Real-Time Monitoring Home Security System Utilizing Iot and Telegram Bot

Jung Kyu Park [1] **and Eun Young Park** [2+]

[1] Department of Computer Software Engineering, Changshin University, Gyeongsangnam-do, Korea

[2] Rinascita College of Liberal Arts and Sciences, Shinhan University, Gyeonggi-do, Korea

jkpark@cs.ac.kr, eypark@shinhan.ac.kr

Article Info
Page Number: 507 – 514
Publication Issue:
Vol. 71 No. 3 (2022)

Article History
Article Received: *12 January 2022*
Revised: *25 February 2022*
Accepted: *20 April 2022*
Publication: *09 June 2022*

Abstract

Home security systems are widely used to monitor situations such as disasters and theft. In the past, a special system was used by a security company to construct a security system, but recently, a security system is constructed using IoT products. In this study, monitoring was implemented using Raspberry Pi and sensors, which are widely used in IoT development, and the monitored data was implemented so that an alarm could be sent to the user using a chatbot. In particular, the Raspberry Pi camera module was used to detect indoor objects, and the temperature sensor and gas sensor were used to monitor indoor air conditions. The home security system produced as a result of the study used Raspberry Pi and sensors, and a messaging system was implemented using a Telegram chat messenger.

Keywords: Chatbot, IoT, Monitoring, Raspberry PI, Sensor

1. Introduction

According to the 2020 National Police Agency crime statistics, the number is on the decline from 1.8 million in 2015, 1.66 million in 2017, and 1.6 million in 2020, but more than 1.6 million crimes are occurring for three consecutive years. In particular, the arrest rate for crimes was 80% in 2015, 85% in 2017, and 81% in 2020, showing a high analogy, exceeding 82% on average over 7 years (Korean National Police Agency, 2020). The reason for such a high arrest rate is that the domestic IoT security system is active. The use of the IoT security system can increase the crime arrest rate in the home and prevent fire accidents that occur in the home (Waseem et al., 2020; Nickson et al., 2021; Daemin et al., 2019; Jung Kyu & Jaeho, 2020).

In a typical residential environment, security systems only detect external intrusions, so it is impossible to prevent accidents such as fires (Jung Kyu & Hyungyun 2020; Jung Kyu

[+] Corresponding author. Tel.: +82-31-870-3776
E-mail address: eypark@shinhan.ac.kr

& Kihun, 2020; Jung Kyu & Jaeho, 2020; Gabriel et al., 2020). In addition, since an alarm is displayed in an environment in which the security system operates, it is difficult to receive real-time information when outside. Because incidents occurring inside the home cannot be reported in real time, there is a problem in that actions against accidents are slow. In general, motion detection sensors are used to monitor the inside of a house, but recently, CCTV has been widely used. Many CCTVs record the environment in real time, but do not notify users in real time (Tsu-Yang et al., 2019; Gregory et al., 2018; Federico et al., 2020). Due to this problem, it is necessary to notify the user in real time when a special situation occurs at home. In this study, we plan to utilize IoT technology to notify users in real time.

Recently, various home security systems are being sold in the market, but they perform only a specific function and do not provide function scalability. In addition, despite providing only one or two types of sensors and a limited number of sensors, they are being sold at a high price. In this study, to solve this problem, the Raspberry Pi board, which is widely used in IoT research, was used. The Raspberry Pi offers better system performance and various scalability compared to a single-function home security system sold in general. In addition, a home security system was implemented using a raspberry dedicated camera sensor, temperature sensor, and gas sensor. Finally, a system that sends messages to users in real time using Telegram, which is widely used as a smartphone messaging application, was implemented and verified.

2. Implementation Method

In this paper, the system development life cycle (SDLC) method is used as shown in Figure 1. SDLC, also known as the application development lifecycle in systems engineering, information systems and software engineering, is the process of planning, creating, testing, and deploying information systems. This cycle generally includes the stages of requirements analysis, design, development and testing, implementation, and documentation evaluation.

Planning: In order to plan the security system, it should be assumed that the following problems may occur. Free access to the house, access without information, lack of security, indoor air condition information, indoor temperature information, etc. should be considered. It is necessary to grasp such information in real time and provide information so that users can easily understand it.

Analysis: This step includes collecting facts, interpreting them, diagnosing problems, and recommending improvements to the system. After identifying the problem, write a general description of the system. In this stage, it is necessary to analyze in detail what is necessary to design and manufacture the system to be implemented. We will use hardware such as Raspberry Pi, temperature/humidity sensor, camera module, LED, buzzer, etc. to implement the system. In addition, the message queue method is used for software implementation.

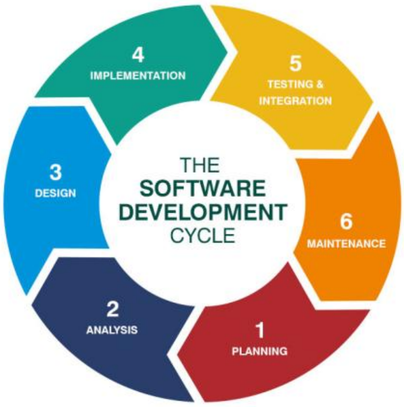


Fig. 1: SDLC Methodologies

Design: In this step, the requirements gathered in the SRS document are used as input and the software architecture used to implement the system development is derived. In the detailed phase, the system design is prepared and the system configuration is established. After that, verify that the system is well configured. Finally, the optimal system configuration is selected.

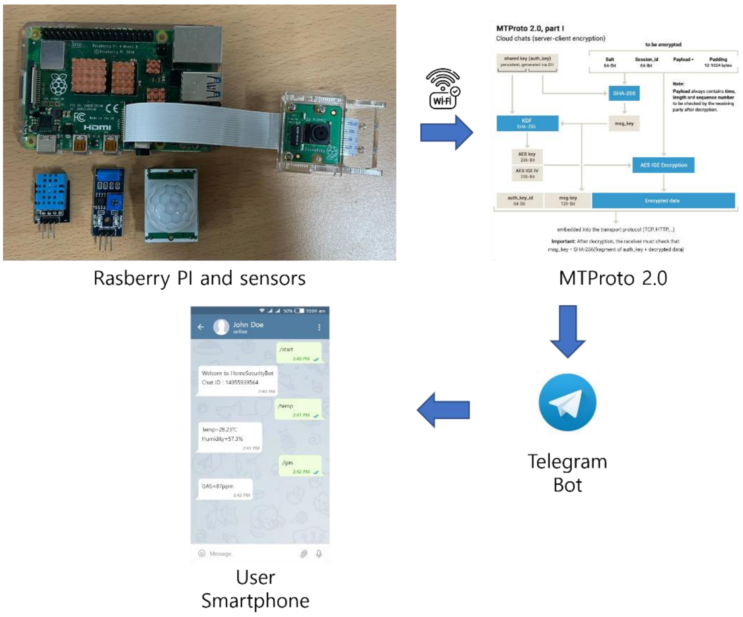
Implementation: Implementation and coding begin when the developer receives the design documentation. The software design is translated into source code. All components of the software are implemented at this stage. In detail, it proceeds in the following order: These are the order of tools and materials preparation, controller installation, module installation of all components, library installation of the entire configuration, installation support software, system configuration, and system testing.

Testing: Testing begins when the coding is complete and the module is released for testing. In this stage, the developed software is thoroughly tested and any defects found are assigned to the developer to be fixed. Retesting and regression testing are performed until the software meets customer expectations. Testers refer to the SRS documentation to ensure that the software meets the customer's standards.

Maintenance: Once the product is tested, it is either deployed to a production environment or UAT (User Acceptance Test) is done according to customer expectations. After the product is deployed into production, it is maintained by the developer, i.e. if something goes wrong and needs to be fixed or improved.

3. Proposed System

The purpose of this paper is to design and implement a home security system using Raspberry Pi (Brittany et al., 2019; Nirnay et al., 2020; Sungjin et al., 2021). The proposed system can measure the gas and temperature inside the home and is designed to monitor outside intruders. In this way, by monitoring the state of the house, when a problem occurs, it is possible to notify the user in real time. The proposed system is illustrated in Figure 1. A camera sensor, gas sensor, and temperature/humidity sensor are connected to the Raspberry Pi used in the designed system, and users can monitor in real time through the messaging system.


Rasberry PI and sensors
Wi-Fi
MTProto 2.0, part I
Cloud chats (server-client encryption)
to be encrypted
shared key (auth_key)
persistent, generated via DH
Salt 64-Bit
Session_id 64-Bit
Payload +
Padding 12-1024 bytes
SHA-256
Note:
Payload always contains time, length and sequence number to be checked by the receiving party after decryption.
KDF SHA-256
msg_key
AES key 256-Bit
AES IGE IV 256-Bit
AES IGE Encryption
auth_key_id 64-Bit
msg key 128-Bit
Encrypted data
embedded into the transport protocol (TCP, HTTP, ..)
Important: After decryption, the receiver must check that msg_key = SHA-256(fragment of auth_key + decrypted data)
MTProto 2.0
Telegram Bot
John Doe
online
Message
User Smartphone

/start 2:40 PM

Welcom to HomeSecurityBot
Chat ID : 14955939564 2:40 PM

/temp 2:41 PM

Temp=28.23℃
Humidity=57.3% 2:41 PM

/gas 2:42 PM

GAS=87ppm 2:42 PM

Fig 2: Home Security System architecture

The proposed security system can be largely divided into hardware and software systems. The hardware system uses Raspberry Pi as a basic system, and here three camera sensors, temperature/humidity sensors, and motion sensors are mounted. This hardware system can be controlled and monitored externally through the Telegram chatbot. This hardware system is connected to the Internet for real-time notifications to users. In this case, the *Mproto* protocol is used to transmit information to the Telegram chatbot. The *Mproto* protocol is provided by Telegram and is used to create chatbots. The *Mproto* protocol consists of commonly used TCP, UDP web sockets, HTTP, and HTTPS.

The security monitoring system proposed in this paper consists of several components such as temperature sensor, gas sensor, PIR sensor and camera module. This system can be controlled and monitored in real time through the telegram service. For this reason, the proposed system can monitor indoor environmental information and can provide security functions through it. Information obtained through the system can be checked in real time and provided to users in real time.

The hardware system developed through research operates by being connected to the Internet through Ethernet or WiFi depending on the operating environment. In particular, it communicates with Telegram through the *Mproto* protocol for real-time information provision and monitoring. Telegram uses the proprietary *Mproto* protocol and has the characteristic of encrypting all messages sent and received. Figure 2 shows the system authentication processing steps using the *Mproto* protocol.

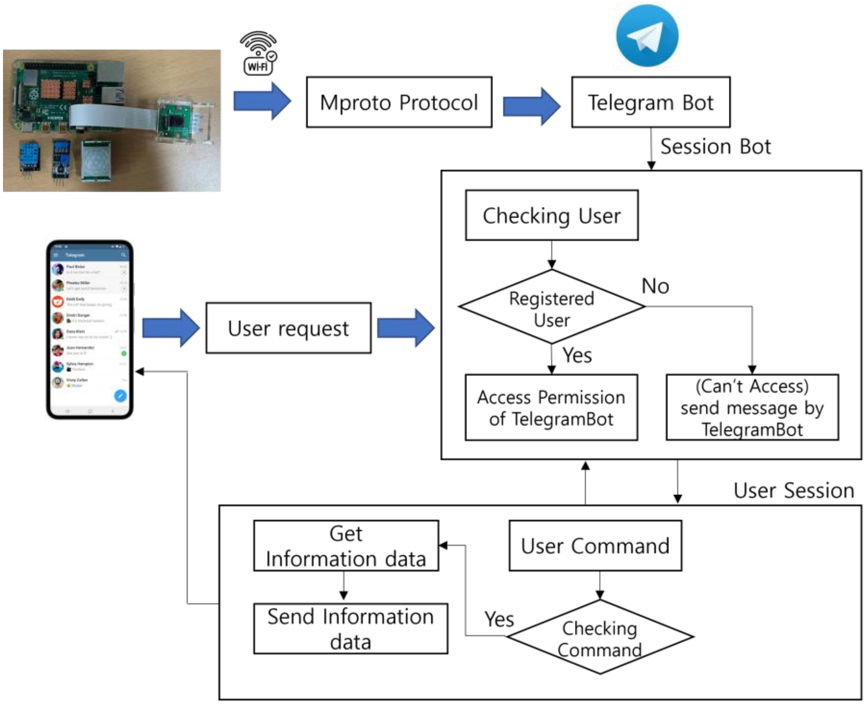


Fig 3: Processing steps of System authentication

Telegram also provides a ‘secure conversation’ function that sends messages directly from a device to a device using a peer-to-peer (P2P) method without going through the server cloud. It is an end-to-end encryption method that allows the terminals to check each other's messages only through the terminal used for the conversation through encryption and decryption. BOT is a tool for developing new applications on Telegram. This BOT is used to process the information acquired from the system and provide the information in real time according to the user's request. The system using Telegram provides security by verifying registered users. In particular, in the registered user's session, the command sent by the user is checked, analyzed whether the system can process it, and if possible, the command is executed. If the user's command cannot be executed, the system notifies the user that the command is incorrect. Conversely, if command execution is possible, the system executes the specified user command and outputs the information. And the system sends the information to the user as a message. In order to process such a step, the following steps are performed. When a user accesses the system, the system checks whether the user is a registered user. The registered user gets access to the system and the user can obtain the desired information by issuing commands to the system.

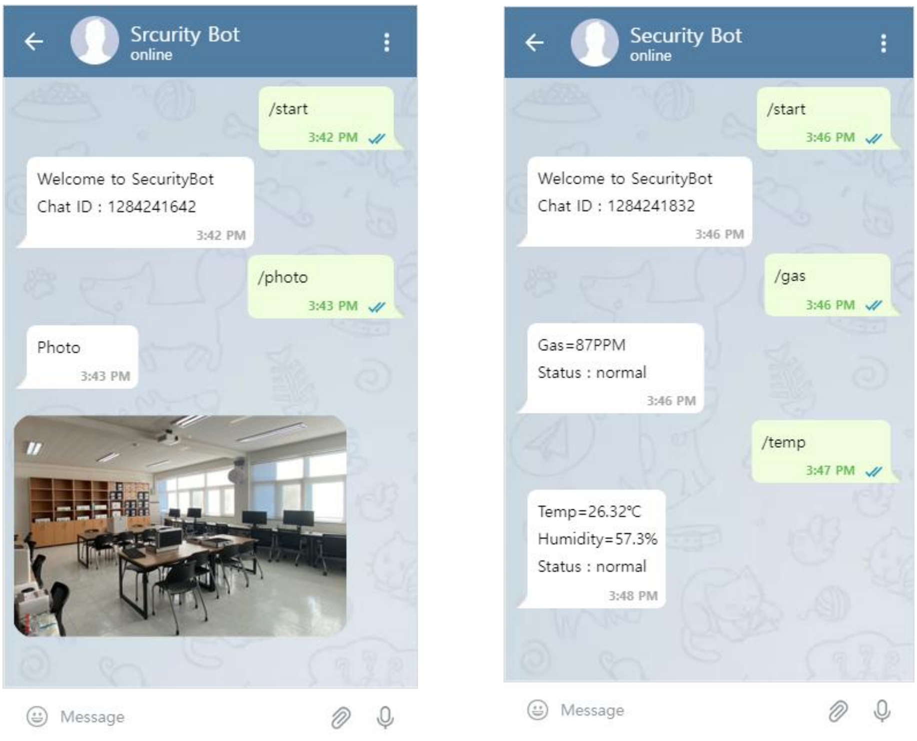
4. Results

To verify the proposed system, a security system was actually implemented with Raspberry Pi and an experiment was performed. In the system implementation, various sensors were used assuming a home environment. In particular, a DHT22 sensor for indoor temperature/humidity identification, a PIR sensor for external intruder monitoring, and a raspberry camera for anomaly detection were used.

To get system information using the telegram bot, we set up some commands. The "/start" command is used when connecting to the system for the first time. Once connected, you can get a Chat ID and send commands. The "/photo" command can take a picture of the environment using a camera attached to the system. In addition, environmental photos are automatically sent when there is an outside intruder. Figure 4(a) shows an environment photo sent with the "/photo" command. Figure 4(b) shows the temperature/humidity and gas information status of the environment using the "temp" and "/gas" commands. Figure 4 shows the results of command execution using the Telegram bot.

Table 1. Rule of gas status

Status	CO(PPM)
Good	0~50 ppm
Normal	51~100 ppm
Warning	101~300 ppm
Dangerous	over 300ppm



/start
3:42 PM

Welcome to SecurityBot
Chat ID : 1284241642
3:42 PM

/photo
3:43 PM

Photo
3:43 PM

/start
3:46 PM

Welcome to SecurityBot
Chat ID : 1284241832
3:46 PM

/gas
3:46 PM

Gas=87PPM
Status : normal
3:46 PM

/temp
3:47 PM

Temp=26.32℃
Humidity=57.3%
Status : normal
3:48 PM

(a) Result of "/photo" command

(b) Result of "/gas" and "/temp" command

Fig 4. Result of command execution

Referring to the data in this study, the CO gas detection threshold was specified as "safe at 50 ppm or less, normal at 51 to 100 ppm, warning at 101 to 300 ppm, and dangerous above 300 ppm. The manufactured system measures the gas sensor value periodically to automatically transmit the gas state value to the Telegram bot account when a set limit is reached or a dangerous state is reached.The system sends the data collected through the sensor to the cloud to store the data and use it for analysis. The state of the gas was divided into four stages as shown in Table 1. In Figure 5, the temperature data transmitted to the cloud is displayed in real time.

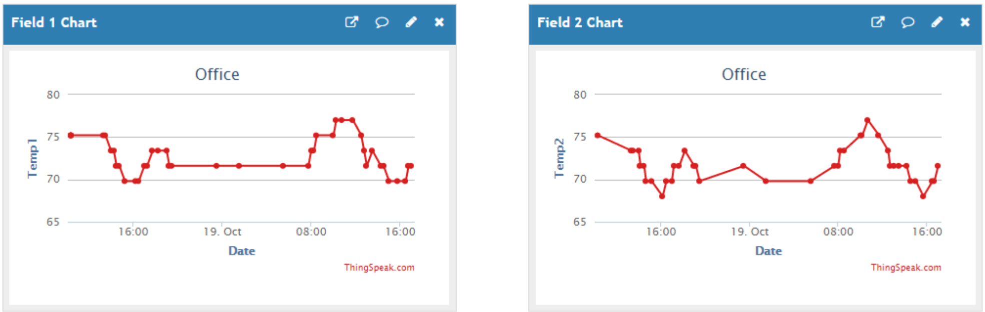


Fig 5. Data representation from Cloud

5. Conclusion

In this study, we designed and implemented a home security system that allows users to monitor their homes in real time. The proposed system is based on Raspberry Pi and is designed to expand various sensors. Sensors attached to the system can collect information about the interior and environment of the home and transmit the information to the user in real time through the network. For real-time transmission from users, a chatbot was created using the *Mproto* protocol provided by Telegram. Using the chatbots produced through the experiment, it was possible to receive real-time notifications about accidents occurring inside the house. In addition, desired sensor information inside the house can be requested and confirmed in real time from the outside.

6. Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2021R1F1A1052129)

7. References

1. Brittany, D.D., Janelle, C.M., Mohd, A., (2020). Vulnerability Studies and Security Postures of IoT Devices: A Smart Home Case Study. IEEE Internet of Things Journal, 7(10), 10102-10110.
2. Daemin, S., Keon, Y., Jiyoon, K. Philip, V.A., Jeong-Nyeo, K., & Ilsun You, (2019). A Security Protocol for Route Optimization in DMM-Based Smart Home IoT Networks. IEEE Access, 7, 142531-142550.
3. Federico, A., Zeyu, F., Yang, L., Ling, S., & Syed, M.N., (2020). 2D Pose-Based Real-Time Human Action Recognition With Occlusion-Handling" IEEE Transactions on Multimedia, 22(6), 1433-1446.

4. Gregory, F., Arun, V., Carlos, C., & Howard S., (2018). A Master Attack Methodology for an AI-Based Automated Attack Planner for Smart Cities. IEEE Access, 6, 48360-48373.
5. Gabriel, R., Vladimir, S.P., (2020). LPWAN Based IoT Surveillance System for Outdoor Fire Detection. IEEE Access, 8, 114900-114909.
6. Jung Kyu, P., & Jaeho, K., (2020). Real-Time Monitoring and Control System of Server Room based on IoT. Journal of The Korea Internet of Things Society, 6(3), 7-13.
7. Jung Kyu, P., & Hyungyoon, S., (2020). ZigBee-Based Smart Fire Detector for Remote Monitoring and Control. International Journal of Advanced Science and Technology, 29(3), 10431-10441.
8. Jung Kyu, P., & Kihun, N., (2020). Implementation of Multiple Sensor Data Fusion Algorithm for Fire Detection System. Journal of The Korea Society of Computer and Information, 25(7), 9-16.
9. Jung Kyu, P., & Jaeho, K., (2020). Smart fire monitoring system with remote control using ZigBee network. Indonesian Journal of Electrical Engineering and Computer Science, 21(2), 1132-1139.
10. Korean National Police Agency, “Crime Statistics”, https://www.police.go.kr/www/open/publice/publice03_2020.jsp, 2020.
11. Nickson, M.K., Nor, M.S., Wencheng, Y., Craig, V., & Victor, R.K., (2021). A Review of Security Standards and Frameworks for IoT-Based Smart Environments. IEEE Access, 9, 121975-121995.
12. Nirnay, G., Saket, C., Vinay, S., & Yuval, E., (2019). SoftAuthZ: A Context-Aware, Behavior-Based Authorization Framework for Home IoT. IEEE Internet of Things Journal, 6(6), 10773-10785.
13. Sungjin, Y., Namsu, J., Youngho, P., (2021). Lightweight Three-Factor-Based Privacy-Preserving Authentication Scheme for IoT-Enabled Smart Homes. IEEE Access, 9, 126186-126197.
14. Tsu-Yang, W., Xiaoning, F., King-Hang, W., Chin-Feng, L., Naixue, X., & Jimmy M.W., (2019). A DNA Computation-Based Image Encryption Scheme for Cloud CCTV Systems. IEEE Access, 7, pp. 181434-181443.
15. Waseem, I., Haider, A., Mahmoud, D., Bilal, R., & Yawar, A.B., (2020). An In-Depth Analysis of IoT Security Requirements, Challenges, and Their Countermeasures via Software-Defined Security. IEEE Internet of Things Journal, 7(10), 10250-10276.