FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

DESIGN AND IMPLEMENTATION FOR SMART HOME SYSTEM WITH KEYLESS ENTRY BASED ON INTERNET OF THINGS

Taha Abdullah ABABAKER

Master's Thesis

DEPARTMENT OF SOFTWARE ENGINEERING

MAY 2021

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering

Master's Thesis

DESIGN AND IMPLEMENTATION FOR SMART HOME SYSTEM WITH KEYLESS ENTRY BASED ON INTERNET OF THINGS

Author
Taha Abdullah ABABAKER

Supervisor
Prof. Dr. Resul DAŞ

MAY 2021
ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering

Master's Thesis

Title: Design and Implementation for Smart Home System with Keyless Entry Based on Internet of Things

Author: Taha Abdullah ABABAKER

Submission Date: 12 April 2021

Defense Date: 21 May 2021

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

		Signature
Supervisor:	Prof. Dr. Resul DAŞ Firat University, Faculty of Technology	Approved
Member:	Dr. Öğr. Üyesi Yunus SANTUR Firat University, Faculty of Technology	Approved
Member:	Dr.Öğr.Üyesi Ahmet Arif AYDIN İnönü University, Faculty of Engineering	Approved

This thesis was approved by the Administrative Board of the Graduate School on

.... / .... / 20 ....

Signature

Doç. Dr. Kürşat Esat ALYAMAÇ
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Master's Thesis titled “ Design and Implementation for Smart Home System with Keyless Entry Based on Internet of Things” in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Firat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

21 May 2021

Taha Abdullah ABABAKER

PREFACE

Our research is about a smart home system based on the Internet of Things (IoT). Our study investigated the approach of applying Advanced Encryption Standard (AES) for smart home system-based IoT. Also, we have introduced the Blynk application and Ubidots IoT cloud platform in our study to allow the use of IoT. Our system presented in this research is able to monitor and control home appliances from any corner of the world at any time and efficiently utilize the power by properly controlling the appliances.

In fact, I could not have accomplished my current level of success without the strong support and help of my supervisor and chair of the software engineering department Prof. Dr. Resul Daş, and He was respectful, patient, and kind with me. He always has a good outlook, and his criticisms were always constructive and beneficial. He is concerned about his students' well-being and always makes his students feel confident. He is encouraging and supportive, and he is always available and eager to assist his students. I was so grateful and lucky to have him as my supervisor. Also, my parents and my wife, who supported me with love and understanding. Finally, I would like to express my gratitude to everyone who has offered me advice, guidance, and encouragement in the research process. Thank you for your unwavering love, everyone.

Taha Abdullah ABABAKER
ELAZIG, 2021

TABLE OF CONTENTS

ABSTRACT

Design and Implementation for Smart Home System with Keyless Entry Based on Internet of Things

Taha Abdullah ABABAKER

Master's Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences
Department of Software Engineering

May 2021, Page: xi + 59

With the exponential growth of the Internet of Things (IoT) usage and devices and with the increasing interest of users in acquiring these devices to improve comfort and optimize their household chores, it is necessary to develop a robust and secure smart home system that allows users to connect all these devices and manage their information in a flexible way, with low implementation costs. Thus, this thesis target is to present a cost reduced and flexible home control and monitoring system based on the concept of IoT using NodeMCU ESP8266 module and Arduino Uno board with Blynk mobile application and Ubidots IoT platform, accessing and controlling devices and appliances at any time and remotely from a mobile phone, a PC or any other device with an internet connection. In this project, three relay switches, a DHT22 sensor for measuring temperature and humidity, IR sensors for detections, and an ultrasonic sensor for measuring the distance with Blynk server and Ubidots cloud are used. In addition, the system also offers highly secure and encrypted data exchange between the ESP8266 and the server. Therefore, a proper cryptographic algorithm is used, which is the Advanced Encryption Standard (AES) technique to protect the system's valuable data. The designed smart home system can unlock door and control lights which are connected with relay switches. Also, the system monitors house surrounding with a dedicated DHT22 sensor, show the status of the door and window by IR1 and IR2 sensors, and measure the distance of an object by the ultrasonic sensor with live update of data in very short interval.

Keywords: Smart Home, Internet of Things, NodeMCU ESP8266, Arduino Uno, Blynk, Ubidots, Advanced Encryption Standard.

ÖZET

Nesnelerin İnterneti Tabanlı Anahtarsız Girişli Akıllı Ev Sistemi Tasarımı ve Uygulaması

Taha Abdullah ABABAKER

Yüksek Lisans Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı

Mayıs 2021, Sayfa: xi + 59

Nesnelerin interneti (IoT) kullanımının ve cihazlarının katlanarak büyümesi ve bu cihazlara sahip olan kullanıcılarının konforlarını arttırma ve ev işlerini optimize etme konusundaki artan ilgisi ile beraber, kullanıcılara bütün cihazlara bağlanma, esnek bir şekilde bilgilerini yönetme ve düşük uygulama masraflarına sahip olan, sağlam ve güvenli bir akıllı ev sistemi geliştirme ihtiyacı doğmuştur. Bu nedenle, bu tezin amacı, Blynk mobil uygulama ve Ubidots IoT platformu ile NodeMCU ESP8266 modülü ve Arduino Uno kartı kullanarak, IoT kavramı üzerine kurulu, düşük masraflı esnek ev kontrol ve izleme sistemi sunmak ve cep telefonundan, bilgisayardan veya internet bağlantısı olan başka herhangi bir cihazdan herhangi bir zamanda cihazlara uzaktan erişim ve kontrol sağlamaktır. Bu projede, üç röle anahtarı, sıcaklığı ve nem oranını ölçmek için bir DHT22 sensörü, algılamalar için IR sensörleri, ve Blynk sunucu ve Ubidots bulut üzerinden uzaklığı ölçek için bir ultrasonik kullanılmıştır. Ayrıca, sistem ESP8266 ve sunucu arasında yüksek güvenlikli ve şifrelenmiş veri aktarımı da sunmaktadır. Bu nedenle, uygun bir kriptografik algoritması, Gelişmiş Şifreleme Standardı (AES) tekniği ile sistemi koruması için kullanılmıştır. Tasarlanan akıllı ev sistemi röle anahtarı bağlantıları ile kapıları açabilir ve ışıkları kontrol edebilir. Ayrıca, sistem özel DHT22 sensörü ile evin çevresini izler, kapı ve camların durumunu IR1 ve IR2 sensörleri ile gösterir ve çok kısa bir zaman aralığında gerçekleşen canlı veri güncellemeleri ile ultrasonik sensörü kullanarak bir objenin mesafesini ölçebilir.

Anahtar Kelimeler: Akıllı Ev, Nesnelerin İnterneti, NodeMCU ESP8266, Arduino Uno, Blynk, Ubidots, Gelişmiş Şifreleme Standardı.

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABBREVIATIONS

Abbreviations

IOT	: Internet of Things
AES	: Advanced Encryption Standard
IT	: Information Technology
NFC	: Near Field Communication
RFID	: Radio Frequency Identification
Wi-Fi	: Wireless Fidelity
M2M	: Machine to Machine
WSN	: Wireless Sensor Network
PC	: Personal Computer
TBSA	: Triangle Based Security Algorithm
IFTTT	: If This Then That
API	: Application Programming Interface
OS	: Operating System
HTTPS	: Hypertext Transfer Protocol Secure
SoC	: System on a Chip
SSL	: Secure Sockets Layer
LED	: Light Emitting Diode
PWM	: Pulse Width Modulation
USB	: Universal Serial Bus
IDE	: Integrated Development Environment
SDK	: Software Development Kit
TCP	: Transmission Control Protocol
IP	: Internet Protocol
GPIO	: General-purpose input/output
AC	: Alternating Current
COM	: Common terminal
NO	: Normally Open Terminal
NC	: Normally Closed Terminal
HMI	: Human Machine Interface
IOS	: iPhone operating system
NIST	: National Institute of Standards and Technology
UART	: Universal Asynchronous Receiver Transmitter
JSON	: JavaScript Object Notation
IC	: Integrated Circuit
SSID	: Service Set Identifier
UART	: Universal Asynchronous Receiver Transmitter
NodeMCU	: Node Microcontroller Unit

1. INTRODUCTION

In recent decades, with the development of technology, the world is becoming closer to everyone, and the future of manually operated processes is heading towards automatic systems for greater interests. A smart home is a trend of technology and automation development that helps individuals to monitor and control the smartness of electronic equipment as well as automatically from or beyond the specific home, making life simpler, safer, more convenient, and often automated.

These days, with the accessibility of the internet, technology has opened a new door called the Internet of Things (IoT), which refers to the networking of physical machines equipped with sensors, actuators, electronics, smartphone applications, and connectivity to the network, allowing the system to capture, monitor and share data. The eventual advantage of IoT would be immense, and our lives are simpler than ever, as we see soon [1, 2]. The IoT seeks to improve the usefulness of the first edition of the internet to make it more usable. With IoT, users can exchange knowledge found in databases created by humans as well as data given by objects in the real world [3]. We may define the IoT as the link between the design objects and the internet and to each other, using various intelligent technology to build a robust computational smart ecosystem. It can also be described as integrating embedded intelligence in individual objects that can notice changes in their physical state [4].

The general concept of IoT is that computers, sensors, and objects are communicating and processing data, so IoT is a modern technological framework mixed with a variety of IT technologies. The IoT integrates multiple applications into a semi-autonomous network. It links the individual devices to the network and each other. There are also network controller systems (software and services) that serve as the brains of the data processing system by evaluating and using connected electronic data gathered to make decisions and execute actions from the same or other devices [5]. The main aim of IoT is to allow us to locate, identify, access, and monitor items individually at any time and everywhere by using the internet [6]. The integrated device networks will offer substantial life, technical and financial benefits to a wide range of clever and autonomous applications and services [7]. There are currently many IoT real-world applications available, and this includes waste management, urban noise charts, smart parking, smart lighting, etc. and also the NFC, Wi-Fi, Radio Frequency Identification System (RFID), Bluetooth, and so on are the technologies used by IoT [8-10].

Smart ecosystem uses a wealth of tiny cryptographic node combinations to classify and provide users with customized resources when communicating and sharing knowledge with the environment [11]. IoT technologies can be used to build smart homes that deliver knowledge, convenience, and improved quality of life. A smart home can be described as a home that is automated with the use of IoT technology and capable of responding to residents' needs, providing them with convenience, protection, security, and entertainment [12, 13]. In the future, IoT will expect major domestic and market applications, enhancing the quality of life and the global economy. With IoT, anytime and anywhere in the world, it is possible to remotely manage and monitor the electrical equipment built in your smart home; for instance, it would allow smart TVs, air conditioning systems, open garage automatically, and other appliances to be operated inside the house. Smart homes are made up of smart sensors and control systems. With the aid of the internet, everything is linked. Basic home automation utilizes timers and clocks to facilitate desired operations, but smart home technologies can perform more complicated operations and activate systems based on input from other devices [14]. Smart homes are fitted with sophisticated automatic systems for different preprogrammed operations and activities such as temperature management, lighting, multimedia, window and door operations, etc. Often referred to as environmental knowledge, the smart home ecosystem is responsive and flexible to current human and social requirements [15, 16]. The smart home is a very exciting sector with diverse advantages, such as improved convenience, better security and safety, more reasonable use of energy and other services, thereby leading to substantial savings [17].

Bringing IoT technologies to our homes leads to new security issues because smart homes based on IoT need high-level security standards. After all, sensitive and private information is accessible in the home environment. Modern devices present both benefits and threats, and an IoT-based smart home is particularly vulnerable to internet attacks. The intruder has the ability to breach the privacy of the customer, steal sensitive information, and track them inside the home if a smart home or smart device has been compromised, and thus necessary precautions need to be taken [18-20]. There is already a growth in the use of IoT software and the online connection of components. As more devices are available from the internet, as the amount of potential attack vectors grows, the security consequences of these devices become more important. The smart home is one of the most significant IoT applications. The smart home appears, positioning itself as the most noteworthy IoT application on a single calculated platform [21, 22]. Users usually have to adjust the home appliance on their own manually. Nowadays, people at least own a computer or a smartphone.

This research aims to design and implement an effective IoT-based smart home system that provides easy access, high security, and control. The research would also provide IoT systems with privacy and security, aiming to strengthen the system and secure the data when linking it to the internet, making it more robust, innovative, and effective. This thesis is a smart home system based on an IoT project that monitors home environment sensors and controls loads with an internet connection via a Wi-Fi connection. A smartphone connected to the internet with Blynk application and computer connected to the internet with Ubidots IoT platform as a control panel, also Arduino

Uno board used to transfer sensor data to NodeMCU ESP8266, in another side NodeMCU ESP8266 module as a controller that receives control commands via Wi-Fi signal. NodeMCU kit is built with an ESP8266 Wi-Fi receiver that able to process and analyze Wi-Fi signal to input the microcontroller. The Wi-Fi receiver and microcontroller are built in one kit to be used as an IoT project. It is called NodeMCU. An IoT-based smart home device is especially vulnerable and insecure to internet attacks when connected to the internet. To protect original data from unauthorized access, we have used the Advanced Encryption Standard (AES) algorithm to encrypt and decrypt sensitive data that need to protect from intruders. AES is one of the worldwide most popular and frequently used symmetric cipher block algorithms [23]. The basic form of this algorithm is used for encrypting and decrypting sensitive information in software and hardware around the world. When encrypting by the AES algorithm, it is tough for hackers to get the actual data. Compared to other algorithms, AES can provide substantially more security [24].

1.1. Motivations

In recent years, the smart home has become increasingly popular. All is now powered by IoT ever-evolving technology, and smart home solutions have become smarter and more sophisticated. To improve the quality of lifestyle, devices must be fully automated, requiring no user interaction in any way. This allows the end-user to communicate with the appliances without difficulty because the appliances learn and react to the user's needs without physically pushing a button. Wired sensor systems are more complex to manage and require a significant amount of wiring to connect the sensors in various locations. One of the major benefits of automating home appliances is that it saves energy. As a consequence, the consumer must be kept informed of the energy usage of automatic devices.

In my nation, Iraq, smart homes are uncommon. Unfamiliarity with them, a lack of use, prices, and a failure to recognize the utility of such systems are all potential reasons. The internet and smartphones, in particular, are progressing at a breakneck pace these days. It is impossible to prevent the use of such technologies and equipment. In a country like Iraq, where people's lives are challenging and stressful, providing effective energy savings and comfort at low prices while maintaining efficiency is crucial. Such houses are critical for our country's stability, economy, and efficient use of electricity. The latest smart homes and IoT articles have created general attention and concern, and these developments have major security consequences in these technologies. Like all other communications systems, the need for protection in smart homes is the same and even stronger than the need for security to ensure that information is not compromised, changed, or refused access. It is clear that in typical households, intruders will only rob or invade their homes if they physically exist near their homes. However, an intruder or an attacker can reach the home from anywhere in the world at any time by connecting a home to the internet via an internet connection. With IoT technologies, attackers can breach consumer privacy, steal private information, and track the occupants in our homes if the smart system is hacked successfully [25].

1.2. Purpose of Thesis

This thesis purposes of creating a prototype for an IoT-based smart home system that allows users to use smartphones or computers to monitor and control electrical equipment such as doors, lamps, and other devices from anywhere and at any time. Furthermore, to protect the system's data from hackers and intruders while connecting with the internet. The following points are the thesis objectives:

- To design a flexible and easy-to-use smart home based on IoT using NodeMCU and Arduino Uno.
- To Provide communication mechanism with Blynk application over Blynk server between smartphone and NodeMCU module.
- To provide communication mechanism with Ubidots IoT platform web application via Ubidots cloud service between PC and NodeMCU ESP8266 kit.
- To develop software for Arduino Uno and NodeMCU microcontrollers to control and monitor appliances remotely from a smartphone application and web application.
- To provide security and protection to the system’s data between NodeMCU and the internet server using the AES technique to encrypt and decrypt data.

Thus, we intend to make it easier for the user to switch ON/OFF the appliance from a smartphone or a computer. There may be a variety of issues as people leave their homes, including save power from electrical wasting when forgetting to switch OFF the electrical devices, saving comfort and effort for people with disabilities, protecting the house from electric stuns or fires causing, by monitoring the home appliances, and the inability of the user to control household devices more accurately and remotely.

1.3. Thesis Organization

This thesis is generally divided into five chapters that will explain the entire process and flow of this project. The thesis is included the introduction, literature review, system design and implementation, application result, and conclusion.

- The second chapter focuses on the literature review and discusses all of the details to which this project is referred. The literature review also implies that, based on the references, each part or process, theory, and fact used in this project is analyzed.

Thus, in this research, the literature review will highlight the interpretation and explanation of IoT-based smart home systems.

- In the third chapter, the system design and implementation are introduced and explained. All the components used in this project are explained. The procedure steps will be explained. This chapter is an important side because it is a process for the successful implementation of the project. The problem may occur during the implementation of the work if the chapter is not well organized.
- In chapter four, the results of the application obtained will be documented and displayed in this chapter. All application results will be included in this chapter. Only after obtaining the desired result of the project can this chapter be completed.
- Finally, the fifth chapter is the project conclusion. This chapter would finalize the result and goal of the project and describe it in depth. Besides, to enhance the project, recommendations on the project will also be included.

A summary of the project to be introduced is given in this chapter of the thesis. This chapter includes a project briefing from the outset to the conclusion of the project. The overview of the project is included motivation, the purpose of the thesis, and thesis organization. The aim is to implement this project based on the issue statement studied and to pursue the project procedure.

2. LITERATURE REVIEW

Home is an environment in which people work and spend much of their time. They are involved in social sports, individual decision-making, well-being, reading, writing, and various other household activities at this time. Consequently, everyone needs their homes to be both secure, intelligent, and human attention is paid to new core problems with the growth of the knowledge age and the increase of the quality of living [26]. Issues such as how household equipment and environmental criteria can be safely used and tracked or managed with remote devices have been called to the attention of many investigators and companies to enhance the standard and quality of life [27, 28].

To thoroughly understand the concept of IoT, we have examined a variety of studies, research, lectures, conference papers, and project reports. We have also reviewed many IoT-based projects that have already been planned and built. In different forms and strategies, the smart home project was applied. The various sensors, microcontrollers, and actuators were used in their designs to control and monitor home appliances and protect the home in different ways.

Hassan et al. (2015) proposed a home control and monitoring system using a simple mobile application. In their study, they demonstrated how lamps, doors, audio, and more could be automated. It monitors and controls anything with a smartphone, including the house itself. The research discusses the concept and implementation of a home automation device using a smartphone and Raspberry Pi to protect our homes. The suggested solution for home security relies on the modern incorporation of motion sensors. Raspberry Pi controls and monitors motion detectors and remote sensing and monitoring, and ultimately performs activities on home appliances such as switch ON/OFF microwave or TV [29].

Daş and Tuna (2015) introduced the machine to machine (M2M) communication-based control of an air conditioner to illustrate M2M usability and realistic implementation. In the proposed application, the SIP server, a Raspberry Pi 2 board with temperature and humidity sensors, and a smart air conditioner were the three main components. The air conditioner automatically changes itself using temperature values given by a group of sensors. The application was a clear instance of howM2M innovations have begun to enhance our everyday lives. The application source code was accessible to readers upon request [30].

Akhtar (2017) presented an IoT-based autonomous office alarm and access control system. When the last individual leaves the premises in the suggested system, the system can automatically turn on the alarm system. The proposed system, however, helps to solve the problems experienced by manual alarm systems. Their system is like every other warning system but is operated by the current wave of IoT computing. In their project, the IoT-based system was cost-effective and offered far more flexibility than the regular alarm system. To prove the viability of the method, a prototype was also built and tested in a real environment [31].

Mohammed and Duman (2017) implemented a smart house application using Wireless Sensor Network (WSN). They considered smart home security to be applied appropriately. They suggested two different ways to connect to the internet to be used and observed from a distance. This can be achieved through the web by using smartphones, laptops, PCs, and tablets after entering their special IP and connecting them via WSN. They also proposed using more than one type of wireless in the same project, such as IR, Bluetooth, and Wi-Fi [32].

Pirbhulal et al. (2017) proposed a secured IoT-based home automation application using WSNs. A security algorithm, namely TBSA, was developed in their study based on a simple and efficient key generation procedure. Because of the use of a low-power Wi-Fi module, the proposed IoT combines low-power Wi-Fi and the proposed TBSA with the internet in WSNs to provide additional benefits of improved coverage and support for many sensor nodes. The use of the proposed TBSA algorithm also consumes less processing time for data encryption. Besides, their study has verified that the IoT platform developed satisfies all primary security needs, including the security of the network [33].

Mandal et al. (2018) suggested the implementation of an IoT-based home automation system using real-time sensors. The study suggests that temperature and humidity DHT11 sensors, gas sensors, NodeMCU ESP8266, Arduino Uno, and Arduino Nano microcontrollers would be required for the system. The system went through multiple phases to evaluate the decision after the retrieval of the data from sensors. Their research recommended a two-step approach that involved the stage of pre-processing and the stage of decision. The pre-processing stage includes collecting and processing data, while the decision stage involves the management of home components and voice control operations using IFTTT. Also, decisions on temperature and humidity data are held on the cloud platform for further testing purposes [34].

Almosawi and Djupsjo (2018) proposed IoT security applied on a smart door lock application. Their thesis describes creating an IoT framework focused on the digitalization of a smart door lock to make it connected to the internet and identify employees who work in the office. They were mainly concerned with security aspects, described in general the typical challenges of security in IoT systems, and summarized to create a functional and safe product from scratch. The microcontroller was chosen for the project, and the test environment was designed to experiment and create security breaches. For the API being built and even for the android application, architectural designs were selected. The Azure active directory multi-master database and its significance for the defense of an important security violation were defined in detail. In their project, the transmission protocol using Bluetooth beacons was served using a new technique called Eddy stone. The last step of the project was to complete android application development and ensure the communication of all subsystems to each other and ensure that the IoT's device flows are usable and stable [35].

Alonso et al. (2018) presented an IoT solution for data sensing in a smart campus using smartphone sensors. They discussed the android operating system (OS) in their paper as a means of controlling and operating various sensors embedded inside smartphones that are widely used in participatory sensing-oriented applications such as urban traffic navigation and smart campus mobility. So many processes in the IoT architecture were tested for the potential uses of the Android OS in which participatory sensing was carried out. For analysis and visualization, data transmission on a cloud platform was important in the project architecture. The paper also proposed designing and implementing an IoT system solution for participatory sensing applications in the smart campus. Their approach uses mobile sensors such as IoT device-embedded sensors and the android API to gather the information from them (e.g., GPS data). To further analyze and display data, the data sensing was stored in the local database and then transferred to the Ubidots cloud servers via HTTPS [36].

Qiang et al. (2018) presented NodeMCU-based low-cost smart home node design. The proposed design was a cost-effective node-based Wi-Fi network solution. The goal of the project was to solve the issue of smart home popularity due to high real-life costs and is suitable for applications in the smart home sector. This approach reduces cost, utilizes SoC, and has very few peripheral circuits. The NodeMCU was used to directly capture sensor data, save a separate main control chip, and further reduce cost. Different hardware interfaces have been established for the expansion of various sensor types. The data were imported onto a back-end server, and the hardware was configured for various usable nodes. The data was aggregated between different nodes based on software processing and submitted to the cloud platform for viewing to achieve remote control and access [37].

Mahindar et al. (2018) suggested an IoT-based home appliances control system using NodeMCU and Blynk server. A framework has been proposed that monitors home appliances worldwide and regulates the use of electricity. Blynk technology was used to read data from devices in the household and operate home appliances based on it. The applications could not read sensor data continuously to engage in the regular scheduling of users. So it sends an emergency alert on the mobile application [38].

Singh et al. (2018) proposed an IoT-based smart home automation system using a sensor node. Their paper developed the prototype and tested it by using different home electrical equipment and automating them with no or limited user interaction. By using IoT technology, they achieved the development of the smart home. Based on studies, they have found that a smart home for a better and greener future is cost-effective, versatile, and energy-effective [39].

Durani et al. (2018) proposed a smart automated home application using IoT with the Blynk application. Their paper was mostly about solving the daily problems facing people around the world in which frequent power cuts, unregulated urbanization, the scarcity of workers in agriculture are apparent. Their prototypical system was used for home security in real-time, automation, remote system monitoring, and control. Their system implementation provided an intelligent, convenient, energy-efficient home automation system. It also allows older people and people with a disability to monitor their home devices more safely and easily [40].

Majhi (2019) presented an IoT-based home automation system. The key goal of this project was to incorporate IoT with the sensor, Blynk framework, and Raspberry pi. The expected outcomes were obtained, and it can be assumed that the project was completed. Furthermore, the project covers the primary feature of so-called perfect smart home automation, such as calculating air temperature, light control, etc. Even with high voltage, but with good safeguards and protection, the powerful relay allowed them to work [41].

Imam (2019) presented a simple smart home based on IoT using NodeMCU and Blynk. The study suggested a solution that can save energy by allowing Blynk application to monitor the building's lighting with its smart home. The lamps are operated by switches distributed across the building at the same time as a certain electrical installation and by the Blynk application. This research produced a basic smart home design or an easy and low-cost way to usually manage loads over Wi-Fi [42].

Wai et al. (2019) proposed an IoT-based healthcare monitoring system using NodeMCU and Arduino UNO. The suggested system was capable of monitoring, diagnosing, and advising patients continuously. It decreases time by the usage of the system. Based on their results, the doctor examined the patient from anywhere and at any time. Doctors or nurses were allowed to conveniently use the computer to search and save in the database. Moreover, with NodeMCU and Ubidots IoT platforms, doctors could use mobile phones and be deployed in a global network [43].

Aldawira et al. (2019) suggested a security monitoring system based on IoT technology. The system was based on ESP32. The suggested scheme was able to monitor and close the door remotely, receiving a warning when motion is detected near the door, granting access to the door to trusted individuals to operate the door, displaying the door access history log and user access, receiving notice that the door is still open after the time limit has expired, and turning on the alarm should the door be opened violently. The simulation outcome suggested that the system would detect the movement exactly at 1,6 meters when the sensor is placed at the height of 48,5 cm with the path moving down, but that if the movement is more significant than 1,6 meters, the detection will be less precise. Results obtained have revealed that they are appropriately secured by Secure Sockets Layer (SSL) encryption when messages are published or signed between mobile phones and door locks, so messages sent are safe from hackers. Other individuals will not interpret or modify the message because hackers do not have the key to decrypt messages [44].

Bohara et al. (2020) proposed an IoT-based smart home using the Blynk framework. The project addressed in this paper was designed to address significant problems facing Nepalese people in their everyday lives. It was intended for mobile control and monitoring of appliances using Wi-Fi as a networking protocol and Raspberry Pi as a private server. The NodeMCU microcontroller, which acts as the gateway to the internet, links all appliances and sensors to the internet. Even if the consumer goes offline, the system is programmed to turn to automatically monitor the devices according to the reading of the sensors. Furthermore, the data was logged into the server for possible data mining [45].

Surriani et al. (2020) presented a design of power monitoring application. The researchers were interested in designing an android-based IoT application that can control electricity consumption and track electricity consumption data history. The monitoring application was able to measure the power used every hour as well as the approximate costs. The IoT, electronic design, ACS712 current sensor, and NodeMCU 8260 were used to track kWh meters. Based on the testing, it was concluded that the architecture of the kWh meter monitoring system with the current sensor ACS712 using the Ubidots IoT platform has successfully controlled IoT monitoring of electrical power consumption, which is 0.35% mistaken [46].

Table 2.1 illustrates a comparative summary for the previous studies that are mentioned in this chapter of the thesis.

Table 2.1. Comparative summary for the previous studies

Ref.	Purpose	Technology	Application	Result
[29]	Provide essential security to homes and associate control operations	Raspberry Pi	Home control and monitoring system	The smart home system successfully designed and implemented
[30]	Design a smart air conditioning application for smart home services	Raspberry Pi, M2M	Control of an air conditioner	Successfully design a smart air conditioner
[31]	To design and implement a cost-effective IoT based autonomous alarm and access control system	Ethernet module, IoT	Control the states of the alarm, set an alarm, lock, and unlock the door, monitor the state of the door	The aim of developing a cost-effective IoT based autonomous alarm and access control system is accomplished

[32]	To give ease in everyday life, provide comfort, authentication and security, and additionally spare power and human endeavor	Arduino Mega 2560 Arduino UNO R3 GSM GPRS ESP8266 Ethernet Shield Router	Controlling and indicating electrical and electronic devices in houses and buildings	The implementation of the system was managed successfully, the system can work with all its pieces through the internet properly
[33]	To provide secure data transmission among several associated sensor nodes in the network	Intel Galileo Board, Wi-Fi Module (N-2320), IoT	Secure data transmission, facilitate energy-efficient data encryption	The experimental results showed that the proposed TBSA algorithm consumed less energy in comparison with some existing methods
[34]	To implement an IoT based home automation system through data processing	Arduino Uno, NodeMCU ESP8266 and Arduino Nano, IoT	Home control and monitoring system detect the presence of harmful gas and measure temperature and humidity	Successfully can control the proposed system remotely by android smart phone via internet
[35]	To study and evaluate a suitable set to develop a smart door lock which is intended to offer high security, easy access, and control	IoT, Microcontroller, Bluetooth beacons	Smart door lock	A fully functional smart door application was accomplished, and the result of security implementation of the product was an overall, more robust
[36]	To explores the Android OS as a medium to manage and operate different sensors embedded within smartphones	IoT	Extract the data from sensors, data in the local database, Transfer data via HTTPS to Ubidots	The Android application completed successfully all the tasks that were programmed
[37]	To solve the problem of the popularity of smart homes due to high costs in real life	NodeMCU ESP8266	Smart home control system	The system worked normally, and the monitoring data can be reflected in the monitoring interface in real-time
[38]	To optimize performance and saves unnecessary wastage of power	NodeMCU	Home control and monitor	Successfully proposed a user-friendly system that can be used for benefiting the mass. Also, the cost of the system is within reach

[39]	To develop a new solution which controls some home appliances	NodeMCU ESP8266, IoT Arduino UNO,	Provides information about the energy consumed, check the level of gas detects the human object,	The system achieved the development of smart home by using the IoT
[40]	To present a small IoT system designed and created by utilizing a WLAN network based on Node MCU ESP8266	NodeMCU ESP8266, IoT	Real-time home security, automation, monitoring, and controlling of remote systems	Successfully implemented an intelligent, comfortable, and energy-efficient system
[41]	To present a cost reduced and flexible home control and monitoring system	Raspberry Pi, IoT	Accessing and controlling devices and appliances remotely	The projects achieved desired results and can be said to be done successfully
[42]	To presents a simple prototype of smart home, or the easy way and low cost to control loads by Wi-Fi connection generally	NodeMCU ESP8266, IoT	Home control and monitoring system	Based on the results of analysis of all data obtained by testing the smart home, the system accomplished successfully
[43]	To constantly monitor the patient's physiological parameters	NodeMCU ESP8266, IoT Arduino UNO,	Healthcare monitoring system	The system achieved the desired result
[44]	To monitor the status of the door, controlling the door, and increasing security in a house	ESP32, IoT	Door security system	The motion detection sensor can detect movement accurately up to 1,6 meters ahead, and messages published encrypted properly
[45]	To switch to automated state controlling the appliances automatically as per the sensors' readings	IoT, NodeMCU, Raspberry Pi	Home control and monitoring system	The project was designed and implemented successfully
[46]	To design a system to help reduce the energy consumption	Arduino, IoT	Monitor electricity consumption and trace the history of electricity consumption data	The system successfully worked to control the monitoring of the amount of electrical energy usage through IoT

Our work	To create a prototype for an IoT-based smart home system to monitor and control electrical equipment and to protect the system's data from hackers and intruders while connecting with the internet	Arduino Uno, NodeMCU ESP8266, IoT	Home control and monitor remotely from a smartphone application and web application, Door unlock, Door status, Window status, system encryption, and decryption	Our work accomplished to designs a smart home system that enables homeowners to control and monitor appliances remotely, and also the data of the system was encrypted and decrypted properly

Based on Table 2.1, we can analyze that every author comes with different technology and concept for the smart home system. With these different technologies, different designs, different algorithms, flowcharts were also implemented. The common purpose of every author's work was to smoothen the life of homeowners, old aged and handicapped people, to save electricity, to give ease in everyday life, provide comfort, authentication, and security, a cost-reduced and flexible home control, and monitoring system, and additionally spare power and human endeavor. The technologies were used are Arduino, NodeMCU ESP8266, Raspberry Pi, ESP32, Ethernet Shielded, etc. As compared to the above smart home system applications, every system has its feature, mostly providing home control and monitoring system, door security system, healthcare monitoring system, and so on. Some of them provide automation, and some provide security, and some provide both services, automation, and security. As a result, they designed and implemented the smart home systems successfully and achieved the desired results. Our IoT -based smart home system can be accessed remotely from anywhere. A laptop, computer, or smart phones can be used as a GUI to give commands to the system. All sensors and loads are connected to the central system, so easily accessible and manageable. With the AES encryption algorithm, the system is more comfortable, protected, and more secure from hackers and intruders while connecting with the internet.

3. SYSTEM DESIGN AND IMPLEMENTATION

In this chapter, the system design, as well as the implementation procedures, are undertaken were listed in detail.

3.1. System Design

To start the project, let's get an idea of all the materials we are using for the project. There are two key parts of the system: hardware design and software design. The hardware is built by arranging microcontrollers, sensors, and actuators, while programming that is written and uploaded to the microcontroller requires software design. According to the design needs and desires, the system block diagram shows in Figure 3.1 is made. Our system block diagram shows how all the parts of the system are connected. The system is purposed to enhance security, adaptability, and effectiveness. The framework is planned to give ease in everyday life and additionally spares power and human endeavor.

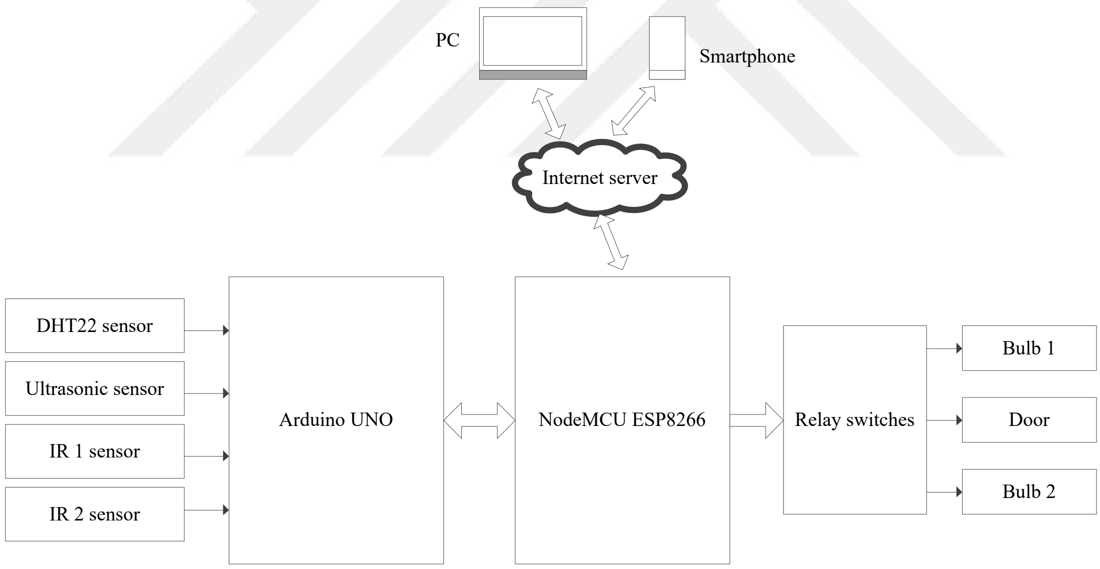

Figure 3.1. Block diagram of the system

The system allows users to remotely control their home appliances from anywhere and anytime using smartphones or PCs. Using the mobile application and IoT platform web application, the user can control his/her appliances through the internet and easily monitor the parameters of his/her household environment by reading sensor information and controlling the appliances in his/her mobile application and IoT platform webpage. Figure 3.2 shows the entire flowchart that provides the complete operation of the method.

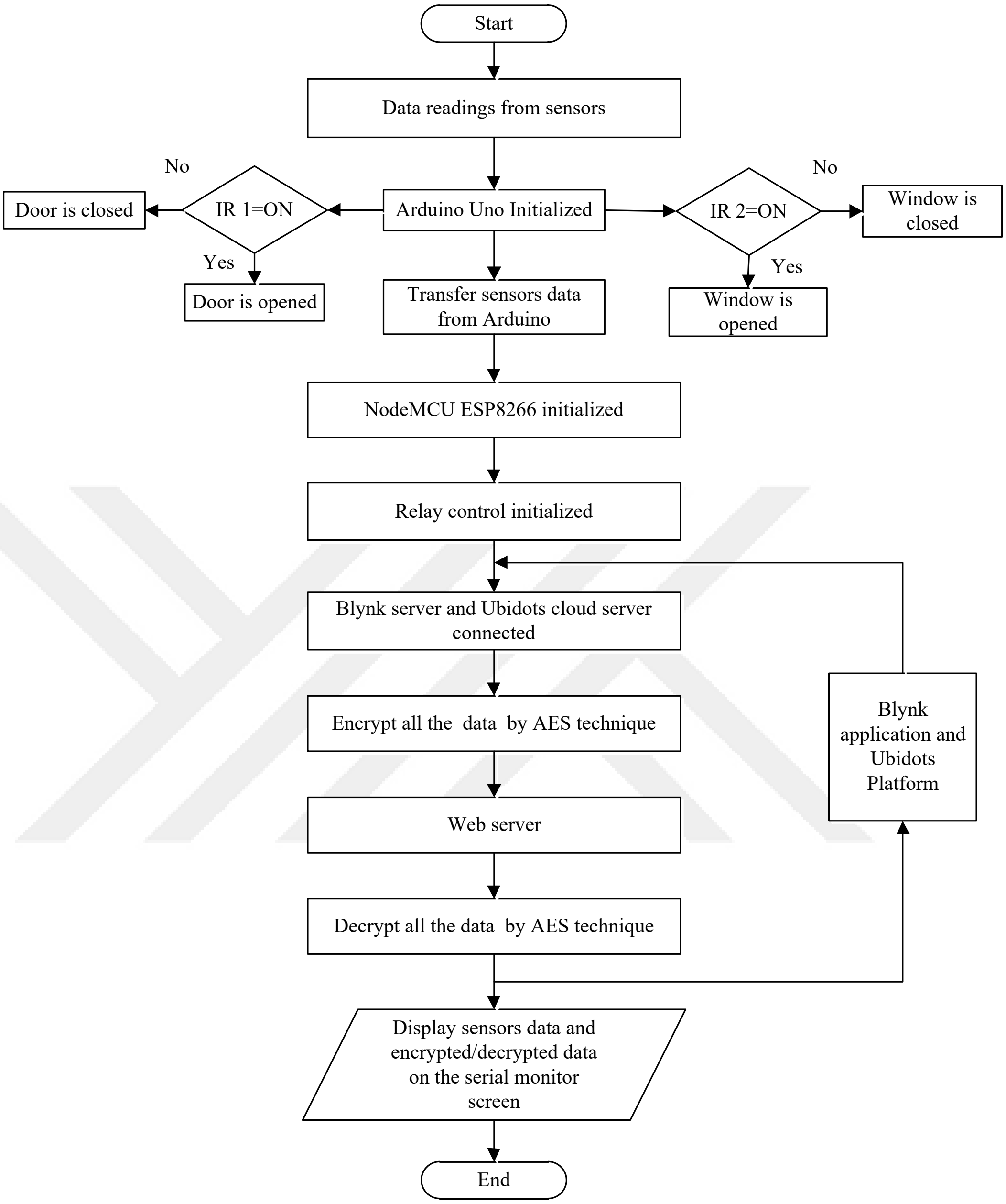

Figure 3.2. Flowchart diagram of the performed system

The schematic diagram of the system in Figure 3.3 explains and shows how all the project parts which are Arduino Uno, NodeMCU ESP8266, DHT22 digital temperature and humidity sensor, HC-SR04 ultrasonic distance sensor, IR infrared obstacle avoidance sensor, electric door lock solenoid, bulbs, and 5V Relay switches are wired together.

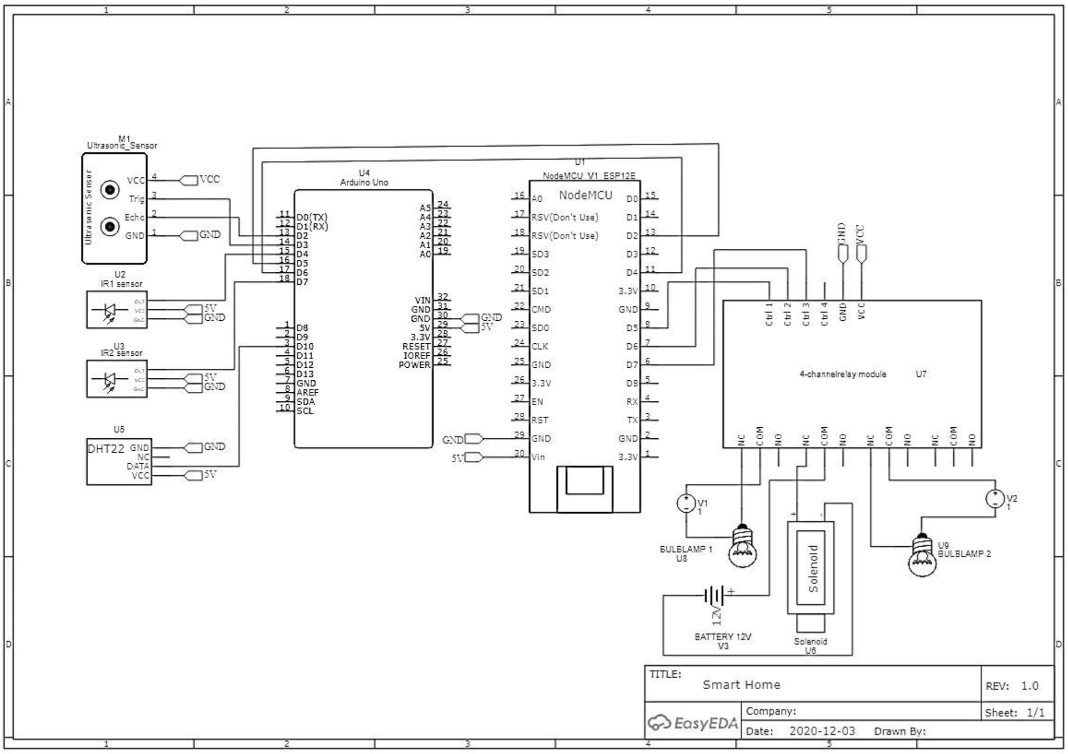

Figure 3.3. Schematic diagram of the system

3.1.1. Hardware Design

It is necessary to know all the specification details of the hardware. The requirements and information used in the system regarding the different components and functional units are defined as follows:

Arduino Uno

Arduino is an open-source platform for electronics focused on hardware and software. Arduino boards can read inputs-light from a sensor, finger from a button, or message from Twitter- and turn it on output-activate a motor, switch on an LED, or perform other actions. So, the Arduino Uno is a microcontroller module based on ATmega328. It has 14 digital input/output pins, 6 of which can be used as PWM outputs, six analog inputs, a ceramic 16 MHz resonator, a USB interface, a power jack, a reset button, and an ICSP header. It contains all the necessary components for programming the microcontroller; simply link it to a computer with a USB cable, or turn it on to operate with an AC adapter or a battery. The Uno is different from all other boards in that it does not use an FTDI USB to serial chip. Alternatively, it features the Atmega16U2 programmed as a USB-to - serial converter (Atmega8U2 up to version R2) [47, 48]. Figure 3.4 shows the Arduino UNO board. "Uno" means one in Italian and reflects Arduino IDE release 1.0. The Uno board and version 1.0 of Arduino Software (IDE) are the reference versions of Arduino.

Figure 3.4. Arduino UNO board

NodeMCU ESP8266

NodeMCU is an IoT open-source platform. It contains both the firmware from the ESP8266 from Espressif and the hardware which is based on the ESP12 board. By design, the term "NodeMCU" applies to firmware rather than dev kits. The firmware uses the language of the Lua script. It is based on the eLua project and designed for ESP8266 on the Espressif Non-OS SDK. The module utilizes several open source technologies, including lua-cjson and spiffs [49, 50]. Figure 3.5 shows the NodeMCU module, and Figure 3.6 illustrates the pinouts diagram of NodeMCU.

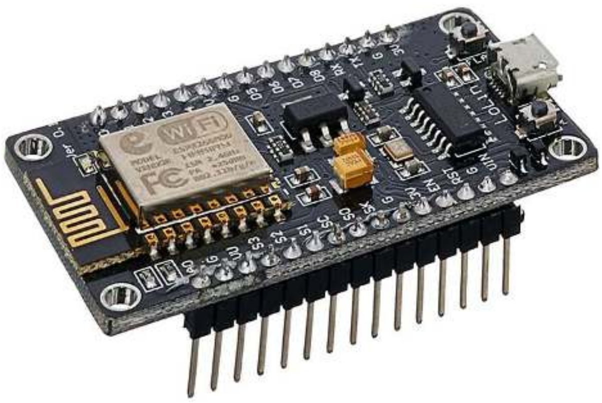

Figure 3.5. NodeMCU ESP8266 module

NodeMCU was developed shortly after the release of the ESP8266. The Espressif device started generating ESP8266 on 30 December 2013. The ESP8266 is a Wi-Fi SoC commonly used in IoT applications combined with a Tensilica Xtensa LX106 core. The ESP8266 has a complete TCP/IP stack and microcontroller capability. This module gives access to the General Purpose

Input/output (GPIO) subsystem. All access is based on the I/O index number of the NodeMCU dev kits.

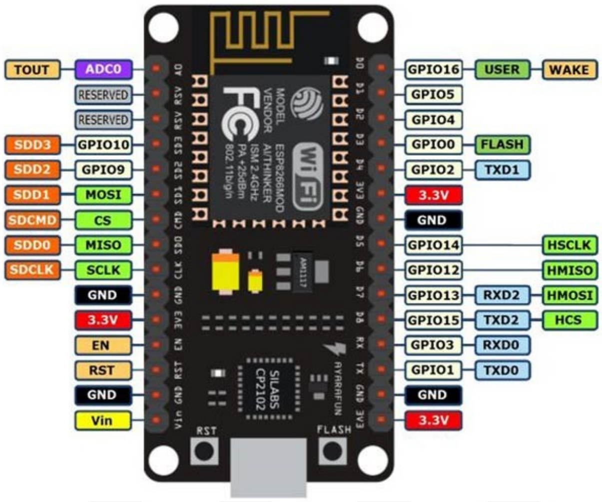

Figure 3.6. NodeMCU ESP8266 pinout diagram

Sensor Modules

In this system, a variety of sensors are used to calculate different environmental parameters. The sensor readings are employed to assess the system operations and decisions. Three types of sensor modules are used in the project as follow:

1) Temperature and humidity DHT22 sensor

The DHT22 is a digital temperature and humidity sensor that is simple and low-cost. It tests the ambient air with a capacitive humidity sensor and thermistor and generates a digital signal on a data-out pin (no analog input pins required). It is relatively easy to use, but it needs careful scheduling for data collection [51]. The only real downside to this sensor is that every two seconds can only get new data. Figure 3.7 shows the DHT22 sensor module.

Figure 3.7. DHT22 temperature and humidity sensor

This sensor is more sensitive, more reliable, and operates in a broader temperature and humidity range compared to the DHT11, but it is bigger and costlier. Also, the humidity readings are between 0-100% with 2-5% accuracy and -40 to 80°C temperature readings with ±0.5°C accuracy.

2) Ultrasonic sensor module HC-SR04

The HC-SR04 is an ultrasonic distance sensor that tells you whether an object is in front of it and also provides the distance between the object and the sensor. This cost-effective sensor offers non-contact measurement functionality from 2 to 400 cm with a range of precision that can reach up to 3 mm. Each unit has a transmitter, receiver circuit, and control circuit [52]. Figure 3.8 shows the HC-SR04 ultrasonic sensor.

Figure 3.8. HC-SR04 ultrasonic distance sensor

It uses sound in its path to assess the distance between the sensor and the nearest object. This sensor is simply a sound sensor, but they function at a frequency above human's hearing. The sensor sends a sound wave to a particular frequency. It then listens to the particular sound wave to bounce off an entity and return to it. The sensor measures the time difference between sending and receiving a sound wave as can be seen from Figure 3.9.

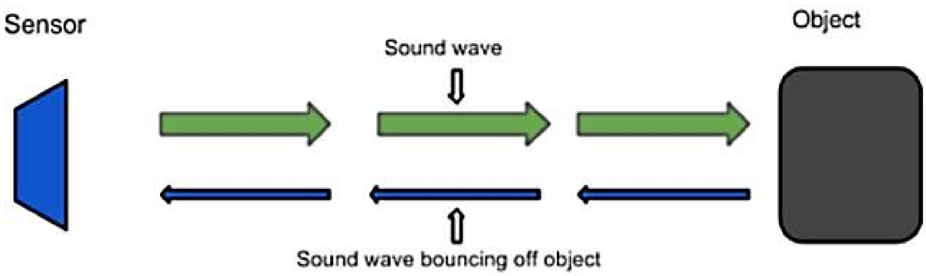

Figure 3.9. Sound wave to bounce off of an object and come back

3) IR infrared obstacle avoidance sensor

The infrared sensor is an electronic system used for heat and object detection. It works by detecting infrared radiation and heat variations in the surrounding environment. This sensor can be commonly used in robot obstacle-avoidance, obstacle-avoidance vehicles, line count, black and white line tracking, and so on and other electronics applications. The IR sensor has a pair of tubes that transmit and receive infrared. Once the transmitted light waves are reflected, the receiver tube will receive the reflected IR waves. The onboard comparator circuitry handles the processing and the green LED indicator comes to life [53]. Figure 3.10 shows the IR sensor.

Figure 3.10. IR infrared obstacle avoidance sensor

Solenoid Electric Door Lock 12V

Solenoids are essentially electromagnets; they are made of a large coil of copper wire with an armature in the center of a slug of metal. The product is ultra-thin, compact, and exquisite, simple to mount, safe, and stable. In particular, this solenoid lock is solid and features a slug with a slant cut and a decent mounting frame [54]. Figure 3.11 shows the electric lock solenoid cabinet.

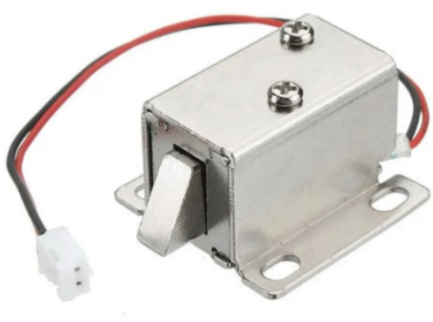

Figure 3.11. Electric door lock solenoid

Its working method is to power on and unlocks the door lock, which is very convenient to use, low power consumption, voltage 12V, current only about 400 mA, and the power supply is immediately triggered. To drive a solenoid lock with an Arduino, it will need an ideal power supply for the relay module because many currents would rush into the solenoid to charge the electromagnetic.

Relay Module

A relay is a switch that is electrically operated. Some relays use an electromagnet to control a switch mechanically, although other operating concepts, such as solid-state relays, are often used. Relays are used where the circuit must be controlled by a separate low-power signal or where multiple circuits must be controlled by a single signal. This module is a suitable board for monitoring high-voltage, high-current loads such as motors, lamps, solenoid valves, and AC loads. It is designed for microcontroller interfaces such as Arduino, PIC, etc. The relays terminal (common (COM), normally open (NO), and normally closed (NC)) is being brought out with a screw terminal. To indicate the status of the relay, it also comes with an LED [55]. Figure 3.12 shows the relay module.

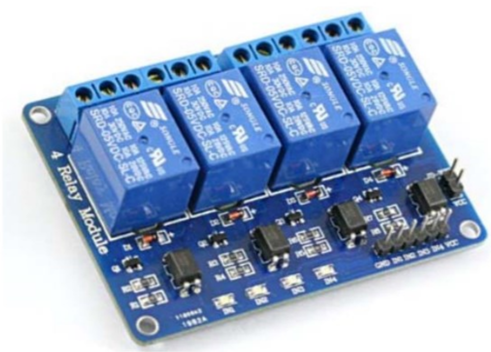

Figure 3.12. Four-channel 5V relay module

3.1.2. Software Design

The software design for developing an IoT-based smart home system will be discussed and explained in detail in this section.

Blynk Platform

Blynk is an IoT platform aimed at simply developing IOS or Android smartphone applications that are used to power Arduino, Raspberry Pi, and NodeMCU over the internet without any significant effort and struggle. To build a graphical interface or human-machine interface (HMI), this application is used by compiling and providing the required address on the widgets available. Blynk has been developed for IoT applications. It can remotely control and monitor hardware, view sensor data, store data, visualize it, and do many other cool things. The platform has three main components as following [56]:

- Blynk Application: It allows you to use different widgets that are offered to create excellent interfaces for your projects.
- Blynk Server: It is responsible for both smartphone and hardware communications. The Blynk server is a bridge that connects smartphones with IoT devices. Users may also run their own local Blynk private server. You may use the Blynk cloud or run Blynk locally. It is open-source; it could manage thousands of devices easily. There several advantages of having the local Blynk server, such as maximum privacy, maximum security, lower latency, and better stability.
- Blynk Libraries: It allows all common hardware platforms to communicate with the server and process all incoming and out-coming commands. The mechanism that happens when someone clicks the button in the Blynk application is that the data will transfer to the Blynk cloud, where data magically makes its way to the mounting hardware. It operates in the opposite direction, and all that happens in the blink of an eye is given in Figure 3.13.

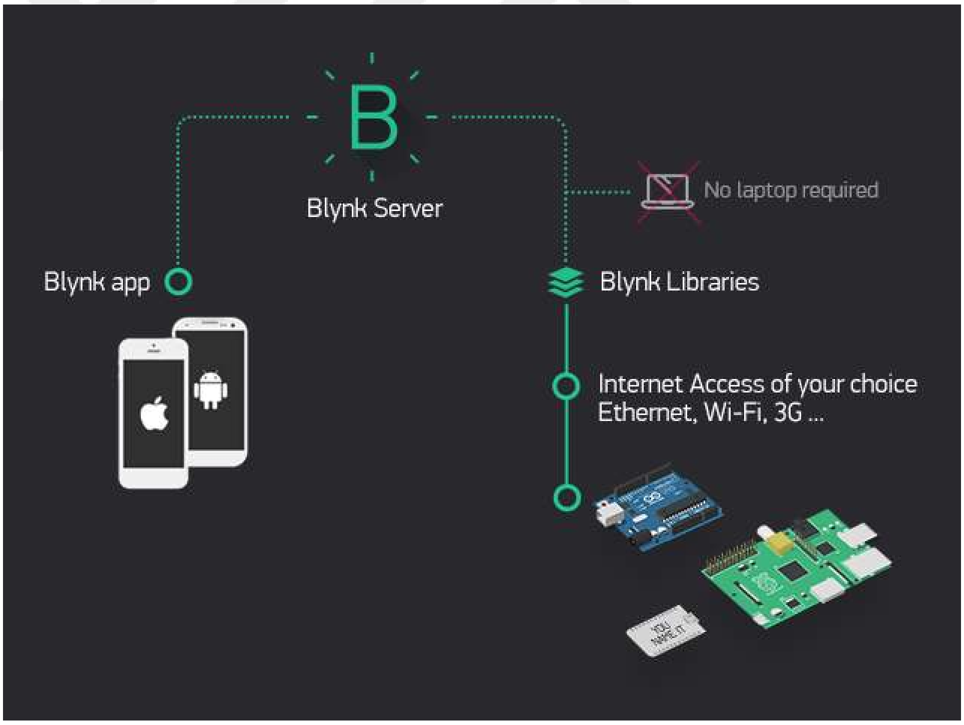

Figure 3.13. System architecture in Blynk platform

Ubidots IoT Platform

Ubidots is a cloud storage system that uses IoT to store a mass of data and record several values for every second. It displays the data linked through the internet in real-time. Within hardware, software, embedded engineering, and manufacturing circles, Ubidots has become recognized as the most inexpensive, secure, and accessible platform in an ecosystem of IoT platforms filled with heavy competition [57]. Figure 3.14 shows the webpage for Ubidots software. Each user of Ubidots has API credentials that must be entered into the code. Then Ubidots will be linked to the hardware, and the data calculated by the hardware will be sent to this IoT system. The device in Ubidots is a virtual representation of the data source. It takes data from the sensor and transmits data via a communication protocol to the Ubidots cloud. Once the device is built, it receives hardware data and is displayed inside the device in a variable. This is achieved in the code by using variable names. After the data is displayed in a variable, a widget is added to the dashboards to visualize the data. Users may display the data in the form of a table, an indicator or a graph, etc. Using events, a warning message or email may also be given to users.

Figure 3.14. Ubidots webpage software

Advanced Encryption Standard (AES)

The AES is one of the most common and widely used block cipher algorithms in the world. This basic method for the encryption and decryption of confidential data is used for hardware and software worldwide. AES is a method used for encrypting or scrambling information. A key is used to encrypt information to securely store and exchange information over a network. Only people with the key will decrypt such information. The same key is used for encryption and decryption of data using an algorithm for symmetric key encryption. The AES was initially named "Rijndael Cipher" because they were the names of the creators. The National Institute of Standards held a competition for a modern, safe encryption process in 1997 in which both developers presented their algorithm. They were the winners of the competition, and the algorithm was called "AES" for the Advanced Encryption Standard in 2001 [58, 59].

The scrambling of the unrecognizable message and output data is the responsibility of the AES algorithm. The data is combined so that it can be stored and distributed securely across the network, and the information can only be decrypted with a password. This is a symmetric key algorithm. Data in 16-byte blocks can either be encrypted or decrypted. This process is operated internally during a variable number of rounds. For 128, 192, or 256-bit cipher keys, 10, 12, and 14 rounds may be needed, respectively [60]. Figure 3.15 shows the process of the AES method.

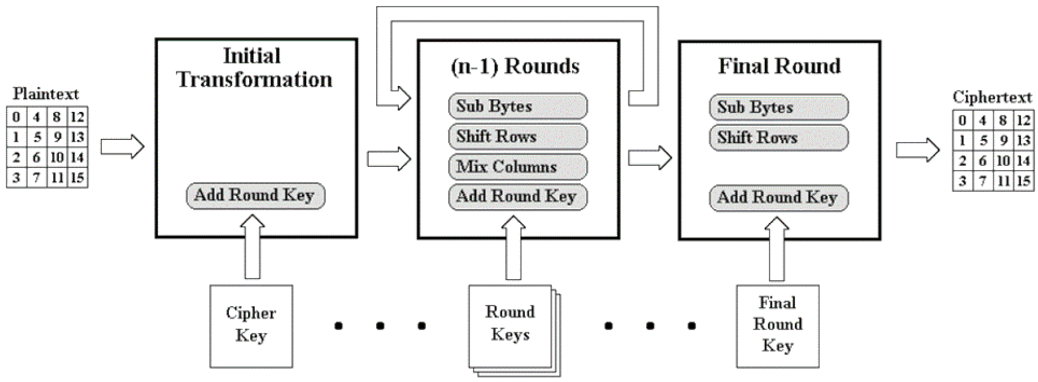

Figure 3.15. The AES algorithm

To test the algorithms submitted by cryptography experts, NIST used three significant criteria, as below:

- Security is one of NIST's most critical features in choosing an algorithm. The key reasons for selecting AES were to highlight the main objectives of the DES algorithm security issue. AES can secure sensitive data from hackers and cannot crack encryption information compared to other proposed algorithms. This was accomplished by performing various experiments on AES in 2017 against theoretical and realistic threats to cryptography and network security.
- The criteria outlined by NIST in the evaluation of algorithms have cost. Again, the reasons behind this measure were also evident in improving the low performance of DES because of another key objective of the AES algorithm. AES was one of the NIST-named algorithms because it could primarily be used for high-speed broadband connections with numerous different applications.
- At this point, the flexibility, simplicity, and suitability of the algorithm for the range of hardware and software implementation were assessed in some important aspects.

Encryption is a common technique that plays a crucial role in protecting information from unauthorized access. To provide the best security, the AES algorithm uses a unique structure to encrypt data. It relies on several rounds to do that and consists of four sub-processes within each round. Each round consists of the four following steps to encrypt 128-bit blocks which are

(SubByes, ShiftRows, MixColumns, and AddRoundKey) but, the transformation of MixColumns is not present in the final round lines. These are linear and nonlinear and are referred to as transformation layers for any round. For every round, there are round keys derived from the original key. From these round transformations and steps, intermediate data is generated and this is called states [61]. The state is a rectangular byte array of four rows and a column, depending on the size of the key length. Figure 3.16 illustrates the basic AES encryption structure.

- Add round key: The plain text will be applied with the initial key in this Add Round Key operation. The critical programming algorithm is used to produce the next round key. As the plain text is 128 bits and the key is also 128 bits, the performance of this operation also is 128 bits.
- Box operation or Substitute byte operation: A S-box is a 16*16 matrix that allows a collection of values according to the inputs of the new values generated by the additional round key. Each element constitutes a byte or 8 bits in the state array matrix. So, we pick the element from the S-box using the element of size 1 byte in the matrix, thus creating a new state array matrix.
- Shift rows: A new status array matrix will be created after the S-box operation. On the newly created matrix, the shift row operations must be performed. On each row, this operation is performed. On each row, the left shift operation is carried out. For the first row, the shift operation takes place just once, and similarly for the second and third row, whether it is the zeroth row, the left shift is performed zero times.
- Mix column operation: In the state array, the array that was created after the shift row operation is stored. This state array is then subjected to a column mixing process. There will be a predefined matrix in this column mix process. Each column of the state matrix multiplies this preset matrix. This results in the old column being replaced.

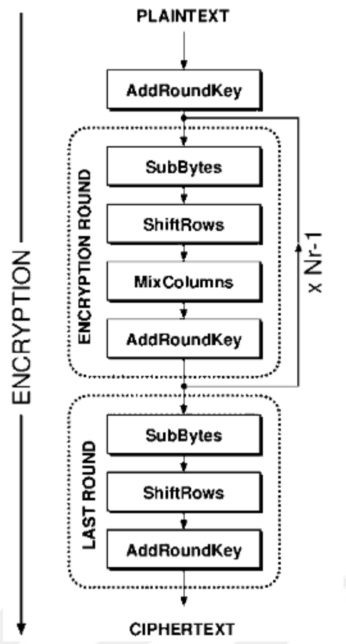

Figure 3.16. The AES encryption algorithm basic structure

The decryption method is the process of retrieving the original encrypted data. This process is based on the key received from the data sender. The AES decryption procedures are identical to the reverse order encryption method, and both sender and receiver have the same data encryption and decryption key. There are three steps in the last round of the decryption process, InvShiftRows, InvSubBytes, and AddRoundKey. Figure 3.17 illustrates the basic AES decryption structure.

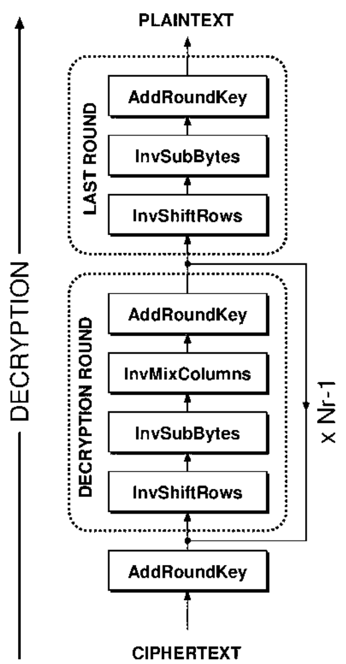

Figure 3.17. The AES decryption algorithm basic structure

3.2. Implementation

The system implementation has two main parts which are the hardware implementation and software implementation. The hardware is implemented by organizing microcontrollers, sensors, and actuators, while software implementation requires programming. The system has been configured to monitor and control household devices by a microcontroller connected to the sensor modules and actuator modules.

3.2.1. Hardware Implementation

The hardware system contains two main parts: the first part is Arduino Uno, and the second is NodeMCU ESP8266. The Arduino Uno is in charge of sending all the data to the NodeMCU board by Universal Asynchronous Receiver Transmitter (UART) protocol which is a physical circuit in a microcontroller and the main purpose of UART is to transmit and receive serial data. The sensor modules used in this project are DHT22 temperature and humidity, HC-SR04 ultrasonic distance sensor, and IR infrared obstacle avoidance sensor are connected and wired with the Arduino Uno digital pins. The second part is the NodeMCU ESP8266 board that is connected to a web server. The NodeMCU is responsible for controlling the loads connected to the relay and monitoring the status of the door and window, and reading the other sensor's status.

Interfacing NodeMCU ESP8266 with Arduino Uno

NodeMCU is the main part of the system. It is used for producing an interface between microcontroller and IoT. NodeMCU is great for connecting to the cloud, and Arduino is great at talking with different sensors. So, on the Arduino side, we had taken samples periodically and sent a JavaScript Object Notation (JSON) message over software serial to NodeMCU. On the NodeMCU side, we receive the messages sent by Arduino Uno over serial. Arduino will send the messages in JSON format. NodeMCU ESP8266 and Arduino Uno are connected by serial communication UART protocol. Figure 3.18 illustrates the interfacing between the main two parts of the system.

The Vin and GND pins of NodeMCU connect with +5V, and GND pins of Arduino and GPIO 2 and GPIO 4 pins of NodeMCU declared as RX and TX connect with digital pin 5 as RX and digital pin 6 as TX of Arduino Uno.

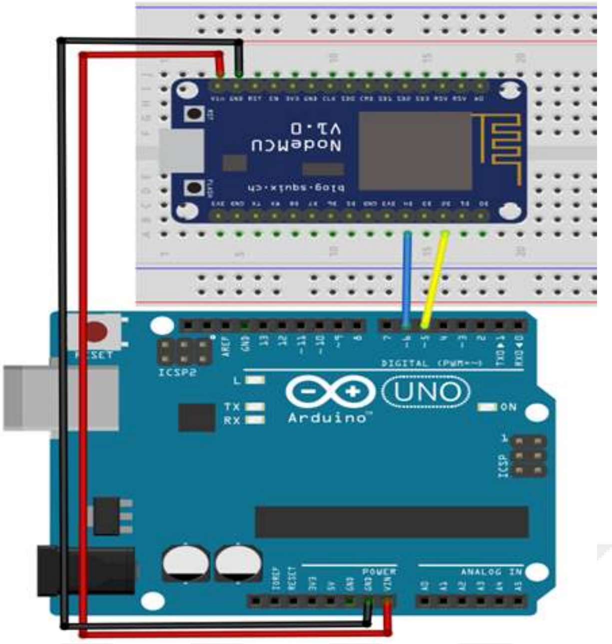

Figure 3.18. NodeMCU module connection with Arduino Uno

Interfacing DHT22 Temperature and Humidity Sensor

The DHT22 sensor is made up of two measuring components: the humidity sensing component and the NTC temperature sensor or a thermistor. On the rear side, there is an IC that allows the Arduino to read the readings. The humidity sensor comes into play while measuring the humidity. The humidity sensor has two electrodes and a layer of the substrate between those electrodes. So the conductivity of the substrate changes as the humidity changes, or you might tell that the resistance between electrodes changes. The IC is then given this change in resistance, which makes it read by the Arduino. The NTC temperature sensor or thermistor comes into play when calculating the temperature. The thermistor's resistance changes as the temperature changes. These are negative coefficients of temperatures (NTCs), which decrease resistance with the temperature increase. Figure 3.19 shows the wiring diagram of the DHT22 sensor with Arduino Uno in this system.

We have included the DHT22 library in the sketch file to communicate with the sensor. It reads the DHT22 sensor and prints the results to the Arduino Uno console on the computer connected to the USB port. The VCC and GND pins of the DHT22 sensor connect with +5V, and the GND of Arduino Uno and the data pin of the DHT22 sensor connect with digital pin 10 of Arduino.

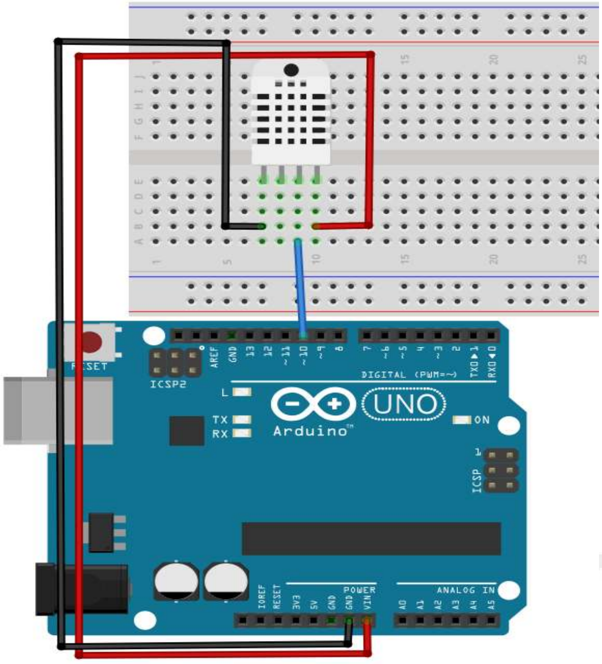

Figure 3.19. DHT22 sensor connection with Arduino Uno

Interfacing HC-SR04 Ultrasonic Distance Sensor

The ultrasonic sensors are widely used for obstacle detection and distance measurement applications. This sensor uses the same radar concept, which turns electrical energy into acoustic waves and vice-versa. The ultrasonic sensor is based on the SONAR basics, short for Sound Navigation Ranging. By giving a 10μS HIGH-level signal on the TRIGGER input pin, the functionality of this sensor can be initiated. Then the module sends out eight ultrasonic bursts of 40KHz. These ultrasound waves pass through the air, and if an object or obstacle is located in their direction, it immediately reflects or bounces back to the module receiver. If the module senses the signal back, a HIGH pulse in the ECHO pin is sent for the duration of the time it takes for the ultrasonic waves to return. The distance of the obstacle can now be measured using the duration of the time. Figure 3.20 shows the wiring diagram of the sensor.

The VCC and GND pins of the ultrasonic distance sensor connect with +5V and GND of Arduino Uno, and the input trigger pin and output ECHO pin of the ultrasonic sensor connect with the digital pin 3 and digital pin 2 of Arduino Uno respectively.

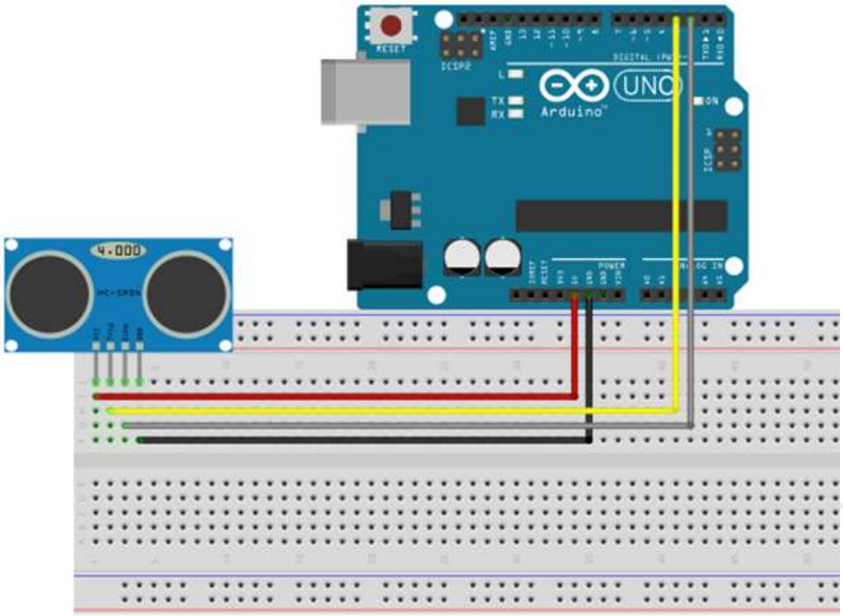

Figure 3.20. HC-SR04 ultrasonic sensor connection with Arduino Uno

Interfacing IR Infrared Obstacle Avoidance Sensor

The IR sensor module has the great adaptive capability of the ambient light; through the potentiometer, the module detection range can be adapted and have small interference, simple to assemble, simple for use.

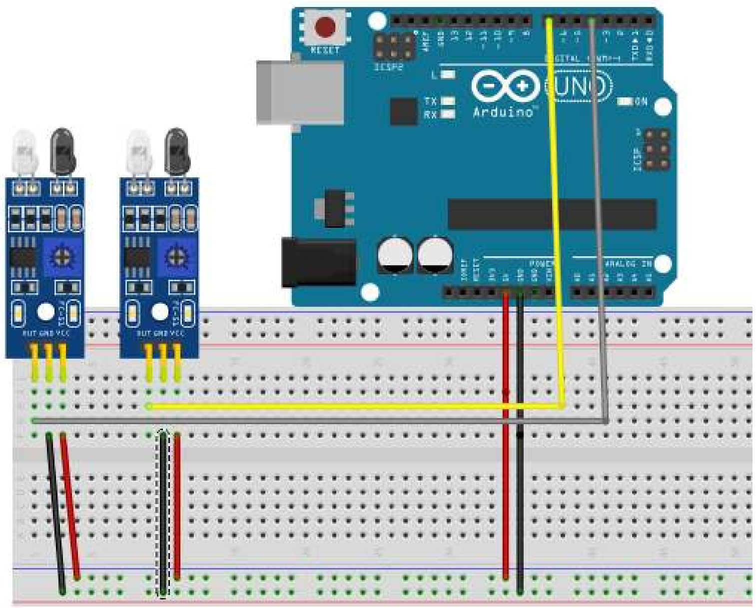

Figure 3.21. IR sensors connection with Arduino Uno

The VCC and GND pins of the IR 1 sensor connect with +5V and GND pins of Arduino Uno, and the OUT pin of IR 1 is connected to digital pin 4 of Arduino Uno monitors the status of the window. The VCC and GND pins of the IR 2 sensor connect with +5V and GND pins of Arduino Uno, and the OUT pin of IR 2 is connected to digital pin 7 of Arduino Uno monitors the status of the door. The output of the IR sensor functions as a switch. When an IR receiver detects infrared radiation, the output will be high. Otherwise, the output will be below. This is essentially identical to the push button and switch features. Figure 3.21 illustrates the connection between the sensor and Arduino.

Interfacing Relay Module with NodeMCU ESP8266

The relay module is used for interfacing between the NodeMCU and the house devices. In this study, the three switches of the relay board and NodeMCU ESP8266 are used to connect with devices. 5V is used as the power source for relay switches. When the relay is turned on, the bulbs get the power from the 230V AC source, and the solenoid gets the power from the 12V power supply source. Moreover, two switches of the relay board used for the bulbs and one switch used for the solenoid, and all of them are controlled by the switch button via the Blynk android application and Ubidots IoT platform web application. Figure 3.22 shows the wiring connection between the relay board and the NodeMCU and the wiring diagram of relay switches with loads shown in Figure 3.23.

The VCC and GND pins of the relay board connect with Vin and GND pins of NodeMCU and IN1, IN2, and IN3 of relay board connect with D5, D6, and D7 NodeMCU.

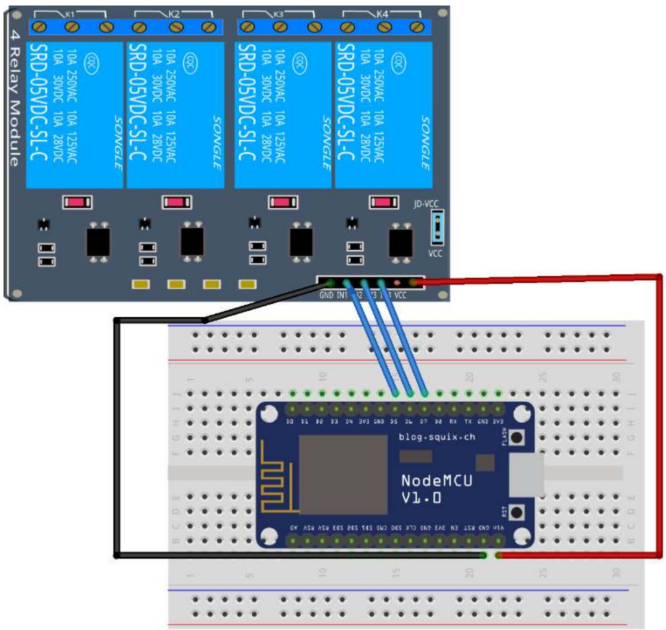

Figure 3.22. Relay switches connection with NodeMCU ESP8266

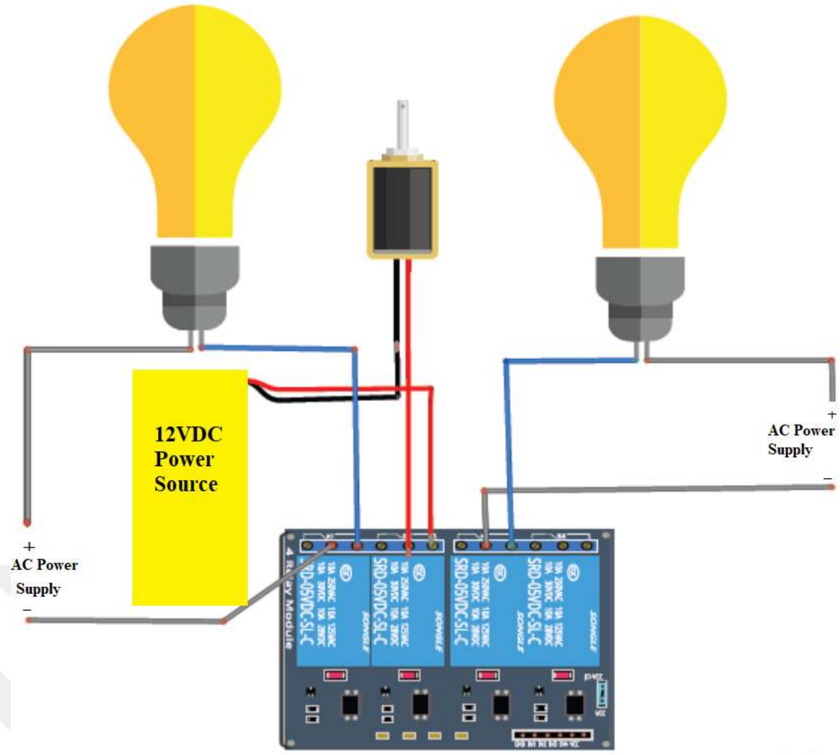

Figure 3.23. Wiring diagram of relay switches with loads

The Normally Open (NO) of relay switch IN1 connects with +ve of bulb 1, the common terminal (COM) of relay switch IN1 connects with AC live power supply, and the -ve of bulb 1 connects with AC neutral power supply. The Normally Open (NO) relay switch IN2 connects with 12VDC, the common terminal (COM) of relay switch IN2 connects with 12V of the solenoid, and the ground of the solenoid connects with the ground of the battery. The Normally Open (NO) relay switch IN3 connects with +ve of bulb 2, the common terminal (COM) of relay switch IN3 connects with AC live power supply, and the -ve of the bulb 2 connects with AC neutral power supply.

3.2.2. Software Implementation

The software is divided into two key portions. The first part relates to the exchange data side. In which the Arduino microcontroller was programmed to transmit sensor modules data to NodeMCU ESP8266 by UART serial communication protocol. Arduino will send the messages in JSON format. The second part is related to the control and communication side. In which, NodeMCU receives the messages sent by Arduino Uno over serial. NodeMCU ESP8266 looks for the Service Set Identifier (SSID) preset and links to the internet automatically. The controller NodeMCU is used to connect the system with the Ubidots IoT platform webpage and Blynk android application by its Wi-Fi for establishing remote controlling and monitoring continuously with a smartphone or PC. The NodeMCU is programmed within Arduino Integrated Development

Environment (IDE) by using USB to let NodeMCU know what to do. In this system, the AES technique is used to encrypt and decrypt the system's sensitive data. The encryption and decryption process allows the connection between the NodeMCU ESP8266 and the server to be authenticated.

Blynk Android Application

In this thesis, we have used the Blynk android application, which is a mobile application that has its server to process user requests. To prepare the android application, we have performed the following steps:

- Download the android Blynk application from Play Store.
- Create an account in the Blynk application.
- Create a project in the Blynk application.
- Name the project to "Smart Home”.
- Select the NodeMCU from the drop-down list.
- Receive an authentication token in our registered email.
- Tap on the screen to open the Widget Box.

Three different buttons were prepared for the three appliances named Light1, Solenoid, and Light2 connected to Blynk digital pins D5, D6, and D7 respectively. Two gauges are set to temperature and humidity on Blynk virtual pins V0 and V1. Also, two styled buttons are set for door status and window status on V3 and V4 from IR1 and IR2 sensors. Finally, one value display is set for the ultrasonic sensor named length on V2. Figure 3.24 and Figure 3.25 show the Blynk home screen and create a project in Blynk respectively. Also, Figure 3.26 displays the Blynk android application screenshots.

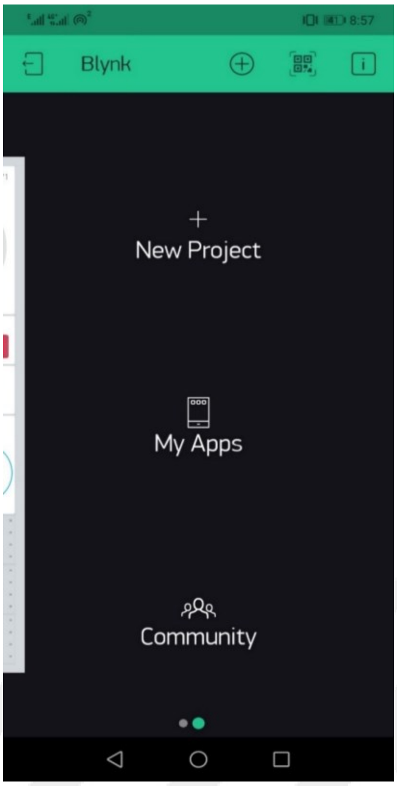

Figure 3.24. Blynk application home screen

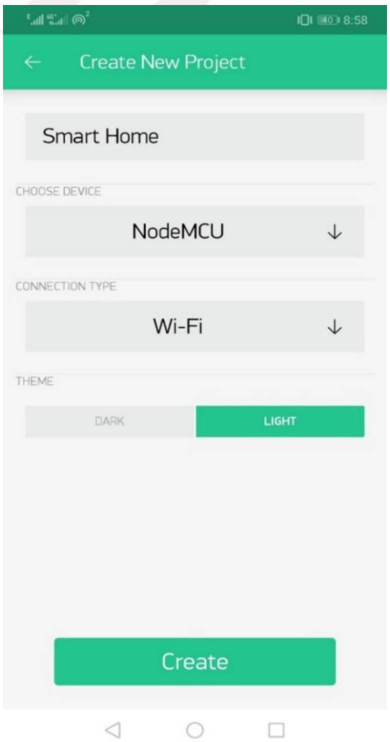

Figure 3.25. Creating a new project in Blynk

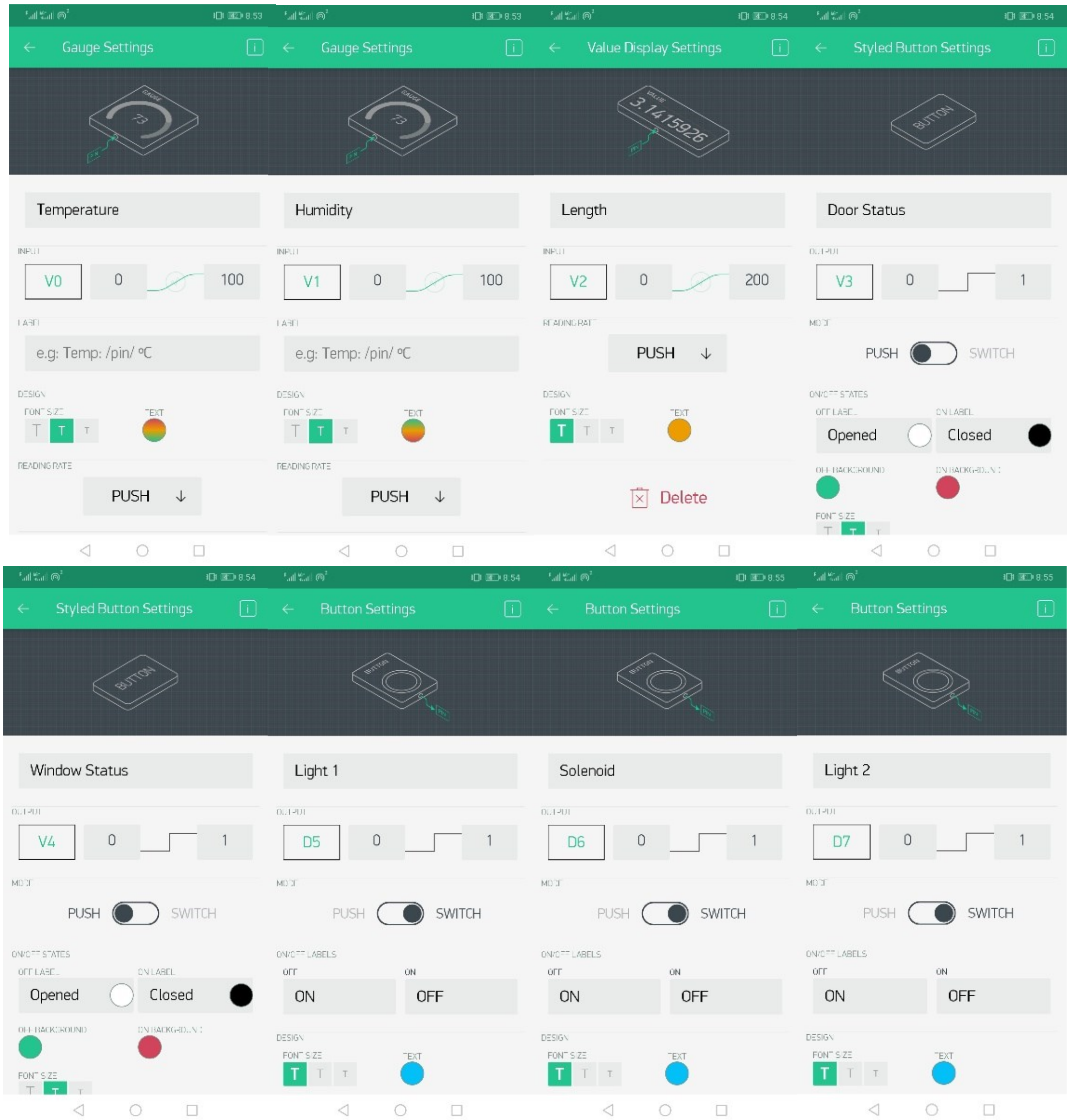

Figure 3.26. Screenshots of setting Blynk android application

The system is based on the NodeMCU ESP8266 module as an IoT system. As the NodeMCU has an ESP8266 circuit to connect to the internet, the NodeMCU is connected to the internet through a Wi-Fi connection. To be connected to the Wi-Fi, NodeMCU must first be identified by the Wi-Fi name, password, and token code that allows Blynk's server to connect them. The software part of the project was prepared in Arduino IDE. The Blynk libraries are ZIP files that can be downloaded and imported to the Arduino IDE library from the Github website. Blynk server will check whether there is an internet connection. The NodeMCU code includes the token code, the SSID, and its password. For ESP8266 to connect to Wi-Fi as a channel to exchange commands between the smartphone and NodeMCU, the information included in the code must match the

Wi-Fi information. The remaining processes are only commands sent to NodeMCU from the Blynk application to control loads that are linked to the relay module. Furthermore, also sensor values from the NodeMCU module are forwarded reverse to the Blynk application. Figure 3.27 illustrates the flowcharts diagram of switching ON/OFF loads.

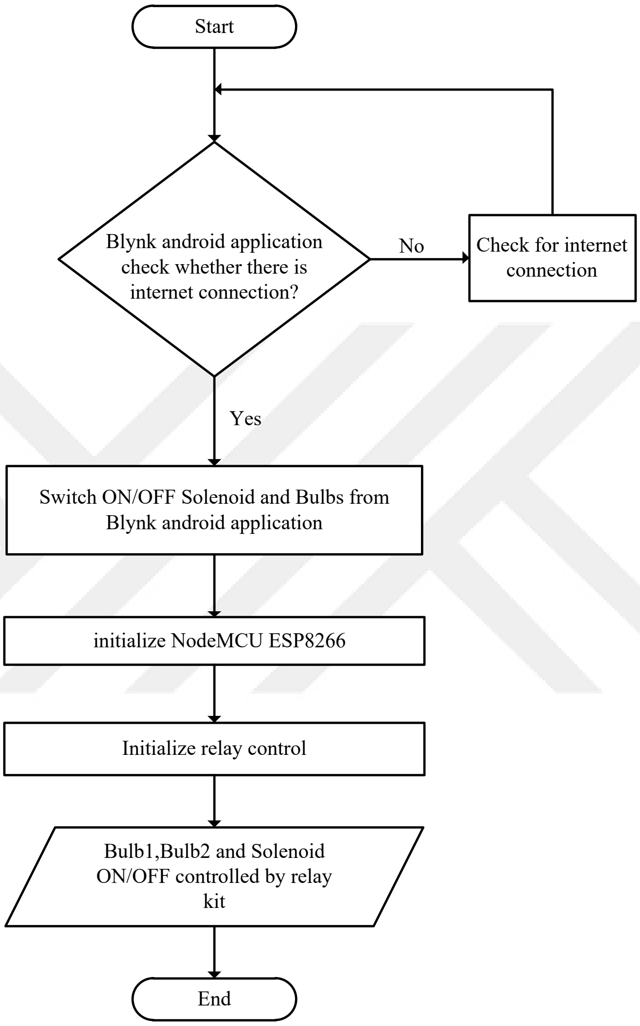

Figure 3.27. Flowchart diagram of switching ON/OFF loads

To display the value of the sensor modules on the android display, NodeMCU returns the output values of the sensor to the Blynk application. Like the last flowchart of the ON/OFF process, the Blynk server will check the sensor output value for internet connection, Wi-Fi name, and password to view the data properly. The data are displayed in the Blynk application, as shown in Figure 3.28.

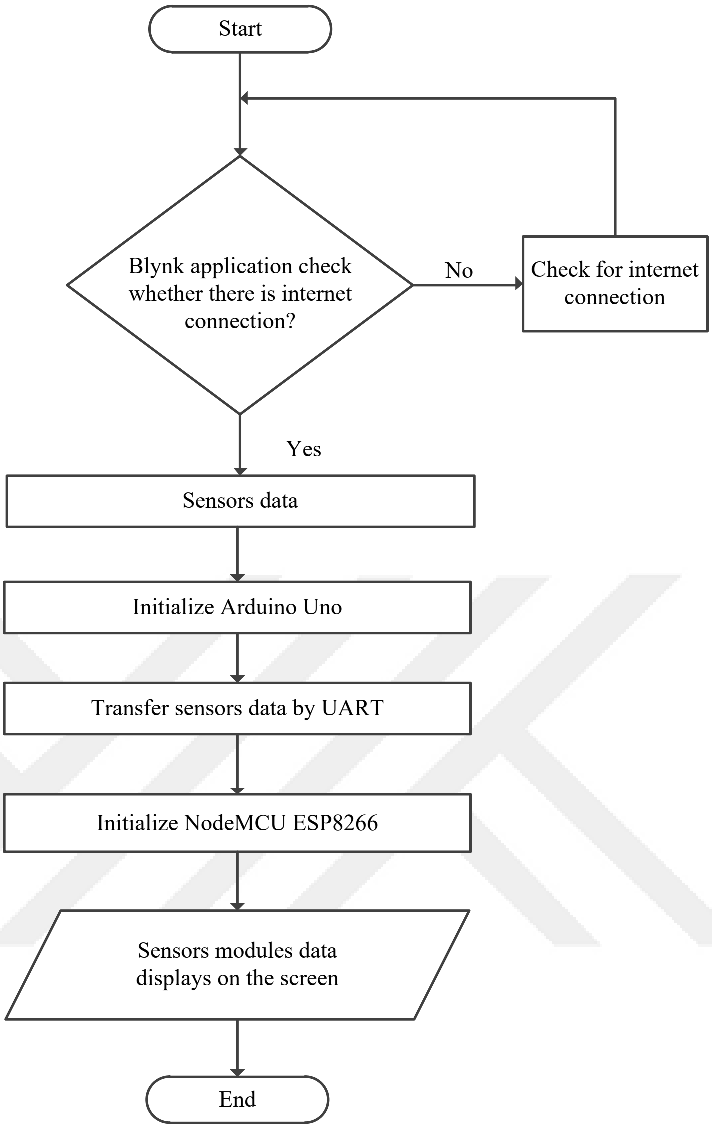

Figure 3.28. Flowchart diagram of sensors data in Blynk application

The sketch starts by including the Blynk library.
#include <BlynkSimpleEsp8266.h>

The code contains the name of the Wi-Fi and the android password match. When authentication (autho token) is provided as an email by the Blynk application and the SSID is the Wi-Fi name.
#define WIFINAME "HUAWEI nova 3e"
#define WIFIPASS "taha12345"
#define BLYNK_PRINT Serial
char auth[] = "PXuwy-Tup7RiK7_fanjoXIj22EOSrh8N";

We define virtual PINs for our sensors output and input pins on the Blynk application, So, that we can access special functions related to the library. The relay input does not need to be defined by the code, as it is included in [Blynk.run();].

```
Blynk.virtualWrite(V0, t);
Blynk.virtualWrite(V1, h);
Blynk.virtualWrite(V2, cm);
Blynk.virtualWrite(V3, door);
Blynk.virtualWrite(V4, win);
```

Ubidots IoT platform webpage

Ubidots IoT platform was also used in this project to monitor and manage home devices. Ubidots is software that helps as a barrier to transfer the data to remote users with the help of a Wi-Fi facility. IoT cloud platform Ubidots interfaced with the NodeMCU ESP8266 to send temperature, humidity, door status, window status, distance data from the sensor modules, lights, and the solenoid data from the loads. Uploaded data can also be visualized on Ubidots dashboard with the help of widgets.

First, we create an account on Ubidots for education and complete the account setup. By clicking on the user dropdown menu and click on API documentation, we get the "API Tokens Key" and copied that token key, and pasted it into the Arduino IDE code. The tokens key is used to authenticate the API requests. Figure 3.29 and Figure 3.30 show the steps mentioned above.

Figure 3.29. Creating Ubidots account

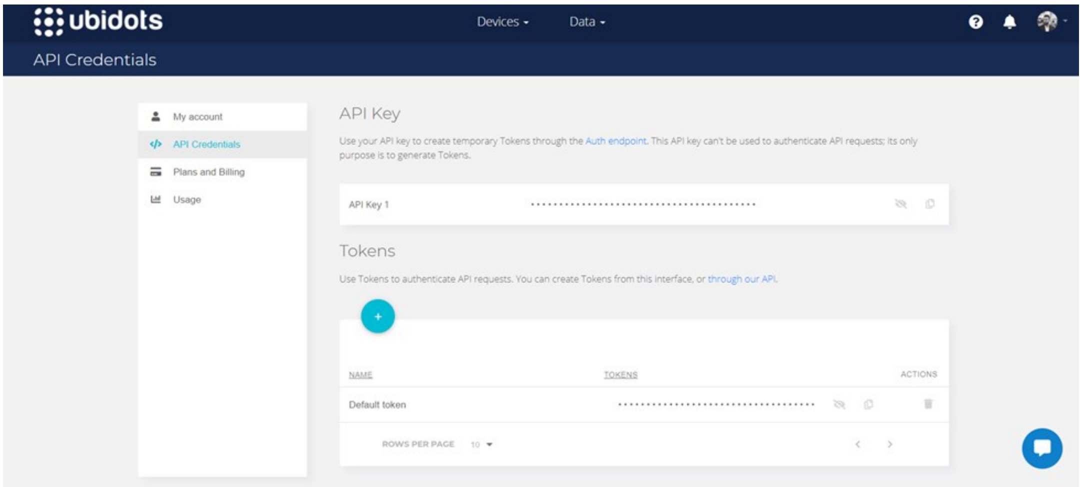

Figure 3.30. Creating Tokens key from Ubidots account

After logging in to the education account created by Ubidots, we will be requested to generate “devices”. To create a new device, we click on Add device and name the device by double click on the device, and then we will be prompted to create "variables". Figure 3.31 illustrates the created devices on Ubidots. By clicking on add variables to create variables and selecting the default, we name the variables like temperature, humidity, door status, window status, length, solenoid, bulb 1, and bulb 2. Figure 3.32 and Figure 3.33 illustrate the created variables for the loads and sensors respectively.

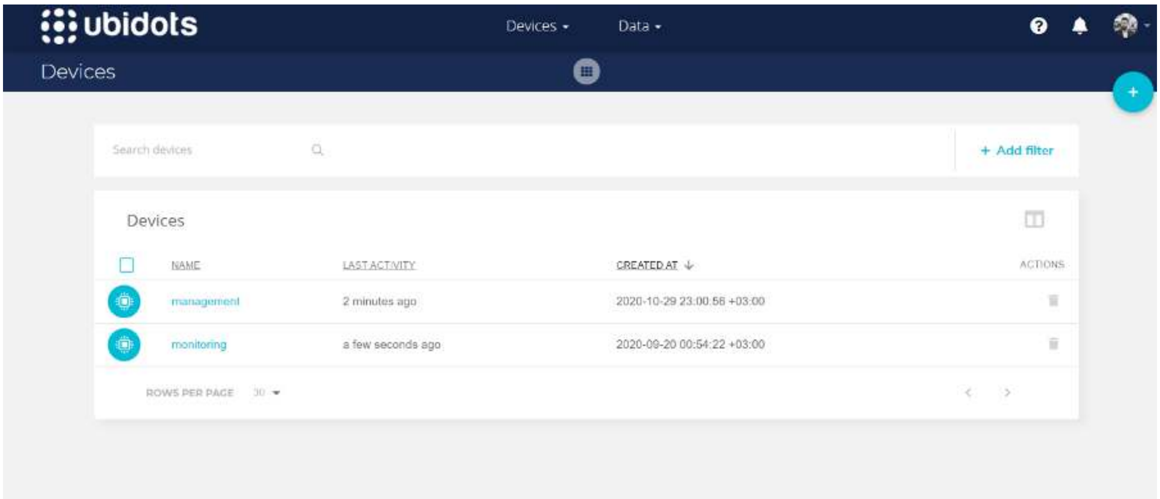

Figure 3.31. Creating devices on Ubidots

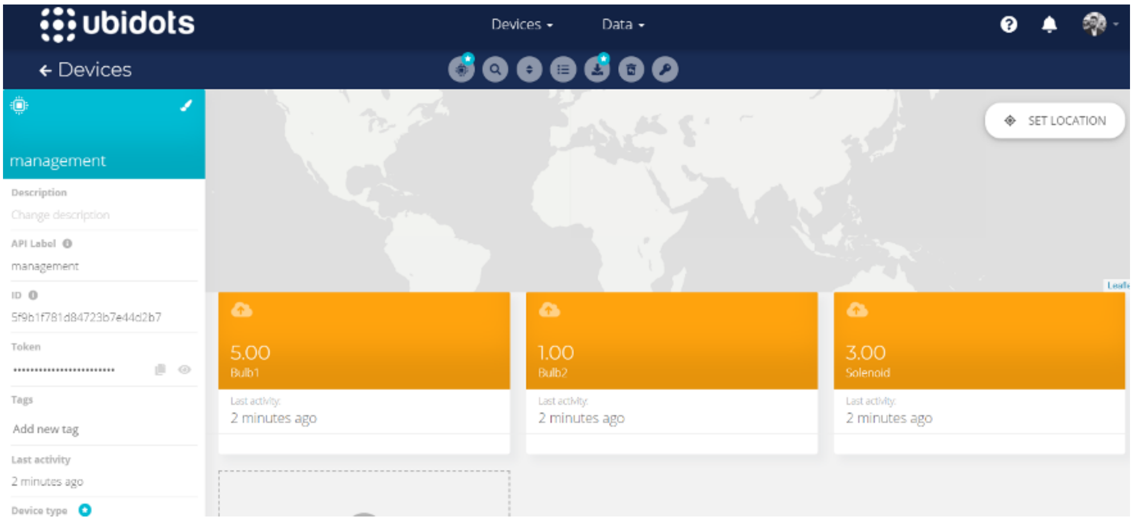

Figure 3.32. Creating variables for the loads

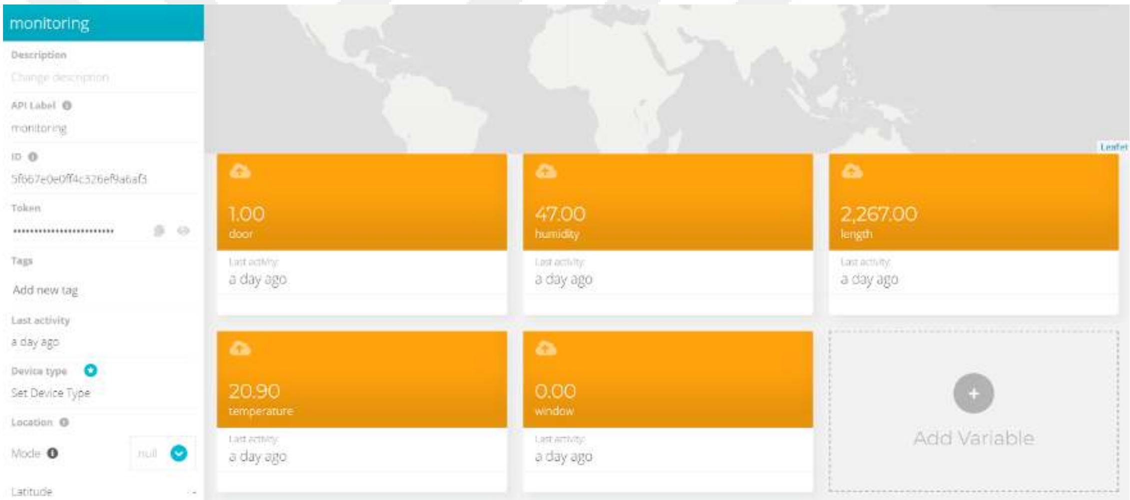

Figure 3.33. Creating variables for the sensors

After successfully the variables created, we move to the dashboards to create a widget to visualize the variables. By clicking on + to add a widget, we can visualize the data from the device in the form of metrics, tank, and thermometer. To visualize the length, humidity, and temperature in values, the "last value" has been selected. Two indicators were selected to visualize the door status and window status, and for both of them, "set by dashboard" are selected. Also, three switches were selected for bulb1, bulb2, and the solenoid. Figure 3.34 shows the types of the widget in Ubidots dashboard.

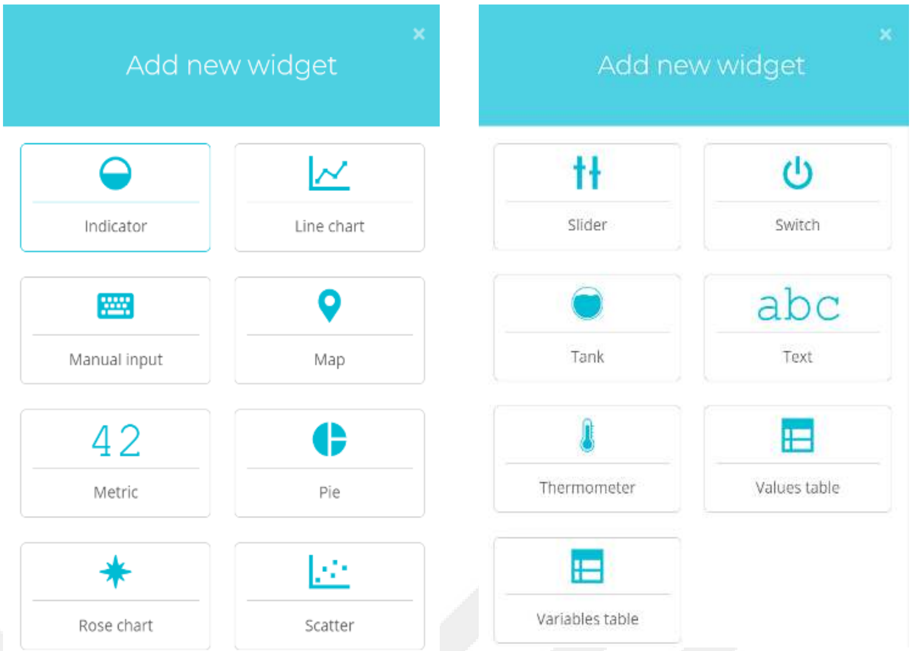

Figure 3.34. Adding widget to see data in real-time

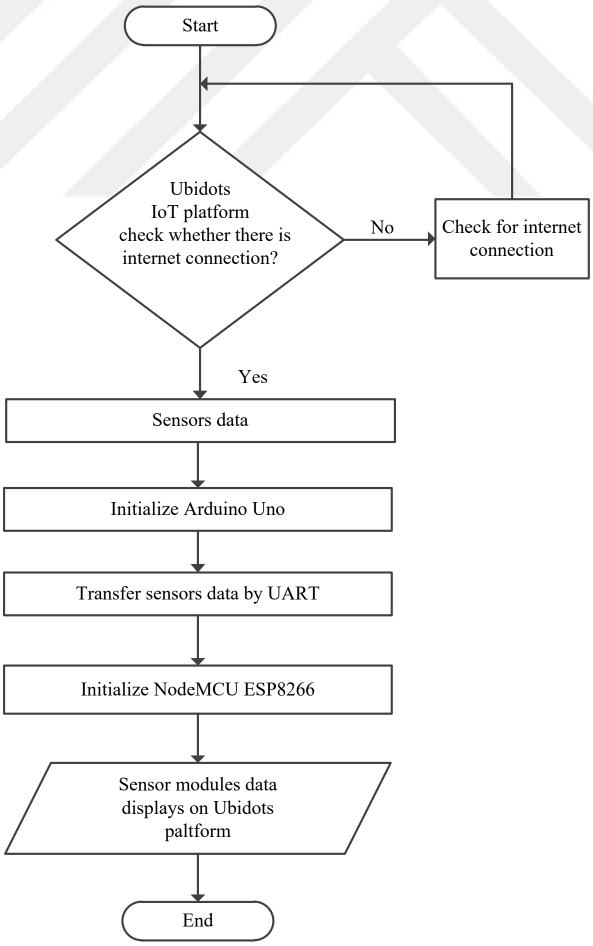

Figure 3.35. Displays sensor data readings on Ubidots

Figure 3.35 illustrates the flowchart diagram of displaying sensors reading on Ubidots account. The sensors are connected with Arduino pins, and their data transferred to NodeMCU ESP8266 by serial communication. Also, the relay switches are connected with NodeMCU. An MQTT-based server cloud is created for the connection protocol. MQTT is beneficial to push data from the cloud to the devices. The Wi-Fi network is established by adding the necessary library files to Arduino IDE software and NodeMCU files so that the data can be stored in Ubidots. Figure 3.36 shows the steps of load control and display on the Ubidots system.

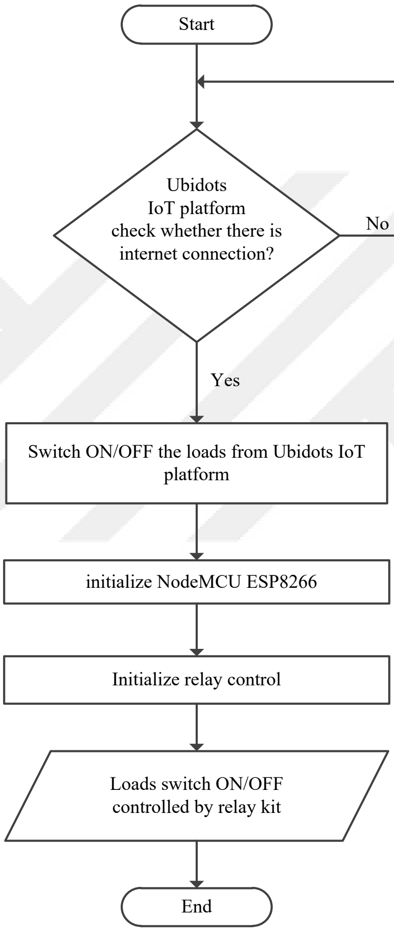

Figure 3.36. Loads are controlled and displayed by Ubidots

To start the code, we include and download all the required libraries, extract them, and paste them into the Arduino library folder to create an MQTT connection with the Ubidots.

#include "UbidotsESPMQTT.h"

To connect with our account in Ubidots we enter the token key from API credentials and SSID name and its password to setup a connection between NodeMCU and Ubidots.

#define TOKEN “BBFF-6wYJCW3nz6f5oMUlqmutB014fsjvpl"

#define WIFINAME "HUAWEI nova 3e"

```
#define WIFIPASS "taha12345"
```

We create an object named client for the class Ubidots with our token as the parameter.

```
Ubidots client(TOKEN);
```

In the setup loop, we initialize the Serial and Wi-Fi connection as shown below.

```
Serial.begin(115200);
 s.begin(115200);
 while (!Serial) continue;
 client.wifiConnection(WIFINAME, WIFIPASS);
client.begin(callback);
```

Also, we insert the dataSource and Variable's Labels.

```
client.ubidotsSubscribe(DEVICE_LABEL, VARIABLE1);
 client.ubidotsSubscribe(DEVICE_LABEL, VARIABLE2);
 client.ubidotsSubscribe(DEVICE_LABEL, VARIABLE3);
```

And finally, as shown below we use the client object to publish the temperature, humidity, length, window and door values to the Ubidots cloud.

```
client.add("Temperature",t);
client.ubidotsPublish("Monitoring");
client.add("Humidity",h);
client.ubidotsPublish("Monitoring");
client.add("Length",cm);
client.ubidotsPublish("Monitoring");
client.add("Door",door);
client.ubidotsPublish("Monitoring");
client.add("Window",win);
client.ubidotsPublish("Monitoring");
```

Encryption Algorithm

In this system, we use the AES algorithm to secure our smart home systems data. AES is one of the most popular and widely used symmetric block cipher algorithms in the world. This algorithm has its basic framework for the encryption and decryption of sensitive data and is applied worldwide in hardware and software.

The key idea is to encrypt all transactions in the ESP8266 NodeMCU and server, and to supplement the proposed encryption technique, the base64 encoding and decoding technique is used to make the encrypted transaction safer. This research lists the processes involved as follows:

- Use AES -128 technique to encrypt and decrypt data.
- Encode and decode the encrypted data.

Some side operations were involved in the proposed technique, and each side has its processes. The process starts with NodeMCU ESP8266 at first gets the original sensor modules data from the Arduino with the original load's data connected with NodeMCU through relay switches. These data will be encrypted into a string by the AES algorithm, only the system knows what these generated data means are, and after the data are encrypted, it will be encoded before it sends out to the server to initiate the session. The server processes the data by decrypt and decode. In Figure 3.37, the steps in the encryption and encoding process are given. The connection between the ESP8266 and the server needs to be authenticated. The ESP8266 is used as a special credential key. The key is read encrypted into the string by the AES technique and then goes through the encoding process. After all the data has been encrypted and encoded, the strings are sent to the server. The strings obtained by the server are decrypted and decoded, and the connection authentication is recognized as a credential between ESP8266 and the server.

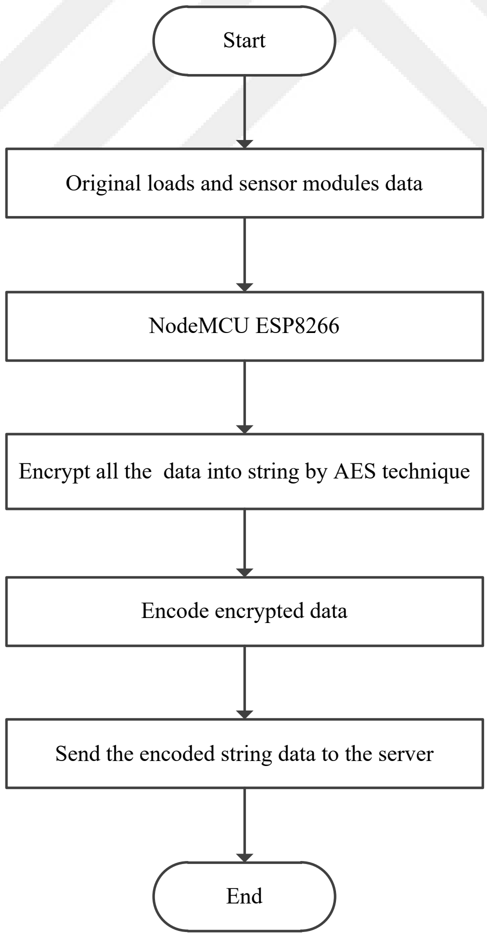

Figure 3.37. Data encrypt and encode process

Once the server has received the credentials successfully, the chipper text is produced and sent to NodeMCU ESP82666. Figure 3.38 shows the decrypt and decode process.

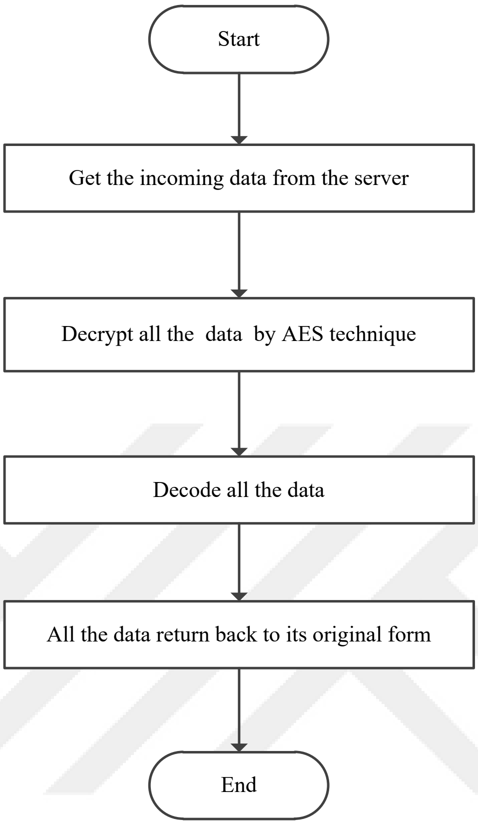

Figure 3.38. Data decrypt and decode process

To start the encryption and decryption functions, we include the AES library in the Arduino IDE code which exposes the set of functionalities needed to use the AES encryption algorithm.

```
#include "AESLib.h"
```

First, we read the DHT22 temperature and humidity sensor values:

```
String readBuffer4 = String(t, DEC);
```

Encrypt the string received from the DHT22 sensor and encoded the values in a string.

```
uint16_t clen4 = String(cleartext).length();
String encrypted4 = encrypt(cleartext, clen4, enc_iv);
```

Then decrypt the string received from the DHT22 sensor and decode the values in a string.

```
uint16_t dlen4 = encrypted4.length();
String decrypted4 = decrypt( ciphertext, dlen4, dec_iv);
```

When the system decrypts the string, it will print the word "SUCCES”, if not it will print “FAILURE” as shown below.

```
 if (decrypted4.equals(cleartext)) {
 Serial.println("SUCCES");
   }
        else
{
 Serial.println("FAILURE");
   }
String readBuffer3 = String(h, DEC);
uint16_t clen3 = String(cleartext).length();
String encrypted3 = encrypt(cleartext, clen3, enc_iv);
uint16_t dlen3 = encrypted3.length();
String decrypted3 = decrypt( ciphertext, dlen3, dec_iv);
```

Other sensors such as ultrasonic, IR1, and IR2 use the same process and functions.

For the loads connected with relay switches, we use the same function to read the values of the variables, as shown below for the solenoid:

```
String readBuffer0 = String(VARIABLE2);
```

Encrypt the string received from VARIABLE2 which is a solenoid, and encoded the values in a string.

```
uint16_t clen0 = String(cleartext).length();
String encrypted0 = encrypt(cleartext, clen0, enc_iv);
```

Then decrypt the string received from VARIABLE2 and decode the values in a string.

```
uint16_t dlen0 = encrypted0.length();
String decrypted0 = decrypt( ciphertext, dlen0, dec_iv);
```

We repeat the same process and functions for the other loads bulb1 and bulb2.

4. APPLICATION RESULT

In this chapter, the results of the smart home application implementation based on IoT will be discussed. This system can connect with the internet to control and monitor the current environment inside the house anywhere at any time through the NodeMCU ESP8266 Wi-Fi module. The main hardware parts of the system are the NodeMCU ESP8266 module, Arduino Uno board, and relay board. Also, the Ubidots IoT platform and Blynk application have been introduced to allow the use of IoT.

The various sensor modules data in the home environment are sent through the Arduino Uno as input to the NodeMCU by serial communication UART protocol. The NodeMCU ESP8266 is coded via Arduino IDE software with the USB to tell the NodeMCU what to do. NodeMCU ESP8266 controls three switches of relay board and receives instructions to show the temperature, humidity, door status, window status, and distance measurements from Blynk mobile application and Ubidots IoT platform web application and processes them to monitor the sensors and control switching ON/OFF loads such as lights and solenoid door lock. Also, in this system, the AES technique was implemented to increase confidentiality in data security. All the information in the system was encrypted and decrypted. Figure 4.1 shows the entire setup of the smart home system.

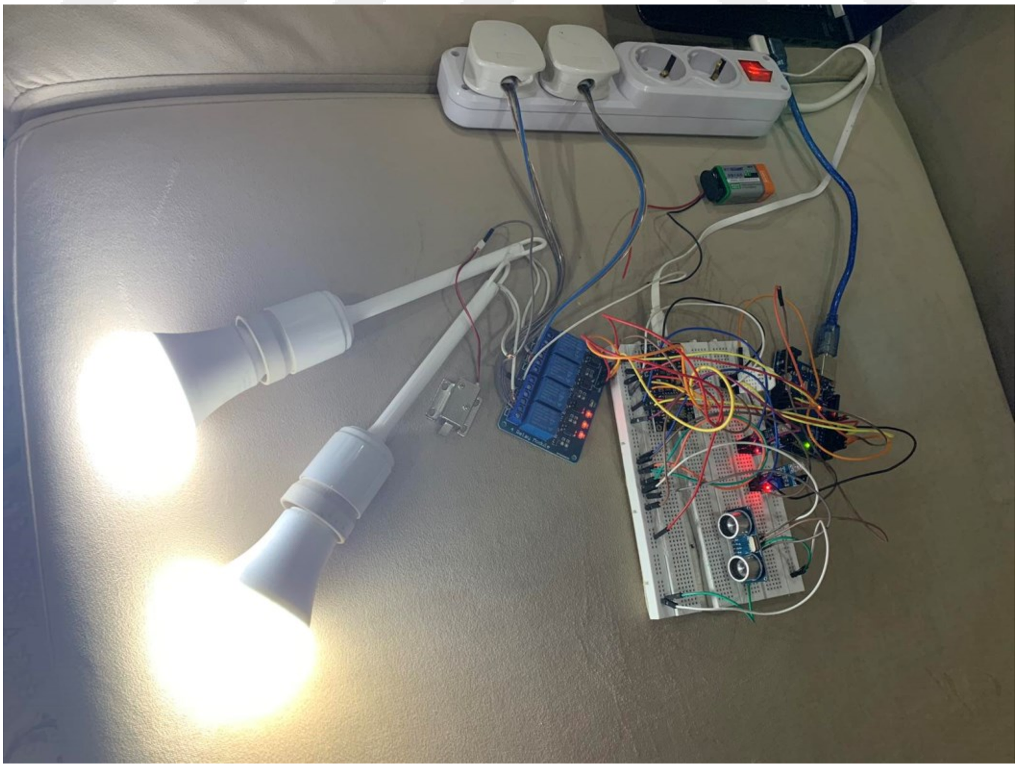

Figure 4.1. The entire setup of the smart home system

4.1. Hardware system

In the performed system, all the hardware parts were checked and tested at first for the certainty that all devices were working correctly, then each part was tested separately with the microcontrollers, it was obtained as a result of this test all hardware parts were perfectly functioning. Thus, all the components of the hardware were linked to each other, and the entire system was checked. Also, the main important parts of the system, such as the connection between Arduino Uno and NodeMCU ESP8266 for exchanging sensor data, were checked to make sure if it operated properly. Furthermore, we checked the communication of the ESP8266 Wi-Fi module with the internet to be sure that the connection was made correctly. Figure 4.2 illustrates the sensor readings data on the serial monitor.

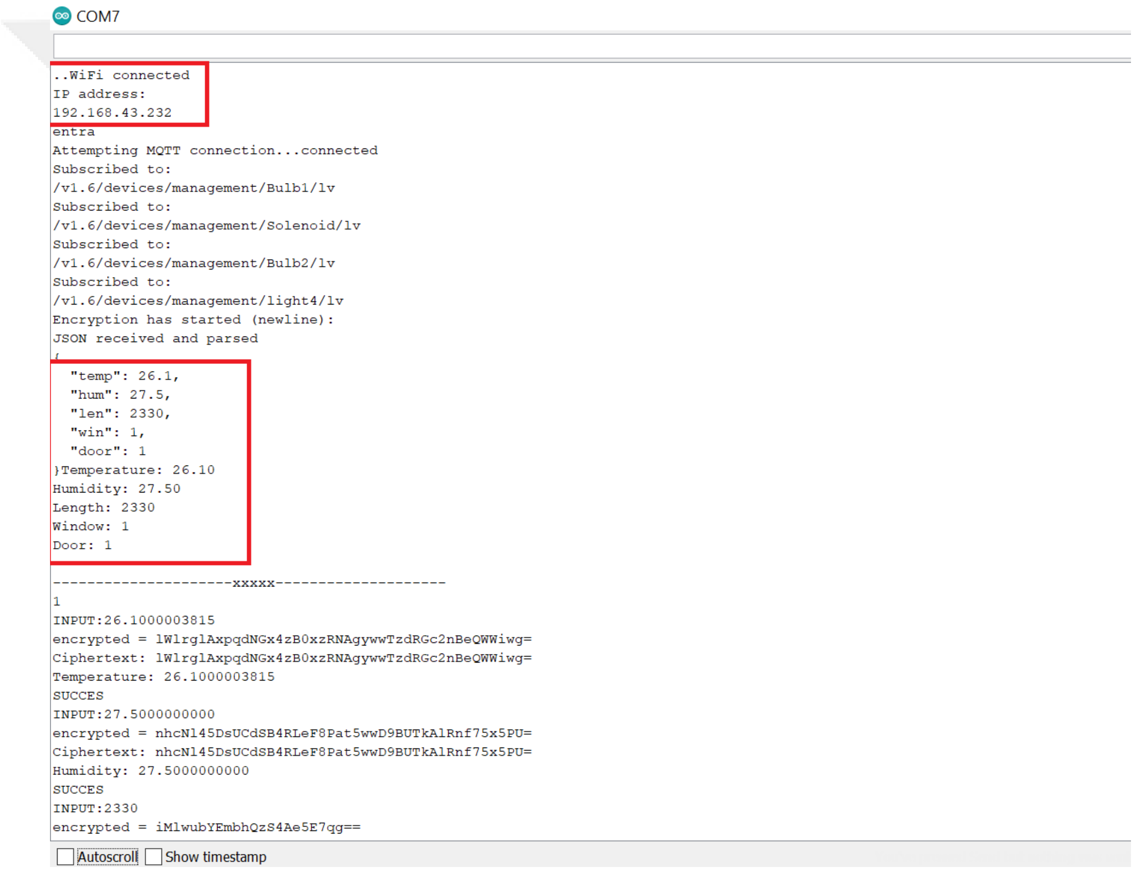

Figure 4.2. Sensors reading data on the serial monitor

The figure above shows that the NodeMCU ESP8266 Wi-Fi module is connected to the internet and obtained the IP address. Furthermore, it displays the sensor's readings data, which are the temperature and humidity degrees reading in real-time, and their incoming data comes from the DHT22 sensor. The distance of an object measured by the ultrasonic sensor is also displayed on the serial monitor identified as a length in the Arduino IDE code. The door status and window status named "win" and "door" in the code to show and monitor the status of the door and window if it is opened or closed, and their data comes from IR1 and IR2 sensors. When the door and window are closed, it displays "0", but when it is open, it displays "1" as can be seen from the figure above.

4.2. Software System

In this research, we use the software part with the hardware system to control and monitor the smart home system through the internet. Arduino IDE software has been used as the interface between the software and hardware of this project. The microcontroller needs a program to operate and execute the process associated with performed design. It is easy to verify and compile after writing the code. The Blynk android application and Ubidots IoT platform web page were created to monitor the actual sensor value, control electronic devices, and see the information from the NodeMCU ESP8266. Also, we applied the AES algorithm to encrypt and decrypt the smart home system data throughout NodeMCU ESP8266 and the server.

4.2.1. Blynk Platform with Android Application

The Blynk application makes it easy to read sensor data and easily control appliances. The android version of the Blynk application with a custom-built layout and buttons was used for the user interface to monitor and control different linked items.

With the NodeMCU ESP8266 controller chip, the relay module device connected to the home appliances to be operated and connected via Wi-Fi. The home appliances can be operated from any remote location and at any time by pressing the button on your smartphone. There are three separate tabs for three appliances, namely Light 1, Solenoid, and Light 2. These loads can be controlled by pressing the Blynk application on the respective android smartphone on the ON/OFF button widget. This is achieved after the device has been enabled and connected to Wi-Fi. If the internet connection is lost or a weak signal at any moment, it also affects the performance of the system. Sensors collect various environmental conditions and transmit them to the microcontroller, which separately processes the data transmitted by each sensor and sends the acquired data to the Blynk server simultaneously. The user can access the readings of each sensor from anywhere at any time. The application consisting of multiple sensors; the application uses three sensors to calculate different environmental parameters, a temperature and humidity measurement by DHT22 sensor, an IR sensor for motion detection, and an ultrasound sensor for distance measurement.

So, in our android Blynk application results, there are two display widgets named “Temperature” and “Humidity” from the DHT22 sensor that displayed the temperature and humidity degree. Two widgets named “Door status” and “Window status” from IR1 and IR2 sensors show the status of the door and window. Finally, the distance is measured by the “Length” widget from the ultrasonic sensor. Figure 4.3 shows the Blynk android application before running and pressing the play button on the Blynk application to show the output. The screenshots of the results of the designed system after running obtained on the android application show the results of controlling three appliances and monitor four sensor modules, as can be seen from Figure 4.4.

Figure 4.3. Blynk application before running

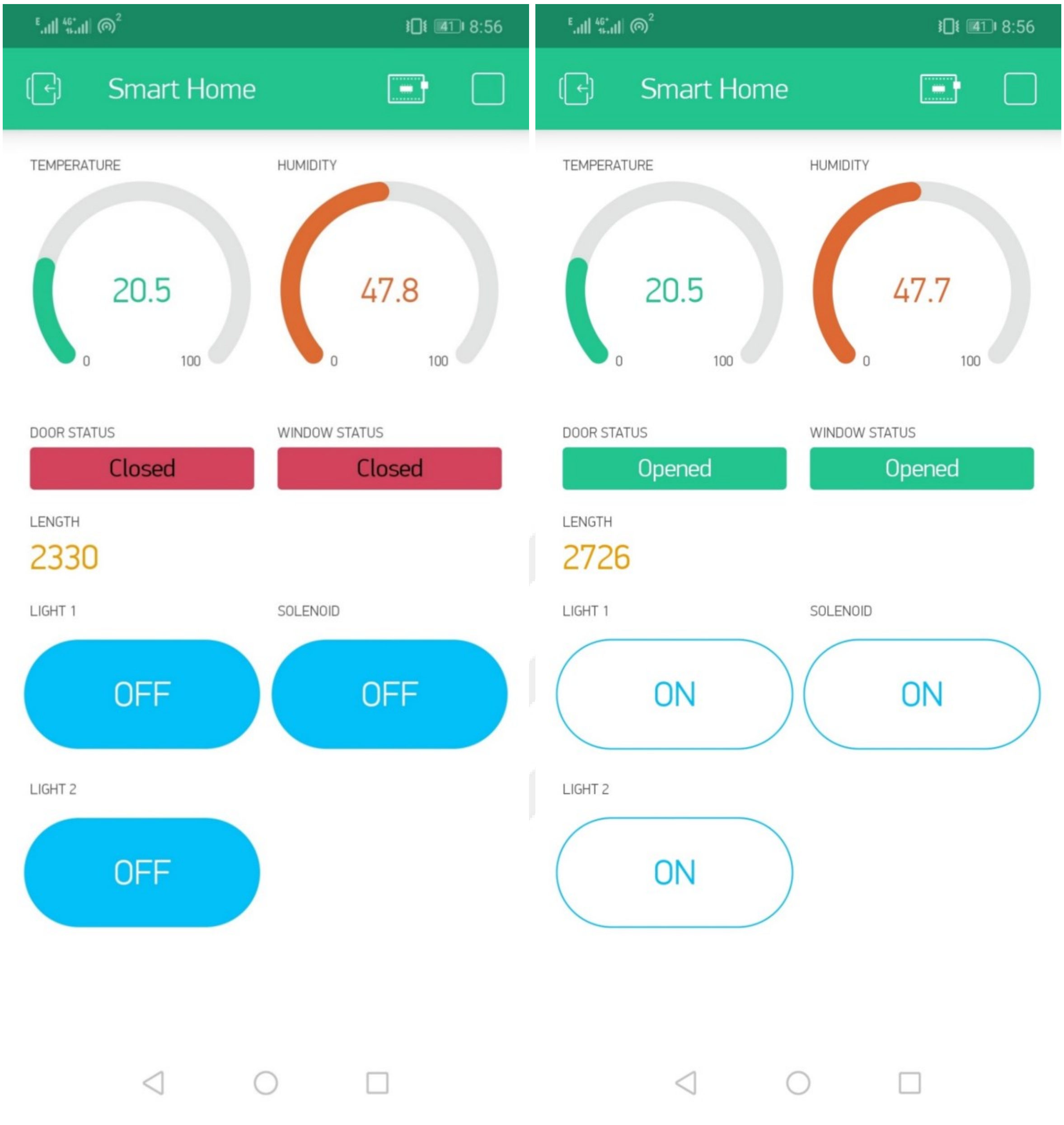

Figure 4.4. Screenshots of Blynk application after running

4.2.2. Ubidots IoT Platform Webpage

The results are obtained and can be visualized on the internet created on the Ubidots IoT platform after hardware and software implementation of the system. Ubidots provides developers with a framework that allows them to easily capture sensor data and transform it into useful information and easily control devices. To send data to the cloud from any internet-enabled device, use the Ubidots platform. The results of this application were successfully obtained the smart home system based on IoT.

The controller NodeMCU ESP8266 connects the system with the Ubidots platform by its Wi-Fi for establishing remote monitoring and controlling continuously with a laptop or tablet. The ESP8666 processes the data received by the sensors and the loads and then sent to the Ubidots web server via the MQTT protocol, where a dashboard shows the collected data transducer variables. The dashboards use widgets such as metrics, indicators, tanks, thermometers, and switches to display the data.

The dashboard generated in our Ubidots platform uses thermometer and tank widgets to visually show the temperature and humidity degrees reading in real-time incoming data from DHT22. The indicators used on the IoT platform dashboard named door status and window status to show and monitor the status of the door and window if it is opened or closed, and their data comes from IR1 and IR2 sensors. The distance of an object measured by the ultrasonic sensor using a metric widget identified as a length on the dashboard and three switches generated on Ubidots platform dashboard to switch ON/OFF relay board connected to bulb1 bulb2 and solenoid to unlock the door. Figure 4.5 shows the results uploaded to the Ubidots platform for our smart home application.

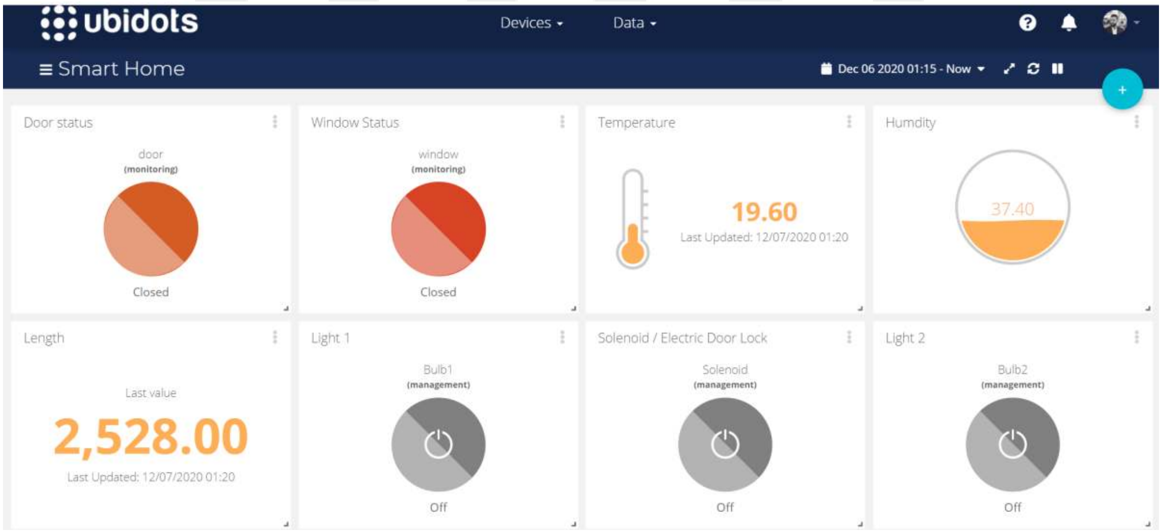

Figure 4.5. Ubidots software webpage of the system

As shown in Figure 4.6, we note that the status of the door and window has changed from "Closed" to "Opened" after motions detected by IR1 and IR2 sensors, and it also shows that the distance measurements reading changed using an ultrasonic sensor. In the dashboard, bulbs, and solenoid electric door lock status also changed from switching OFF to ON.

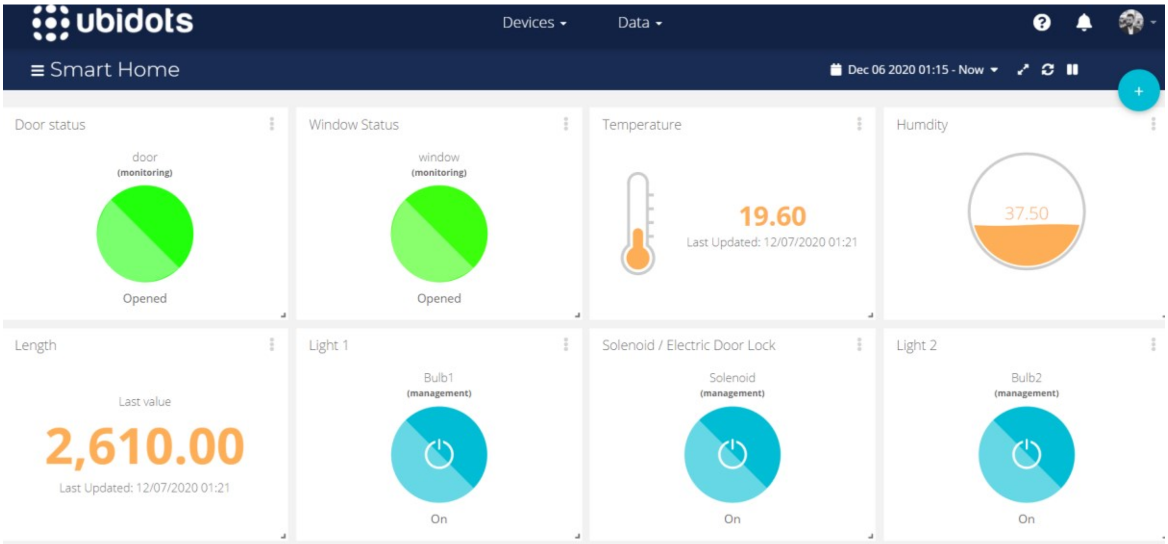

Figure 4.6. Ubidots software webpage of the system after sensors and loads status changed

4.2.3. Algorithm Result

The results of the encryption and encoding process of the NodeMCU ESP8266 side and decryption and decoding process of the server-side will be presented in this chapter of the thesis. The AES provides features that allow encryption and decryption of data. Three essential criteria, i.e., security, costs, and implementation, can be achieved. Encryption is a mechanism through which the original data is transformed into an encrypted form, and decryption is a method by which the encrypted data is transferred back to its original form. This method for each encryption depends on the size of the big data that needs to be processed, and the decryption process also depends on the size of the large data that needs to be processed.

The result of the performed technique was captured on the serial monitor while adapting it into the IoT device. Figure 4.7 shows the smart home system data after the encryption and decryption by AES technique was applied to the performed system. The illustrated result in the figure below consists of various processes and techniques to achieve the desired output; after all, the data was encrypted/encoded and decrypted/decoded throughout the NodeMCU ESP8266 webserver. As can be seen from the result, different data sizes for the encryption have been processed. Firstly, the original input data is encrypted into a string using the AES technique; only the system knows what this generated data means. Secondly, it is sent to the ciphertext before it sends out to the server. After all, data had been encrypted and encoded, and the server receives the strings. The strings received by the server are decrypted and decoded to convert the data to its original form.

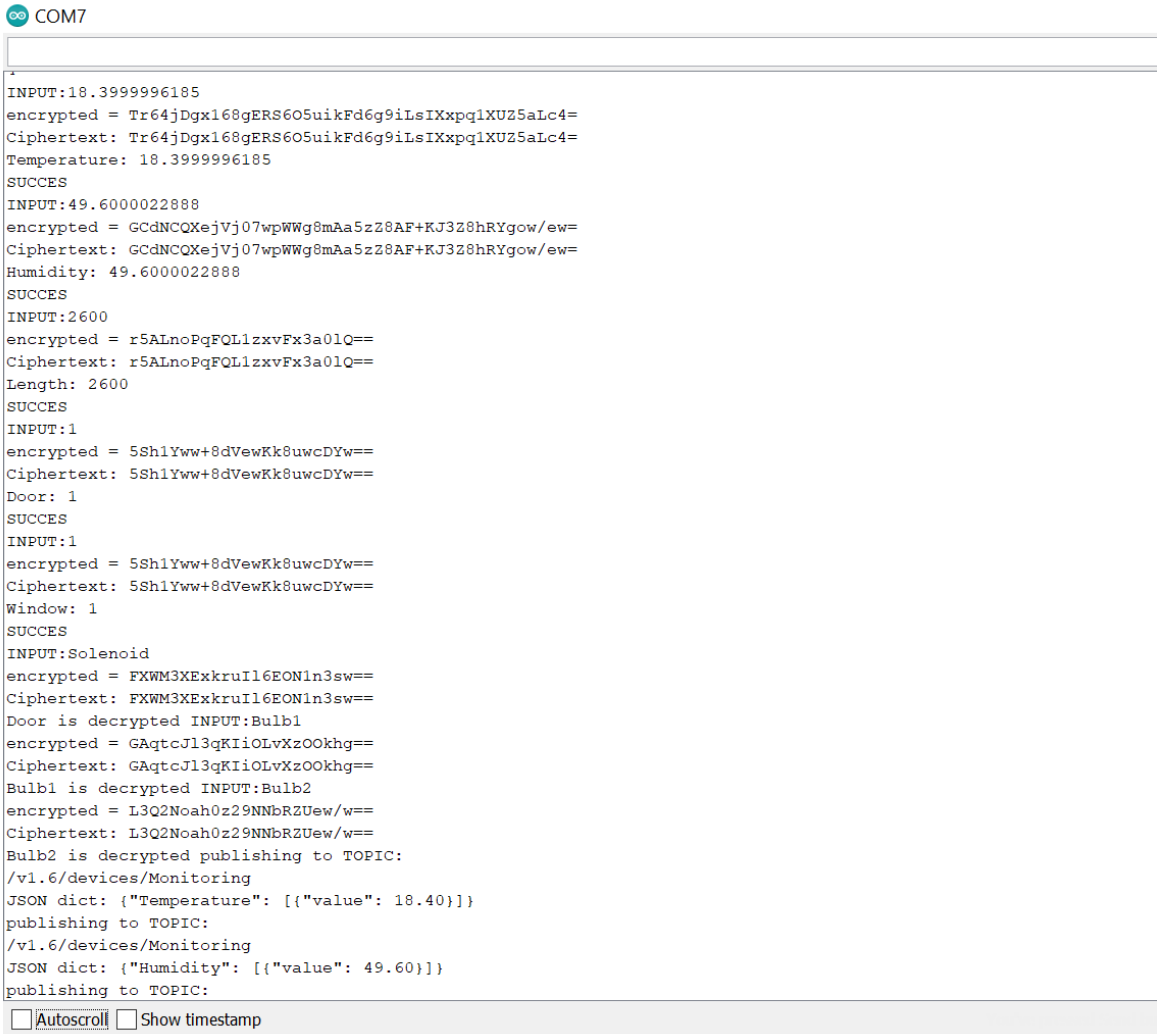

Figure 4.7. The encrypted and decrypted data by AES technique on the serial monitor screen

5. CONCLUSIONS

Smart Home system with NodeMCU ESP8266 module and Arduino Uno board based on IoT can be built with different hardware and software support modules to be incorporated into a smart home system. This project successfully designs a smart home system that enables homeowners to control and monitor appliances remotely from smartphone applications and web applications over the internet. The goal of the study was to develop an intelligent home system that controls home appliances like doors and lights with a simple mobile application called Blynk and Ubidots IoT platform webpage, Furthermore, recording the sensor module data and sending the real-time data to the Blynk application and Ubidots platform. All the data are sent to the Blynk server and Ubidots cloud through NodeMCU ESP8266. Besides, our system was developed for more security and data encryption by using the AES technique. The key concept was to encrypt all data between the ESP8266 NodeMCU and the server. Since it is a very reliable cryptography technique, AES-128 was proposed. In securing data communication, which is encrypting and encoding data and decrypting and decoding data, two processes were involved.

This project mainly targeted the application of IoT with the sensors, relay kit, Arduino Uno, NodeMCU ESP8266 module, Ubidots IoT platform, and Blynk application. The project accomplished the objective, and the project can be considered a success. Furthermore, the project covers the main feature of the smart home system, such as light control, door opening, temperature and humidity measurements, etc. even with high voltage, but with strong precautions and safety, the powerful relay switches let us operate.

In the future, the proposal involves a range of changes and improvements to the existing system to increase the accuracy and reliability of the overall system. This system is only a fundamental framework of another system that is complete. All the basic needs of a typical house have been fulfilled. The tasks we have completed are not the only tasks that the components will perform. There are several other scopes for this project. More devices with a powerful relay module can be connected to this framework. Use wireless connection without internet to control and monitor household devices when the user locates at home. We may add sensors for which the appliances work automatically. We can add any appliances that are running when you are not at home to get a warning via SMS to your phone so that we can properly and immediately respond to the appliances. Also, in the smart home, we can incorporate voice recognition and a sensor that automatically adjusts the fan speed based on room temperature and ready-to-use IoT appliances.

REFERENCES

[1] K. Bing, L. Fu, Y. Zhuo, and L. Yanlei, "Design of an Internet of Things-based smart home system," in 2011 2nd International Conference on Intelligent Control and Information Processing, 2011, pp. 921-924.

[2] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, "A survey based on Smart Homes system using Internet-of-Things," in 2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC), 2015, pp. 0330-0335.

[3] C. Paul, A. Ganesh, and C. Sunitha, "An overview of IoT based smart homes," in 2018 2nd International Conference on Inventive Systems and Control (ICISC), 2018, pp. 43-46.

[4] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," Journal of Cleaner Production, vol. 140, pp. 1454-1464, 2017.

[5] L. M. Satapathy, S. K. Bastia, and N. Mohanty, "Arduino based home automation using Internet of things (IoT)," International Journal of Pure and Applied Mathematics, vol. 118, pp. 769-778, 2018.

[6] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of things," International journal of communication systems, vol. 25, p. 1101, 2012.

[7] L. Calderoni, A. Magnani, and D. Maio, "IoT Manager: a case study of the design and implementation of an Open Source IoT Platform," in 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019, pp. 749-754.

[8] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang, "A vision of IoT: Applications, challenges, and opportunities with china perspective," IEEE Internet of Things journal, vol. 1, pp. 349-359, 2014.

[9] H. Chen, X. Jia, and H. Li, "A brief introduction to IoT gateway," in IET international conference on communication technology and application (ICCTA 2011), 2011, pp. 610-613.

[10] P. Gokhale, O. Bhat, and S. Bhat, "Introduction to IOT," International Advanced Research Journal in Science, Engineering and Technology, vol. 5, pp. 41-44, 2018.

[11] K. Agarwal, A. Agarwal, and G. Misra, "Review and Performance Analysis on Wireless Smart Home and Home Automation using IoT," in 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2019, pp. 629-633.

[12] R. Das, A. Tuna, S. Demirel, and M. K. Yurdakul, "A survey on the Internet of Things solutions for the elderly and disabled: Applications, prospects, and challenges," International Journal of Computer Networks and Applications, vol. 4, pp. 1-9, 2017.

[13] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future generation computer systems, vol. 29, pp. 1645-1660, 2013.

[14] X. Mao, K. Li, Z. Zhang, and J. Liang, "Design and implementation of a new smart home control system based on internet of things," in 2017 international smart cities conference (ISC2), 2017, pp. 1-5.

[15] V. D. Vaidya and P. Vishwakarma, "A comparative analysis on smart home system to control, monitor and secure home, based on technologies like GSM, IOT, Bluetooth and PIC Microcontroller with

ZigBee Modulation," in 2018 International Conference on Smart City and Emerging Technology (ICSCET), 2018, pp. 1-4.

[16] T. Adiono, S. Harimurti, B. A. Manangkalangi, and W. Adijarto, "Design of smart home mobile application with high security and automatic features," in 2018 3rd international conference on intelligent green building and smart grid (IGBSG), 2018, pp. 1-4.

[17] V. Govindraj, M. Sathiyanarayanan, and B. Abubakar, "Customary homes to smart homes using Internet of Things (IoT) and mobile application," in 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), 2017, pp. 1059-1063.

[18] D. Geneiatakis, I. Kounelis, R. Neisse, I. Nai-Fovino, G. Steri, and G. Baldini, "Security and privacy issues for an IoT based smart home," in 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2017, pp. 1292-1297.

[19] E. J. Lee and S. J. Park, "A Framework of Smart-Home Service for Elderly's Biophilic Experience," Sustainability, vol. 12, p. 8572, 2020.

[20] J.-H. Han, Y. Jeon, and J. Kim, "Security considerations for secure and trustworthy smart home system in the IoT environment," in 2015 International Conference on Information and Communication Technology Convergence (ICTC), 2015, pp. 1116-1118.

[21] M. Larsson and L. Lipkin, "On Security Threats and Solutions for the Future Smart Home Exploring the Viability of Intrusion Detection in a Centralized Smart Hub," 2019.

[22] S. Sezer, "T1C: IoT Security:-Threats, Security Challenges and IoT Security Research and Technology Trends," in 2018 31st IEEE International System-on-Chip Conference (SOCC), 2018, pp. 1-2.

[23] P. Mahajan and A. Sachdeva, "A study of encryption algorithms AES, DES and RSA for security," Global Journal of Computer Science and Technology, 2013.

[24] S. Heron, "Advanced encryption standard (AES)," Network Security, vol. 2009, pp. 8-12, 2009.

[25] W. Li, T. Logenthiran, V.-T. Phan, and W. L. Woo, "A novel smart energy theft system (SETS) for IoT-based smart home," IEEE Internet of Things Journal, vol. 6, pp. 5531-5539, 2019.

[26] L. Jiang, D.-Y. Liu, and B. Yang, "Smart home research," in Proceedings of 2004 international conference on machine learning and cybernetics (IEEE Cat. No. 04EX826), 2004, pp. 659-663.

[27] A. Jacobsson, M. Boldt, and B. Carlsson, "A risk analysis of a smart home automation system," Future Generation Computer Systems, vol. 56, pp. 719-733, 2016.

[28] Y. Jie, J. Y. Pei, L. Jun, G. Yun, and X. Wei, "Smart home system based on iot technologies," in 2013 International conference on computational and information sciences, 2013, pp. 1789-1791.

[29] A. H. Hassan, A.-f. Y. Awad, and A. H. El-Amin, "Smart Home: Home Monitoring & control System using Simple Mobile App," Sudan University of Science and Technology, 2015.

[30] R. Daş and G. Tuna, "Machine-to-machine communications for smart homes," International Journal of Computer Networks and Applications, vol. 2, pp. 196-202, 2015.

[31] L. P. Akhtar, "IoT based autonomous office alarm and access control system," Master's Thesis, Technical University of Denmark, 2017.

[32] I. Mohammed and E. Duman, "Implementation of a smart house application using wireless sensor networks," in 9th International Conference on Networks & Communications. Computer Science & Information Technology, 2017, pp. 53-70.

[33] S. Pirbhulal, H. Zhang, M. E. E Alahi, H. Ghayvat, S. C. Mukhopadhyay, Y.-T. Zhang, et al., "A novel secure IoT-based smart home automation system using a wireless sensor network," Sensors, vol. 17, p. 69, 2017.

[34] D. Mandal, M. Hassan, and J. S. Joydhar, "Implementation of an IoT based home automation system using real time sensors," BRAC University, 2018.

[35] M. Almosawi and K. Djupsjö, "IoT Security Applied on a Smart Door Lock Application," ed, 2018.

[36] J. Alonso, C. Bayona, O. Rojas, M. Terán, J. Aranda, H. Carrillo, et al., "Iot solution for data sensing in a smart campus using smartphone sensors," in 2018 IEEE Colombian Conference on Communications and Computing (COLCOM), 2018, pp. 1-6.

[37] T. Qiang, G. Guangling, C. Lina, and W. Han, "Nodemcu-based Low-cost Smart Home Node Design," in IOP Conference Series: Materials Science and Engineering, 2018, p. 012013.

[38] R. Mahindar, M. Prakash, S. Ghosh, S. Mukherjee, and R. Ghosh, "IoT-based home appliances control system using NodeMCU and Blynk server," International Advanced Research Journal in Science, Engineering and Technology, vol. 5, pp. 16-22, 2018.

[39] H. Singh, V. Pallagani, V. Khandelwal, and U. Venkanna, "IoT based smart home automation system using sensor node," in 2018 4th International Conference on Recent Advances in Information Technology (RAIT), 2018, pp. 1-5.

[40] H. Durani, M. Sheth, M. Vaghasia, and S. Kotech, "Smart automated home application using IoT with Blynk app," in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 393-397.

[41] R. Majhi, "IoT: Home Automation," 2019.

[42] A. H. Imam, "A Simple Smart Home Based On Iot Using Nodemcu And Blynk," Universitas Muhammadiyah Surakarta, 2019.

[43] K. T. Wai, N. P. Aung, and L. L. Htay, "Internet of Things (IoT) Based Healthcare Monitoring System using NodeMCU and Arduino UNO," 2019.

[44] C. R. Aldawira, H. W. Putra, N. Hanafiah, S. Surjarwo, and A. Wibisurya, "Door security system for home monitoring based on ESp32," Procedia Computer Science, vol. 157, pp. 673-682, 2019.

[45] B. Bohara, S. Maharjan, and B. R. Shrestha, "IoT based smart home using Blynk framework," arXiv preprint arXiv:2007.13714, 2020.

[46] A. Surriani, A. Pradana, M. Arrofiq, J. Putra, M. Budiyanto, and L. Subekti, "Design of Power Monitoring Application," in IOP Conference Series: Materials Science and Engineering, 2020, p. 012069.

[47] A. G. Smith, "Introduction to Arduino," Alan G. Smith, 2011.

[48] L. Louis, "working principle of Arduino and u sing it," International Journal of Control, Automation, Communication and Systems (IJCACS), vol. 1, pp. 21-29, 2016.

[49] Y. S. Parihar, "Internet of Things and Nodemcu," Revista de Tecnologías Emergentes e Investigación Innovadora, vol. 6, pp. 1085-1088, 2019.

[50] A. Patel and P. Devaki, "Survey on NodeMCU and Raspberry pi: IoT," International Research Journal of Engineering and Technology (IRJET), vol. 6, pp. 5101-5105, 2019.

[51] W. Adhiwibowo, A. F. Daru, and A. M. Hirzan, "Temperature and Humidity Monitoring Using DHT22 Sensor and Cayenne API," Jurnal Transformatika, vol. 17, pp. 209-214, 2020.

[52] A. Dimitrov and D. Minchev, "Ultrasonic sensor explorer," in 2016 19th International Symposium on Electrical Apparatus and Technologies (SIELA), 2016, pp. 1-5.

[53] B. Mustapha, A. Zayegh, and R. Begg, "Multiple sensors based obstacle detection system," in 2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012), 2012, pp. 562-566.

[54] T. Adiono, S. Fuada, S. F. Anindya, I. G. Purwanda, and M. Y. Fathany, "IoT-enabled door lock system," Int. J. Adv. Comput. Sci. Appl, vol. 10, pp. 445-449, 2019.

[55] N. Sadikin, M. Sari, and B. Sanjaya, "Smarthome Using Android Smartphone, Arduino uno Microcontroller and Relay Module," in Journal of Physics: Conference Series, 2019, p. 012035.

[56] P. Serikul, N. Nakpong, and N. Nakjuatong, "Smart farm monitoring via the Blynk IoT platform: case study: humidity monitoring and data recording," in 2018 16th International Conference on ICT and Knowledge Engineering (ICT&KE), 2018, pp. 1-6.

[57] N. S. Mohammed and N. H. Selman, "Home Energy Management and Monitoring Using Ubidots Platform," Al-Furat Journal of Innovation in Electronic and Computer Engineering, vol. 1, pp. 14-21, 2020.

[58] J. Daemen and V. Rijmen, "Reijndael: The Advanced Encryption Standard," Dr. Dobb's Journal: Software Tools for the Professional Programmer, vol. 26, pp. 137-139, 2001.

[59] D. Selent, "Advanced encryption standard," Rivier Academic Journal, vol. 6, pp. 1-14, 2010.

[60] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, et al., "Report on the development of the Advanced Encryption Standard (AES)," Journal of Research of the National Institute of Standards and Technology, vol. 106, p. 511, 2001.

[61] W. Stallings, "The advanced encryption standard," Cryptologia, vol. 26, pp. 165-188, 2002.

CURRICULUM VITAE

Taha Abdullah ABABAKER

[redacted]

[redacted]

[redacted]

RESEARCH EXPERIENCES

- ✓ Computer Programming languages available (C programming language)
- ✓ Computer programs: (C, C++, Matlab, Embedded system, Cisco packet tracer, Computer networking, Computer architecture, Java programming language, IoT, Python)

WORK EXPERIENCE

[redacted]