Quest Journals
Journal of Research in Mechanical Engineering
Volume 9 ~ Issue 11 (2023) pp: 01-14
ISSN(Online):2321-8185
www.questjournals.org

Research Paper

Design of Semiconductor Manufacturing Environment Monitoring System Based on the IoT

Hongyu Yang Yanting Ni*
School of Mechanical Engineering, Chengdu University, China
Corresponding Author: Yanting Ni

ABSTRACT: *The semiconductor production and manufacturing environment monitoring system designed in this article is based on the Arduino development board to monitor the semiconductor production and manufacturing environment. Implement intelligent management by transferring data from the ESP8266 board to the Arduino development board, and set up a human-machine interface to monitor and manage environmental data. The main components of the system are the Arduino module, monitoring module, alarm module, and display module. Through the Arduino motherboard, various sensors for signal collection are controlled, and the detection amount is collected through a certain range of temperature, humidity, cleanliness, and gas composition. When the data is abnormal, an alarm device is used to alert, and the staff further operates through the display module's data.*
KEYWORDS: *Semiconductor production; Internet of Things(IoT); Environmental testing*

R*eceived 05 Nov., 2023;* ***R****evised 18 Nov., 2023;* ***A****ccepted 20 Nov., 2023*
P*ublished with open access at* ***www.questjournals.org***

I. INTRODUCTION

According to the investigation and research results, in order to ensure the safety of equipment and staff in semiconductor production and manufacturing plants, it is necessary to monitor and adjust the temperature, humidity, cleanliness, and harmful gases in the semiconductor production environment in real time, and equip it with an alarm mechanism. This can ensure that there are corresponding monitoring equipment in every corner of the production workshop, thereby ensuring the safety and stability of the entire production line. Under the conditions of the Internet of Things, conduct comprehensive monitoring of the production and manufacturing environment of semiconductors, including but not limited to multiple aspects of monitoring.

The semiconductor production and manufacturing environment monitoring system constructed in this article achieves real-time monitoring and control of the semiconductor production and manufacturing environment on the Arduino development board. Transfer the data from the ESP8266 board to the Arduino development board and set up a human-machine interface to achieve intelligent management and monitoring of environmental data.

The main components of the system include the Arduino module, monitoring module, alarm module, and display module, which together constitute the core architecture of the system. Use the Arduino motherboard to control various sensors for signal collection, and collect temperature, humidity, cleanliness, and gas composition within a certain range for detection. When there is an abnormal situation in the data, an alarm device is used to alert and further operations are carried out through the display module's data..

II. OVERALL DESIGN SCHEME

In the investigation and research, it was found that in order to ensure the safety of equipment and staff in semiconductor production and manufacturing plants, real-time detection, control, and alarm should be carried out for temperature, humidity, cleanliness, and harmful gases in the semiconductor production environment. Through these aspects, the overall module composition of the system can be designed.

The semiconductor production and manufacturing environment monitoring system designed in this article is mainly based on the Arduino development board to monitor the environment of the computer room. It uses temperature and humidity sensors, PM2.5 dust sensors, and CO sensors to transmit data to the ESP8266 module, and then uploads it to the upper cloud for visualization processing. Realize the detection and control of environmental data for semiconductor wafer factories. The selection of each hardware module of this system is shown in the figure.

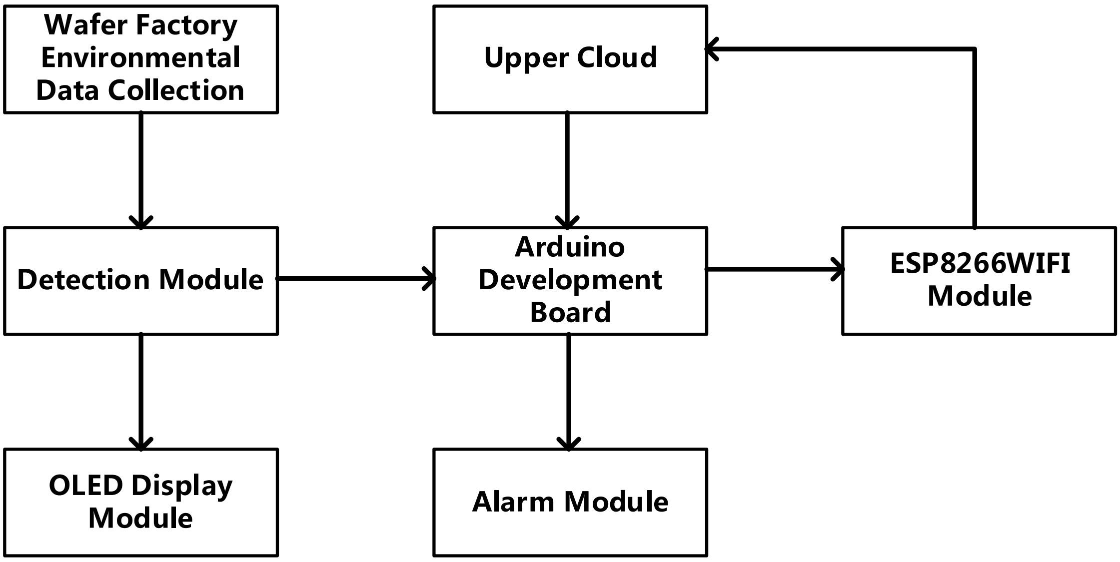

Diagram 1: Overall system framework

In the investigation and research, it was found that in order to ensure the safety of equipment and staff in semiconductor production and manufacturing plants, real-time detection, control, and alarm should be carried out for temperature, humidity, cleanliness, and harmful gases in the semiconductor production environment. Through these aspects, the overall module composition of the system can be designed. The semiconductor production and manufacturing environment monitoring system designed in this article is mainly based on the Arduino development board to monitor the environment of the computer room. It uses temperature and humidity sensors, PM2.5 dust sensors, and CO sensors to transmit data to the ESP8266 module, and then uploads it to the upper cloud for visualization processing. Realize the detection and control of environmental data for semiconductor wafer factories. The selection of each hardware module of this system is shown in the figure.

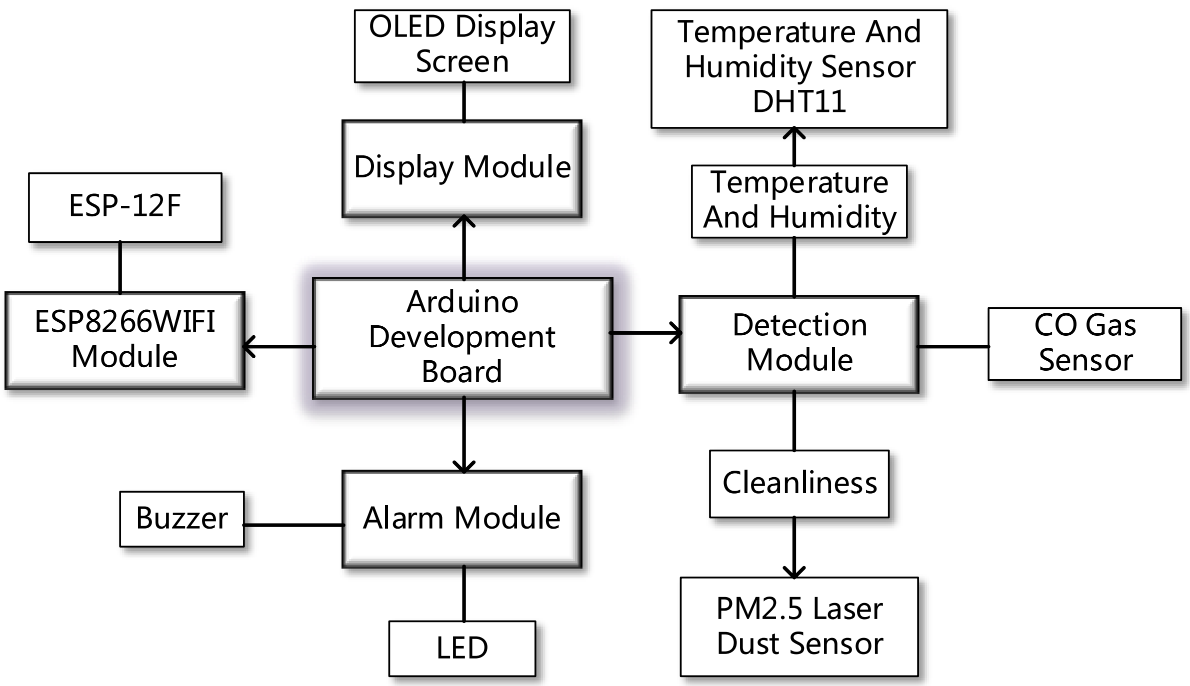

Diagram 2: Hardware Selection

III. HARDWARE CIRCUIT DESIGN

ESP8266 has a self-contained Wi Fi network function, with two functions: independent application and slave loading to run on other host MCUs. It also supports two methods: Bluetooth connection and wireless access, and can communicate wirelessly with devices such as smartphones and tablets. In the case of independent application of ESP8266, it can be directly started from the external flash [14]. Improving system performance and optimizing storage systems, the built-in cache memory plays an indispensable role. Another scenario is that ESP8266 can be used as a Wi Fi adapter through the SPI/SDIO interface or UART interface, suitable for any microcontroller design. Due to the use of low-power technology, it can achieve higher energy efficiency across the entire chip area. ESP8266 has excellent chip processing and storage capabilities, allowing it to integrate various sensors and other specific devices through GPIO ports, significantly reducing early development costs.

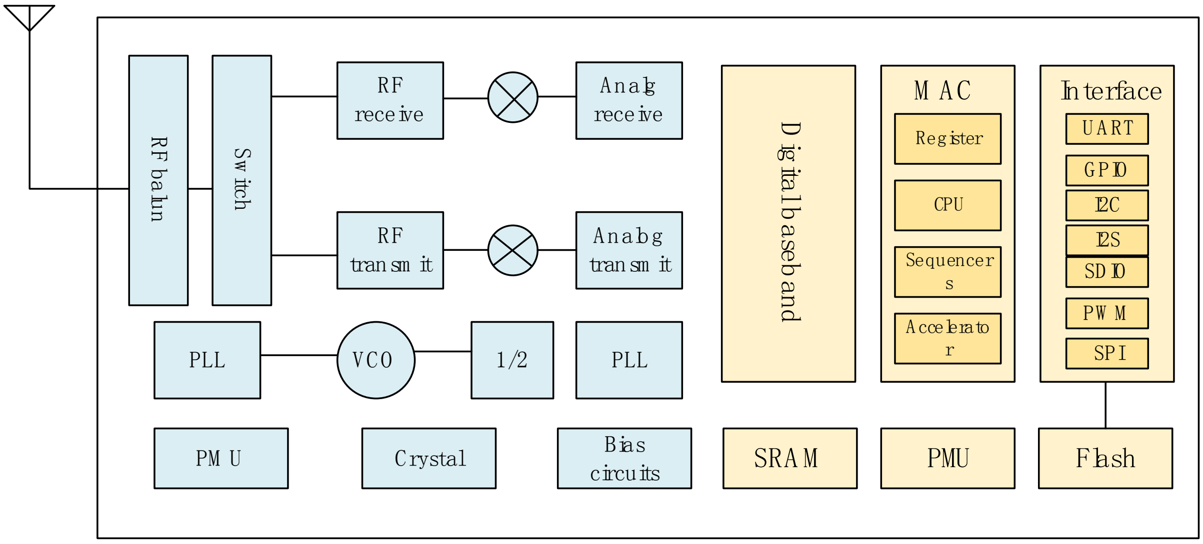

Diagram 3: ESP8266 Internal Structure

The motherboard used in this article is from Anxinke Company. The NodeMCU-8266 development board is a core development board designed by Anxinke for the ESP8266 module, which leads out all I/O to the pins on both sides. Developers can connect peripherals according to their own needs. When using a breadboard for development and debugging, the standard arrangement of pins on both sides can make the operation simpler and more convenient.

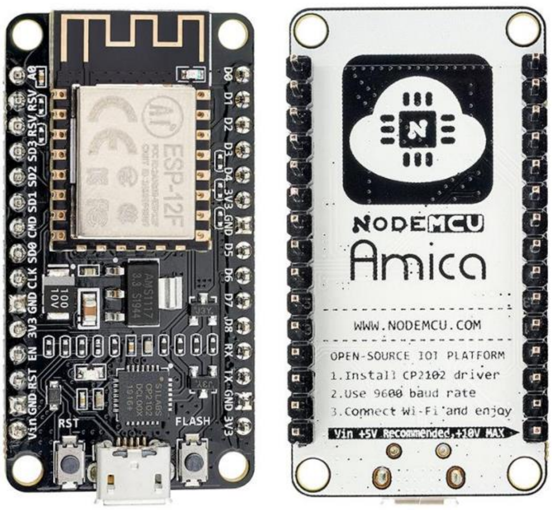

Diagram 4: NodeMCU-8266 Development Board

NodeMCU-8266_ The V1.0 development board module has a total of 30 interfaces connected, as shown in the pin diagram. The pin function definition table is the interface definition.

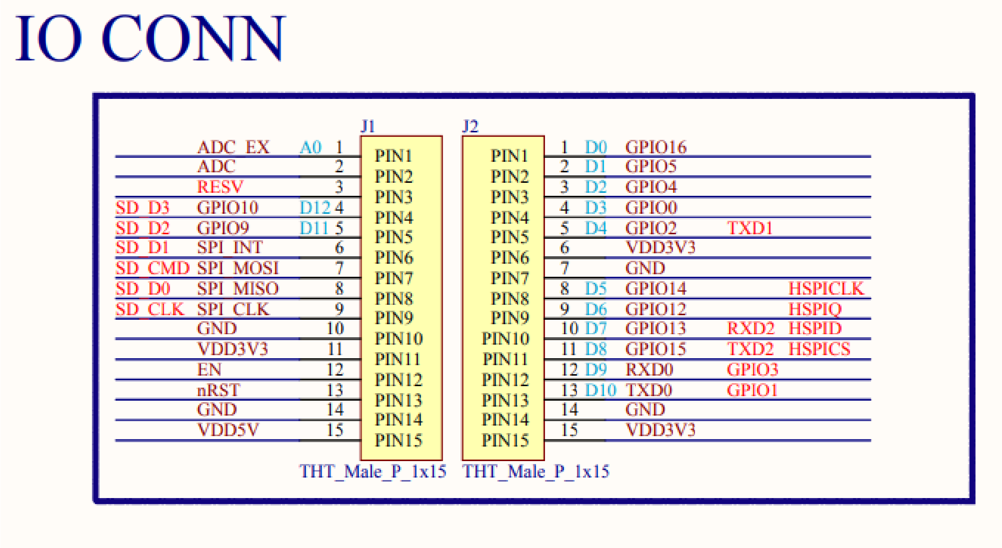

Diagram 5: NodeMCU-8266 Pin Diagram

In the system constructed in this article, ESP8266 is designed as a controller that can utilize GPIO and UART technology to receive sensor data on the board, while also serving as an access point (AP) and station (STA), providing strong support for the stable operation of the system. In addition, the controller also communicates with a cloud server to process and store sensor information, as well as provide users with various network service functions. This device is a wireless access device with the ability to autonomously search, filter, and confirm service requests .

When the user needs to use the device for data communication, it will trigger corresponding operations to complete the communication process. Therefore, in the case of using it as an AP, we can set the Wi Fi name and password for the connection by sending instructions, and then input the IP address and port number of the cloud server to achieve more flexible and efficient data transmission. By establishing connections with Wi Fi networks and using them as predefined parameters for STA, interconnection with cloud servers is achieved, thus achieving the goal of seamless connectivity.

This system can be applied to environmental monitoring in schools, hospitals, and households . Before data transmission, it is necessary to preprocess the data to ensure its integrity and accuracy. In order to improve transmission efficiency, it is necessary to analyze and process the collected information, then use corresponding algorithms for calculations, and finally send the results to the cloud server. Once temperature, humidity, carbon monoxide, and PM2.5 concentration data are obtained, they will be uploaded in the format specified by OneNET. Due to differences between different devices, corresponding communication protocols should be used to transmit data to ensure that the data transmission process is not disturbed. To ensure the reliability of data transmission, it is necessary to store and backup the collected data to ensure its integrity and availability. This article mainly studies the design and implementation of temperature and humidity sensor arrays based on embedded systems. In practical application scenarios, when the ESP8266 module is powered on, each sensor module also starts the power supply at the same time, thereby achieving pin control program burning through serial port, enabling efficient data transmission of each component. Due to the fact that serial port is a special communication interface, its security issues must be taken into consideration in design.

The CP2102 chip can achieve conversion between ordinary serial ports and USB, and its RXD and TXD are equipped with serial input and serial output, respectively, achieving efficient data transmission. During use, corresponding types of serial ports can be selected according to different needs. Communicate the microcontroller with the serial port pins, and then use software to implement corresponding functional operations. Among them, pull-up and pull-down are two important work steps. pull-up is responsible for collecting external signals to the internal, while pull-down is responsible for processing internal data to the external. Since the ESP8266 module integrates a pull-up resistor, it is unnecessary to consider the impact of pull-up resistor in the circuit design. We have conducted in-depth analysis of this section and proposed a specific design plan. Through the introduction of hardware circuit design and software design, a wireless charging system based on microcontroller control has been implemented. The overall circuit diagram is shown in the figure.

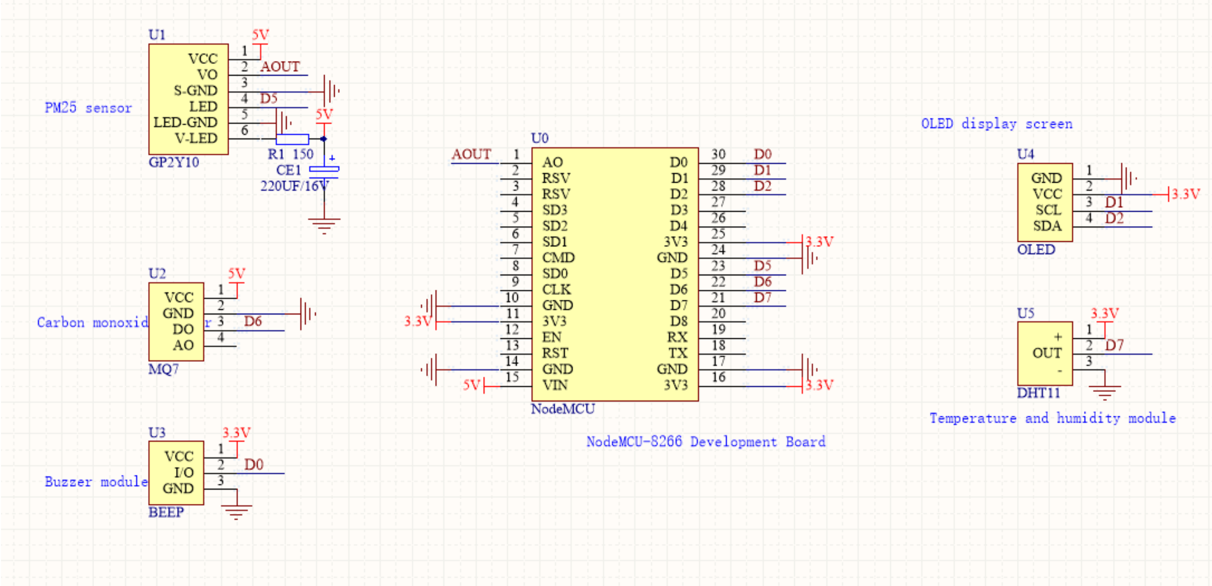

Diagram 5: Overall circuit design

IV. DESIGN OF SYSTEM SOFTWARE

After the design and connection of the hardware modules are completed, programming is needed to implement these functions. For an application, developing appropriate driver code and compiling it into corresponding packages is the key to the success of the entire project. The tool we use is the Arduino IDE software, which can be used for programming on almost all Arduino development boards. In order to enable users to have a more intuitive understanding of the various features of the software platform, this article will focus on discussing issues such as building its development environment, selecting drivers, and interfaces. This section mainly explores the process of writing driver programs for each module in the system, as well as the steps of initializing each module with the main function. In addition, it also involves the selection of communication methods between various modules and the setting of related parameters. The final section provides a detailed explanation of the entire process of system testing and an in-depth analysis of the test results. Through test cases, it has been verified that this system can achieve the expected results and operate normally.

Arduino is an open-source prototype platform based on easy-to-use hardware and software, providing users with an efficient operating experience and unparalleled convenience during use. The platform adopts a modular design concept, which has good scalability, portability, and compatibility. The operating system is designed for openness, allowing users to conduct secondary development within it to meet diverse application needs. The system adopts a programmable microcontroller circuit board and an Arduino IDE software integrated development environment to achieve computer code uploading on the physical board. By analyzing the test results under different hardware acceleration conditions, the impact of hardware and software configuration parameters on system performance is explored. This test case includes multiple applications and corresponding functional components.

Arduino Core For ESP8266 has rich resources, not only supporting compatibility with most Arduino libraries, but also providing support for multiple third-party libraries. With just a click on library management, you can easily view and install the Arduino IDE. This program implements the porting of multiple operating system platforms and can be easily extended to various embedded systems. The program also provides a series of innovative features, including an automatic start system, remote control devices, real-time data display, and data exchange between wireless modules and servers, bringing users a more efficient user experience. These characteristics make it very suitable for use in various application scenarios.

Due to ESP8266 (https://github.com/esp8266/Arduino) With the support of Arduino IDE, developers can quickly start projects and directly call a large number of interface functions, such as hardware initialization, network parameter settings, and timer initialization, effectively reducing development difficulty and cost [24]. Adopting open-source design methods ensures that there is no negative impact on the performance of other modules due to code issues during the development process, while also avoiding tedious operations caused by program modifications. Therefore, this system is developed using open-source libraries. All functions in the library are open to users, and developers can customize more options according to their personalized needs to meet their personalized needs. In addition, secondary development can be carried out according to different application scenarios to achieve more user-friendly operation methods. It provides great convenience for developers, allowing them to easily customize the required applications, thereby improving development efficiency. In order to achieve Arduino IDE support, ESP8266 needs to first introduce driver support, and its successfully configured user interface can be seen in the figure.

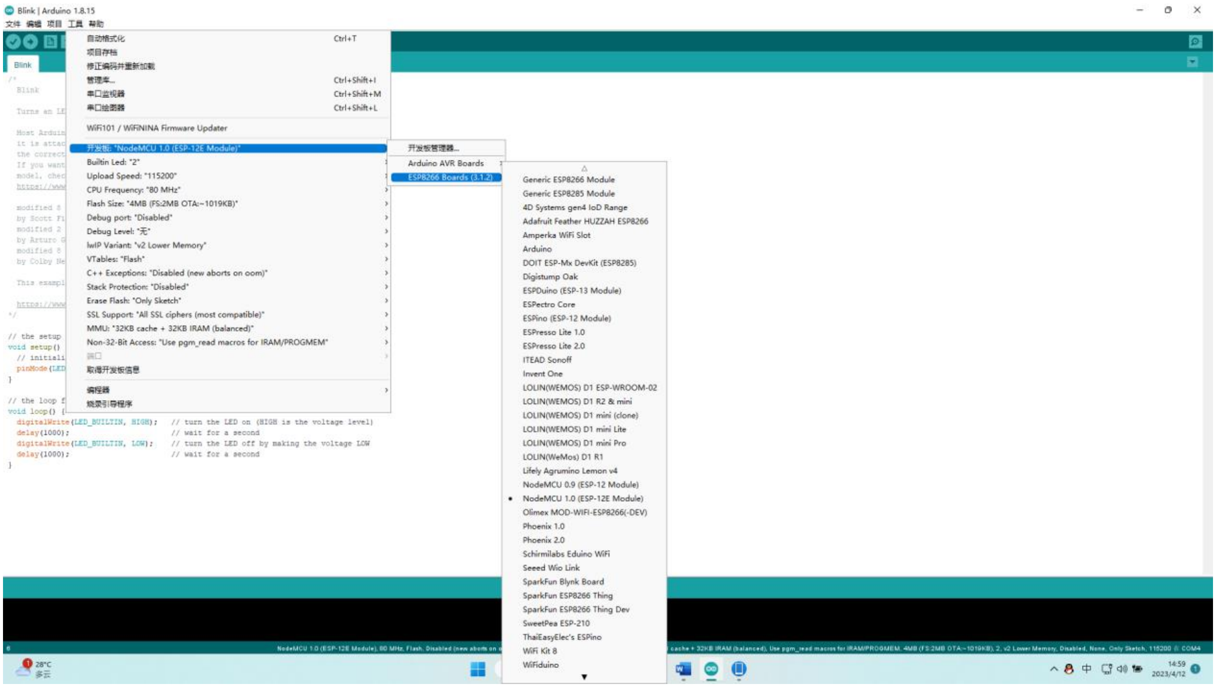

Diagram 6: Programming environment construction

4.1 Arduino Programming Method

The programming of Arduino is achieved through the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). By using this method, a complete application can be quickly built, and then these data can be stored in a database at runtime and displayed for easy query. This method does not require the use of cumbersome compilation tools or database techniques, and does not have any impact on the performance of the program itself. Therefore, this method can be adopted in many fields. Arduino's project provides users with the opportunity to interact with other computer software and web APIs, thereby enhancing the user experience. This approach allows users to easily complete all functional requirements using a simple and easy-to-use application. In this way, users do not need to have any professional knowledge of the operating system to easily use the system for operation.

Arduino is a cross platform microcontroller interface that can run on operating systems such as Windows, Linux, and Macintosh OS X. Its uniqueness lies in its differentiation from most other microcontroller interfaces. It allows users to implement their application functions through a universal programming environment. The system provides a highly flexible and easy-to-use user interface, allowing users to easily use their own designed graphical interfaces or other forms of applications. By embedding it into a common software framework, users can easily develop various applications.

Arduino's program can be divided into three main components: the elements that make up the structure, variables and constants, and the elements of the function. Each component has unique functions that operate independently of each other to achieve control and management of the application. Different types of program code are running on each section and their content is represented by corresponding data structures. When starting a sketch, the Setup() function will be called in the software structure. It can display graphics unrelated to the system on the screen. The unique feature of this function is that it sets a unique value for each instruction and only sets one parameter, which makes it very special in functionality.

By using these values, program code can be easily modified and made independent of external systems. By using this tool, operations such as variable initialization, adjusting pin modes, and enabling libraries can be achieved. All variables in the program can be set values through it, and the state during execution can be controlled based on these values. When a value needs to be changed, simply enter a specific numerical value, which is defined by the user. Due to its lack of any restrictions, it is very convenient to use on various operating systems. After powering on or resetting the Arduino board, the Setup function can only be run in one operation and cannot be run in other operations. Every time you start, you need to download and save the file, and then reload the new library to complete the call to the entire application. If you want to make modifications to the new library, you need to reload it into the system for execution, which will bring a lot of inconvenience. In fact, as long as you use simple commands, you can easily integrate these functions into an application. The p() function is constructed as the setup() function, which is used to initialize and set initial values. Your program will be able to actively control Arduino through continuous loop modifications and responses, thereby achieving more efficient operations.

Considering the open-source nature of the software, the source code for the Java environment has been released under the General Public License (GPL), while the C/C++microcontroller library runs under LGPL to ensure efficient program operation . Therefore, we will develop a simple application program interface library for calling the functions provided by the library. In this way, we can easily implement interoperability between applications under different platforms and operating systems, thus improving work efficiency. In order to enable software developers to better use the software, this article will test and analyze the system. Start the Arduino program, and for this application named Arduino, its programming environment can be viewed in the figure.

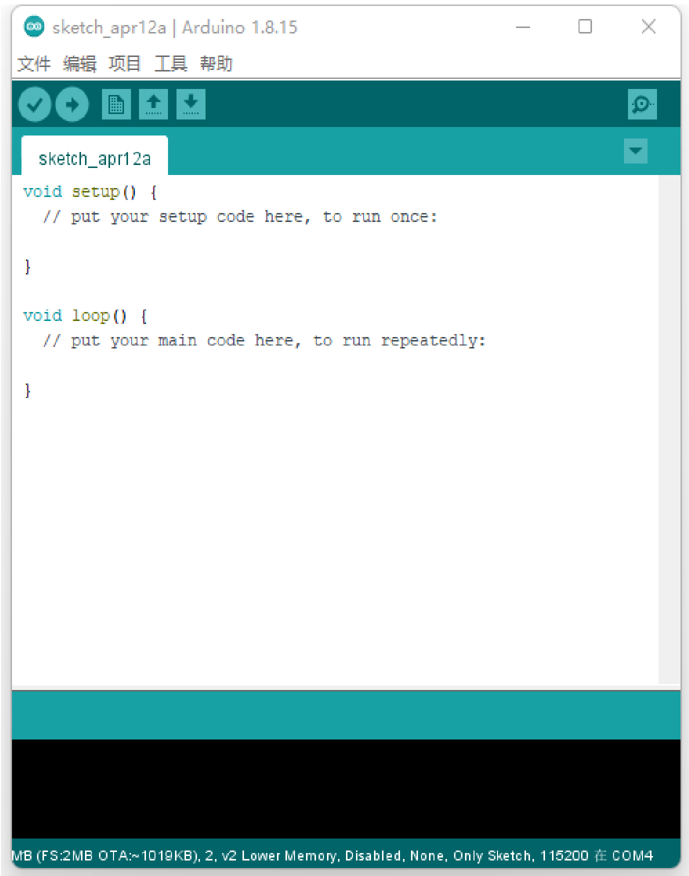

Diagram 7: Programming Interface

4.2 Design of Sensor Acquisition Program

All the network functions provided by the ESP8266 IoT platform have been implemented in the library, but due to a lack of transparency, it is difficult to provide clear information to users, resulting in a certain degree of information uncertainty. In order to enable users to manage these resources more conveniently, it is necessary to encapsulate them into a unified interface for users to access. Therefore, users do not need to create a separate database for each device, but can collaborate with other devices to achieve more efficient data management.

This article adopts a universal interface design method to solve this problem. On user_ In main, users can initialize the interface and add more functions, such as hardware initialization, network parameter settings, and timer initialization, to improve the performance and reliability of the interface. At the same time, in order to meet the needs of different users for different services, corresponding service programs have been designed, including message sending and receiving, data parsing, and stored procedures, etc.

In the system, each module is independent of each other and interconnected through external interfaces and fixed communication protocols, forming a highly interconnected structure. This system demonstrates excellent scalability and portability, enabling it to be widely applied in various fields that require large-scale development work. The wireless sensor network system designed and implemented in this paper is a distributed control system composed of multiple intelligent terminals. Adopting a modular design scheme ensures the stability of the system while also providing convenience for maintenance and optimization. The operation process of sensor nodes in the entire system is presented in a clear diagram as follows.

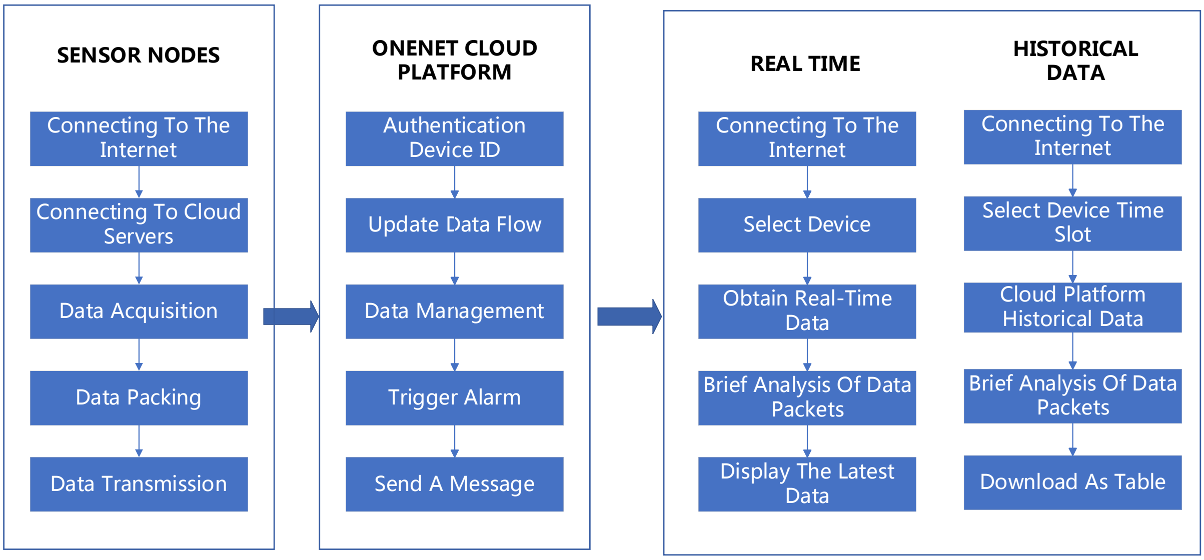

Diagram 8: Sensor workflow diagram

The hardware and software system we have established is compatible with multiple sensor nodes and can encapsulate the data collected by ESP8266 from sensors in a specific format. When a sensor node fails, it can automatically switch to other normal working states, thereby making the system more stable and reliable. In this situation, if one or more sensor nodes are mistakenly activated and unable to properly process the received data packets, it may cause the entire network to fall into a state of paralysis. A solution based on wireless sensor network technology is proposed to address this issue. In order to evaluate the interoperability between the system and different types of nodes, we have constructed two different types of sensor nodes and conducted a data transmission cycle every N seconds to ensure the continuity of data transmission. In the experiment, we selected two representative typical examples to demonstrate the effectiveness of this design scheme. After experimental verification, this scheme has been proven to meet practical application requirements and effectively improve network performance. For web pages using the ESP8266 service, the update frequency represented by N can be flexibly adjusted to meet different user needs

. To ensure that the network transmission process is not disturbed, we adopt an asynchronous event mechanism to ensure that potential errors are detected in a timely manner before each request is sent. By adopting asynchronous event processing methods, the ESP8266Wi Fi module can achieve ultra-low power deep sleep mode, thereby maintaining efficient operation during data transmission intervals. This is different from the query method in loops, while also considering the stability and power consumption of hardware programs. In addition, the design also provides a data synchronization protocol based on priority scheduling strategy. In the event of network interruption or failure, this algorithm can restore the integrity of communication links at an astonishing speed. In addition, how to use this scheme to reduce energy consumption in wireless sensor networks is also studied. Before entering deep sleep mode, network connectivity may be disrupted. After entering deep sleep, the system will provide a new service request to the user and choose whether to restart based on the current time. When the RTC (clock chip) wakes up, if the module fails to establish a connection with the access point or remote server, the system will automatically switch to deep sleep mode to avoid multiple attempts to reconnect. This approach can effectively improve the reliability and availability of data. The execution process is shown in the figure, and the main program for collecting sensor information plays an important role in the execution process, as shown in the following figure.

*Corresponding Author: Yanting Ni

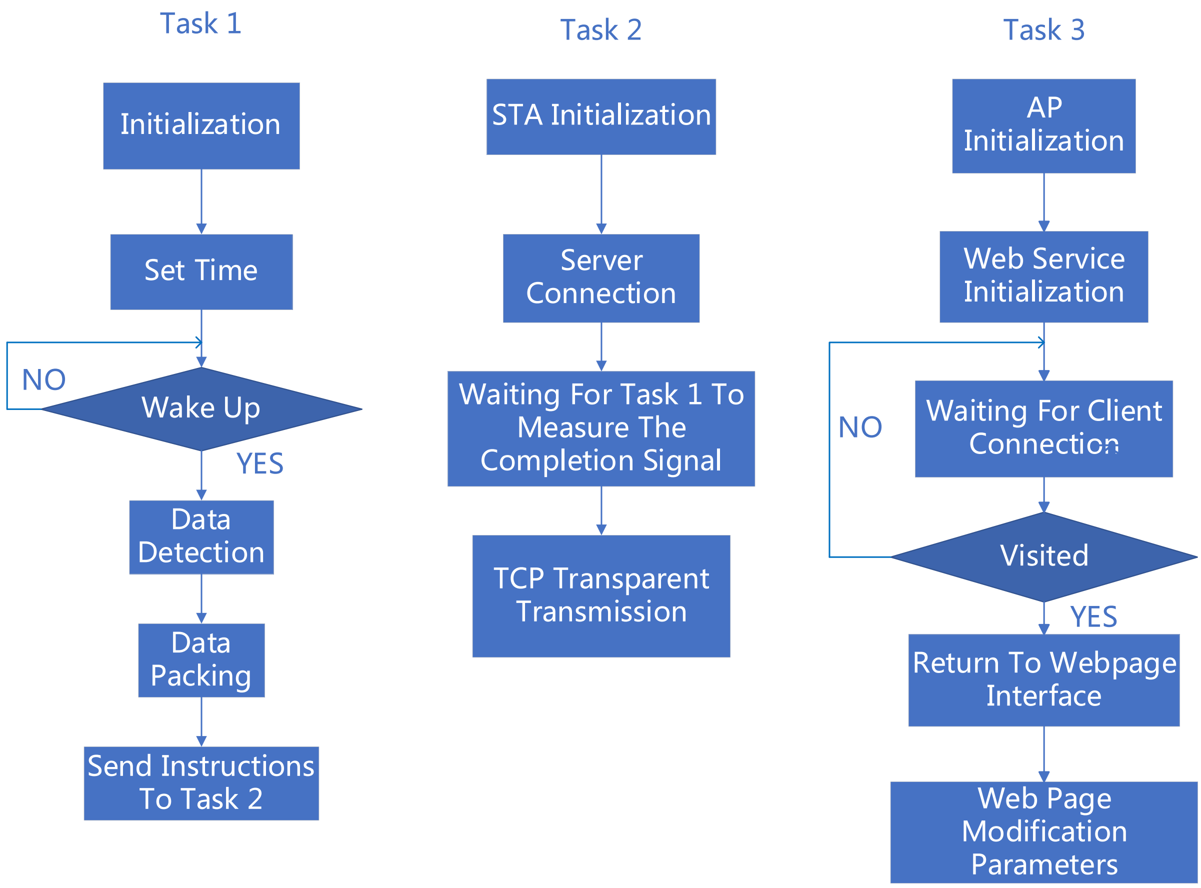

Diagram 9: Data collection program flowchart

4.3 ESP8266WiFi Programming

ESP8266 implements three working modes, namely, wireless access point mode (AP), wireless terminal mode and hybrid mode (the mixture of the above two modes). In STA mode, the ESP8266 module is connected to an internet phone or computer through a router to achieve remote control of the device. This control method belongs to client mode. The characteristic of this mode is that there is no need to add any hardware and software support in the wireless network. Using AP mode, set ATK_ The ESP8266 module defaults to the module as a hotspot, enabling direct communication between mobile phones or computers and the module, thus achieving wireless control of the local area network, equivalent to using it as a router and emitting WIF I signals.

STA+AP mode: A mode that combines two modes, that is, seamless switching is achieved through internet control, making it convenient for users to operate. This method solves the above problems through programming. The ESP8266 module will enable wireless terminal mode and connect to the most recently used access point without any parameter adjustments. This connection method is based on the WiFi configuration previously saved in flash memory.

The overall concept of this WiFi solution is to set a WiFi name and password, establish an unconnected state, monitor connection status, and output WiFi status and connection status through a serial port. If there is a connection failure or inability to connect to the network normally, the alarm system will be entered, and the user will choose whether to restart or reinstall the system to solve the above problems. The operation processes of two operation modes are presented in the figure.

V. CONSTRUCTION OF ONENET IOT PLATFORM

5.1 Cloud Platform Selection

The Internet of Things platform is an intelligent platform that utilizes advanced methods such as cloud computing, big data analysis, and IoT communication technology to interconnect various intelligent devices, sensors, and other products, thereby achieving data sharing and interaction between devices. IoT platforms can provide users with more convenient services and experiences. With the increasing maturity and widespread application of Internet of Things technology, more and more people are becoming interested in IoT platforms. In this case, IoT platforms have become an essential part of people's daily lives. In recent years, with the vigorous development of Internet of Things technology, the application scenarios of cloud platforms have become increasingly widespread, and many large enterprises have established their own cloud platform websites. Therefore, research on IoT cloud platforms is of great significance.

The classification of IoT cloud platforms usually includes private IoT cloud platforms and general IoT cloud platforms, both of which are common types of cloud platforms. In these IoT cloud platforms, public cloud platforms are typically composed of multiple hosts, each with different types of devices installed on it. A private IoT cloud platform is a type of IoT server that provides services specifically for a single host, and its service target is limited to a specific customer [31]. Suppose a farmer installs an IoT temperature and humidity meter in their greenhouse and transmits temperature and humidity data to a host through the network. There is a sensor installed on this host to measure parameters such as temperature and humidity inside the greenhouse. The host runs a specific program to record and analyze the temperature and humidity inside the greenhouse to ensure the accuracy and reliability of the data. If the host malfunctions, it will affect other melon households and cause huge economic losses. Unlike ordinary IoT cloud platforms, a universal IoT cloud platform is a large-scale IoT cloud platform developed by professional institutions. It provides free or paid services to the IoT industry market, with huge customer traffic, a large number of devices, and various types of data, providing strong support for the development of the IoT.

Alibaba Cloud, Amazon Cloud, Microsoft Cloud, Tencent Cloud, Baidu Cloud, Mobile OneNET, and China Unicom, these large IoT platform suppliers are currently widely used in China. They all have their own independent cloud computing systems in their respective products. Compared to traditional IoT servers, this large-scale IoT platform has obvious advantages, as traditional servers require a lot of manpower and resources to build a huge infrastructure, and also need to find and collaborate with embedded developers.

In large-scale IoT applications, these technologies are no longer sufficient to meet the needs, so a new technology is needed to support the entire system. The universal large-scale IoT platform provides device side SDKs, making cloud connectivity more efficient, while supporting global device access, heterogeneous network device access, multi-protocol device access, and device access in multiple environments. In addition, in the era of cloud computing, traditional servers cannot meet the needs.

The scalable architecture of traditional servers requires independent design, which makes the process from device granular scheduling to servers extremely difficult, and also requires the construction of infrastructure such as load balancing. Therefore, the universal Internet of Things platform has great application value and market space in cloud computing. The universal Internet of Things platform provides multiple security guarantees, ensuring the security of the black cloud, while ensuring the uniqueness of device authentication, transmission encryption to ensure data is not tampered with, cloud shield protection and permission verification to ensure cloud security.

5.2 OneNET Cloud Platform

OneNET is an IoT open cloud platform launched by China Mobile, aiming to provide solutions for various cross platform IoT applications and also provide solutions for IoT device access issues. Provide a safe, stable, and reliable solution, and improve and perfect existing technologies. This platform provides multiple protocols that enable users and a large number of customers to easily access, store, compute, and showcase IoT products, effectively reducing the cost of IoT development. It is also one of the most mature and stable technologies in the current IoT industry. OneNET has been widely penetrated into people's daily life, covering many fields such as smart life, smart home, greenhouse, environmental detection [32]. Through accurate and fast access to platform protocols, OneNET has realized multiple functions such as data transmission, storage and management.

The OneNET Internet of Things platform supports access to multiple protocols, including 6 communication protocols such as MQTT, EDP, HTTP, etc. EDP is a public protocol based on TCP, specifically designed for the characteristics of the Internet of Things, and supports platform customization of data and messages. This article provides a detailed explanation and analysis of it. MQTT is data centric, while HTTP is file centric, with significant differences in data transmission and management between the two. Therefore, by converting between different protocols, users can more conveniently access the information they need. In addition, MQTT provides customers with an independent way of existence, thereby enhancing the stability and reliability of the entire system. This paper designs and implements an IoT application platform with good portability, which can adapt well to the communication needs of different devices and environments. In the event of a client failure, the entire system can operate continuously and stably without any interference. This article ESP8266 uses the MQTT protocol, which also provides users with options for Last will&Testparameter and Retained messages. These two options can be used to save and restore messages. In the event of an unexpected interruption of the client's connection, all subscribed clients will obtain relevant information from the agent. The second indicates that the client has received new information and can make modifications to it. When a newly subscribed client receives a message, it will immediately receive the latest status updates to ensure the integrity and accuracy of the information. The second requirement is that the agent must send a request to the proxy server to confirm whether the message can be obtained. However, the HTTP protocol does not have these features.

5.3 Building a cloud platform environment for the Internet of Things

In today's increasingly mature development of IoT technology, many commonly used cloud platforms have been opened to the public. This paper adopts the OneNET cloud platform, which is an IoT platform running in the upper cloud. ESP8266 achieves WiFi connectivity through the MQTT protocol, which is suitable for application scenarios where long connections need to be maintained between devices and platforms. The MQTT protocol is characterized by the ability to achieve message unicast and multicast between devices, without relying on other services (command issuing and push services) to enable devices to manage and control real devices through application servers.

First, visit the OneNET IoT cloud platform website, register your OneNET platform account, log in and select the console, access and select the developer center, click on all the products and services above, select multi-protocol access and click on "add product", "create a new product", and use the old version MQTT protocol to achieve the characteristics of uploading and publishing data, which can be freely obtained. After clicking on the established product, add the required components, and carefully fill in the device ID, authentication information, and access method. The relevant data flow module can be set or not, and the required data flow can be received when communication is normal

5.4 OneNET interface design

After creating product and device information on the multi-protocol access page of the cloud platform, further click on the data flow template interface to create multiple data streams. In the semiconductor environment detection system designed in this article, the cloud needs to receive data on temperature and humidity, air quality, PM2.5 concentration, and alarm status. These data are uploaded to the OneNET IoT cloud platform in real-time for display and recording of historical data. As shown in the figure, the data streams of several detected data are created.

After creating the required data stream on the OneNET IoT cloud platform, compile the server address, device number, product number, authentication information, etc. using the Arduino IDE, package and integrate the data we want to upload into an array through data encapsulation letters, store the attribute names and values of the data received by the sensors in the array, and connect the ESP8266 program to the mobile hotspot network, From this, the data detected by the sensor can be uploaded to the OneNET data stream. The collection of temperature, humidity, air quality and other data has been displayed locally and uploaded to the cloud, which means the data can be viewed through the data stream on the OneNET cloud platform.

OneNET provides an efficient application management service that allows us to quickly design a user interface for receiving environmental data. By sending this data to the client, users can monitor the system's operation in real time and learn about some important historical details. After every 1 second, the data collected by the sensor is refreshed, and this interface can be used on personal computers, mobile applications, or websites. It includes analyzing this information to generate visual graphical displays. The user interface provided by this platform is easy to develop, has excellent visibility, and can easily view a small amount of data before the current time node, so that users can obtain current data and trend information. In addition, the system allows developers to apply it to other types of software. From another perspective, user interface design can be achieved by adding controls to the platform, allowing developers to easily place controls and set corresponding data sources. Due to the use of object-oriented technology in this system, it allows developers to focus on their own concerns without having to focus too much on other aspects. Therefore, it provides efficient solutions for developers to shorten the development cycle. Additionally, due to its distributed architecture, the platform can be easily deployed to different locations. On this platform, users can browse historical data, query device connection records, and view the number of triggering information. In addition, a simple and easy-to-use graphical interface is provided to facilitate developers to intuitively understand the system operation and data change process. As shown in the figure, its user interface presents a visual effect.

VI. IMPLEMENTATION OF PHYSICAL FUNCTIONS

6.1 Overall physical connection

The connection of each module on a breadboard is shown in the figure below, which is an indispensable and important tool for electronic experiments. It has two advantages:

Simple: Due to the many small sockets on the board, various electronic components can be inserted or pulled out as needed, eliminating the welding process and saving circuit assembly time;

Reuse: Components can be reused, making them very suitable for assembly, debugging, and training of electronic circuits.

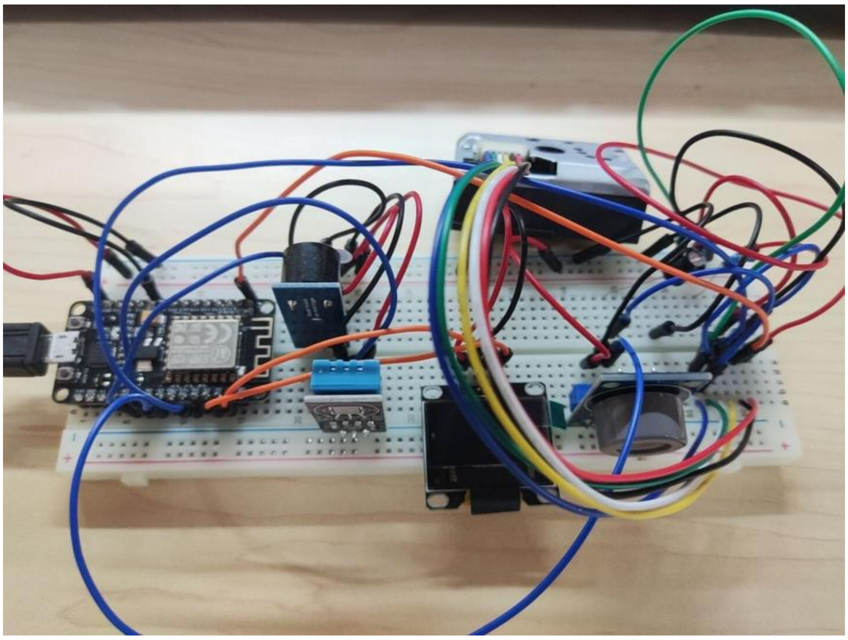

Diagram 10：Physical wiring diagram

6.2 Monitoring and visualization of data in the environment

After connecting each module according to the circuit diagram, open the WIFI with the ID set on the phone. After connecting the data cable using USB, it was found that the small light on the ESP8266 module was flashing to indicate successful connection. After pressing the RST reset button at this time, the ESP8266 module will automatically detect the WIFI and connect to the internet. When the OLED display temperature is between 10 and 30 ℃, the humidity is between 40% and 70%, and the air quality value is less than 400, the system is in normal operation state. On the contrary, when these data are abnormal, the buzzer will give an alarm, which will be fed back to the workers in the semiconductor manufacturing environment for timely adjustment.

At this point, open the device we have already created on the OneNet platform and click on the edited data stream. Enter the visualization page we have built. At this point, the ESP8266 module connected to the internet will be able to upload the received data to our visualization page through various data streams. Due to limited funding, under the free visualization function on the OneNet platform, the functionality and display on our visualization page are limited. The following figure shows the real-time monitoring data displayed on the visualization page.

6.3 Simulated harmful gas experiment

In a temperature, humidity, and clean indoor environment, ignite a wooden stick near the carbon monoxide sensor and blow it out to generate CO gas, which is used to simulate the sudden occurrence of toxic and harmful gases in the semiconductor production environment. The early combustion process of wood is characterized by the production of a large amount of smoke and a certain amount of CO, and the temperature gradually steadily increases. In the later stage, the combustion material decreases, the temperature gradually decreases, and smoke and CO are generated. This causes the carbon monoxide sensor and PM2.5 sensor to receive abnormal data, thereby causing the buzzer to alarm. In this way, the alarm status of carbon monoxide can be detected.

VII. CONCLUSION

The main research content and theme of this paper is to design a detection system that meets the requirements of semiconductor production environment using Internet of Things technology. This system mainly assists in monitoring important production processes in semiconductor factories, assisting semiconductor factory staff in managing and controlling production, and providing simple and effective assistance for semiconductor production processes. The main content and work completed in this article include:

Exploring the practical application of Internet of Things technology in the field of semiconductor production and manufacturing, and elaborating in detail on the communication method of wireless data transmission to servers using WiFi technology in this system. Subsequently, an in-depth analysis was conducted on the functional and technical requirements of the system, and a complete system architecture scheme was proposed, including environmental data collection, main controller, and user webpage monitoring platform. When conducting the overall scheme design analysis, we carefully divided the software and hardware, and ultimately chose a low-cost and high-precision scheme

By utilizing DHT11 temperature and humidity sensors, CO sensors, and PM2.5 dust sensors, and adopting a modular sensor node design based on ESP8266, the overall hardware circuit of the Internet of Things monitoring system was successfully connected, and the sensing layer was successfully deployed. This sensor network adopts wireless sensing technology, which has advantages such as low power consumption and low cost. The information collected by the sensor can be directly transmitted to the GPIO port of the ESP8266 Wi Fi module, and the sensor data can be encapsulated and uploaded through the TCP protocol to achieve real-time monitoring of the sensor data. A low-cost, low-power, and highly stable smart home remote monitoring platform has been constructed using wireless sensor network technology, which can provide users with a safe and comfortable home environment. After experimental verification, the accuracy of the sensor shows extremely high reliability, the data transmission of the sensor nodes is stable and reliable, and the entire system operates stably with zero packet loss rate.

By building the Arduino platform, the system successfully achieved rapid programming of ESP8266, thus completing the design phase of the information collection software. Different information collection programs are designed according to different sensor types, through which ESP8266 can be configured to connect with Wi Fi by one key to complete the AP mode parameter setting of ESP8266 module and data transmission of STA mode.

In practical application scenarios, we have chosen OneNET as our cloud platform to meet our needs. At the same time, corresponding client software was developed and system testing was conducted. By establishing a remote monitoring system control interface for semiconductor production and manufacturing environment on the device cloud platform, we have successfully achieved the goal of online monitoring and historical data query on the computer PC. At the same time, it can also timely and accurately transmit various collected information to users for analysis and processing. After experimental verification, the cloud platform successfully achieved comprehensive collection and recording of changes in long-term production environment parameters, and no abnormal or missing fragments were found. The system implements real-time data monitoring and historical data display visualization functions, providing users with an efficient and convenient data interaction experience.

REFERENCES

[1]. Liu Hao Design of Gas Detection and Monitoring System for Semiconductor Electronics Factory [J] Instrument Technology, 2020 (03): 27-30
[2]. Dong Xuexin Energy saving research on the air conditioning system of semiconductor factory buildings [J] Value Engineering, 2020, 39 (15): 233-234
[3]. Wang Xuening A Brief Discussion on AMC Control of HVAC in Semiconductor Clean Plants [J] Residence, 2021 (12): 22-23
[4]. Liu Li Design of a Clean Room Air Conditioning System for a Semiconductor Factory [J] Technology Wind, 2019 (26): 5-6
[5]. Yu Qingjie Design and Application Analysis of Purification Air Conditioning System for Semiconductor Factory Buildings [J] China Science and Technology Forum, 2020 (3): 52-53
[6]. Wan Minghui Microseismic control technology for a clean workshop of a certain ultra silicon semiconductor [J] Building Materials and Decoration, 2019 (24): 226-227
[7]. Tong Lu Discussion on the use of automated instruments and meters in the dust-free room monitoring system of semiconductor electronics factories [J] Science and Information Technology, 2019 (18): 74-77
[8]. Chen Xingzhong Application of Fuzzy Control in Clean Room Temperature and Humidity Control of Semiconductor Plants [D]. Shanghai Jiao Tong University, 2010
[9]. Gai Haoyu, Zhang Zhen, Zhu Lian, Zhang Youchun. Design of a real-time monitoring system for semiconductor factory environment based on LoRa technology [J]. Journal of Qiqihar University (Natural Science Edition), 2022, 38 (04): 14-19
[10]. Liu Zhifeng. Design of Intelligent Security Systems in Semiconductor Manufacturing Factories [J]. Integrated Circuit Applications, 2022, 39 (07): 45-47. DOI: 10.19339/j.issn.1674-2583.2022.07.014
[11]. Ying Huanya Research on human-machine interactive bathroom product design based on the era of the Internet of Things [D]. Jingdezhen Ceramic University, 2022. DOI: 10.27191/d. cnki. gjdtc. 2022.000055
[12]. Gao Wanming Design of a Student Physical Monitoring System Based on the Internet of Things [D]. Huaibei Normal University, 2022. DOI: 10.27699/dcnki. ghbmt. 2022.000262
[13]. Lin Xiaoying. Pure Gas System and Example of Semiconductor Ultra Clean Factory [J]. Cleanliness and Air Conditioning Technology, 2002 (02): 66-68
[14]. Shi Jingyi Energy saving research on the air conditioning system of semiconductor factories [D]. Xi'an University of Science and Technology, 2014
[15]. Luo Yuangui. Energy saving analysis of ventilation and air conditioning in large-scale semiconductor device factories [J]. HVAC, 2001 (06): 83-85
[16]. Wang Feiyang Research on the Greenhouse Internet of Things Information Collection System Based on ESP8266 [D]. Zhejiang University, 2020. DOI: 10.27461/d. cnki. gzjdx.2020.004317

[17]. Liao Yinshuang Research on Fire Detection and Alarm System Based on the Internet of Things [D]. Guizhou University, 2021. DOI: 10.27047/dcnki. ggudu.2021.002814
[18]. Wang Guangming Design of Remote Environmental Detection Data Center [D]. Inner Mongolia University of Science and Technology, 2015
[19]. Tian Xin Research on the Application of Internet of Things Technology in Middle School Physics Experimental Teaching [D]. Tianjin Normal University, 2022. DOI: 10.27363/d. cnki. gtsfu.2022.000782