# Smart Home Information Detection System Based on Internet of Things

Xueke Cheng
dept. Automation, Tsinghua University, Beijing, China
cxk21@mails.tsinghua.edu.cn

Zhaohui Ye*
dept. Automation, Tsinghua University, Beijing, China,
Corresponding author:
yezhaohui@tsinghua.edu.cn

## ABSTRACT

The integration of the Internet of Things (IoT) with home products gives rise to a smart home system that significantly enhances the comfort and safety of household living, thereby effectively improving the overall quality of life. The currently available products in the market encompass smart detection devices, smart control devices, and smart security devices. However, these smart home products are limited in functionality, as they are often designed to only cater to a few specific functions. Additionally, the education sector at universities and colleges suffers from a dearth of curriculum that focuses on the construction of smart home systems. This article proposes the use of IoT technology to develop a comprehensive smart home system capable of sensor data collection, video monitoring, remote inquiry, and remote control. Furthermore, programmable development boards are employed to support secondary development and serve as an experimental system for educational purposes.

## CCS CONCEPTS

• **Computer systems organization** → Embedded and cyber-physical systems; System on a chip.

## KEYWORDS

Internet of Things, Smart Home, Programmable Development Boards

**ACM Reference Format:**
Xueke Cheng and Zhaohui Ye*. 2023. Smart Home Information Detection System Based on Internet of Things. In *2023 5th International Conference on Internet of Things, Automation and Artificial Intelligence (IoTAAI 2023), November 24–26, 2023, Nanchang, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3653081.3653194

## 1 INTRODUCTION

With the advancements in communication and network technology, the Internet of Things (IoT) has emerged as a transformative concept that has greatly improved the quality of people's lives. By harnessing communication technology, the IoT allows for the effortless collection and transmission of data from various devices and sensors, enabling seamless information exchange among multiple devices. Applications of the IoT have seen a significant surge in recent years, including the creation of smart cities, smart homes, and intelligent transportation systems [1].

The Internet of Things (IoT) technology facilitates home automation and enhances intelligence. Upon connecting home electronic products to the IoT, users can conveniently monitor and adjust the home environment [2]. Common home products can be classified into three categories: smart detection devices, smart control devices, and smart security devices. Smart detection devices employ sensors to detect the environment and display the information on electronic screens. Examples of these devices include temperature recorders and humidity monitors, as well as gas detectors. On the other hand, smart control devices are capable of receiving external control signals, allowing users to change their operating modes. For instance, smart speakers play or switch songs based on voice commands. Lastly, smart security devices utilize advanced analysis algorithms to determine if collected environmental data poses a threat. Notable products in this category include smart door locks and alarm systems. Furthermore, some smart security devices offer remote monitoring through cloud services for users who are away from home.

The space requirement for smart home systems is gradually expanding. In addition to controlling household appliances within the home, they can now be remotely monitored and controlled. This includes smart electric heaters and air conditioners. Cloud services have emerged to meet these demands by bridging the home's local network with the external internet. A substantial volume of data flows from the local network to the cloud server, and the external network accesses the cloud server to retrieve information.

Broadly speaking, smart home systems are not limited to individual family-based systems that solely focus on housing units. Instead, these systems can be extended to encompass buildings and neighborhoods, forming part of a larger smart city system. This expanded smart system has significant demands for data storage and processing. As a result, it can benefit from utilizing technologies like cloud storage and cloud computing to establish efficient and comfortable smart living systems.

This study investigates the functions and architecture of IoT-based smart homes and their application in education. Utilizing existing products and research findings, we developed a smart home system capable of gathering environmental data and offering users various features, including remote video surveillance, remote information retrieval, and remote control.

In summary, this study has the following contributions:

*1)* This study implemented multiple functions, including sensor information collection, video information collection, fire safety detection, remote monitoring, remote querying, and control.



*IoTAAI 2023, November 24–26, 2023, Nanchang, China*

ACM ISBN 979-8-4007-1648-5/23/11
https://doi.org/10.1145/3653081.3653194

*2)* The low-power development board PSoC 6 was utilized to construct sensor nodes, resulting in reduced power consumption of the system.

*3)* The smart home system developed in this study not only supports secondary development but can also be used for teaching in university laboratory courses.

This article is structured into several sections. Section II offers an overview of previous research conducted on smart home systems. In Section III, the architecture of the smart home system implemented in this study and the design principles of each component are introduced. The design results and discussion are presented in Section IV. Finally, Section V summarizes the findings of the study and provides future directions for further research.

## 2 LITERATURE SURVEY

Smart home systems that can be remotely monitored typically encompass features such as environmental detection, remote viewing, remote control, and security detection. Previous studies on smart home systems predominantly employ a development board as the central hardware, which is interconnected with diverse sensors and cameras through wired connections in order to gather environmental data. The development board also establishes a connection to the network, facilitating remote control. Certain systems additionally incorporate capabilities like object recognition, facial recognition, and alarm activation.

Thirrunavukkarasu R R et al. [3] proposed a customized control switch system that utilizes the Arduino Uno as the core control structure. The system captures environmental data through an infrared (IR) sensor and employs the ESP8266 as a Wi-Fi module to establish a connection to the Ubidots IoT platform for remote control. H.S. Sridhar et al. [4] implemented a multi-appliance control system that utilizes the ESP8266 as the core development board and connects it to the network through the Arduino IoT platform. This system facilitates remote control of multiple switches and motors. Nesreen Alsbou et al. [2] designed an automated home system for the care of elderly and disabled individuals. The system employs the Arduino MKR100 development board, integrated with WiFi, as its core. It utilizes various sensors including temperature sensors, light sensors, rotary angle sensors, PIR motion sensors, and humidity sensors to detect the household environment. It alerts the user through email and text messages. Shehzad Nur Tayus et al. [5] proposed an automated home monitoring system that utilizes the Raspberry Pi as the core development board, providing users with the capability to remotely access data and control household appliances. This system utilizes temperature and humidity sensors, as well as ultrasonic sensors, to measure room temperature, humidity, and detect visitors at the door. Additionally, it incorporates a camera to capture visitor images and utilizes deep learning algorithms for face recognition. In the event that an unrecognized face is detected, the system sends the image of the unknown person to the cloud via an MQTT broker, allowing customers to verify if it is an intruder.

The application of the Internet of Things encompasses both software and hardware components. There is a growing demand for well-rounded expertise in talent development. As a result, several universities have established dedicated research laboratories and introduced relevant courses to facilitate skills acquisition through a combination of theoretical instruction and hands-on experiments.

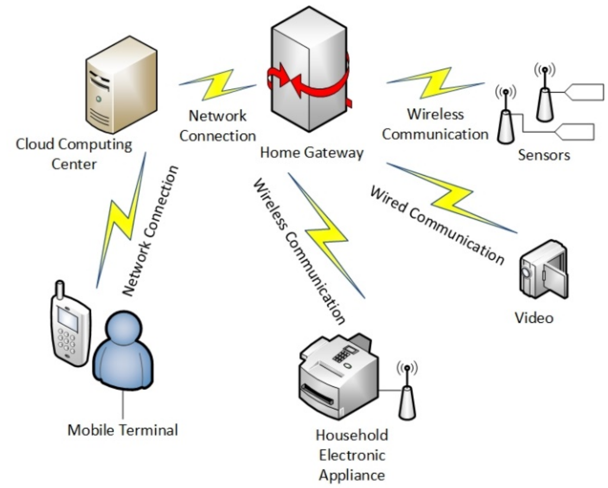


**Figure 1: System block diagram**

Juan Luo et al. [6] conducted a survey on the training programs offered by 31 Chinese universities for the Internet of Things (IoT) major. Their findings concluded that IoT teaching typically covers communication and electronics technology, sensor networks, embedded design, IoT and cloud computing technology, and software design. In a separate study, Yun Guo et al. [7] developed an IoT course based on the Niagara framework experimental platform, specifically in the context of industry-academia cooperation. Similarly, Mohamed El-Sharkawy et al. [8] focused on teaching a course that emphasized MEMS/NEMS and IoT applications; they also gathered feedback from students. Meenaxi M. Raikar et al. [9] developed a comprehensive IoT course that covered the foundational elements of IoT and taught related concepts in terminal devices, network layer, computing layer, and application layer, along with conducting experiments. Lastly, Lau Gim Seng et al. [10] summarized the evaluation results of various IoT course trials and recommended the inclusion of additional technologies such as network security, compatibility testing, and power consumption testing.

## 3 SYSTEM ARCHITECTURE AND IMPLEMENTATION

The research institute has implemented an IoT smart home system with specific objectives. Firstly, the system is designed to monitor the home environment and connect the detection module to the home gateway. Secondly, it allows users to remotely access information about the home environment and send control instructions to the home network. Previous research surveys have identified various modules that are essential for an intelligent home system. These include the home gateway module, sensor node module, video surveillance node module, cloud computing module, remote mobile terminal module, and home appliance control module. Figure 1 illustrates the system architecture.

The following section provides a detailed description of the construction process for key modules and the implementation of module interconnection.

## 3.1 Construction of Key Modules

The system comprises several essential modules, namely home gateways, sensor nodes, and video nodes. These modules must be constructed using hardware components.

*3.1.1 Hardware selection for the home gateway.* The home gateway serves as the central component of the entire system and is implemented using a programmable development board. The selection of the programmable development board necessitates consideration of processor performance, transmission methods, and programming protocols. Regarding processor performance, the development board should possess low power consumption and high integration capabilities. As for transmission methods, it should support wireless communication technologies like Bluetooth and WiFi to obtain information from sensor nodes, while also accommodating wired connections for capturing video streams from cameras. In terms of programming protocols, it should facilitate chip programming for data calculation and transmission, and provide support for multi-threading to enable simultaneous communication with other devices in the home network as well as cloud server. Taking all these factors into account, the Raspberry Pi 4 Model B development board is chosen as the hardware device for the home gateway. The Raspbian system is then installed on the Raspberry Pi 4 Model B and Python 3 is deployed.

*3.1.2 Design and construction of sensor nodes.* Sensor and programmable chips are used to create the sensor node module. The sensor directly detects variables, and the programmable chip converts the electrical signals into data that can be processed and analyzed. The sensor node module then transmits the data to the home gateway using its wireless module.

*Hardware selection for sensor node.* The sensor node's hardware core is carefully selected to be a low-power, I/O supporting development board that can accommodate multiple wired communication protocols. In particular, we have chosen the integrated Bluetooth PSoC 6 BLE development board. To measure the environmental data, we have opted for the BME280 sensor. The BME280 sensor operates at either 5V or 3.3V, with a typical current consumption of 2.8$\mu$A. It has the apability to measure temperature within the range of -40 to 85℃, with a resolution of 0.01℃. Additionally, it can measure humidity within the range of 0 to 100%RH, with a resolution of 0.008%RH, and pressure within the range of 300 to 1100 hPa, with a resolution of 0.18Pa [11]. Refer to Figure 2 for the visual representation of the BME280 sensor. Considering the simplicity and efficiency of the I2C protocol, which allows for multi-master and multi-slave communication with only two lines [12], our work utilizes this protocol to collect sensor data.

*Temperature and humidity acquisition.* The process of acquiring temperature and humidity data can be divided into two stages: data reading and writing, and data calculation. The specific register addresses and data information for the BME280 sensor can be found in Figure 3. Initially, the data is written to the registers at addresses 0xF2, 0xF4, and 0xF5. This is done to set the sampling rate to 16 times, enable continuous measurement, and set the standby time to 0.5ms. Subsequently, the chip ID is read from the register at address 0xD0. Next, the data is read from registers with addresses 0xFA to 0xFE. Each register stores eight bits of binary data. Registers at addresses 0xFA to 0xFC contain temperature-related data, while registers at addresses 0xFD and 0xFE contain humidity-related data. A combination of bit shifting and bitwise OR operations is used to merge the 24-bit temperature-related data and the 16-bit humidity-related data. Furthermore, the calibration data of the sensor is read from the registers at addresses 0x88 to 0x8D, 0xA1, and 0xE1 to 0xE7. The compensation coefficients dig_T1 to dig_T3 and dig_H1 to dig_H6 are calculated according to Table 1, and the temperature and humidity are then determined based on Figure 4 and Figure 5.

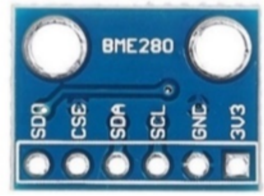


**Figure 2: BME280 sensor**

**Table 1: Compensation parameter of BME280 [11]**

| Register Address | Register content | Data type |
|---|---|---|
| 0x88/0x89 | dig_T1[7:0]/[15:8] | unsigned short |
| 0x8A/0x8B | dig_T2[7:0]/[15:8] | signed short |
| 0x8C/0x8D | dig_T3[7:0]/[15:8] | signed short |
| 0xA1 | dig_H1[7:0] | unsigned char |
| 0xE1/0xE2 | dig_H2[7:0]/[15:8] | signed short |
| 0xE3 | dig_H3[7:0] | unsigned char |
| 0xE4/0xE5[3:0] | dig_H4[11:4]/[3:0] | signed short |
| 0xE5[7:4]/0xE6 | dig_H5[3:0]/[11:4] | signed short |
| 0xE7 | dig_H6 | signed char |

*3.1.3 Construction of video nodes.* The video surveillance module performs several functions, such as capturing videos using a camera and transmitting the data to a home gateway for further image processing.

*Hardware selection and environment configuration.* The Raspberry Pi's Pi Camera is an expansion camera with 5 million pixels. It is capable of capturing static images at a maximum resolution of 2592×1944 pixels and can record 1080p videos at 30fps. The camera's appearance is depicted in Figure 6, where the white component represents the CSI cable. During the preparation phase, we install MJPG on the Raspberry Pi, set the port to 8080, and configure the OpenCV environment to display video capture results on a webpage as well as for graphics processing. The video is accessible by users at the address 192.168.1.106:8080/stream.html, as illustrated in the figure. Additionally, Figure 7 demonstrates that the webpage allows rotation and flipping of the video (indicated by the red box)

| Register Name | Address | bit7 | bit6 | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 | Reset state |
|---|---|---|---|---|---|---|---|---|---|---|
| hum_lsb | 0xFE | hum_lsb<7:0> | | | | | | | | 0x00 |
| hum_msb | 0xFD | hum_msb<7:0> | | | | | | | | 0x80 |
| temp_xlsb | 0xFC | temp_xlsb<7:4> | | | | 0 | 0 | 0 | 0 | 0x00 |
| temp_lsb | 0xFB | temp_lsb<7:0> | | | | | | | | 0x00 |
| temp_msb | 0xFA | temp_msb<7:0> | | | | | | | | 0x80 |
| press_xlsb | 0xF9 | press_xlsb<7:4> | | | | 0 | 0 | 0 | 0 | 0x00 |
| press_lsb | 0xF8 | press_lsb<7:0> | | | | | | | | 0x00 |
| press_msb | 0xF7 | press_msb<7:0> | | | | | | | | 0x80 |
| config | 0xF5 | t_sb[2:0] | | | filter[2:0] | | | | spi3w_en[0] | 0x00 |
| ctrl_meas | 0xF4 | osrs_t[2:0] | | | osrs_p[2:0] | | | mode[1:0] | | 0x00 |
| status | 0xF3 | | | | | measuring[0] | | | im_update[0] | 0x00 |
| ctrl_hum | 0xF2 | | | | | | osrs_h[2:0] | | | 0x00 |
| calib26..calib41 | 0xE1…0xF0 | calibration data | | | | | | | | individual |
| reset | 0xE0 | reset[7:0] | | | | | | | | 0x00 |
| id | 0xD0 | chip_id[7:0] | | | | | | | | 0x60 |
| calib00..calib25 | 0x88…0xA1 | calibration data | | | | | | | | individual |

| Registers: | Reserved registers | Calibration data | Control registers | Data registers | Status registers | Chip ID | Reset |
|---|---|---|---|---|---|---|---|
| Type: | do not change | read only | read / write | read only | read only | read only | write only |

Figure 3: Memory map of BME280 [11]

```
float Read_Temperture(void)
{
    int32_t var1, var2;
    int32_t adc_T = Sensor_Read24(BME280_ADDRESS,BME280_REGISTER_TEMPDATA);//0xFA
    adc_T >>= 4; //24bit to 20bit

    var1 = ((((adc_T>>3) - ((int32_t)_bme280_calib.dig_T1 <<1))) *
            ((int32_t)_bme280_calib.dig_T2)) >> 11;

    var2 = (((((adc_T>>4) - ((int32_t)_bme280_calib.dig_T1)) *
              ((adc_T>>4) - ((int32_t)_bme280_calib.dig_T1))) >> 12) *
            ((int32_t)_bme280_calib.dig_T3)) >> 14;
    t_fine = var1 + var2;
    float T = (t_fine * 5 + 128) >> 8;
    T = T / 100;
    return T;
}
```

Figure 4: Temperature compensation calculation

```
float Read_Humidity(void)
{
    int32_t adc_H = Sensor_Read16(BME280_ADDRESS,BME280_REGISTER_HUMIDDATA);//0xFD
    adc_H = Change_HighLow(adc_H);
    int32_t v_x1_u32r;

    v_x1_u32r = (t_fine - ((int32_t)76800));

    v_x1_u32r = (((((adc_H << 14) - (((int32_t)_bme280_calib.dig_H4) << 20) -
                    (((int32_t)_bme280_calib.dig_H5) * v_x1_u32r)) + ((int32_t)16384)) >> 15) *
                 (((((((v_x1_u32r * ((int32_t)_bme280_calib.dig_H6)) >> 10) *
                      (((v_x1_u32r * ((int32_t)_bme280_calib.dig_H3)) >> 11) + ((int32_t)32768))) >> 10) +
                    ((int32_t)2097152)) * ((int32_t)_bme280_calib.dig_H2) + 8192) >> 14));

    v_x1_u32r = (v_x1_u32r - (((((v_x1_u32r >> 15) * (v_x1_u32r >> 15)) >> 7) *
                               ((int32_t)_bme280_calib.dig_H1)) >> 4));

    v_x1_u32r = (v_x1_u32r < 0) ? 0 : v_x1_u32r;
    v_x1_u32r = (v_x1_u32r > 419430400) ? 419430400 : v_x1_u32r;
    float h = (v_x1_u32r>>12);
    h =  h / 1024.0;
    return h;
}
```

Figure 5: Humidity compensation calculation

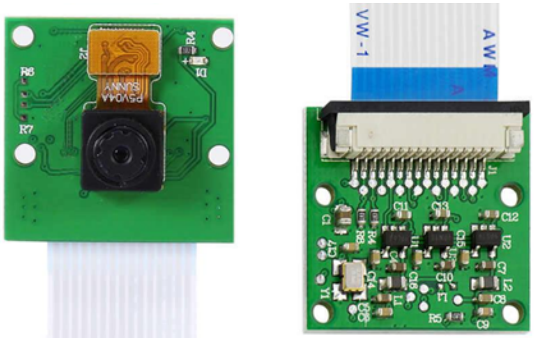

Figure 6: Pi Camera

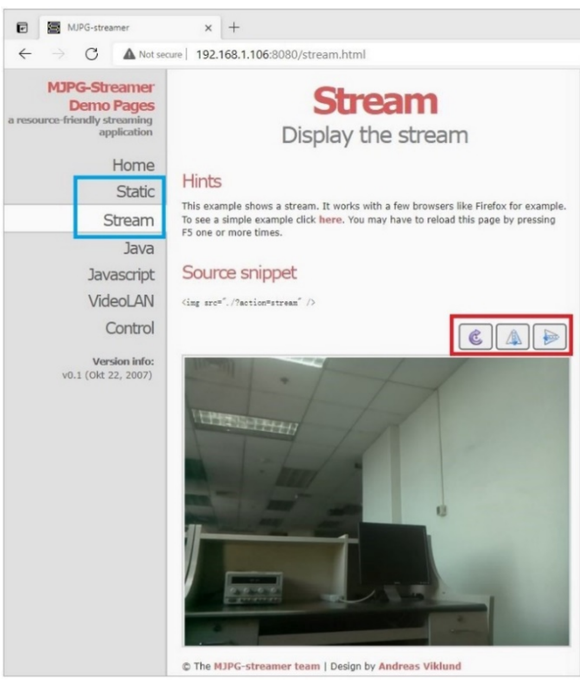


Figure 7: Get video with MJPG-streamer

and the option to switch between static images and dynamic videos (indicated by the blue box).

*Flame detection algorithm.* Flame detection can be achieved by processing the collected video using OpenCV. In general, flames have a relatively large red component and green component in terms of color, with the green component being larger than the blue component. Additionally, the saturation component also exhibits certain characteristics. To determine flame presence, we extract the RGB channels of the image and compute the saturation value of the S channel. Chen et al. [13] proposed decision rules for flame detection, as shown in (1), where $(x, y)$ represents the pixel position coordinate, $R_{mean}$ is the average value of the red component throughout the entire image, and $S_t$ and $R_t$ denote the saturation threshold and red component threshold, respectively.

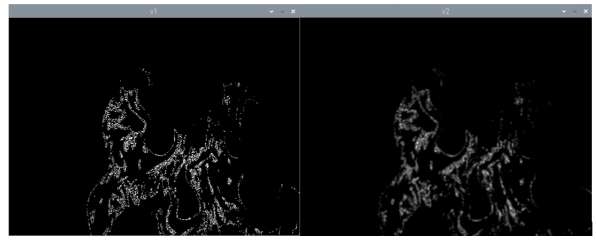

Figure 8: The mask of fire before (left) and after (right) filtering

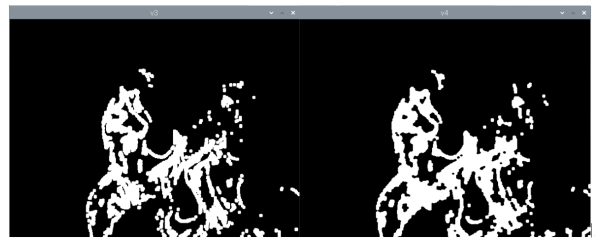

Figure 9: The mask of fire before (left) and after (right) the closing operation

For our experiments, we have chosen $S_t = 55$, $R_t = 150$. By applying this formula, we can initially determine the location of the flame, resulting in a flame mask. Subsequently, morphological processing is utilized on the preliminary flame mask. Gaussian filtering is applied to achieve smoothing, as depicted in Figure 8. After this step, the granularity is reduced and the overall image becomes gray. Following an increase in brightness, a closing operation is conducted to eliminate black dots in the mask and achieve further smoothing. Regions of interest with stronger overall coherence are represented by the white portions in Figure 9. Based on the mask, a closed curve is outlined as the flame and marked on the original image. The flame is approximately outlined according to its range, as shown in Figure 10. After the flame detection process, the temperature information from sensors is combined to confirm the presence of fire.

$$\begin{cases} S > 0.2 \\ R(x,y) > G(x,y) > B(x,y) \\ R(x,y) > R_{mean} \\ S > \frac{(255-R)\times S_t}{R_t} \end{cases} \tag{1}$$

## 3.2 Interconnection of Modules

This study employs wireless communication and network technology to establish connectivity between modules. The transmission of data between the home gateway and sensor nodes is achieved using the Bluetooth protocol. Meanwhile, the transmission of data between the home gateway and the cloud server, as well as between the cloud server and the remote mobile terminal, is accomplished through network protocols. Additionally, the remote mobile terminal's access to video utilizes intranet penetration technology.

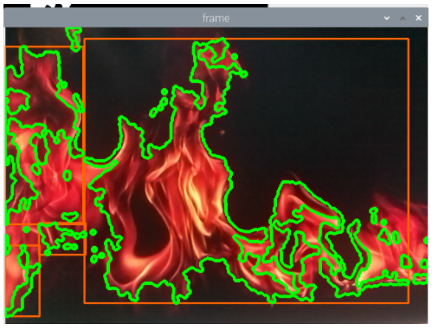


**Figure 10: Flame detection results**

*3.2.1 Home gateway collects data from the sensor nodes.* Environmental data from sensor nodes and home gateways is transmitted using the Bluetooth protocol. Due to the limitations of the Bluetooth protocol, the collected data for household environment is separated into an integer part and a decimal part. In the case of temperature and humidity measurements, four integers are transmitted after each measurement. To distinguish between adjacent sets of data, a flag bit is added to each set of four integers. Since the measurement result includes two decimal places and the integer part is always below 100, the hundreds digit can be utilized as the flag bit. Once a complete set of data is received, the home gateway converts it to retrieve the original temperature and humidity values.

*3.2.2 Data Transmission between home gateway and cloud server.* Various services are provided by existing cloud computing platforms. In this research, suitable cloud server are selected for network communication.

*Cloud server selection.* Cloud computing can be categorized into three services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). These services have a hierarchical relationship, with various products in the market offering all three. IaaS, the lowest level service, provides resources such as processing, storage, and networking. PaaS, built on top of IaaS, facilitates the development and hosting of web applications. SaaS, the highest level service, offers easily accessible client services [14]. Users can customize application software services according to their specific requirements. In this study, we opt for an Aliyun Cloud server with the PaaS option and select the Linux operating system to utilize Python for data processing.

*Implementation of network communication.* The implementation of network communication involves two parties: the home gateway and the cloud server. The home gateway acts as the client, responsible for uploading data to the cloud server for external queries. To establish a connection, the home gateway creates a socket and connects to the public IP address and open port of the cloud server. Before transmission, the data is encoded in UTF-8 and then decoded upon reception. Each data packet includes a Raspberry Pi identifier and temperature/humidity readings. The identifier and readings are separated by a hyphen, while the temperature and humidity data are separated by a plus sign. On the other hand, the cloud server acts as the server and creates a socket to bind to the internal IP address and port. Enhanced security settings are implemented.


**Figure 11: Socket transfer send and receive results**

The cloud server allows for monitoring multiple devices and parses the received data packets. The identifier at the beginning of the data is used to identify the source. If the data is from the Raspberry Pi, it is stored in the database along with the current time. If the data is from an external device, the queried data is sent back to the device. For data storage, a MySQL database is utilized, and the pymysql package is used to establish a connection and insert data using the INSERT INTO command. To ensure timely updates, the home gateway simultaneously sends and receives data from the sensor nodes. This is achieved by initiating two threads on the home gateway side, each responsible for receiving and sending data. The received and sent data packets are stored as global variables. Additionally, the sending thread updates the data at regular intervals. To ensure timely updates, the home gateway simultaneously sends and receives data from the sensor nodes. This is achieved by initiating two threads on the home gateway side, each responsible for receiving and sending data. The received and sent data packets are stored as global variables. Additionally, the sending thread updates the data at regular intervals. Figure 11 visually represents the data received and forwarded time at the Raspberry Pi end on the left side, while the data received and time of reception at the cloud server end is displayed on the right side. The red box indicates that the data is sourced from the Raspberry Pi. The time delay during the transmission process is minimal, being less than one second and can be considered negligible. Figure 12 provides a visual representation of the database storage results, with each column representing the time, temperature, and humidity readings, respectively.

*3.2.3 Data transmission between cloud server and mobile terminal.* Communication between mobile terminal and cloud server allows users to remotely query and control their home environment using mobile devices. The mobile terminal serves as the client in this communication, utilizing a public IP address and port to establish a socket connection and create an output stream for data transmission. When querying information, the output message is labeled as "android-Searching," whereas for home control, the output message originates from the editable text box with an added "android-" tag for differentiation. On the server-side, the cloud server functions as the server, receiving messages from the mobile terminal and determining whether they are queries or controls. If a query, the server

| time | temperature | humidity |
|---|---|---|
| 2021-05-25 11:42:30 | 31.35 | 31.71 |
| 2021-05-25 11:42:40 | 31.20 | 31.68 |
| 2021-05-25 11:42:45 | 31.23 | 32.49 |
| 2021-05-25 11:42:54 | 31.23 | 31.71 |
| 2021-05-25 11:42:59 | 31.23 | 31.59 |
| 2021-05-25 11:43:08 | 31.24 | 32.16 |
| 2021-05-25 11:43:16 | 31.24 | 32.53 |
| 2021-05-25 11:43:22 | 31.24 | 32.57 |
| 2021-05-25 11:43:32 | 31.24 | 32.55 |
| 2021-05-25 11:43:42 | 31.24 | 32.55 |
| 2021-05-25 11:43:52 | 31.24 | 32.29 |
| 2021-05-25 11:43:59 | 31.24 | 32.51 |
| 2021-05-25 11:44:09 | 31.23 | 32.90 |

**Figure 12: Database storage result**

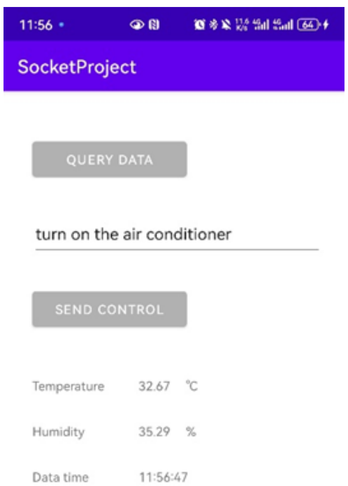


**Figure 13: Mobile app for query and control**

retrieves the latest environmental information from the database table and returns it to the mobile terminal. If a control, it sends instructions to the home gateway instead. The mobile application is developed using Android Studio, incorporating the control home button and editable text box at the top of the user interface for sending control commands. The query data button at the bottom is utilized for sending query messages to the cloud server, and the resulting information which includes temperature, humidity, and detection time is displayed at the bottom of the program page, as shown in Figure 13.

*3.2.4 Viewing home videos from remote mobile terminal.* Video streaming is a form of big data, making it inconvenient for large-scale transmission. To address this issue, intranet penetration is utilized to enable external devices to view home videos in real-time. The reverse proxy software FRP facilitates the provision of services to external devices located behind the intranet or firewall, as depicted in Figure 14. A cloud server with a fixed IP serves as a gateway to allow external device access to the home network port. Installation and configuration of the FRP application are necessary for both the Raspberry Pi and the cloud server. The Raspberry Pi

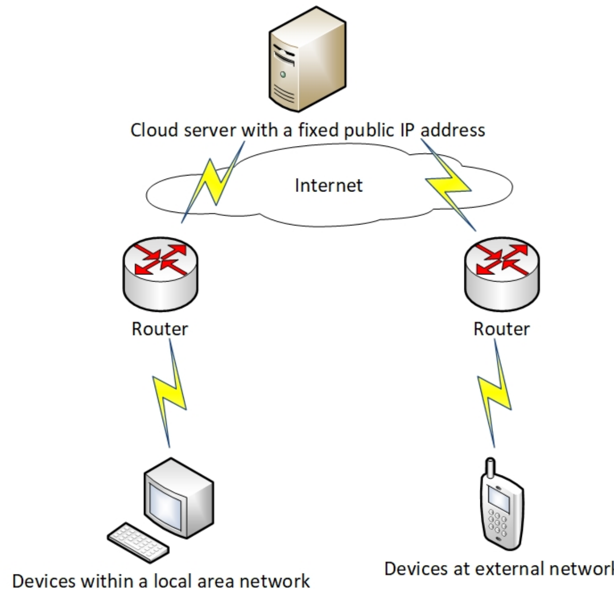


**Figure 14: FRP intranet penetration diagram**

functions as the client and requires configuration of the frpc.ini file to set the FRP and local website port numbers. The cloud server, on the other hand, acts as the server and necessitates configuration of the frps.ini file to set the same FRP and local website port numbers, as well as the public IP of the server. Once the aforementioned configuration steps are completed, the local website accessed by the Raspberry Pi IP is mapped to the publicly accessible website accessed through the cloud server's public IP. To ensure automatic startup, the service can be initiated at startup by adding the configured script to the startup unit through the creation of a service file. It is important to note that the FRP application should be run after the basic network application has started. Figure 15 displays the outcome of viewing the video stream, where the red box indicates the utilization of the 4G network provided by the mobile network operator as the external network of the mobile device, and the blue box represents the entry of the cloud server's public IP in the browser.

## 4 RESULT AND DISCUSSION

The results of the smart home system implemented in this study are presented in Figure 16. The primary functions of the system encompass data collection, data transmission, data processing, and remote access. The data collection phase involves the utilization of sensors and cameras to detect and provide users with prompt and accurate information about the home environment. Data transmission is subdivided into two categories: propagation within the home network and transmission to the external network. Propagation within the home network integrates wired and wireless transmission methods to consolidate the gathered information at the home gateway. On the other hand, transmission to the external network involves the home gateway sending the data to the cloud

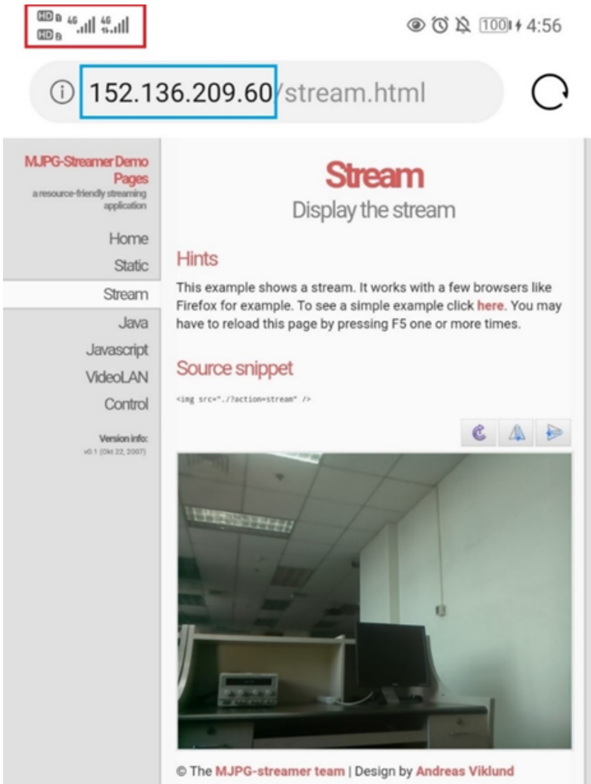


**Figure 15: Accessing video surveillance from external network**

server through network protocols, thereby reducing local memory consumption. Data processing entails analyzing video streams using color analysis and morphological processing techniques, in conjunction with temperature and humidity data, to determine the occurrence of a fire. Lastly, remote access involves connecting an external mobile terminal to the cloud server using network protocols or employing network penetration technology to establish an external network access interface, enabling the acquisition of home information and the transmission of control instructions.

This study presents a collection of electronic systems designed for the purpose of teaching students about the various technologies involved in intelligent home systems, including software and hardware technologies, data processing technologies, and network communication technologies. These systems are focused on enabling students to gain a comprehensive understanding of the detection, control, and security functions employed in intelligent home systems. Additionally, this system facilitates further software and hardware development, allowing for the possibility of changing the communication protocol to different protocols. In terms of hardware, the use of FPGA can be utilized to carry out specific functions. The Verilog language can be employed on the FPGA development board to implement communication protocols like I2C, serving as a replacement for the sensor node development board. Furthermore, the FPGA development board can also support the implementation of a CPU based on the RISC-V instruction set [15] to handle the

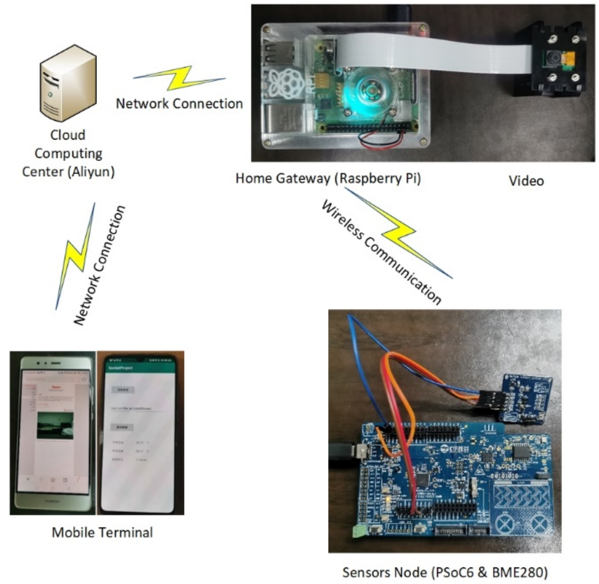


**Figure 16: Results of the smart home system**

computational aspect of video flame detection. Additional sensors, such as human infrared sensors, smoke sensors, and GPRS sensors, can also be incorporated into the hardware to detect environmental data from various dimensions. Data processing capabilities can be enhanced through the utilization of algorithms for video stream processing, such as neural networks for facial recognition. Lastly, the development of the mobile terminal app can be continued.

## 5 CONCLUSION AND FUTURE WORK

This study developed a smart home system for experimental teaching and development. The system achieved various functionalities including home environment collection, home security detection, home and external network interconnection, and remote monitoring. In terms of home environment collection, the system collected temperature, humidity, and video information. The temperature and humidity data were collected by a sensor node consisting of a BME280 sensor and a PSoC 6 BLE development board. Video information was captured through a camera connected to the Raspberry Pi CSI interface and displayed on a web page. Home security was addressed by detecting flames in video streams using color space and morphological operations. To aggregate and store home information, a home gateway and a cloud server were constructed using Bluetooth and TCP protocols. The home gateway operated with a dual-thread for receiving and sending data, and the information was stored in a cloud database with timestamps. For remote monitoring, an Android application was developed, allowing users to view home information and send control commands through the TCP protocol. This improved the system's interactivity and convenience. Additionally, the system used FRP intranet penetration technology to provide external network video access by mapping the home intranet web page to the public network of the cloud service.

This study aimed to investigate smart home systems and develop a smart home information detection system using Internet of Things (IoT) technology. The developed system effectively enables users to remotely monitor their home environment and holds practical significance. Additionally, it has potential applications in experimental teaching.

This study offers opportunities for improvement and further development. Future endeavors should aim to enhance the hardware structure, data processing, and software development. Regarding hardware, incorporating additional sensors would yield more extensive home information. Moreover, replacing the existing development board with a programmed core processor on an FPGA would be advantageous. Additionally, integrating servo motor control would facilitate home appliance manipulation. Concerning data processing, extracting keyframes from the video stream and implementing neural networks for object and face recognition should be considered. In terms of software development, optimizing the user interface of the mobile app is essential.

## REFERENCES


[1] P. Negi, V. Pachouri, S. Pandey, S. Chattopadhyay, A. Gehlot and K. Joshi, "Digitalized Infrastructure of Smart Homes Using Iot and Cloud Computing," 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), Greater Noida, India, 2023, pp. 510-514, doi: 10.1109/CISES58720.2023.10183553.

[2] N. Alsbou, N. M. Thirunilath and I. Ali, "Smart Home Automation IoT System for Disabled and Elderly," 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 2022, pp. 1-5, doi: 10.1109/IEMTRONICS55184.2022.9795738.

[3] R. R. Thirrunavukkarasu, S. Mohan Kumar, P. Praveen, T. Meera Devi, S. Pradeep and S. Ganesh Prabu, "Customization In Home Automation Using IoT," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 1062-1067, doi: 10.1109/ICACCS51430.2021.9441979.

[4] H. S. Sridhar, S. Somvanshi, M. Ramsinghani and N. H. C., "Designing of Smart Home Automation Using IoT," 2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT), Mandya, India, 2022, pp. 1-6, doi: 10.1109/ICERECT56837.2022.10059802.

[5] S. N. Tayus, A. K. M. Kamrul Alam Kakon and M. Ullah, "IoT Based Web Controlled Multiple Home Automation and Monitoring with Raspberry Pi," 2022 3rd International Conference for Emerging Technology (INCET), Belgaum, India, 2022, pp. 1-4, doi: 10.1109/INCET54531.2022.9825087.

[6] J. Luo, C. Wang, Y. Zheng, H. Zhao, Z. Li and D. Xiao, "Exploration and Consideration on the Course System Construction of IoT Specialty," 2018 13th International Conference on Computer Science & Education (ICCSE), Colombo, Sri Lanka, 2018, pp. 1-5, doi: 10.1109/ICCSE.2018.8468695.

[7] Y. Guo and M. Li, "IoT Course Construction in General Education against the Background of China's University-Industry Collaboration," 2022 4th International Conference on Computer Science and Technologies in Education (CSTE), Xi'an, China, 2022, pp. 128-132, doi: 10.1109/CSTE55932.2022.00030.

[8] M. El-Sharkawy, S. Shayesteh and M. Rizkalla, "Integrating NEMS/MEMS with IoT applications into an innovative ECE senior elective course," 2017 IEEE Frontiers in Education Conference (FIE), Indianapolis, IN, USA, 2017, pp. 1-5, doi: 10.1109/FIE.2017.8190447.

[9] M. M. Raikar, P. Desai and J. G. Naragund, "Active Learning Explored in Open Elective Course: Internet of Things (IoT)," 2016 IEEE Eighth International Conference on Technology for Education (T4E), Mumbai, India, 2016, pp. 15-18, doi: 10.1109/T4E.2016.012.

[10] L. G. Seng, K. L. K. Wei and S. J. Narciso, "Effective Industry Ready IoT Applied Courseware - Teaching IoT Design AND Validation," 2020 IEEE Global Engineering Education Conference (EDUCON), Porto, Portugal, 2020, pp. 1579-1583, doi: 10.1109/EDUCON45650.2020.9125366.

[11] Bosch Sensortec, "BME280 Combined humidity and pressure sensor" BME280 datasheet, Jan. 2022, Available: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bme280-ds002.pdf.

[12] J. Chen and S. Huang, "Analysis and Comparison of UART, SPI and I2C," 2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 2023, pp. 272-276, doi: 10.1109/EEBDA56825.2023.10090677.

[13] Thou-Ho Chen, Ping-Hsueh Wu and Yung-Chuen Chiou, "An early fire-detection method based on image processing," 2004 International Conference on Image Processing, 2004. ICIP '04., Singapore, 2004, pp. 1707-1710 Vol. 3, doi: 10.1109/ICIP.2004.1421401.

[14] M. Saraswat and R. C. Tripathi, "Cloud Computing: Analysis of Top 5 CSPs in SaaS, PaaS and IaaS Platforms," 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART), Moradabad, India, 2020, pp. 300-305, doi: 10.1109/SMART50582.2020.9337157.

[15] A. Waterman, Y. Lee, D. Patterson, and K. Asanović, "The RISC-V Instruction Set Manual, Volume I: User- Level ISA, Version 2.1," 2016. Available: https://digitalassets.lib.berkeley.edu/techreports/ucb/text/EECS-2016-118.pdf.