International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 4, July-Aug 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Real-Time Monitoring Application for Smart Home Automation of Household Appliances

Soulith SENGMANOTHAM *, Amone Chanthaphavong**, Phonesouda SOUPHAMITH***
*(Department of Computer Science, National University of Laos, Vientiane Capital, Lao P. D. R
Email : s.sengmanotham@nuol.edu.la)
** (Department of Computer Science, National University of Laos, Vientiane Capital, Lao P. D. R
Email : a.chanthaphavong@nuol.edu.la)
*** (Department of Computer Science, National University of Laos, Vientiane Capital, Lao P. D. R
Email : phone.souphamith@gmail.com)

--************************----------------------------------

Abstract:

The internet of Thing (IoT), following the rise of internet technology, marks a transformative wave in the information industry, profoundly impacting political, military, economic, and environmental domains. As an integral part of the next generation of information technology, IoT connects physical objects through the internet using technologies such as RFID, sensors, GPS and scanners, enabling intelligent identification, monitoring, tracking, and management of objects. A core feature of IoT is the deployment of numerous real-time sensors that gather and transmit environmental data, facilitating seamless communication between people, objects, and networks. This real-time connectivity supports efficient control and management across various sectors. This paper explores the development of smart home systems based on IoT and Internet technologies. A smart home integrates home security, network services, and automation systems using advanced computing, communication, and wiring technologies. It creates a safe, comfortable, and intelligent living environment by connecting appliances and security systems for centralized or remote control and management. As consumer expectations for convenience and quality of life continue to rise, smart homes are evolving into multifunctional platforms offering services such as video intercoms, remote appliance monitoring, telemedicine, online education, and home entertainment. Although the current market share remains modest compared to traditional housing sectors, the increasing affordability and technological advancement of smart home systems promise substantial future growth and widespread adoption.

Keywords **— Internet of Thing (IoT), Smart Home, Intelligent Systems, Sensors, Remote Control.**

--************************----------------------------------

I. INTRODUCTION

The Internet of Thing (IoT) has emerged as a transformative force in modern information and communication technology, enabling seamless interconnectivity among devices, systems, and people. IoT extends the capabilities of the traditional Internet by incorporating everyday objects such as sensors, mobile terminals, smart home devices, surveillance systems, and industrial equipment into a unified, intelligent ecosystem [1]. Through advanced technologies including radio frequency identification (RFID), wireless sensor networks, global positioning systems (GPS), and laser scanning, IoT facilitates real-time data exchange, remote monitoring, and automated control over a wide range of applications [2]. A key feature of the IoT is its ability to integrate physical infrastructure with digital intelligence, bridging the divide between conventional construction elements—such as buildings, roads, and transportation systems—and IT components like data centres, broadband networks, and smart terminals [3]. In this interconnected environment, devices can autonomously "communicate" with each other without human intervention, creating a more efficient, responsive, and intelligent world. Among its many applications, smart home systems stand out as one of the most impactful and accessible implementations of IoT in daily life. A smart home refers to a residential system that incorporates home automation, networked communication, and intelligent control technologies to enhance safety, convenience, energy efficiency, and overall quality of life. This includes the centralized or remote management of lighting, climate control, home appliances, surveillance systems, and entertainment platforms via mobile or web-based interfaces [4]. For instance, intelligent curtains may close automatically based on sunlight intensity, or air conditioners can adjust temperature and humidity levels without user input. These capabilities are achieved by connecting household devices to an intelligent platform that monitors and controls them in real time. The concept of smart homes originated in the United States in the 1970s and quickly spread to Europe, Japan, and other technologically advanced regions [5]. In China and Southeast Asia, adoption began more gradually in the late 1990s but has since gained momentum. Today, many modern communities are being developed with embedded smart home systems. Despite significant progress, smart home technology still occupies a relatively small share of the broader building and housing industry. However, rapid advancements in wireless communication, cloud computing, and sensor miniaturization are driving down implementation costs, making smart home solutions increasingly accessible to a wider population [6]. With growing consumer demand for personalized, intelligent, and secure living environments, smart homes are evolving into multifunctional platforms that offer visual intercoms, telemedicine, remote diagnostics, energy optimization, online education, and entertainment systems. These systems aim to not only improve quality of life but also reduce energy consumption, enhance home security, and support independent living for the elderly and disabled [7]. This paper explores the development and application of smart home systems based on IoT technologies. It examines the architecture, communication protocols, system components, and integration methods that enable real-time automation and monitoring of home environments. The study also discusses key challenges such as interoperability, data security, scalability, and cost, and presents current trends and future prospects for the smart home industry.

II. THE PROPOSED AND IMPLEMENTATION SYSTEM

The smart home system comprises several hardware modules such as a GSM remote control, temperature and smoke sensors, a voice-prompt password lock, and a child safety module. Each module is treated as a device in embedded development and requires a Linux kernel device driver for operation. A device driver is kernel-level software that facilitates communication between the hardware and user applications. It handles data transfer, parameter configuration, and interrupt management. Applications use system calls (e.g., open(), read(), write(), ioctl()) to interact with these drivers via the Linux kernel. Linux drivers are typically implemented as kernel modules, which can be loaded or unloaded at runtime. This modularity improves flexibility, debugging, and system maintainability. Drivers are categorized into character devices, block devices, and network interfaces. In this system, only character device drivers are used, which allow sequential byte-by-byte data access (not random access).

A. Implementation of gsm remote control module

In the actual programming process, it mainly consists of two components, one is the hardware driver of the serial port, and the other is the application on the file system. The Mini2440 development board uses the S3C2440 processor, which includes three UART serial ports (UART0, UART1, UART2). UART0 is RS232 level and is reserved for booting via NFS, UART1 and UART2 are TTL level. In this project, UART2 is used to communication with the GSM module via a level-conversion circuit (TTL to RS232). The Linux kernel provided for the board already includes a serial port driver, so no additional driver programming is needed. Accessing the serial port in Linux, serial ports are treated as files (e.g., /dev/ttySAC1). The serial port is accessed using the open() system call: int fd = open("/dev/ttySAC1", O_RDWR); If the port cannot be opened, an error is returned. Seral Port Configuration to use the serial port properly, parameters like baud rate, parity, data bits, and stop bits must be configured using the struct termios structure: struct termios {

```
    unsigned short c_iflag;  // input mode flags
    unsigned short c_oflag;  // output mode flags
    unsigned short c_cflag;  // control mode flags
    unsigned short c_lflag;  // local mode flags
    unsigned char  c_line;   // line discipline
    unsigned char  c_cc[NCC]; // control characters
  };
```

The following functions are used to get and set terminal attributes:

- tcgetattr(fd, &termios_struct) – Get current serial settings
- Modify desired settings in termios_struct
- tcsetattr(fd, TCSANOW, &termios_struct) – Apply new settings
  in this system, the serial port is configured as follows:
- Baud rate: 9600 bps
- Parity: None
- Data bits: 8
- Stop bits: 1

Data Transmission, once configured the serial port can be used like a file:

- Send data: write(fd, buffer, length);
- Receive data: read(fd, buffer, length);

B. Sending SMS via Serial Port

The Mini2440 smart home system uses UART2 and a MAX232 circuit to communicate with the TC35 GSM module, sending AT commands over a serial port to transmit English SMS messages for remote appliance monitoring and control. The TC35, which operates at RS232 level, requires level conversion from the board's TTL output. SMS messages are sent by configuring the module with AT commands, specifying the format and recipient, and sending the message content followed by Ctrl+Z. The system only processes messages from authorized phone numbers, identifying the sender and content before executing appliance commands. Messages from unknown users are ignored for security.

C. Implementation of temperature acquisition module

DS18B20 temperature sensor and its application in Multi-Point temperature measurement, the DS18B20 is a digital temperature sensor that provides 9- to 12-bit binary temperature readings. It communicates with the host CPU via a single-wire interface, requiring only one common data line. There are two power supply modes for the DS18B20. Parasitic Power Mode – In this mode, the sensor draws power directly from the data line, reducing wiring requirements. However, it results in slower temperature conversion due to limited power supply current. External Power Supply Mode – This design uses an external power supply connected to the VDD pin. In this mode, the I/O line does not need a strong pull-up resistor, and the sensor receives sufficient current for faster and more accurate temperature conversion.

The DS18B20 supports a wide temperature range of -55°C to +125°C with a resolution of 0.5°C, and typically completes a conversion in 1 second. Each sensor is factory-programmed with a unique 64-bit serial number stored in its internal ROM, enabling multiple DS18B20 devices to coexist on the same data bus for multi-point temperature monitoring. To perform multi-channel measurements, the system uses the ROM search command to identify all DS18B20 devices on the bus. After identification, it sends the Skip ROM command followed by the temperature conversion command to trigger all sensors. Once the conversion is complete, the system can use the Match ROM and Read Scratchpad commands to access the temperature data from each individual sensor.

The necessary ROM and RAM operation commands for these procedures are summarized in Table 1 and Table 2, respectively.

TABLE I
ROM MEMORY OPERATION COMMANDS

Rom instruction	Code	Command function
Read ROM	33H	Read the 64-bit device serial number of ds18b20
Match	55H	Bus control locates a particular ds18b20 on a single line
Skip ROM	CCH	The bus controller can operate on the memory without sending the

		device serial number
Search ROM	FOH	Identify the serial number of each single-wire device on a single-wire bus
Alarm ROM	ECH	Warning search response for devices whose temperature exceeds the upper and lower limits of the alarm

TABLE III
RAM MEMORY OPERATION COMMANDS

Ram command	Code	Command function
Write Scratchpad	4EH	Write data to memory bytes 2, 3 and 4
Read Scratchpad	BEH	Read scratch memory
Copy Scratchpad	48H	Complex the contents of th, tl and configuration registers in the scratchpad to e2prom
Convert Temperature	44H	Start ds18b20 temperature conversion
Recall EPROM	B8H	Write the contents of th, tl and configuration registers in e2prom back to the scratchpad
Read Power supply	B4H	Read power

The DS18B20 digital temperature sensor converts analogy temperature into a digital signal and transmits it via a one-wire serial interface. Accurate temperature acquisition relies heavily on the communication protocol between the DS18B20 and the S3C2440 processor. This protocol includes device initialization, ROM commands, and memory operations. Communication begins with an initialization sequence involving a reset pulse and presence detection. Data transmission follows a strict timing-based protocol, where all commands and data are transferred byte by byte, least significant bit first. To write data, the processor pulls the DQ line low. A write "0" keeps the line low for the full duration (60–120 µs), while a write "1" pulls the line low for only <15 µs before releasing it. At least 1 µs recovery time is required between write operations. This precise timing ensures reliable data transmission between the sensor and the processor.

D. Implementation of the smoke sensor module

The MQ-2 gas sensor is capable of detecting various flammable gases. When such gases are present in the surrounding environment, the sensor's conductivity increases in proportion to the gas concentration. This change in conductivity is converted into a corresponding analoy voltage signal through a simple circuit. The higher the output voltage, the higher the gas concentration. In this system, the S3C2440 processor is used to read the MQ-2 sensor's output. The S3C2440 includes a built-in A/D (analog-to-digital) converter, which digitizes the sensor's voltage signal for further processing. In the embedded Linux environment, the A/D converter is accessed through the device file: /dev/adc.

To read gas levels:

- The system first opens the ADC device using: int fd_adctest = open("/dev/adc", O_RDWR); If the open operation fails, the function returns -1.
- After the ADC device is successfully opened, further operations (e.g., reading the voltage) can be performed using fd_adctest.

To stop using the gas sensor, the system calls: close(fd_adctest); figures, as shown in Fig. 1.

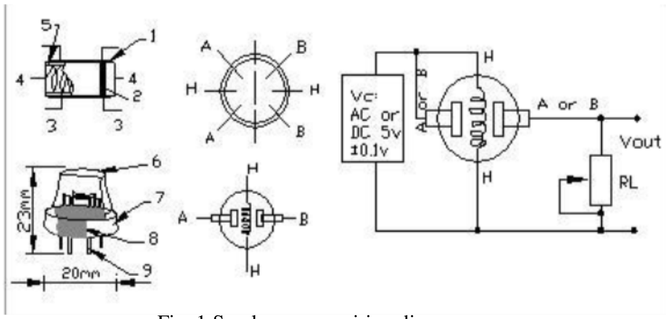

Fig. 1 Smoke sensor wiring diagram

E. Implementation of password lock anti-theft module with voice prompt function.

The system continuously scans the keypad to detect if a key is pressed. When a button press is detected, a debounce delay is applied to filter out noise, after which the system determines which key was pressed and calculates its corresponding value. A predefined password is initialized in the program. The system constantly monitors user input, and if the correct password is entered, the door is unlocked. At this point, the system disables the external interrupt signal triggered by the infrared (red light) sensor. If the door is opened by unauthorized means (e.g., forced entry), the indoor photoelectric sensor detects an abnormal signal. This triggers an interrupt service routine (ISR), which sends a signal to activate the ISD1760 voice alarm chip. The password lock module begins by initializing external interrupts. The program constantly checks whether the infrared sensor has detected a signal and whether an interrupt has occurred. If an interrupt is detected:

- It first checks if the correct password has been entered.
- If correct, the door opens.
- If incorrect, the voice alarm module is triggered?

The photoelectric sensor is used to detect changes near important objects (e.g., safes). It outputs a high signal by default, and when someone approaches, it switches to a low signal. The system uses this falling-edge transition as an interrupt trigger. In the embedded Linux system, the device file /dev/red_sensor is used. The program opens the device using open("/dev/red_sensor", O_RDWR) and reads the state. If the value is 1, it indicates an interrupt, and a signal is sent to trigger the voice alarm. In the Qt-based application, a timer checks the state of the photoelectric sensor every second. If a falling-edge signal is detected, it triggers the alarm module. The voice alarm system uses the ISD1760 chip, which is a high-quality voice recording/playback IC. It supports multiple features like:

- Independent or embedded control mode
- Voice alert (vAlert)
- Customizable playback signals

This module ensures real-time password verification, intrusion detection, and voice alarm feedback, enhancing both security and user interaction in the smart home system. This module uses a 4*4 keyboard as the input device. The physical map and circuit diagrams used in the system are shown in Fig 2.

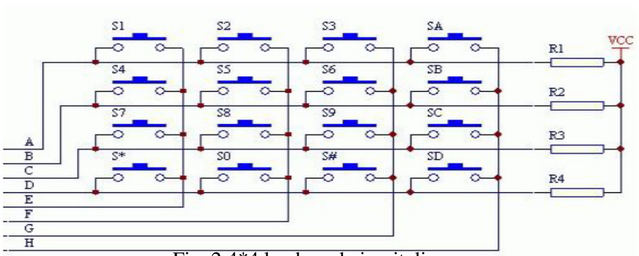

Fig. 2 4*4 keyboard circuit diagram

F. Preventing children falling from the building module

To address the growing concern of children accidentally falling from windows, this smart home module introduces an automated safety mechanism that uses a human infrared sensor and a stepper motor to monitor and control window access. A passive infrared (PIR) sensor is installed near the window to detect the presence of a person. When someone—presumably a child—approaches the window, the system evaluates the current window status. If the window is open, the system automatically activates a stepper motor to close it. Once the individual moves away from the window, the system reopens it, ensuring both safety and usability. In the prototype, a single infrared sensor is used, which cannot differentiate between children and adults. However, in practical implementations, multiple infrared sensors can be installed at various heights to distinguish the height profile of the approaching person, allowing the system to more accurately identify children. The PIR sensor outputs a high-level voltage (~2.3V) when idle and 0V when motion is detected. A 74LS04 inverter chip is integrated to reverse the logic for processing ease. This logic inversion helps the controller easily identify detection states and respond accordingly. A 28BYJ-48 stepper motor is employed to physically control the window. This motor operates using a four-phase, eight-beat stepping sequence and responds to electrical pulse signals to produce accurate angular displacement. The motor is powered by a DC 5V or 12V supply and enables precise control of window position based on pulse count. The system runs continuously, polling the sensor via a timer to check for motion. When a person is detected near the window, it automatically initiates a window-close command and continues to monitor the area. Once the area is clear, the system reopens the window after a brief delay. This module provides a real-time, automated safety solution to prevent child-related window accidents, enhancing home security through responsive sensing and precise motion control. Show in Fig 3

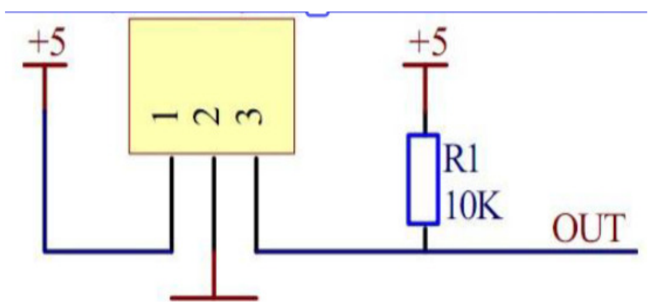

Fig. 3 Infrared sensor pin diagram wiring diagram

III. RESULTS AND DISCUSSION

The proposed smart home system was tested across various modules, including the password lock, infrared detection, photoelectric sensors, stepper motor control, and voice alarm functions. The system successfully responded to user inputs, sensor activations, and interrupt signals in real-time. During testing, the password authentication module reliably opened the door upon correct input and triggered the alarm when an incorrect password was entered during intrusion scenarios. The stepper motor also performed consistently in opening and closing the window based on human proximity detection.

A. Remote monitoring of home app lance module function test

When the system is turned on, the desk lamp and fan are turned off, as show in Fig 4. At this point we send the "tdk" command on the phone to turn on the desk lamp. The desk lamp and the fan are in working state. The following sends a "qjk" command to the system to monitor the working state of the appliance

Fig. 4 Smart home system program interface

B. Password lock anti-theft module function test

Next, this the password locks anti-theft module with voice prompt function, first enter the password, as show in Fig 5 below. The door will open when the correct password is entered

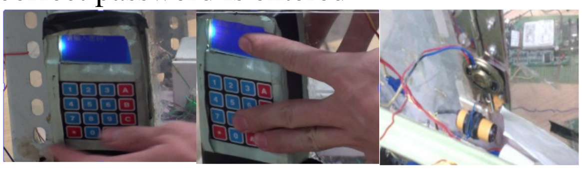

Fig. 5 Detects an abnormality and performs a voice alarm.

C. Smoke alarm module function test

The smoke alarm module displays "The air quality is well" when the indoor air quality is good, when the smoke fog is high, the controller displays "the air quality is not well", and the system turns the exhaust fan to open the indoor harmful gas. Exhaust outdoors; when the smoke concentration is too high, the controller displays "the air quality is bad" and starts the buzzer to perform a smoke alarm.

During the test, the smoke sensor is added with smoke by a lighter, and the smoke sensor is tested for detecting the harmful gas of the lighter. After testing, the function module works normally. Is show in Fig 6 below shows the smoke sensor test lighter age and the aversive concentration is high, the system and the exhaust fan take harmful gas to the outside.

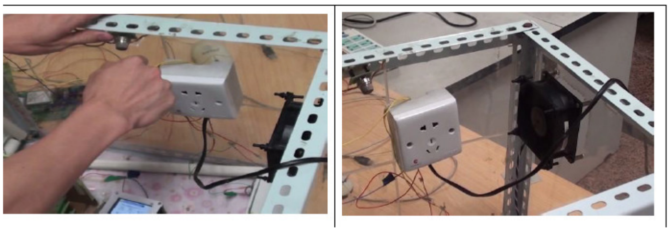

Fig. 6 the system will make a voice alarm when the door is opened.

D. Preventing children from falling off the module function test

This work is intended solely as a home model rather than a real-life installation. In the model, we use only a single human-body infrared sensor, which cannot distinguish whether the person near the window is an adult or a child. However, in an actual project, infrared sensors could be installed at different heights, enabling accurate detection and differentiation between adults and children. Fig 7 shows the window in an open state, with no human presence detected nearby. the infrared sensor detects movement near the window (simulated by a hand test), prompting the window to close automatically. Once the window is closed and the handle is released, the system will automatically reopen the window.

Fig. 7 the system will make a voice alarm when the door is opened.

IV. CONCLUSION

With the rapid advancement of electronic, communication, and automation technologies, smart home systems have become an essential component of modern living. They not only offer comfort, safety, and convenience but also enable real-time monitoring and intelligent management of household appliances and security infrastructure. This paper presents the design and implementation of an ARM-based embedded smart home system that integrates key modules including password protection, environmental monitoring, GSM-based remote control, voice alarm functionality, and a child safety mechanism.

Throughout the development process, we established a complete embedded system platform, successfully configured and ported an embedded Linux environment, and built a multi-module framework where each subsystem operates independently yet harmoniously. The system demonstrated reliable functionality in real-time response, accurate sensing, and automated control during practical testing. However, developing such a comprehensive system posed several challenges. The interdisciplinary nature of smart home design spanning software, hardware, network communication, and human factors required learning beyond traditional boundaries. Limitations such as a lack of user-friendly interface design, basic model wiring, and the system's susceptibility to interference were identified and provide clear direction for future improvement.

Despite these challenges, the system effectively validates the feasibility of using embedded ARM platforms for intelligent home automation. It reflects the potential for scalable, efficient, and modular smart home solutions that align with the evolving demands of modern lifestyles. Looking forward, further enhancement is necessary in areas such as interface optimization, sensor accuracy, wireless communication stability, and integration with cloud services. As smart home technology continues to evolve, it will play an increasingly important role in improving quality of life and reshaping how people interact with their living environments. This pepper lays a solid foundation for future studies and implementations in the field of smart homes, and contributes valuable insights into the development of intelligent, user-cantered, and scalable home automation systems.

V. FUTURE WORK

Although the development of the smart home system in this study has demonstrated significant potential, there are still several areas requiring enhancement and further research to meet the demands of real-world applications. The following future work directions are proposed based on current limitations and observations during system implementation:

*1) **Enhancement of Peripheral Circuit Design:*** A level-1 To improve user interaction and control simplicity, the home main controller should incorporate a richer set of peripheral components. This includes adding a display screen, a more intuitive and user-friendly input interface, and enhanced status indicators. Such additions will significantly improve usability, real-time feedback, and overall user experience.

2) ***Real-Device Testing and System Integration:*** A Due to constraints in experimental conditions, current testing was conducted primarily through analog simulations. In future work, the system should be integrated and tested with actual household appliances to validate its performance under real operating conditions. This would help in refining control logic and verifying system reliability during extended usage.

3) Advanced Security Monitoring with High-Speed Video: The current design supports only low-speed image acquisition, which is insufficient for modern home security needs. Future enhancements should include the integration of high-speed video capture using high-performance embedded processors such as DSP (Digital Signal Processors) or ARM Cortex-A series. This would enable more accurate motion detection, facial recognition, and real-time video surveillance.

4) Expansion of Functional Modules and Network Architecture: The present work focuses primarily on appliance control and image acquisition modules. However, a fully functional smart home system requires the integration of multiple additional modules, such as: Environmental monitoring (temperature, humidity, gas), Energy management, Lighting and curtain automation, Voice assistant integration. A more comprehensive home network architecture should also be developed to ensure seamless communication, scalability, and interoperability among all devices.

ACKNOWLEDGMENT

The author would like to express sincere gratitude to the Faculty of Computer Science, University of Laos, for providing the opportunity, resources, and academic support throughout the course of this research.

Special thanks are extended to the professors, lecturers, and staff members whose guidance and teaching have laid the foundation for this work. The author also wishes to thank fellow classmates and project collaborators for their cooperation, encouragement, and shared knowledge during the development of this smart home system.

Appreciation is also given to family and friends for their unwavering support, patience, and motivation throughout the research and writing process.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[3] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and challenges for realising the Internet of Things," *European Commission*, 2010.

[4] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[5] B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal, "Sentinel: Occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings," in *Proc. ACM SenSys*, 2013, pp. 17–30.

[6] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.

[7] M. A. Rahman, M. S. Hossain, and A. Almogren, "Trust-based secure resource management for fog computing in smart cities," *IEEE Access*, vol. 6, pp. 36041–36051, 2018.

[8] M. Chan, E. Campo, D. Estève, and J. Fourniols, "Smart homes – Current features and future perspectives," *Maturitas*, vol. 64, no. 2, pp. 90–97, 2009.

[9] D. J. Cook, G. M. Youngblood, E. O. Heierman III, et al., "MavHome: An agent-based smart home," in *Proc. IEEE Int. Conf. Pervasive Computing and Communications*, 2003.

[10] D. J. Cook, M. Youngblood, E. O. Heierman III, et al., *MavHome: An Agent-Based Smart Home*, 2003.

[11] D. M. Han and J. H. Lim, "Smart home energy management system using IEEE 802.15.4 and ZigBee," *IEEE Trans. Consumer Electron.*, vol. 56, no. 3, pp. 1403–1410, 2010.

[12] D. M. Han and J. H. Lim, "Design and implementation of smart home energy management systems based on ZigBee," *IEEE Trans. Consumer Electron.*, vol. 56, no. 3, pp. 1417–1425, 2010.

[13] S. K. Das, D. J. Cook, A. Battacharya, et al., "The role of prediction algorithms in the MavHome smart home architecture," *IEEE Wireless Commun.*, vol. 9, no. 6, pp. 77–84, 2002.

[14] L. Chen, C. D. Nugent, and H. Wang, "A knowledge-driven approach to activity recognition in smart homes," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 961–974, 2012.

[15] L. Chen, C. D. Nugent, and H. Wang, *A Knowledge-Driven Approach to Activity Recognition in Smart Homes*, IEEE Educational Activities Dept., 2012

[16] G. Demiris, M. J. Rantz, M. A. Aud, et al., "Older adults' attitudes towards and perceptions of smart home technologies: A pilot study," *Medical Informatics*, vol. 29, no. 2, pp. 8, 2004.

[17] M. A. A. Pedrasa, T. D. Spooner, and I. F. MacGill, "Coordinated scheduling of residential distributed energy resources to optimize smart home energy services," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 134–143, 2010.

[18] M. Skubic, G. Alexander, M. Popescu, et al., "A smart home application to eldercare: Current status and lessons learned," *Technology and Health Care*, vol. 17, no. 3, pp. 183–201, 2009.

[19] P. Rashidi and D. J. Cook, "Keeping the resident in the loop: Adapting the smart home to the user," *IEEE Trans. Syst., Man, Cybern., Part A: Syst. Humans*, vol. 39, no. 5, pp. 949–959, 2009.

[20] A. Fleury, M. Vacher, and N. Noury, "SVM-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 274–283, 2010.

[21] X. Hong, C. Nugent, M. Mulvenna, et al., "Evidential fusion of sensor data for activity recognition in smart homes," *Pervasive and Mobile Computing*, vol. 5, no. 3, pp. 236–252, 2009.

[22] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—Past, present, and future," *IEEE Trans. Syst., Man, Cybern., Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190–1203, 2012.

[23] S. Davidoff, M. K. Lee, C. Yiu, et al., "Principles of smart home control," in *Proc. Int. Conf. Ubiquitous Computing*, Springer, Berlin, Heidelberg, 2006, pp. 19–34.

[24] S. Davidoff, M. K. Lee, C. Yiu, et al., "Principles of smart home control," *Proc. Ubicomp*, vol. 4206, pp. 19–34, 2006.

[25] D. Valtchev, "Service gateway architecture for a smart home," *IEEE Commun. Mag.*, vol. 40, no. 4, pp. 126–132, 2002.

[26] C. L. Wu, C. F. Liao, and L. C. Fu, "Service-oriented smart-home architecture based on OSGi and mobile-agent technology," *IEEE Trans. Syst., Man Cybern., Part C (Applications and Reviews)*, vol. 37, no. 2, pp. 193–205, 200