Design and Implementation Smart Home System Based on The Internet of Things (IoT) With A Laravel Web Interface

Laila Khoirunnisa Nurul Imani, Muhammad Khosyi'in
Electrical Engineering Department, Universitas Islam Sultan Agung,
Semarang, Indonesia

Abstract
The implementation of IoT in smart home systems faces a number of obstacles, such as a lack of integration between components, limitations in real-time control, and reliability of the system in responding to needs. Many smart home solutions require special applications to access them, making them impractical for device transfers. Sensor data is used to control components such as lights or stepper motors and displayed on the website. Data transmission done through an MQTT broker to a Laravel-based website. The website allows users to monitor and control components from any device with a browser. The success rate of the control process via the website reached 100%. Response time from reading to component response is also fast, with an average response of 1.271 s for infrared sensors, 1.182 s for rain sensors, and 0.420 s for temperature sensors. With this, users can monitor and control the smart home system efficiently.
Keywords: *smart home, Internet of Things, Arduino UNO, ESP8266, MQTT, Laravel.*

1. Introduction

A smart home system serves as a support system in today's era of high mobility. Internet of Things (IoT) is an infrastructure of objects connected via the internet [1]. The utilization of the Internet of Things (IoT) in smart home systems offers a solution to enhance the comfort and efficiency of modern living. The Internet of Things (IoT) is the foundation for transforming physical objects into virtual objects with embedded intelligence. Besides being used as a communication tool, the Internet can also be used as a remote-control tool. This control system can be utilized to control household devices commonly known as Smart Home [2]. Sensors are then integrated with various actuators or control devices to respond to the data obtained and/or can also be used for data monitoring itself [3].

The purpose of a smart home system includes increasing comfort and efficiency for its users. The selection of the website as a control tool in the context of a smart home was chosen because of its convenience. Through the website, users can easily control household appliances from anywhere, anytime, provided that the device used is connected to the internet and equipped with a web browser [4]. This flexibility allows homeowners to control their appliances without time and space constraints. This simplicity of access creates ease of use, making the website a practical and reliable interface for remote control of household appliances. In addition, the website can also be a place to view the components to be monitored.

2. Related Work

In recent years, the Internet of Things has become a topic of interest to researchers. IoT is considered a part of the future that involves billions of devices where each device can communicate with each other [5]. By utilizing various technologies, various devices in the home are equipped with remote control systems. Smart homes have the potential to improve comfort, convenience, security, and energy management [6]. Smart homes allow users to manage energy and increase savings by controlling lighting, windows, and so on [7]. By utilizing technology, smart home systems create intelligent and environmentally friendly automation [8]. The importance of achieving the main goal of smart home systems, namely increased comfort and efficiency for residents, is the main focus in the literature review [4]. Along with the development of technology, IoT-related research is increasingly being explored. For example, in 2020 there was a study entitled "Implementation of a Smart Home Control System Using Laravel". In this research, a website is used as a means of controlling the device by utilizing an Arduino microcontroller. The use of the website is intended so that the control of devices in the house is not limited to distance and place [9]. Thus, the development of an IoT-based smart home system not only covers technical aspects but also has the potential to have a significant impact on modern lifestyles, improve energy efficiency, and expand the possibilities of remote control for comfort. Sensors are integrated with various actuators or control devices to respond to the data obtained and or can also be used for monitoring data itself [3].

3. Methodology

3.1 System Requirements Analysis and Tool Design

At this stage an analysis is carried out regarding how the system will be made. It identifies what equipment is needed and will be included in the system. System requirements analysis involves an in-depth understanding of the desired functions, performance, and features in the system. Next, the system design is carried out, which includes electronic, mechanical, and program aspects. System design is done by ensuring that each element can integrate efficiently and run as expected.

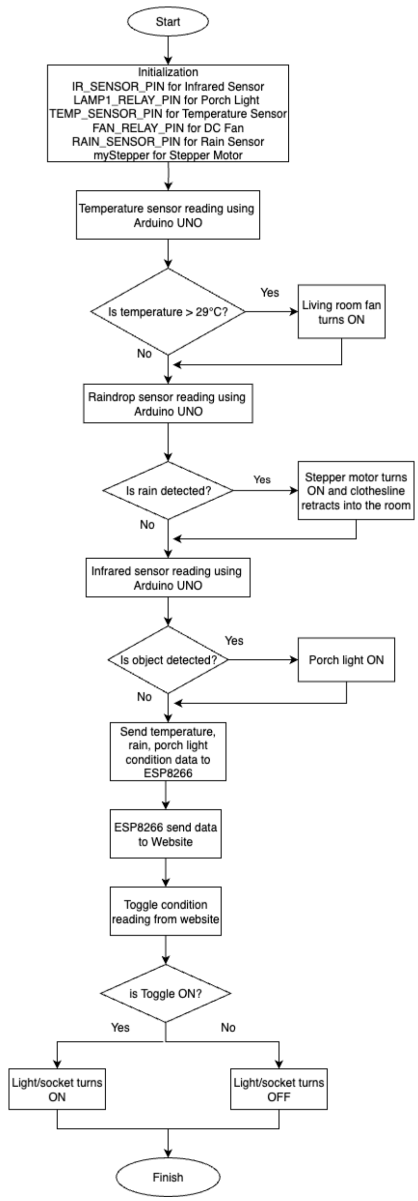

Fig. 1 Smart home system flow chart

The system starts with the temperature reading on the Arduino UNO. If temperature is > 29°C, fan on living room will turn ON. After that Arduino UNO continue reading raindrop sensor. If rain detected, motor stepper will ON and clothesline will start retracts into the room. Then Arduino UNO read infrared sensor condition. If object is detected, porch light will turn ON. After finish reading all sensor Arduino UNO will send all sensor condition and value to ESP8266 through serial communication. ESP8266 will publish sensor data to website through MQTT protocol. Besides sending data to the website, the ESP8266 also receives data from the website to toggle the state of lights and sockets. The data from website will process and control light and socket that connected to ESP8266.

3.2 Electronic design and prototype of smart home

In this system, several components are used such as lights, stepper motors, DC fans, DS18B20 temperature sensors, infrared sensors, and rain sensors. On the side of the terrace there is an infrared sensor which is placed on the terrace ceiling and if it detects an object, it will turn on the terrace lights which are also installed on the front of the prototype. On the front room page, there is a DS18B20 temperature sensor which, if it detects a temperature of > 29 ° C, will automatically turn on the DC fan mounted on the front room wall. At the back there are 2 rooms, where each room has a lamp called lamp 1 and lamp 2. Each of these lights will be controlled via the website. In addition, in the back room there is a rain sensor, which if it detects water droplets, it will automatically turn on the stepper motor and rotate the clothesline to enter the room. Once the rain is no longer detected, the stepper motor will turn on again and rotate the reverse direction so that the clothesline will come back out.

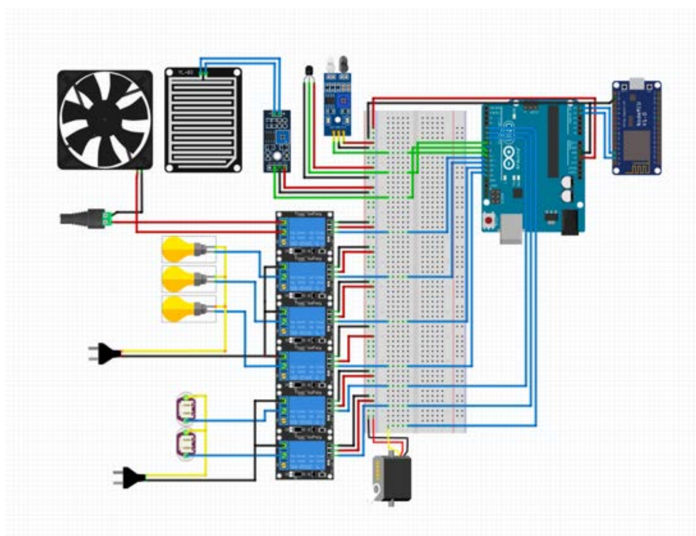

Fig. 2 Electronic design of smart home

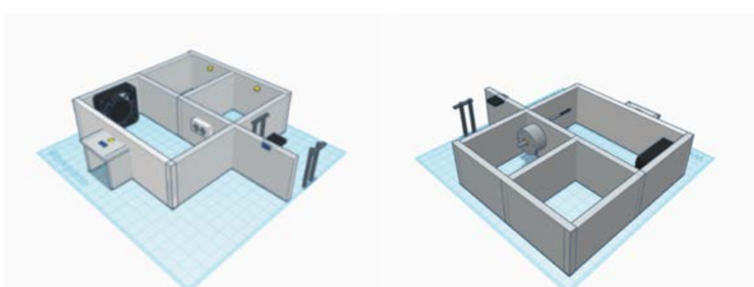

Fig. 3 Smart home prototype design

3.3 Data Transfer System Between Arduino and ESP8266

On Arduino there are components such as temperature sensors, rain sensors, and porch light conditions. The three-component data is sent serially to ESP8288 via the RX/TX pin. The data structure is as follows:

Table 1: Data exchange between Arduino and ESP8266

Component	Condition	Data
Temperature Sensor	Temperature	*temp*: (float *temp*erature),
Rain Sensor	Rain	*rain*: 1,
	No Rain	*rain*: 0,
Infrared Sensor	Object Detected	porch_light: 1
	Object Not Detected	porch_light: 0

3.4 Data Exchange System Between ESP8266 and MQTT

After ESP8266 receives data from Arduino, ESP8266 will continue to send data to MQTT. MQTT will be the intermediary between ESP8266 and the website. The following is the data structure sent via MQTT:

Table 2: Data exchange between ESP8266 and MQTT

Component	Condition	Data
Temperature Sensor	Temperature	/*temp*(float *temp*erature)
Rain Sensor	Rain	/*rain*/1
	No Rain	/*rain*/0
Infrared Sensor	Object Detected	/porch_light/1
	Object Not Detected	/porch_light/0

3.5 Data Exchange System Between MQTT and ESP8266

In addition to sending data, ESP8266 will also receive data from the website via MQTT. The data received is the toggle ON/OFF action of lamp 1, lamp2, and the socket. The following is the data structure:

Table 3: Data exchange between MQTT and ESP8266

Component	Condition	Data
Lamp 1	ON	/lamp2/1
	OFF	/lamp2/0
Lamp 2	ON	/lamp3/1
	OFF	/lamp3/0
Lamp 3	ON	/socket/1
	OFF	/socket/0

4. Implementation and Results

4.1 Implementation of smart home control and monitoring website

Login page contains a form to log in into the website. Previously the user was registered through the database. For register new user, there is dedicated user page.

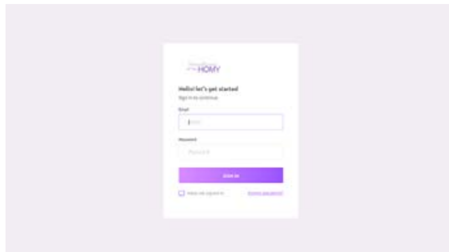

Fig. 4 Login page

After the user has successfully logged in, the user will be directed to the control page on the website. On this page there is an ON/OFF toggle to control lamp 1, lamp2, and socket. In addition, there is also an indicator to see the ON / OFF condition of the terrace lights, where the ON / OFF condition is determined from the condition of the infrared sensor reading on the terrace.

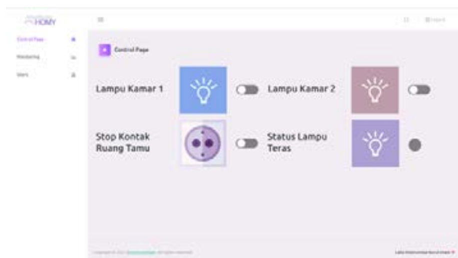

Fig. 5 Control page

On monitoring page there is a graph of the temperature data obtained from the temperature sensor. In addition, there are also indicators obtained from the rain sensor. If the rain sensor does not detect rain, a picture of the sun will be displayed. Meanwhile, if the rain sensor detects rain, the image displayed is a picture of rain.

Fig. 6 Monitoring page with rain state false

IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 12 Issue 09, September 2025
ISSN (Online) 2348 – 7968 | Impact Factor – 6.72
www.ijiset.com

Fig. 7 Monitoring page with rain state true

4.2 System Testing

4.2.1 Manual Control Website Testing

Testing is done by performing ON/OFF commands for lamp 1, lamp 2, and socket from the website and seeing the results on physical components.

Table 4: Manual control testing website

Component	state	Freq Inst	Website Status	Lamp Status	AVG Respons TIme
Lamp 1	ON	10x	ON	ON	1.393 s
Lamp 1	OFF	10x	OFF	OFF	1.331 s
Lamp 2	ON	10x	ON	ON	1.357 s
Lamp 2	OFF	10x	OFF	OFF	1.342 s
Socket	ON	10x	ON	ON	1.290 s
Socket	OFF	10x	OFF	OFF	1.058 s

Based on the test results, response time ranges from 1.058 s to 1.393 s. The test was conducted repeatedly to obtain measurement data with a high degree of precision [10]. It was found that in 10x tests for all control components, the results were correct. The ON/OFF status of the components follows the commands sent through the website. In addition, the status on the website is also always appropriate in 10x testing. This shows that the communication between the website and ESP8266 is going well.

4.2.2 Infrared Sensor Testing

Infrared sensor testing is carried out to determine the success rate of the sensor in reading the condition of the terrace and the response of the lights on the terrace.

Table 5: Infrared sensor testing

Object	Time Detection	Lamp Status	Lamp Status on Website	Porch Light Response Time
Exist	12:32:23.946	ON	Green	1.087 s
None	12:32:25.033	OFF	Grey	1.400 s
Exist	12:32:26.433	ON	Green	0.760 s

Object	Time Detection	Lamp Status	Lamp Status on Website	Porch Light Response Time
None	12:32:27.193	OFF	Grey	1.742 s
Exist	12:32:28.935	ON	Green	0.735 s
None	12:32:29.670	OFF	Grey	3.182 s
Exist	12:32:32.852	ON	Green	0.580 s
None	12:32:33.432	OFF	Grey	0.687 s
Average response time				1.271 s

In the Infrared Sensor test results, when an object is detected, the porch lights will turn on and the status of the lights on the website will be green. When the object is considered gone, the porch light will turn off and the status of the lights on the website will turn gray. In 8x tests, the status of the lights on the website and the on/off condition of the website are always as expected. The response time of the lights turning on/off when an object is detected varies. Out of 8x tests, the fastest response time is less than 1 s or exactly 0.580 s. While the longest response time is less than 4 s or more precisely 3.182 s. The average response time for infrared sensor testing is 1.271s.

4.2.3 Stepper Motor and Rain Sensor Testing

This test was conducted to determine how quickly and accurately the stepper motor and rain sensor responded to the given conditions. In this test, rain conditions were not tested directly in the field, but were represented by wetting the surface of the rain sensor. Sunny conditions were simulated by drying the rain sensor after it had been wet. During the test, changes in the rain sensor status and the stepper motor response were observed. The response time was also recorded, which is the difference between when the condition (wet or dry) was applied to the sensor and when the stepper motor moved according to the predetermined logic. With this method, it can be analyzed whether the system is able to respond quickly and accurately to changes in simulated environmental conditions.

Table 6: Stepper motor and rain sensor testing

Weather Condition	Time Detection	Motor Stepper Status	Response Time
Rain	12:43:22.000	Rotate left	0.420 s
Sunny	12:43:54.000	Rotate right	0.570 s
Rain	12:44:27.000	Rotate left	0.444 s
Sunny	12:45:01.000	Rotate right	0.844 s
Rain	12:45:06.000	Rotate left	4.466 s
Sunny	12:45:41.000	Rotate right	0.831 s
Rain	12:46:14.000	Rotate left	0.954 s
Sunny	12:46:47.000	Rotate right	0.933 s
Average response time			1.182 s

From eight tests, when the rain sensor detected water droplets, the stepper motor rotated to the left. Meanwhile, when the rain sensor no longer detects water droplets, the stepper motor will rotate to the right. The response time from the sensor detecting water droplets to the rotation of the stepper motor varies. The fastest response time was recorded at 0.420s and the longest response time was 4.466s. So, it was found that the average time required for the stepper motor to respond to the input received by the rain sensor was 1.182 s.

4.2.4 DC Fan and DS18B20 Temperature Sensor Testing

Testing was conducted by changing the temperature dramatically, either by warming or cooling the temperature sensor. From the action then observed the response of the actuator, namely the DC Fan.

Table 7: DC fan and temperature sensor testing

Temperature Condition	Time Detection	DC Fan Status	Response Time
29°C	12:49:35	ON	0.438 s
28.87°C	12:49:53	OFF	0.693 s
29°C	12:50:24	ON	0.201 s
28.87°C	12:50:43	OFF	0.496 s
29°C	12:51:04	ON	0.388 s
28.94°C	12:51:18	OFF	0.246 s
29°C	12:52:03	ON	0.483 s
28.87°C	12:52:23	OFF	0.416 s
Average response time			0.420 s

Testing the DC fan and DS18B20 temperature sensor if the temperature is detected to exceed 29 ° C, the fan will turn on. Testing is done 8x and the results are as expected. When the temperature reaches 29 ° C, the DC fan will turn on. If the temperature has dropped below 29 ° C, the DC fan will automatically turn off. The response time from when the temperature is detected 29 ° C or < 29 ° C varies. The average response time is 0.420 s with the fastest response time of 0.201 s and the longest of 0.693 s.

4. Conclusions

The smart home system is capable of monitoring temperature, detecting rain, and controlling electronic components such as lights, fans, sockets, and stepper motors. It integrates Arduino UNO and ESP8266 through serial communication, while the ESP8266 connects to the website using the MQTT protocol. The system enables remote control and monitoring with fast response times, providing both convenience and efficiency.

References

[1] R. Sokullu, M. A. Akkaş, and E. Demir, "IoT supported smart home for the elderly," Internet of Things (Netherlands), vol. 11, Sep. 2020, doi: 10.1016/j.iot.2020.100239.

[2] A. S. Rafika, M. Surya, H. Putra, and W. Larasati, "SMART HOME AUTOMATIC MENGGUNAKAN MEDIA BLUETOOTH BERBASIS MIKROKONTROLLER ATMEGA 328," 2015.

[3] N. Surantha and W. R. Wicaksono, "Design of Smart Home Security System using Object Recognition and PIR Sensor," Procedia Comput Sci, vol. 135, pp. 465–472, 2018, doi: https://doi.org/10.1016/j.procs.2018.08.198.

[4] Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A Literature Review. Journal of Computer and Communications, 03(05), 164–173. https://doi.org/10.4236/jcc.2015.35021.

[5] Li, S., Xu, L. Da, & Zhao, S. (2015). The internet of things: a survey. Information Systems Frontiers, 17(2), 243–259. https://doi.org/10.1007/s10796-014-9492-7

[6] P. Mahamuni, M. R. Kadam, M. Pranav Mahamuni, and M. Y. Parikh, "Smart Home System," 2015. [Online]. Available: www.ijirae.com

[7] T. S. Gunawan et al., "Prototype design of smart home system using internet of things," Indonesian Journal of Electrical Engineering and Computer Science, vol. 7, no. 1, pp. 107–115, Jul. 2017, doi: 10.11591/ijeecs.v7.i1.pp107-115.

[8] Y. L. Hsu et al., "Design and implementation of a smart home system using multisensor data fusion technology," Sensors (Switzerland), vol. 17, no. 7, Jul. 2017, doi: 10.3390/s17071631.

[9] Khoirunnisa Nurul Imani, L., Alicia, N., & Yusmaniar Oktiawati, U. (2020). IMPLEMENTASI SISTEM PENGENDALI RUMAH PINTAR MENGGUNAKAN LARAVEL (Vol. 1, Issue 1).

[10] M. Khosyi'in, A. A. Nugroho, and A. Yulistiyanto, "Three-Phase Power Data Logger Using IEM 3255 Schneider Module Based On Internet Of Things(IOT)," Conference SENATIK STT Adisutjipto Yogyakarta, vol. 5, Nov. 2019, doi: 10.28989/senatik.v5i0.359.

Author Biographies

Laila Khoirunnisa Nurul Imani received the Diploma (D3) degree in Electrical Engineering from Universitas Gadjah Mada, Indonesia. She is currently working as a Software Engineer while pursuing her Bachelor's degree in Electrical Engineering through the extension program at Universitas Islam Sultan Agung Semarang, Indonesia. She has previously published academic work in the field of electrical engineering. Her current research interests include the Internet of Things (IoT), smart home systems, and web-based control applications.

Muhammad Khosyi'in was born in Jepara, Central Java, Indonesia, on July 25, 1979. He received his Bachelor's degree in Electrical Engineering from Universitas Islam Sultan Agung (UNISSULA), Semarang, Indonesia. He completed his Master's degree in Electrical Engineering at Institut Teknologi Sepuluh Nopember (ITS), Surabaya, and earned his Doctorate in Electrical Engineering from Universitas Sriwijaya, Palembang, Indonesia. He is currently a lecturer in the Department of Electrical Engineering at Universitas Islam Sultan Agung, Semarang. His research focuses on electronic technology, particularly navigation systems, electrical instrumentation, and sensor–actuator systems.