IoT Based Automatic Blind Controller

LINXI LI
l.linxi@wustl.edu

WENXUAN ZHU
540516776@qq.com

JIANJUN WEI
jianjun@wustl.edu

Abstract

Seeking to be much more general to users, smart home researchers have focused on the concept of control. They attempt to minimize user's intervention in monitoring home settings and controlling home appliances by developing much more intelligent advice. This paper presents an application of the smart home by integrating internet of Things(IoT) with wireless technology. We build a system that could control the blind automatically without frequent human interference. Our system encompasses Raspberry PI, a sensitive light sensor, a powerful motor and basic room settings. We report our experience and lessons learned from different trails of the system based on different external environments. Our experiences show the feasibility of achieving reliable adjustment of light within the room, using a wireless network and infrastructure in ordinary room. Also, we highlight the real-time response of the system and provides outperformed user interface to monitor the environment.

Keywords: Wireless, Raspberry Pi, IoT, Smart home

ACM Reference Format:
LINXI LI, WENXUAN ZHU, and JIANJUN WEI. 2021. IoT Based Automatic Blind Controller. In *Proceedings of ACM Conference (Conference'17)*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

In today's digital era, technology brings many visible benefits such as provide a smart home and can save users' time and money and reduce the use of heat and energy by turning off appliances automatically when not in use with the help of smart applications that can control the hardware at home remotely such as closing the smart garage door and locking the smart door when leaving the house[10, 18]. People's daily life become more comfortable and easier to see with the development of IoT that has already been widely used in various of fields such as factories and hospitals[6]. Our work and life now are highly dependent on IoT.

Also, we could easily find that some of the smartness with interconnected devices such as thermostat devices, self-guided vacuum cleaners, automated lighting or door entry/security systems could be employed to achieve chore automation.

Blinds is a very widely-used tool for America family to control the indoor light intensity. And in the sunny parts of the world, the indoor light intensity can change dramatically in one day. For example, windows face west will allow much more sunshine into the room in afternoon. Thus, to maintain a stable level of light intensity in room, people may need to adjust the blinds frequently, which can be very annoy since people have to stop what they are doing and do adjustments. Starting from this point, we find that it is important to have a customer focus in providing devices in home automation system to enhance user options and this forms the motivation for this paper. Thus, we plan to use the wireless systems to create an auto blinds controller to set people in the sunny area free from annoyance of altering blinds from time to time.

We plan to use the wireless sensor network to solve this problem, which allows us to collect data with physical sensor and send information through internet and communicate with the user. The overall blueprint for our WSN is that light sensor can constantly collect indoor light intensity and raspberry pi can process the data and give the AWS IoT[2] playing a role of linker between user and raspberry pi can exchange data. Then the user can communicate with the raspberry pi who can control the blinds according to users' instructions.

The article is organized as follows: section 2 specifies the main problem and clarify our goals and requirements. Section 2 explains the whole system design. Section 3 shows all details in implementation. Section 4 is about the experiment we did to test our production. Section 5 is about other work that related to this project. Section 6 will summarize what we learned from the whole process of building WSN. Section 7 concludes all information and discuss what parts we can improve in the future. All materials including videos and source code have been uploaded on the github[1].

2 Goals and Requirements

Based on the overall idea of auto blinds controller, we tried to specify the problem into more concrete goals. The system can be divided into three parts:user interface, AWS IoT, blinds control part. First for the user interface part, our goal is to allow the user to input their desired light intensity according

[1]https://github.com/weijj27/CSE521-wireless-IoT

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

to the current light intensity. Also, users can set a specific time to open or close the blinds. To achieve this essential function, we need to make both raspberry pi and user can send and receive message through AWS IoT. For the blinds control part, we found that the most traditional and common blinds is a kind of blinds that its leaf can rotate horizontally and all leaves can be controlled by a pole. Thus, we decided to only focus on the method to automatically control this kind of blinds.

To achieve our ideas, for the UI part, we choose is React framework to construct the webpage and raspberry pi 3 B+ to process data and control motor, whose input power is 5V/2.5A DC. Light sensor we choose is TSL2591, which has an extremely wide dynamic range: from 1 to 600,000,000 and it allowed voltage range is 3.3-5V[3]. The motor dirver controller is L298N and the power supply for the motor is 12V and 5A[7].

3 Design

In this section, we will discuss our design in detail. The core of our system is the Raspberry Pi and it is used as CPU and could combine all the components together as a whole, which is shown in Figure 1.

In order to control the light intensity in the room, we first need to gain the light data inside. Thus, a light sensor is required to feel and collect the environment data, and we connect the light sensor to the Raspberry Pi so that environment data could be received by the core in real time.

Then, another module L298N is also designed to be connected by the core so that we could control the motor in a much more powerful and precise way. The L298N is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage. In other words, L298N could not only share part of the burden on the Raspberry Pi but also supply additional power source for the motor.

In addition, we have built some interesting algorithms to make the motor work in a much smarter way. Intuitively, we could make it run in both forward direction and backward direction. Through this way, we could open and close the blind just as what we hope. Then, we find the most suitable speed according to the experiments so that we could set how many rounds we should run in each adjustment. With the speed and rounds we set in the program, we could make sure that the difference between the real time light intensity and the threshold we set, which is the light intensity we expect inside the room, could be confined into 10, almost closing to the ideal situation we hope. Since our blinds can change its rotation angle by rotating its controlling stick, we can connect the motor and blinds directly. Once we activate the motor, the controlling stick will rotate with it together in same direction.

After that, the data would be sent to AWS so that we could get the real time data from remote. Thanks to AWS IoT Core, devices and clients could use the MQTT and the MQTT over WebSocket Secure (WSS) protocols to publish and subscribe to messages. AWS IoT manages device communication through a message broker. Devices and clients publish messages to the message broker and also subscribe to messages that the message broker publishes. Messages are identified by an application-defined topic. When the message broker receives a message published by a device or client, it republishes that message to the devices and clients that have subscribed to the message's topic.

The message broker also forwards messages to the AWS IoT rules engine, which can act on the content of the message. However, we choose to put the rules into the message so that users could set the threshold as they expect and would not depends on the AWS IoT rules engine any more.

Moreover, we have designed a website to display the data in real time. The website is both a tool for data visualization and a controller which could do some configurations according to users' preference. Through the website, users could set the light intensity they expect and set the time opening or closing the blind. Also, there is a disable button so that users could close the system and enjoy the light. These kinds of settings would be sent back to the AWS and finally reached the Raspberry Pi. And then, the core would work just as we set.

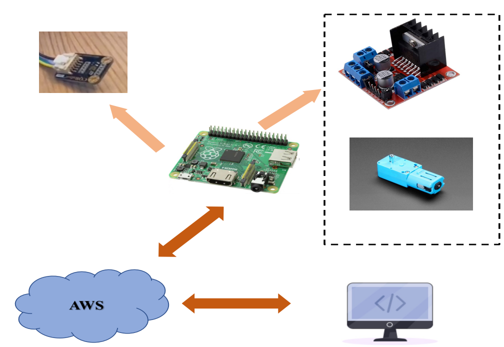

Figure 1. Data flow of our control system.

4 Implementation

In this section, the implementation of our system would be discussed in detail. There are two parts that we need to take into consideration: the user interface and the implementation for the software and hardware.

4.1 Implementation for User Interface

The user interface part is shown as Figure 2. It contains several part. On the left top it shows the current light intensity and the status of the blinds. Then there are three input box that have three different functions. Users can use the first input box to input their desired indoor light intensity value and the set button will send this value to raspberry pi. And the second and third input box are used to set the time to open and close the blinds completely, which just serve as a clock but remind people by the change of light intensity. And the right part is a line chart which can show the change of light intensity with time. The red line is the value collect by the light sensor and the green line is the value set by the user. In ideal situation, the green line and the red line should completely overlap, which means that the indoor light intensity equal to the value user set.

The basic structure of this user interface contains a front-end which is implement by the React framework and a back-end which is implement by the python flask. The method to connect flask and React refers to [1]. The front-end can use POST function to send the value inputted by users to the back-end and use GET information to get the current light intensity from the back-end. On the other hand, the back end can subscribe data from the raspberry pi and publish the user’s needs through MQTT[11]. Every time the back-end receive the information from the Raspberry Pi, it refreshes the data in the front-end shown data, which make sure that user can see the real time light intensity value.

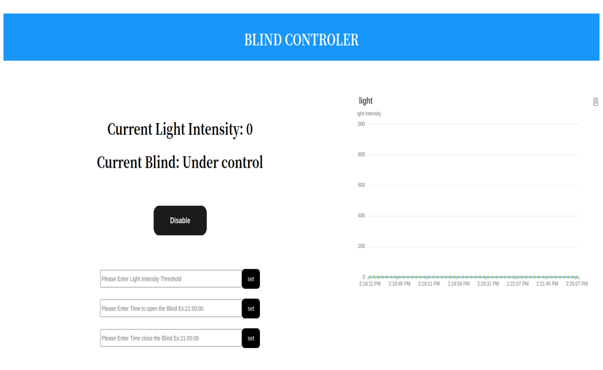

Figure 2. User Interface for Monitor and Control

4.2 Implementation for Software and Hardware

We now present how we implement the software and combine it with the hardware by using Python programming to make the whole system work.

4.2.1 Collect the data from the light sensor.

The light sensor here we use is TSL2591. It is high-sensitivity digital ambient light sensor with 600M:1 wide dynamic range, capable of detecting light intensity up to 88000Lux, I2C interface control, low power consumption, suitable for working under various lighting conditions.

For the first step, we need connect it to the Raspberry Pi so that real time environment parameters could be transferred into our computation core. The specific line connection is shown in Figure 3.

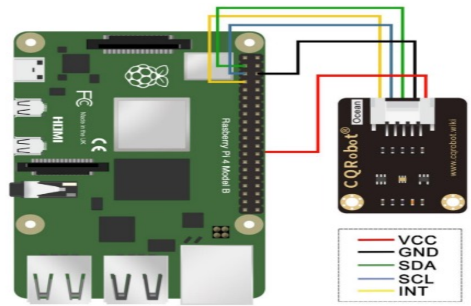

Figure 3. Connect light sensor with Raspberry Pi.[4]

We could see that the red line and black is for VCC and GND respectively. VCC is the higher voltage with respect to GND (ground). VCC is the power input of a device. It may be positive or negative with respect to GND, but most of the world runs on positive voltages. However, GND is normally at zero volts or the zero voltage point for a power supply and circuit[8]. Importance should be put on the connection especially for the VCC and GND and the reason is that once we make a mistake on the positive voltage input position and the negative output position, the Raspberry Pi could not afford the high temperature and would be burned, which is such a terrible catastrophe that we should pay more attention on and try our best to avoid.

At hardware level, SCL (Serial Clock) is the clock signal. SCL is used to synchronize all data transfers over the I2C bus. The clock signal is always generated by the current bus controller. Some peripheral devices may force the clock low at times to delay the controller sending more data, or they may require more time to prepare data before the controller attempts to clock it out. On the other hand, SDA (Serial Data) is the data signal, and it is actually the data line. With the help of these two lines, data could be transferred successfully from the light sensor to our computation core.

For software, we import the waveshare, which is a powerful library designed for read environment parameter using ambient light sensor such as TSL2591. With such a great library, we could get not only the luminescence but also the infrared intensity and the full spectrum. Here we will take the luminescence and use it in our afterword implementation.

After obtaining the real-time data, we would then transfer it to the Raspberry Pi so that it can do some judgment and could be smart enough to understand what we think in our mind.

4.2.2 Data Transfer.

Devices connect to AWS IoT and other services through AWS IoT Core. Through AWS IoT Core, devices send and receive messages using device endpoints that are specific to the user account. Here we choose MQTT as protocol to send and receive messages between our user interface and the AWS and publish and subscribe topic between Raspberry Pi and the AWS[2, 5].

After initializing the MQTT client in our Python script and setting the connection configuration, channel to the AWS core would be created. Then, by reading data we have received from the light sensor, we could send it to the cloud which has the same topic as we published. Receiving data on devices are similar to the previous progress[16].

Besides, we will use JSON format for our message payload, which enables the AWS IoT engine to parse our messages and apply SQL queries to it for examination. Another reason we choose JSON is that it is self-describing. The syntax and hierarchical structure of the JSON strings can in some cases be interpreted by applications that do not already know what data to expect. Since it is easy to learn, easy to read, and easy to understand, we may ave lots of time transforming data from different formats when data flows from our computation core passing through AWS and finally reaching our website. The opposite direction data transportation enjoys the same benefits brought by JSON. Therefore, we could not afford not using it.

4.2.3 Activate the motor when there is a request.

To control the motor, we need a chip called L298N. This chip extends the potential of motor and provides additional power supplication for it. With the help of L298N, we could control the direction and the speed of the motor more precisely. We connect the L298N to the Raspberry Pi through three pins. The detail for the L298N has been shown on Figure 4.

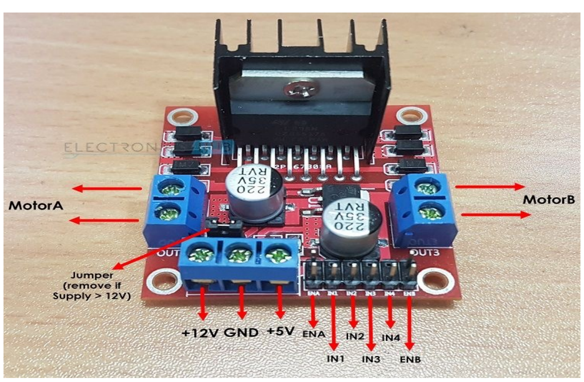

Figure 4. Detail for L298N.

We connect ENA, IN1 and IN2 to 25, 23 and 24 pin of Raspberry Pi. By this way, we are able to activate and control its direction by different output level of pin 23 and pin 24. Raspberry will loop for getting new light data and compare it with the threshold, once there is too much difference between the threshold and light data. The motor will activate to change the angle.

In addition, we take four situations into consideration. "Current light" means the luminosity now inside room and the "Threshold" is the light intensity we set. Here we introduce the parameter "Difference" so that we could release the threshold to a loose extent and tolerate some offset. Usually, we set the "Difference" as 10, which means the luminosity in the room will always lies in[Threshold - Difference, Threshold + Difference]. Moreover, we set the parameter "Location" as how many rounds we could take. The reason for this configuration is to protect the blind from being over twisted. Under the speed of 70, setting the maximum of "Location" as 10 would be safe and sound. We discuss the each of the situation below.

(1) *Current light > Threshold + Difference.* When the current light is much larger than the threshold and it is not after opening time or closing time, we may set the speed as 70. Besides, we may decrease the "Location" by 1 until it equals zero.
(2) *Current light < Threshold - Difference.* When the current light is much smaller than the threshold, and it's not after opening time or closing time, we may still set the speed of the motor into 70. However, we control the motor to run in an opposite direction so that we could open the blind and let the light in. In addition, we may increase the "Location" by 1 until it reaches 10.
(3) *Time we need to open the blind.* When we need to open the blind automatically at the default time, we would keep the motor rotating until it reaches the threshold we set, or it rotates 10 times.
(4) *Time we need to close the blind.* Similar to the previous situation, when we hope to close the blind automatically at the time we set, we would also keep rotating the motor until "Location" turns into zero. Here we do not set the luminosity as the pause condition because there must be light inside the room and this light intensity would be the one that we could not control.

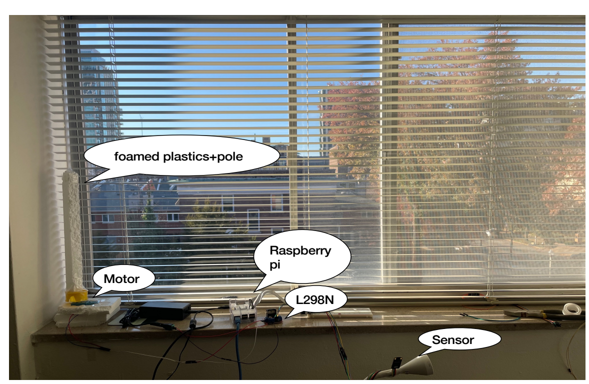

Figure 5. Whole connection of the system.

Speed	10	20	30	40	50
Location	20	19	18	16	15
Speed	60	70	80	90	100
Location	14	10	7	5	3

Table 1. Relationship between speed and Location

4.3 Connection between motor and blinds

The motor we brought is designed for the car model. The motor can easily connect with a little car wheel. What we do to change the rotation of car wheel into the rotation of blinds pole is that we use the glass tape connect the car wheel with a piece of long cuboid foam plastic then we plugged the pole into that cuboid foam plastic. Finally, we use glass tape to fasten the motor with the platform. The high coefficient of friction between the blinds pole and the foam plastic is very high, thus slippery between them almost never happened, which turned out to be a very important property in our whole system. Since the power of motor is not sufficient enough to deal with the slippery problem between pole and the material connect it with the motor. The other option to connect motor with the pole we have been considered is rope, however rope is far less stable than the foam plastic and the ratio to change the work of motor into the movement of pole will be extremely small. Thus, we choose the foam plastic instead of rope. The whole setting has been shown in Figure5.

5 Experiments

We set up the experiment as a whole, which is shown in Figure5. As we mentioned before, we apply our project on the windows blind in our room. By this way, we can do some experiments in a much more explicit way. Here we mainly have 2 experiments. The first one pays more attention on the precision of the system and the second one would put more emphasis on the additional function. Besides, we set up an additional experiment to show the relationship between the rotation speed of the motor and the maximum rounds it could rotates. The results have shown in Table 1.

From the Table 1, we could easily find that too high the speed or too low the speed is neither the best choice since it erases the possibility in the range of the maximum luminosity and the minimum luminosity inside the room. It is also highly likely that rotation in this speed would damage the blind. However, when we set the speed lower than 50, the motor may not have enough power to overcome the resistance of the blind to make it work. Locations over 60 may be a good choice that is suitable for the experience. Here we set the speed as 70 in the following experiments.

For the first experiment, we hope the user to set the threshold and visualize the data with multiple data.

Figure 6. Current Light VS Threshold.

In Figure 6, we set the threshold follow the sequence of 800, 200, 700 respectively. After the threshold changed, the motor will activate to change the current light intensity. As we shown in the graph, the green line is the threshold we set and the red line the current light intensity. Red line can always be close to the green line. We can see some fluctuation of red line, according to our observation, it is because some sudden change of outside environment and in this situation the light sensor can sensitively capture the change and drive the motor to rotate. Thus, we can see that eventually the red line can still back to the safe range.

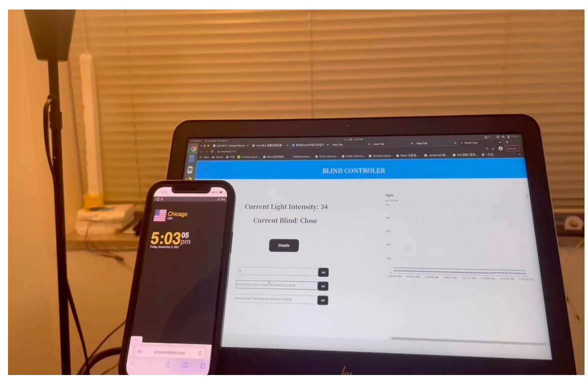

Figure 7. Test for the closing time and opening time

For the second experiment, we hope the function could be worked as an alert. When we setting the closing time and opening time, the blind would be opened to maximum angle or closed the blind to its initial position. In Figure 7, we set up the environment for this experiment. When we set the closing time on 5:03 PM and opening time on 5:10PM, it actually works at exact time. We also provide the videos on the github so that the results could be shown in a much more direct way.

6 Related Works

In[9], the author focuses on the wireless system of smart home. A prototype that has the advantage of maintaining home security by provided the facilities such as turning on and off the lights without having to use the switch and know the lighting conditions has been produced with the combination of software and hardware. With this system, users who are outside the home or work area will get info from their smartphone in real-time so that they could monitor the security condition of their home anywhere. In[14], Yuda has proposed a smart home light-based service oriented Architecture and IoT which is equipped with a token as system security. Tests have been carried out by controlling house lights remotely using a smartphone via the internet network and the results can be controlled properly.

Although there are lots of powerful devices on the market that concerns the smart home, they are expensive and could only be fully used by integrating with other high technology devices[13, 15, 17]. Since it may spend lots of money both on the smart home device with light sensor and other affiliated devices, it is not suitable for most of the middle-class family. In addition, most of the smart home devices are too complicated and could not to be installed easily. Although most the software and hardware are both highly integrated, we still need to do some confused setting or even programming to make it run. Therefore, easy-installation and highly integrated smart home devices are need so that most of users who are not familiar with technology could enjoy the fun of wireless and intelligent devices.

7 Lessons Learned

The smart home system based on Internet of Things has the following characteristics[12]:

- *Ubiquitous service.* The real-time smart home information can be obtained conveniently no matter where the users are.
- *Comprehensive perception.* With various of physical and logical sensors, home devices could be monitored in a comprehensive and real-time way.
- *Conveniently control.* The smart home system can be controlled by the mobile terminal, PC, and many other communication equipment. Meanwhile, the real time data could be shown and visualized through all sorts of interfaces.

In addition, we could find that the opportunity here lies in the fact that many of today's IoT applications are comprised of the same components: for example, the same visualizations on top of a standard workflow engine, deployed on the same lambda architecture-based pattern with the same data type-specific ingestion pipelines, and using the same machine-learning and statistical techniques. Even domain-specific core components like object models, and application logic carry over from application to application. This commonality serves as the basis for reuse and as a foundation on which to build.

Moreover, machine learning method or deep learning method could be used for model development in order to improve the precision of the IoT devices. Also, machine learning can help demystify the hidden patterns in IoT data by analyzing massive volumes of data using sophisticated algorithms. Combined with IoT smart devices, we could provide much more reliable data for training and deploy the model on cloud, edge and device. As a result, such an interesting combination could provide customers diverse services, which would be fun and exciting.

8 Conclusion and Future Work

In conclusion, wireless networks and light sensor combined with Raspberry pi have shown promise for real-time adjustment of angle of the blind according to the exterior environment. This paper presents the lessons and insights learned from experiments based on light conditions. Salient features of our system are easy-installation, automatic control, integration with ordinary existing home infrastructure and data visualization based on the web services. We present important and interesting findings that our system is feasible and reliable, which improves our life experience in a much smart way.

For this project, the next step, we will first focus on the problem of data delay. Now, the data on user interface cannot change with the rotation of motor. We are going to further optimize the whole algorithm in the user interface part to decrease reaction time. What's more, we will try to give users some suggestions for their operations in different situation. For example, in cloudy day, we may remind our user of the biggest light intensity it can ever reach and if users give the value bigger than the upper value of light intensity, we will remind them that you can try to open the light to make the room brighter. We will also build our system as a whole to be a mature product so that it could not only be installed in home but also in hospital with a much wider use. Combined with other smart home applications, we could build a large ecosystem to make our life better.

Acknowledgments

We would like to thank to Chris Donnell for recommending us a higher torque TT motors. This give us about twice the torque at half the speed, improving the accuracy.

References

[1] 2020. https://blog.miguelgrinberg.com/post/how-to-create-a-react--flask-project
[2] 2021. https://docs.aws.amazon.com/iot/index.html
[3] 2021. https://www.adafruit.com/product/1980
[4] 2021. http://www.cqrobot.wiki/index.php/TSL25911FN_Ambient_Light_Sensor_SKU:_CQRTSL25911
[5] 2021. https://aws.amazon.com/iot-core/?nc=sn&loc=2&dn=3
[6] Feras N Hasoon Alattar and Athar Azeez. 2021. Design and Implementation of an Energy Meter System for Optimized Cost using Internet of Things (IOT) Technology. *Applied Computing Journal* (2021), 55–65.
[7] Amazon. 2017. https://www.amazon.com/Qunqi-Controller-Module-Stepper-Arduino/dp/B014KMHSW6

[8] Yasirli Amri and Mukhammad Andri Setiawan. 2018. Improving smart home concept with the internet of things concept using raspberrypi and nodemcu. In *IOP Conference Series: Materials Science and Engineering*, Vol. 325. IOP Publishing, 012021.

[9] Muhammad Akmal Md Asran and Jumadi Abdul Sukor. 2021. Smart Home Wireless System. *Progress in Engineering Application and Technology* 2, 2 (2021), 448–455.

[10] Marc Jayson Baucas, Stephen Andrew Gadsden, and Petros Spachos. 2021. IoT-based Smart Home Device Monitor Using Private Blockchain Technology and Localization. *IEEE Networking Letters* 3, 2 (2021), 52–55.

[11] Smriti Bhatt, Farhan Patwa, and Ravi Sandhu. 2017. Access control model for AWS internet of things. In *International Conference on Network and System Security*. Springer, 721–736.

[12] Gao Chong, Ling Zhihao, and Yuan Yifeng. 2011. The research and implement of smart home system based on internet of things. In *2011 International Conference on Electronics, Communications and Control (ICECC)*. IEEE, 2944–2947.

[13] Ivan Cvitić, Dragan Peraković, Marko Periša, and Brij Gupta. 2021. Ensemble machine learning approach for classification of IoT devices in smart home. *International Journal of Machine Learning and Cybernetics* (2021), 1–24.

[14] Yuda Irawan, Ari Linarta, Anita Febriani, et al. 2021. Smart Home Light Based Service Oriented Architecture and IoT. In *Journal of Physics: Conference Series*, Vol. 1845. IOP Publishing, 012070.

[15] S Pradeep Kumar, Vemuri Richard Ranjan Samson, U Bharath Sai, PLSD Malleswara Rao, and K Kedar Eswar. 2017. Smart health monitoring system of patient through IoT. In *2017 international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*. IEEE, 551–556.

[16] Vrushali Ladekar and Rohin Daruwala. 2019. Indoor air quality monitoring on AWS using MQTT protocol. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 1–6.

[17] Sitalakshmi Venkatraman, Anthony Overmars, and Minh Thong. 2021. Smart Home Automation—Use Cases of a Secure and Integrated Voice-Control System. *Systems* 9, 4 (2021), 77.

[18] Ain Zulaikha Mohd Zaki, Fitri Yakub, Alyani Nadhiya Fakharulrazi, Azizul Azizan, Aizul Nahar Harun, and Zulhasni Abdul Rahim. 2021. Building a Smart Gardening System and Plant Monitoring Using IoT. *Journal of Sustainable Natural Resources* 2, 1 (2021), 1–6.