Quest Journals
Journal of Electronics and Communication Engineering Research
Volume 9 ~ Issue 3 (2023) pp: 16-25
ISSN(Online) : 2321-5941
www.questjournals.org

Research Paper

Intelligent Voice Home Control System

Xu Wang, Yao Wang, Yanting Ni
School of Mechanical Engineering, Chengdu University, China
Corresponding Author: Yanting Ni

ABSTRACT: *Based on the reason that the current smart home system is limited by factors such as network and transmission control distance, this paper innovatively proposes an intelligent voice home system controlled by offline voice technology, online Tmall Genie voice control technology and Internet of Things remote transmission APP. This design is composed of home master and wireless network master node and sub node established by two NodeMCU development kits (ESP8266 wireless WiFi module). Among them, the main control part of smart home is built with Arduino development board and various household devices, and connected with the network master node through the serial port. The network master node can be connected to the home router gateway or the cellular network generated by the 4G hotspot to access the Blinker Internet of Things server. The mobile phone APP designed by the Blinker Internet of Things platform can be used to communicate with the home master in both directions using the MQTT protocol to realize the remote control of Tian-mao Smart audio online voice and mobile phone APP. In addition, the network sub-node and ASR voice module are used to form an internal LAN, which can realize the function of offline voice control of household equipment, and realize the language control in the case of home disconnection, voice control in the case of network, and remote APP control. The design of this system is based on TCP/IP, WiFi, MQTT and other network communication protocol stacks, which ensures the reliability of information transmission, has strong convenience and practicability, and can be widely used in modern homes.*

KEYWORDS: *Smart home, voice, WiFi, Arduino, Internet of Things platform*

R*eceived 06 Mar., 2023;* ***R****evised 17 Mar., 2023;* ***A****ccepted 19 Mar., 2023 © The author(s) 2023.*
P*ublished with open access at* ***www.questjournals.org***

I. INTRODUCTION

Smart home is a residential environment that integrates convenience, security and artistry, based on the residence platform and using various communication, network, control technology, image recognition and voice recognition technologies.

Using speech recognition technology to complete human-computer interaction provides users with great convenience. For the control mode of smart home, voice control is a more efficient and flexible experience mode than the traditional button or finger operation, and is the mainstream way to achieve smart home control [1,2].

At present, smart home solutions mostly use Zigbee technology for communication. Zigbee is a short-range, low-rate transmission protocol with general penetration ability, and has certain limitations in the remote control of indoor multi-scenes [3].

When using voice to control household equipment, several problems will occur: first, the smart speaker or voice assistant needs to connect to the network. If the network speed is slow or disconnected, the control will be delayed or ineffective [4]. Secondly, the transmission range of smart speakers is limited. If they are not at home or in the bedroom, they will not be able to control the home equipment in real time [5,6]. Finally, the user's privacy and security are worrying. Most cloud voice recognition will be recorded in the background, which is equivalent to that the voice of the user has been backed up and uploaded to the cloud for storage without knowing it. At present, the application of voice recognition technology in smart home is limited by network, communication distance and user privacy [7,8].

In order to solve the above problems, this paper proposes a smart home design scheme. This design uses WiFi technology to access the Internet, establish a connection with the Internet of Things cloud server, and realize remote mobile APP control and Tmall Smart voice control. WiFi module is used to build internal LAN, and high-resolution offline voice recognition module is used to communicate with home master control, which

solves the problem of voice control and recognition distance in the case of network instability. The combination of voice control and software control will further improve the convenience, reliability and security of the home.

II. OVERALL SYSTEM DESIGN SCHEME

As shown in Figure 1, is the design scheme block diagram of intelligent voice home control system. The whole system is divided into three parts: home master design, network server master node design and voice recognition system and networking design. The scheme design of the system can be divided into two parts, namely, network-free control scheme and network control scheme.

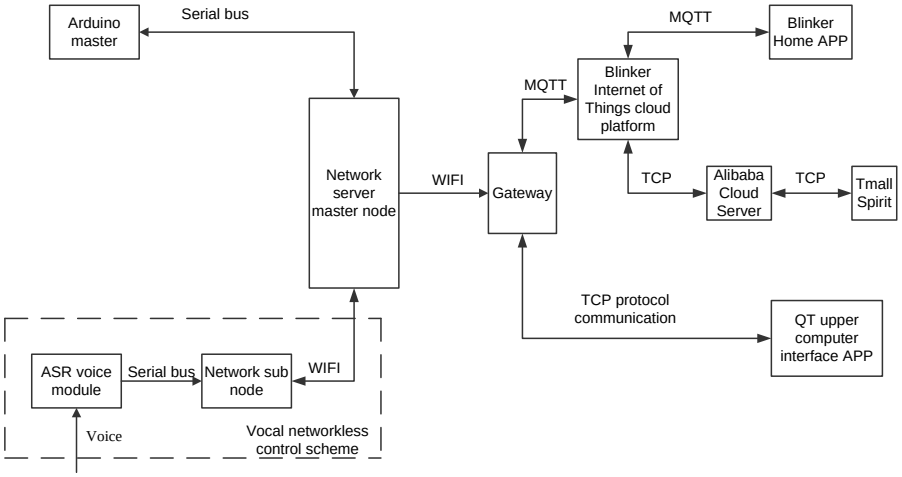

Figure1: Block diagram of overall system design scheme

The system provides users with a variety of interactive control methods and means. In the network-free control scheme, users can directly issue voice commands to control household equipment. In the online solution, users can use the mobile APP or the QT application on the PC to view the home environment information and device status in software mode and implement the control operation of the device. In addition, it can also cooperate with Tmall Smart smart speakers to query the temperature and humidity of the indoor environment and control the household equipment through online voice interaction.

The system uses the Internet of Things and WiFi technology to establish network coverage, and is equipped with diversified control methods to achieve intelligent control under different occasions and different network conditions, meet the design requirements of modern smart home, meet the needs of users under various conditions, and improve the interaction experience. The system takes the main node of the network server as the core of the network organization, and plays different roles and functions in the three parts of the design.

In the home master design, the master node connects to the home master through the serial port downward, connects to the gateway through the WiFi protocol upward, and accesses the Internet to establish MQTT connection and TCP connection with the mobile APP client and the QT application client. The master node serves as the network bridge and transceiver node between the home master and Tmall Smart and APP to realize two-way connection and communication at both ends.

In the design of network server master node, the master node acts as the server of APP and Tmall Smart. When the master node goes up, the home master control will process and analyze the switch status and temperature, humidity and illumination information of each device through the master node, and the data format will be coordinated and unified. The MQTT protocol is used to control the transmission to the mobile phone APP client and Tmall Smart or to upload the TCP protocol to the QT client. When the master node goes down, the software client or Tmall Smart sends control instructions to the server as the master node, and then it reaches the home master control serial port after being forwarded by the master node server. The MCU then controls and executes the corresponding instructions.

In the design of offline voice module and networking, the master node acts as the gateway of the voice module sub-node, acts as the AP device, and emits hot spots for the sub-node to access. The two constitute two-way communication within the LAN, namely WiFi communication. When the voice command is recognized by the ASR offline voice module, it is sent to the sub-node through the serial port, and the voice signal command is transmitted to the MCU master through the wireless WiFi network. The MCU controller judges the command content and drives the home equipment or controls the running status of the home equipment, and controls the home equipment.

III. HARDWARE DESIGN

3.1 HOME CONTROL DESIGN

As shown in Figure 2, the home master module is designed based on the Arduino Mega 2560 development board, and the NodeMCU development board with ESP8266 as the core is used as the network transceiver master node. The household hardware equipment includes OLED display screen, electric fan, LED light, alarm module, DHT22 temperature and humidity sensor module, BH1750 illumination sensor module, RFID radio frequency module, MQ2 smoke sensor module, stepper motor and its drive plate, and steering gear module.

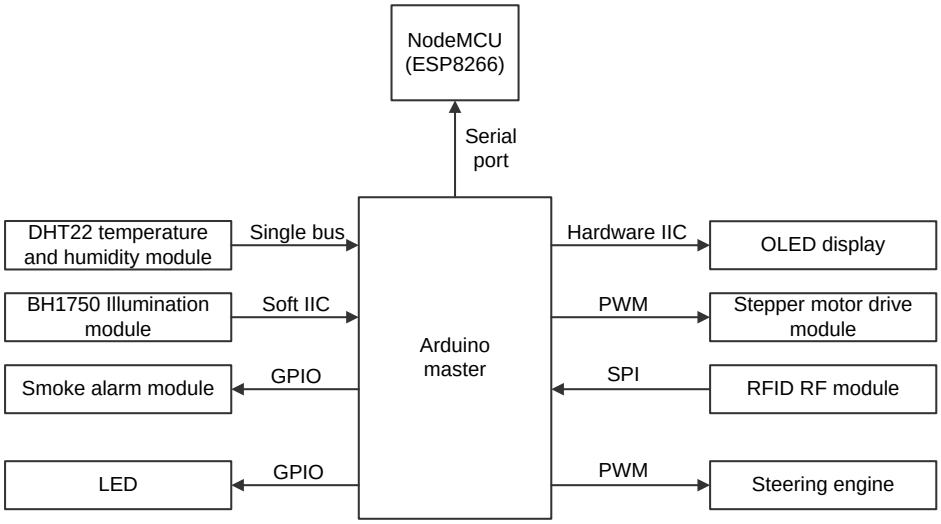

Figure2: Home master hardware block diagram

The temperature and humidity module is used to measure the indoor temperature and humidity. Compared with the commonly used DHT11 module, the DHT22 temperature and humidity module is used in this design. Compared with DHT11, DHT22 has higher measurement accuracy, with temperature error less than 0.1 ℃ and humidity error of 0.5% RH. The measuring temperature range is from - 10 ℃ to 50 ℃, which is very suitable for household systems.

The function of Arduino MCU is to receive the instructions from the master node through the serial port, and then execute the instructions to control the work of hardware equipment after recognition. In addition, the home master also has the function of self-test control.

As the access control system of the family, the RFID RF module can only drive the steering gear to open the LED lights in the door and bedroom after the user has swiped the IC card and passed the authentication.

DHT22 temperature and humidity module will automatically turn on the fan when it detects that the indoor temperature is too high. BH1750 illuminance sensor is used to measure the external light intensity. When the external light intensity is detected to be too large, the MCU will control the driving step motor to reverse, representing the process of opening the curtain at dawn; On the contrary, the stepper motor turns forward to open and close the curtain. The function of MQ2 smoke sensor is to measure the concentration of combustible gas and harmful gas in the environment, and the concentration exceeding alarm system is activated. The MCU will notify the master node to feed back to the APP through the network, and can also realize the function of sending alarm messages to the user's mobile phone and pushing alarm notifications through WeChat. OLED display can display the indoor temperature and humidity and the switch status of LED lights and other equipment in real time.

ESP8266 refers to the general name of WiFi modules using ESP8266EX chips. For the WiFi transmission part, the SoC chip ESP6288EX of Lexin Information Technology (Shanghai) Co., Ltd. is selected. The chip is built with Tensilica L106 ultra-low power 32-bit microMCU, integrated with Wi-Fi MAC/BB/RF/PA/LNA, and adapted to various wireless WiFi environments. Built-in Wi-Fi protocol stack and TCP/IP protocol stack support real-time operating system (RTOS) and non-operating system (NONOS) development.

ESP8266 series modules can be developed in the following ways: use the AT command firmware package from Lexin to interact with external host or MCU to provide WiFi connection function. Use the Lexin SDK development package and the provided Eclipse IDE to develop the underlying drivers in C to generate BIN files. Use NodeMCU, brush in the firmware package, and use the lightweight embedded scripting language Lua

script to develop in the ESPlorer software environment. Use Arduino IDE to write firmware, compile and burn it into NodeMCU to use internal resources [4].

NodeMCU is a development board with ESP8266 series module ESP-12E as the core, and is a common development platform for the Internet of Things. It leads out all chip pins with row pins, and supports the USB port of CH340 or CP2102 chips, which is convenient for developing and burning programs. Because the project is based on Arduino development environment, NodeMCU is selected to use Arduino integration library to write code.

The ESP8266 WiFi module has three working modes: STA mode, AP mode, and STA mixed AP mode. In STA mode, the WiFi module can connect to the gateway device, which can be a wireless router or a mobile hotspot AP, access the network, and connect to the MQTT server, Onenet, Smart Cloud, Alibaba Cloud and other cloud servers by running MQTT protocol or other protocols. In AP mode, the ESP8266 WiFi module can act as an AP device, run the WiFi protocol stack, and transmit wireless network signals for the access of hosts supporting WiFi protocol. In the STA+AP mode, the WiFi module can be connected to the gateway to access the Internet server or connected to other hosts as a gateway without routing function [10,11].

3.2 NETWORK MASTER NODE DESIGN

3.2.1 NETWORK PROTOCOL

The network protocol network server master node design includes a variety of network protocols, such as TCP/IP transmission protocol, WiFi wireless protocol, MQTT transmission protocol, etc. In this paper, two hosts using TCP/IP protocol are divided into server and client. The client requests to connect and sends instructions, and the server responds and provides services. When the client sends a packet, the packet will be transmitted from the application layer to the network interface layer from top to bottom. The application program of the client adds control information at the application layer, that is, defines the format of data transmission to form a message stream and transmit it to the transport layer. The transport layer adds the received message flow to the control information of the layer, that is, adds the port number of the client to form a message segment and then transfers it to the network layer. The network layer runs the IP protocol and ARP protocol. If the communication parties are not in a subnet, the routing protocol will run. The routing protocol is run by the gateway, router and other devices. The network layer will identify the client IP address and resolve it to MAC address to form IP datagram. After entering the network interface layer, the data packet is parsed and assembled into a frame, and sent to all hosts in the subnet in the form of broadcast. The server identifies whether the MAC address is the same as its own. If it is the same, the received data frame is transmitted to the hardware device in the form of bitstream. If it is different, the data frame will be discarded. On the other hand, the client receives the server data packets from the network interface layer to the application layer from the bottom up, parses the control information of each layer and finally reaches the client application.

MQTT transmission protocol is the most commonly used communication protocol in modern Internet of Things solutions. In order to realize the communication between devices, TCP/IP protocol, the mode of request response, can no longer meet the requirements. MQTT protocol is a lightweight communication protocol based on TCP/IP protocol. The biggest advantage of MQTT protocol is that it has low bandwidth consumption, low cost and can provide reliable real-time message service for the machines connected to the server, which makes it widely used in the scenario of IOT with high power consumption requirements. The publish/subscribe mode is different from the request/reply mode. It is a mechanism of message distribution. The publishing and subscribing ends are two device ends, and a proxy server, namely MQTT server, is required between the device ends. The MQTT server can be built locally using the Linux system itself, but it is only limited to the device communication in the LAN. If you want to realize the communication between intranet devices and extranet devices, you need to use cloud servers. In this design, the Blinker cloud server is the MQTT server of the public network. In this way, subscribers and publishers do not need to establish direct connections and contacts, but only need to distribute messages through MQTT cloud servers according to the topics of subscribers and publishers. The advantage of this design pattern is that the subscription and the publisher do not need to be directly associated, nor do they need to cause message blocking and deadlock due to requests and replies, which means that the devices at both ends do not need to be online at the same time. The publisher publishes topic messages, and the subscriber can receive messages after connecting to the MQTT server online.

3.2.2 MAIN NODE DESIGN

As shown in Figure 3, it is a block diagram of the master node design scheme. NodeMCU is in the coexistence mode of STA and AP. The gateway runs WiFi protocol and transmits radio wave hotspots outward. NodeMCU receives signals and connects to the wireless LAN port of the gateway. The gateway uses NAT technology to allocate the IP address of the internal LAN for NodeMCU. The QT application runs on the PC side. After the PC connects to the WiFi of the gateway, the QT application is also assigned to the IP address in the same network segment as the NodeMCU. Both can establish a point-to-point connection for TCP communication in the LAN. With NodeMCU as the server and QT application as the client, the QT client

requests the connection through NodeMCU port number and IP address. After NodeMCU confirms the connection, it can display the indoor temperature and humidity information in the QT application and control the household equipment.

Because the gateway has the routing and NAT functions, the NodeMCU accessing the router can access the public network server through the routing function. Here, the NodeMCU intranet private address is converted into the public network address through the NAT technology of the multi-layer router, the connection request is forwarded to the Blinker cloud server, and the MQTT connection is established. After the mobile APP is online, it will automatically connect to the Blinker cloud server. NodeMCU and APP are both subscribers and publishers. Through the subject message distribution of the proxy server, the APP can receive information and remote control at any place. Blinker cloud account can be bound with Alibaba Cloud server to synchronize the device information to Tmall Smart. The same as APP control, NodeMCU connects to Alibaba Cloud server based on MQTT protocol, and can use Tmall Smart to control home devices.

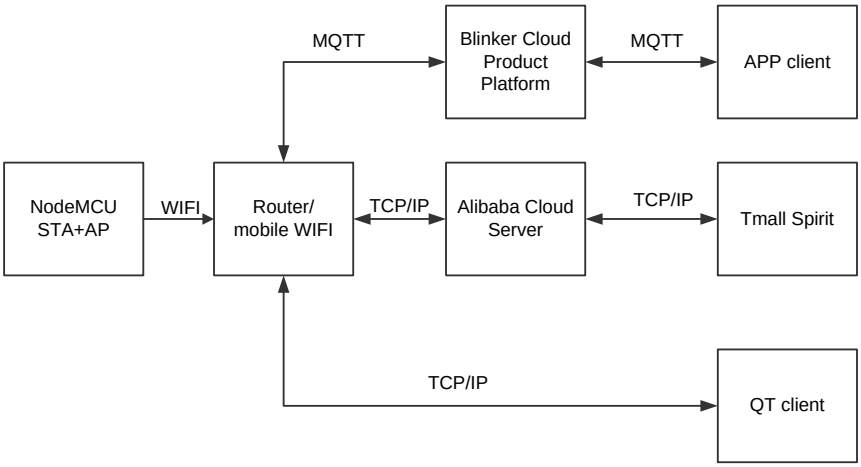

Figure3: Main node design block diagram

3.3 SPEECH RECOGNITION MODULE AND NETWORKING

The speech recognition part is composed of NodeMCU sub-node and ASR voice module. As shown in Figure 4, it is the composition block diagram of the language recognition system. Speech recognition system is a discrete small control system, which adopts the idea of distributed design. As the offline control solution of the system, the distributed networking design reduces the coupling with the networked control scheme.

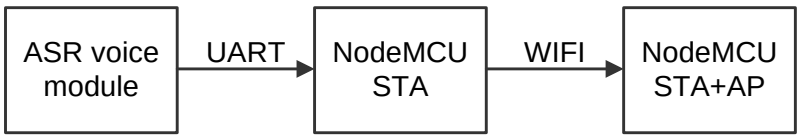

Figure4: Network block diagram of speech recognition system

ASR speech recognition module is designed for integration, with high integration. Built-in 32-bit RISC high-performance low-power LD3320 speech recognition chip, internally integrated with VS1003 decoder, flash with voice model and TF-Card of offline speech recognition kit. This module has powerful functions. It can customize wake-up words, voice command words and reply words online, support Chinese and English language recognition, support continuous voice recognition after wake-up, and has a high recognition rate. It also supports cloud programming configuration and automatic generation of firmware.

The ASR voice recognition module configuration process is as follows: log in to the smart AD cloud server, register an account and log in, create a new offline scheme, select the ASR chip, and start configuring the voice module functions and instructions. After configuration, generate the SDK online. After the SDK is generated successfully, compile and generate the firmware bin file. Use the USB serial port burner to burn the program to the ASR voice recognition module.

The block diagram of voice recognition system networking scheme is shown in Figure 4. The NodeMCU of the sub-node adopts the STA working mode. The master node sets the AP mode to diverge the WiFi hotspot. The sub-node connects to the master node as a client, assigns the private IP address of the LAN, and both establish the point-to-point TCP communication. Here, two ESP8266 modules are used to build local area networks independently, without gateway participation, so as to realize offline network-free control.

IV. SOFTWARE DESIGN

4.1 HOME MASTER PROGRAM DESIGN

Arduino is a convenient, flexible and easy-to-use open source electronic prototype platform, including hardware (Arduino development board) and software (Arduino IDE), which can be programmed and developed in C language and support cross-platform development. Arduino has C++class libraries provided by the official community and third parties. Developers do not need to have too much knowledge of the underlying principles and hardware, and it is convenient and flexible to develop. Arduino mega 2560 is used in the design. The development board uses Atmel's low-cost AVR single chip microcomputer, with built-in ATmega 2560 chip, which has larger memory and richer peripheral interfaces.

When designing the master program, we should first build the Arduino programming development environment. The Arduino IDE can be downloaded and installed on the Arduino official website. Because the ESP8266 module is subsequently used, the ESP8266 environment needs to be configured. After installation, open the Arduino IDE, enter the preferences of the file menu, add the URL image address of ESP8266 in the address of the additional development board manager, and then you can search for esp8266 in the development board management of the tool menu, download and install it.

As shown in Figure 5, it is the flow chart of home master program design. Multiple hardware is used in the master control program. It is necessary to download and install the development support package of Arduino library, and import it as a header file by precompiling instructions. Figure is the logic flow chart of the master program. The first step is to create and instantiate multiple hardware device objects, and define global flag bits and pin macro definitions. In the process, multiple hardware switch function functions are defined to facilitate calling and improve code readability.

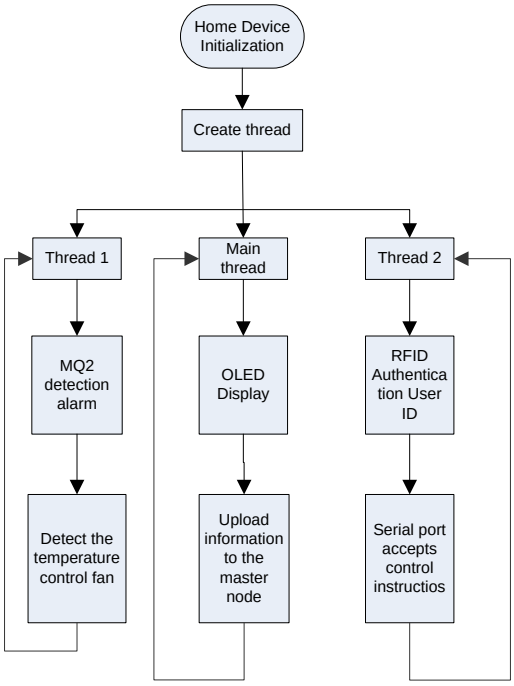

Figure5: Flow chart of home master control program

In the initialization process, the first step is to initialize the serial port monitor between the development board and the PC and the hardware serial port connected to the NodeMCU. Because the serial port is different from RS485 and other buses, it is vulnerable to electromagnetic interference, and may be distorted due to too fast speed in long-distance transmission. After a long time of testing, 4800 baud rate is selected to replace the commonly used 115200 baud rate to improve the accuracy and stability of transmission and ensure the reliability of the system. Next, initialize the LED and fan light hardware pin modes, and then initially use OLED, RFID and other hardware of IIC interface and SPI interface.

In order to improve the response speed of the master controller to instructions and special situations, and improve the real-time performance, after the initialization is completed, the official SCoop library is used to simulate the multithreading technology and create two threads. The Arduino controller does not support the operating system, so the multithreading here is essentially realized by using the timer interrupt method to simulate the working principle of multithreading. The use of multithreading can realize concurrent operation,

enable the system to handle multiple tasks at the same time, and greatly improve the system operation efficiency.

In the main thread, first obtain the temperature and humidity values and light intensity values detected by the temperature and humidity sensor and the illumination sensor, then detect the switch status of LED and other devices, and then drive the OLED screen to work and display these information. Finally, the information needs to be encapsulated in Json data format and sent to the master node. Json is a lightweight data interaction format. Json is not used here because of its simple data format and small bandwidth, which is conducive to server parsing and reading. It also supports multiple languages, with strong compatibility, and is suitable for network transmission and communication.

The task of thread 1 is to detect the smoke concentration value and temperature value in real time. An alarm will be given if the smoke concentration exceeds the standard. If the RFID verification is passed and the temperature is too high, the fan will be turned on automatically.

Thread 2 has the following functions: RFID RF module reads ID card information to verify user identity, and serial port circularly reads instructions. Instructions are defined in the program using macros, and the format is a single character for easy recognition and response.

4.2 MASTER NODE PROGRAM DESIGN

As shown in Figure 6, the main node program flow chart. The first step of the program is also to initialize the serial port. The master node should be connected with Tmall Smart, and the Tmall Smart interface should be macro defined. In addition, in order to associate the components of the mobile app with the program, you need to define the component object in the program, keep the key name of the object consistent with the APP, write the callback function, and register the callback function.

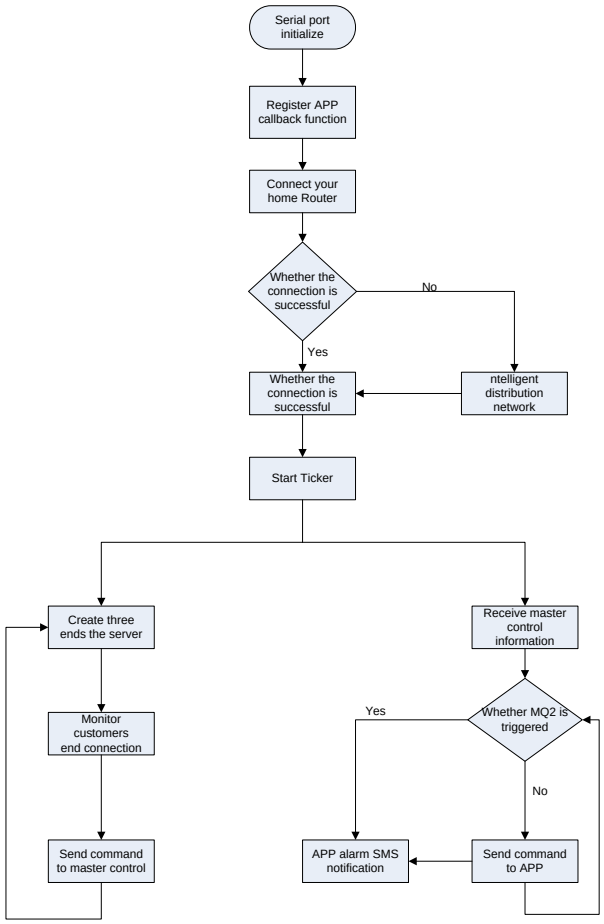

Figure6: Main node program flow chart

Set the NodeMCU master node bit STA+AP mode. Connect the router in STA mode and wait for a period of time. If the router fails to connect repeatedly, NodeMCU will start the smart distribution network (smartConfig). The ESP-Touch APP officially developed by Lexin uses Smart Config technology. When NodeMCU is in the distribution network mode, it will listen for all message information existing in the network.

ESP-Touch parses and encodes the name (SSID) and password (Password) of the mobile phone's WiFi into the UDP message, and sends it to the network through multicast and broadcast. NodeMCU will decode the UDP message after receiving it, and actively connect the WiFi device to complete the connection. If the distribution network reaches the specified number of times and fails to connect, it means that the home network is disconnected. At this time, NodeMCU will actively interrupt the networking operation and establish a connection with the sub-node. At this time, the user will use the offline voice control device in the network-free state.

If the connection to the router is successful or the intelligent distribution network is successful, the Ticker timer will be enabled and the function function of the server receiving instructions will be called regularly. The function is also to simulate multithreading to process multiple tasks. In the function of receiving commands from the server, first configure the APP, QT and voice servers, register and bind the IP and port, then listen and process the connection requests of the client, receive the commands sent by the client connected to the server, and forward them to the Arduino controller.

The main program is mainly responsible for receiving the Json format information from the Arduino controller and sending it to the client that has accessed the main node. At the same time, the main program will also use Json library to parse the received information, and then screen out the key value corresponding to the smoke concentration sensor, and judge whether the concentration exceeds the standard according to the key value. If the concentration exceeds the standard, the user will receive SMS notification and WeChat reminder, and both QT and mobile APP will have alarm prompts.

4.3 SUB-NODE PROGRAMMING

The sub-node program flow is shown in Figure 7. After the serial port is initialized, set the AP mode to connect to the master node, and judge whether the connection to the master node is successful. If the connection fails, continue to connect. Until the connection is successful, the command sent by the voice recognition module will be received, and then transmitted to the master node. After that, continue to judge whether the connection is successful.

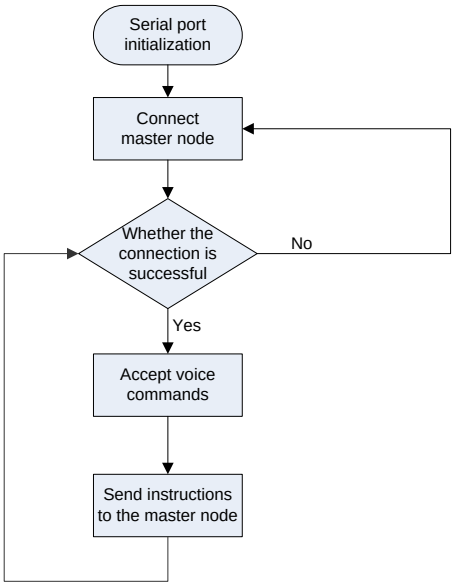

Figure7: Sub-node program flow chart

4.4 MOBILE APP AND QT INTERFACE DESIGN

The Blinker IoT development platform supports Arduino device access and provides APP design functions and API interface functions. Before designing the mobile app, you need to download and install the Blinker client, register and log in, add Arduino device, enter the device terminal, and start designing the UI interface. Users can access the device to the software and use the customized control panel function to achieve personalized control. The development process is as follows: first enter the opening interface for editing, then add components with display, control or communication functions, click components to enter the parameter setting interface, and users can change the color, text content, data key name and display style of various parameters to adjust components. You can drag and drop components, change the location of components, complete the customized layout, and finally save the interface. The mobile APP interface and functions are shown in Figure 8.

Figure8: Mobile APP interface

QT is a cross-platform C++graphical user interface framework. QT supports the development of application interfaces for Windows, Linux, Mac and other systems. It has powerful C++class library support, object-oriented programming, and good scalability. It also allows component programming for simple interface development. The GUI application developed using QT Creator is shown in Figure 9.

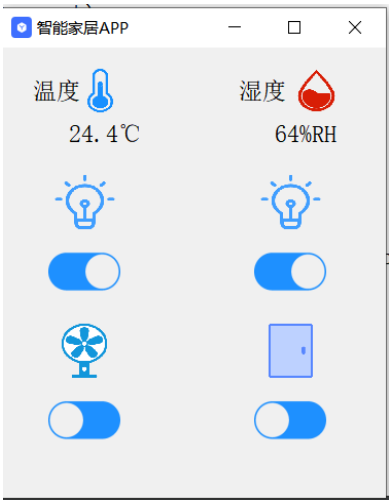

Figure9: QT client interface

V. CONCLUSION

This design uses ASR voice recognition module and two ESP8266 WiFi modules to build a home LAN to realize offline voice control of home equipment. ESP8266 access gateway to realize communication with QT client. Through network access to the Internet of Things cloud platform, mobile APP and Tmall Genie control home devices.

There are still some deficiencies and needs to be improved in the design. Although the system optimizes the serial communication, the serial communication speed is relatively slow, and there is still the possibility of data loss due to interference. A better solution is to use RS485 or I2C communication bus with high speed and strong anti-interference ability to connect.

In addition, the mobile app needs to obtain the switch information of the device and update the status in real time. The intercommunication between mobile APP and devices relies on the Blinker Internet of Things platform, which restricts the amount of data uploaded in order to control traffic and avoid resource waste. The update of APP is delayed and has poor real-time performance. For this, you can choose to rent Tencent or Alibaba's cloud server to build your own MQTT server or pay a certain fee to open the professional version of the Blinker platform to solve the problem of real-time data transmission.

REFERENCES

[1]. Liu Ronghui, Peng Shiguo, and Liu Guoying, Embedded speech recognition system based on smart home control. Journal of Guangdong University of Technology, 2014,31(02):p.49-53 .

[2]. Wang Junzhi, mart home system based on voice interaction. South China University, 2020. DOI: 10.27234/d. cnki.gnhuu.2020. 000290.

[3]. Chen Yanxiu, Design of Zigbee-based voice home control system. Journal of Liaocheng University (Natural Science Edition), 2014,27 (02):p. 98-102.

[4]. Luo Wei, Liu Cien, and Ni Youshun, et al, Design of wireless object finder based on ESP8266EX. Science and technology horizon, 2016(23):p.381-382.DOI:10.19694/j.cnki.issn2095-2457.2016.23.311.

[5]. Wang Aiyun, Application of speech recognition technology in smart home. Software, 2015,36 (07): p.104-107.

[6]. Wang Shuifa and Fan Weihua, Development and research of intelligent safety window monitoring system based on Arduino. Electronic Quality, 2019 (10):p. 43-45.

[7]. Wu Bin, Development status, existing problems and countermeasures of smart home system in China. Wireless Internet Technology, 2016 (24): p.61-63.

[8]. Miao Wei, Intelligent Voice Control Solutions. Shanghai Information Technology, 2019 (10):p. 66-68.

[9]. Zhang Hushi, Lin Weilong, Yang Fazhu, etc., IoT temperature monitoring system based on ESP8266 WiFi module. IoT technology, 2020,10 (12):p. 32-35. DOI: 10.16667/j.issn.2095-1302.2020.12010.

[10]. Huang Xiaoyi, Research on smart home application based on speech recognition and wireless sensor technology. Real Estate Guide, 2016 (06): p.85.

[11]. Zhong Minghui, Cao Naiwen, Huang Hanming, et al, Chinese digital speech recognition system based on continuous HMM. Guangxi Physics, 2007 (04):p. 24-26.