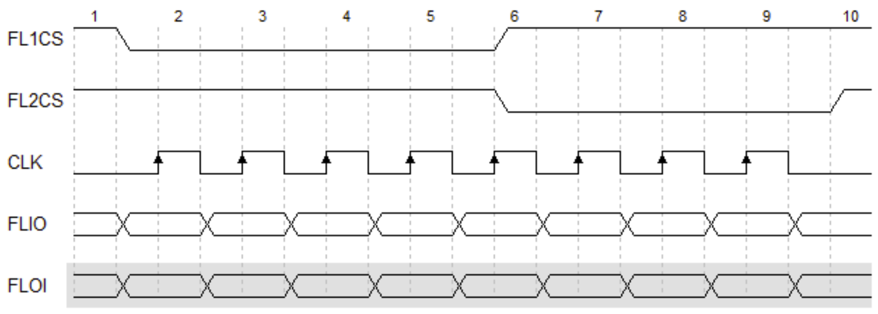


Figure 32: Timing of backup flash memory

3.8 Wireless Communication Module

The wireless communication module is a key component in the system and the hardware basis for wireless communication. It is selectable between Bluetooth-based or Wi-Fi-based modules to implement node communication function.

3.8.1 Bluetooth Communication Module

The advantage of the Bluetooth communication module is low power consumption. The low-power Bluetooth module used in this design actually measures only 0.3 μA@3.3 V offline current, the online low-power broadcasting current is 0.8 μA@3.3 V, online full-speed communication current is 1.23 mA@3.3 V, and it can be online for long enough to ensure the real-time communication requirement. However, the problem of Bluetooth is that the communication distance is short and the communication bandwidth is low. Therefore, when collecting data from the node, the user still needs to close to the node.

3.8.2 Wi-Fi Communication Module

The advantage of the Wi-Fi communication module is that the communication distance is long and the communication bandwidth is high. However, the most significant limitation is the power consumption of the module. The Wi-Fi module used in this design has a maximum startup current of 330 mA@5 V and a maximum continuous sending and receiving current of 112 mA@5 V. The online standby current can reach 72 mA@5 V, as shown in Figure 33, which prevents the node from being online for a long period when using the Wi-Fi module and the battery, otherwise, the limited lithium battery power supply may be exhausted rapidly. However, by setting up communication between Wi-Fi modules, a node network can be established, which makes it possible to access all other nodes in the network through a single node.

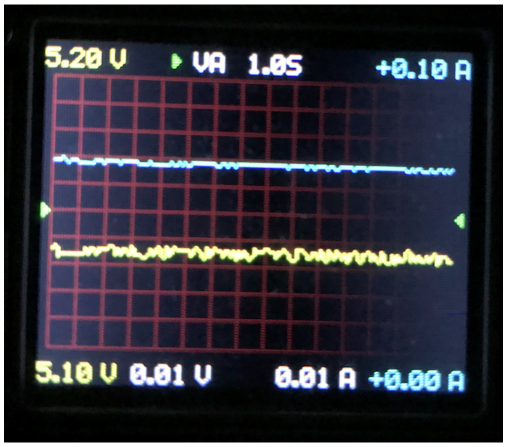


Figure 33: Standby current of Wi-Fi module

3.8.3 Cellular Communication Module

The power consumption of cellular communication modules is extremely high, so cellular communication modules cannot be mounted on low-power nodes, and it is of little practical significance to equip them together with low-power nodes. However, cellular communication makes it possible to connect nodes to the WAN wirelessly, which makes it possible to obtain node data far from the site where the node is installed.

3.9 LED Indicator

In consideration of low power consumption and space-saving design, not all versions of system design include a screen by default. Therefore, it is necessary to indicate the working status of the node through LED lights. The hardware design includes two LED indicators, one LED is directly connected to the power supply to indicate the power supply status, and another LED is controlled by the system, as shown in Figure 34. It can indicate the system status by constant lighting, blinking frequency, and turning off.

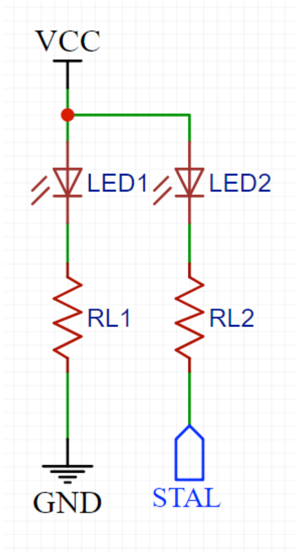


Figure 34: LED indicator circuit

3.10 Power Module Interface

The power module interface is a three-pin interface, and its main design purpose is to deal with different battery inputs. As shown in Figure 35, the negative pole of the battery is used as the ground and the positive pole of the battery is connected to the power module. The power module converts the battery input voltage to the system operating voltage to provide a stable voltage as the system power supply. Because the system is designed to take power from the power module and not directly through the battery, the system can also be used without a battery, as long as the power module can provide the system with the required working power voltage through the interface. When the battery is not used but the power is supplied through the power module, the first pin of the power module interface can be left floating.

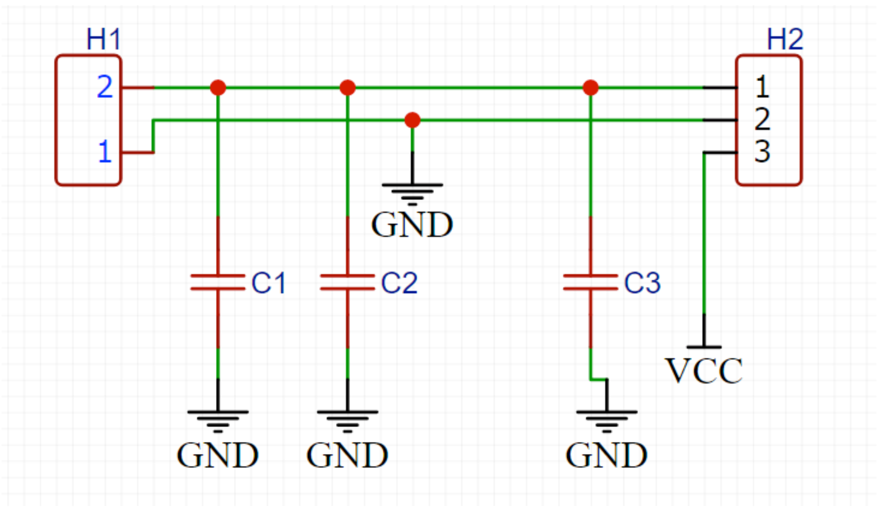


Figure 35: Battery and power module interface circuit

3.11 Battery Power Monitoring Circuit

When the battery is continuously discharged, the battery voltage will gradually decrease. As shown in Figure 36, by measuring the voltage across the battery, the remaining battery power can be roughly estimated. When the battery voltage is lower than the predetermined threshold, the battery power is considered exhausted, and the system will stop the measurement and data collection before the battery runs out. It will synchronize the collected data to the flash memory to prevent power loss that damages the flash or data loss. The battery status and battery voltage can be monitored through the analog-to-digital converter provided by the microcontroller and external circuit. It should be noted that the battery voltage should not be too high to prevent damage to the measurement circuit and the microcontroller.

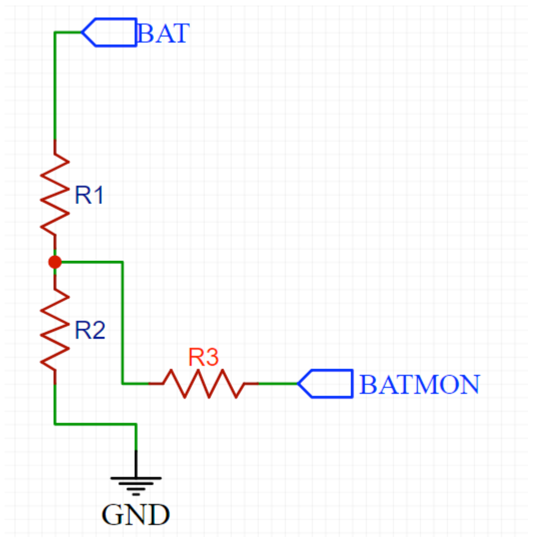


Figure 36: Battery power monitoring circuit

3.12 USB Circuit

The communication between the device and the host adopts the USB2.0 standard, and the device side uses a Micro USB interface to reduce space occupation. Because the device does not use the OTG function, the ID pin is directly grounded to make the device act as a slave, as shown in Figure 37. The design with USB enables the device to obtain power directly from the host site without the need to be equipped with batteries and power modules.

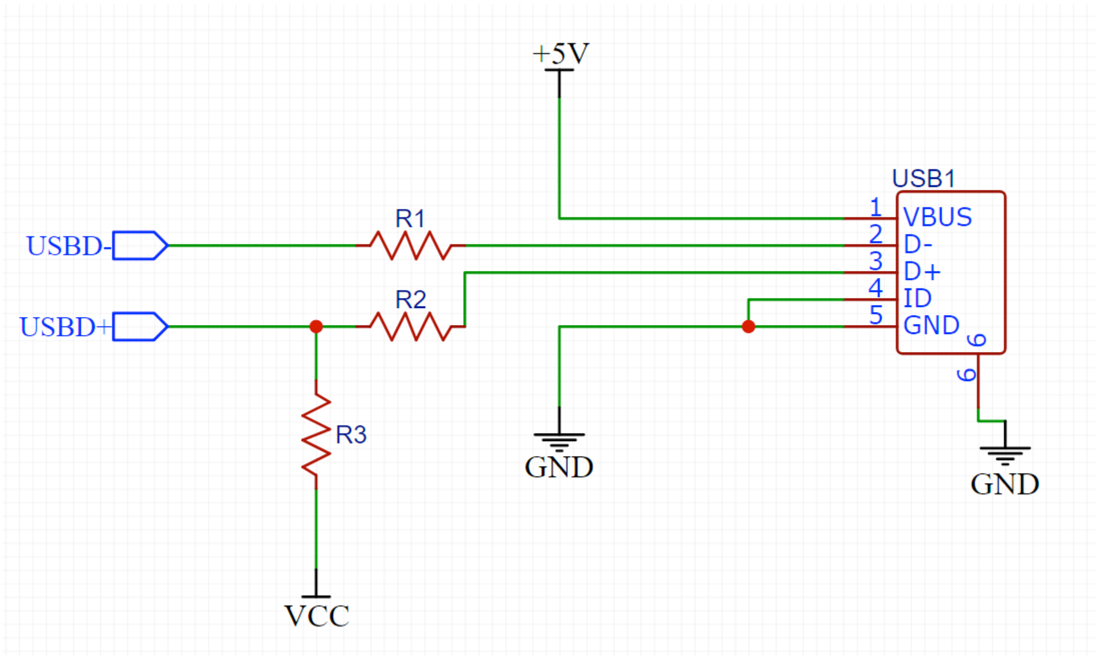


Figure 37: USB circuit

3.13 Expansion Module Interface

To realize the modular design, it is necessary to provide the expansion module interface. In the face of different needs, the nodes can be configured to meet the required functions by freely configuring the modules. The expansion module interface can supply power to the module. However, it should be noted that too many modules or modules with complex functions may consume excessive power and affect the battery life of the node.

3.13.1 Screen Module

Since the node does not contain a display screen, when a small amount of graphics and text information needs to be displayed, the information can be displayed by connecting the screen to the node through the expansion module interface. The modular designed screen can ensure that the work of the node will not be interrupted even if the screen is damaged, and it is relatively simple to replace the damaged screen.

3.13.2 Sensor Module

The most important purpose of the node is still to collect information, so the sensor module is an important part to help the system perceive the external environment. This includes the temperature sensor, humidity sensor, pressure sensor, light sensor, and sound sensor, which connect different sensors to the expansion module interface. Thus, the node can collect different kinds of environmental information. Sensor modules can be moved between different nodes to reduce unnecessary sensor module usage and, thus, reduce power consumption and cost.

3.13.3 Cache Module

Generally, the cache module is not necessary. Although the cache module is also a non-volatile memory, it usually has a small capacity, only 2KB-32KB, so it cannot be used to store a large amount of data. Because the erase-write cycle of the cache module is very high, it can be used to repeatedly erase and write data, especially when it needs to read in and write out data instantly. For example, when performing edge calculations, some iterative calculation algorithms may affect the data in the array and will perform a random rewrite. If the flash memory is used, repeated erasing will rapidly diminish the life of the flash memory. This problem can be avoided by adding cache modules.

3.13.4 Flash Memory Module

When the node's built-in flash memory still cannot meet the usage requirements, the flash memory modules can be connected to the node to save more data. However, the communication speed of the expansion module interface is much lower than that of the node's built-in flash memory. Therefore, if the data is collected in real time, with regard to performance requirements, the data operation should avoid reading and writing large amounts of data on the external flash memory module, because this will exhaust the bus resources, keep the bus busy and cause the system to fail to respond to other modules on the expansion module interface.

3.13.5 Detection Module

The detection module involves human-computer interaction, usually applied to human detection and on-site intrusion. It is necessary to equip the node with a detecting module if it is required to trigger recording based on certain situations, such as monitoring the environment changes at the on-site environment if the human activities took place, or the environment changes at the on-site environment after devices are triggered. However, it should be noted that the random trigger feature of the detecting module may increase the power consumption of the system and reduce battery life.

3.14 Control Node Hardware Design

The control node hardware design adopts the integration scheme based on the ARMCortex A series CPU as the core of the control node. The expansion circuit is equipped with a Wi-Fi module to communicate with the Wi-Fi module on the data node. The control node will be used as a control core of the data node domain.

3.15 Deployment Requirements

- When supplying power from the host through USB, it is necessary to consider the possible insufficient power supply of the host, especially when the node is connected to the host through the USB HUB as the current supply may be insufficient. Typically, the USB2.0 port should have a maximum current power supply capacity of 500 mA@ 5 V, which far exceeds the current required for continuous operation of the node, but it should be noted that when the Wi-Fi module is started, a peak current of up to 330 mA@5 V will be generated. The insufficient current supply may cause the node to fail to start.

- Due to the high power consumption of the Wi-Fi module, the module itself can generate a certain amount of heat. Although there is no need to install an active thermal fan, when the node is equipped together with temperature sensors, humidity sensors, and other sensors that are sensitive to the changes in the surrounding air environment, the Wi-Fi module should be connected through a flat cable and be kept as far as possible from the sensor module. At the same time, the shell of the node should use heat insulation material to separate the Wi-Fi module area and the sensor module area to prevent the heating of the Wi-Fi module, which may affect the measurement data.

Chapter 4

Node System Software Design

4.1 System Architecture

The system is composed of the following parts: data node, control node, network, server, online client, offline client, and database. Several data nodes and one control node form a node domain. Each node domain can only have one control node, and one data node can only belong to one node network. When the data node is far from the control node and cannot directly communicate with the control node, the remote data node can forward the data pack to the control node via established connections with other data nodes and allow other data nodes to forward the data pack to the control node. The control node can also send data to the remote data node by establishing connections with other data nodes. When the control node collects the data from the data node, the data pack will be temporarily stored in the control node. Once the control node can go online and connect to the remote server, the control node will upload the data to the server. The control node can connect with the server via Ethernet, Wi-Fi, or cellular networks. After receiving the data, the server will verify the integrity of the data and write the data into the database. The database runs as an independent system and is not affected by the server's running status. When the server and the database are running normally, the online client can obtain the information reported by the control node and send data to the node by communicating with the server. The sent data will be forwarded when the node network performs the next data synchronization. Some data nodes have direct communication ports, such as USB. These nodes can be used directly through offline clients. In this case, the nodes will be used as independent nodes and no longer need to connect to the control node. When the application environment does not allow wireless signals to prevent electromagnetic interference, the data node can be used in a wired connection. The offline client can also manage a node domain by accessing the control node. When the application environment does not allow data to pass through the Internet or LAN (to prevent information leakage), an offline node network can be constructed. The use of offline clients means that users must be located near the independent data nodes or control nodes. The architecture is shown in Figure 38.

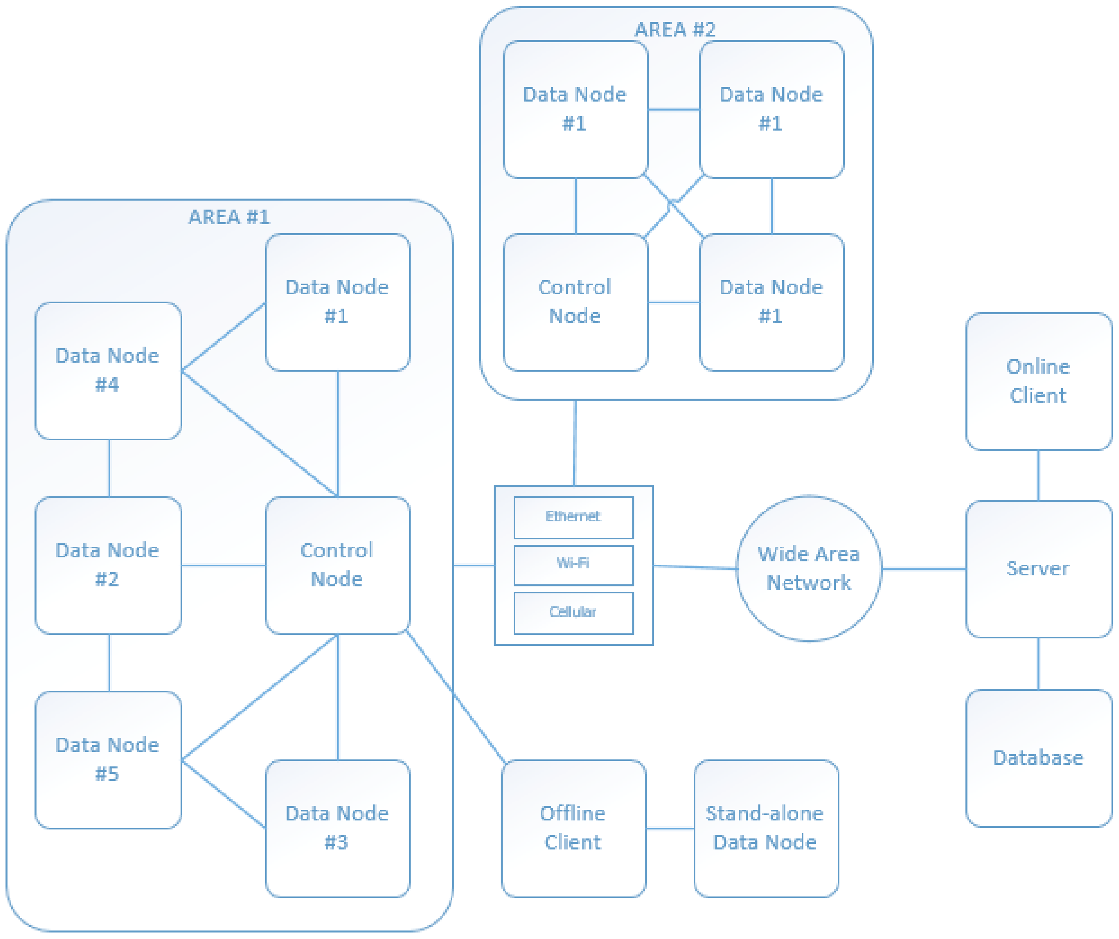


Figure 38: Architecture of system

4.2 Low-Power Design

4.2.1 Low-Power Design of Node

When a node collects environmental data through the sensor module, it does not need to continuously obtain data from the sensor module. The sensor itself often requires a certain period to update its status. Based on the requirements of the application, the interval of each information collection cycle may range from a few seconds to a few minutes. During this period, the node does not need to collect information, so the node can be placed in sleep mode during idle time to reduce the power consumption of the system and increase battery life. Depending on the interval time, the system can enter two different sleep modes to save power. When the data collection interval is less than two seconds, the node will enter the simple sleep mode. In this mode, only the power supply of the core in the microcontroller is cut off, and the peripheral devices will continue in normal running status. At this time, most of the system power is consumed by external peripheral devices, and a small part is consumed by the microcontroller on the node. The greater the number of external modules equipped, the greater the overall system power consumption will be. The core of the microcontroller will be awakened by the system clock or software interrupt at the next pre-defined timeout and enter the next working cycle. Since only the core needs to be awakened, the node can guarantee instant wake-up. The data-collection time point will not be missed because the wake-up time is short. The process is shown in Figure 39.

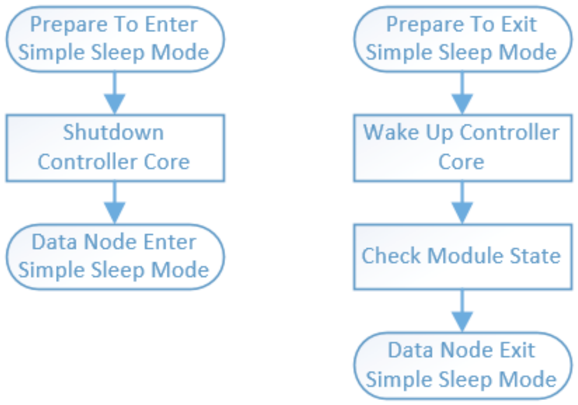

Prepare To Enter Simple Sleep Mode
Shutdown Controller Core
Data Node Enter Simple Sleep Mode
Prepare To Exit Simple Sleep Mode
Wake Up Controller Core
Check Module State
Data Node Exit Simple Sleep Mode

Figure 39: Process of enter sleep and wake

When the data collection interval is longer than two seconds, the node will attempt to enter the deep sleep mode. In this mode, the high-speed oscillator of the node will stop working, the system clock will stop, all software interrupts will not respond, and the system will only reserve a small amount of power supply to drive real-time clock and memory. Only part of the system area is powered, and the flash memory will synchronize information before deep sleep and enter sleep mode before the system enters deep sleep. Based on specific conditions and driver support and application requirements, external peripheral modules may enter sleep, partially enter sleep, or never enter sleep. The power supply of the external module will not be cut off. The node only sends a sleep command to the module to tell the module to enter the sleep mode. Whether to respond to the sleep command and how to enter sleep mode is determined by the external module. In the deep sleep mode, the power consumption of the node is extremely low; usually, the current of the system is only tens to hundreds of μA. The overall power consumption of the system is almost entirely determined by the power consumption of the external modules. The process is shown in Figure 40 and Figure 41.

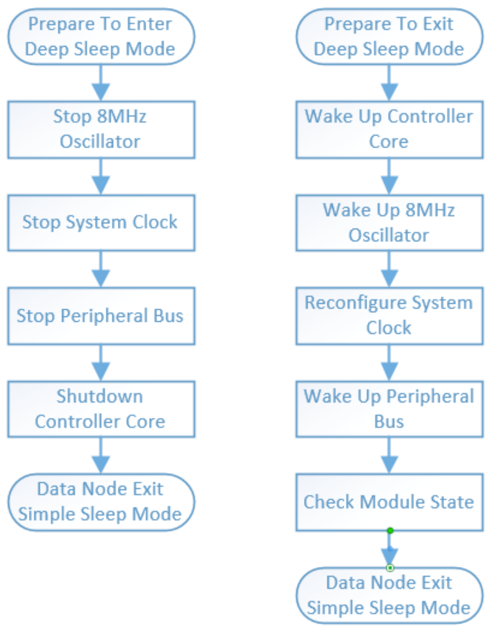


Figure 40: Process of enter deep sleep and wake

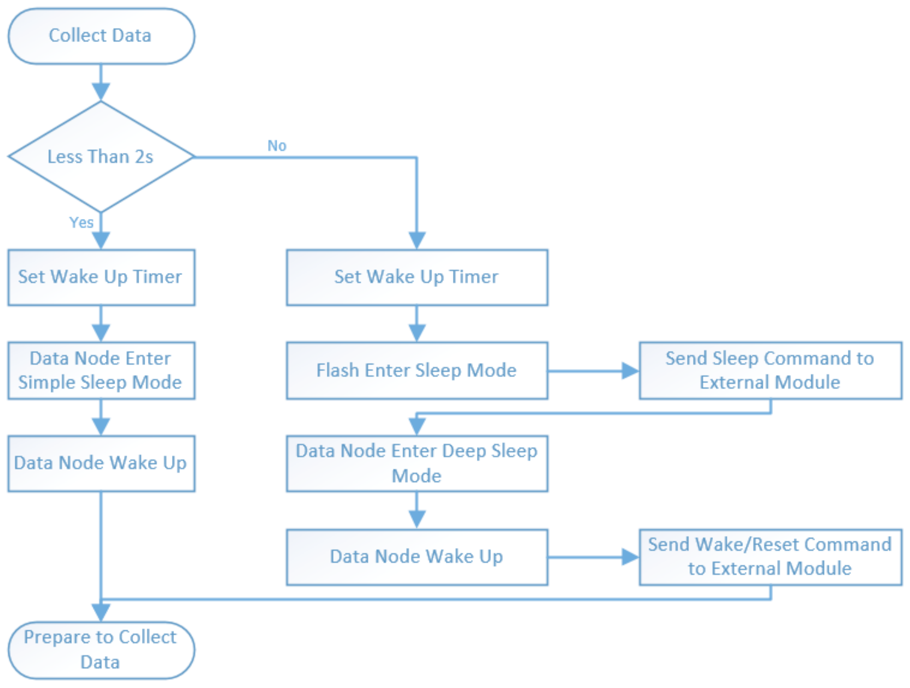

Collect Data
Less Than 2s
No
Yes
Set Wake Up Timer
Data Node Enter Simple Sleep Mode
Data Node Wake Up
Set Wake Up Timer
Flash Enter Sleep Mode
Send Sleep Command to External Module
Data Node Enter Deep Sleep Mode
Data Node Wake Up
Send Wake/Reset Command to External Module
Prepare to Collect Data

Figure 41: Process of working cycle

Since the deep sleep mode stops the high-speed clock and shuts down the communication with peripheral devices, it takes a certain amount of time to reactivate the system clock and wake up peripheral devices after the node system wakes up. To make the node enter the deep sleep mode, the node system will only enter the deep sleep mode when the data collection interval can provide sufficient wake-up time for the node to recover from the deep sleep mode. Otherwise, the long wake-up time will cause the data-collection time point to be missed.

The simple sleep mode is suitable for applications that have certain requirements for the collection interval. Compared with the normal operation mode, the simple sleep mode cannot reduce the power consumption significantly. Therefore, it is recommended to extend the data collection interval in common application scenarios to trigger the microcontroller to place the system into deep sleep mode and reduce the power consumption.

4.2.2 Low-Power Design of Bluetooth Module

Since the introduction of Bluetooth 4.0, which supports Bluetooth low-energy broadcasting, the power consumption of Bluetooth, therefore, is reduced. To reduce the power consumption of the Bluetooth module, it is only required to reduce the Bluetooth broadcasting and scanning frequency and extend the interval by changing the firmware configuration. Based on this feature, and the fact that the establishment of Bluetooth communication requires completing the pairing process between two devices, a triggered broadcast method can be designed. Since the node knows the device address of the offline communication client (as the client needs to be paired with the node, the offline communication client frequently broadcasts the client information while waiting for the device. When the node finds the offline client in one of the scanning periods, the node will broadcast the node information immediately, and then the offline communication client can discover the node and connect to the node. After the node disconnects from the offline communication client, if the node cannot detect the broadcast from the offline client within a specific period, the scan interval of the node will be extended until the scan interval of half a minute is reached. The Bluetooth module will be put into a sleep state to save power consumption during the scan interval. The process is shown in Figure 42.

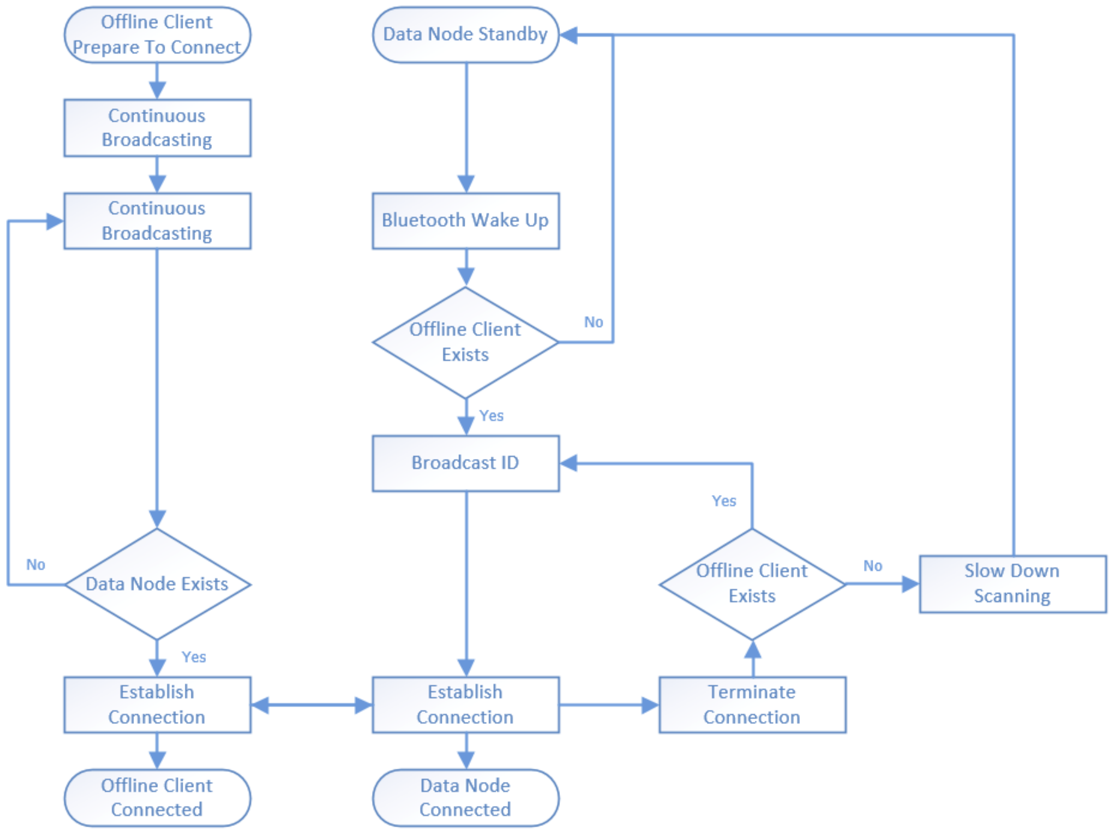


Figure 42: Process of low-power Bluetooth

4.2.3 Low-Power Design of Wi-Fi Module

The Wi-Fi module is designed to create a node network. Due to the high power consumption of Wi-Fi communication, the module cannot be powered up for a long period when the power supply is limited. However, Wi-Fi can form a large area network. The node can be connected to a remote server to avoid using offline clients through the node network, and the Wi-Fi module can be activated or deactivated by the microcontroller. The data node will first synchronize the current network time from the control node and then turn off the Wi-Fi module after calculating the next synchronization time point of the node network according to the pre-defined synchronization period. At the next synchronization time point, all data nodes will start the Wi-Fi module at the same time and attempt to communicate with nearby data nodes and establish connections. Regardless of whether the data is sent, once the pre-defined synchronization period expires, the microcontroller will immediately shut down the Wi-Fi module, and the data that has not been sent will wait until the next synchronization period of the node network. Since Wi-Fi communication can send and receive a large amount of data at a time, the interval of the synchronization period can be extended to save additional power. The process is shown in Figure 43.

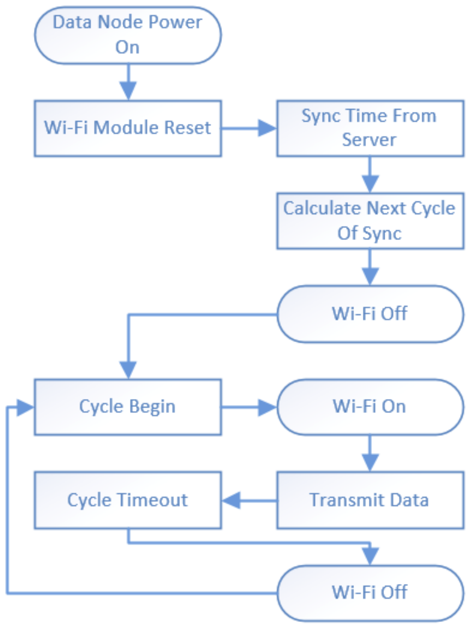


Figure 43: Process of low-power Wi-Fi

4.3 Data Storage

4.3.1 Data Range and Accuracy

In this demo application scenario, each report contains four items: timestamp, ambient temperature, ambient pressure, and ambient humidity. The timestamp is recorded in seconds, the content is the number of seconds from 1970-01-01 to the recording time point, and the environmental information is provided by the sensor. The data ranges are shown in Table 7.

Table 7: Ranges of data

Timestamp	0 - 4294967296 s
Ambient temperature	-40 - +85 °C
Ambient pressure	300 - 1100 hPa
Ambient humidity	0-100 %RH

The timestamp is provided by the real-time clock, so the accuracy of the timestamp is not in question. The absolute accuracies of the environmental information are shown in Table 8.

Table 8: Accuracies of data

Ambient temperature	±0.5°C @ 25°C, ±1°C @ 65°C
Ambient pressure	±1 hPa @ (300-1100 hPa & 0-65°C)
Ambient humidity	±3 %RH @ 20-80 %RH

The effective decimal place can be determined from the environmental information. The timestamp does not include milliseconds, so there is no decimal place. The environmental data resolutions are shown in Table 9.

Table 9: Resolutions of data

Ambient temperature	0.01°C
Ambient pressure	0.0018 hPa
Ambient humidity	0.008 %RH

The above data indicates that the longest integer and decimal places of each data are as shown in Table 10.

Table 10: Digits of data

Parameters	Longest integer	Longest decimal places	Signed number
Timestamp	10ˆ{10}	0	False
Ambient temperature	2	2	True
Ambient pressure	4	4	False
Ambient humidity	3	3	False

4.3.2 Data Type and Length

According to the precision and signed or unsigned of the number, the data types can be defined as presented in Table 11.

Table 11: Types of data

Timestamp	unsigned int
Ambient temperature	double
Ambient pressure	double
Ambient humidity	double

Although using float to define decimals can also meet the requirements of the longest decimal places, when fetching data from the sensor, a transformation calculation of the data is needed. In the transformation process, a variable that is defined as double is required in the calculation to improve the accuracy of the reference value, so the final data saved in memory, which has decimals, should be defined as double type. Based on this, it can be calculated that the size of one piece of full record saved in the memory should be 28 bytes.

4.3.3 Data Compression

Flash memory designed for use in embedded systems usually has a very limited capacity. The optional capacity is generally from 512KB to 32MB. To store as many records as possible, the raw data needs to be compressed. However, due to the limitation of low power consumption, the computing performance of the system is usually low, and the RAM of the system is very limited. In this demo application, only 20KB of RAM is available in the entire system. Limited RAM makes it is impossible to store a large amount of data in the RAM at any given time. The writing characteristic of the flash memory causes the flash memory to undergo a low-to-high transition when modifying data and the flash memory needs to be erased in 4KB units. Therefore, to extend the life of the flash memory and reduce the context calculation time during erasing, one must avoid excessive erasing and writing. Thus, the data compression method needs to meet the following conditions: 1. The compressed data should always be organized in fixed lengths. 2. The compressed data length should be a multiple of 4096 bytes. 3. The algorithm should be as simple as possible to reduce CPU instruction execution time when compressing the raw data. Based on the above requirements, the following compression method can be designed, as shown in Figure 44: For decimal numbers, it can be divided into bits and use four bits to represent a number, so one byte can store two numbers. The higher the ratio of the number of decimal places to be represented to the number of integers plus the total number of decimal places, the higher the compression ratio. However, this compression method is limited by the total number of digits. When the total number of stored integers plus decimal places reaches eight, the data cannot be compressed further.

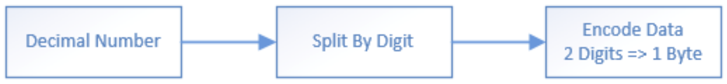


Figure 44: Process of data encoding

The elapsed time may increase to the maximum value of unsigned integers, so all data needs to be retained. The range of the ambient temperature value is from -45.00 to +85.00. After the compression method is applied, the data length will be fixed into two bytes. Because the temperature is a signed number, an extra byte will be occupied for saving symbols, and the total data length is three bytes. The range of ambient pressure value is from 300.0000 to 1100.0000, the raw data has the same data length, regardless of whether it is stored in raw or compressed form, so the data can be stored directly. The range of ambient humidity value is from 0.000 to 100.000. After the compression method is applied, the data length will be fixed into three bytes. After data compression, the total length of each record will occupy 14 bytes, as shown in Figure 45, which is two bytes shorter than the raw data in the write-out buffer. The compression rate is 87.5% and the total length of each record is 14 bytes shorter than the raw data in memory, so the compression rate is 50%. When the data increases, the compression rate may increase.

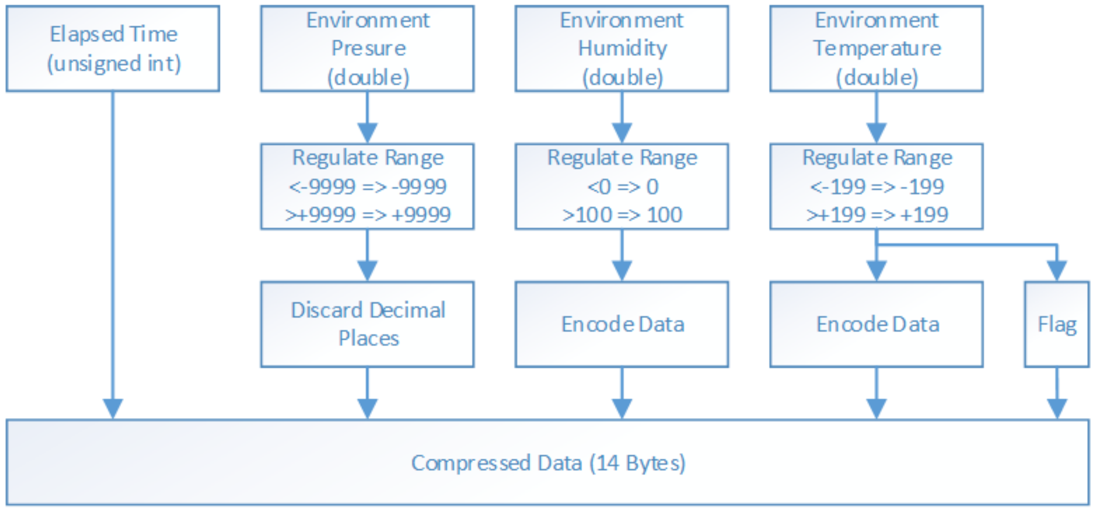

Elapsed Time (unsigned int)
Environment Presure (double)
Environment Humidity (double)
Environment Temperature (double)
Regulate Range <-9999 => -9999 >+9999 => +9999
Regulate Range <0 => 0 >100 => 100
Regulate Range <-199 => -199 >+199 => +199
Discard Decimal Places
Encode Data
Encode Data
Flag
Compressed Data (14 Bytes)

Figure 45: Process of data compression

The advantages of this data compression method include a fast execution speed and the fact that it requires little RAM. During the compression process, only eight bytes or 16 bytes of additional temporary space are required to calculate each record. Since the temporary space can be reused, the compression process ultimately requires only

8 bytes or 16 bytes temporary space, regardless of how many values or records are processed. It should be noted that the memory space can be dynamically allocated according to the actual length of the data because the allocation and destruction of the memory space require additional CPU cycles to wait for memory finishing tasks. However, as there are restrictions on dynamic memory management of the embedded system, the general solution is to allocate a fixed length of the memory space and to reserve this memory space for continuous compression throughout the entire hardware-running period, which will reduce the waiting time and increase the stability of the system. In terms of instruction execution of the microcontroller, there are no complicated mathematical calculations, only data splicing and displacement is involved, there are few CPU instructions, the execution cycle is small, and the running speed is fast. Therefore, it is especially suitable for hardware platforms that have a limited power supply and limited computing performance.

4.3.4 Data Cache

Data caching can be used to partially expand the memory during edge computing, for example, to save calculation results during matrix calculation, and to temporarily store the data to be calculated that was received from the management client and server. Since the chip used in the data cache can be modified in sections, there is no need to erase beforehand when modifying the data cache, which can reduce the waiting time and speed up the calculation compared to that of flash memory. The data cache chip has a long erase-rewrite cycle, about 1-10 million cycles, so there is no need for concern about the damage caused by repeated erasing and rewriting of the chip. When a new node is being initialized or an old node needs to be replaced, the data cache can also be used to mirror or transfer system settings. This solves the special application scenario in which the node needs to be set up when it cannot connect with online or offline clients or provide configuration information required for initialization when batch settings are required on new nodes. The data cache can also be used to urgently export a small section of the flash memory of a node. When the communication port of the node is damaged, some key data can be quickly extracted in this way.

4.3.5 Flash Memory Partition and Read Write Strategy

To make the node go online and automatically build the node network, the node needs a certain amount of memory space to save the configuration information that is needed when joining the node network. Regardless of the total capacity of the flash memory, the system will always reserve the last 4096 bytes of the flash memory area to save the system configuration. As shown in Figure 46, when the system has multiple built-in flash memories, the last 4096 bytes of each built-in flash memory will be used to save the same configuration information to prevent flash memory damage, which will prevent a situation in which the node cannot join an online network due to missing configurations.

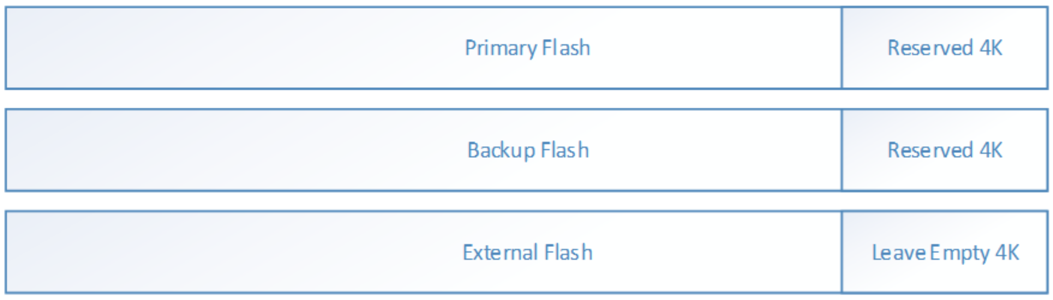


Figure 46: Reserved area for flash

Generally, the flash memory can be erased and written only 10,000 to 100,000 times per unit. Therefore, it is necessary to balance the writing as far as possible to prevent the overwriting of and damage to a certain storage unit. Since data records are written gradually in strips, one by one, the cycle recording method is used to balance the erasing and writing cycle of each storage unit as far as possible. When the memory is full, the oldest unit will be cleared, and the latest data will be overwritten to the cleared unit, as shown in Figure 47.

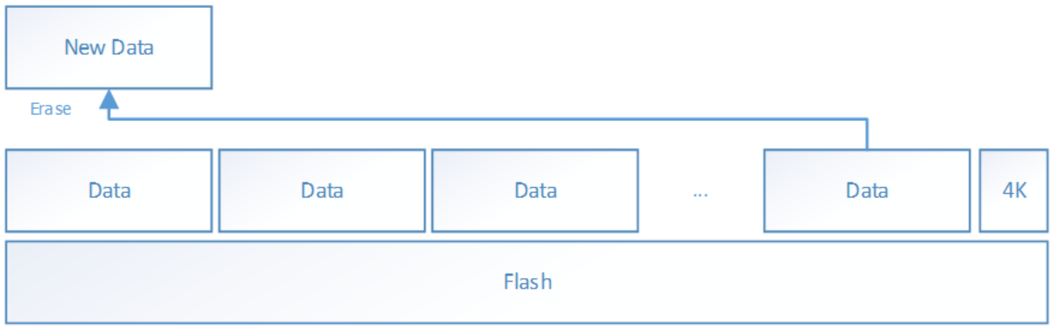


Figure 47: Circle write for flash

4.3.6 Data Verification

Since the length of each compressed record is 14 bytes, to align the data storage to 4096 bytes for storage, every 18 records are combined as a group, and the length of each group is 14*18=252 bytes. Because the length of each CRC32 checksum is always fixed into 4 bytes, the total length of each group can be fixed by adding a CRC32 check code at the end of each group, and then every 16 groups of records can be aligned to 4096 bytes. When reading the records, the group will be used as a unit, and every 18 records will be checked together. If the checked checksum does not match the recorded checksum, this group of data will be discarded. The calculation process of CRC32 on the data node will need the hardware calculation accelerator provided by the microcontroller to reduce the calculation time and power consumption. The CRC32 verification process on other system parts mainly relies on software to finish, which enhances software compatibility and portability.

4.3.7 Data Backup

Data backup relies on more storage chips. It can only be performed when the motherboard of the node comes with backup flash memory or an external flash memory module is attached. The data backup strategy can be selected from real-time backup and lazy backup, as shown in Figure 48. The real-time backup strategy will write to the backup flash memory and the external flash memory simultaneously when the primary flash memory is written. This mode can ensure that data will not be lost, even if the primary flash memory is damaged during the writing process. The disadvantage is that each write-out command needs to wait for all chips to finish writing and for data verification to complete, which will increase the wake-up time and consume more power. The lazy backup strategy will synchronize data from the primary flash memory to another flash memory after a certain number of writes. Because multiple sets of data are synchronized at one time, the overall wake-up time and power consumption can be reduced. However, the obvious disadvantage is that the backup period will be extended. Due to the occupancy of the flash memory, the information collection work will be temporarily suspended during the data backup period. Therefore, if the backup process takes a long time, it may cause the data collection time point to be missed, and when the primary flash memory is damaged during the two backup periods, the data recorded during the last backup time point until the next backup time point will be lost.

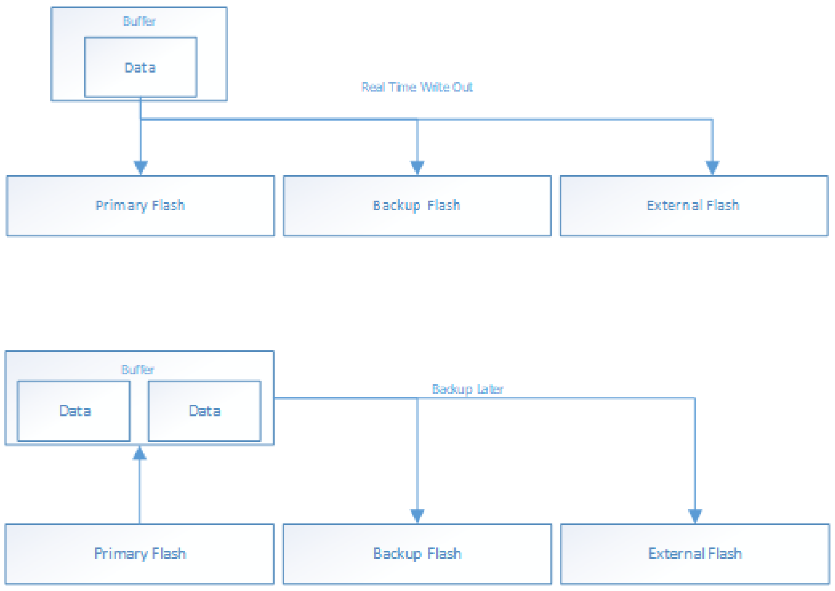


Figure 48: Process of two backup strategies

4.4 System Configuration

Because the system has multiple hardware designs and there are limitations in interchip detection, it is necessary to predefine the device configuration in the system. The configuration content includes software ID, board type identification, hardware configuration, and other information. If the node is equipped with a Bluetooth module, it must contain additional information, such as device name, Bluetooth communication key, and trusted connection client address. If the node is equipped with a Wi-Fi module, it is required to contain additional information, such as network name and network key. The pre-defined information that is required by some external modules will also be stored in this section, and the total length of all system configurations cannot exceed 4096 bytes.

4.5 System Startup

The start-up process of the system is divided into several stages. The microcontroller and the attached external modules take power from the power supply at the same time. Therefore, even if the microcontroller does not generate the command signal, the devices on the motherboard and attached external modules can still obtain power from the power bus. Most modules will start self-testing after being powered up, and the microcontroller will start self-testing synchronously. After completing the self-test, the microcontroller will enter the software initialization process. The system will first attempt to obtain the clock signal. If the suitable clock signal is reached, the boot-up process will enter the next stage. If the suitable clock signal cannot be obtained, the system will stop, and the status indicator will keep lit up. Thereafter, the system will activate all external interfaces and configure the interface status. If the interface is successfully activated, the boot-up process will enter the next stage. If the interface is not successfully activated, the system will stop and the status indicator will keep lit up. This step will enable the offline communication interface on the board type that contains the offline communication interface. If the interface is successfully enabled, the status indicator will turn off at this step, and if the interface cannot be initialized, the status indicator will keep lit up. In the next step, the system will initialize all communication protocols. This step is the final preparation work for the system to have the ability to communicate with peripherals. If the initialization completes successfully, the status indicator will blink once, and if it fails, the status indicator will keep lit up. Next, the system will first try to communicate with the built-in flash memory. If it receives a response from the built-in flash memory, it will attempt to obtain configuration information from the built-in flash memory and initialize the system and other modules based on the configuration information. If the configuration information states that the system contains a Bluetooth module, the system will initialize the Bluetooth module and commence Bluetooth broadcasting according to the configuration information. It will also enter the adaptive broadcasting mode after a certain period to reduce power consumption. If the configuration information states that the system contains a WiFi module, the system will attempt to join the node network according to the configuration information. The Wi-Fi module will always remain in online mode before successfully joining the node network for the first time. After successfully joining the node network, the Wi-Fi module will close until the next communication time point reaches. If there is no flash memory module, the wireless module is initialized directly, and the system configuration information is provided by the wireless module. After the system has completed all the startup steps, the system will regularly blink the status indicator, indicating that the system has fully initialized and started normal working. The process is shown in Figure 49.

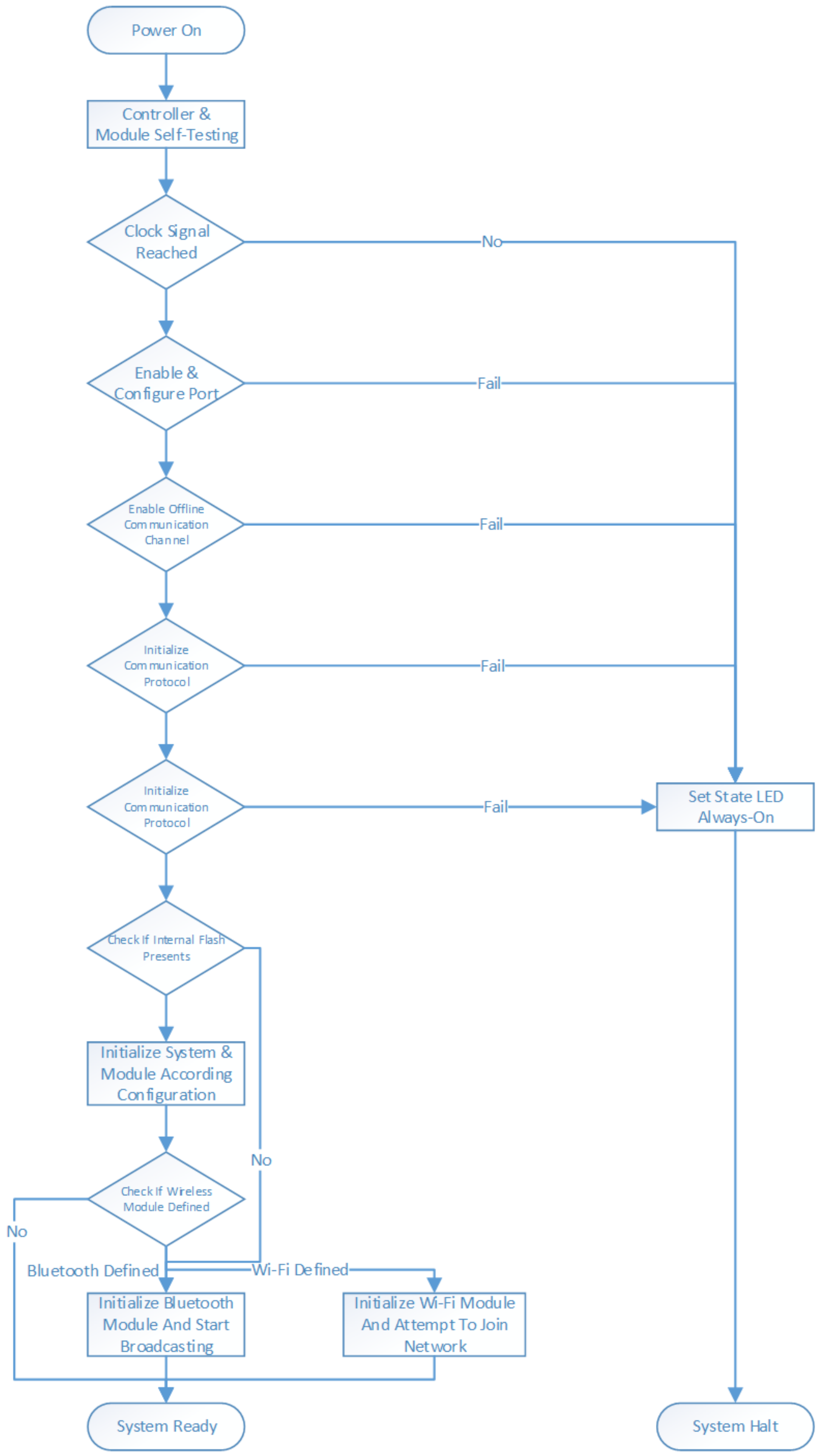

Power On
Controller & Module Self-Testing
Clock Signal Reached
No
Enable & Configure Port
Fail
Enable Offline Communication Channel
Fail
Initialize Communication Protocol
Fail
Initialize Communication Protocol
Fail
Set State LED Always-On
Check If Internal Flash Presents
Initialize System & Module According Configuration
No
Check If Wireless Module Defined
No
Bluetooth Defined
Wi-Fi Defined
Initialize Bluetooth Module And Start Broadcasting
Initialize Wi-Fi Module And Attempt To Join Network
System Ready
System Halt

Figure 49: Process of system startup

4.6 Wi-Fi Network

The typical Wi-Fi network structure is centralized, as shown in Figure 50. The center of the network is provided by the access point. The device connects to the network by communicating with the access point. The device needs to be located inside the signal coverage range of the access point, and the device needs to have the ability to transmit wireless signals to the access point to complete the communication with the access point. However, according to the requirement of the node network, the data node needs to have the ability to complete communication without external Wi-Fi network signal coverage. Therefore, a node network constructing method is designed to transmit, receive, and redirect data by establishing Wi-Fi direct connection, and nodes need to have the ability to connect to each other, as shown in Figure 51, then a large-area node network can be constructed through connections between nodes.

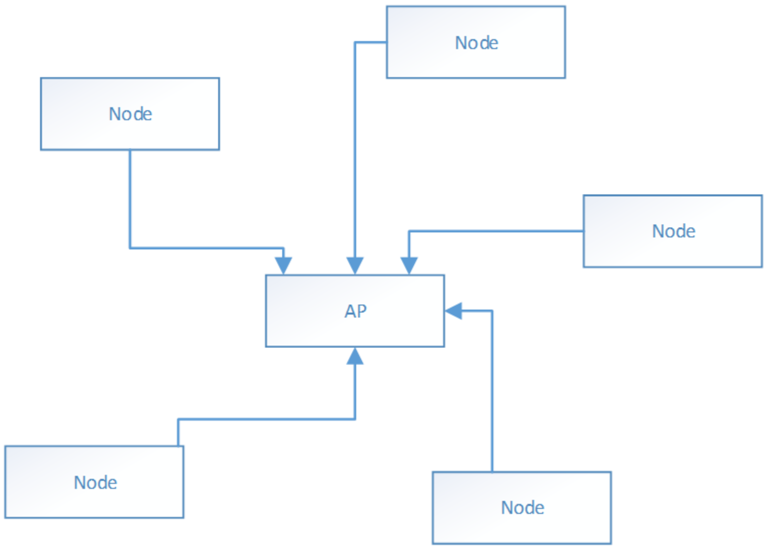


Figure 50: Structure of typical Wi-Fi network

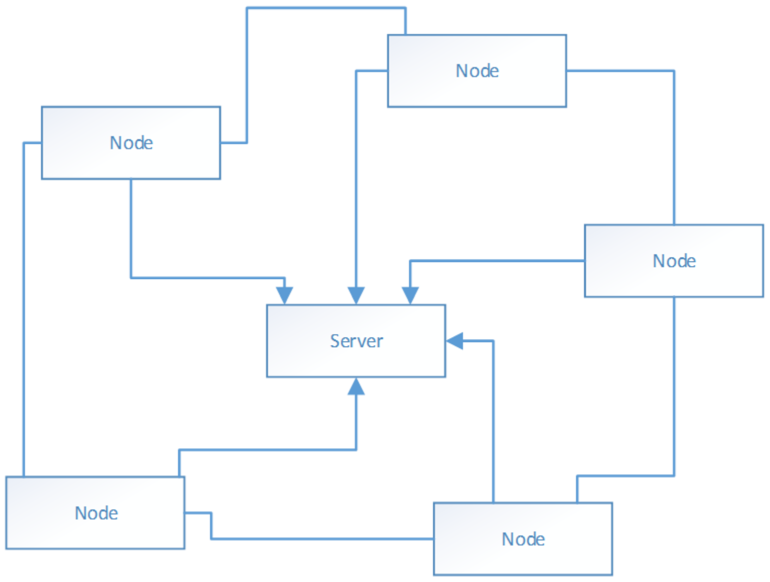


Figure 51: Structure of node network

4.6.1 Physical Connection

The node network construction is based on Wi-Fi communication; the selected frequency band is 2.4 GHz. Relying on the Wi-Fi standard, the conflicts of communication channels between different Wi-Fi devices can be automatically reduced, and the amount of data synchronized by the node network is limited during each period. Thus, compared to the communication methods such as Bluetooth and ZigBee, Wi-Fi networks have superior carrying capacity when a large number of nodes must communicate at the same time. Modern Wi-Fi chips can work at the access point mode and the station mode at the same time on a single chip, which provides a basis for building a network between nodes. Each node will start an access point and operate in station mode at the same time after the communication time point is reached. First, the Wi-Fi module will scan the broadcast signals from other nodes that may exist in the surrounding area according to the node's pre-defined node network information. After joining a surrounding node, the access point mode will be activated to broadcast the information of the node itself according to the node's predefined information. This will allow other nodes that are far from the control node to join the upper sub-network through this node. As more nodes join the node network, the robustness of the network becomes stronger. The process is shown in Figure 52.

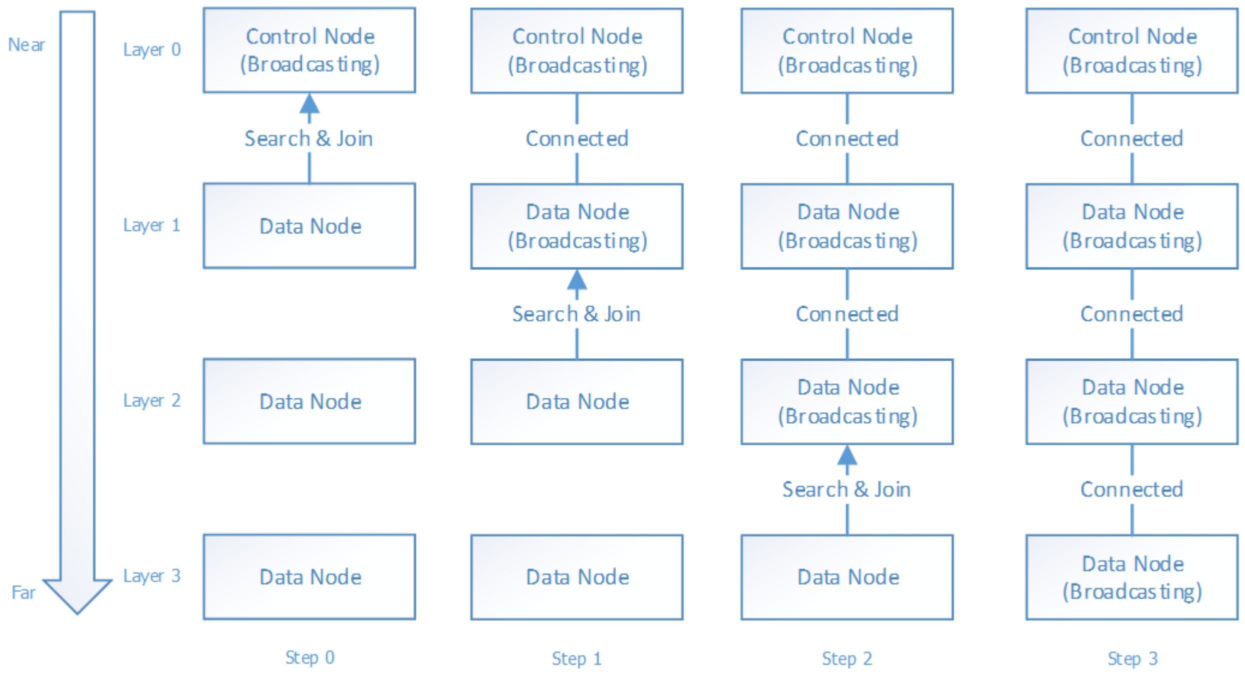


Figure 52: Process of join node network

4.6.2 Network Identification

The physical connection provides the basis for building a node network, but it is not enough to create a complete network. In practical applications, it may face more complicated situations, so the identification layer needs to be added.

The first purpose of the identification layer is to scan and identify the broadcast signals from other nodes. Multiple Wi-Fi signals may be present in the environment where the nodes are deployed, such as public Wi-Fi access points in public places and Wi-Fi access points from the network printers. These Wi-Fi access points that are not part of the node network and will cause interference when establishing connections between nodes, so identification information needs to be added to distinguish the node network from other networks. The most basic method is to add a prefix to the name when the node broadcasts itself. This will allow other nodes to quickly discover and identify the node network provided by other nodes when scanning the broadcast signals and to exclude irrelevant non-node networks. At the same time, different prefixes can also be used to perform node network isolation between nodes. When multiple node networks need to coexist in the same area, different prefixes will immediately identify different node networks, which will allow multiple node networks in the same area to communicate with other nodes that belong to the corresponding node network.

The second purpose of the identification layer is to prevent mistakenly joining access points that are not part of the node network. The identification layer will contain the verification of user-defined password combinations. This password combination will be used to authenticate each node when joining other nodes. Only when both parties hold the same password combination can complete authentication take place. Then the node can connect to other nodes, which will greatly reduce the possibility of third-party stations joining the node network inadvertently, and protect the node network, as shown in Figure 53.

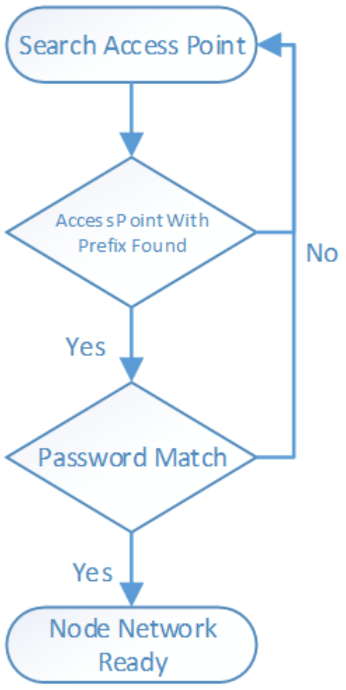


Figure 53: Process of security check when join node network

The third purpose of the identification layer is to identify the control node. Regardless of how many nodes exist in a node network, according to the system architecture, all the data of the data node needs to be sent to the control node and processed by the control node. Therefore, the related information of the control node is always the same, so the relevant information of the control node is directly included in the identification layer of each node network configuration, and the data is transmitted to the control node, level by level, during data forwarding. Since the identification layer can only contain information about one control node, it prevents repeated data transmission and receiving when there are multiple control nodes in the node network due to mistaken configuration.

The identification layer will also contain the information of the node itself and the information of the target node, as shown in Figure 54. Thus, it can ensure that the source of the data can be traced during the transmission process, which will prevent confusion arising from transferring the same data pack over the node network. It can also help other nodes to identify the current node more easily by adding an identification layer. Moreover, when the control node sends information to the data node, the data pack can be forwarded between nodes by exchanging the identification head of source and target, until the node that matches the identification head receives the data pack from the control node.

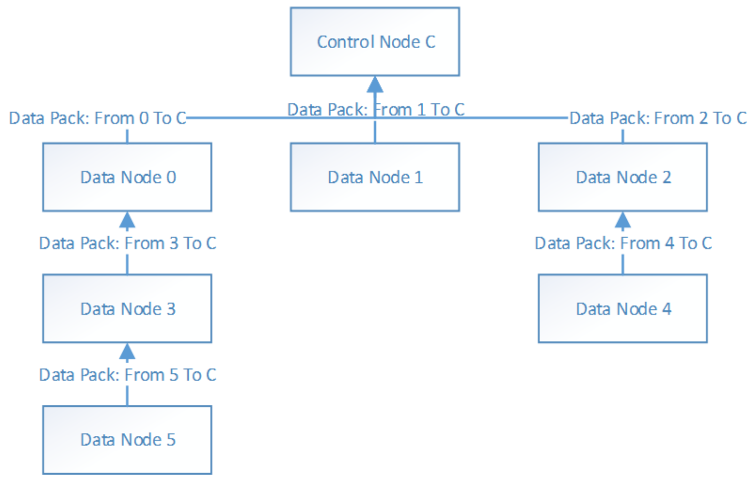


Figure 54: Additional mark for identification

4.6.3 Network Stack

Because the node network depends on wireless communication, unstable network communication is inevitable, such as signal blocking, network congestion, or communication interruption caused by too many wireless stations. Therefore, whether the data packet comes from the node itself or other nodes, the node that functions as a transceiver needs to be able to buffer the data packet. Thus, a circular stack must be implemented to buffer the data packets on the node network. The stack adopts the first-in-first-out rule to send and receive data packets. Because of the system memory management limitation, the memory space of the network stack must be one-time allocated, the length of the memory space needs to be static, and the number of data packets that can be buffered is fixed at 80 data packets. When the stack is full and the network is still blocked, the buffered data packet still cannot be forwarded. If any other nodes request to forward data via this node, the request will be immediately rejected and an error code is sent to the source node. Knowing that the upstream node network is blocked, the source node will suspend the data packet sending process, wait for a period, and then initiate retransmission to prevent data loss or frozen network due to simultaneous transmission. The processes and status are shown in Figure 55.

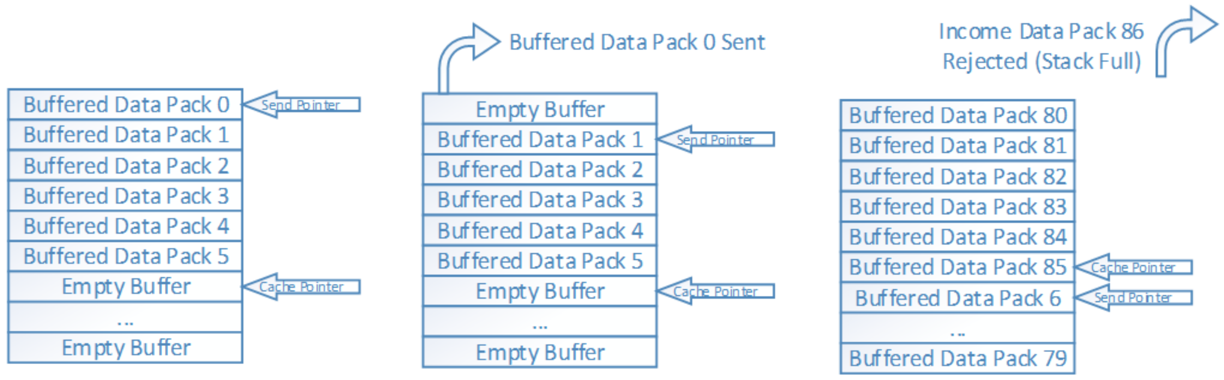


Figure 55: Different status of network stack

4.6.4 Data Packet Transmission

Relying on the network stack, each node can send and receive data packets in an orderly manner. Due to the additional target and source information provided by the identification layer, the node network not only can send data packs from data nodes to control nodes, the control nodes also can send data packs to data nodes. However, there are certain differences between the two ways of data pack transmission.

Sending data from the control node to the data node may significantly increase the load on the node network. According to the network structure, the control node is always at the top level of a node network logically, regardless of where the control node is located physically. Therefore, while the data node is sending information to the control node, it only needs to ensure that the information is sent to the upper-level node, and the upper-level node will continue to forward upwards. Thus, the data pack will eventually reach the control node. However, there is uncertainty when sending information from the control node to the data node. Although the control node can estimate the transfer direction of a data node according to the last data node that forwards the information, due to the limited memory on the data node, each data node cannot store information on which nodes are located at the next level of the subordinate nodes. Therefore, the data packet sent from the control node to the data node will be conducted in a directional broadcast mode. The control node can obtain an estimated forwarding direction according to the data node directly connected to the control node and the information about the source node of the data contained in the identification information. Then it will broadcast the data pack on each of the data nodes in this direction, that is, the data node in this direction and all its subordinate data nodes will receive and forward the data packet until it finally reaches the target data node. The information will be confirmed by the target data node and will no longer be forwarded. Therefore, the worst case in the entire downward transmission process is that only one data node is directly connected to the control node, and the target data node is located at the outermost edge of the entire network. At this time, each data pack will be forwarded over the entire node network and all the nodes within this node network will receive the data pack. Moreover, all the data nodes that have subordinate data nodes must forward the data pack.

As shown in Figure 56, when data node 5 and data node 4 send information to control node C, the information sent from data node 5 will only be forwarded by data node 3 and data node 0, and then finally arrive at control node C. During the transmission process, the data nodes 5 and data node three do not need to know the logical location of the control node C; they only need to forward the data pack, level by level. Even if there are other data nodes under the data node 0, the information from data node 5 will not be forwarded to other data nodes, and the data pack will never be forwarded between nodes located on the same logical level during the process of sending from the data node to the control node. Similarly, the information sent from data node 4 will only reach control node C through data node 2, and data node 1 will not participate in information forwarding during the whole process. Therefore, there is only one path for data sent from the data node to the control node. Only the direct superior data node of the sending node will participate in the forwarding process.

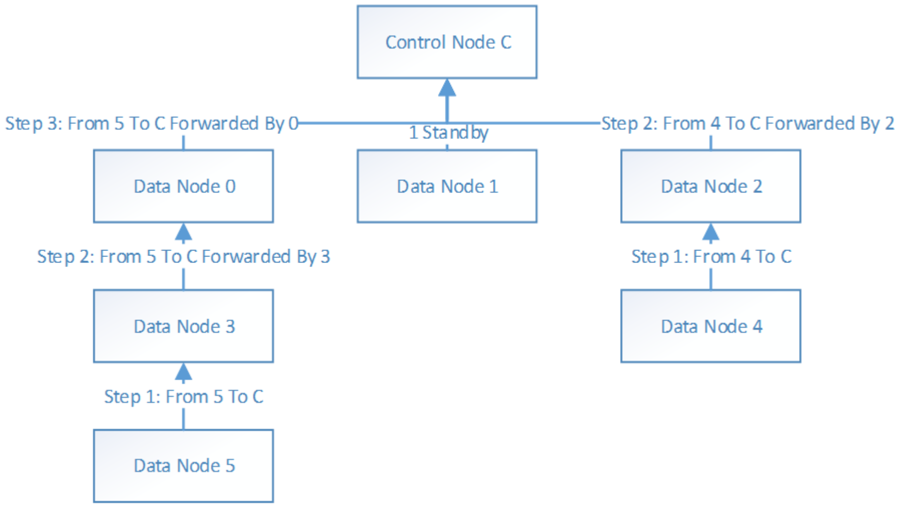


Figure 56: Process of sending from 5 and 4 to C

However, there is a difference when the information is sent from the control node to the data node. The example in Figure 57 below shows the forwarding process in the ideal state. The data has only one path, and the forwarding process only takes place between related nodes. Since the data node will actively send handshake data packets to the control node during the establishment stage of the node network, the control node can determine the general direction of a data node according to the forwarding process and identification information. As shown in the figure above, the data pack that was sent by the data node 5 will finally be forwarded by the data node 0 that is directly connected to the control node from which the control node can know that data node 5 is located below data node 0. Thus, when the control node sends information to data node 5, it only needs to send information to data node 0 for downward forwarding, and the data pack will eventually reach data node 5. Similarly, if the control node sends information to data node 4, it is only required to send the information to data node 2 for downward forwarding and will eventually reach data node 4.

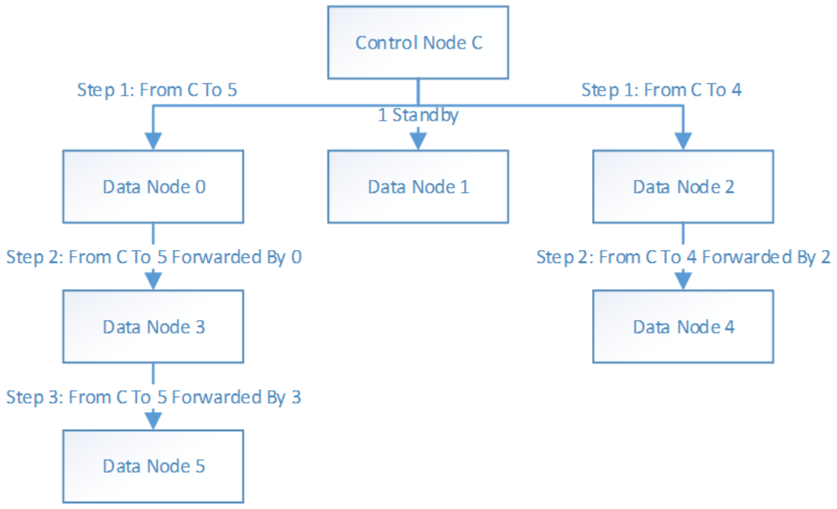


Figure 57: Process of sending from C to 5 and 4

However, downward information transmission is not always ideal. The following Figure 58 is an extension of the above network. Now data node 0 has a sub-data node 30, data node 30 has sub-data node 50, and data node 50 has sub-data node 60. When the same information must be sent from the control node C to the data node 5, it is the same as sending the data from the control to the data node 0. Then it will be forwarded downward by data node 0 because each data node can only know the data node directly connected to it and cannot know the node distribution located at the level after the next level. Thus, if the information is sent to the data node 3 or the data node 30, the data pack can be directly forwarded to the corresponding data node via data node 0. However, when a data pack needs to be sent to data node 5, data node 0 does not know the direction of data node 5, so it has to broadcast the information downward on the entire subordinate network, so the information will be forwarded to data node 30 and data node 3. Since data node 5 is located under data node 3, the information will be successfully delivered. At this time, data node 5 will no longer forward information to data node 6. Data node 6 and data node 1 will not participate in the process of data pack forwarding, but data node 50 does not know whether data node 5 will be located in the level after the next level of its own. Thus, it will keep forwarding until the data node 60 is located at the outermost edge of the entire node network, then the information forwarding ends. Therefore, it is easy to infer that when there is another data node below data node 60, the information that should be sent to data node 5 will be broadcast to the deeper area of the node network. This will consume a significant amount of network resources, occupy the network stack of the data node, and increase the load of part of the node network.

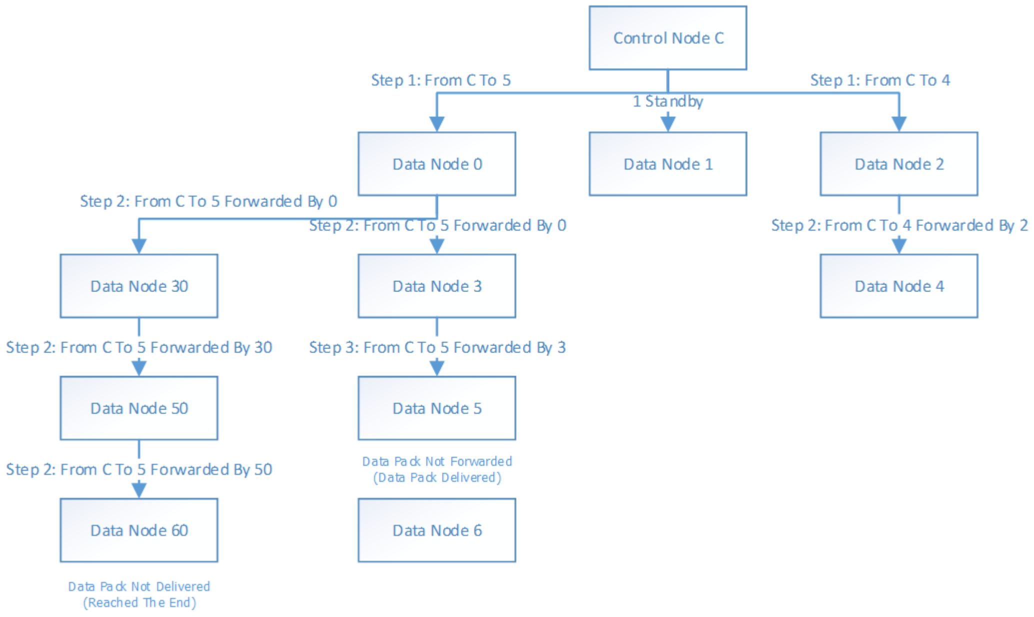


Figure 58: Process of sending from C to 5 and 4, extra forwarding process

Then consider the worst case. As shown in Figure 59 below, only one data node 0 is directly connected to the control node. Therefore, provided the information needs to be forwarded to the data node located at the level after the next level of data node 0, the data will be broadcasted in most of the data nodes. In this node network, if the information needs to be sent to data node 6, and data node 6 does not have any nodes under it, this situation constitutes the worst case, that is, the information needs to be broadcast on the entire network, especially when there are more child nodes under data node 1 and its subordinate data. Therefore, data nodes 1 and all the branches under data node 1 must participate in data forwarding. Although data node 6 is not located at the deep logical level in the node network, the information sent to data node 6 may be broadcast to a deeper logical level.

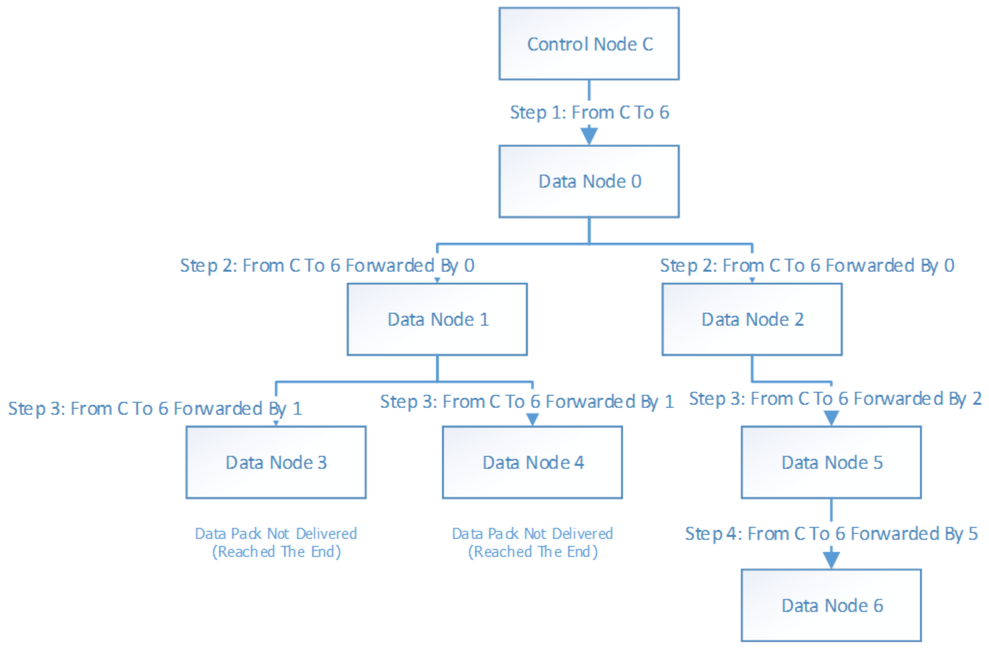

Control Node C
Step 1: From C To 6
Data Node 0
Step 2: From C To 6 Forwarded By 0
Step 2: From C To 6 Forwarded By 0
Data Node 1
Data Node 2
Step 3: From C To 6 Forwarded By 1
Step 3: From C To 6 Forwarded By 1
Step 3: From C To 6 Forwarded By 2
Data Node 3
Data Node 4
Data Node 5
Data Pack Not Delivered (Reached The End)
Data Pack Not Delivered (Reached The End)
Step 4: From C To 6 Forwarded By 5
Data Node 6

Figure 59: Process of sending from C to 6, worst case

However, to ensure integrity during data transmission and reception, certain strategies have to be applied, which will reduce the performance of the node network. Relying only on the physical connection will not ensure integrity during data transmission.

4.6.5 Transmission and Response Mechanism

In each process of sending a data packet, there is a mandatory response and an optional response. The mandatory response is usually provided by the node that receives the data packet and so identifies the receiving state of the data. The sender will wait for the mandatory response each time after sending a data packet to other nodes, and the receiver will return a successful response to the sender immediately after the data packet successfully enters the node network stack, which means that the data packet has entered the next level of forwarding process. As shown in Figure 60, when data node 2 transmits data to control node C, the data pack needs to be forwarded by data node 1. Thus, data node 2 first sends the data pack to data node 1, and when data node 1 stores the data packet in the node network stack, it will inform data node 2 that the data pack is successfully sent, and data node 2 can now remove the data packet 0 from its network stack.

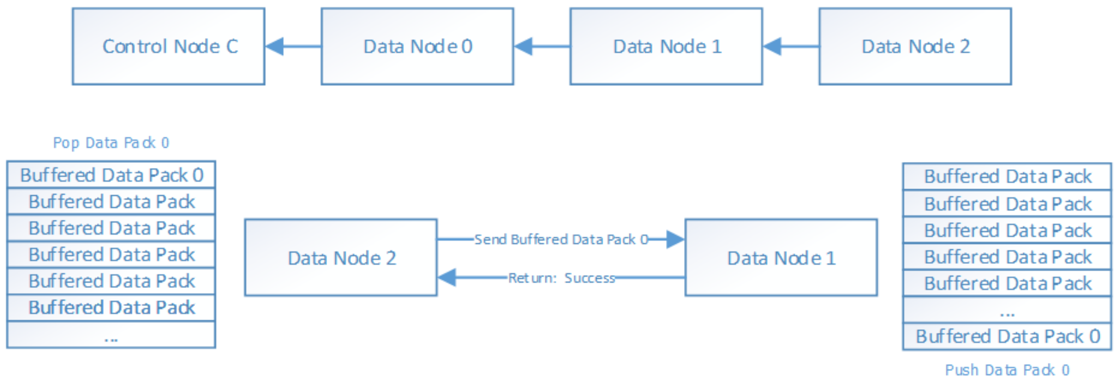


Figure 60: Success response between nodes

However, there may have unexpected situations during this transmission process. When the next level node in the forward process loses its connection to the node located at the level after the next level, the next level node will enter the reconnection state at this time. All incoming data packets will be directly discarded and a response will be generated and sent to the source of the sending process to indicate the node is in an abnormal state. As shown in Figure 61, the data packet sent by data node 2 to control node C also needs to be forwarded through data node 0. When data node

2 attempts to send data packet 0 to data node 1, data node 1 already knows that the connection with data node 0 is lost, so data node 1 will immediately discard data packet 0 and notify data node 2 that the connection has disconnected.

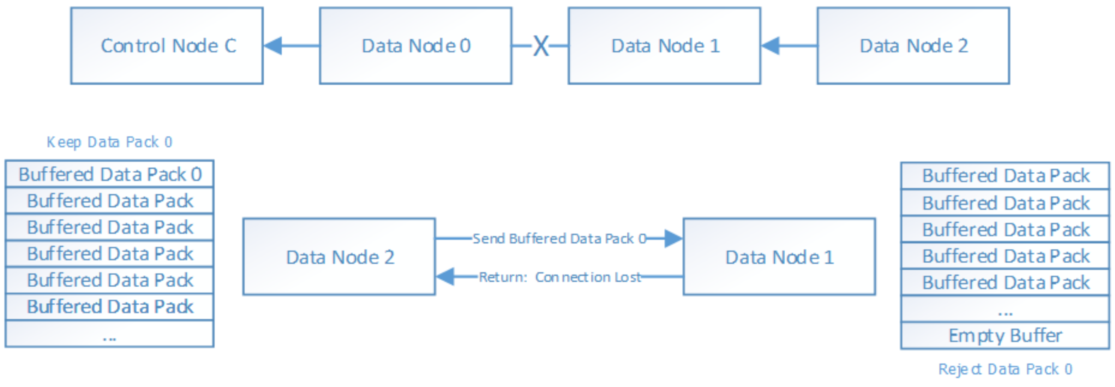


Figure 61: Connection lost response between nodes

When the network stack of the next level node is full, the data packet will be immediately discarded, and the source will receive a response that indicates the node is full. As shown in Figure 62, although the quality of network connection is good, since the stack of data node 1 cannot save more data packs, a response will be generated and return to data node 2, indicating that the stack is full. At this time, data node 2 will retain data pack 0 and try to resend it after a period.

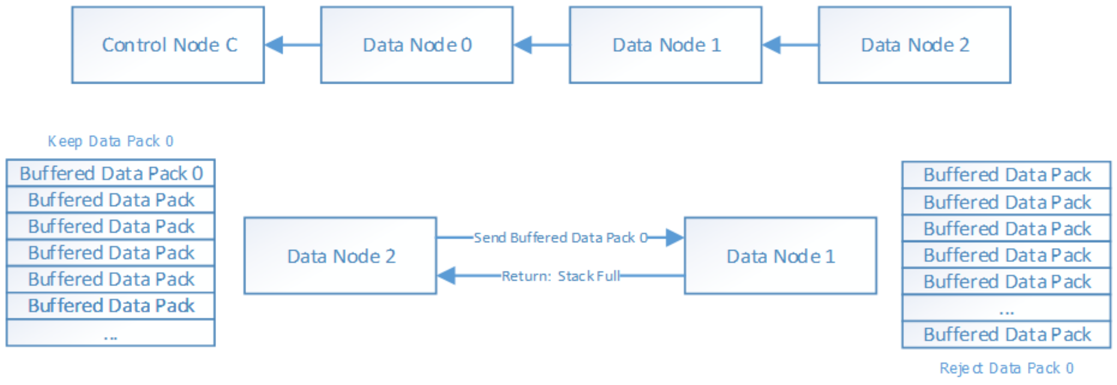


Figure 62: Stack full response between nodes

In addition to the response from the next level node, when the connection between the node itself and the subordinate node is disconnected, the node itself will generate a response that indicates that it cannot send data packs. As shown in Figure 63, data node 2 loses its connection with data node 1, so it cannot transmit data packets. At this time, the work of data node 1 is not affected, and data node 2 will keep data packet 0 and try to re-establish the connection with the node network.

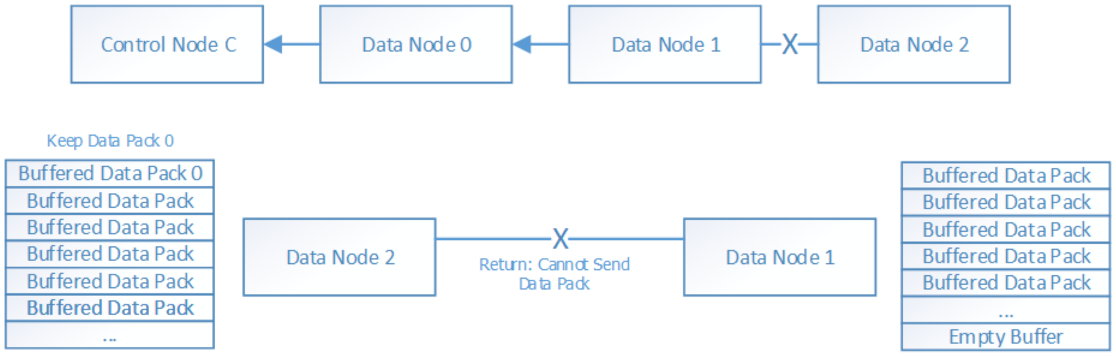


Figure 63: Situation of lost connection among nodes

The optional response can only be provided by the final target during the data packet transmission, usually, it will be provided by the control node. When the first node generates the data pack and sends it, once it is successfully delivered to the next level node, it usually defaults to be considered as successfully sent, but in an extreme case, the next level node may be offline forever. For example, the physical distance between two nodes on the transfer chain is too long. At this time, if an intermediate node is permanently offline, only the middle node that has buffered the data pack can know that it can no longer continue the forwarding process, but the first node obviously cannot know whether the data packet has successfully reached the control node. At this time, once the network can never recover or the middle node is offline again, the buffered data pack will be lost permanently. As shown in Figure 64, data node 2 successfully transfers the data pack to data node 1 and then data node 1 returns the successful transmission response, then data node 2 pops the data pack from its network stack. At this time, only data node 1 buffered the data pack, and since the connection from data node 0 to the control node is disconnected, so data node 0 will refuse to forward the information from data node 1. At this time, data node 1 keeps the data packet in its network stack. However, if data node 1 is accidentally offline at this time, the data pack that was sent by data node 2 will be lost permanently. Although data node 2 received a successful transmission response, the control node did not successfully receive the data packet.

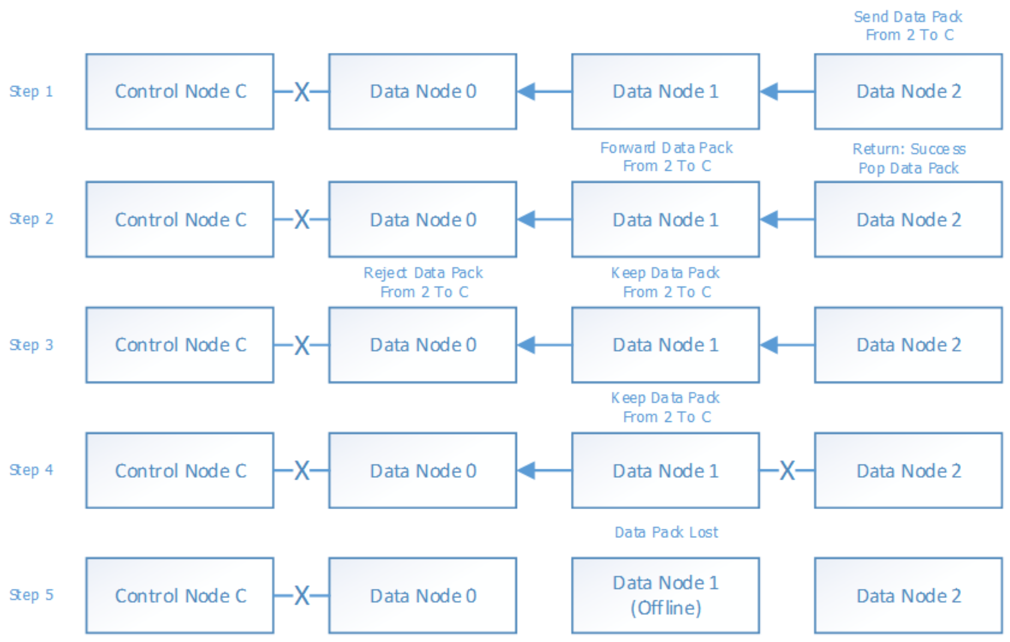


Figure 64: Data lost during the forwarding process

Therefore, an optional response mechanism is added. When the control node receives the data packet and confirms that it is valid, the control node will reply with an optional response to the first node. The first node will only confirm that the data pack is completely delivered after receiving the optional response from the control node. Although it can be known from the previous section that sending information from the control node to the data node will increase some of the network load, this will be a very effective way to ensure that the information is delivered without loss in an environment plagued by uncertain network conditions. As shown in Figure 65, after the data packet from data node 2 is forwarded to the control node, level by level, the control node will broadcast the optional response information on the node network, and the optional response will finally reach data node 2 after the level-by-level forwarding. Data node 2 will only pop the buffered data pack after receiving the optional response from the control node.

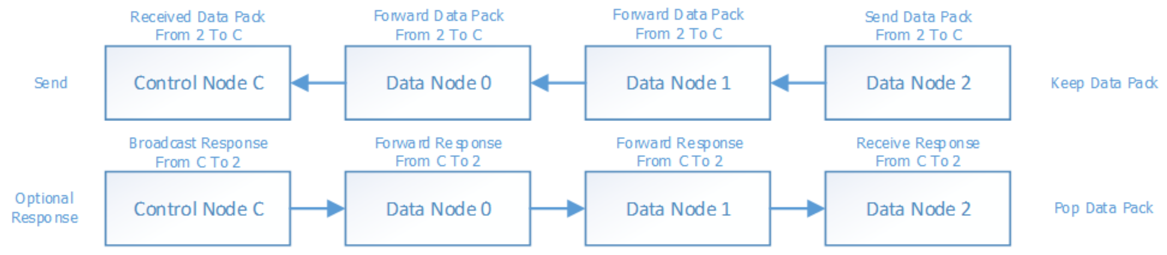

Received Data Pack From 2 To C
Forward Data Pack From 2 To C
Forward Data Pack From 2 To C
Send Data Pack From 2 To C
Send
Control Node C
Data Node 0
Data Node 1
Data Node 2
Keep Data Pack
Broadcast Response From C To 2
Forward Response From C To 2
Forward Response From C To 2
Receive Response From C To 2
Optional Response
Control Node C
Data Node 0
Data Node 1
Data Node 2
Pop Data Pack

Figure 65: Process of optional response

Regardless of whether the data pack sent by data node 2 arrives at the control node, under the optional response mechanism, provided data node 2 does not receive a response from the control node within the specified time, it will be treated as a data pack transmission failure. Thus, data node 2 will always buffer the data pack and try to resend the data packet after a period. At this time, the data node 2 will generate a sending timeout response by itself, as shown in Figure 66.

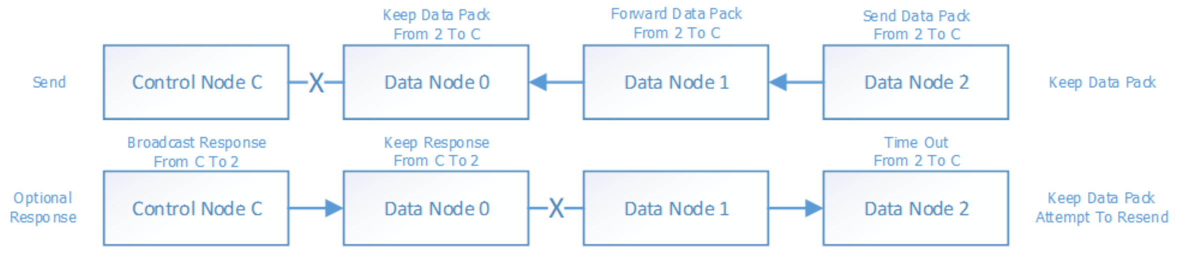

Keep Data Pack From 2 To C
Forward Data Pack From 2 To C
Send Data Pack From 2 To C
Send
Control Node C
X
Data Node 0
Data Node 1
Data Node 2
Keep Data Pack
Broadcast Response From C To 2
Keep Response From C To 2
Time Out From 2 To C
Optional Response
Control Node C
Data Node 0
X
Data Node 1
Data Node 2
Keep Data Pack Attempt To Resend

Figure 66: Optional response to avoid data loss

4.6.6 Network Security and Malicious Attacks

Although the identification layer has been created to distinguish from other networks and protect the node network by adding prefixes and setting password combinations, it is still possible to create maliciously fake stations to attack node network due to the weakness of Wi-Fi communication itself. that is, by scanning the surrounding WiFi signal, the node network can be attacked by creating a station that has a cloned name and station address to mimic a node. However, due to the existence of the password combination, the fake node cannot join the node network. But this can only ensure that the node network is not maliciously joined, it cannot prevent the node network from communicating with the fake node when the node network is initializing. As shown in Figure 67, provided the signal strength of the fake node exceeds any other data node, it can prevent the data node that is trying to join the node network to establish correct communication between real data nodes. Even in daily life, cloned networks entice the public to join through a cloned public Wi-Fi that is visible everywhere. The identification layer can only increase the network security, but it cannot be used to determine which the real station is if two stations have the same attributes, such as the name, address, and frequency band of the station. Similar to FM broadcast, the radio can only hear the sound from the source with the strongest signal. However, before the password combination is leaked, the data on the node network is always safe.

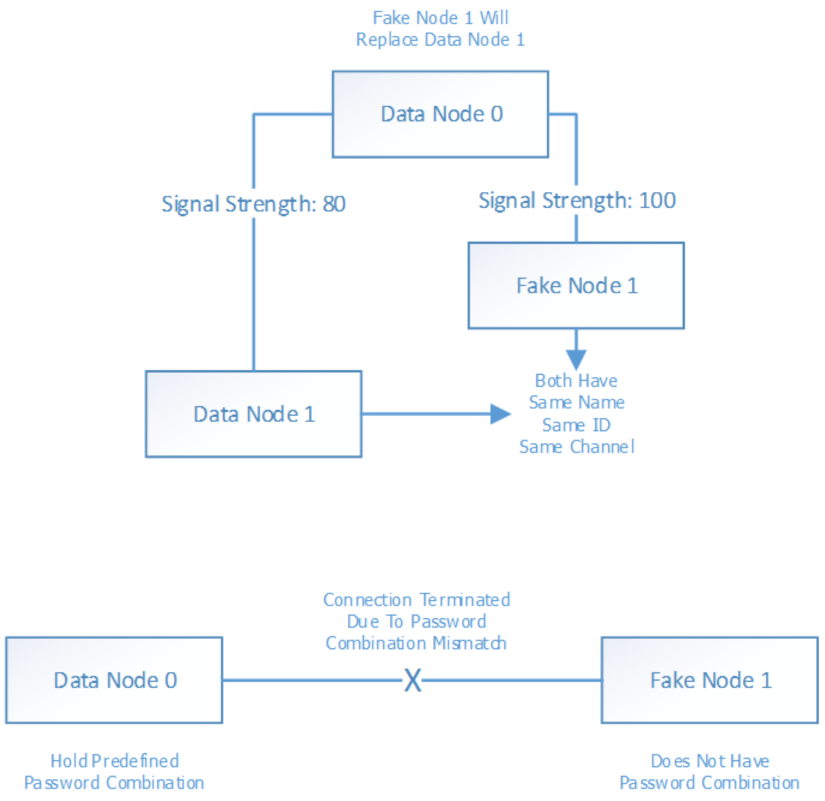

Fake Node 1 Will Replace Data Node 1
Data Node 0
Signal Strength: 80
Signal Strength: 100
Fake Node 1
Data Node 1
Both Have Same Name Same ID Same Channel
Connection Terminated Due To Password Combination Mismatch
Data Node 0
Fake Node 1
Hold Predefined Password Combination
Does Not Have Password Combination

Figure 67: Fake node attacks

4.7 Simple Data Calculation

The data node itself can perform simple calculations on the data. In this example, after each unit of environmental information has been collected, the data node will correct the acquired environmental information according to the correction parameters provided by the module, and the corrected result from the calculation is the final result to write into the storage. When the node network exists, the data node can also perform a certain kind of calculation locally based on the parameters from a remote server. For example, some constants may vary according to the time or region. At this time, these constants can be distributed to each data node through the control node.

Chapter 5

Management System Design

5.1 System Database Design

The entity is designed as follows, as shown in Figure 68: The record information is used to save the environmental information collected from the data node, which includes the internal ID, the timestamp when the information was collected, and the ambient temperature, humidity, and pressure at that time. The internal ID is stored in a self-increasing unique sequence, the timestamp is stored in a time-stamp data format, and the ambient temperature, humidity, and pressure are stored using floating numbers. The report log is used to save the time point when the node uploads data. According to this log, the system can know the last online time point of the node and the time point of historical data upload, the report log includes the internal ID and the timestamp of the data report. The internal ID is stored in a self-increasing unique sequence, and the timestamp is stored in a time-stamp data format. The node information is used to save the relevant software and hardware information of each node, including internal ID, node ID, node description, and firmware version. The internal ID is stored in a self-increasing unique sequence, the node ID and node description are stored in character format, and the firmware version is stored in floating numbers. Network information is used to save the relevant configuration information of the node network, including internal ID, network prefix, password combination, and network ID. The internal ID is stored in a self-increasing unique sequence, and the network prefix, password combination, and network ID are stored in characters. The node type is used to store different node hardware forms, including the type name and type description. The internal ID is stored in a self-increasing unique sequence, and the type name and type description are stored in character format. System logs are used to save logs generated by system operation and related logs generated by user operations, including internal ID, timestamp, and log content. The internal ID is stored in a self-increasing unique sequence, the timestamp is stored in a time-stamp data format, and the log content is stored in a character format. User information is used to record relevant user information in the system and is used to complete identity verification when a user logs into the system. It includes internal ID, user name, real name, password, and account-activation status information. The internal ID is stored in a self-increasing unique sequence, the user name, real name, and login password are stored in character format, and the account activation status is stored in a Boolean value. The permission table is used to save the information of permissions that can be given to the user in the system, which contains the name of the permission and description of the permission. The name and description are stored in character format. User types are used to record the types of users at different levels in the system and divide users group by group. The internal ID is stored in a self-increasing unique sequence, and the name and description are stored in character types. Each node's information corresponds to the information of multiple records and report logs, each network's information and each node type correspond to the information of multiple nodes, and each user's information corresponds to multiple system logs, multiple permissions, and each user type corresponds to multiple user information.

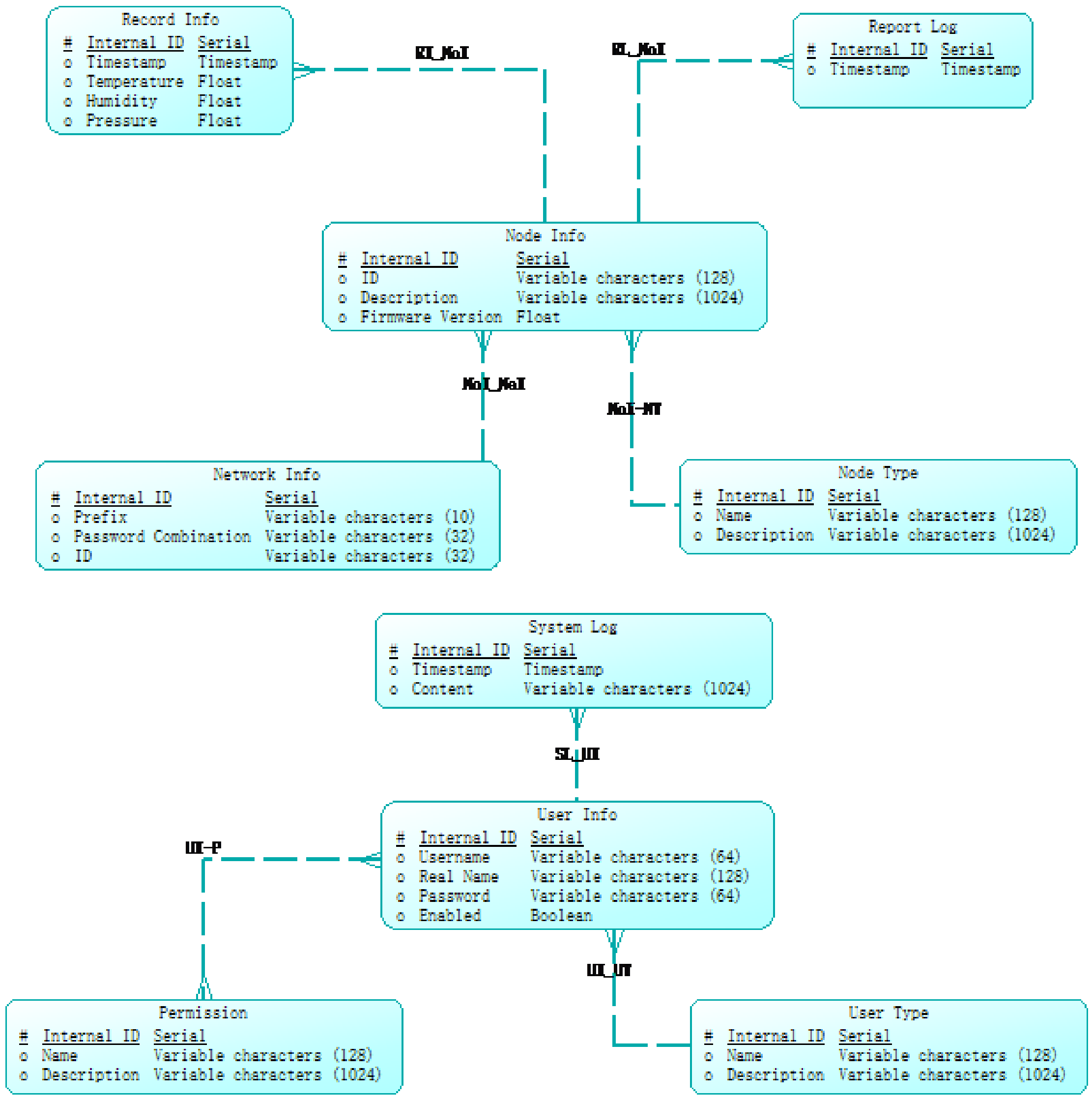

Record Info
# Internal ID Serial
o Timestamp Timestamp
o Temperature Float
o Humidity Float
o Pressure Float
RI_NoI
RL_NoI
Report Log
# Internal ID Serial
o Timestamp Timestamp
Node Info
# Internal ID Serial
o ID Variable characters (128)
o Description Variable characters (1024)
o Firmware Version Float
NoI_NeI
NoI-NT
Network Info
# Internal ID Serial
o Prefix Variable characters (10)
o Password Combination Variable characters (32)
o ID Variable characters (32)
Node Type
# Internal ID Serial
o Name Variable characters (128)
o Description Variable characters (1024)
System Log
# Internal ID Serial
o Timestamp Timestamp
o Content Variable characters (1024)
SL_UI
UI-P
User Info
# Internal ID Serial
o Username Variable characters (64)
o Real Name Variable characters (128)
o Password Variable characters (64)
o Enabled Boolean
UI_UT
Permission
# Internal ID Serial
o Name Variable characters (128)
o Description Variable characters (1024)
User Type
# Internal ID Serial
o Name Variable characters (128)
o Description Variable characters (1024)

Figure 68: ER model of the database

According to the ER model, a physical data model can be constructed, as shown in Figure 69. The table information required by the database can be formed by adding the corresponding relationship based on the entity. The relationship between the entities will exist in the related tables in the form of keys, when constructing the system, the required information can be obtained by just operate the tables by following the relationships between tables, the separated entities can increase the efficiency of database query and prevent redundant information causing trouble when operating the database. In the database table, node information is used as the foreign key for record information, node information is used as the foreign key for report log, network information is used as the foreign key for node information, and node type is used as the foreign key for node information. Furthermore, user information is used as the foreign key for system log, and user type is used as the foreign key for user information. Because there are multiple correspondences between user information and permissions, there is a middle table for auxiliary query.

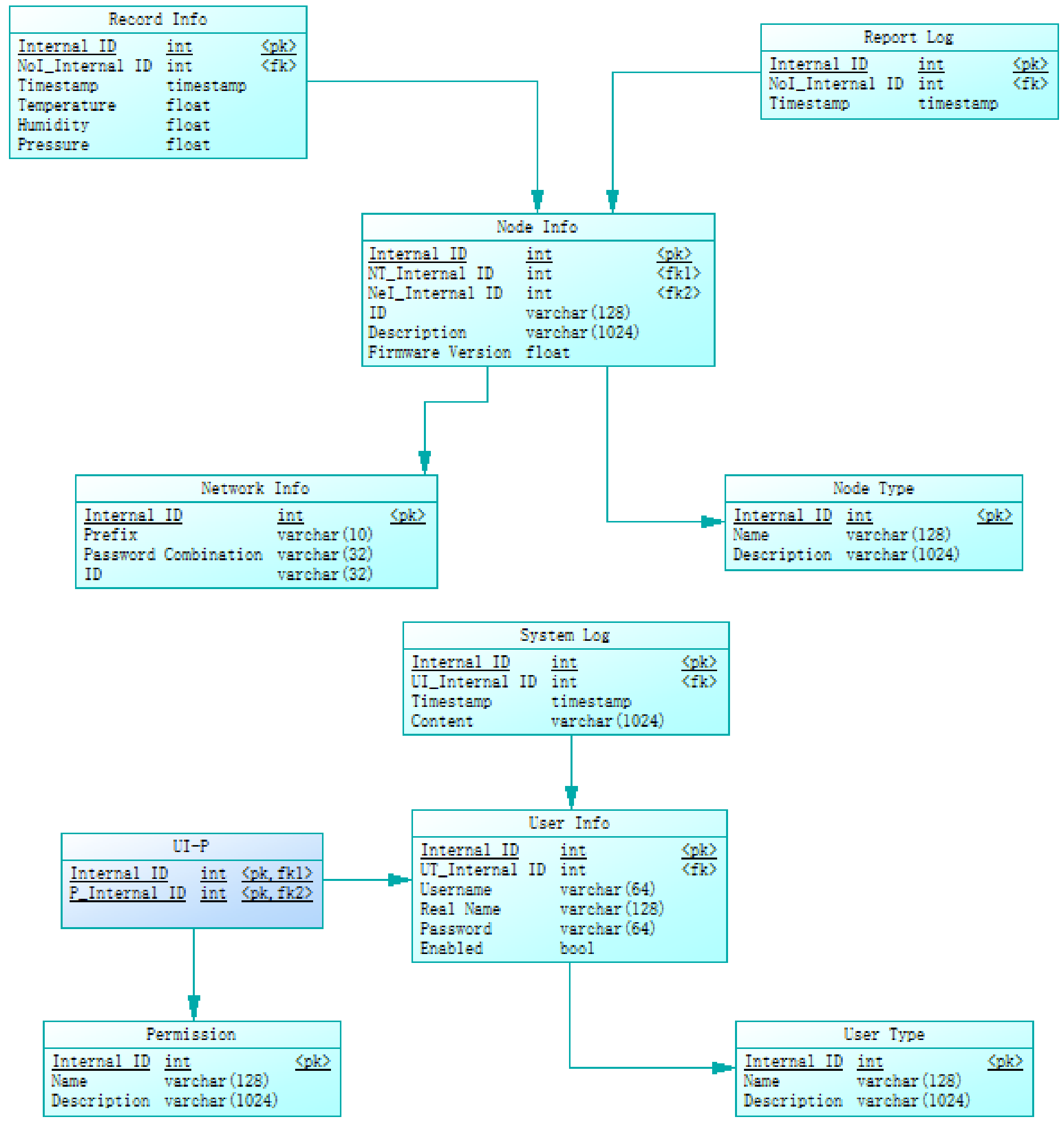


Figure 69: Physical model of the database

5.2 System Information Security and Malicious Attacks

Since the system needs to send and receive data on the network, a certain strategy is to be applied in software design to ensure the data will not leak. Due to the data packet-broadcasting characteristic of computer networks, it is almost impossible to prevent data packets from being intercepted. Therefore, to keep the information security, it is required to focus on how to ensure the data cannot be decrypted even if intercepted. At the same time, how to protect the system from attacks that come from the network is also important. For the data transmission process on the network, the most basic requirement is that the data should be encrypted before the transmission process. The introduction of the AES encryption algorithm can prevent information from being transmitted in plaintext. Even if a third party intercepts the data pack, the time cost for decryption will greatly exceed the value of the packet itself. The AES algorithm is a symmetric encryption algorithm, so the encryption and decryption speed is very fast. However, the symmetric encryption algorithm requires that both parties in the communication know the key used for encryption and decryption in advance, which will greatly limit the flexibility of encryption and decryption and reduce the system security. If the key is leaked, that means anyone who intercepts the data packet can quickly decrypt the information through the leaked key. Therefore, to increase the security level by changing the AES key frequently, the DH algorithm is introduced to perform the key exchanging process, and the AES key will be negotiated temporarily when each communication process begins. In this way, a one-time AES key can be generated for each network connection. Even if the AES key of a certain communication is cracked by force, the leaked key cannot be used to decrypt new communications and other communications. Even so, the communication process can still be intercepted by the intermediary. When the intermediary exists, both parties in the communication think that they are communicating with each other. However, the communication process has already been intercepted by the intermediary, because the intermediary can obtain the keys of both parties from the key exchange process. Thus, all encryption measures are inadequate. To prevent the existence of an intermediary, a certificate verification mechanism has been introduced, that is, the management server holds a certificate and its private key, and all control nodes hold the public key of the certificate published by the management server. The certificate verification will be performed during the key exchanging process to check the identity of the management server. If the public key and private key of the certificate do not match, then it can be regarded as a risk, and the network connection will be immediately terminated. In addition to the security of the connection, the security of the system itself is also crucial. The server and related databases and other systems need to be configured with firewall protections. If allowed, all network ports that are not related to the necessary services need to be closed to prevent the system privilege escalation caused by the service that has security vulnerabilities. This kind of privilege escalation may lead to high-privilege damage to the server. The most significant risk for system users is the leakage of system user names and passwords due to malware or accidental leaks. In this case, the system logs can be queried to verify the login status of the relevant users, and the system operation records can be queried to check whether the user name and password may be leaked and, if necessary, the account at risk can be blocked by deactivating it.

5.3 Management Server Design

The management server is a system service without a graphical interface. It is configured to receive the reported data collected from the data node and submitted by the remote control node. It then organizes and writes the data into the database according to the relevant information, and it then responds to the management client during the login process and provides information according to the related user operations. The management server always listens on a fixed network port and performs different processing according to the content of the incoming message. The overall architecture of the server is shown in the figure, including port listener, database manager, report analyzer, backup scheduler, and export interface as, shown in Figure 70.

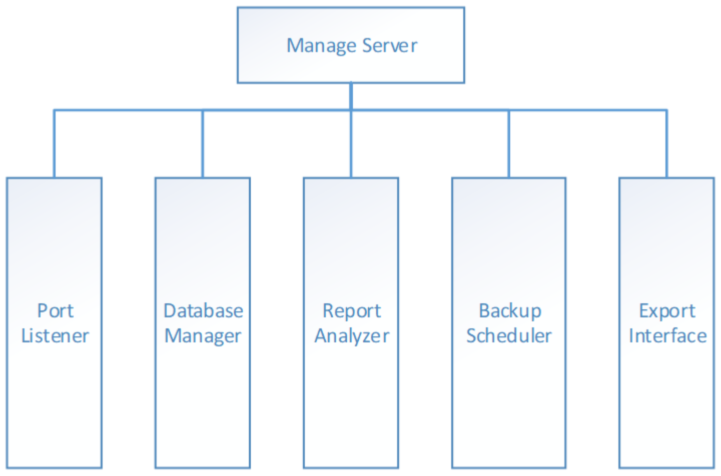


Figure 70: Architecture of server

5.3.1 Port Listener

The port listener is responsible for sending, receiving, and filtering information over the network port and responding to user operations. It is also responsible for checking and authenticating incoming connections. Only incoming connections that can pass the final verification and pass the communication key exchange will be allowed, other connections will be cut off. The port listener is the only module that directly connects the server with the client. Any type of client can only communicate with the server by establishing a connection with the port listener. It is not allowed to bypass the port listener and directly access other modules of the server.
The port listener includes an authentication module, an event filter, and a command interpreter, as shown in Figure 71. The authentication module includes a codec. When an incoming connection arrives, the event filter will forward the incoming data to the authentication module. Then the server will authenticate the identity and exchange the key with the incoming client. After the communication key is confirmed, the authentication module will add the connection and its communication key to the queue, and the subsequent data transmission and reception will be processed by the authentication module and the codec. When the data packet is decoded, it will be

sent to the event filter again. The event filter will analyze the content contained in the data packet. If the content is the report that is uploaded by the remote management client, the report in the data packet will be split and handed to the report analyzer for subsequent processing. If the content of the data packet is a command, the command part will be split separately and handed to the command interpreter for analysis and execution. The result of the command execution will be passed to the event filter again, and the event filter will repackage the execution result and pass it to the authentication module. It is then sent back to the target client by the authentication module inside the port listener.

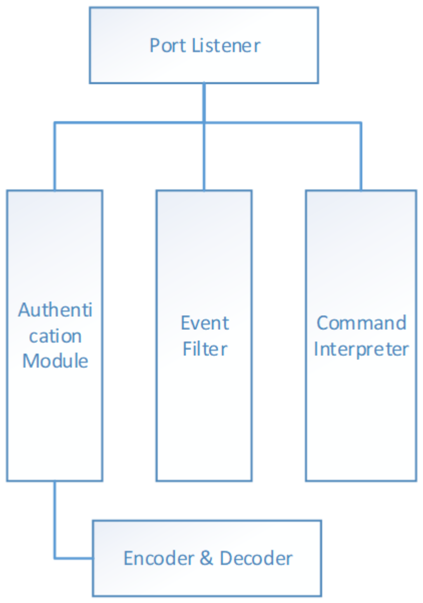


Figure 71: Architecture of port listener

5.3.2 Database Manager

The database manager is responsible for all operations related to the database, whether it is storing the environmental information reported by the control node in the database, saving the logs generated by the system operation, or reading the database when the client or third-party plug-ins try to obtain information from the system. All these operations need to go through the database manager. An independent database manager can make it possible to be compatible with more databases. It is only required to expand the functions of the database manager to support more databases.

The database manager includes a command builder, connection watchdog, security, and availability checker, as shown in Figure 72. When the system requests to operate the database, the command builder will construct the database operation command according to the relevant operation requested and send it out through the connection that is established with the database. The result of the command execution will be transmitted to the event filter under the port listener to deal with. The connection watchdog is used to maintain the connection with the database. Any interruption of the connection with the database will cause the server to become unavailable. The connection watchdog will continue to monitor the connection status with the database. If the connection is unexpectedly terminated, the watchdog will attempt to re-establish the connection to the database, but if it still cannot connect to the database within a limited time, the connection watchdog will notify the entire server to enter the unavailable state. The security and availability checker is used to test the various states of the database and check the security configuration of the database. In some cases, the database may be accidentally reconfigured. At this time, the database server may lose the database or table, and the existing table may be changed. As a result, the database cannot satisfy the requirements of the server operation. For example, the loss of the system log table will render the server unable to save the system log, and the wrong security configuration may grant other users or programs permissions to access the database and table that should only be accessed by the management server. All these problems will reduce the stability of the server, so the checker will check the relevant information of the database after each time when connected to the database to ensure the security and availability of the database.

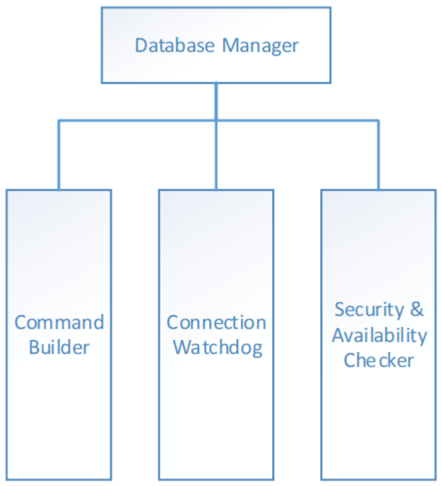


Figure 72: Architecture of data manager

5.3.3 Report Analyzer

The report analyzer processes all the work related to the report submitted by the control node. Regardless of how the data is uploaded by the data node changes, it only needs to expand the function of the report analyzer to be compatible with data reports in different formats. This design will enhance the flexibility of the server.

The report analyzer includes a report decoder, a checksum verifier, and a compliance checker, as shown in Figure 73. After the report decoder receives the report content that is extracted from the event analyzer under the port listener, the checksum verifier will first verify the checksum attached in the report. To ensure that the report that is transmitted remotely from the control node through the network is still complete and correct when received, the control node will attach a checksum after each report when submitting the report. By recalculating the checksum and comparing it with the attached one, the management server can determine whether the report is transmitted correctly. If the checksum cannot be verified, the report analyzer will send the verification failure notification to the event analyzer and notify the control node to retransmit. To reduce the amount of data transmitted on the network and to reduce network bandwidth usage and data traffic usage when using the cellular network, the control node may enable data compression function when submitting reports. The compressed data must be decompressed by the report decoder that will extract the reports systematically from the content and send them to the compliance checker to ensure that there is no unexpected data format. The data in the incorrect format may cause the database manager to fail to write the report into the database. For example, the temperature record should be a numerical value, and writing a string will cause an exception when writing to the database. The compliance checker will analyze the incoming records one by one. After ensuring that the database requirements are met, the compliance checker will hand the records to the database manager one by one, and the database manager is responsible for writing records to the database.

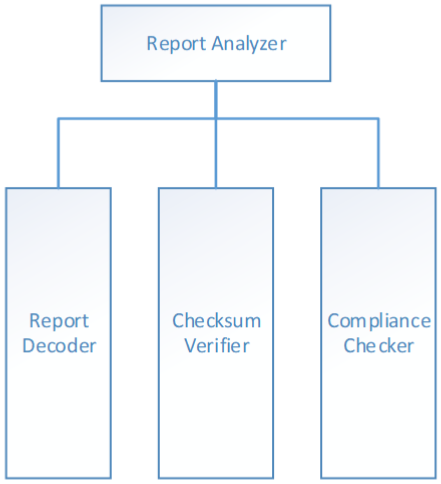


Figure 73: Architecture of report analyzer

5.3.4 Backup Scheduler

The backup scheduler is primarily used for the scheduled backup of system data to prevent the loss of critical data caused by the failure of the system. The backup scheduler also can restore the system. When the system is rebuilt or encounters an unrecoverable failure, the system can be restored from previous backups.

The backup scheduler includes a backup module and a recovery module, as shown in Figure 74. The backup module will continue to check the status of the backup storage, such as the remaining space of the storage. When the storage space is about to full, the backup module will delete the oldest backup or suspend the backup according to the backup strategy and notify the system. If it is a physical storage, the backup module will also check the health status of the physical storage device, such as the temperature of the device and the health status. When the storage status is in the warning state, the backup module will suspend the backup and allow the backup files to be migrated to the new storage to stop the backup from being be corrupted. The recovery module can be used to load data from the backup and restore the system to any previous time point when the backup has been created, but it should be noted that once the system is restored to a specific time point, all data generated since that backup time point will be lost. The restore module will not make any changes to the backup files, so the system can be switched to any backup time point. However, the backup module will be set to a suspended state after each restoration. Unsure that important information has been backed up before restarting the backup module. Once the backup module is restarted, to prevent conflicts between the backup files, all subsequent backups from the restored time point will be deleted, which will result in the permanent loss of subsequent backups, as shown in Figure 75.

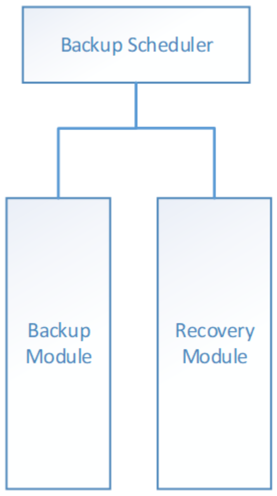


Figure 74: Architecture of backup scheduler

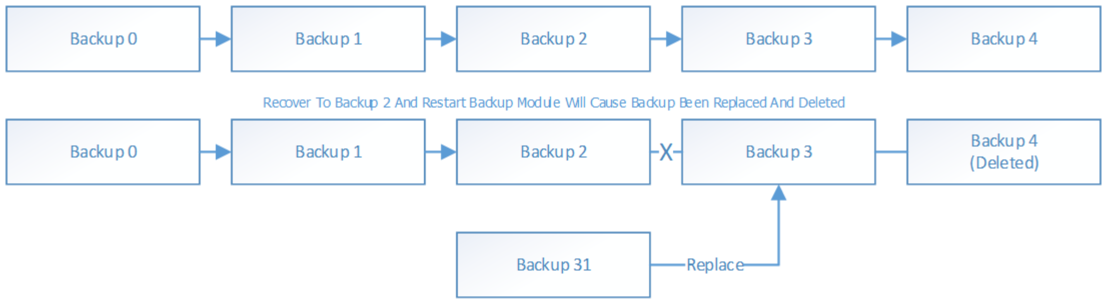


Figure 75: Process of recovery

5.3.5 Export Interface

The export interface is used to provide support for the development of third-party clients and plug-ins. The third-party clients can communicate with the server by calling the server API, or they can access the server through compatible third-party protocols. The export interface can help to enhance the system compatibility.

The export interface includes server API and third-party protocol support, as shown in Figure 76. The third-party client that supports the server API is not much different from the management client provided by the system in the function that operates the server. To the server, the primary difference is the identification of the client. The third-party client can perform secondary development to provide more functions based on the server API or use the server API to embed the client into more complex programs. For example, when developing data analysis software, the data can be collected directly from the server for subsequent analysis by using server API. The support for third-party protocol is to provide third-party protocol support for existing software. For example, some closed-source software or commercial software can import data through certain protocols or fixed formats, but users cannot modify such kind of software, the third-party protocol support is to provide active compatibility for such kind of software.

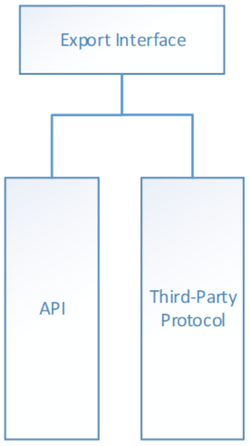


Figure 76: Architecture of export interface

5.4 Management Client Design

Because the server does not allow direct operation due to security considerations, and to meet the needs of multiple users to access the service at the same time, the management client is designed to work with the server to facilitate users to operate the server. The management client has a graphical interface for simplified operation. The management client needs to be able to establish communication with the server. The management client includes the following parts, node manager, report generator, user manager, system configurator, and node network configurator, as shown in Figure 77.

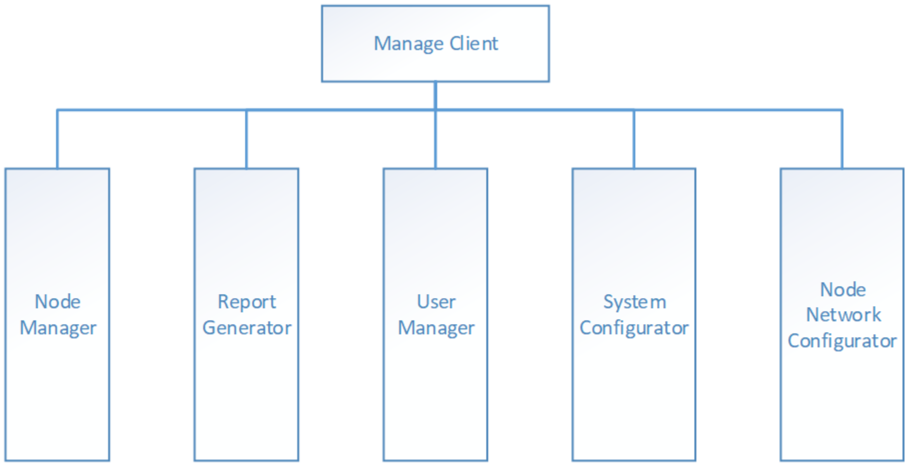


Figure 77: Architecture of management client

5.4.1 Node Manager

The node manager is responsible for all the work related to the node. Although the node communicates with the server during the communication process, the configuration of the node still needs to be done through the management client. The management client will communicate with the server at the same time during the process of configuring the node. The node information will be synchronized and uploaded to the server. The firmware of the node will also be distributed through the server, so the management client is not always offline.

The node manager includes node tester, node monitor, firmware upgrader, data downloader, and node configurator, as shown in Figure 78. The node tester can be used to test the health status of nodes in batches. Although the nodes will also perform software and hardware self-tests during the startup process, any unexpected software and hardware errors that cannot be handled will directly cause the node system to stop running. Thus, manual observation is required to determine the health status of the nodes, which may lead to mistakes in the checking process and require a certain training to perform the check, and cannot be tested in batches. The node tester can automatically check the nodes connected to the management client and generate a health report. The node tester will also detect the difference between the configuration file of the node and the hardware. When the configuration file declares a non-existent hardware module, the node tester will generate an error report. In the actual application of the node, the wrong configuration file may cause the node system to terminate unexpectedly. The node monitor is prepared for nodes that support offline use. By connecting the node directly to the management client, the node can be used as an external sensor, which provides convenience for on-site data collecting. The hardware of the node can be flexibly configured on site. Therefore, one node can satisfy most requirements of the on-site temporary data collecting and provides convenience during the process. The firmware upgrader can update the firmware of the node. To be compatible with more modules continuously, the firmware of the node will be continuously updated. The updated firmware will be submitted to the server, which will be pushed to each management client and can be downloaded to the node later. The data downloader is prepared for nodes with an offline communication interface. When the node is offline for an extended period or suffers accidental damage and cannot function normally, the data in the on-board flash memory can be downloaded by connecting the node to the management client. Moreover, when the node is not allowed to enter the online mode, the download manager can download data through a wired communication method. The download manager can also download data from multiple nodes at the same time, which will save substantial download time compared to a single-point communication method such as Bluetooth. The node configurator can configure the information of the node, whether it is the on-board device statement or the node network-related information. The configuration process can be completed through the node configurator. The generated configuration information will be written into the node, and the node configurator can read and backup the configuration information from existing nodes. Moreover, the node configurator can also configure multiple nodes at the same time, and configuration operations, such as configuring multiple nodes at the same time by using a node as a template, can be completed through the node configurator.

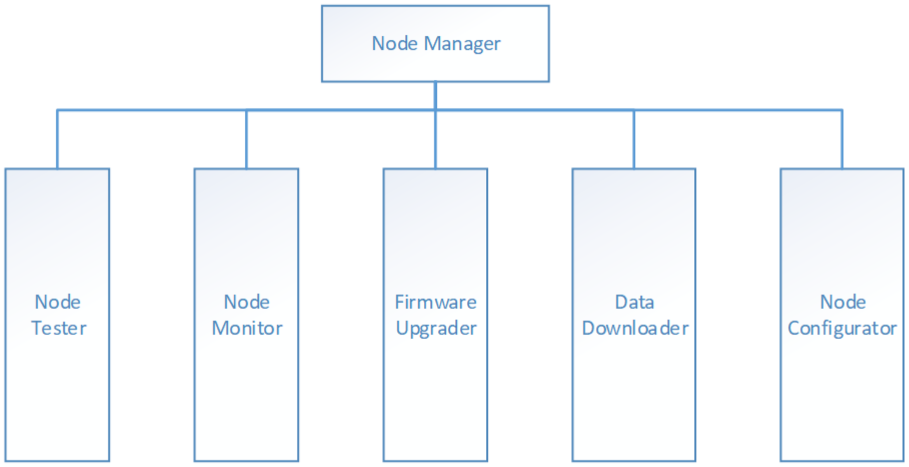


Figure 78: Architecture of node manager

5.4.2 Report Generator

The report generator is responsible for exporting all the information generated or collected by the system in the form of reports. Some data can be exported in the form of charts, and the rest of the information will be exported in the form of texts. The source of the exported information is provided by the server. The data that downloaded from the data node connected to the management client through wired connection needs to be uploaded to the server. After being processed by the server, the records can be exported through the report generator.

The report generator includes a data plotter, a log exporter, and a node state exporter, as shown in Figure 79. The data plotter is mainly used to process the environmental information collected by the node. The data plotter can draw a line chart in the form of periods for the data collected from the server. The highest, lowest, and average value can be automatically calculated and drawn on the chart and exported with the report together. The log exporter is used to export-related logs generated during the system operation, such as user login records, system health check results, database running status, system storage status, and other related logs, as well as the load changes during server running and other information. This type of report can only be exported by the user who has suitable permission, the report can be used to analyze system status and safety status, and provide references when performing system maintenance. The node state exporter is used to generate a report on the status of the node that is registered in the system. The report content includes the previous data submitting the time point of the node, the firmware version, and the hardware type of the node, serial number, and network configuration information. Through this report, the amount and status of nodes inside each note network can be observed directly, and the report can provide a reference for further adjustment during the node deployment process.

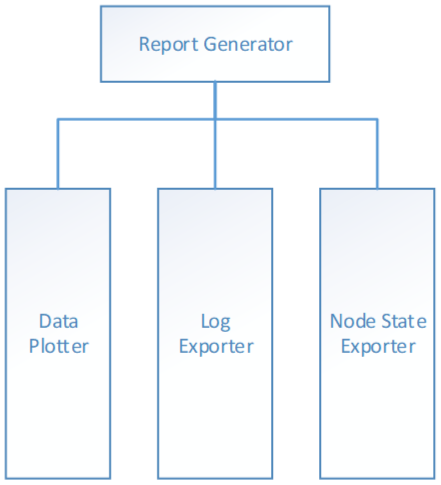


Figure 79: Architecture of report generator

5.4.3 User Manager

The user manager is used to manage the users and user groups that exist in the system. All users who access the system need to log in through the management client before they can use various functions. There may be multiple users in a system at the same time. To apply different restrictions on different users of the system, there are various operation permissions in the system. Different permissions can be applied to users located at different levels by configuring user groups and add users to user groups. After the initial configuration of the system, a temporary highest-privileged user will be enabled by default. After the first login, the system administrator needs to create a dedicated administrator account as soon as possible and log in with the new account. The temporary highest-privileged user will be automatically disabled after the new administrator logs in.

The user manager includes two parts: the group manager and the user manager, as shown in Figure 80. The group manager is used to simplify the configuration of permissions for the user in the same group. The system administrator can create different user groups, assign different system permissions to different groups, and then add users that have the same permissions to the same user group to complete the batch configuration of permissions for the user. For a system that has nodes that use online mode, dedicated users and user groups are required in the system to allow the node to submit data, and submission permission is required by these users and user groups. In the user group manager, different permissions that allowed to be granted in the system and their descriptions can be checked here. The user manager is used to create different users in the system. Users can be added, deleted, and modified. The activation and deactivation of users can also be modified here. User group members that have permission to modify other users can manage other users; otherwise; users only modify the information related to themselves in the user manager, and the user's login password modification is also completed in the user manager.

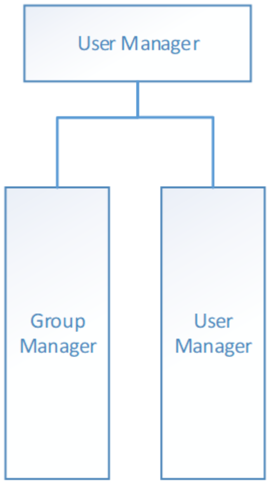


Figure 80: Architecture of user manager

5.4.4 System Configurator

The system configurator is used to command the server by constructing related commands and completing the related configuration of the server through the management client.

The system configurator includes a server commander and a server configurator, as shown in Figure 81. The server commander is used to submit command execution requests to the server, such as log cleanup, manual backup, server data restoration, server shutdown, and other operations. The server configurator is used to set the security level of the server, and the related operating parameters of the server, such as a backup period and backup target.

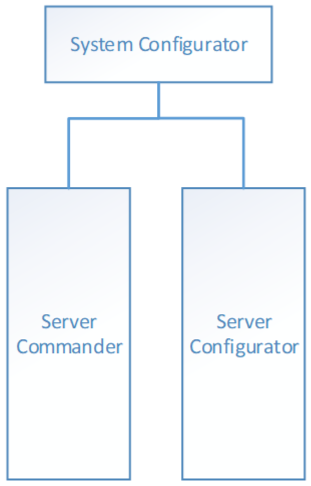


Figure 81: Architecture of system configurator

5.4.5 Node Network Configurator

The node network configurator is used to configure the relevant information of a network as a whole. The node manager is the relevant settings in the node network configurator that affect an entire node network, but the node manager configures specific nodes. When the node enters the online mode, builds up a node network, and can establish a connection with the server through the control node, the node manager will no longer be able to configure the node, and all online configurations will be performed in the manner of the node network.

The node network configurator includes a node network commander and a node network director, as shown in Figure 82. The node network commander is used to send commands to the control node. The command will first be sent to the server by the management client and then pushed to the control node of the corresponding network by the server. After receiving the relevant commands, the control node will broadcast the command to the data node over the node network when the node network is built up next time. Thus, the longer the synchronization interval, the longer it takes commands to be delivered to each data node. Normally, very few commands require each data node to participate during the execution process.

Generally, this situation may occur when the data node network is at risk, such as due to the leakage of password combinations, and the data-node installation area does not allow to be entered immediately. At the time, the new password combination can be pushed to each data node remotely through online configurations. After the new password combination reaches each data node, the control node will notify all data nodes to switch to the new password combination and then notify the server that the command execution is complete, which may require multiple instances of node network building to complete the switch. Since the control node is always online, most of the commands that only need the control node to respond will be executed once received, without waiting for the buildup cycle of the node network to commence. The node network director is used to create different node network parameters, such as prefix and password combinations, to generate node network configurations. These statements will be stored on the server and can be used to download to different data nodes in the node manager for networking configuration.

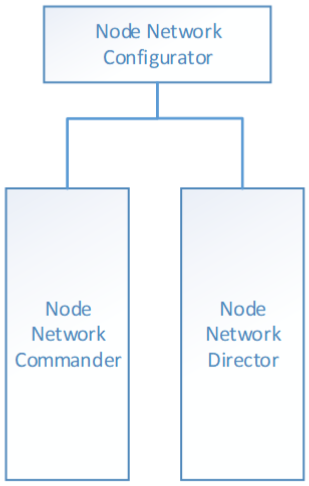


Figure 82: Architecture of node network configurator

5.5 Control Node Software Design

The control node is the bridge between the data node network and the server. Although the network is created based on Wi-Fi between data nodes, the data node network cannot be connected to the server directly because the node network is not constructed in a traditional central network mode. The control node inserts itself into the data node network, collects data in the data node network, and communicates with the server through the standard network connection methods that are compatible with the control node such as Ethernet, standard Wi-Fi network, and cellular network. Moreover, it forwards the data that is collected in the node network to the server, so the control node has to consume substantial power and cannot be compatible with energy-saving strategies as the data node is. However, due to the presence of the control node, the data node can remove a large amount of hardware that is not related to data collection and thus save energy by placing the system into sleep mode.

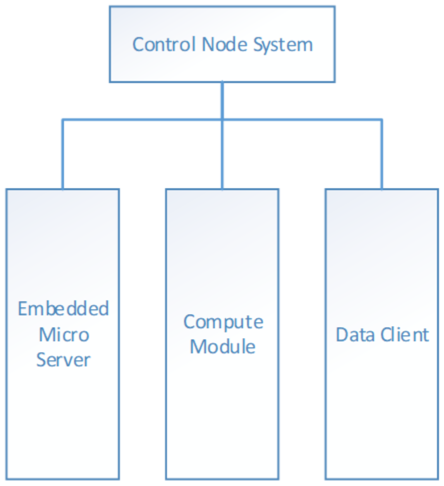


Figure 83: Architecture of control node software

The program for the control node includes an embedded microserver, computing module, and data client, as shown in Figure 83. The embedded microserver is used as a network center in the data node network. All data nodes will be configured to communicate with the embedded micro server of the control node, and the data will be collected by the embedded microserver. When a command issued from the management client requires a data node or all data nodes to respond, the related command will be forwarded by the embedded microserver. The calculation module is used for operations such as data statistics and sorting compression. When there is a need for mathematical calculations, the calculation can also be completed on the control node, and the control node can run a variety of scripting languages. Since the control node needs to submit data to the remote server, it needs to act as a client. The data client is used to submit data to the remote server. The data client needs a system account with data submission permission to complete the data submission process. To prevent security issues, the account of the data client should not have other system management permissions, and each control node should have a different data submission account.

Chapter 6

Cost and Performance Evaluation

The main purpose of this chapter is to evaluate the sensor platform and its operating status.

6.1 Hardware Cost

The embedded microcontroller uses a 32-bit 48-pin ARM core chip, the full speed of the CPU runs at 72 MHz, 20 KB RAM is available in total, the price is 1 USD. The Wi-Fi part selects a full-featured module, including an on-chip antenna and radio frequency circuit that can provide complete hardware support, the firmware of the Wi-Fi module needs to be developed and programmed, the price is 1.5 USD. The storage chip selects a flash memory that has a capacity of 128 Mb, can provide 16 MB of storage space in total, each memory chip costs 0.5 USD, the standard version node requires two chips, cost 1 USD in total. The high-precision high-speed oscillator and low-speed oscillators are used to provide clocks to support the running of controller and peripherals, the total cost of which is 1 USD. Peripheral electronic components, such as capacitor, resistor, LED, and USB interface, cost 2 USD in total. The battery power module costs 2 USD. The PCB board is produced by the factory and the price is 1 USD. The soldering process of the components costs 1 USD. The sensor module costs 1.5 USD. The lithium battery costs 3 USD. Based on the above materials, the total cost of three design versions and an environmental sensor is shown in Table 12.

Table 12: Costs of different board types

Board Type	Cost
Standard version	15 USD
Mini version	14.2 USD
Micro version	13.6 USD

6.2 Hardware Dimensions

The standard version has the largest board size since the standard version design contains all the functions, the dimensions are 54.23 mm long, 34.42 mm wide, and the area of the board is 1866.6 mm^2. The design is shown in Figure 84.

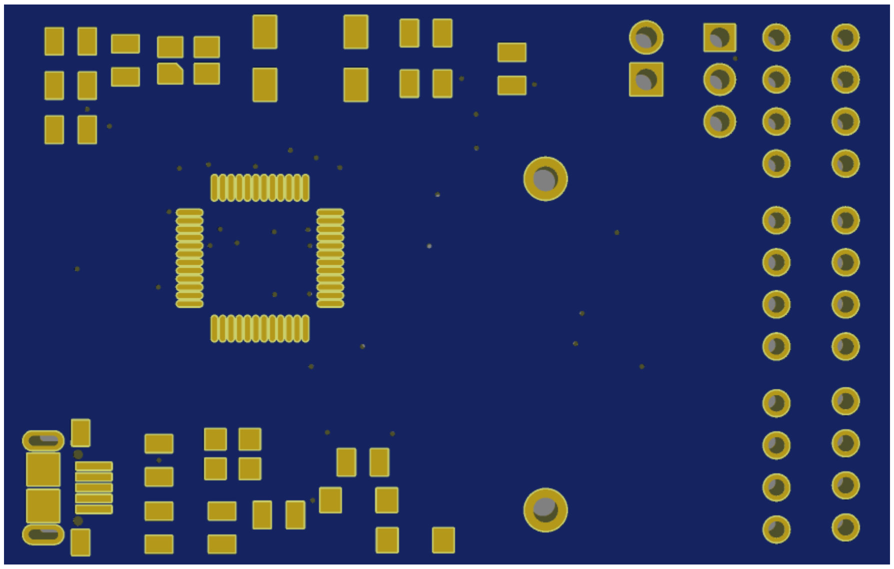


Figure 84: Design of standard board

The mini version has reduced the number of components due to the removal of some functions, and the size of the motherboard has been reduced. The dimensions are 38.87 mm long, 32.52 mm wide, and the area of the board is 1264.06 mm^2. The design is shown in Figure 85.

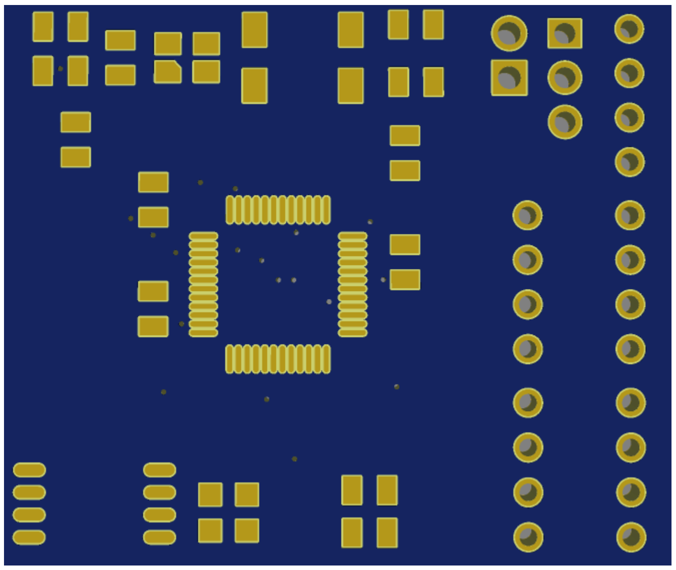


Figure 85: Design of mini board

The micro version removed more functions, so the amount of components is further reduced, and the main board size is further reduced. The dimensions are 31.75 mm long, 29.34 mm wide, and the area of the board is 931.55 mm^2. The design is shown in Figure 86.

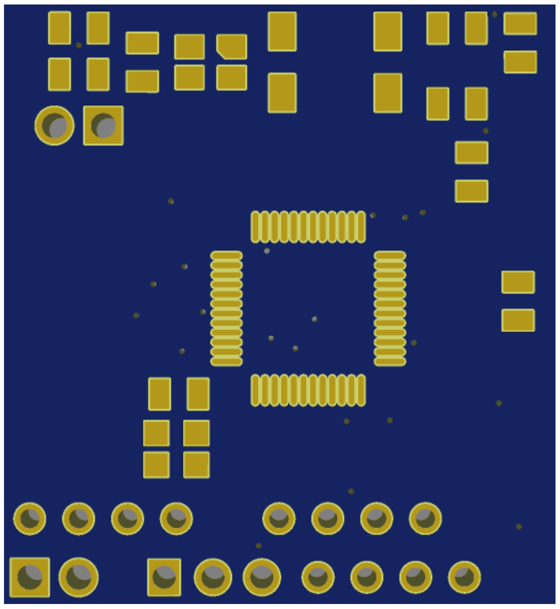


Figure 86: Design of micro board

Table 13: Board area comparisons

Board Type	Board area compared with Raspberry Pi 3	Board area compared with Arduino UNO
Standard version	39.3%	51.1%
Mini version	26.6%	34.6%
Micro version	19.6%	25.5%

The physical dimensions of Raspberry Pi 3 are 85 mm X 56 mm, the area of the board is 4760 mm^2, the physical dimensions of Arduino UNO is 68.6mm X 53.3mm, the area of the board is 3656.38 mm^2. In contrast, for the self-designed platform, the area of the board of the standard version is reduced by 60.7% compared with Raspberry Pi 3 and is reduced by 48.9% compared with Arduino UNO. The area of the board of the mini version is reduced by 73.4% compared with Raspberry Pi 3 and is reduced by 65.4% compared with Arduino UNO. The area of the board of the micro version is reduced by 80.4% compared with Raspberry Pi 3 and is reduced by 74.5% compared with Arduino UNO. The comparison is shown in Table 13.

6.3 Power Consumption

The data nodes will be powered by lithium batteries. To reduce the battery replacement time and extend the running time of the platform, the platform uses a low-power design during the design process. The power consumption test of the platform is evaluated by the power meter and related calculations. The actual energy consumption and battery life may vary due to the communication period defined in the system configuration.

The test is performed by connecting a power meter in series with the USB power supply, then connecting the power meter in series with a transformer, and the transformer will directly provide power supply to the platform.

First, disconnect the platform from the transformer, and the measured power of the transformer is 0.0425 watts, as shown in Figure 87.

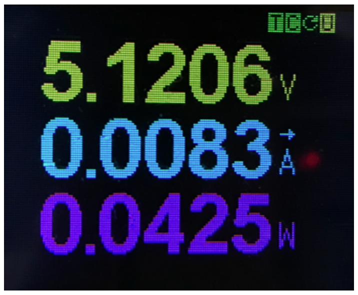


Figure 87: Power of transformer

After that, connect the platform to the transformer. After the platform is initialized, the overall power of the platform is 0.5506 watts before the platform enters the sleep mode, as shown in Figure 88. Subtract the transformer power to get the power is 0.5506-0.0425=0.5081 watts when the platform is in running mode.

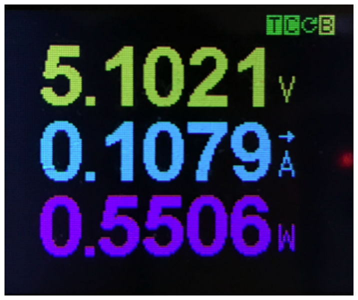


Figure 88: Total power of platform @ running

Waiting for the platform to finish the process of environment data collection and enter the sleep mode, the overall power at this time is 0.0446 watts, as shown in Figure 89, and the power of the platform in the sleep mode is 0.0446-0.0425=0.0021 watts after subtracting the transformer power.

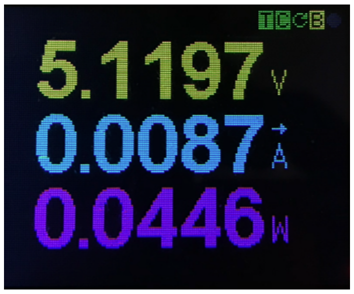


Figure 89: Total power of platform @ sleep

Continue to wait for several data collection periods are finished, until the platform entering the node-network construction stage. After the node joined the node network, the overall power at this time is 0.6788 watts, as shown in Figure 90. Subtract the transformer power to get the power is 0.6788-0.0425=0.6363 watts when the platform is in running mode after joining the network.

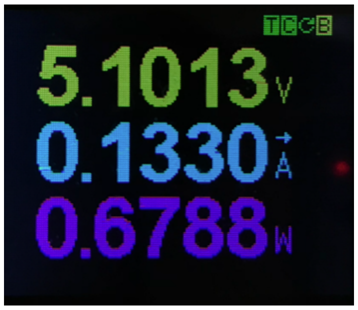


Figure 90: Total power of platform @ joined network

When the node joined the node network and needs to send and receive data, the platform reaches the stage of maximum power at this time, the overall power is 0.8142 watts, as shown in Figure 91, subtracting the transformer power to obtain the power is 0.8142-0.0425=0.7717 watts when the platform is sending and receiving data.

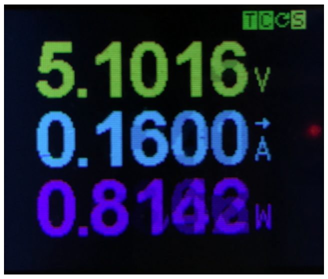


Figure 91: Total power of platform @ send and receive data

The platform is not always sending and receiving data. Because the data is sent and received individually in the form of packets, the duration of the data sending and receiving process is relatively short, as shown in Figure 92, and the duration is about 20 ms for each data packet. Only during this period, the platform will reach maximum power consumption. Therefore, it is necessary to balance the interval of data submission and energy consumption when configuring the period of data submission.

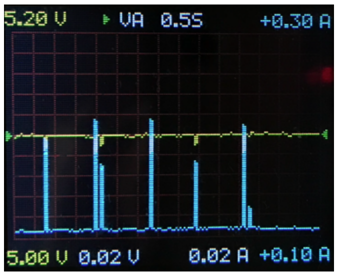


Figure 92: Peak current @ send and receive data

Based on the above data, the average current of the platform can be estimated when the system is powered by the lithium battery. The typical charging voltage of the lithium battery is 4.2 v, and the discharge process will start from 4.2 v when fully charged. The voltage gradually decreases with the discharge process of the battery. Generally, the lithium battery can be safely discharged until the voltage is reduced to 2.5 v, but when the voltage is too low, it will not be able to satisfy the voltage requirement for the platform to run normally. To protect the battery, the platform will stop running and cut off the system power when the battery voltage drops to

3.75 v.

For a qualified lithium battery, 90% of the battery power should have been released at this time. Since the power consumption is constant, the current will increase with the decrease of the battery voltage when discharging, the average value of the upper and lower limits of the discharge voltage is taken to evaluate the platform current. The average voltage is (4.2+3.75)/2=3.975 v, according to the calculation, the average current during running is I=P/U=0.5081/3.975=0.1278a, the average current during sleep is I=0.0021/3.975=0.000529a, and the average current after the platform joins the network is I=0.6363/3.975= 0.1601a, the average current when the platform sends and receives data is I=0.7717/3.975=0.1941a.

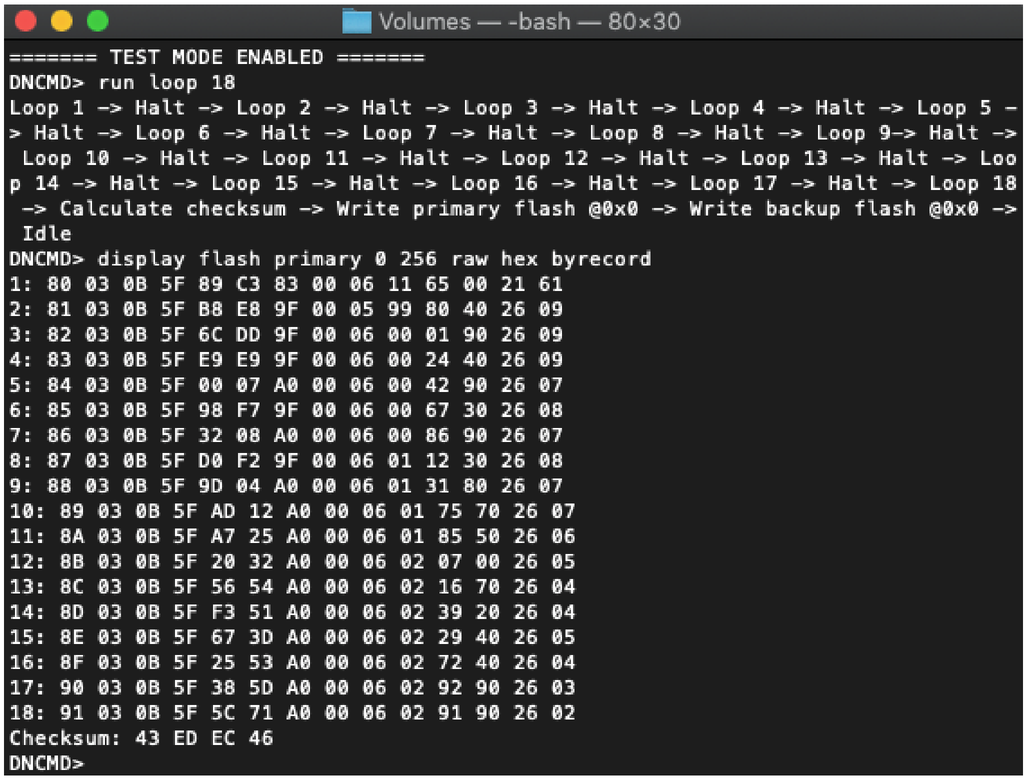
6.4 Battery Life

According to the average current, the battery life of the platform when running under the condition of lithium battery power supply can be further estimated. The capacity of 18650 lithium battery is between 1800 mAh and 2600 mAh, and the average value is (1800+2600)/2=2200 mAh, and the 90% capacity is 2200*90%=1980 mAh. When the discharge voltage is constant, the battery can be discharged for one hour at a current of 1980 milliamperes. If the system remains in running state, one 18650-lithium battery can provide about 1980/127.8=15.493 hours of battery life. If the system remains in sleeping mode, one lithium battery can provide about 1980/0.529=3742.92 hours of battery life. If the system keeps running after joined the node network, one lithium battery can provide about 1980/160.1=12.367 hours of battery life. If the system keeps sending and receiving data packets, one lithium battery can provide about 1980/194.1=10.201 hours of battery life. Since the duration of sending and receiving each data packet is about 20 ms, one lithium battery can be used to send and receive about 10.201*3600/0.02=1836180 data packets.

Based on the above data, the approximate battery life of the system when running under different configurations can be evaluated. If the environmental data will be collected when the system time is a multiple of ten minutes, such as 12:00, 12:10, 12:20, etc., the data needs to be collected six times per hour, the data needs to be collected 24*6=144 times every day. The system needs 500 ms to finish the collection process each time, and the system needs to write to the flash memory after every 18 instances of data collection, so it needs to write 144/18=8 times per day, and the system needs 1.5 s to write to the flash each time. Thus, the system will continue in the running state for 144*0.5+8*1.5=84 s per day, the battery needs to provide 127.8*84/3600=2.982 mAh. Assuming that the system will upload data through the network once a day, and keep the node network online for 5 minutes every time to establish a network, upload data, and forward data for other nodes. Each complete report requires three data packets to finish the sending and confirmation process, which requires 3*8 = 24 data packets to complete the data upload process for one data node. Assuming there are 20 nodes in the network, there are theoretically at about 24*20=480 data packets that need to be transferred for one data node during the network synchronization process. Assuming that the communication quality is poor and there are two retransmissions, in the worst case, there may be 480*3=1440 data packets that need to be transferred. The total running time of data sending and receiving is 1440*0.02=28.8 s, and the period during which the system joined the node network but not transfer data is about 5*60-28.8=271.2 s. The lithium battery is required to provide 160.1*271.2/3600 +194.1*28.8/3600=13.614 mAh. Therefore, in the case of collecting data every 10 minutes and uploading data once a day, a total of 13.614+2.982=16.596 mAh of power is required per day for lithium batteries to provide. In this configuration mode, one lithium battery can provide about 1980 /16.596=119.305 days of battery life.

6.5 Storage

To check the status of the data storage, a data node that contains a test mode and is only for system-function testing and verification is used. The system is tested by placing it in test mode and manually operating the node system. First, let the system run for 18 cycles. Every time the system is started, the task is completed, and it enters the sleep mode is regarded as one cycle. Eighteen cycles are required to run because the system writes to the flash memory once after 18 cycles. According to the log, the system writes data to the main flash memory and backup flash memory after 18 cycles at position 0. Then it commands the system to display the data recorded in the primary flash memory from position 0 to position 256, as shown in Figure 93. The records will be displayed in raw data in hexadecimal format and split individually. A total of 18 records and checksum are displayed. The system is set to collect data every 1 second. Therefore, the last byte of the time of each record will gradually increase from 0x80 to 0x91. Each block has a total of 18 records and one check code, totaling 256 bytes, which is 36 bytes shorter than the 292 bytes original data in the buffer and 252 bytes shorter than the 508 bytes original data in RAM.



```
======= TEST MODE ENABLED =======
DNCMD> run loop 18
Loop 1 -> Halt -> Loop 2 -> Halt -> Loop 3 -> Halt -> Loop 4 -> Halt -> Loop 5 -
> Halt -> Loop 6 -> Halt -> Loop 7 -> Halt -> Loop 8 -> Halt -> Loop 9-> Halt ->
 Loop 10 -> Halt -> Loop 11 -> Halt -> Loop 12 -> Halt -> Loop 13 -> Halt -> Loo
p 14 -> Halt -> Loop 15 -> Halt -> Loop 16 -> Halt -> Loop 17 -> Halt -> Loop 18
 -> Calculate checksum -> Write primary flash @0x0 -> Write backup flash @0x0 ->
 Idle
DNCMD> display flash primary 0 256 raw hex byrecord
1: 80 03 0B 5F 89 C3 83 00 06 11 65 00 21 61
2: 81 03 0B 5F B8 E8 9F 00 05 99 80 40 26 09
3: 82 03 0B 5F 6C DD 9F 00 06 00 01 90 26 09
4: 83 03 0B 5F E9 E9 9F 00 06 00 24 40 26 09
5: 84 03 0B 5F 00 07 A0 00 06 00 42 90 26 07
6: 85 03 0B 5F 98 F7 9F 00 06 00 67 30 26 08
7: 86 03 0B 5F 32 08 A0 00 06 00 86 90 26 07
8: 87 03 0B 5F D0 F2 9F 00 06 01 12 30 26 08
9: 88 03 0B 5F 9D 04 A0 00 06 01 31 80 26 07
10: 89 03 0B 5F AD 12 A0 00 06 01 75 70 26 07
11: 8A 03 0B 5F A7 25 A0 00 06 01 85 50 26 06
12: 8B 03 0B 5F 20 32 A0 00 06 02 07 00 26 05
13: 8C 03 0B 5F 56 54 A0 00 06 02 16 70 26 04
14: 8D 03 0B 5F F3 51 A0 00 06 02 39 20 26 04
15: 8E 03 0B 5F 67 3D A0 00 06 02 29 40 26 05
16: 8F 03 0B 5F 25 53 A0 00 06 02 72 40 26 04
17: 90 03 0B 5F 38 5D A0 00 06 02 92 90 26 03
18: 91 03 0B 5F 5C 71 A0 00 06 02 91 90 26 02
Checksum: 43 ED EC 46
DNCMD>
```

Figure 93: Raw data in flash memory

6.6 Communication

The communication test is performed on a public site. The structure of the test site is shown in Figure 94. The dark green point is the location of the control node and the light green point is the location of the data node. There is a closed area X between B2 and B3 where one cannot enter to install the data node. Thus, a data node is installed on the corridor wall between B2 and B3, the main purpose of which is to help B2 and B3 to form a stronger node network. A2 is a large conference room. Since the human traffic may be high and the communication signals between data nodes may be blocked by human activities, an additional data node will be installed on the corridor wall of the conference room to enhance the stability of the network. Because the control node needs to be continuously online and will consume significant power, it is deployed in a power switch room that can supply power to the control node for an extended period. As the door of the power switch room is normally closed and the metal door will block the wireless signal, an additional data node will be installed near the door of the power switch room to assist network communication, and data upload to the server is completed through the cellular network connection established by the control node.

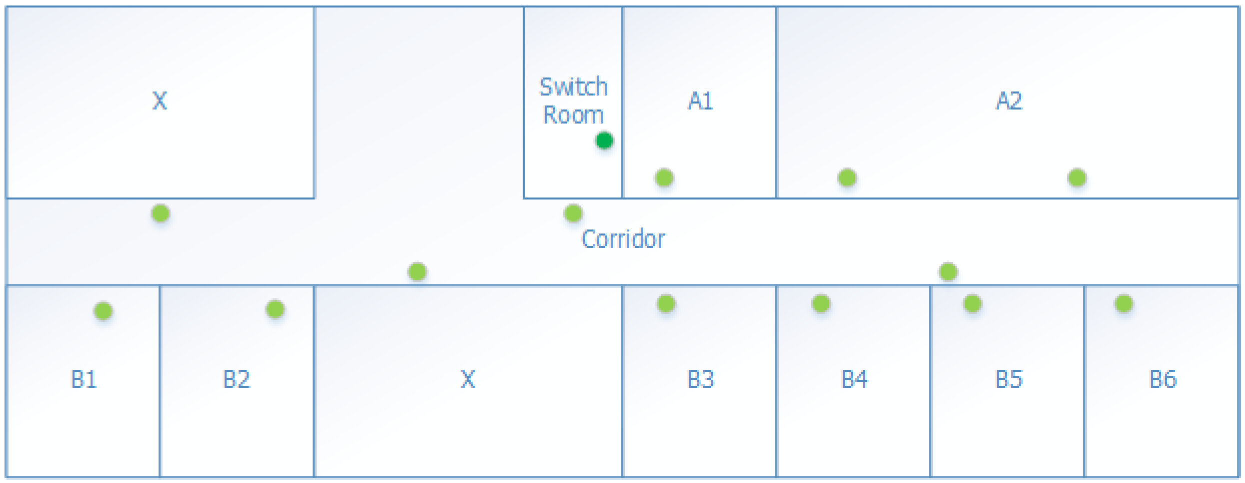
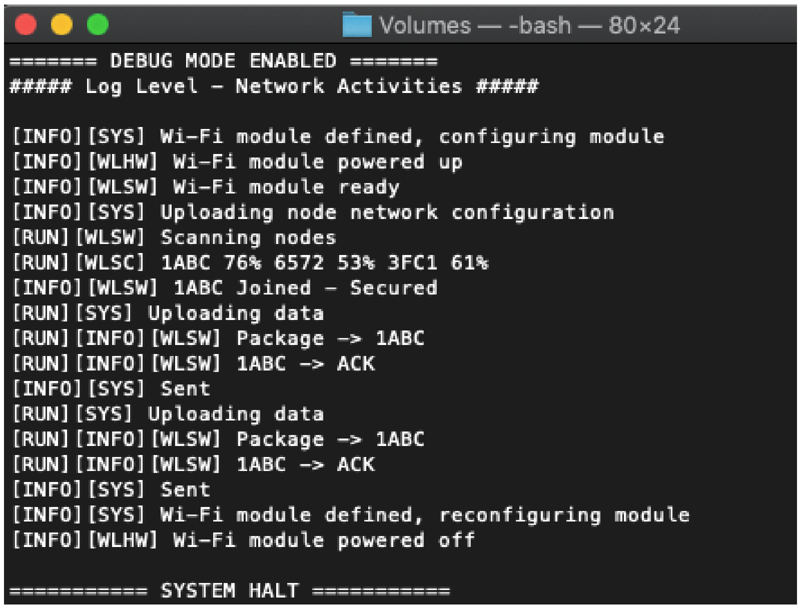


Figure 94: Structure and distribution of nodes @ test place

The nodes in rooms B1 and B6 are located at the far end of the spatial distribution, and the node located at the corridor wall of the power switch room is located at the nearest end of the spatial distribution. Therefore, these data nodes can be set to debug mode and capture logs for analysis. First, check the B1 debugging logs, as shown in Figure 95. It can be observed that the data node in room B1 first opened the Wi-Fi module according to the configuration and tested the module. After receiving the ready signal, the module scanned other data nodes and found three nodes nearby and the corresponding signal strength. It then selected the data node with the strongest signal strength and joined the node network, after which the system started to send data packets to the control node. The data packets are handed over to the 1ABC node, which is the node on the opposite corridor wall to B1. After receiving the response that confirmed that the data has arrived, the next data packet is sent. After a timeout, the Wi-Fi module will be powered off, the system will enter sleep mode, and the debug log ends. Since B6 is also located at the edge of the network, it is similar to the B1 debug log, as shown in Figure 96.

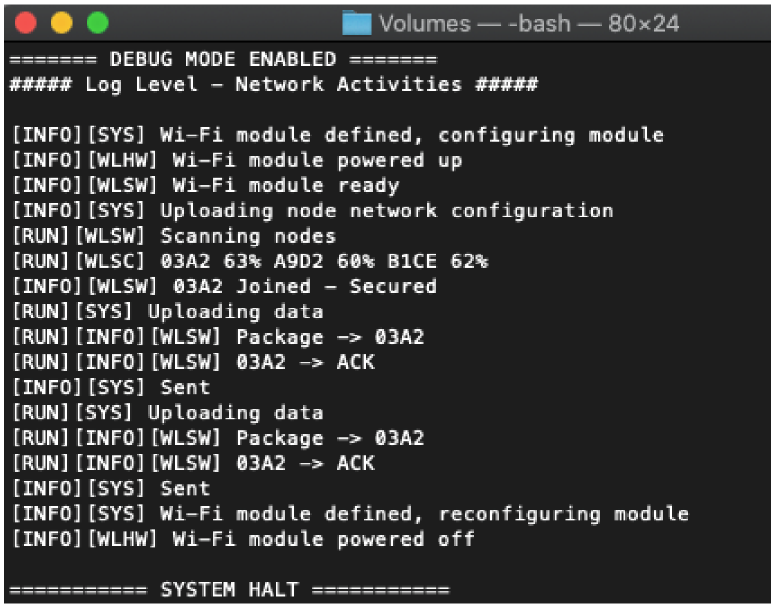


```
======= DEBUG MODE ENABLED =======
##### Log Level - Network Activities #####

[INFO][SYS] Wi-Fi module defined, configuring module
[INFO][WLHW] Wi-Fi module powered up
[INFO][WLSW] Wi-Fi module ready
[INFO][SYS] Uploading node network configuration
[RUN][WLSW] Scanning nodes
[RUN][WLSC] 1ABC 76% 6572 53% 3FC1 61%
[INFO][WLSW] 1ABC Joined - Secured
[RUN][SYS] Uploading data
[RUN][INFO][WLSW] Package -> 1ABC
[RUN][INFO][WLSW] 1ABC -> ACK
[INFO][SYS] Sent
[RUN][SYS] Uploading data
[RUN][INFO][WLSW] Package -> 1ABC
[RUN][INFO][WLSW] 1ABC -> ACK
[INFO][SYS] Sent
[INFO][SYS] Wi-Fi module defined, reconfiguring module
[INFO][WLHW] Wi-Fi module powered off

=========== SYSTEM HALT ===========
```

Figure 95: Debug log of B1 node



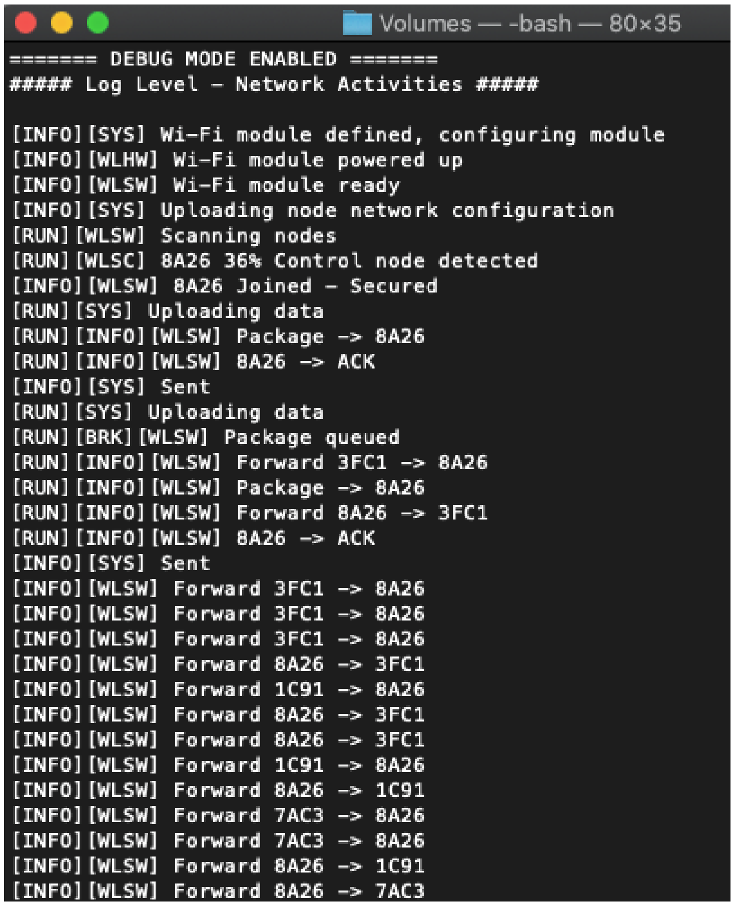
```
======= DEBUG MODE ENABLED =======
##### Log Level - Network Activities #####

[INFO][SYS] Wi-Fi module defined, configuring module
[INFO][WLHW] Wi-Fi module powered up
[INFO][WLSW] Wi-Fi module ready
[INFO][SYS] Uploading node network configuration
[RUN][WLSW] Scanning nodes
[RUN][WLSC] 03A2 63% A9D2 60% B1CE 62%
[INFO][WLSW] 03A2 Joined - Secured
[RUN][SYS] Uploading data
[RUN][INFO][WLSW] Package -> 03A2
[RUN][INFO][WLSW] 03A2 -> ACK
[INFO][SYS] Sent
[RUN][SYS] Uploading data
[RUN][INFO][WLSW] Package -> 03A2
[RUN][INFO][WLSW] 03A2 -> ACK
[INFO][SYS] Sent
[INFO][SYS] Wi-Fi module defined, reconfiguring module
[INFO][WLHW] Wi-Fi module powered off

=========== SYSTEM HALT ===========
```

Figure 96: Debug log of B6 node

The data nodes outside the power switch room need to forward all the data packets of the entire node network, so the debugging log is too long. Only some representative logs are intercepted for analysis here, as shown in Figure 97. The difference from the previous nodes is that the data nodes outside the power switch room can detect the control node, although the signal strength is weak. After detecting the control node and confirming that the signal is strong enough to establish the connection, the system will first send its own data. This is the same as the case of the first two nodes. However, when it is preparing to send the next data packet, other nodes have forwarded data packets to this node at this time. To avoid holding too many data packets from other nodes, the local data packets that have not been sent will be placed in the queue, and the data packets forwarded by other nodes will be sent first. The local data packets in the queue will be sent, and then the reply from the control node will be forwarded to other data nodes. After all the local data packets are sent, the node will focus on forwarding data packets from other nodes. In the intercepted log, it can be determined that at least three nodes are connected to this node and have forwarded data, and this node will forward the replies from the control node to other nodes, one by one.



```
======= DEBUG MODE ENABLED =======
##### Log Level - Network Activities #####

[INFO][SYS] Wi-Fi module defined, configuring module
[INFO][WLHW] Wi-Fi module powered up
[INFO][WLSW] Wi-Fi module ready
[INFO][SYS] Uploading node network configuration
[RUN][WLSW] Scanning nodes
[RUN][WLSC] 8A26 36% Control node detected
[INFO][WLSW] 8A26 Joined - Secured
[RUN][SYS] Uploading data
[RUN][INFO][WLSW] Package -> 8A26
[RUN][INFO][WLSW] 8A26 -> ACK
[INFO][SYS] Sent
[RUN][SYS] Uploading data
[RUN][BRK][WLSW] Package queued
[RUN][INFO][WLSW] Forward 3FC1 -> 8A26
[RUN][INFO][WLSW] Package -> 8A26
[RUN][INFO][WLSW] Forward 8A26 -> 3FC1
[RUN][INFO][WLSW] 8A26 -> ACK
[INFO][SYS] Sent
[INFO][WLSW] Forward 3FC1 -> 8A26
[INFO][WLSW] Forward 3FC1 -> 8A26
[INFO][WLSW] Forward 3FC1 -> 8A26
[INFO][WLSW] Forward 8A26 -> 3FC1
[INFO][WLSW] Forward 1C91 -> 8A26
[INFO][WLSW] Forward 8A26 -> 3FC1
[INFO][WLSW] Forward 8A26 -> 3FC1
[INFO][WLSW] Forward 1C91 -> 8A26
[INFO][WLSW] Forward 8A26 -> 1C91
[INFO][WLSW] Forward 7AC3 -> 8A26
[INFO][WLSW] Forward 7AC3 -> 8A26
[INFO][WLSW] Forward 8A26 -> 1C91
[INFO][WLSW] Forward 8A26 -> 7AC3
```

Figure 97: Part of the debug log of the switch-room node

6.7 Information Management System

In this section, part of the functions and interfaces of the information management systems are shown.

6.7.1 System Login

The system login interface is the first to be displayed when the management client launches, as shown in Figure 98. After the user enters relevant information and presses login, the program will calculate the password to an MD5 value and send it to the server to avoid plaintext transmission on the network. If the login is successful, the user can enter the management interface. If login fails, a login failed message box will pop up.

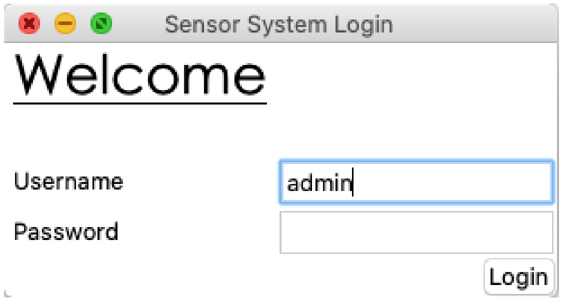


Figure 98: Login interface of client

6.7.2 Management Client Main Interface

According to the system design chapter, the main interface of the system includes node manager, report manager, user manager, system configurator, and node network configurator, which correspond to different functions, as shown in Figure 99, and can be clicked to enter. If the user does not have permission to access the function, the access buttons for related functions will not appear. The permission of the user will be returned by the server after login confirmation.

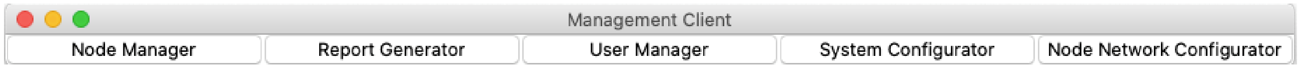
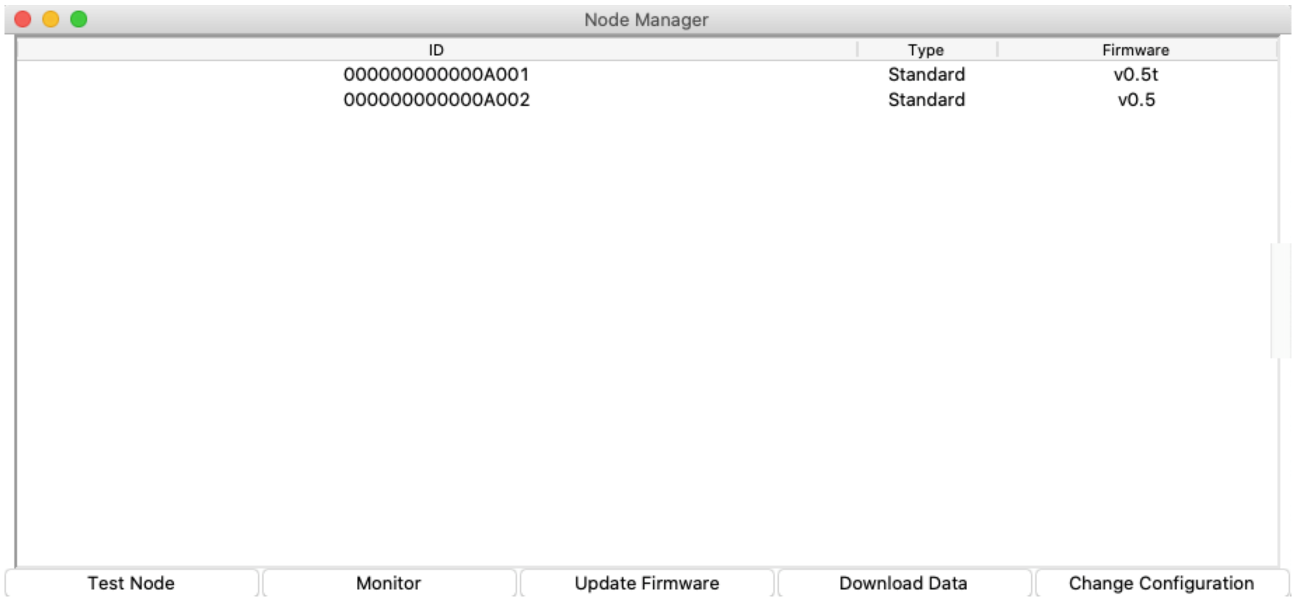


Figure 99: Main interface of client

6.7.3 Node Manager

The node manager will automatically refresh the nodes connected to the management client, and the detected nodes will be displayed in the list, as shown in Figure 100. The relevant information includes node ID, node board type, and node firmware version.

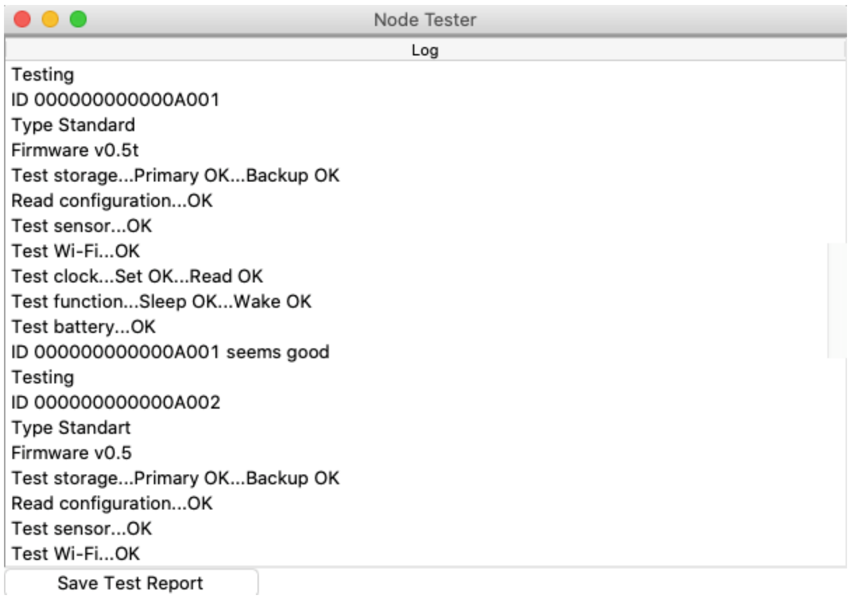


ID	Type	Firmware
000000000000A001	Standard	v0.5t
000000000000A002	Standard	v0.5

Figure 100: Node manager interface of client

6.7.4 Node Tester

After selecting the node in the list and pressing the test button, the system will automatically test the node. The relevant information and test results of the node will be displayed in the log window, and the user can save the test log to a file after the test, as shown in Figure 101.



```
Testing
ID 000000000000A001
Type Standard
Firmware v0.5t
Test storage...Primary OK...Backup OK
Read configuration...OK
Test sensor...OK
Test Wi-Fi...OK
Test clock...Set OK...Read OK
Test function...Sleep OK...Wake OK
Test battery...OK
ID 000000000000A001 seems good
Testing
ID 000000000000A002
Type Standart
Firmware v0.5
Test storage...Primary OK...Backup OK
Read configuration...OK
Test sensor...OK
Test Wi-Fi...OK
```

Figure 101: Node tester interface of client

6.7.5 Monitor

The monitor can read data from the supported sensor nodes in real-time and automatically update the display gradually, as shown in Figure 102. The figure shows the three data sets of temperature, humidity, and pressure. Moreover, it can translate the data into points on a graph. The horizontal axis is the time in the period, and the vertical axis is the corresponding data scale, The red dot is the node data when data is collected, and the blue line is the periodically update and the old data will be replaced by the new data. However, the old data will be submitted to the server in fixed groups in the background, and the data collected in real-time can be directly saved to a file.

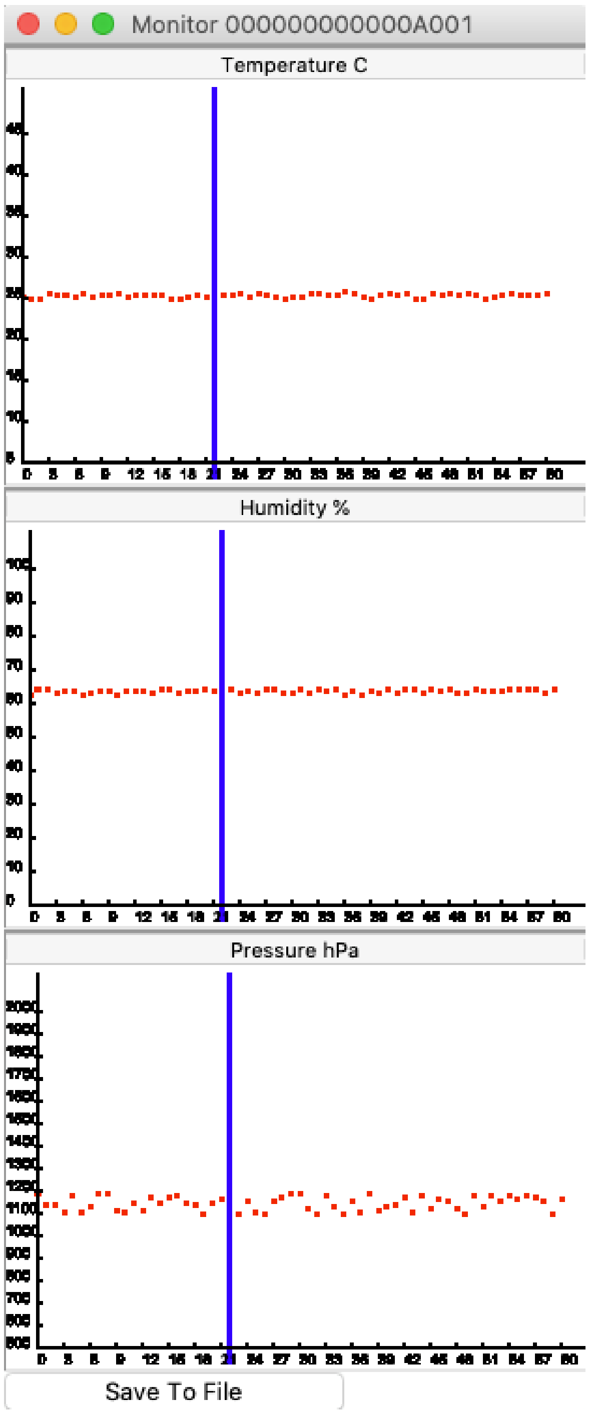
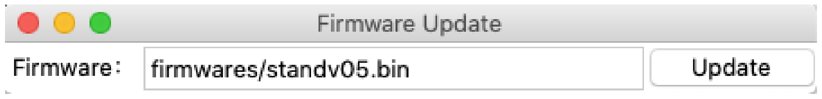


Figure 102: Monitor interface of client

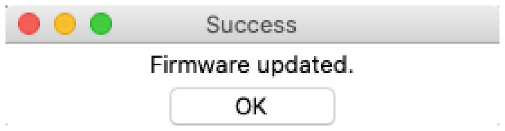
6.7.6 Firmware Update

The firmware update function can update the firmware on the data node, as shown in Figure 103. The firmware can be selected from local or from the server. According to the settings of the server, the firmware from local may be rejected because of trust issues. After the firmware update is successful, the client will display a message box, as shown in Figure 104.



Firmware: firmwares/standv05.bin

Figure 103: Firmware update interface of client

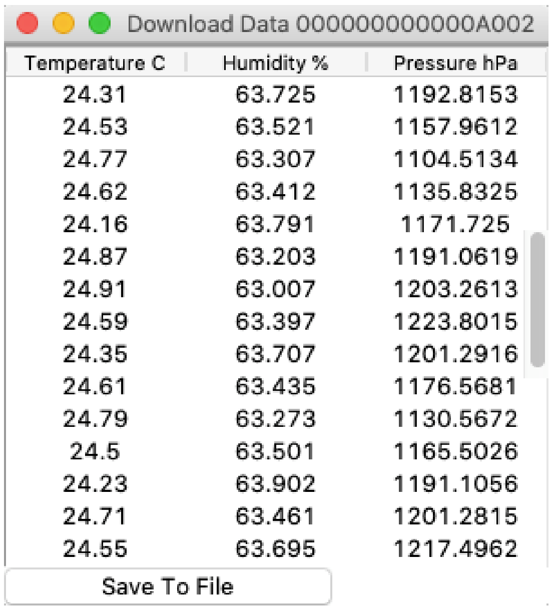


Firmware updated.

Figure 104: Firmware update success messagebox of client

6.7.7 Download Data

The management client can download data from the data node through offline connection methods, as shown in Figure 105. When the data is downloaded to the local machine, it will also be submitted to the server in a fixed group in the background. The data downloaded from the node can be saved to a file.

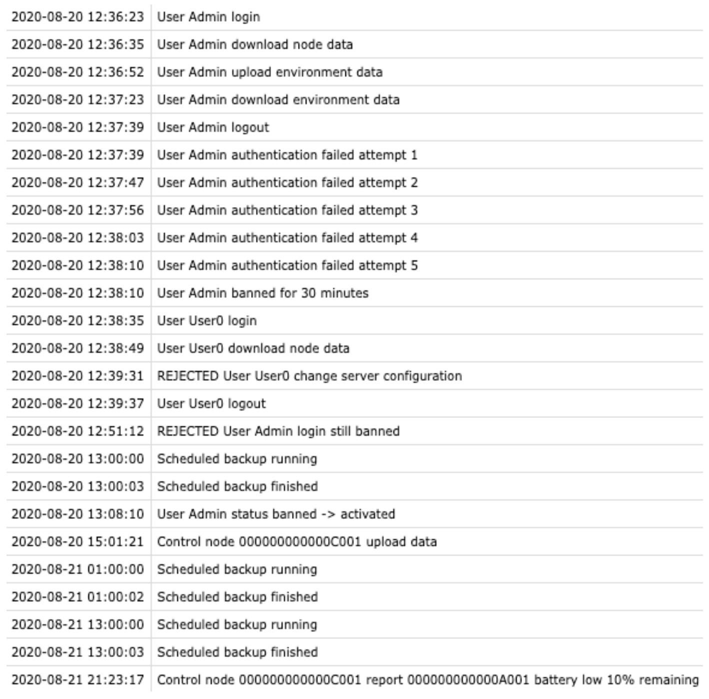

Download Data 000000000000A002

Temperature C	Humidity %	Pressure hPa
24.31	63.725	1192.8153
24.53	63.521	1157.9612
24.77	63.307	1104.5134
24.62	63.412	1135.8325
24.16	63.791	1171.725
24.87	63.203	1191.0619
24.91	63.007	1203.2613
24.59	63.397	1223.8015
24.35	63.707	1201.2916
24.61	63.435	1176.5681
24.79	63.273	1130.5672
24.5	63.501	1165.5026
24.23	63.902	1191.1056
24.71	63.461	1201.2815
24.55	63.695	1217.4962

Figure 105: Download data interface of client

6.7.8 System Log

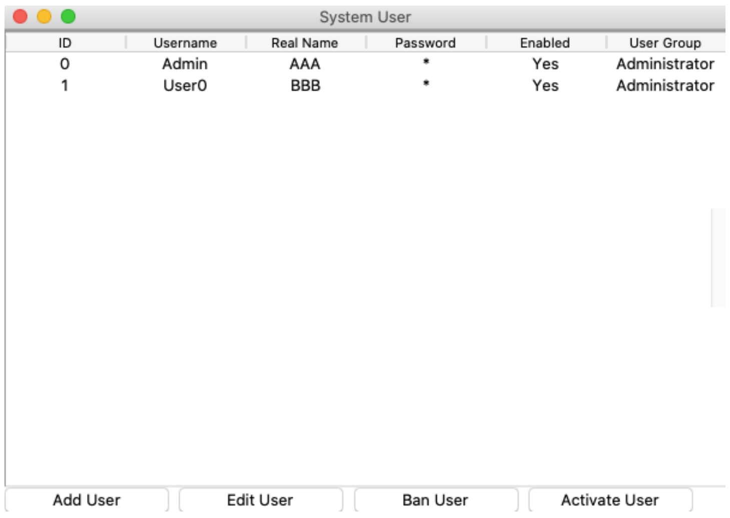
Through the client, the system log from the server can be exported to a file. Figure 106 shows part of the system log. It can be observed that the user Admin has logged in to the system and downloaded the node data from the server, uploaded the environment information, downloaded the environment information, and finally logged out. The user was banned by the server for 30 minutes due to repeated attempts to login to the system with incorrect passwords. Thereafter, User0 logged in to the system and downloaded the node information. However, User0 was rejected when trying to obtain node data due to insufficient permissions. After that, User0 logged out of the system. The log also indicates that the system automatically backed up system data at 13:00 and 01:00. Furthermore, the control node uploaded data once and reported that the remaining power of one of the data nodes was low.



```
2020-08-20 12:36:23  User Admin login
2020-08-20 12:36:35  User Admin download node data
2020-08-20 12:36:52  User Admin upload environment data
2020-08-20 12:37:23  User Admin download environment data
2020-08-20 12:37:39  User Admin logout
2020-08-20 12:37:39  User Admin authentication failed attempt 1
2020-08-20 12:37:47  User Admin authentication failed attempt 2
2020-08-20 12:37:56  User Admin authentication failed attempt 3
2020-08-20 12:38:03  User Admin authentication failed attempt 4
2020-08-20 12:38:10  User Admin authentication failed attempt 5
2020-08-20 12:38:10  User Admin banned for 30 minutes
2020-08-20 12:38:35  User User0 login
2020-08-20 12:38:49  User User0 download node data
2020-08-20 12:39:31  REJECTED User User0 change server configuration
2020-08-20 12:39:37  User User0 logout
2020-08-20 12:51:12  REJECTED User Admin login still banned
2020-08-20 13:00:00  Scheduled backup running
2020-08-20 13:00:03  Scheduled backup finished
2020-08-20 13:08:10  User Admin status banned -> activated
2020-08-20 15:01:21  Control node 000000000000C001 upload data
2020-08-21 01:00:00  Scheduled backup running
2020-08-21 01:00:02  Scheduled backup finished
2020-08-21 13:00:00  Scheduled backup running
2020-08-21 13:00:03  Scheduled backup finished
2020-08-21 21:23:17  Control node 000000000000C001 report 000000000000A001 battery low 10% remaining
```

Figure 106: System log interface of client

6.7.9 User Manager



ID	Username	Real Name	Password	Enabled	User Group
0	Admin	AAA	*	Yes	Administrator
1	User0	BBB	*	Yes	Administrator

Figure 107: User manager interface of client

The user manager interface is used to add users to the system and modify users in the system, as shown in Figure 107. Users can be activated and blocked through this interface. The interface will also list the users in the system. Only users with user management permission can list and modify other users in the system, otherwise, users can only modify their account information through this page and cannot activate or block their account.

Chapter 7

Conclusion and Future Work

7.1 Conclusions

In this thesis, by analyzing the problems encountered and related requirements, we have developed a user-friendly and intelligent system. The hardware of the system is a modular design. Users can select needs-based peripherals to complete the process of collecting environmental information. Automatic node network construction reduces the cost of system deployment and a single portal based on a cellular network can cover a larger area. The low-power design allows the system to run for longer. The supporting software system includes server and client. The clean interface can complete management whether in offline or online use. The system, therefore, achieves the identified goals.

The hardware and software are separated, the server and the client are separated, and each part has its own customizable parts. In terms of hardware, we attempt to provide more functions and improve the performance of computing under the premise of ensuring battery life. Although the hardware platform has limited computing performance due to the limitations of the selected microcontroller, nodes that need to ensure battery life have balanced computing performance can enable the data node to break away from dependence on cloud computing services, which is crucial for delay-sensitive application scenarios. The unique dual-storage native backup function provides a higher level of protection for data storage. The expansion interface provided by the hardware platform is not only be used to connect sensors, it also can be used to connect to control nodes. Once the platform can obtain a permanent power supply, the sensor platform can immediately be converted into an all-in-one platform that has both sensing and controlling abilities. With certain computing capabilities, it can be used to complete the field control function. The battery design can increase stability in unexpected situations of mains power loss, even under control mode. These functions and designs originally applied to the sensor platform still have advantages when used as a control platform. The rechargeable lithium batteries can also reduce environmental pollution caused by the frequent replacement of dry batteries. The life of a lithium battery is approximately 200 to 300 charge-discharge cycles. If a fully charged battery can provide three months of battery life, 200 cycles will power a data node for about 50 years.

The existence of the server in the software design makes it possible to access the system regardless of where the node is deployed and where the user is located, provided the network environment is unblocked. The separated design can ensure the stability of the system; even if the system is used in offline mode, it will not be affected if the management client is in an unstable state. Data storage separation ensures that data can be exported normally even if the system server is damaged. Automatic backup scheduling provides double guarantees for system data safety. The process of data transmission on the network also considers security factors to prevent data leakage. In addition to functional requirements, the system's safety, and its robust design have also been considered.

As one of the key functions of this system, the network construction between nodes has also undergone extensive testing and tuning. Since the Wi-Fi module itself only focuses on the Wi-Fi communication itself, all the advanced functions need to be developed, which gives us the possibility to step outside the limitations of the traditional Wi-Fi network standards. In short, we have implemented our own protocol with the help of Wi-Fi communication, which can build a large network from a multi-level Wi-Fi network. In the self-designed multi-level network, IP addresses and the protocols that are necessary for traditional network communication are no

longer needed. Therefore, the data that need to be broadcast on the network and the complexity of network services are reduced. This is critical for power-sensitive platforms. Reducing the amount of calculation and data sending and receiving on the network can significantly extend the battery life and save system memory. While simplifying the structure, the security of the network is also considered. Multilayer identity authentication and data verification ensure the safety and accuracy of the data transmission as far as possible, which increases the power consumption slightly. Nevertheless, given the importance of the data, this is acceptable.
The client with the graphical interface provides support for the actual application of the system. It only requires a simple familiarization process to operate the system. Even if the user does not have a deep understanding of computer software, the user will not encounter too many obstacles when using the system.

7.2 Contributions

Our system has made the following contributions:

- Reduced costs: the cost price per platform is about 13 USD - 16 USD.
- Reduced the platform size: the maximum size is only 2.15*1.36 inches.
- Reduced the difficulty of system reconstruction and enhanced platform compatibility in a modular design.
- Wi-Fi-based multi-level inter-device network auto-building function, with high bandwidth and signal strength that surpasses other local wireless communication methods.
- Network security guarantee, multi-level identity authentication, data encryption, data verification methods, and data retransmission strategy significantly reduce the possibility of data tampering and prevent transmission errors caused by poor signals.
- Native data-backup strategy to enhance data reliability.

- With edge calculation function, the node can perform a certain degree of local calculation, including matrix, and FFT.
- Separate modular power supply design, reserve interfaces for various power inputs.
- Server-client design with roles and permissions supports multi-user operation and separation of roles.
- Communication security guarantee provides a security guarantee for the communication between the client and the server over the Internet. The server database is equipped with system security checks and encryption to ensure data security as far as possible.

7.3 Future Work

At present, the system can still be improved in related software functions, such as adding more data analysis methods and introducing more mathematical calculations. This will transform the system into a sensor-based scientific computing platform from a sensor-based platform. In terms of node design, there is room for optimization of the node network. There is currently no flow-control method to address the situation when the node network is in a congested state. If the flow control method is implemented, the stability of the node network can be increased when the node network is under high load, and the network blocking can be avoided. At present, the compatibility of peripherals mainly relies on the integrated code being as compatible as possible with peripherals. In the future, a driver framework can be added and related APIs can be provided to simplify the process of peripheral driver development and achieve native driver-level compatibility for related peripherals. At present, there can only be one control node in the node network, which will partially reduce the fault tolerance of the node network. In the future design, the ability of multiple control nodes to coexist in the node network will be added to improve the ability to deal with accidents when one of the control nodes is offline.

Bibliography

[1] Abo-Zahhad, M., Ahmed, S. M., Farrag, M., Ahmed, M. F. A., and Ali, A. Design and implementation of building energy monitoring and management system based on wireless sensor networks. In *2015 Tenth International Conference on Computer Engineering Systems (ICCES)* (2015), pp. 230–233.

[2] Arom-oon, U. An aes cryptosystem for small scale network. In *2017 Third Asian Conference on Defence Technology (ACDT)* (2017), pp. 49–53.

[3] Ayaz, M., Ammad-uddin, M., Baig, I., and e. M. Aggoune. Wireless sensor's civil applications, prototypes, and future integration possibilities: A review. *IEEE Sensors Journal 18*, 1 (2018), 4–30.

[4] Aziz, M. V. G., Wijaya, R., Prihatmanto, A. S., and Henriyan, D. Hash md5 function implementation at 8-bit microcontroller. In *2013 Joint International Conference on Rural Information Communication Technology and Electric-Vehicle Technology (rICT ICeV-T)* (2013), pp. 1–5.

[5] Bai, Y. *ARMÂ® Microcontroller Architectures.* 2016, pp. 13–82.

[6] Becari, W., and Ramirez-Fernandez, F. J. Electrogoniometer sensor with usb connectivity based on the ieee1451 standard. In *2016 IEEE International Symposium on Consumer Electronics (ISCE)* (2016), pp. 41–42.

[7] Bouaziz, S., Fan, M., Reynaud, R., and Maurin, T. Multi-sensors and environment simulator for collision avoidance applications. In *Proceedings*

Fifth IEEE International Workshop on Computer Architectures for Machine Perception (2000), pp. 127–130.

[8] Caldari, M., Conti, M., Crippa, P., Orcioni, S., Sbrega, M., and Turchetti, C. Object-oriented design methodology applied to the modeling of usb device and bus interface layers. In *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No.02CH37353)* (2002), vol. 2, pp. II–II.

[9] Chatterjee, A., Shah, R. J., and Hasan, K. S. Efficient data compression for iot devices using huffman coding based techniques. In *2018 IEEE International Conference on Big Data (Big Data)* (2018), pp. 5137–5141.

[10] Dano, E. B. Importance of reuse and modularity in system architecture. In *2019 International Symposium on Systems Engineering (ISSE)* (2019), pp. 1–8.

[11] Dengwei Wang, Yinghua Lu, Hongxin Zhang, Che Shuliang, and Yunan Han. A wireless sensor network based on bluetooth for telemedicine monitoring system. In *2005 IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications* (2005), vol. 2, pp. 1361–1364 Vol. 2.

[12] Deshpande, P., Santhanalakshmi, S., Lakshmi, P., and Vishwa, A. Experimental study of diffie-hellman key exchange algorithm on embedded devices. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)* (2017), pp. 2042–2047.

[13] Fan, H., Mao, J., Feng, Q., Qi, X., Li, D., Fen, L., Hu, D., Diao, X., Ghannam, R., and Heidari, H. The design of intelligent sensor interface circuit based on 1451.2. In *2019 IEEE 2nd British and Irish Conference on Optics and Photonics (BICOP)* (2019), pp. 1–4.

[14] Fang Rao, and Jianjun Tan. Energy consumption research of aes encryption algorithm in zigbee. In *International Conference on Cyberspace Technology (CCT 2014)* (2014), pp. 1–6.

[15] Gui-hong, L., Hua, Z., and Gui-zhi, L. Building a secure web server based on openssl and apache. In *2010 International Conference on E-Business and E-Government* (2010), pp. 1307–1310.

[16] Guler, T., Gross, G., Litvinov, E., and Coutu, R. Quantification of market performance as a function of system security. *IEEE Transactions on Power Systems 22*, 4 (2007), 1602–1611.

[17] Harikrishnan, V. S., Irene, S., and Pitchiah, R. Implementation of transducer electronic data sheet for zigbee wireless sensors in smart building. In *2013 Seventh International Conference on Sensing Technology (ICST)* (2013), pp. 906–911.

[18] Hayashikoshi, M., Noda, H., Kawai, H., Nii, K., and Kondo, H. Low-power multi-sensor system with task scheduling and autonomous standby mode transition control for iot applications. In *2017 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)* (2017), pp. 1–3.

[19] Heydari, B., and Dalili, K. Emergence of modularity in system of systems: Complex networks in heterogeneous environments. *IEEE Systems Journal 9*, 1 (2015), 223–231.

[20] Hsieh, S. L., Lai, F., Cheng, P. H., Chen, J. L., Lee, H. H., Tsai, W. N., Weng, Y. C., Hsieh, S. H., Hsu, K. P., Ko, L. F., Yang, T. H., Chen, C. H., and Chen, C. H. An integrated healthcare enterprise information portal and healthcare information system framework. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society* (2006), pp. 4731–4734.

[21] Huang, Y., Lin, W., and Zheng, H. A decision support system based on gis for flood prevention of quanzhou city. In *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics* (2013), vol. 1, pp. 50–53.

[22] Jeongho Kang, Yongduk Kim, Hyungpyo Kim, Junwhan Jeong, and Sekwang Park. Comfort sensing system for indoor environment. In *Proceedings of International Solid State Sensors and Actuators Conference (Transducers '97)* (1997), vol. 1, pp. 311–314 vol.1.

[23] Jurjens, J., Lehrhuber, M., and Wimmel, G. Model-based design and analysis of permission-based security. In *10th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'05)* (2005), pp. 224–233.

[24] Kari, H. H., Saikkonen, H. K., Park, N., and Lombardi, F. Analysis of repair algorithms for mirrored-disk systems. *IEEE Transactions on Reliability 46*, 2 (1997), 193–200.

[25] Kattan, A., and Poli, R. Evolutionary lossless compression with gp-zip. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)* (2008), pp. 2468–2472.

[26] Kaur, R., and Singh, M. A survey on zero-day polymorphic worm detection techniques. *IEEE Communications Surveys Tutorials 16*, 3 (2014), 1520–1549.

[27] Kumar, A., and Hancke, G. P. An energy-efficient smart comfort sensing system based on the ieee 1451 standard for green buildings. *IEEE Sensors Journal 14*, 12 (2014), 4245–4252.

[28] Liu, A. X., and Gouda, M. G. Firewall policy queries. *IEEE Transactions on Parallel and Distributed Systems 20*, 6 (2009), 766–777.

[29] Liu, K., Jiang, J., Ding, X., and Sun, H. Design and development of management information system for research project process based on front-end and back-end separation. In *2017 International Conference on Computing Intelligence and Information System (CIIS)* (2017), pp. 338–342.

[30] MÃ¼ller, R., Esser, M., JanBen, M., and Corves, B. Modular control system for reconfigurable robot applications. In *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)* (2011), pp. 1–5.

[31] Moon, Y., Choi, Y., Hong, S., and Lee, I. Sensor data management system in sensor network for low power. In *2008 10th International Conference on Advanced Communication Technology* (2008), vol. 1, pp. 504–507.

[32] Moon, Y., Lee, J., and Park, S. Sensor network node management and implementation. In *2008 10th International Conference on Advanced Communication Technology* (2008), vol. 2, pp. 1321–1324.

[33] Mustapha, A. M., Hannan, M., Hussain, A., and Basri, H. Implementing gis in bus identification and monitoring system. In *International Conference on Electrical, Control and Computer Engineering 2011 (InECCE)* (2011), pp. 461–465.

[34] Nan Li. Research on diffie-hellman key exchange protocol. In *2010 2nd International Conference on Computer Engineering and Technology* (2010), vol. 4, pp. V4–634–V4–637.

[35] Ophir, L., Bitran, Y., and Sherman, I. Wi-fi (ieee 802.11) and bluetooth coexistence: issues and solutions. In *2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No.04TH8754)* (2004), vol. 2, pp. 847–852 Vol.2.

[36] Patterson, and Sequin. A vlsi risc. *Computer 15*, 9 (1982), 8–21.

[37] Pleskach, V., Pleskach, M., and Zelikovska, O. Information security management system in distributed information systems. In *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)* (2019), pp. 300–303.

[38] Putri Ratna, A. A., Dewi Purnamasari, P., Shaugi, A., and Salman, M. Analysis and comparison of md5 and sha-1 algorithm implementation in simple-o authentication based security system. In *2013 International Conference on QiR* (2013), pp. 99–104.

[39] Qin, L., Huang, T., Zhang, H., and Gu, J. Development of archives management information system based on. net multi-tier architecture. In *2009 3rd IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications* (2009), pp. 1210–1213.

[40] Ramamurthy, H., Prabhu, B. S., Gadh, R., and Madni, A. M. Wireless industrial monitoring and control using a smart sensor platform. *IEEE Sensors Journal 7*, 5 (2007), 611–618.

[41] RamÃrez, C. A., BarragÃ¡n, R. C., GarcÃa-Torales, G., and Larios, V. M. Low-power device for wireless sensor network for smart cities. In *2016 IEEE MTT-S Latin America Microwave Conference (LAMC)* (2016), pp. 1–3.

[42] Rautmare, S., and Bhalerao, D. M. Mysql and nosql database comparison for iot application. In *2016 IEEE International Conference on Advances in Computer Applications (ICACA)* (2016), pp. 235–238.

[43] Renping Xu, Kunqian Wang, Xiaonan Shu, and Chao Zhang. Interface management for industrial design. In *2009 IEEE 10th International Conference on Computer-Aided Industrial Design Conceptual Design* (2009), pp. 116–119.

[44] Schmalzel, J., Figueroa, F., Morris, J., Mandayam, S., and Polikar, R. An architecture for intelligent systems based on smart sensors. *IEEE Transactions on Instrumentation and Measurement 54*, 4 (2005), 1612–1616.

[45] Sharma, D. Response time based balancing of load in web server clusters. In *2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)* (Aug 2018), pp. 471–476.

[46] Shen, J., Wang, H., Meng, Z., and Huo, P. Management information system design of safecoal-production based on c/s and b/s. In *2010 2nd IEEE*

International Conference on Information Management and Engineering (2010), pp. 305–308.

[47] Soyler, A., and Sala-Diakanda, S. A model-based systems engineering approach to capturing disaster management systems. In *2010 IEEE International Systems Conference* (2010), pp. 283–287.

[48] Stone, J., Greenwald, M., Partridge, C., and Hughes, J. Performance of checksums and crcs over real data. *IEEE/ACM Transactions on Networking 6*, 5 (1998), 529–543.

[49] Tongkaw, S., and Tongkaw, A. A comparison of database performance of mariadb and mysql with oltp workload. In *2016 IEEE Conference on Open Systems (ICOS)* (2016), pp. 117–119.

[50] Tvrdikova, M. Information system integrated security. In *2008 7th Computer Information Systems and Industrial Management Applications* (2008), pp. 153–154.

[51] Wang, Z., Jia, Z., Ju, L., and Chen, R. Asip-based design and implementation of rsa for embedded systems. In *2012 IEEE 14th International Conference on High Performance Computing and Communication 2012 IEEE 9th International Conference on Embedded Software and Systems* (2012), pp. 1375–1382.

[52] Wei, X., and Li, Y. Role control based workflow management for research projects. In *2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering* (2013), vol. 1, pp. 472–475.

[53] Xianmin Wei. Research and development of rubber tree germplasm resource information management system based on c/s and b/s integrated structure. In *2011 3rd International Conference on Advanced Computer Control* (Jan 2011), pp. 404–407.

[54] Xiaolin, Y., Nanzhong, C., Zhigang, J., and Xiaobo, C. Trusted communication system based on rsa authentication. In *2010 Second International Workshop on Education Technology and Computer Science* (2010), vol. 1, pp. 329–332.

[55] Yang, F., Wu, T., and Chiu, S. A secure control protocol for usb mass storage devices. *IEEE Transactions on Consumer Electronics 56*, 4 (November 2010), 2239–2343.

[56] Zeng, Y. Axiomatic theory of design modeling. *Trans. SDPS 6* (2002), 1–28.

[57] Zeng, Y., and Cheng, G. On the logic of design. *Design Studies 12* (07 1991), 137–141.

[58] Zeng, Y., and Gu, P. A science-based approach to product design theory part i: formulation and formalization of design process. *Robotics and Computer-integrated Manufacturing 15* (1999), 331–339.

[59] Zhao, Y., and Lu, N. Research and implementation of data storage backup. In *2018 IEEE International Conference on Energy Internet (ICEI)* (2018), pp. 181–184.

[60] Zhou, H., Li, S., Chen, S., Zhang, Q., Liu, W., and Guo, X. Enabling low cost flexible smart packaging system with internet-of-things connectivity via flexible hybrid integration of silicon rfid chip and printed polymer sensors. *IEEE Sensors Journal 20*, 9 (2020), 5004–5011.