Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 154 (2019) 40–46

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Congress of Information and Communication Technology, ICICT 2019

Smart Gate System Design and Implementation Based on Cloud Platform

Yue Jia Xin[a] , Wang Zhong[b*] , Lai Hong[b]

[a]Sichuan Film and Television *University*, Chengdu, 610035,China
[b] College of Electrical Engineering and *Information* ,Sichuan University,Chengdu, 610000,China

Abstract

In view of the deficiency of traditional entrance guard control, a more convenient intelligent entrance guard control system suitable for home remote control is designed. WeChat small program development is adopted at the front end of the system， and the management control center is built by cloud server, and the hardware controller is designed with Raspberry Pi ,which is an embedded development board with low power consumption and high performance. The functions of user management, device management and unlocked control are realized. The response performance and safety performance of the system are optimized and the response time is reduced.

Keywords: smart home; Cloud platform; WeChat small program; Raspberry Pi; The entrance guard control

1. Introduction

In recent years, the wave of themes of smart cities, smart life, and smart homes has constantly impacted the traditional access control system, resulting in the constant enlargement of traditional access control deficiencies.For password lock,it is cumbersome,the password is easy to leak, the security is low and we got less using scene.For magnetic card, it is prone to the phenomenon that the key is forgotten.Also, the magnetic card has the disadvantages

* Corresponding author.
E-mail : 1648277629@qq.com

10.1016/j.procs.2019.06.008

of being easily damaged and demagnetized.For biometric access control, the cost is high. It also has a high requirement for users and the using environment, because the users' characteristics change will lead to unrecognized phenomenon[1].

With the widespread use of WeChat, it is of great significance to develop a convenient access control management system based on mobile devices with excellent security performance and compatible performance using small programs born in the WeChat ecosystem.

2. The Overall Design of the System

The design of the intelligent gate control system is based on the ideas provided by the Internet of Things (IoT) and the services provided by the cloud computing model, enabling users to remotely control the door through the Internet.The system structure is divided into user front end, cloud management control center and hardware control end.The front end faces the user and provides the user with a simple management and control function of the device;the cloud management and control center implements the logic control of the entire system, including user identification, device identification, and the forwarding of unlock commands;the hardware control end connects with the background server and accepts the message instruction from the server to complete the unlocking function[2].

The framework of the system is shown in Figure.1.The user accesses the background management and control service program located in the cloud through the application program on the mobile phone. After the user is authenticated and authorized, the device name in the user request data and the GPIO pins controlled by the device are stored in the database or forwarded[3]. The hardware control service program built for the Raspberry Pi is verified as an available pin, and a hardware control program is called to control the GPIO pin to output a high level and the high level can control the relay to disconnect the main circuit it controls. The electromagnetic lock unlocking function is realized.

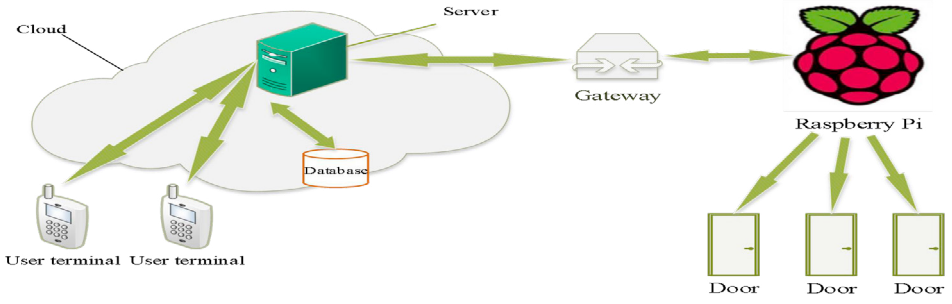

Figure.1 The frame structure of the system

3. Hardware Selection and Programming Control

The hardware uses Raspberry Pi development board. The Raspberry Pi Development Board is an ARM-based computer motherboard. As an embedded device whose size is only the size of an ID card, it not only supports a rich development environment, but also is a high-performance, low-power (IOT) product[4].

The Raspberry Pi supports the linux system. The system must be burned into the SD card in advance.The Raspberry Pi has a WiFi wireless network card and an Ethernet wired interface, both of which can connect to the Internet. At the same time the device is equipped with HDMI interface, login Raspberry Pi can use HDMI external display or SSH login through the network.

The Raspberry Pi has a GPIO interface, and the GPIO driver has been compiled in the kernel[5]. The user can control the GPIO pins to implement many interesting functions by calling the system's GPIO library.Commonly used GPIO operation libraries include Python-based Python GPIO, C-based WiringPi, and C-based BCM2835 C Library driver.The raspbian operating system based on debian is chosen in this paper. The system includes two versions of Python 2.7.14 and 3.6.5 for free choice

Table.1 Raspberry Pi 40pin GPIO Interface List

Wiringpi coding	BCM coding	function name	Physical pin BOARD coding		function name	BCM coding	Wiringpi coding
		3.3V	1	2	5.0V		
8	2	SDA.1	3	4	5.0V		
9	3	SCL.1	5	6	GND		
2	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29

With the Raspberry Pi model 3B, 1G RAM, and 40Pin GPIO port, one device can control multiple doors. Table.1 shows the function list of the Raspberry Pi GPIO pins. The BOARD encoding mode is used to control the GPIO ports.

Using the Python programming control module incorporated GPIO output pin-level settings. The code is as follows.

```
import RPi.GPIO as GPIO
import time
import sys
GPIO_PIN = int(sys.argv[1])
GPIO.setmode(GPIO.BOARD)
GPIO.setup(GPIO_PIN, GPIO.OUT)
GPIO.output(GPIO_PIN,GPIO.HIGH)
time.sleep(10)/**Setting level output time
GPIO.cleanup(GPIO_PIN)
```

4. Software Design

4.1 WeChat Applet design

The system uses the WeChat applet at the front end, and its development framework is

WXML+WXSS+JS+JSON. It has the following features during development.

Do not support cookies.

Provide a local cache API.

The network request adopts HTTPS protocol, which enhances the security of the service.

The Request request URL only supports the domain name,the IP address will be treated as an illegal domain name.

The access control has three function modules, which are a user management module, a device management module, and a device control module.Among them, user management implements functions of usr's registration and login control.Device management implements functions such as device addition, pin addition, and device viewing. Device control refers to the unlock function.

Details of business processes on Wechat applet are as follows.The user opens the applet and first loads the login page. After the registered user enters the user name and password, it enters the personal home page of the applet after being verified in the background. The unregistered user needs to click the registration button to perform the registration operation.The user home page provides device addition, device viewing, and logout features.Enter the device add page, enter the device information that meets the requirements can be added successfully, the program will automatically jump to the current user's device console to view the information added before.Select a device to enter the pin console under the device to view the GPIO pin information. To enable the door opening function, the user only needs to select the device and click unlock to finish.

4.2 Cloud management system design

The system cloud uses the node.js and npm management packages to set up an HTTP server to implement the management and control center.Node.js is a JavaScript language that runs on the server, which greatly facilitates the development of the front end and back end of the system.node.js features asynchronous callback for highly concurrent multi-user access to the HTTP server, relying on the use of non-blocking asynchronous programming using the system, you do not need to wait for the next one thread can perform a complete thread[6].

The database uses mongodb. Mongodb is a document-oriented nosql database with very strong extensibility. With the use of the system, the growth of storage data and the optimization of the structure, mongo can meet the system requirements and optimize the system to some extent.

The communication between the cloud and the Raspberry Pi device is implemented by means of an HTTP request response[7]. The peanut shell tool is used to configure the network environment where the device is located to implement the intranet penetration function.

The cloud mainly implements the logic control of the entire system, including user management, device management, and the function of unlocking instructions.

User management is mainly for user identification. When the user registration and login information is transmitted to the cloud, the user's identity is verified by invoking the query database to implement the user's management control. The system verifies the user's login status when performing other operations[8]. The specific implementation requires the server to refer to the cookie-parser library and the express-session library for parsing cookies and creating sessions. When the user logs in successfully, it creates a session that indicates the user's login status, and sends the encrypted sessionID to the client applet through the set-cookie. The next session to perform other operations requires verification of the session user.

The device management module is mainly device add function, pin information addition, list view function. In the device adding function, the system design adopts a third-party intranet penetration tool. To access the Raspberry Pi, you need to obtain the domain name and the service opening port number. The device identifies the device with a user-defined device name. Therefore, the added device information is the domain name, port number, and device name.When designing a database, some of the logical structure needs to be completed by code[9]. The user adds device information, transmits it to the server, and the server queries the database. The result feedback is designed in three cases. First, the user already owns the device name, adds duplicates, and the system fails to respond to device addition. Second, the user did not add the device, and the device library does not have the IP and device name of the device[10]. The system responds that the device is added successfully. Third, the device already exists, but the user does not add the device. The system responds to device binding successfully,this applies to the control of family

members.

The server-side interaction with the database requires a database driver. Use require ('mongodb').MongoClient call the database driver, MongoClient.connect (url, function (err, db) {}) to connect to the database, db.collection.find () to query the database information, complete the verification of the device to add information, db .collection.update() completes the insertion and update of data.

The forwarding process of the unlocking instruction is shown in Figure.2 . When the user clicks the unlock button, the cloud accepts the device information transmitted by the client, performs a database query to obtain the domain name, the port number, and the GPIO pin serial number to creates an HTTP request to send to the Raspberry Pi device. The Raspberry Pi accepts instructions, calls the python program to unlock, and the user waits for the device to respond.

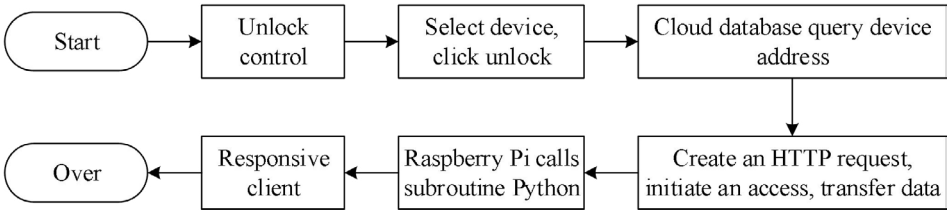

Figure.2 Equipment Control Flowchart

4.3 Cloud Management Control Center Optimization

Improving the WeChat access control by using intranet penetration tool, and WebSocket was used to achieve the coupling of cloud management and Raspberry Pi hardware control. The Raspberry Pi device and the server establish a long two-way communication channel.The cloud creates a WebSocket server that listens for port connection events and data transfer events[11]. When other devices connect to the server, the unique socketid[7] is stored. The socketid flag is different for different connection events of the same Raspberry Pi device. In order to identify different devices, the system needs to set a unique identifier for the entire network.

The data transmission format uses JSON serialization. To distinguish between connection objects, the system uses the source tag object, pi stands for the Raspberry Pi terminal control device, and user stands for the user. If the connection object is a hardware controller, the server will store the devicename and socketid of the received data in the mongodb database and determine that the device is in a connected state[12]. If the connection object is a user, it means that the user sends an unlock command and queries the socketid of the database information according to the devicename in the transmission data to determine whether the device is online. The device sends an unlock command online, controls the device to unlock, and the device responds immediately when the device is offline.

The server also listens for disconnected events on the device and removes the devicename and socketid in the database.

Specific development using socket.io library of npm open source library, socket.io version 2.1.0, ease of use, good support for cross-platform compatibility, and adaptive features of real-time communication mode of choice[13].

WebSocket is based on the HTTP protocol, relying on the HTTP service to complete the first handshake, so the server uses socket.io mounted in the HTTP server mode, the two listen on the same port, can complete the HTTP and WebSocket provide services at the same time[14].

5. Design Results

5.1 Unlock Function Test

The system's unlocking function is tested, click the unlock button, after a short delay, the Raspberry Pi responds to the request, pin 12 output level 3.32V, duration 10s. As shown in Figure.3, the system operates normally and the unlocking function is completed.

Figure.3 Unlock Response Level Output

5.2 Analysis of Response Performance

When testing the work condition of the gate control function, both control systems respond to the client after a delay of a certain time. Test the response time and compare the response performance of the two device control systems.

Table.2 Performance Comparison Before and After Improvement

Testing frequency	1	2	3	4	5	Average response time
HTTP WeChat Access Control System Response	11.81	11.37	11.25	11.13	11.06	11.324
Improved Websocket device control response	10.88	10.83	10.81	10.74	11.03	10.858

Note: The output duration of the Raspberry Pi GPIO pin is set to 10s, which means the unlocking duration is 10s. So the actual response time is the test time -10s.

As shown in Table.2, comparing the response performance before and after improvement, the improved control is better than HTTP's WeChat access control, and the average response time is 0.466s less.

The analysis shows that the HTTP-based gating system communication includes a third-party tool intranet penetration tool, and the background access server needs to create an HTTP request service, that is, an HTTP handshake action is required [5]. In the communication of the WebSocket device control system, The front-end and server are upgraded via HTTP handshake to establish a WebSocket connection. The server and the device only need to transmit data. There is no need to send a handshake request and there is no third party to resolve the server query[15]. It can be seen that the improved device control performance response after the improvement is better than the HTTP-based WeChat access control system, which is in agreement with the test results.

6. Conclusion

This paper takes the cloud platform as the management and control center, uses the advantages of WeChat applet, such as no-downloading, large user traffic, many interface APIs, and simple development . It selects the low-power and high-performance embedded development board Raspberry Pi as hardware control, development and design a remote door opening system. The system design of this article still has deficiencies in terms of safety performance and integrity, and further optimization is needed in future in-depth studies.

Acknowledgement

The authors would like to thank the reviewers for the detailed reviews and constructive comments, which have helped improve the quality of this paper. This work is supported by Science and technology support project of Sichuan science and Technology Office under Grant No. 2015FZ061 and the key projects of education department in

Sichuan Province under Grant No. 18ZA0307, 18ZA0308 and the aviation science foundation project under Grant No. 20100011904.

References

1. Zhang Yaoguang. “Opening the door of mobile phones” will lead the development of smart access control industry[J]. China Public Security, 2016(5):71-72.
2. Hulusi. The Future of Access Control Technology [J]. Computer Security, 2012(11):16-17.
3. Yan Ping, Yi Runzhong, Tong Liang. DDNS and NAT based dynamic mapping between servers inside and outside networks[J]. Computer Engineering, 2008, 34(20):136-137.
4. Alex Bradbury, BenEverard. Raspberry Pi Python Programming Guide [M]. Machinery Industry Press, 2015.
5. Xie Xiren. Computer Network (Fifth Edition)[M]. Publishing House of Electronics Industry, 2008.
6. Chodorow K, Dirolf M. MongoDB: The Definitive Guide[M]. O'Reilly Media, Inc. 2013.
7. Cosmina I, Harrop R, Schaefer C, et al. WebSocket[J]. 2017.
8. Zhu Hangjiang, Pan Zhenfu, Zhu Yongli. "Internet +" Intelligent Access Control System [J]. Electronic Technology Applications, 2017, 43(3): 124-126.
9. Liu Lina. Talking about Session mechanism and Cookie mechanism [J]. Computer Programming Skills and Maintenance, 2008(16):28-29.
10. 10. Springer S. Node.js[M]. Galileo Press, 2013.
11. 11. Hui Chunyang, Chen Zhihao, Hu Tingting, etc. Access Control System Based on WiFi and Smartphone[J]. Internet of Things Technology, 2016, 6(3):7-7.
12. 12. Zheng Jieyi. Research on Intelligent Access Control System Based on Bluetooth Technology[J]. Wireless Network Technology, 2017(2):45
13. 13. Hongyan L. Design and realization of smart home terminal applications based on IOT technology[J]. International Journal of Smart Home, 2015, 9(8):123-132.
14. 14. Chan M, Campo E, Estã¨Ve D, et al. Smart homes - current features and future perspectives[J]. Maturitas, 2009, 64(2):90-97.
15. 15. Rui W, Song H. Design and Implementation of Blind Family Members based on WeChat Applet[J]. China Computer & Communication, 2017.