Journal of Research in Science and Engineering (JRSE)
ISSN: 1656-1996 Volume-7, Issue-9, September 2025
DOI: 10.53469/jrse.2025.07(09).04

The Smart Home Control System based on Raspberry Pi

Ronghua Jiang

Jiandong Vocational and Technical College, Changzhou, Jiangsu, China

Abstract: ***With the development of Internet of Things and sensor network technology, the implementations of smart home control system have become more various than before. In this paper, a new smart home control system based on Raspberry Pi is proposed, with the features of human movement alert, temperature and humidity monitor as well as outlet control. The system is available for both desktop and mobile platforms, enhancing the security of the house and providing more convenience to the user.***

Keywords: Internet of Things, Smart Home, Raspberry Pi**.**

1. Introduction

Smart home control system is usually based on Artificial Intelligence, Internet of Things and automatic control technology, connecting multiple appliances in the house such as power outlet and door lock, achieving the central control for the user, and sometimes there are cloud computation, sensor network involved, which expands the functionality in various aspects. The main goal of smart home control system is to make people's life more comfortable, safe and flexible, improving the life quality.

Nowadays, there are various smart home control systems in our life. For the house light, the system usually offers the remote on and off switch, adjusting the brightness and color, sometimes also the situation mode, which usually contains like standard, cinema, gaming and other scenarios, and sometimes the light system can be connected with curtains to the windows as well. When it comes to home security alert, the real time video streaming is necessary, enabling the user to monitor the house status remotely. Based on the video data analysis, some systems can detect if there is movement in the video feed, and send alert messages to the user if unusual movement happens in the video feed. Other than the video alert, it is possible to integrate other intrusion detection such as human infrared and pressure pads detection, which are triggered if a person enters the detection area or steps on the pressure pads. Meanwhile, some control systems introduce the gas leak detection with certain gas sensors, sending alert messages once the gas reaches a certain value in the house.

Moreover, many control systems can be activated by certain words that user needs to say, by understanding which the system can control and communicate with more appliances easily and wisely. Besides the basic control by the voice, the voice-based smart home control system can work as the personal assistant by answering various questions and having daily conversation with the user.

2. System Architecture

This system architecture mainly contains two parts, respectively for the daily use and maintenance purposes. For the daily use, the system is available via various browsers, available on desktop, tablet and mobile phone platforms. Other than the daily use purpose, when the system needs to be maintained or updated, the user can log in to the system via SSH client while the SSH server runs on the Raspberry Pi OS, which is usually conducted on a desktop platform. By logging in the system, the username and password are needed during the login procedure. Once the user is authenticated successfully, the system operation becomes available with sudo permission, where the user can edit the smart home control system and perform other operations. The system architecture is shown as Figure 1.

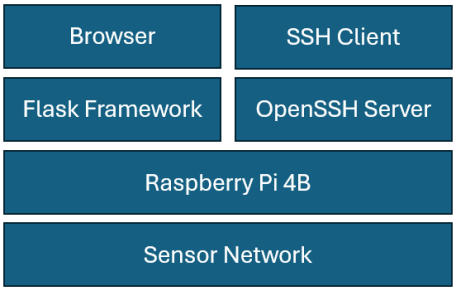

Figure 1: System Architecture

3. Hardware Modules

The main hardware of this smart home control system is Raspberry Pi 4B model shown as Figure 2. It is a small development board with 64-bit processor and 2GB RAM. The 4B model has the standard GPIO pins, also known as General-Purpose Input/Output pins. It is powered at DC 5V by the Raspberry Pi official AC/DC adaptor, whose input voltage is AC 220V. The TF card is installed on the Raspberry Pi with the official Raspberry Pi OS. After everything is ready, it is time to connect the USB keyboard, HDMI monitor and the Type-C power adaptor, waiting for the Raspberry Pi OS boot procedure completes, where there might be a reboot to make changes valid.

Figure 2: Raspberry Pi 4B model

Once the boot initialization finishes, user needs to create a system account with username and password. For the test

purpose, in this system the username is set jrh and password is set the same as username, but for production environment, the password should be strong and kept safe. The default hostname is raspberrypi, which can be changed by the hostname file if necessary.

Various sensors are involved in the smart home control system, and all the sensors are working as a sensor network to implement the designed functionality. The human infrared detection feature is based on the HC-SR 501 sensor, which is shown as Figure 3. HC-SR 501 is capable to detect human infrared, supporting both single and repeatable modes. When working at the single mode, the output value goes to high on detecting human infrared and sustains high value output till the delay timeout, which can be set on board directly. However, the repeatable mode automatically resets the counting down to delay timeout every time the human infrared is detected, and outputs low value after the timeout of delay time without human infrared detected. In this system, the HC-SR 501 sensor is set to repeatable mode so it keeps outputting high value unless the human leaves after the delay time, and it is connected to the Raspberry Pin by the GPIO pins and Dupont cables.

Figure 3: HC-SR 501 Sensor

For temperature and humidity measurement, the Modbus-RTU temperature and humidity sensor is used in this system. Modbus-RTU is widely used in sensor network area, which is usually based on RS-485 interface. The Raspberry Pi 4B model works as the master node while the sensor works as the slave node. Only the Raspberry Pi sends queries to the sensor, while the sensor always stays passive and never sends request to the Raspberry Pi. By implementing sensors with Modbus-RTU, it works efficiently by sending binary encoded data with strong robust check, easy for system integration and debugging purpose. The temperature and humidity sensor is shown as Figure 4, where there are five pins defined as RS-485 A, RS-485 B, DC power positive, DC power negative and the empty pin. The RS-485 A and B pins are connected to the USB to RS485 converter and DC power positive and negative pins are connected to the DC 12V adaptor.

Figure 4: Temperature and Humidity Sensor

To communicate with Raspberry Pi, a USB to RS485 converter is needed. It works as a media to connect between the RS485 devices and Raspberry Pi, where the USB plug is connected to Raspberry Pi and the RS485 A/B are connected with the devices while keeping the G and TR pin empty, which is shown as Figure 5. After the converter is connected, the temperature and humidity sensors and relay driver can be powered on by DC adapter with DC 12V 3A power supply.

Figure 5: USB to RS485 Converter

The RS485 relay driver is needed to control the outlet on and off status, where a four-channel relay driver is used in this system, which is shown as Figure 6. There are NC, NO, COM pins for the relay output, where the NC means Normally Closed, NO means Normally Open, and COM means common. All the relay modules are activated by high value of signal with separate LED indicators. The power for the RS485 module is DC 12V while the signal is transmitted from the Raspberry Pi 4B model and converted by the USB to RS485 converter.

Figure 6: RS485 Relay Driver

4. Software Modules

The official Raspberry Pi OS website offers various options to download. In general, there are three types of OS, which are OS with desktop, OS with desktop and recommended apps, and the OS lite. Different types of OS target to different users, and for beginners the OS with desktop environment is more friendly and easy to use. User can also choose the desktop version with recommended applications if user finds the applications useful for development. In this system the OS lite is used for Raspberry Pi 4B model, which does not include the desktop environment by default and consumes less hardware resource compared with the desktop OS and desktop OS with recommended applications.

As there is no default desktop environment offered by the OS lite version, all the operations are supposed to be implemented with the console window. After the system configuration procedure, its recommended to enable the built-in SSH server, which allows login from user computer to operate remotely without a dedicated keyboard and screen. The default SSH port is TCP 22, which can be changed to another TCP port via the OpenSSH server configuration file, which is shown as Figure 7.

```
jrh@raspberrypi:~ $ cat /etc/ssh/sshd_config
# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

Include /etc/ssh/sshd_config.d/*.conf

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

Figure 7: OpenSSH Server Configuration File

In the sshd_config file, it first includes the configuration directory by the Include keyword. Then the four lines starting with hashtag indicate the default server configurations, where the hashtag needs to be removed when user makes change to the items. In this smart home control system, the default port is used for test purpose. The Address Family item indicates the IP protocol type, consisting of any for both IPv4 and IPv6, inet for IPv4 only, and inet6 for IPv6 only. The ListenAddress item takes two separate lines, which indicate the IPv4 binding address and IPv6 binding address, where 0.0.0.0 and: indicate all IPv4 and IPv6 addresses. When a certain IP address is requested to be listened, user replaces the default value with the static IP address. It is advised to test the OpenSSH server configuration file after making changes, which can be done by providing -t option when running the sshd command with sudo privilege.

For the backend service, the flask framework is used for hosting the web content on the Raspberry Pi. Flask is lightweight framework for web application development, and it is widely used in various small and middle scale of web applications. There are mainly basic routing, request processing, and modular rendering features in the default flask framework, making it flexible and simple to use. The flask API is easy to understand, providing detailed documents and manuals, which is friendly to beginners.

5. System Test

The smart home control system is running on the Raspberry Pi 4B model with 2GB RAM, with the OS version of May 13, 2025 with 32-bit architecture, whose kernel version is 6.12. The backend flask server for this smart home control system is hosted by default python, with the version of 3.11.2.

To Communicate with the Modbus-RTU temperature and humidity sensor and relay driver, the minimalmodbus library is necessary to be installed and imported. During the main program initialization procedure, the flask application connects to the human infrared sensor, temperature and humidity sensor, and the relay driver modules, then binds all the network interface cards by the IP of 0.0.0.0, whose port is 1024. Then user needs to get the local IP of the Raspberry Pi, and replace the 0.0.0.0 with the obtained IP address, which can be done via SSH. The backend flask application is running shown as Figure 8, and it can be terminated by pressing Ctrl and C key. Note that the firewall rule of flask application needs to be added so the user request can go through the firewall without being blocked.

```
jrh@raspberrypi:~ $ python main.py
Initializing the Smart Home Control System...
The System is running at http://0.0.0.0:1024
Use a Browser to access the above address.
Press Ctrl+C to Exit...
```

Figure 8: Backend Flask Application

Besides the python flask backend application, the frontend user interface is shown as Figure 9, which contains 6 box elements. The First box is the title of the smart home control system, and displays the current system time to the user. The second box is the temperature and humidity data measured by the Modbus-RTU sensor, which can be updated every 3 seconds automatically or by the button manually, and the text below the values shows the connection status of the sensor, where the color stays green when connected and goes to red when disconnected. Then the third box is for the human infrared detection relying on the HC-SR 501 sensor, which is set to repeatable mode, so when there is no one in the detection range, the status shows clear in green, otherwise it goes to red with the text Danger, whose data can be updated manually as well by clicking the button. On the second row, there are 3 outlets involved in this system, which can be controlled separately, with the return of current status, where the user clicks ON button to turn on the outlet and OFF button to turn off the outlet.

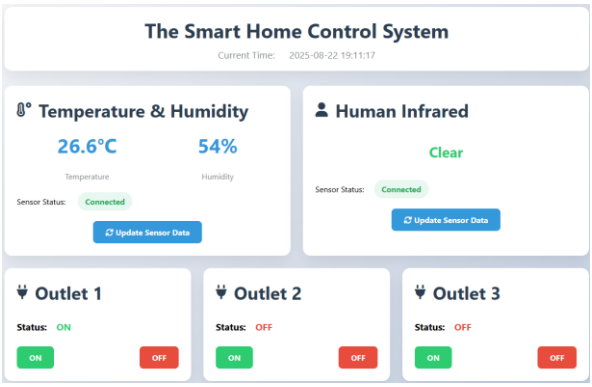

Figure 9: Frontend User Interface

6. Summary

The smart control system is designed to be integrated with human infrared sensor, temperature and humidity sensor, relay driver, aimed to bring people a low cost and high efficiency system based on embedded system technology, electronic engineering, and web development.

When it comes to modular concept, this system is considered as sensor module, control module, service module and application module. The sensor module consists of HC-SR 501 and temperature and humidity sensor, which measure the data and transmit the data to the Raspberry Pi. The Raspberry Pi works as the control module, which is the core module of the whole system, where the Raspberry Pi OS is running. The control module communicates the sensor module via GPIO pins and USB to RS485 converter. The service module is different from traditional implementation, and it is based on the lightweight flask framework, which offers the RESTful API for necessary operations, providing web data for information and control purpose. The application module is usually the user browser, which is used to access the control system user interface by the static IP address. To set static IP address assigned to the Raspberry Pi, the router management is needed by binding Raspberry Pi ethernet MAC address with a static IP address, which is 10.0.0.100 in this system. When it is migrated to a different network environment, the change of router DHCP assignment is needed before using this system.

In the future, there are still improvements that can be

implemented to improve the system user experience. To improve the system security, the password authentication can be used when accessing the frontend user interface, recording the number of failure attempt and alerting the user that illegal operations might be undergoing and actions might be needed to take for staying safe. Database module can also be useful for storing the history temperature and humidity data, human infrared data, and outlet usage, so the user is capable to review the history when necessary, which can be implemented by lightweight database such as SQLite.

To make the system more intelligent, the artificial intelligence technology can be introduced as well, analyzing the past human existence to study the routine of user being at home, predicting the future temperature and humidity information for reminding user to stay warm and healthy, alerting the user when outlet is turned on unexpected based on the user habit and preventing potential risks in advance.

References

[1] Peng Yanning. Research on Scene-Based Smart Home Design [J]. Furniture and Interior Decoration. 2024, 31(2).
[2] Wang Erzhuo, Yuan Xiang, Li Shiyan. Research on Active Interaction Design of Conversational Intelligent Agents in Smart Home Scenarios [J]. Journal of Graphics. 2020, (4).
[3] Nie Hao. Application and Practice of Automation Technology in Smart Home Systems [J]. Television Technology. 2024, 48(2)
[4] Zhu Limin. Design of Smart Home Environmental Information Monitoring System Based on ZigBee Technology [J]. Television Technology. 2023, 47(11).
[5] Ma Yuru, Liang Yungji, Lin Kejiao, et al. Exploration of the Development of Smart Homes Based on the Internet of Things [J]. Intelligent Building and Smart City. 2023, (7).
[6] Gai Sen, Lu Yi, Zhang Yinshuai, et al. Research on Active Interaction Design for Smart Home Systems [J]. Journal of Computer-Aided Design and Computer Graphics. 2023, 35(2).
[7] Sheng Wenjian. Application of Smart Home Control System in Architectural Electrical Design [J]. Value Engineering. 2024, 43(25).
[8] Zou Haizhen. Research on Intelligent Control Strategies for Home Lighting Systems Based on IoT [J]. Household Appliances. 2024, (7).
[9] Fang Yamin. Exploring the Application of Interactive Design in Smart Home Products [J]. Popular Standardization. 2022, (2).
[10] Liu Zhenglai. Design of Smart Home System Application Scheme Based on Internet of Things Technology [J]. Information and Computer. 2024, 36(12).
[11] Yu Yadong, Li Chunjiang, Yang Li. Intelligent Home IoT System Based on Speech Recognition [J]. Computer Applications. 2022, 42(z1).
[12] Dang Dandan, Chen Jiancong. Research and Implementation of Cloud-based Smart Home Internet of Things System [J]. Modern Information Technology. 2020, (16).

Author Profile

Ronghua Jiang received the B.S. degree in Internet of Things Engineering from Huzhou University in 2020 and the M.S. degree in Computer Technology from Xinjiang Normal University in 2023. She is currently affiliated with Changzhou Jiandong Vocational and Technical College, with research interests focusing on the Internet of Things and Deep Learning.