

Design and Implementation of an Internet of Things System for Air Conditioner Monitoring and Control

Wilfred Brenaldo^{1*}, Agus Triyono², Fara Triadi³

^{1,2,3}Computer Engineering Technology Study Program, Departemen of Information Technology State Polytechnic of Samarinda
wilfredbrenaldo@gmail.com¹, triyono@polnes.ac.id², fara_triadi@polnes.ac.id³

Abstract

An Air Conditioner (AC) is widely used to keep indoor spaces comfortable, but its operation is commonly managed manually through a remote control, which often leads to operational negligence such as forgetting to switch off the unit when a room is empty. This negligence results in energy waste and increased electricity costs, a problem that is particularly relevant for spaces with high-intensity, scheduled usage such as houses of worship. This study aims to design and build an Internet of Things (IoT) system for automatic and remote monitoring and control of an AC unit, using the Kibaid Church congregation in Samarinda as a case study. The system is built around an ESP32 microcontroller integrated with a DHT22 sensor for temperature and humidity monitoring, a PZEM-004T module for real-time power-consumption monitoring, a rain-drop sensor for water-leak detection, and a Solid State Relay (SSR) for switching the AC unit on or off, with a local 16x4 LCD display and the Blynk mobile application for remote monitoring, manual control, and push notifications. The DHT22 temperature sensor was first calibrated against the AC thermostat setpoint, and functional testing was then carried out on nine core features in a controlled pre-testing environment prior to on-site deployment. The calibration results show an average temperature deviation of only 0.17 degrees Celsius, and all nine functional test scenarios - covering sensor readings, leak detection, local display, cloud data transmission, automatic and manual AC control, high-temperature notification, and a 24-hour endurance run - were completed successfully. These findings indicate that the developed system can reliably automate AC operation, provide real-time remote monitoring, and help reduce energy waste caused by operational negligence.

Keywords: *Internet of Things; Air Conditioner; ESP32; Energy Monitoring; Automatic Control; Blynk*

1. Introduction

An Air Conditioner (AC) is an electronic device used to regulate indoor temperature and humidity so that a room remains stable and comfortable for its occupants. AC units are widely used across various types of spaces, including homes, offices, schools, and other public buildings. In general, however, AC operation still relies on a manual remote control that is entirely managed by a human operator, which creates a considerable risk of operational negligence, such as forgetting to switch off the unit when a room is empty. This not only wastes energy but also increases electricity consumption unnecessarily. A system capable of controlling AC operation automatically and remotely, built on more efficient and modern technology, is therefore needed.

As technology has advanced, the concept of the Internet of Things (IoT) has brought significant change to the way people interact with the devices around them. IoT connects physical devices such as sensors, actuators, and control systems to the internet so that they can exchange data, be controlled, and be monitored remotely with minimal direct human intervention. In the context of AC control systems, IoT technology allows users to monitor temperature, humidity, and power consumption, and to control the AC from a smartphone application over an internet connection, thereby improving energy efficiency significantly.

This project uses AC usage at the Kibaid Church congregation in Samarinda as its case study. This house of worship is used for extended periods, particularly during Sunday services and other congregational activities, with a fairly large number of attendees, which causes room temperature to rise, especially given that Samarinda is known for its high average daytime temperature of 32-34 degrees Celsius. In practice, however, operational negligence - such as forgetting to switch off the AC after an activity ends, or using the AC for an inefficient duration - frequently occurs, leading to wasted electrical energy and a high monthly bill. A brief interview with the church management confirmed that the electrical load for the AC is one of the largest recurring expenses, indicating a clear need for a technology-based AC monitoring and control system.

Several related studies have explored IoT-based AC monitoring and control. Pratomo et al. [1] developed a web-server-based AC remoting system using a NodeMCU ESP8266, a DHT22 sensor, and an LDR, achieving an average command-response time of around 10 seconds and demonstrating that AC control is achievable through a browser without requiring an additional application, although response time was somewhat slow and the system did not include power-consumption monitoring or leak detection. Isyanto et al. [2] designed a portable smart AC system based on a NodeMCU and the Blynk application, monitoring temperature, humidity, current, and motion with a reported error of only 1-3%, though the circuit complexity was relatively high. Afandi et al. [3] built an automated AC controller using a DHT11 sensor and a PIR motion sensor, reducing temperature-reading error from 3.11% to 0.29% through a moving-average filter, but relied on the less accurate DHT11 sensor and did not provide local data display. Waluyo et al. [4] proposed an ESP32-based AC control system incorporating a motion sensor and Blynk, achieving 13-22% energy savings compared to manual control. Beta et al. [5] developed an AC monitoring and control prototype for smart classrooms using a NodeMCU ESP8266 and a Telegram bot, reporting a 100% success rate in simulation testing, although the system remained at the simulation stage without a direct connection to a real AC unit.

Building on these works, the present study integrates temperature and humidity monitoring, real-time power-consumption monitoring via a PZEM-004T module, and water-leak detection via a rain-drop sensor within a single ESP32-based system, an integration that is not found together in the related studies above, and specifically targets a house-of-worship setting with a high-intensity, scheduled usage pattern. Based on this background, the research problem addressed in this study is how to design and build an Internet of Things (IoT) system for monitoring and controlling an Air Conditioner (AC) using an ESP32 microcontroller so that it can operate automatically and be accessed remotely. Accordingly, this study aims to design and build an IoT-based AC monitoring and control system using an ESP32 microcontroller that is capable of operating automatically and can be monitored and controlled remotely through the Blynk application.

2. Research Methods

This research was conducted at the Kibaid Church congregation in Samarinda, East Kalimantan, Indonesia, over a period of approximately six months from May to October 2025. This location was selected as the case study because it is directly relevant to the research problem, namely a high intensity of AC usage combined with a considerable risk of energy waste due to operational negligence. The entire process from system design and implementation to functionality testing of sensor readings, power monitoring, and Blynk-based control was intended to be carried out on site, covering requirement analysis, prototype design, field testing, and final effectiveness analysis.

The primary object of this research is the usage pattern and operation of the Air Conditioner (AC) at the Kibaid Church congregation, which plays a crucial role in maintaining the comfort of the congregation during services given the generally high average temperature in Samarinda. A secondary object of this research is the phenomenon of energy inefficiency arising from operational negligence, such as forgetting to switch off the AC unit once a room is no longer in use, which leads to wasted electrical energy and increased cost.

2.1. Research stages

The system was developed using the waterfall development method, in which each development phase must be fully completed before the next phase begins, thereby minimizing the risk of error and preserving clear documentation at every step. The sequential workflow used in this study is illustrated in Fig. 1.

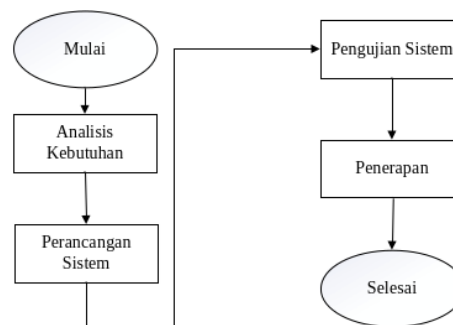


Fig. 1: Research development flow

2.2. Tools and materials

The tools and materials used during the design and testing stages of the system are summarized in Table 1.

No.	Component	Function
1	ESP32 Development Board	Acts as the main control center, processing sensor data, managing control logic, and sending data to the Blynk application via Wi-Fi.
2	Sensor DHT22	Accurately measures room temperature and humidity, used as the basis for automatic AC on/off decisions.
3	Rain Drop Sensor	Detects the presence of water droplets, particularly from AC leakage, and triggers an early-warning notification.
4	PZEM-004T	Monitors the AC unit's real-time electrical power consumption to help control energy usage and prevent waste.

5	Solid State Relay (SSR 25A)	Switches electrical current to the AC on or off based on system or user commands; more durable and faster than a mechanical relay.
6	LCD 16x4 + I2C Module	Displays key information such as temperature, humidity, AC status, and power consumption locally on site.

2.3. System architecture

The overall system architecture, illustrating the interaction between the ESP32 microcontroller, the sensor array, the Solid State Relay actuator, and the Blynk cloud platform, is presented in Fig. 2.

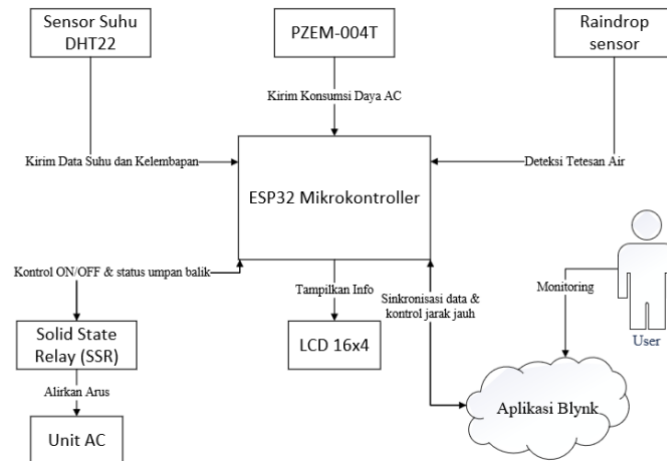


Fig. 2: System block diagram

Based on this architecture, the ESP32 acts as the central processing unit that continuously reads data from the DHT22 temperature-and-humidity sensor, the PZEM-004T power-monitoring module, and the rain-drop leak sensor, and uses this data to drive the Solid State Relay, which switches the AC unit on or off. All readings and the current AC status are simultaneously displayed on the local 16x4 LCD and transmitted over Wi-Fi to the Blynk cloud platform, from which the mobile application renders real-time values, allows manual override of the AC, and pushes notifications when an abnormal condition - such as an overheated room or a detected water leak - is identified.

2.4. Control logic and testing method

The complete decision logic of the automatic control system is presented as a flowchart in Fig. 3.

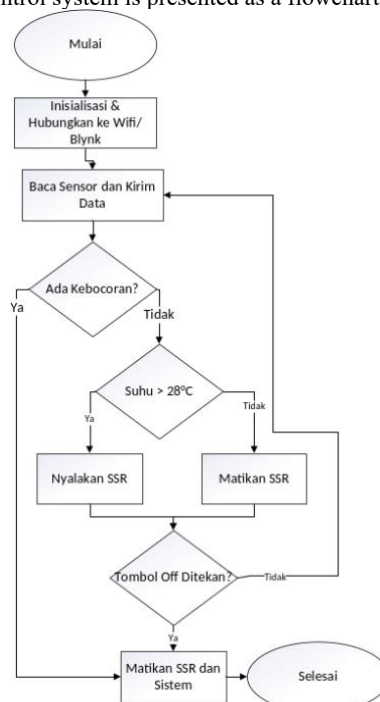


Fig. 3: System control flowchart

Two primary automatic control rules govern the system: the SSR is switched on (activating the AC) whenever the DHT22 reading exceeds a defined temperature threshold, and the SSR is immediately switched off, accompanied by a "Leak" notification, whenever the rain-drop sensor detects water. In addition, the user may issue a manual ON/OFF command at any time through the Blynk application, which is

executed by the ESP32 with the same priority as the automatic logic. A push notification is also generated whenever the room temperature exceeds a defined high-temperature threshold, independent of whether the AC is switched on automatically or manually.

System evaluation was conducted in two stages: first, a temperature-sensor calibration procedure comparing the DHT22 reading against the AC thermostat setpoint at several temperature levels to verify sensor accuracy; and second, functional black-box testing of nine core feature scenarios - covering sensor reading, leak detection, local display, cloud data transmission, automatic control, manual control, high-temperature notification, and a 24-hour endurance test - carried out in a controlled pre-testing environment using a residential AC unit prior to final on-site deployment.

3. Result and Discussion

3.1. System implementation

The hardware components were assembled into an integrated control unit consisting of the ESP32 control circuit, the DHT22, PZEM-004T, and rain-drop sensors, the SSR-driven AC switching mechanism, and the LCD display, with the wiring and sensor integration process documented in Fig. 4.

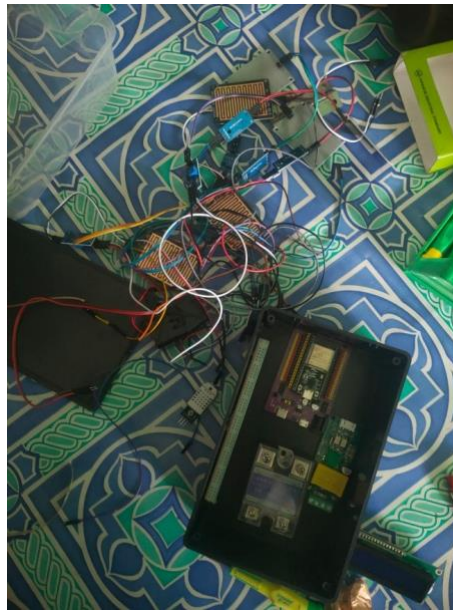


Fig. 4: Wiring and sensor integration process

On the software side, the ESP32 firmware was developed to continuously sample sensor data, evaluate the threshold-based control logic described in Section 2, and synchronize each reading and the AC status with the Blynk cloud platform. The resulting mobile dashboard, allowing users to monitor temperature, humidity, power consumption, and AC status in real time as well as issue manual control commands, is shown in Fig. 5.

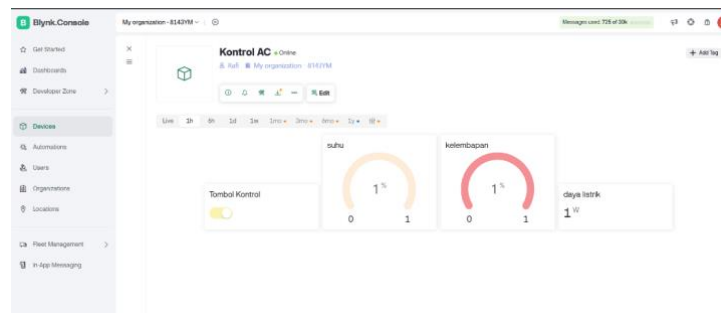


Fig. 5: Blynk application monitoring dashboard

3.2. Sensor calibration results

Before functional testing was carried out, the DHT22 temperature sensor was calibrated against the AC thermostat setpoint at three temperature levels, with the results summarized in Table 2.

Table 2: DHT22 temperature sensor calibration results

No.	AC Setpoint Temperature (°C, Reference)	Sensor Reading (°C)	Deviation (°C)
1	18.0	18.2	+0.2
2	22.0	22.1	+0.1
3	26.0	25.8	-0.2

As shown in Table 2, the DHT22 temperature sensor demonstrated a very good level of accuracy, with a deviation ranging between -0.2 and +0.2 degrees Celsius and an average deviation of only about 0.17 degrees Celsius across the three test points. This deviation is small enough to be tolerated without significantly affecting the performance of the automatic control logic, confirming that the sensor used in this prototype is valid and reliable for the subsequent testing stage.

3.3. Functional testing results

Functional (pre-)testing was carried out in a controlled environment using a residential AC unit to validate that all core features of the prototype - from sensor reading to control logic - operate correctly prior to final installation. Nine test scenarios were designed and executed, with the results summarized in Table 3.

Table 3: Functional test scenarios and results (pre-testing)

No.	Feature Tested	Input	Expected Output	Result
1	DHT22 sensor reading	Room temp. 27°C, humidity 60% RH	~27°C and ~60% RH shown on LCD and Blynk	27.2°C / 61% RH displayed. Passed
2	PZEM-004T sensor reading	AC power consumption ~750 W	~750 W shown on LCD and Blynk	Stable at 748-752 W. Passed
3	Water leak detection	Rain-drop sensor exposed to water	AC switches off; "Leak" notification appears	SSR deactivated; LCD and Blynk notified. Passed
4	Local data display	Sensor data read by system	All data shown consistently on LCD	All data displayed clearly and updated in real time. Passed
5	Data transmission to Blynk	Sensor data updated periodically	Blynk app receives real-time data	Data updated roughly every 2 seconds. Passed
6	Automatic AC control	Temperature rises to 29°C (>28°C threshold)	AC turns on automatically (SSR HIGH)	SSR activated at 28.1°C; status shows "AUTOMATIC". Passed
7	Manual AC control	"ON"/"OFF" button pressed in Blynk	SSR responds to command	Average response time 1.4 s; LCD status synced. Passed
8	High-temperature notification	Temperature exceeds 30°C	Push notification "High Temperature" sent	Notification received at 30.1°C. Passed
9	24-hour endurance test	System run continuously for 24 hours	No crash or hang	System ran stably for 24 hours with no crash or disconnection. Passed

As shown in Table 3, all nine functional test scenarios were completed successfully during pre-testing, indicating that the prototype's core functionality - sensor reading, leak detection, local display, cloud synchronization, automatic and manual control, notification, and long-duration stability - operates as intended in a controlled environment.

3.4. On-site deployment status

Final on-site testing was intended to be carried out after the prototype was installed at the Kibaid Church congregation, in order to evaluate system performance under real operating conditions, including a larger room volume and potential wireless-signal interference compared with the pre-testing environment. However, at the time this research was reported, the church building itself was still under construction, and therefore final on-site testing had not yet been carried out. This section is accordingly limited to describing the planned on-site testing scenario, which mirrors the pre-testing scenario in Table 3 so that the two data sets can later be validly compared once the location is ready.

3.5. Discussion

The calibration and pre-testing results confirm that the core functionality of the prototype - from sensor reading and cloud synchronization to automatic and manual AC control - operates reliably under controlled conditions. The DHT22 sensor's average deviation of 0.17 degrees Celsius indicates that the temperature data underlying the automatic control decisions is trustworthy, while the successful completion of all nine functional test scenarios, including a 24-hour endurance run without a crash or disconnection, indicates that the system is stable enough to be considered for continuous field operation.

With respect to the research problem defined in Section 1, the pre-testing results confirm that an IoT-based AC monitoring and control system using an ESP32 microcontroller can indeed operate automatically - switching the AC on or off based on temperature thresholds and leak conditions without manual intervention - and can be monitored and controlled remotely through the Blynk application, with a manual-control response time averaging 1.4 seconds. These results address the automation and remote-access requirements defined in the research problem, at least under the controlled pre-testing conditions in which they were evaluated.

Compared with the related studies discussed in Section 1, the system developed in this study is distinguished by its combination of temperature-and-humidity monitoring, real-time power-consumption monitoring, and water-leak detection within a single ESP32-based platform - a combination not found together in the related works reviewed, several of which lacked either power monitoring (as in Isyanto et al. [2] and Afandi et al. [3]) or leak detection (as in all five related studies). This broader parameter coverage gives facility managers, such as the church administration in this case study, a more complete picture of both AC performance and potential maintenance issues from a single dashboard.

The most significant limitation of this study is that final on-site validation at the actual case-study location could not be completed, since the building was still under construction at the time of reporting. As a result, the system's performance under real environmental conditions - including a larger room volume, real Wi-Fi infrastructure at the site, and long-term operational stability outside a controlled setting - remains to be confirmed through the planned on-site testing described above. This limitation should be read as a scope boundary of the present study rather than a functional shortcoming of the prototype itself, since every feature that could be tested under controlled conditions performed successfully. Future work is recommended to complete the planned on-site testing once the location becomes available, to extend the system toward multi-unit AC control from a single dashboard, to incorporate a PIR motion sensor so that the AC can be switched

off automatically when a room is detected as empty, and to add a scheduling feature to the Blynk interface so that users can pre-set automatic on/off times.

4. Conclusion

Based on the design, implementation, and testing process described in this study, an IoT-based Air Conditioner monitoring and control system was successfully developed by integrating an ESP32 microcontroller as the main control center, a DHT22 sensor for temperature and humidity reading, a PZEM-004T module for power-consumption monitoring, and a rain-drop sensor for water-leak detection. The system was shown to operate automatically, adjusting AC performance based on temperature readings and leak conditions without requiring manual intervention, and was also shown to be accessible and controllable remotely through the Blynk application, allowing users to monitor AC conditions in real time and perform manual control whenever needed. Calibration testing showed an average temperature-sensor deviation of only 0.17 degrees Celsius, and all nine functional test scenarios evaluated during pre-testing - including a 24-hour endurance run - were completed successfully, indicating that the prototype offers an effective, real-time monitoring and intelligent control solution for reducing energy waste caused by AC operational negligence, particularly for the Kibaid Church congregation case study. Future development is recommended to complete on-site validation once the case-study location is ready, and to extend the system toward multi-unit control, motion-based automation, and schedule-based operation.

Acknowledgement

The authors would like to express their gratitude to the management of the Kibaid Church congregation in Samarinda for granting access and providing valuable input during the design and testing stages of this research, and to the academic supervisors of the Computer Engineering Technology study program, Department of Information Technology, Politeknik Negeri Samarinda, for their guidance throughout this work.

References

- [1] R. Pratomo, G. B. Putra, and M. Jumnahdi, "Sistem Remoting pada Air Conditioner Berbasis Web Server," 2023.
- [2] R. Isyanto, et al., "Rancang Bangun Smart AC Portable Berbasis Internet of Things (IoT)," 2021.
- [3] Afandi, et al., "Automated Air Conditioner Controller and Monitoring Based on Internet of Things," 2021.
- [4] Waluyo, A. Widura, F. Hadiatna, and R. Syamsul Fikri, "IoT-Based Air Conditioning Control System for Energy Saving," *Jurnal Teknik Elektro*, vol. 13, no. 1, 2022.
- [5] Beta, et al., "Prototype Air Conditioning Monitoring and Control System for Smart Classroom Based on the Internet of Things," 2023.
- [6] A. R. Nugraha and A. Hasan, "Kendali Perangkat Elektronik Menggunakan Aplikasi Berbasis Web dengan Arduino," *JUMANTAKA*, vol. 2, no. 1, 2019.
- [7] R. Safarudin, D. Setiawan, and D. Saripurna, "Rancang Bangun Sistem Kendali dan Monitoring Smart Classroom Menggunakan RFID Berbasis Internet of Things (IoT)," *Jurnal Sistem Komputer TGD*, vol. 2, no. 6, pp. 298-306, 2023.
- [8] A. Widodo and A. Sumaedi, "Prototipe Deteksi Hujan Berbasis Arduino Uno Menggunakan Rain Drop Sensor Module," *Jurnal Teknik Informatika STMIK Antar Bangsa*, 2023.
- [9] InnovatorsGuru, "PZEM-004T V3.0 User Manual," 2019.
- [10] Futurlec, "SSR25A Solid State Relay Datasheet," *Futurlec Technical Datasheet*, pp. 57-59, 2025.