Application of XML and TCP communication on Dynamic customization of the smart home system

Zhihao Li*, Jie Sun, Danzengouzhu, Hai Lan and Wenbo Lu
School of computer science and technology, Jilin University, Changchun 130022, China.

*Corresponding author Email: lizh2718@mails.jlu.edu.cn

Abstract. Aiming at the problem of personalized customization of a smart home system, a scheme of personalized customization of the smart home system based on parsing XML authorization files is proposed. First, develop an XML authorization file generator, and use it to generate personalized XML authorization files; secondly, use the developed app to parse the XML authorization file to realize the legal authorization of the app and the dynamic generation of the home control interface; finally, click buttons on the control interface to trigger the TCP communication between the app and ESP8266, and then with the help of esp8266 to realize the switch of electrical appliances. Experimental results show that dynamic customization of the smart home system based on XML can implement dynamic customization of control interfaces for various home layouts, and can effectively control the home furnishings.

Keywords: Smart home system, XML, ESP8266, TCP communication.

1. Introduction
Smart home (smart home, home automation) uses residential as a platform, which uses integrated wiring technology, network communication technology, security technology, automatic control technology, audio and video technology to integrate facilities related to home life, build an efficient management system of residential facilities and family daily affairs, improve home safety, convenience, comfort and artistry, and realize an environmentally friendly and energy-saving living environment [1]. The smart home system can perform wireless and convenient control of the home equipment in the home, which is a necessary configuration for the daily life of families in the future. The Smart home system has become a hot topic in the research of the Internet of Things. The traditional smart home system generally uses Bluetooth and Wi-Fi to connect smart home gateways to mobile phones and control them indoors through mobile apps (App), but its personalized customization is not yet perfect and the cost is high [2]. In response to the above problems, a dynamic customization of the smart home system based on XML is designed. The system can generate electrical control interfaces in a personalized manner by parsing XML files and use the ESP8266 with a WIFI module as a module of electrical control to greatly save the cost of development. The system is mainly divided into three modules: the XML authorization file generator module, the mobile app module, and the electrical control module based on ESP8266.
2. The whole frame

2.1. The design of the whole structure of the system.

The system is divided into three modules: the XML authorization file generator module, the mobile app module, and the electrical control module based on ESP8266. The XML authorization file generator module is used to generate XML authorization files; the mobile app module is used to parse XML authorization files, authorize app users and dynamically generate control interfaces based on XML files, and realize TCP communication with ESP8266 to control electrical switches; the electrical control module based on ESP8266 is used to realize TCP communication with the mobile phone app and control the switches of the electrical appliances according to the received mobile phone app instructions.

2.2. The system flow.

First, obtain the personal information of the user and the distribution of electrical appliances in the home, and input the corresponding information through the XML authorization file generator to generate an XML authorization file. The authorization of the user is achieved by MD5 encryption on the mac address of the user's mobile phone.

Secondly, the mobile app copies the XML authorization file to the app's exclusive directory and parses it. First, the app obtains the mac address of the mobile phone for MD5 encryption and compares it with the MD5 encryption field in the XML authorization file. If they are consistent, the authorization is successful and the app dynamically generates control interfaces based on the XML authorization file, for example, there is a bathroom and a living room in the home, and there are two lights in both rooms. Input these information in the XML authorization file generator and send it to the app. After the app reads the XML file, the bathroom and the living room each generate an interface, and two switch buttons are generated in each interface to control the corresponding two lights; Otherwise, the authorization fails and the app cannot be used.

Finally, the ESP266 receives the command sent by the app through the WIFI module to control lights on and off. In addition, it can also control lights on and off through physical switches.

The basic flow chart of the system is shown in Fig. 1.
3. Functional module

3.1. The XML authorization file generator.

The XML authorization file generator is a PC program to store the input text information as nodes of an XML file by creating a Document object and finally generate an XML authorization file. The main nodes of an XML authorization file are:

User information node:

There are multiple sub-nodes related to user information, such as user name, user mobile phone number, user mobile phone mac address and so on. The code sub-node saves the string obtained after MD5 encryption of the mobile phone mac address, which is used for legal authorization of app users.

Multiple room information nodes:

There are multiple electrical equipment sub-nodes under these nodes, which are used to store electrical equipment information, such as equipment name, equipment IP address, equipment port number, equipment control signaling and so on. The IP address and port number are the IP address and
port number of the ESP8266 that controls the corresponding appliance, and the control signaling is the command received by the 8266.

3.2. Mobile app.

The mobile app is mainly divided into three interfaces, the authorization login interface, the control interface, and the setting interface.

Use Pull to parse the XML authorization file. First create a Bean object corresponding to XML, and then create an XMLPULL parser to parse each node under the XML authorization file.

The authorization login interface: grant the app the permission to read the phone storage and the phone mac address. Firstly, judge whether there is an XML authorization file under the specified file path. If it exists, then determine whether the mobile phone mac address encrypted by MD5 is equal to the encrypted field (code) in the XML authorization file. If they are equal, the authorization is successful and enter the control interface.

The control interface: after parsing the XML authorization file, use ViewPager and Fragment to dynamically generate multiple cyclically sliding control interfaces. There are corresponding control buttons under each control interface, which are used to control the switch of the corresponding electrical appliance. When a control button is pressed, it is first judged whether the app has established a TCP connection with the corresponding 8266. If it is not established, a connection is established first, and then a control command is sent to the 8266 to control the electrical switch.

The setting interface: the current setting interface has realized the function of deleting the XML authorization file in the app's exclusive directory, and more setting options can be added later according to user needs.

The flow chart of app usage is shown in Fig.2.

Fig. 2 The flow chart of app usage
3.3. **Electrical control based on ESP8266.**

ESP8266 is a complete and self-contained Wi-Fi network solution that can carry software applications or uninstall all Wi-Fi network functions through another application processor.

The power supply voltage for ESP8266 is 5V, and the household voltage is 220V, so it needs to go through the 220V to 5V step-down power module in Fig.4 to provide 5V working voltage for ESP8266. The step-down power supply module supplies power to the 8266 by connecting to the Vin pin in Fig.3.

The output voltage of ESP8266 is 5V, so the 5V output must be converted to 220V through the relay in Fig.4 to control the household appliances. The VU and GND pins in Fig.3 are used to supply power to the relay, and the D5 pin is used to control the relay to pull in or float, and then control Electrical switches.

ESP8266 has three working modes, namely wireless access point mode (AP), wireless terminal mode (Wireless Station) and hybrid mode (a combination of the above two modes) [3]. What we want to achieve is to connect the ESP8266 and the user's mobile phone to the same local area network and communicate with each other through the TCP protocol. This requires that the IP address and port number of the 8266 are always fixed, so that the mobile phone can communicate with the 8266 reliably.

![Fig. 3 ESP8266 (NodeMCU) Pin diagram [4].](image-url)
4. Experimental results

4.1. Running result.

By parsing the XML authorization file, the Android app can dynamically generate multiple home control interfaces corresponding to the XML authorization file, as shown in Fig. 5. Click buttons in the mobile app control interface to turn on/off the bulb, the corresponding ESP8266 receives the command to turn on/off the bulb and controls the relay through the D5 pin to make bulbs in the on/off state, as shown in Fig. 6.

4.2. Analysis of results.

The test results show that it is completely feasible to dynamically customize the control interface for different users by parsing XML authorization files, as shown in Fig. 5, after the mobile app is authorized successfully, by parsing the XML authorization file, multiple home control interfaces and control buttons can be dynamically generated to control the switch of the electrical appliances. In addition, ESP8266 can effectively control the electrical appliances. Fig. 6 shows the ESP8266 electrical control module which has connected to the circuit. When control buttons on the app are clicked, the corresponding ESP8266 receives the control commands sent by the app through TCP.
communication and controls the pull-in and suspension of the COM and NO interfaces of the relay by controlling the D5 pin, and then controls the light bulbs’ switches. In summary, in view of the problem of customizing a personalized smart home system, it is practical to propose a solution for customizing a personalized smart home system based on parsing XML authorization files.

5. Summary
Aiming at the problem of customizing a personalized smart home system, this paper proposes a plan for customizing a personalized smart home system based on parsing XML authorization files, and elaborates on it in the article. The system is divided into three major design modules, namely the XML authorization file generator module, the mobile app module, and the ESP8266 electrical control module. In the article, these three modules are deeply analyzed and explained. In addition, the article also clarified the test analysis of the system. The test results show that it is completely feasible for the mobile app to dynamically customize the control interface for different users by parsing the XML authorization file, and the ESP8266 can receive control commands through TCP communication to effectively control electrical appliances. Therefore, in view of the problem of customizing a personalized smart home system, it is practical to propose a solution for customizing a personalized smart home system based on parsing XML authorization files.

References
[1] Ruan Xing, Cai Chuanghua. Implementation of an application example of intelligent lighting based on ZigBee protocol[J]. Journal of Chifeng University: Natural Science Edition, 2011 (8).
[2] Ye Gaoyang, Bi Ran. Design and implementation of smart home system based on the Internet of Things [J]. Computer Applications, 2014, 34 (S1): 318-319, 351.
[3] Information on: https://wenku.baidu.com/view/035d72da773231126edc6f1af00bed5b8f37334.html
[4] Information on: https://www.yiboard.com/thread-1548-1-1.html