International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 10, pp 147-154, October-2024

https://doi.org/10.47001/IRJIET/2024.810020

A Smart, Cloud-Enabled Internet of Things System that Optimizes Home Energy Distribution, Safety, and Consumption

[1*]Ifeagwu E.N., [2]Ejimofor, Ihekeremma A.U.

[1]Department of Electrical and Electronic Engineering, Federal University Otuoke, Bayelsa State, Nigeria
[2]Department of Computer Engineering, Madonna University, Akpugo Campus, Nigeria
*Corresponding Author's E-mail: ifeagwuen@fuotuoke.edu.ng

***Abstract* - This paper focused ona smart, cloud-enabled internet of things system that optimizes home energy distribution, safety, and consumption. The materials used in this paper include Servo motor, lamp holder, energy bulbs, 13A socket, rectifier, jumper wires, electric cables, Blynk interface, motion sensor, temperature/humidity sensor, Esp32, and smoker sensor. The development of the system's code, which was essential for optimizing the system's operations, was carried out using the Arduino Integrated Development Environment. A testbed comprising of the designed work was done in house located in Anambra State, Nigeria over a six-month period, where eight appliances were used to gather used energy by deploying smart plugs, and a smart meter that measures the major energy load of the house. Results show that both implementations collected, stored, and controlled the energy according to the smart, cloud-enabled internet of things system. This work showed that energy providers, and technology developers implement a cloud-enabled Internet of Things system for the best possible household energy usage, safety, and distribution. This technology can significantly increase energy efficiency, save costs, and improve home safety to 100%.**

Keywords: Arduino, Internet of Things, smart home, IoT, Home energy distribution, Home safety, Consumption.

I. INTRODUCTION

The Internet of Things (IoT) and its rapid growth have made it possible to develop novel technologies that can greatly improve a number of elements of daily life. Optimizing home energy distribution, safety, and consumption is one exciting use of IoT technology (Alor, *et al.*, 2022). This background research investigates the potential for combining IoT and cloud computing to create a system that would improve houses' safety, energy efficiency, and connectivity to energy distribution networks. The Internet of Things (IoT) is a network of actual physical objects, gadgets, cars, appliances, and other things that are integrated with sensors, software, and connectivity to allow them to gather and share data on their own (Enebe, *et al*, 2017).

The Internet of Things (IoT) idea imagines a seamlessly connected society in which objects work together to accomplish tasks more quickly. IoT devices gather information about patterns in energy consumption and send it to the cloud for analysis (Idigo, *et al.*, 2011). These data are processed by sophisticated algorithms that yield conclusions and suggestions for cutting energy use. Smart thermostats, for example, can change heating and cooling systems automatically based on user preferences, maximizing energy efficiency without sacrificing comfort (Ifeagwu, *et al.*, 2015a). It is crucial to guarantee safety and security in residential settings. Smart locks, security cameras, and smoke detectors are just a few examples of the linked devices that IoT systems use to improve home safety. These devices, which are managed by a centralized cloud platform, provide for mobile device accessibility to real-time monitoring and warnings.

In contrast, cloud computing makes use of the internet to deliver scalable and adaptable computer resources (Ifeagwu, *et al.*, 2015b). IoT systems can benefit from increased data storage, processing power, and analytics capabilities by utilizing the cloud. Real-time data analysis, remote device monitoring, and control are made possible by integrating IoT with cloud computing, which makes intelligent and automated systems possible. A large amount of the overall energy demand is made up of domestic energy consumption. Effectively managing home energy use, distribution, and safety is still a major concern in residential settings all over the world. Traditional energy management systems in homes generally lack the intelligence and automation needed to optimize energy usage efficiently (Alam, *et al*, 2022). Traditional energy management systems frequently lack the sophisticated features necessary to efficiently track and control energy consumption, which raises operating expenses, has an adverse effect on the environment, and jeopardizes safety precautions. Interoperability issues arise when different devices are integrated for home automation and security,

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 10, pp 147-154, October-2024

https://doi.org/10.47001/IRJIET/2024.810020

impeding the smooth functioning of a networked system. Both the environmental effect and energy bills may rise as a result of this inefficiency (Stoikloska, *et al*, 2017). Through the use of Internet of Things-capable gadgets like voltage regulators, smart meters, and appliances, residential clients can enhance their ability to monitor and manage their energy usage. The emergence of cloud-enabled Internet of Things (IoT) technologies offers a promising solution to these challenges. Using cloud computing for data processing, storage, and analytics in conjunction with IoT-enabled devices that have sensors and actuators, households can potentially achieve real-time monitoring, intelligent energy management, and enhanced security measures. However, the deployment of such integrated systems encounters significant challenges, including data privacy concerns, cybersecurity vulnerabilities, and the need for standardized communication protocols among diverse IoT devices. For example, smart smoke detectors can detect unusual activity and immediately notify homeowners or authorities. (Gubbi, *et al*, 2023).The goal of this article is to develop a state-of-the-art, cloud-enabled Internet of Things (IoT) system that will optimize home energy consumption, improve safety, and guarantee effective energy distribution—all of which will contribute to smart and sustainable living environments.

II. SMOKE DETECTOR

A smoke detector is a device designed to alert building occupants to the presence of fire before it reaches a rapidly spreading stage, facilitating escape or firefighting efforts. Upon detecting smoke, these detectors emit a loud, high-pitched alarm tone (Zanella, *et al*, 2014), typically warbling or intermittent, and are often accompanied by a flashing light.

Types of Smoke Detectors:

1. **Ionization Smoke Detector:** (Zi. *et al*, 2020) Utilizes americium-241 to ionize air and create an electric current that smoke disrupts, triggering an alarm. It quickly detects fast-flaming fires and is cost-effective but is less effective for slow-burning fires and prone to false alarms from cooking or steam, with concerns about radioactive material.
2. **Heat Smoke Detector:** Responds to significant temperature changes rather than smoke, triggering an alarm when temperature rises. It is reliable in dusty or harsh environments and complements other detectors but has a delayed response to heat changes and is not ideal for detecting smoldering fires.
3. **Optical Smoke Detector:** (Zhang, *et al*, 2018) Uses a light source and sensor; smoke particles scatter the light, activating the alarm. It effectively detects smoldering fires, has fewer false alarms, and is reliable for residential spaces but is slightly slower for fast-flaming fires and may be less effective for very fine smoke particles.
4. **Combination Smoke Detector:** Integrates both ionization and optical sensors for broad fire detection. It provides comprehensive coverage and reduces false alarms but involves complex maintenance, higher cost, and greater power consumption.

III. MATERIALS AND METHODOLOGY

Materials

The materials used in this paper include: Servo motor, lamp holder, energy bulbs, 13A socket, rectifier, jumper wires, electric cables, Blynk interface, motion sensor, temperature/humidity sensor, Esp32, and smoker sensor

The precise specs of the ESP32, the Servo motor, the Hi-link rectifier, the current sensor, and the 8-channel relay are displayed in Table 1-Table 5 of this paper. The specifications for the DTH11 (temperature and humidity sensor) are listed in Table 6, whereas the specs for the ultrasonic sensor are listed in Table 7.

Components and their Specifications

Table 1: Eight-Channel relay specifications

Parameter	Rating
Supply Voltage	3.75V to 6V
Trigger Current	5mA
Active Relay current Level	70mA (One relay), 600mA (8 relay)
Relay Maximum Contact Voltage	250VAC, 30VDC
Relay Maximum Current	10A

Table 2: ESP32 specifications

ESP-32	Core	Architecture
DESCRIPTION	2	32 bits
ESP-32	Clock	WiFi
DESCRIPTION	Tensilica Xtensa LX106 160-240MHz	IEEE802.11 b/g/n
ESP-32	Bluetooth	RAM
DESCRIPTION	Yes - classic & BLE	520KB
ESP-32	Flash	GPIO
DESCRIPTION	External QSPI - 16MB	22
ESP-32	DAC	ADC
DESCRIPTION	2	18
ESP-32	Interfaces	
DESCRIPTION	SPI-I2C-UART-I2S-CAN	

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 10, pp 147-154, October-2024

https://doi.org/10.47001/IRJIET/2024.810020

Table 3: Current sensor specifications

Pin Configuration

Pin Number	Pin Name	Description
1	Vcc	Input voltage is +5V for typical applications
2	Output	Outputs Analog voltage proportional to current
3	Ground	Connected to ground of circuit
T1	Wire In	The wire through current has to be measured is connected here
T2	Wire Out	Measures both AC and DC current
		Available as 5A, 20A and 30A module
		Provides isolation from the load
		Easy to integrate with MCU, since it outputs analog voltage
		Scale Factor

Table 4: Servo Motor specifications

Dimension	23x12.2x29mm
Torque	0.5kg/cm
Stall torque	1.5kg/cm at 5V
Operating speed	0.3sec/60degree at 4.8V
Operating voltage	4.2-6V
Temperature range	0-55 °C
Dead band width	10us
Gear medium	Nylon

Table 5: Hi link Rectifier specifications

AC Input Voltage Range	100-240VAC
Output DC Voltage	5VDC
Output Power	5W
Maximum Input Current	<0.2A
Input Current Surge	<10A
Load Rated Output Voltage	+5±0.1
Short-term Maximum Output Current	1000mA
Relative Humidity	5 ~ 95%
Input Low Voltage Efficiency	Vin=110VAc, Output full-load: 69%
Input High Voltage Efficiency	Vin=220VAc, output full-load: 70%
Operating Temperature	-20 ~ +60°C
Store Temperature	-40 ~ +80°C
Dimensions	38 x 23 x 18mm
Net Weight	25g

Table 6: DTH11 (Temperature and Humidity sensor) specifications

Operating Voltage	3.5V to 5.5V
Operating	current 0.3mA (measuring) 60uA (standby)
Output	Serial data
Temperature Range	0°C to 50°C
Humidity Range	20% to 90%
Resolution Temperature and Humidity both are	16-bit
Accuracy	±1°C and ±1%

Smoke detector specifications

Smoke sensitivity; 2.31+/-1.37%/FT Obscuration (UL standard), 0.086dB/m~0.140dB/m Power supply: 9V battery; 100/110/120/220/230/240 VAC, 50/60HZ for AC versions Battery Life: At least 30 months under normal conditions with Alkaline cell Battery Low Battery life: Up to 30 days warning signal Alarm Indicator: Continuously emitting red light and sounder exceeding 85dB at 10 feet.

Table 7: Ultrasonic sensor specifications

Power supply	5V DC
Quiescent current	<15Ma
Effectual angle	<15o
Ranging distance	2cm – 350cm
Resolution	0.3 cm
Output cycle	50ms

IV. METHODOLOGY

This section encompasses of the various methods / step by step ways in which we used to achieve this project.

4.1 System Connection and Analysis

4.1.1 ESP32

The Espressif Systems-developed ESP32 is a popular and reasonably priced microcontroller chip for Internet of Things applications. It has a dual-core Tensilica Xtensa LX6 CPU that can run at up to 240 MHz, 520 KB of SRAM, and support for up to 16 MB of external flash memory. The ESP32 provides extensive communication capabilities, including as Bluetooth 4.2 with BLE and Wi-Fi (802.11 b/g/n). It has two 8-bit DACs, 18 channels of 12-bit ADC, a large number of GPIO ports, and capacitive touch sensors. It runs in the voltage range of 2.2V to 3.6V and is intended for low-power operation with several sleep modes. The chip is supported by a robust developer community and may be programmed using platforms such as Micro Python, ESP-IDF, and Arduino IDE. It is frequently utilized in wearable technology, automation systems, smart home appliances, sensors, and educational

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 10, pp 147-154, October-2024

https://doi.org/10.47001/IRJIET/2024.810020

initiatives. The ESP32 is a go-to solution for developers wishing to construct connected products with a powerful microcontroller that supports both Wi-Fi and Bluetooth.

Steps for Using ESP32:

Step 1: Go to the Arduino website and download the most recent version of the Arduino IDE for your operating system (Windows, macOS, or Linux). We then installed the Arduino IDE.

After that, we launched the Arduino IDE on our computer and installed the ESP32 Board Manager.

To access the Board URL, go here: Go to Preferences under File. In the "Additional Board Manager URLs" section, add the following URL: https://dl.espressif.com/dl/package_esp32_index.json

Next, through the Board Manager's Opening: Go to Boards Manager under Tools > Board.

Looking for the ESP32: Install the "esp32 by Espressif Systems" package after typing "ESP32" into the search bar.

Step 2: Selecting the ESP32 Board:

We Connected the ESP32 by Plugging our ESP32 board into computer using a USB cable.

Then; Select Board: Go to Tools > Board and choose the appropriate ESP32 board (e.g., "ESP32 Dev Module").

Select Port: Go to Tools > Port and select the COM port associated with your ESP32 board

Install Required Libraries:

We Opened our Library Manager and went to; Go to Sketch > Include Library > Manage Libraries.

Search for libraries we looked for things, such as:

- ✓ WiFi: For Wi-Fi connectivity.
- ✓ PubSubClient: For MQTT communication.
- ✓ DHT: For temperature and humidity sensors.
- ✓ Adafruit Sensor: For various sensors.

Step 3: This focused on writing our Sketch (Programming it) through:

- ✓ Open a New Sketch: Go to File > New to open a new sketch.
- ✓ Write Code: Start writing the code in the editor.

Step 4: Compiling and Uploading the Code:

- ✓ Compile the Sketch: Click the checkmark button in the top-left corner of the IDE to compile your code. These checks for errors in your code.
- ✓ Upload the Sketch: Click the right arrow button (next to the checkmark) to upload the code to the ESP32.
- ✓ Monitor Output: Open the Serial Monitor (Tools > Serial Monitor) to view output from your ESP32. Ensure the baud rate matches what you've set in your code (Serial. Begin (115200)).

Step 5: Troubleshooting the Common Issues:

This concentrated on joining the system's parts. Using jumper wires, a circuit was created on a board and connected to the ESP32. The Arduino IDE was used to write the project's code. Next, using the Arduino IDE's "Upload" button, the code was uploaded to ESP32. The serial output was tracked for debugging purposes when needed. Until the project achieved the required functionality, it was tested and refined.

Figure 1: ESP32 Pins Layouts

4.1.2 8-Channel Relays

All the appliances i.e. switches and lamps were connected to the relay using the back wire(Neutral) to be connected to the common ports (COM) of the relay knowing the relay for this project is Normally closed. Hence a loop through each NO (Normally-Open) across the eight relays, the looped wired facilitate the relay to send current to open and close when a Read or Write command is given by the ESP32 Controller.

5VDC input and output is linked out to the controller using jumper wires.

Procedures for Connection:

Power Connections:

- Connect the VCC pin of the relay module to the 5V pin of the ESP32 or an external 5V power supply.
- Connect the GND pin of the relay module to the GND pin of the ESP32.

Control Connections:

- Connect each INx pin on the relay module to a corresponding GPIO pin on the ESP32. For example:

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 10, pp 147-154, October-2024

https://doi.org/10.47001/IRJIET/2024.810020

- IN1 -> GPIO 14 (Relay 1)
- IN2 -> GPIO 27 (Relay 2)
- IN3 -> GPIO 26 (Relay 3)
- IN4 -> GPIO 25 (Relay 4)
- IN5 -> GPIO 33 (Relay 5)
- IN6 -> GPIO 32 (Relay 6)
- IN7 -> GPIO 13 (Relay 7)
- IN8 -> GPIO 12 (Relay 8)

Connect Electrical Loads

Relay Output Connections:

- Connect the COM terminal of each relay to the live (L) wire of the power supply.
- Connect the NO (Normally Open) terminal of each relay to one terminal of the electrical load (e.g., a light or appliance).
- Connect the other terminal of the load to the neutral (N) wire of the power supply.

NC vs. NO: Use the NO terminal if you want the relay to turn on the load when the relay is activated. Use the NC terminal if you want the load to be on by default and turn off when the relay is activated.

Testing and Debugging

Implement and Enhance

Install the System: Set up the relay module and ESP32 where you want them.

Optimize Control Logic: Make adjustments to the control logic in response to feedback and real-time data from the cloud.

Safety and upkeep: Ensure the system is safe on a regular basis and update the firmware when necessary. With the aid of this tutorial, you will be able to link an 8-channel relay with the ESP32 and use a cloud-enabled Internet of Things system to automate and control electrical devices remotely.

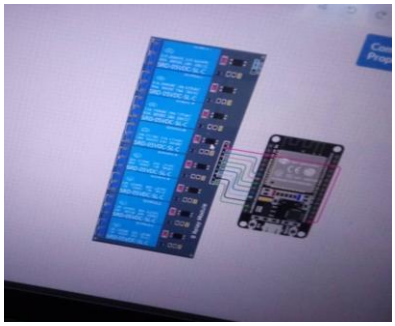

Figure 2: 8-Channel Relay Circuit Diagram

4.1.3 DHT11 (Temperature and Humidity) Sensor

Connection: The Vcc, GND, and Data pin of the temperature and humidity sensor were connected to the 3V3, GND and D4 pins of the ESP32 respectively. It specified Temperature range of 0-50°C, Humidity range of 20-90% RH, provides a digital output.

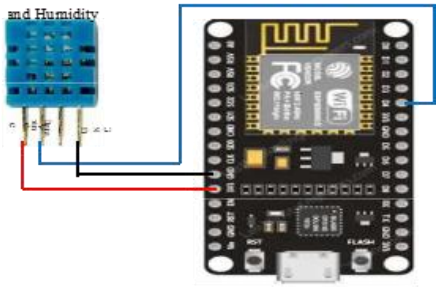

Figure 3: DHT11 (Temperature and Humidity) Sensor Circuit Diagram

4.1.4 Smoke Detector Sensor

Hardware setup, coding, and cloud integration are among the stages involved in connecting a smoke detector to an ESP32 for an Internet of Things system that is cloud-enabled. We performance the operations in these steps:

We choose a smoke detector with a digital output signal that can interface with the ESP32. We decided to use MQ-2 or MQ-135 gas sensors for the smoke detection.

Hardware Connection Links

Power: We attached the smoke detectors VCC pin to the ESP32's 3.3V pin.

Ground: Join the ESP32's GND pin and the smoke detectors GND pin.

Signal: We then attached the smoke detector's digital output (DO) pin to one of the ESP32's GPIO pins (such as GPIO 5).

We proceeded to write codes to facilitate the controls via the Arduino IDE to make the MQ sensor smart; below are the codes initiated (check appemdix)

Integration of Clouds

We then selected a cloud platform (such as Thingspeak, AWS IoT, or Firebase).

Created a new project or object after creating an account.

In the Arduino IDE, we sent data to the cloud platform when smoke is detected.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 10, pp 147-154, October-2024

https://doi.org/10.47001/IRJIET/2024.810020

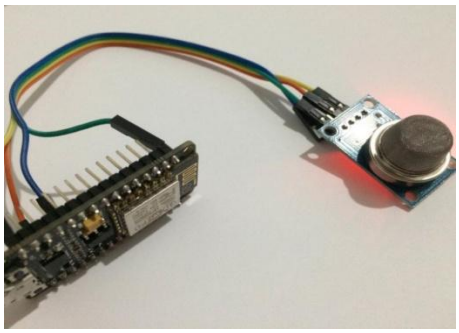

Figure 4: Smoke Detector Sensor Circuit Diagram

4.1.5 Lamp Holders and Accessories

We connected 6 lamp holders to an 8-channel relay for a cloud-enabled IoT system using an ESP32, following these steps:

1. Safety Precautions

- **High Voltage Handling:** Since the lamps will operate on AC mains voltage, be very careful when handling wires. Ensure that the circuit is powered off before making any connections.
- **Insulation:** Use proper insulation for all exposed wires to prevent short circuits or electric shocks.
- **Relay Specifications:** Ensure that the relay can handle the current and voltage of the lamps.

2. Wiring the Lamp Holders to the Relay

- **Common Wire:** Connect the neutral wire (from the mains) directly to one terminal of each lamp holder.
- **Relay Connections:**
  1. Cut the live wire (hot wire) of the AC mains and connect one end to the COM (Common) terminal of the first relay channel.
  2. Connect the corresponding NO (Normally Open) terminal of the relay to the other terminal of the first lamp holder.
  3. Repeat the above steps for each of the 6 lamp holders, using separate relay channels.

3. Relay Module to ESP32 Connections

- **VCC:** Connect the VCC pin of the relay module to the 5V pin on the ESP32 (or to an external power supply if required by the relay).
- **GND:** Connect the GND pin of the relay module to the GND pin on the ESP32.
- **IN Pins:** Connect the relay input pins (IN1 to IN6) to the GPIO pins on the ESP32 (e.g., GPIO 12, 14, 27, 26, 25, and 33).

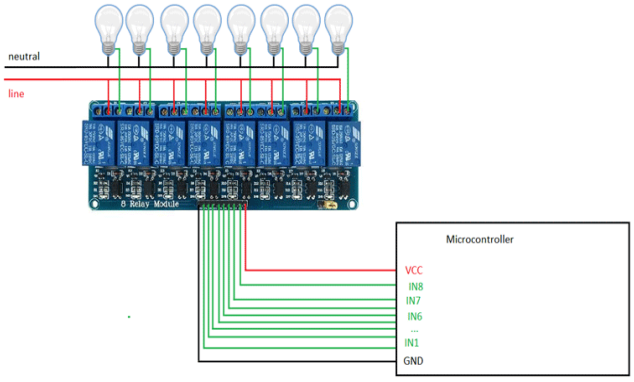

Figure 5: Lamp and accessories circuit

4.2 Working Principle of the Overall System

The setup includes; a motion sensor which was mounted on top a servo motor for precise rotation once a motion is detected, a temperature and Humidity sensor, Current sensor, the 8-channel Relay, Lamp Holders, bulbs and 13amp switches. The ESP32 microcontroller is the central commander of the entire system; it is charged with the sole responsibility to give programmed instructions to different parts of the systems simultaneously, accurately without hold ups, The ESP32 was programmed using the Arduino IDE which codes were sketched out and uploaded to it. Each component of the system enables and gives input and output instructions once the intended command is active or sensed. The code was carefully developed to account for and test every component connected to the ESP32, ensuring that each element performed its function as expected. The process involved writing instructions that facilitated the seamless operation of the system, with a focus on precision and reliability.

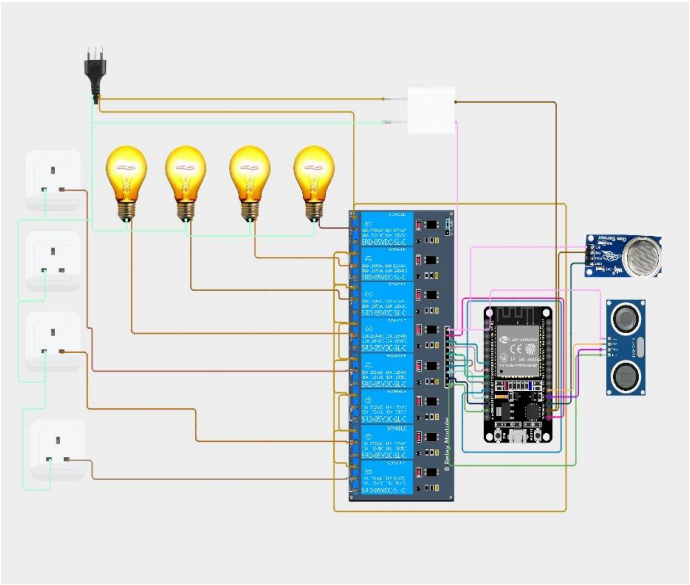

Figure 6: Overall system circuit diagram

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 8, Issue 10, pp 147-154, October-2024

https://doi.org/10.47001/IRJIET/2024.810020

V. RESULTS AND DISCUSSION

5.1 Results

The summary of the results is shown in Table 8 which describes the components functionalities table.

5.2 Components Functionalities Table

Table 8: Components Functionalities Table

Components	Functionality	Performance Range (0 -100%)	Comment
ESP32	Microcontroller	90%	Excellent
8-Channel Relay	Switch	100%	Excellent
Motion Sensor	Detects Electrical Signal	100%	Excellent
Lamp Holder	Secure Mounting And Electrical Connection	97%	Very Good
Energy Bulbs	Illumination	100%	Excellent
Current Sensor	Measures the flow of electric Current	100%	Excellent
Temperature Sensor	Measures the Temperature.	100%	Excellent
Rectifier	Converts alternating current (AC) to direct current (DC),	100%	Excellent
Electrical Accessories	Electrical devices	100%	Excellent
Smoke Sensor	Detects the presence of smoke	90%	Very Good
Blynk Interface	Platform that allows users to create mobile applications	94%	Very Good

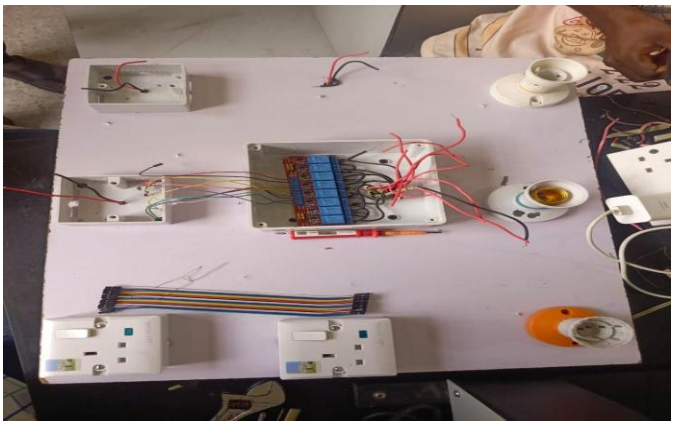

Figure 7(a): Laying of 8-channel relay

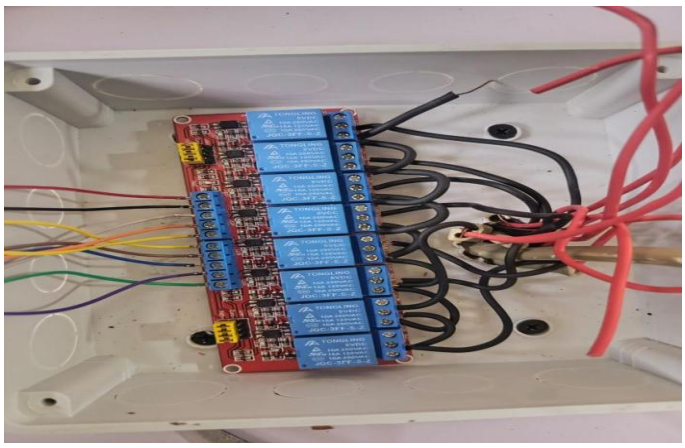

Figure 7(b): 8-channel relay

5.3 Discussion

1. Sensors, smart meters, relays, and other IoT-enabled devices are installed throughout the home to monitor energy usage, control appliances, and detect safety hazards (e.g., smoke detectors, motion sensors).
2. After being gathered by IoT devices, the data is sent to a cloud platform for processing, storing, and analysis. The cloud offers capabilities for machine learning, data analytics, and remote access, all of which contribute to safety and energy efficiency.
3. The system is capable of analyzing real-time data to spot trends in energy usage, recommend energy-saving actions, and automate device operations to cut down on waste. For example, it can change the thermostat according to occupancy or turn off lights or appliances when not in use.
4. Gadgets are able to identify possible safety risks like electrical malfunctions, gas leaks, and fire threats. Through the cloud, alerts can be sent to emergency agencies and homes, guaranteeing quick reactions to potentially harmful circumstances.
5. The system may control how energy is distributed among grid supply, battery storage, and family use in residences equipped with renewable energy sources, such as solar panels. It minimizes reliance on the grid, maximizes the utilization of generated energy, and lowers energy expenses.

ISSN (online): 2581-3048
Volume 8, Issue 10, pp 147-154, October-2024
https://doi.org/10.47001/IRJIET/2024.810020

VI. CONCLUSION

There are several advantages to combining cloud-enabled IoT devices with home energy management, including decreased costs, better efficiency, and safety. By accepting and promoting these technologies, all parties concerned may contribute to ensuring a more secure and sustainable energy future. In order to fully achieve the potential of this revolutionary technology, cooperation between IT companies, energy providers, homeowners, and regulatory bodies is necessary.

It is highly recommended that homeowners, energy providers, and technology developers implement a cloud-enabled Internet of Things system for the best possible household energy usage, safety, and distribution. This technology can significantly increase energy efficiency, save costs, and improve home safety.

REFERENCES

[1] Alam, M. R., Reaz, M. B. I., & Ali, M. A. M. (2022). A review of smart homes—Past, present, and future. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* (Applications and Reviews), 42(6), 1190-1203.

[2] Alor, M.O. Ifeagwu E. N., Abonyi D. O. (2022) "Comparative Study of Least Mean Square And Recursive Least Square Adaptive Beamforming Algorithms On CDMA Based Networks For Improved Performance" *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE),* 17(4), PP 51-56.

[3] Enebe C.C, Ifeagwu N.E, IdigoV.E. (2017) "An investigative study of received signal strength from CDMA 20001X network in a multipath fading channel", *Umudike Journal of Engineering and Technology, UJET*, 2, (1), Pg 14-18.

[4] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2023). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645-1660.

[5] Idigo V.E, Ifeagwu E.N., Azubogu A.C., Akpado K., Oguejiofor O.S., "Simulation and Evaluation of a Simple Adaptive Antenna Array for a WCDMA Mobile Communication", *International Journal of Advanced computer science and Applications, (IJACSA),* ISSN-2156-5570 (online), 25th August, 2011, Page No 1-4.

[6] Ifeagwu E.N. Abba M.O., Obi P.I. "Analysis of Least Mean Square Adaptive Beamforming Algorithm of the Adaptive Antenna for improving the performance of the CDMA20001x base mobile radio network" *International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC),* Vol3, Issue 5, 2015, Pg 3296-3299.

[7] Ifeagwu E.N., Ejiofor H.C., Obi P.I., Okafor C.S., Okoro K.C. "Bit Error Rate Reduction in Code Division Multiple Access (CDMA) 20001x Mobile Communication Network Using Antenna Diversity Technique", *International Journal Of Engineering and Innovative Technology (IJEIT),* Vol 4, Issue 11, 2015, Pg 5-9.

[8] Li, H., Liu, H., & Zhao, X. (2020). Internet of Things: Definitions, challenges, and strategic directions. *International Journal of Communication Systems*, 28(8), 933-948.

[9] Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of Things (IoT): A literature review. *Journal of Computer and Communications*, 3(05), 164.

[10] Stojkoska, B. L. R., & Trivodaliev, K. V. (2017). A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production*, 140, 1454-1464.

[11] Zhang, Y., Qian, C., Wu, D., & Liu, Q. (2018). IoT-based smart home: Design and implementation. *Journal of Physics: Conference Series*, 1069(1), 012020.

[12] Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of Things for smart cities. *IEEE Internet of Things Journal*, 1(1), 22-32.

Citation of this Article:

Ifeagwu E.N., & Ejimofor, Ihekeremma A.U.. (2024). A Smart, Cloud-Enabled Internet of Things System that Optimizes Home Energy Distribution, Safety, and Consumption. *International Research Journal of Innovations in Engineering and Technology - IRJIET*, 8(10), 147-154. Article DOI https://doi.org/10.47001/IRJIET/2024.810020

****

