Hindawi
Journal of Computer Networks and Communications
Volume 2023, Article ID 8571506, 12 pages
https://doi.org/10.1155/2023/8571506

Research Article

SECHA: A Smart Energy-Efficient and Cost-Effective Home Automation System for Developing Countries

Abdulhafiz Ahmed Mustofa,[1] Yohannes Alemayehu Dagnew,[1] Prabhakar Gantela,[2] and M. Javed Idrisi[3]

[1]*Department of Microelectronics Engineering, School of Physical Science and Technology, Yangzhou University, Yangzhou, China*
[2]*Department of Information Technology, School of Computing and Informatics, Mizan-Tepi University, Tepi, Ethiopia*
[3]*Department of Mathematics, College of Natural and Computational Science, Mizan-Tepi University, Tepi, Ethiopia*

Correspondence should be addressed to M. Javed Idrisi; javed@mtu.edu.et

Received 7 October 2022; Revised 15 February 2023; Accepted 23 February 2023; Published 1 March 2023

Academic Editor: Giovanni Pau

Home automation systems are gaining a lot of attraction globally and changing the way we live. They simplify our lives, reduce workloads, improve home safety and security, and pave the way for newer developments. It is no wonder why these systems are in such high demand and why modernization is needed to keep up with consumer needs. Nevertheless, utilizing home automation system technology can be energy-intensive and costly—especially for middle-class families in developing countries. In this article, we discuss SECHA, a smart, energy-efficient, and cost-effective home automation system. It empowers users to automate their homes with IoT regardless of the residence type. SECHA is developed with the goal of being energy-efficient, simple to use, and open-source for everyone's benefit. SECHA has developed a low-cost smart home automation system that incorporates Wi-Fi and GSM technology, enabling remote monitoring and control of appliances through an Android application. This solution enables users to easily monitor and manage their homes. An automation system has been developed using an ESP32 microcontroller equipped with Wi-Fi and GSM SIM800. This impressive setup is further enhanced by the integration of several sensors that enable monitoring of temperature, humidity, movement, and other aspects at home.

1. Introduction

The Internet of Things (IoT) is already a part of our lives, with billions of devices connected to it, and the projection is that this number will only grow in the years to come. This advancement can bring many benefits and conveniences for us in our daily lives. A rapid increase in the global number of IoT devices is expected, from 9.7 billion in 2020 to 29 billion by 2030 [1]. While many developed countries are already leveraging the benefits of IoT technology, most developing and underdeveloped nations are still at the beginning stages of implementation. Despite its relatively recent arrival, IoT has started to open doors and provide immense possibilities for a range of new possibilities that can have a huge impact on our lives. Due to the lack of existing infrastructure, the growth of IoT is hindered by several challenges, the most important of which are the relatively high cost of IoT devices and equipment, a lack of strength, and unstable Internet connections [2, 3]. For example, in a developing country such as Ethiopia, power cuts are frequent, and this is expected to limit the country's population's access to electricity [4, 5]. The high energy demands of IoT devices combined with human abuse, such as turning on lights in an empty room, make this limited energy scarce. It is essential to design and implement cost-effective and energy-efficient IoT systems that are easily accessible and flexible. Understanding how to use the Internet of Things (IoT) can help us reduce energy loss and thus potentially become more efficient [6].

Several other home automation projects are also underway in various countries. All of them are unique in terms of design, features, equipment, elements, and algorithms. They were developed in response to specific requirements and component availability in the respective regions. Some are cheap, while others are quite expensive. The availability of hardware and software used in developing countries was also a requirement. Despite the fact that most of these research papers are useful in some form or the other, we feel compelled to develop an all-inclusive, simple, and easy-to-implementIoT-based home automation system.

As a result, we developed SECHA, a smart, energy-efficient, and cost-effective home automation system in this article. It offers a straightforward and practical method for utilizing the majority of IoT home automation and security technologies. We plan to present a home automation architecture in SECHA that makes use of a single microcontroller, mobile connection, and an Android app. Even when a user is not online, SECHA uses quicker SMS communication offered by Android to deliver emergency messages from anywhere in the world. In addition to smart home automation, we will focus on an energy consumption management solution made up of low-voltage sensors and components that will save a significant amount of energy while also allowing consumers to avoid excess energy consumption by remotely managing equipment. It also prioritizes affordability by utilizing low-cost sensors and components. This can help users save money by reducing appliance energy waste. All costs are kept as low as possible, making it affordable to middle-class families in developing countries, such as Ethiopia.

The objective is to design a prototype using economical and power-efficient equipment that establishes wireless remote control over a network of home appliances. The application software is designed to run on easily accessible Android smartphones and includes voice command control and the ability to check the equipment's status directly from the application. The system can be used in any residence, business, institution, or mall and has a wide range of uses. Smart house devices can be controlled via the Internet or by manually connecting them to switches, and the system can notify users or the fire department in case of an emergency even when no Internet connection is available.

By applying efficient scheduling skills, users can save money as well as conserve energy. The Internet of Things home energy management system provides consumers with optimal results at the lowest possible cost. We designed SECHA to be easily upgradeable and degradable based on user preferences, as some functionality is not applicable to all users. This type of advantage can not only save energy but also unnecessary expenditures, making it more efficient. As a result, users can easily and cost-effectively implement the IoT-basedenergy-saving home automation system concepts discussed in this article.

2. An Overview of Research Papers on Low-Cost IoT

This section is devoted to a brief overview of subsequent IoT papers that have been chosen because they state low cost as a requirement. As we are going to see, this symptom is more often than not linked to a perspective of the issue, which in general gets ignored, and as often as not entirely justified. Most of the papers included in this section are from developing countries, indicating that IoT is considered to improve their quality of life, as well as fill the gap with developed countries. Home automation is the use of technology in the home environment to provide convenience, comfort, security, and energy efficiency to its occupants. Many research papers present various methods for home automation. Each of these has some distinctive features as well as some drawbacks. Some of these papers are discussed further below:

The authors of a study [7] proposed energy-efficient home automation using IoT. The study implements an intelligent home automation approach using the Internet of Things, which would greatly improve the current energy usage difficulties. To do this, a low-cost, low-power ESP8266 with an integrated Wi-Fi module is used to control a relay channel. The major drawback to utilizing ESP8266 is that its near cousin, the ESP32, performs better and has several features that ESP8266 lacks while costing almost as much.

The author of reference [8] provides a low-cost, effective automatic energy management system for households using an Arduino UNO microcontroller board. It combines instruments to deal with natural disasters such as fire in addition to offer a cost-effective solution for managing household energy. The sole drawback of utilizing Arduino UNO is that, in comparison to ESP32, it consumes less power, making it unsuitable for use with a battery. Even without the Wi-Fi shield, the ESP32 is significantly less expensive than the Arduino UNO—about five times less so. Similarly, the authors of reference [9] propose an Arduino-based greenhouse monitoring and control system. The DHT11, soil moisture, LDR, and PH sensors are the primary sensors used in this project, and they provide precise values for temperature, humidity, moisture content, light intensity, and soil PH. This system is designed to control and monitor environmental parameters in a greenhouse via SMS sent from anywhere on the GSM network. Although the majority of the sensors used are low-cost and energy-efficient and the system also operates offline via GSM800; using Arduino is not the most cost-effective or energy-efficient method, as it is relatively expensive and difficult to find in developing countries. The article focuses on using IoT for plant growth monitoring and control rather than general home automation.

A similar study on the development of an IoT-based system for monitoring the electrical energy consumption of smart and rental houses was conducted in Tanzania [10]. Although they used the ESP32 as their main microprocessor and GSM for offline messaging, the article only aims to add value to smart technology rental house users by providing them with a system that provides them with full detailed information about each appliance or power line in their houses, as well as the power consumption of each room. Even though these will assist users in understanding total house power consumption and eliminating unnecessary or unused devices from the consumption line, the article falls short of providing a solution for a diverse range of communities in developing countries.

A comprehensive strategy was adopted in reference [11] to create a low-cost IoT. The holistic idea that the existence of a single low-cost component does not ensure that the same property holds true for the entire project served as the writers' driving force. Although our main focus was on home automation and security systems, which can be used in any developing or underdeveloped country, this was consistent with the system concepts we had in place.

3. SECHA's Architecture

In this section, we provide an overview of SECHA, a smart, energy-efficient, and cost-effective home automation system for any average-earning household in developing or underdeveloped countries by addressing the main issues facing the IoT: affordability, power outages, and unstable internet connections. At SECHA, we attempted to achieve this by bringing together different technologies from different companies that offer low-cost, power-efficient, and easy-to-implement devices and integrating them into a single system that seems to work together. We also tried to reduce the number of sensors and devices required by incorporating most of the home automation while keeping the cost low.

SECHA is built around an ESP32 microcontroller, which processes data from multiple sensors and sends it to a real-time database, which then sends it to the user's Android app to receive instructions. In this project, we created a general prototype that can perform most of the basic home automation tasks by using a single ESP32 as the main microcontroller and connecting it to all of our sensors and gadgets to create a home automation system. We attempted to demonstrate that even with a limited budget, scarce electricity, and an unstable internet connection, as is common in developing countries, an IoT home automation and security system can be implemented. We also designed the SECHA system to be easily upgradeable and degradable, which means that a user can easily add more features by connecting a new sensor or appliance to any relay module that is connected to ESP32, or they can remove unnecessary features based on their preferences, giving SECHA more flexibility.

SECHA's fundamental architecture is divided into multiple phases, which include sensors, gateway connection, processing of data, cloud, and user interface application software (Figure 1). To begin, physical devices such as sensors, appliances, and microcontrollers collect and interpret data from their surroundings. These data are then sent and processed in the cloud, which then sends them to the user using a UI and receives the data as a command. This command is then passed to the microcontroller, which commands the sensors and appliances. In the IoT architecture, microcontrollers and sensors operate as transmitters, transforming energy from one form to another. In our prototype, we used ESP32, ESP32-CAM, SIM800 Module, PIR Sensor, DHT11, MQ2 Gas Sensor, and LDR Sensor as our main components, and we developed an Android app that controls home automation using the Firebase real-time database.

3.1. Why ESP32? The ESP32 microcontroller is a low-cost, low-power system-on-chip microcontroller that supports Wi-Fi and Bluetooth and has a highly integrated structure powered by dual-core microprocessors. The ESP32 is an excellent choice for IoT devices due to its performance and low cost. The ESP32, a microcontroller board used to connect all of the sensors, is at the center of our design. The ESP32 board is programmed with the project's source code. The operational voltage range of the ESP32 is 2.2 to 3.6 V, and it has on-chip memory that can be used to store the source code directly on the chip. The ESP32 will deliver 3.3 V to the chip during typical operation [12]. The ESP32 microcontroller has two cores and can execute many tasks at once. Internal sensors, such as capacitive touch and hall effect sensors, are also included. The ESP32 chip's key benefits are its broad deployment capabilities and compatibility with Wi-Fi protocols. The ESP32 is a 32-bithexa-core microcontroller developed for Internet of Things (IoT) devices. It boasts a large computational capability for such a small chip, as well as on-chip SRAM memory for data and program instructions. It also supports external memory, making it appropriate for more demanding activities such as connecting to cameras, voice recognition, streaming data, image recognition, and IoT sensors. You may read more technical information on why the ESP32 is the best microcontroller at [13, 14].

3.1.1. ESP32 Sleep Modes. When the ESP32 is not in use, it may enter a power-saving mode called "sleep mode," which stores all data in RAM. Any unneeded peripherals will be turned off, and the RAM will be given enough power to maintain its contents. The ESP32 features five programmable power modes as a result of its advanced power management: active mode (160~260 mA), modem sleep mode (3~30 mA), light sleep mode (~0.8 mA), deep sleep mode (~6.5 μA), and hibernation mode (~4.5 μA). The chip may transition between these several power modes depending on the power needed. The ESP32 can operate in these modes at currents even lower than 4.5 μA. ESP32 power-management technology is efficient and adaptable, allowing for the optimal balance of power consumption, wakeup latency, and available wakeup sources. Users can choose from five predetermined power modes of the main processors to meet the application's demands. Furthermore, commands may be conducted by the ultralow-power coprocessor (ULP coprocessor) while the main processors are in deep-sleep mode to conserve power in power-sensitive applications [15].

3.1.2. ESP32 vs. Arduino Uno vs. ESP8266. The ESP32 can run for an astonishingly long time on batteries, making it perfect for developing countries, whereas the Arduino Uno has inefficient power consumption, making it unsuitable for use with a battery. Additionally, compared to the Arduino Uno, it features more GPIO pins. Additionally, the ESP32 has Wi-Fi and Bluetooth built in, removing the need for extra radio modules that are included on most Arduino boards. The ESP32 is also around five times cheaper than the

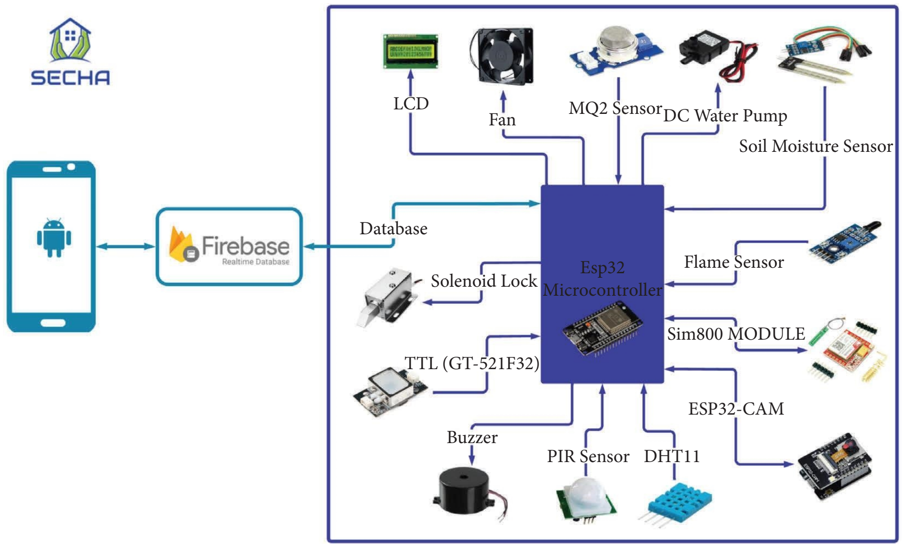


Figure 1: SECHA's architecture.

Arduino Uno without taking into account the external peripherals required for the Arduino Uno.

The ESP32 also surpasses its close cousin, the ESP8266, thanks to extra capabilities such as a CPU core, faster Wi-Fi, Bluetooth (BLE), touch sensitivity pins, built-in hall effect sensors, and a temperature sensor. The ESP32 has more GPIO pins than the ESP8266. The ESP32 features ten capacitive GPROs that sense touch and may be used to trigger events while maintaining around the same price as the ESP8266.

3.2. Why ESP32-CAM? The ESP32-CAM is a simple, low-cost development board based on the ESP32. It has a built-in TF card slot and an OV2640 camera. The ESP32-CAM is suitable for a wide range of smart IoT applications, such as wireless video monitoring, Wi-Fi picture upload, and QR recognition. Both of the high-performance32-bit LX6 CPUs and all three Wi-Fi, conventional Bluetooth, and low-power BLE are integrated. Its main frequency adjustment spans from 80 MHz to 240 MHz, and additional features include an on-chip sensor, a hall effect sensor, a temperature sensor, and other features. It also includes a 7-stage pipeline design. The ESP32-CAM is suited for our projects as it is affordable, simple to use, and ideal for Internet of Things (IoT) devices that require an advanced camera functionality such as face recognition and surveillance [16].

3.3. Why SIM800 Module? The SIM800 is the most recent version, with extra features such as Bluetooth and enhanced signal reception, as well as being less expensive. It is a GSM modem that can be used in a variety of IoT projects. You can use this shield to do almost anything a standard cell phone can do: send SMS text messages, make or receive phone calls, connect to the Internet via GPRS, TCP/IP, and more! To top it all off, the shield supports a quad-band GSM/GPRS network, which means it will work almost anywhere in the world. The SIM800L is a compact cellular module that can transmit GPRS, send and receive SMS, and make and receive voice calls. The low cost, small size, and quad band frequency capabilities of this module make it a great alternative for any project requiring long-distance communication. The SIM800 was developed with power-saving techniques, with current usage as low as 1.2 mA in sleep mode [17]. We used SIM800 in our project because it was useful for sending emergency messages to the user in the event of an unstable connection. Although we could have used an ESP32 with a built-in SIM800 (TTGO T-Call board) to simplify our system and save even more energy, it was more expensive than purchasing the ESP32 and SIM800 modules separately, which contradicted our goal of making our system affordable.

3.4. Why PIR Sensors? The passive infrared sensor (PIR) detects both humans and animals up to 10 metres (30 ft) away. The sensor can be used to detect the presence of humans in the room and at the front gate by detecting infrared light generated by a warm body and sending a signal to the ESP32 [12]. Because the PIR sensor is passive, it is unnoticeable and works well in low-light conditions. It detects motion consistently indoors, day or night, and consumes less energy (0.8 W to 1.0 W) than a microwave sensor. They are less expensive than microwave sensors and are suitable for electrical applications in smaller and more compact spaces. When compared to existing intrusion detection systems, it is perfect to install an IoT system using PIR sensors for intrusion detection that consumes less power, is low cost, simply operable, and easy to install [18].

3.5. Why DHT11? The DHT11 is a low-cost digital temperature and humidity sensor. This sensor is simple to connect to any microcontroller. The DHT11 humidity and temperature sensor is a low-cost peripheral capable of sensing relative humidity between 20 and 90% RH with an accuracy of 5% RH throughout an operational temperature range of 0 to 50°C. The temperature is also measured with a 2°C precision in the range of 0 to 50°C. Both values are returned with a resolution of 8 bits. The gadget consumes 0.5 to 2.5 mA. For those worried about battery longevity, its standby current is specified at 100 to 150 A [19]. The DHT11 is a simple and cheap digital temperature and humidity sensor. It measures the surrounding air with a capacitive humidity sensor and a thermistor and outputs a digital signal on the data port (no analogue input connections are required), and it is pretty simple to operate.

3.6. Why MQ2 Gas Sensors? The gas sensor (MQ2) module detects H2, LPG, CH4, CO, alcohol, smoke, or propane leaks. Because of its high sensitivity and short reaction time, measurements may be carried out as effectively as possible. The MQ2 gas sensor, which is commercially available in most developing nations' local markets, makes it easier and more cost effective than preordered sensors from abroad. When compared to using many sensors per pollutant, this sensor detects 7 distinct types of gas components, saving the node 1/7 of the power used [20].

3.7. Why LDR Sensors? LDRs are light-sensitive devices that are frequently employed to detect the presence or absence of light or to measure the intensity of light. According to previous research, by utilizing LDR sensors, which require very little power and voltage for operation, a system may save an average of 71.39% power consumption while having a low cost, a simple structure, and easy employment [21]. The LDR has numerous advantages: it is inexpensive, simple to install, and has a high light-to-dark resistance ratio. They are also compact in size, have very basic hardware, simple LDR connections, a low frequency response, and only need a low voltage to operate.

4. System Design and Methodology

4.1. Hardware Architecture and Implementation

4.1.1. System Block Diagram. Based on SECHA's architecture, we designed a system structure, as shown in the block diagram (Figure 2). The system is meant to be as simple and affordable as possible by employing a single ESP32 microcontroller. However, it is linked to numerous sensors in order to demonstrate its many capabilities and a wide range of applications that deliver various types of features based on the needs of the customers. It may also be customized with new features and functionality, as well as up to 16 relays for controlling any household appliance. If necessary, our system also supports Alexa, Siri, and Google Voice Commands. All of the sensors and components were individually and collectively tested to guarantee that they would continue to function even if one or more components failed. The ESP32 was programmed directly with its inbuilt USB driver before being integrated into a single IDE.

4.1.2. Working Principles and Flowchart of the System

(1) Setting Up Door Security and Unlocking. When the PIR sensor detects movement in front of the door, it calibrates and activates the ESP32-cam. The ESP32-CAM is programmed using camera.py, enabling it to capture images and stream live video to the user's phone. The OV2640 on the EPS32-CAM records at 12.5 to 25 frames per second depending on the resolution and streams it to the Android app. Photos and videos taken by the ESP32 camera can be shared in real time with the user, uploaded to a server, and stored on an SD card, making them accessible even if the internet connection is unstable. When there is no motion detected by the PIR, it remains in deep sleep mode and ceases video streaming after a certain time of inactivity to conserve resources.

To unlock the door, the system uses TTL (GT-521F32) fingerprint recognition technology, which may also be managed via the Android app. When the main door button is pressed, the ESP32 transfers voltage to the central controller, which activates the door solenoid latch and opens the door. It is also programmed to close the door automatically after five seconds and to sound a buzzer if the fingerprint does not match. A sonar sensor and the ESP32-CAM's facial recognition feature may both be used to automate the opening of the door, but we observed that this approach was less secure. A flowchart of door security is given in Figure 3.

(2) Detecting Flammable Gases, Smoke, and Fire. MQ-2 sensors were used to detect smoke and combustible gases, while flame sensors were used to detect fire. When the sensors' readings surpass the threshold, the system activates the buzzer, which sounds to notify the user of rising smoke/gas levels or fire, while the GSM module sends an emergency message to the user or fire department, and an automated water sprinkler will be activated if the flame sensor reading is very high. The temperature was measured using a DHT11 temperature sensor. The DHT11 sensor value is used to switch on and off the fan in the event of a high temperature (more than 30 degrees Celsius in our scenario), and the MQ-2 reading is used to turn on the fan in the event of a large concentration of smoke or gas in the house. A flowchart of flammable gases, smoke, and fire detecting and alarming is shown in Figure 4.

(3) Controlling Lighting and Household Appliances. All of the lights and household appliances were controlled by the user's Android app, but the outdoor light was also configured to turn on and off automatically. When a user presses a button on the Android app, the ESP32 sends a voltage signal to the central controller, which compares it to the threshold voltage and either activates or deactivates the relay module, which controls the on/off of home appliances. To automate the outside light, we used a light-dependent resistance (LDR) sensor to measure the intensity of the outside light and send a signal to the ESP32, which then sends out a signal to the relay module. If the LDR sensor value is greater than the threshold value (0.3 lux in our prototype), the outside light will be turned off. However, if the light intensity value is equal to or less than the threshold value, the outside light will be turned on automatically. Figure 5 shows the flowchart of light control.

4.1.3. Circuit Design. We used the ESP32 as the microcontroller unit. All components and modules are connected according to the circuit diagram (Figure 6). In this system,

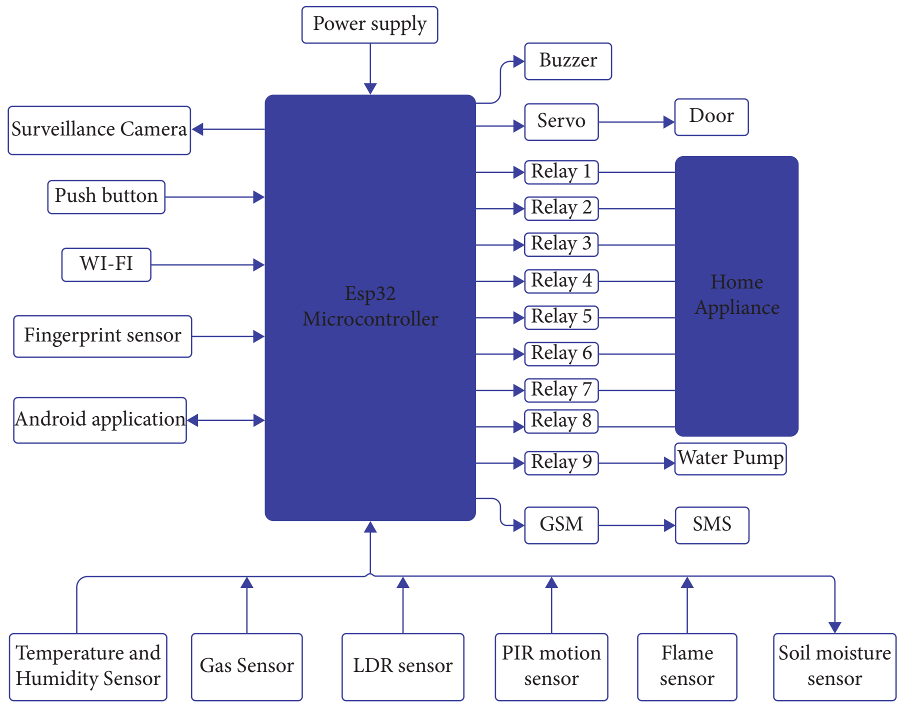


Figure 2: The system block diagram.

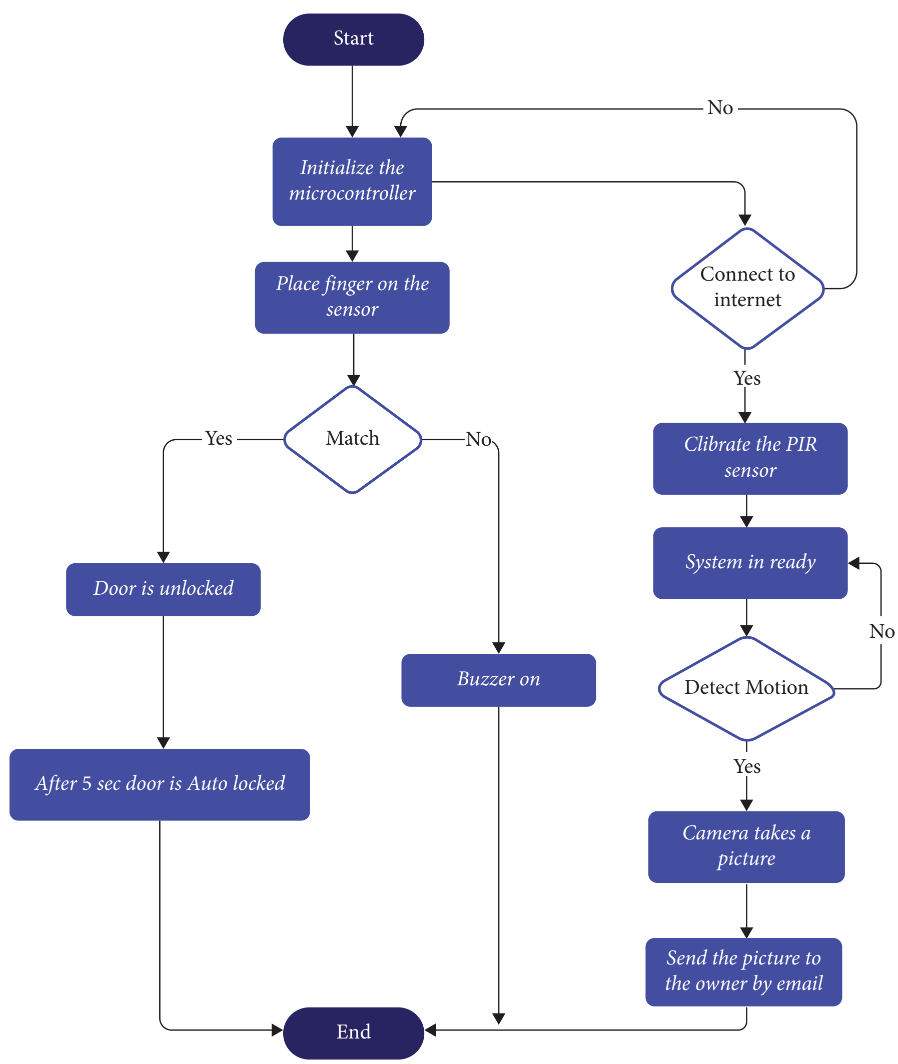


Figure 3: The flowchart of door security.

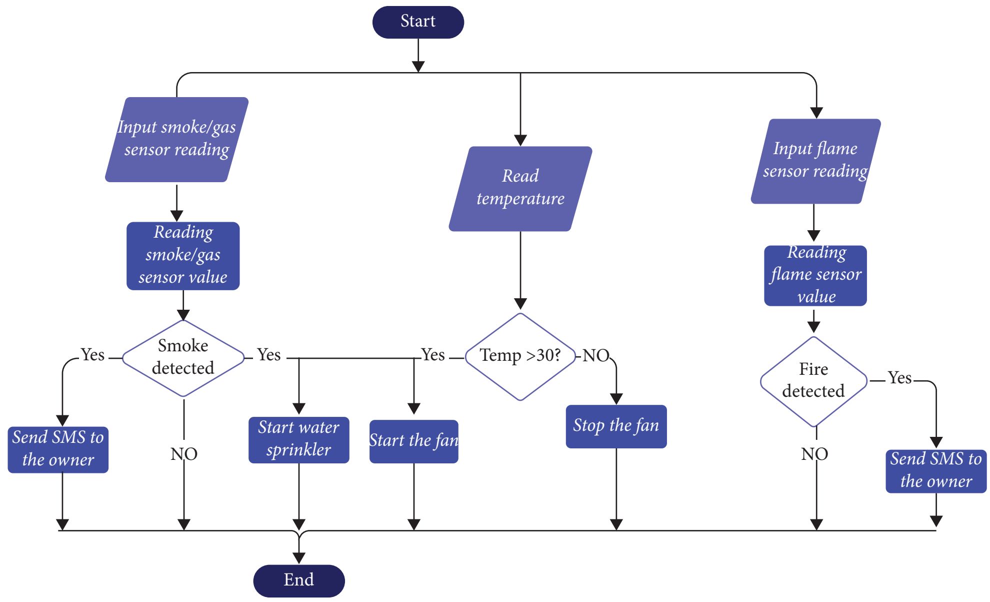


Figure 4: The flowchart of flammable gases, smoke, and fire detecting and alarming.

some components are powered by 3.3 V, so we use two types of power supplies: 5 V or 3.3 V.

4.1.4. Materials Used. To design the SECHA, we have used the materials given in Table 1:

4.2. Software Requirements. Our systems are programmed using a variety of applications and programming languages. The ESP32 is coded directly using the C programming language, while the ESP32-CAM OV2640 is programmed using cam.py, and then integrated into the Arduino IDE. To transfer data between the ESP32 and the user's Android app in real time, we used Google's Firebase database and cloud servers. The Android app is developed using MIT App Inventor, which enabled us to quickly design and implement an Android application that is also easily customizable by the user.

4.2.1. Arduino IDE. It is a free and open-source software development environment that we use to code our ESP32 board in the C programming language. This offers us access to the vast Arduino library, which is continually growing as a result of its large user base. It also simplifies programming our ESP32 board as it can be used to upload code offline, making it ideal for users with unstable or no internet access [22].

4.2.2. Firebase Cloud Server. The Firebase cloud server platform enables us to do real-time data transfer and device connectivity. This platform includes a real-time database, performance monitoring, analytics, networking, and crash reporting capabilities. Because Firebase is supported by Google, it has the greatest level of security. The user's personal data will be adequately safeguarded and free of the risk of being hacked. Firebase Cloud Messaging is a powerful technology for sending notifications to mobile applications through data messages. It can be used to change the behavior of Android applications that are installed on the phone. Firebase is better at saving time on sending data to any mobile application, making it efficient with a low transmission period. Firebase will transmit data whether or not the Internet is available. The benefit of this technology is that when the app connects to the Internet, the cloud will deliver updated data. The Firebase notification is the best in terms of bandwidth use since it pulls the update as soon as the app receives the alert [23].

4.2.3. Android Application Design. We created an Android app to control home appliances and enable security monitoring. This platform is the most prevalent among both developed and developing countries, offering a secure experience. Users can login by using their particular and unique login ID and password.

We used MIT's App Inventor tool [24] to develop the Android app. MIT App Inventor is a simple-to-use visual programming environment that enables anybody to design fully working apps for smartphones and tablets. The Android application's block diagram is provided in reference [25]. An overall view of the Android application is given in Figure 7.

5. Implementation and Results

We first tested the home automation during the implementation process. Sensors were used as inputs, ESP32s as controllers, and lights as outputs in home automation (Figure 8). The optimal and automatic modes are tested independently. The Android app is used to turn ON and OFF the light (Figure 9). We used the app to control the status of the lights in real time. The user activates the security

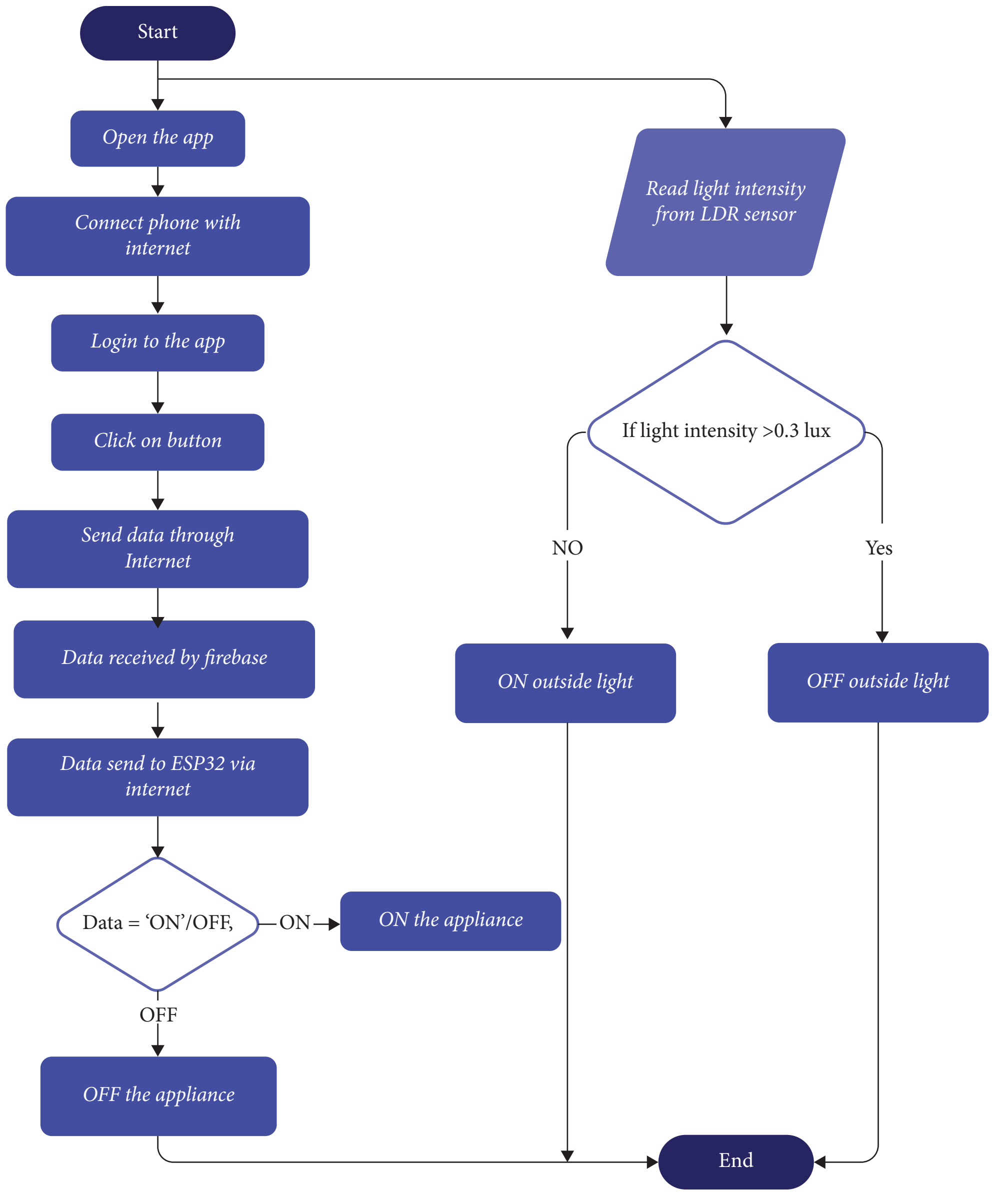


FIGURE 5: The flowchart of light control.

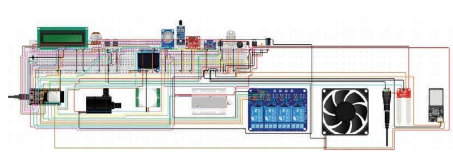


FIGURE 6: The circuit design.

system by placing a finger on the scanner. The door will open if the fingerprints match. Fingerprint scanning takes about 2-3 seconds (Figure 10).

5.1. Door Unlocked by Android Applications. The application is in control of the solenoid lock, which is a hardware unit for the locking mechanism. The user has two commands to open

Table 1: Material used.

	Items	Device type	Quantity
1	ESP32 Wi-Fi module	ESP32	1
2	GSM module	SIM800 MODULE	1
3	ESP32-CAM	—	1
4	DHT11 sensor	—	1
5	Relay module	Electromagnetic relays	1
6	PIR sensor	—	1
7	LDR module	—	1
8	MQ2 gas sensor	—	1
9	Soil moisture sensor	—	1
10	Fingerprint sensor	TTL (GT-521F32)	1
11	Flame sensor	—	5
12	LCD	—	1
13	Fan	—	1
14	Water pump	—	2
15	5 V power supplies	—	1
16	Solenoid lock	—	1
17	Buzzer	—	1
18	PUSH-BUTTON	—	1
19	Resistors	220 ohm	5
20	Jumper wires	—	12

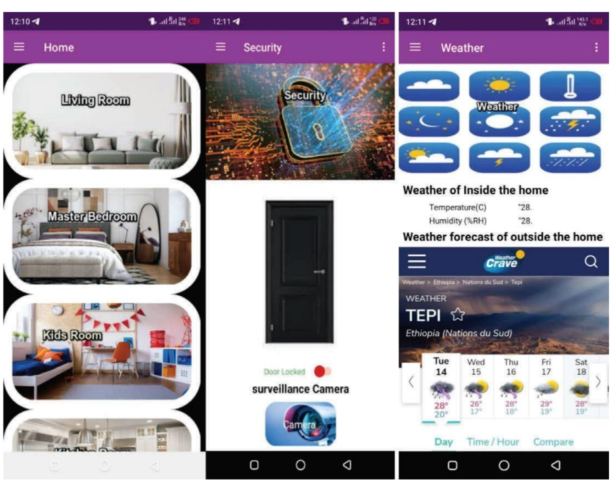


Figure 7: Overall view of the android application.

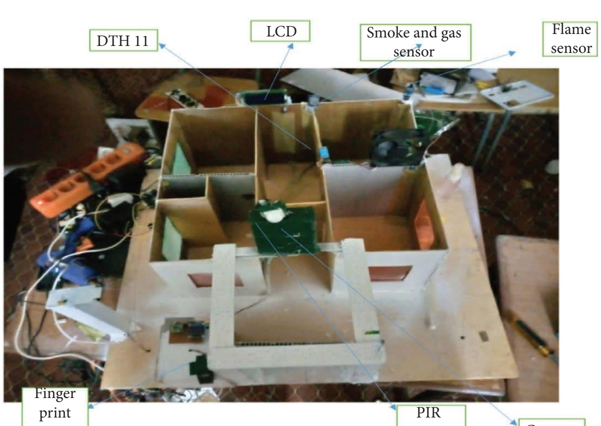


Figure 8: Top view of the home automation prototype.

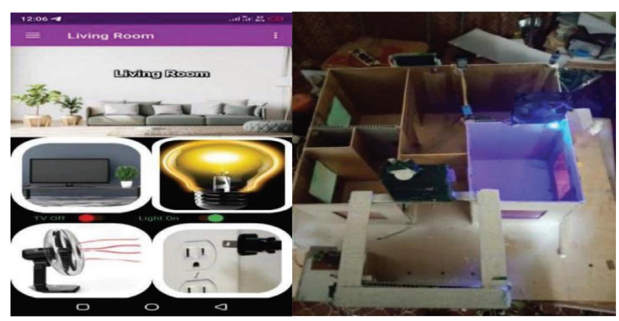


Figure 9: Controlling lights in the room with an Android app.

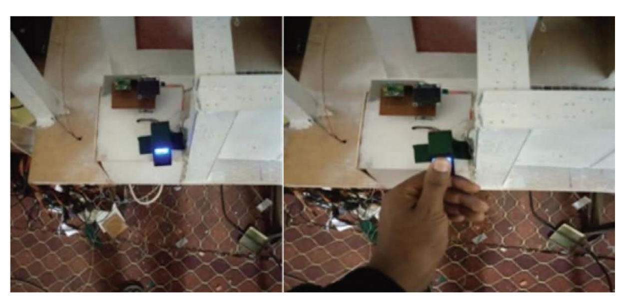
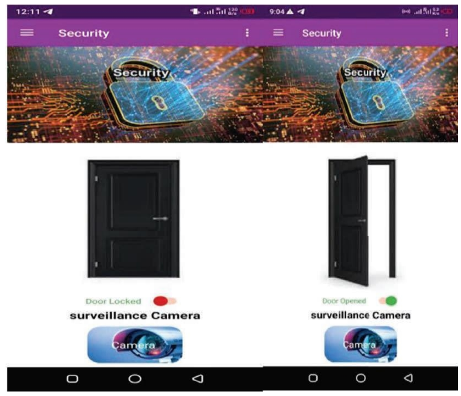


Figure 10: Using a finger print to open a door.

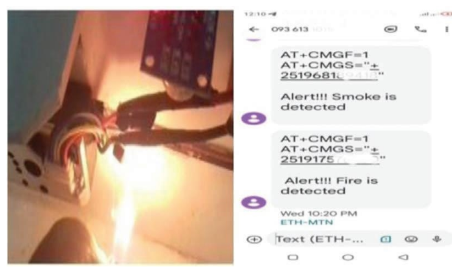
or close the lock in the form of a button in the app (Figure 11), which is controlled by Firebase by a status value (0, 1) indicating whether the door is open or closed. This communication is being carried out with the assistance of a real-time database.

If smoke or fire is detected, an SMS will be sent to the user's phone. We used a lighter near the flame sensor to test the fire alarm system (Figure 12). The LCD screen displays the current temperature and humidity levels, as



(a) (b)

Figure 11: (a) Use the app to close the door; (b) use the app to open the door.



(a) (b)

Figure 12: (a) Fire and smoke sensor testing; (b) SMS notification of fire and smoke.

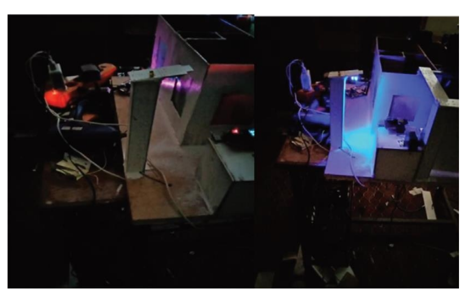


Figure 13: Outdoor light turns ON when it gets dark.

well as a text message sent to the user. SMS alerts are also sent in the case of theft alarms. The Piezo buzzer will start to sound when all of the sensors' detection values are high.

To control the light, we first measure the intensity of the outside light. We used a light-dependent resistor (LDR) to measure light intensity. According to the LDR value, the system will remain off during the day, turn ON automatically when it gets dark, and turn OFF when it gets too dark (Figure 13).

5.2. System Cost and Power Consumption. The cost and power consumption of the system is given in Table 2.

Table 2: Average price and current consumption of used components.

	Average price	Current consumption
ESP32	\$1.80~\$3.20	Active mode: 160~260 mA Modem sleep mode: 3~30 mA Light sleep mode: ~0.8 mA Deep sleep mode: ~6.5 μA Hibernation mode: ~4.5 μA
ESP32-CAM	\$0.78~\$2.22	Maximum brightness and flash ON: 310 mA Flash OFF: 180 mA Modem-sleep mode: up to 20 mA Light-sleep mode: up to 6.7 mA Deep-sleep mode: 6 mA
SIM800 module	\$2.35~\$4.99	Sleep mode: <2.0 mA Idle mode: <7.0 mA GSM transmission (avg): 350 mA GSM transmission (peek): 2000 mA
PIR sensor	\$0.51~\$0.82	Calm output: 40 μA Active output: 400 μA
DHT11	\$0.10~\$0.50	Active mode: 0.3 mA Standby mode: 60 μA
MQ2 gas sensor	\$0.89~\$0.94	~88 mA–160 mA
LDR sensor	\$0.15~\$0.25	Around 0.5 mA
Buzzer	\$0.10~\$0.15	Less than 20 mA
Relay module	\$1.60~\$3.00	Trigger current: 5 mA Active mode: ~70 mA
Solenoid lock	\$1.50~\$4.75	500 mA–900 mA
Soil moisture sensor	\$0.36~\$0.39	Active mode: 4.3 mA Sleep state: <1 mA

6. Conclusion

The Internet of Things is already visible in almost every home, and it is inevitable that it will be the next technological wave in the near future. This necessitates the design of a system that enables IoT home automation in developing and underdeveloped countries. In this article, we propose SECHA, an IoT architecture that can be used as a framework to build a low-cost smart home security system. The main goal of this system is to use cutting-edge technology to make people's lives easier and more comfortable. The system is designed with developing and underdeveloped countries' IoT challenges and opportunities in mind. The system's cost was kept as low as possible when it was designed. Only the most essential features that facilitate home automation while also ensuring resource efficiency are included.

Using the SECHA system, we demonstrated that any developing or underdeveloped country can implement an affordable, energy-efficient, and simple IoT system. We connected all of our sensors to a single ESP32, which sends data to the Android app we developed via a real-time database, making the system as cheap, energy-efficient, and simple to implement as possible. In the event of an emergency, the system could use GSM SIM800 to send SMS notifications even without an Internet connection. Our system can be upgraded or downgraded based on individual cost preferences, and it can also be implemented nationally by institutions or governments or personally by the user.

6.1. Future Scope. This project is a work in progress in the field of IoT applications in home automation systems. In the future, there may be devices that are more reliable, faster, and less expensive. The components we used can be replaced with the most recent device, but it must have the appropriate software and driver. This project's tasks have all been successfully accomplished. We were able to achieve our objectives as outlined in this article. We had time and financial constraints, but we hope that it will serve as a foundation for bringing IoT technologies to developing and underdeveloped countries. There is always room for improvement as technology advances. Machine learning and artificial intelligence can be implemented in various machine operations in the future. New innovations can occur in waste management, efficient water management, and wastewater management. More work and research will be required to accomplish these projects.

Data Availability

The data used to support the findings of this study are included in this research article. For simulation, we have used data from other research papers which are properly cited.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

[1] L. S. Valishery, "Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030," *Stat*, vol. 8, 2022, https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide.

[2] M. N. S. Miazi, Z. Erasmus, M. A. Razzaque, M. Zennaro, and A. Bagula, "Enabling the internet of things in developing countries: opportunities and challenges," in *Proceedings of the 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pp. 564–569, Dhaka, Bangladesh, May 2016.

[3] C. Pham, A. Rahim, and P. Cousin, "Low-cost, Long-range open IoT for smarter rural African villages," in *Proceedings of the 2016 IEEE International Smart Cities Conference (ISC2)*, pp. 1–6, Trento, Italy, June 2016.

[4] T. H. Meles, "Impact of power outages on households in developing countries: evidence from Ethiopia," *Energy Economics*, vol. 91, Article ID 104882, 2020.

[5] K. Kaygusuz, "Energy for sustainable development: a case of developing countries," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 2, pp. 1116–1126, 2012.

[6] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid, and H. Yu, "Green IoT: an investigation on energy saving practices for 2020 and beyond," *IEEE Access*, vol. 5, pp. 15667–15681, 2017, https://ieeexplore.ieee.org/document/7997698.

[7] R. K. Kodali and S. Yerroju, "Energy efficient home automation using IoT," in *Proceedings of the 2018 International Conference on Communication, Computing and Internet of Things (IC3IoT)*, pp. 151–154, Chennai, India, July 2018.

[8] C. Nayyar, B. Valarmathi, and K. Santhi, "Home security and energy efficient home automation system using Arduino," in *Proceedings of the 2017 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1217–1221, IEEE, Chennai, India, April 2017.

[9] P. V. Vimal and K. S. Shivaprakasha, "IOT based greenhouse environment monitoring and controlling system using Arduino platform," in *Proceedings of the 2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, pp. 1514–1519, Kerala, India, April 2017.

[10] H. Msimbe, D. Wilson, J. Salim, F. Rwegoshora, D. R. Sinde, and P. M. kisangiri, "Development of IoT-based system for monitoring electrical energy consumption of the smart and rental houses in Tanzania," *International Journal of Advances in Scientific Research and Engineering*, vol. 08, no. 8, pp. 01–10, 2022, https://ijasre.net/index.php/ijasre/article/view/1557/1968.

[11] A. Ciuffoletti, "Low-cost IoT: a holistic approach," *Journal of Sensor and Actuator Networks*, vol. 7, no. 2, p. 19, 2018.

[12] V. Pravalika and C. R. Prasad, "Internet of things based home monitoring and device control using Esp32," *International Journal of Recent Technology and Engineering*, vol. 8, pp. 58–62, 2019, https://www.ijrte.org/wp-content/uploads/papers/v8i1s4/A10110681S419.pdf.

[13] A. Maier, A. Sharp, and Y. Vagapov, "Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things," in *Proceedings of the 2017 Internet Technologies and Applications (ITA)*, pp. 143–148, Wrexham, UK, November 2017.

[14] M. Babiuch, P. Foltýnek, and P. Smutný, "Using the ESP32 microcontroller for data processing," in *Proceedings of the 2019 20th International Carpathian Control Conference (ICCC)*, pp. 1–6, Krakow-Wieliczka, Poland, May 2019.

[15] ESP32 Technical Reference Manual, "Espressif Systems," 2022, https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf.

[16] W. Budiharto, E. Irwansyah, J. S. Suroso, and A. A. S. Gunawan, "Low-costvision-based face recognition using ESP32-CAM for tracked robot," *ICIC Express Letters Part B: Applications*, vol. 13, pp. 321–327, 2022.

[17] T. Trivedi, A. Narad, S. Verma, N. Purohit, and D. D. Shrivastava, "Audio system using GSM module for announcement," *International Research Journal of Engineering and Technology*, vol. 5, pp. 2816–2818, 2018, https://www.irjet.net/archives/V5/i5/IRJET-V5I5536.pdf.

[18] K. C. Sahoo and U. C. Pati, "IoT based intrusion detection system using PIR sensor," in *Proceedings of the 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information and Communication Technology (RTEICT)*, pp. 1641–1645, Banglore, India, May 2017.

[19] W. Gay, *DHT11 Sensor*, Advanced Raspberry Pi, Apress, Berkeley, CA, USA, 2018.

[20] H. A. H. Nograles, C. P. D. Agbay, I. S. L. Flores, A. L. Manuel, and J. B. C. Salonga, "Low cost Internet based wireless sensor network for air pollution monitoring using Zigbee module," in *Proceedings of the 2014 Fourth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, pp. 310–314, Bangkok, Thailand, May 2014.

[21] G. M. Salim, H. Ismail, N. Debnath, and A. Nadya, "Optimal light power consumption using LDR sensor," in *Proceedings of the 2015 IEEE international symposium on robotics and intelligent sensors (IRIS)*, pp. 144–148, Langkawi, Malaysia, October 2015.

[22] J. Purdum and D. Kidder, *Arduino Projects for Amateur Radio*, McGraw-Hill Education, New York, NY, USA, 1 edition, 2015.

[23] M. A. Mokar, S. O. Fageeri, and S. E. Fattoh, "Using firebase cloud messaging to control mobile applications," in *Proceedings of the 2019 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pp. 1–5, Khartoum, Sudan, April 2019.

[24] MIT, "MIT app inventor," 2022, https://appinventor.mit.edu/.

[25] MIT, "Android Application block diagram from MIT app inventor," 2022, https://ufile.io/jlgo5b26.