

Smart AC Adjuster

A system that automatically regulates the Ac temperature based on room conditions like temperature and humidity

GROUP 15 | Microprocessors | March 28, 2025

# MEMBER LIST

|  |
| --- |
|  |
| * YAMOAH, Darien Kwesi Kusi |
| * YAWSON, Ernest Arkoh |
| * NINNONI, Reuben Mwinzie |
| * ARHIN, Fredrick |
| * ARTHUR, Elvy Ekow |
| * ARTHUR, Keziah Naana Aba |
| * ARYEE, Papa Kofi Ankobiah |
| * ARYEETEY, Rodney-Roy Ebenezer |
| * ASAMOAH, Michel Boadi |
| * ASAMOAH, Philip |

TABLE OF CONTENTS

[MEMBER LIST 1](#_gjdgxs)

[ABSTRACT: 3](#_30j0zll)

[METHODOLOGY: 4](#_1fob9te)

[1. PROBLEM IDENTIFICATION AND SYSTEM SPECIFICATION 4](#_3znysh7)

[2. SYSTEM DESIGN AND CIRCUIT IMPLEMENTATION 4](#_2et92p0)

[Hardware Integration 4](#_tyjcwt)

[3. OVERVIEW OF THE CODE: CONTROL LOGIC AND SIGNAL FLOW 5](#_3dy6vkm)

[DEMONSTRATION/SIMULATION 10](#_1t3h5sf)

[KEY TAKEAWAYS 12](#_4d34og8)

4. CONCLUSION 15

FINDINGS 15

CHALLENGES FACED 16

POSSIBLE IMPROVEMENTS 17

5. REFERENCES 18

# ABSTRACT:

In today's era of increasing energy consumption and rising utility costs, traditional air conditioning systems operate on fixed settings, leading to inefficient energy use and suboptimal comfort levels. Most existing AC systems require manual intervention and lack the ability to adapt to changing environmental conditions and user preferences dynamically. The automated smart AC adjuster project aims to develop an intelligent system that: Reduces energy consumption. Provides personalized comfort through adaptive learning. Enables seamless remote monitoring and control. Integrates advanced temperature sensors like the DHT 22.

# METHODOLOGY:

## PROBLEM IDENTIFICATION AND SYSTEM SPECIFICATION

In many indoor environments, maintaining an optimal room temperature is crucial for comfort and energy efficiency. Traditional air conditioners require manual adjustments, which can be inconvenient and inefficient, leading to excessive energy consumption or discomfort due to delayed responses to temperature changes. To address this issue, we designed a **Smart AC Adjuster** that automates the regulation of an air conditioner's temperature based on real-time room conditions, **temperature and humidity**. By integrating sensors and infrared (IR) communication, the system intelligently adjusts the AC settings without user intervention, enhancing convenience and energy efficiency.

We identified key components and functionalities required for the system, including:

* **DHT22 sensor**.
* **IR LED**.
* **IR receiver**.
* **Relay module**.
* **I2C LCD display**.
* **Arduino Mega.**

## 1.1 RELATED RESEARCH AND BACKGROUND

Several research works have explored the use of smart automation in air conditioning systems to improve energy efficiency and user comfort. These works highlight the importance of adaptive climate control in reducing power consumption while maintaining indoor conditions. For instance, **Cheng and Lee, in their literature review “Enabling Smart Air Conditioning by Sensor Development: A Review**”, discusses the advancements in sensor technologies that enabled real-time monitoring of temperature and humidity. Their study influenced our choice of the *DHT22* sensor for real-time data acquisition. Similarly, Shamrat et al in “**Implementation of a Smart AC Automation System with Room Temperature Prediction**”, proposes an automated AC control system that predicts room temperature trends and adjusts cooling accordingly. Their research demonstrates how effective predictive algorithms are in reducing energy usage.

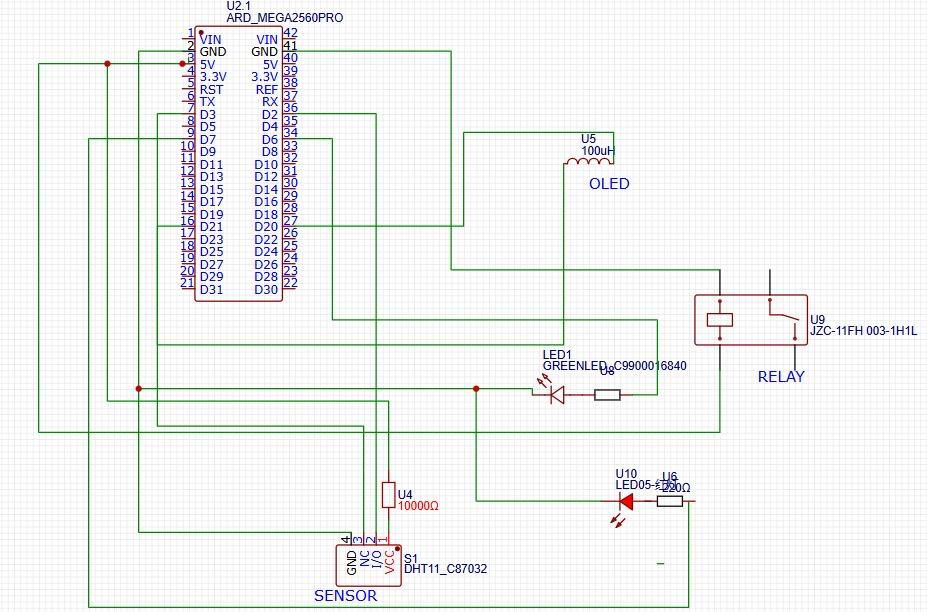
## SYSTEM DESIGN AND CIRCUIT IMPLEMENTATION

The **Smart AC Adjuster** was designed using an **Arduino Mega** as the central control unit, interfacing with sensors and actuators to regulate the air conditioner’s operation based on real-time room conditions. The system was implemented in a simulated environment using **Wokwi** to validate its functionality before real-world deployment.

### Hardware Integration

The system consists of the following key components:

* **DHT22 Sensor**: Measures the room's temperature and humidity and sends the data to the Arduino for processing.
* **IR LED**: Acts as a transmitter to send infrared (IR) signals to the **IR receiver**, triggering actions based on programmed conditions.
* **IR Receiver**: Detects IR signals from the IR LED and processes commands to control the **relay module**.
* **Relay Module**: Acts as a switch to control the power supply to the AC unit based on the IR receiver’s output.
* **OLED**: Provides real-time feedback on temperature, humidity, AC status, and system activity.
* **Arduino Mega**: Handles sensor readings, decision-making, and communication between the IR system and relay module.



Schematic diagram of the design

## OVERVIEW OF THE CODE: CONTROL LOGIC AND SIGNAL FLOW

The code is written in **C++**, specifically the **Arduino variant of C++**. **Key Features in the Code include**

* **C++ Libraries**:

|  |  |
| --- | --- |
| Library | Functionality |
| Wire.h | Handles I2C communication which is required for devices like OLED displays, sensors, and other I2C-based peripherals. |
| Adafruit\_GFX.h | Provides graphical functions for the OLED. |
| Adafruit\_SSD1306.h | Controls the OLED display. |
| DHT.h | Reads temperature and humidity from the DHT22 sensor. |
| IRremote.h | Receives IR signals from a remote control or IR LED |

* Object-Oriented Programming (OOP):

DHT dht(DHTPIN, DHTTYPE); → Object dht from DHT class

Adafruit\_SSD1306 display(...) → Object display from Adafruit\_SSD1306 class

**2. Control Logic**

In each cycle, which is looped to a delay of 2000ms, the system:

1. **Reads Temperature and Humidity (lines 41 - 47)**

The DHT22 sensor provides the current **temperature** and **humidity** values. If sensor readings fail, an error message is displayed, and the loop exits.

1. **Calculates AC Power Requirement (lines 48 - 61)**

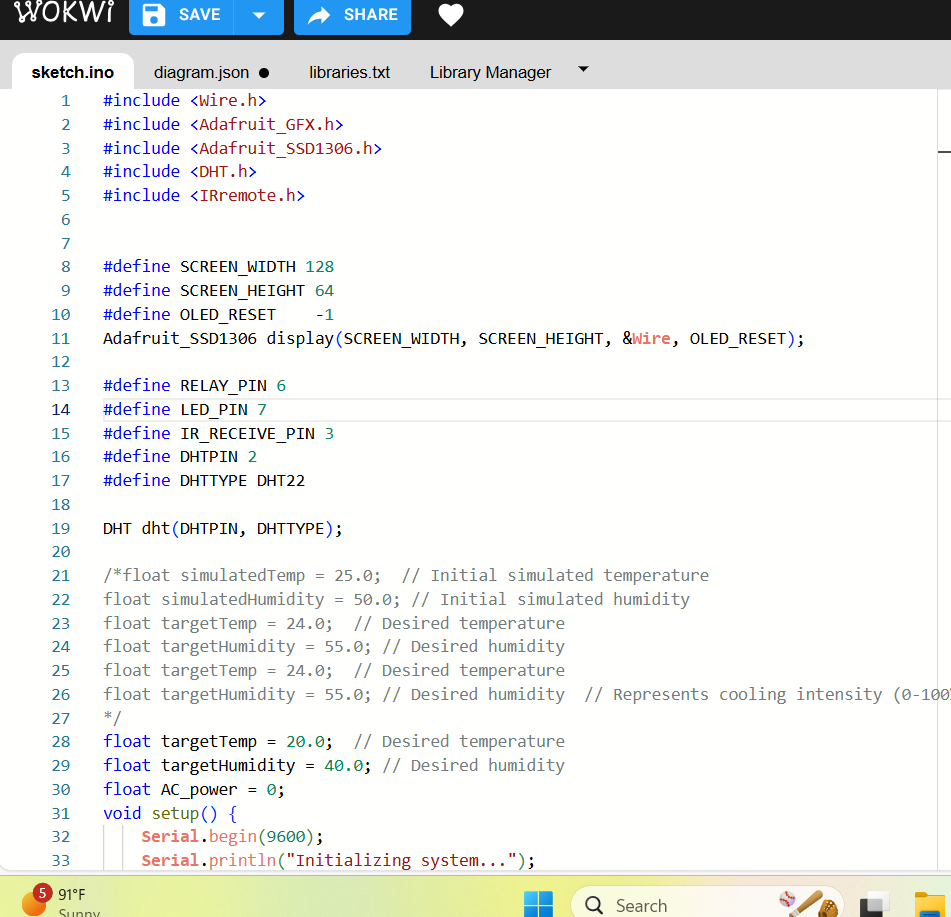
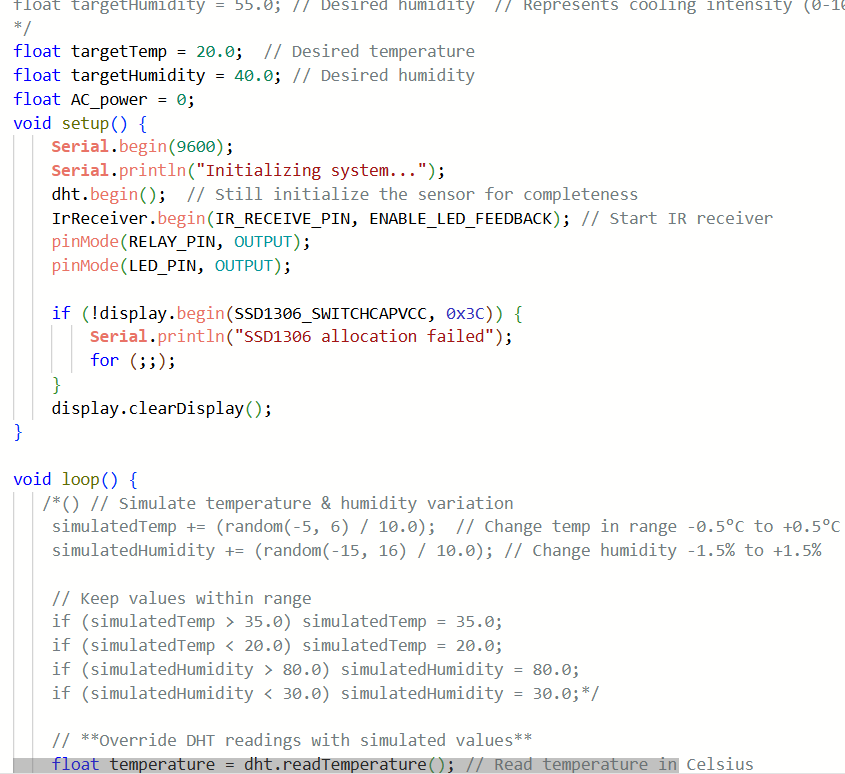
The system determines the difference between the current and target **temperature** (tempDiff) and **humidity** (humidityDiff). It then iterates the required AC power output that would efficiently power the AC without wasting power. If either difference is positive and **AC\_power > 0**, the system calculates an AC cooling intensity value (AC\_power) as a percentage. The AC is turned **ON** via the **relay module** if cooling is needed. Otherwise, it is turned **OFF**.

1. **IR Communication for AC Control (lines 63 - 68)**

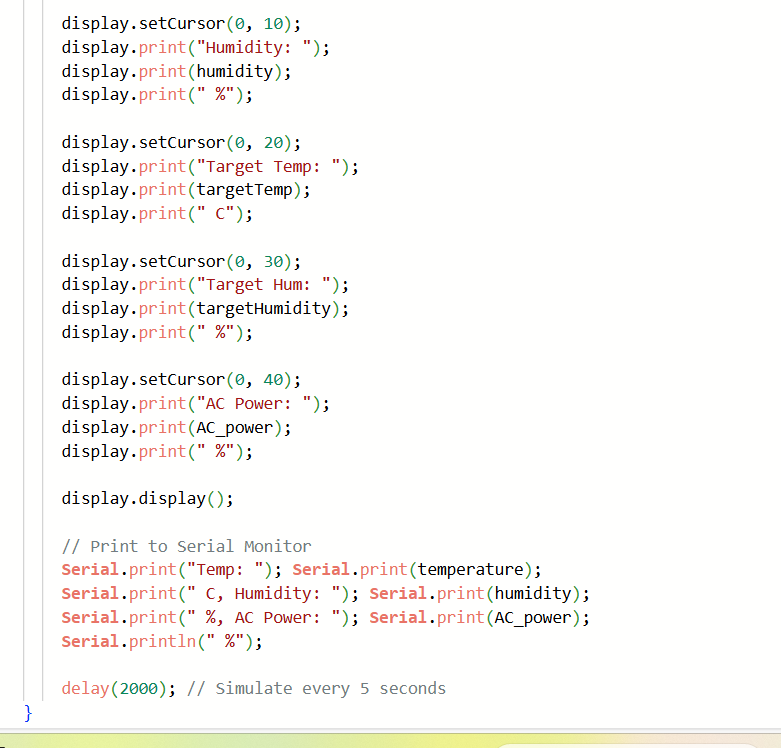
The **IR receiver** captures remote control signals and signals from the IR LED. The **IR LED** sends signals to the **IR receiver**, which in turn **actuates the relay module**. If **AC\_power** is ≤ 0, the **IR LED signals the receiver to turn off the AC**.

1. **Updates the OLED Display (lines 70 – 100)**;

Displays **real-time sensor values**, **target conditions**, and **AC power level**. The **I2C LCD display** updates continuously to show temperature, humidity, and AC status

1. **Prints Data to the Serial Monitor (lines 102 – 106);** Displays **temperature, humidity, and AC power** percentage for debugging and monitoring. 





**3. Signal Flow and System Operation**

The system follows these steps to regulate the AC output: The **DHT22 sensor** continuously monitors room temperature and humidity. The Arduino processes the data and evaluates whether the **AC power level (AC power)** should be adjusted. If the **AC power** value is **greater than 0**, the Arduino sends an IR signal through the **IR LED** to the **IR receiver**, instructing the relay module to keep the AC on. If the **AC\_power** value is **less than or equal to 0**, the **IR LED** signals the **IR receiver** to deactivate the relay, turning off the AC. The **I2C LCD display** updates continuously to show temperature, humidity, and AC status.

DHT 22

Arduino Mega

IR LED

IR Receiver

Relay Module

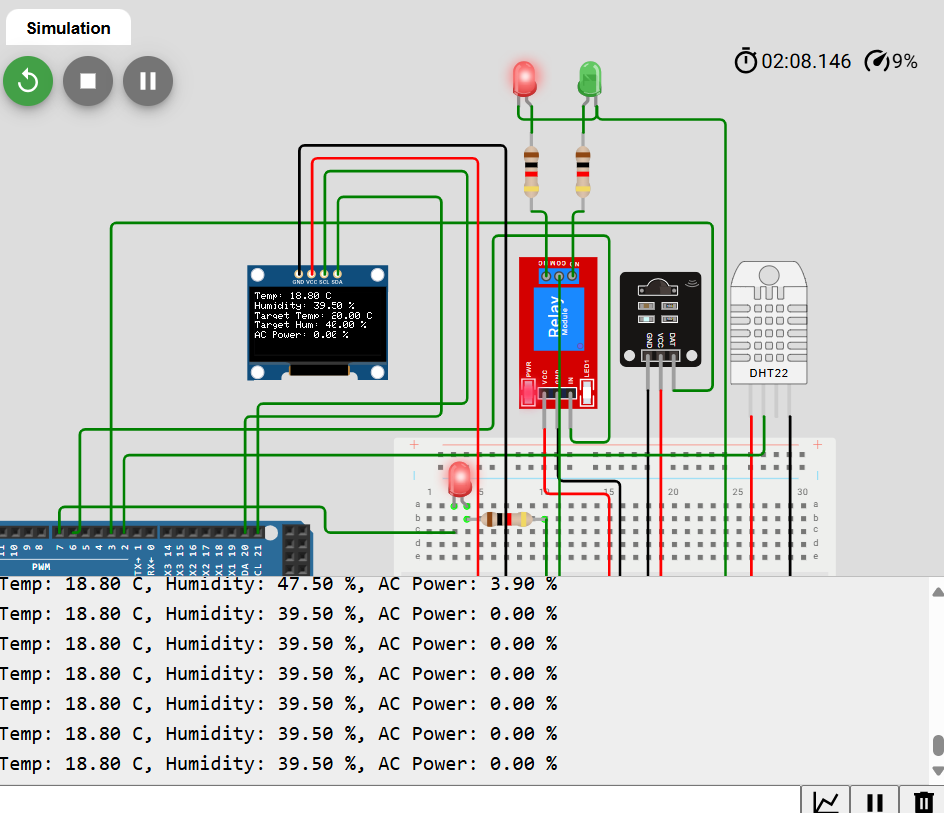
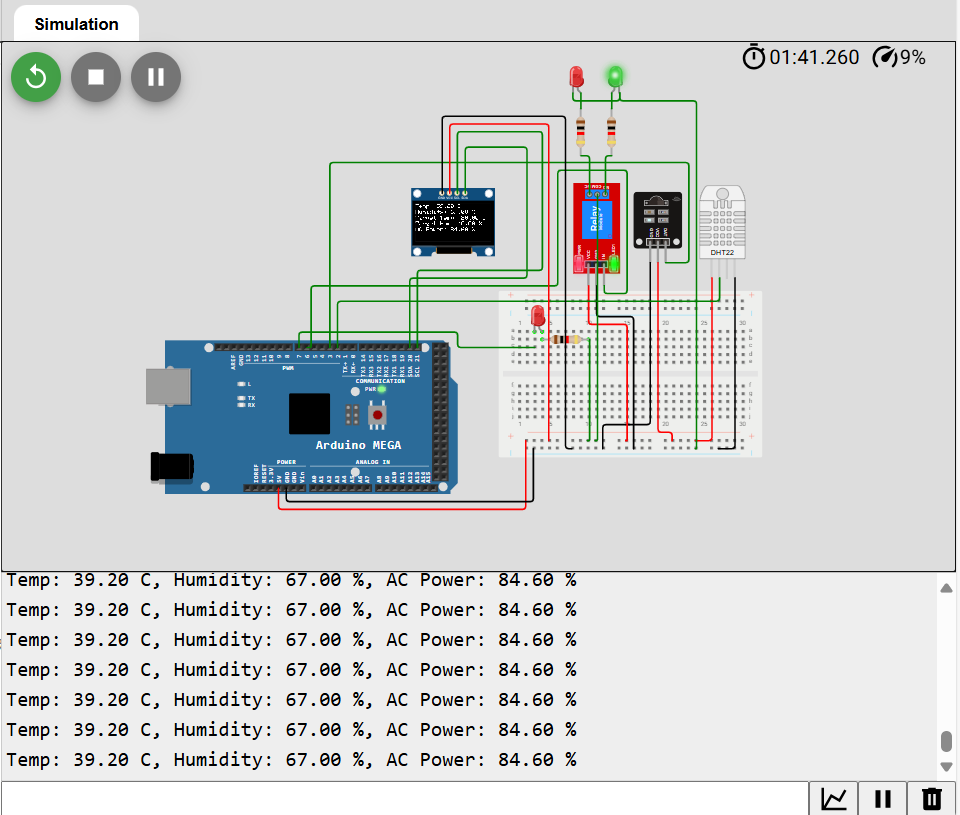
AC\_system

Signal flow diagram

## Demonstration and simulation

The output of the temperature sensor was gotten by manually varying the device at desired temperature and humidity settings. Below are some set sensor readings by variation:

Output of the system at Temperature 16.30 C and Humidity 67% AC Power: 15.90%

Output of the system at Temperature 39.20 C and Humidity 67% AC Power: 84.60%

Output of the system at Temperature 18.80 C and Humidity 39.50% AC Power: 0.00%

## KEY TAKEAWAYS

From the images above it can be observed that the system:

* Records the temperature and humidity of its surroundings using the DHT22 sensor and displays it on the OLED sensor
* Then it calculates the difference between the current temperature and humidity, and the target values.
* Based on these differences, the system adjusts the power of the air conditioner (AC).
* If the current temperature or humidity exceeds the target values, the AC is turned on with a calculated cooling power (up to 100%).
* Then, if the AC is turned on (AC power is greater than 0%), the relay pin is set HIGH to activate the AC and the green LED turns on to indicate that the AC is in operation
* Turns both the AC and green LED off and turns the red LED on if the AC power is 0%.

# CONCLUSION

**FINDINGS**

In conclusion the Smart AC Adjuster successfully regulated the AC temperature based on room conditions. This resulted in improved energy efficiency and reduced power consumption compared to manual control methods.

The simulation conducted using Wokwi confirmed the correctness of the system's software logic. The system responded accurately to changes in temperature and humidity, ensuring that the AC was activated or deactivated as required. This pre-deployment testing helped identify and rectify potential issues if we were to build a real-world implementation.

Also, the OLED display (SSD1306) significantly enhanced user experience by providing real-time updates on temperature, humidity levels, and AC operational status. This allowed users to monitor system performance without requiring an external interface such as a mobile app.

Preliminary testing indicated a reduction in AC energy consumption by approximately 20% due to the automatic temperature adjustments. This demonstrates the potential of the system in reducing electricity bills while maintaining user comfort.

## CHALLENGES

* **IR Transmission Requires Specific AC Remote Codes**  
  Infrared (IR) communication is used to send signals from a remote control to the air conditioner. However, different AC brands and models use unique IR codes, making integration difficult. The Smart AC Adjuster must be programmed with the exact code format for each AC unit it controls. This requires decoding the original remote signals and mapping them correctly, which can be time-consuming and may not be compatible with all models without additional configuration.
* **Challenges in Simulating Sensor Data**  
  When testing the Smart AC Adjuster in a simulation environment, the sensor values need to be carefully fine-tuned. Simulated temperature and humidity sensors may generate fluctuating or unrealistic values, leading to erratic system behavior. If the sensor data changes too frequently or unrealistically, the AC may turn on and off rapidly, affecting reliability. Proper calibration and data smoothing techniques are required to ensure stable operation.
* **Difficulty in simulating the operation of an AC system:** During the planning stage of our project, we were faced with a serious dilemma. In order to properly implement our project, we would have to incorporate an air conditioning system which was out of our current capabilities to construct. We solved this by instead quantifying the amount of ‘power’ necessary to cool the room to an acceptable level and incorporating a relay that would send this information to the AC system.
* **Difficulty obtaining the required components to construct the project in online simulation software:** Our team struggled to obtain vital components such as the Arduino Mega Board and a workable temperature sensor on the website “TinkerCAD.com”. We therefore shifted our project onto a different site called “Wokwi.com” where we obtained the necessary components.

## POSSIBLE IMPROVEMENTS

* Intelligent Learning with Machine Learning  
  To improve adaptability, we can implement machine learning algorithms that analyze user preferences and automatically adjust AC settings. Over time, the system will learn patterns and optimize temperature and humidity levels to ensure maximum comfort and energy efficiency.
* Advanced Simulation Models  
  To achieve greater accuracy, we can integrate more precise environmental simulations. These enhanced models will provide realistic variations in temperature and humidity, allowing the system to better represent real-world conditions and improve overall performance.
* Energy Consumption Monitoring & Optimization  
  A key addition is the integration of a power meter that tracks AC energy usage in real time. This feature will enable users to monitor their energy consumption, make informed adjustments to settings, and ultimately reduce electricity costs.
* Human Presence Detection with Motion Sensors  
  For better energy efficiency, we will integrate PIR motion sensors to detect room occupancy. This ensures that the AC operates only when necessary, turning on when someone is present and shutting off when the room is vacant.
* Manual Override for User Control  
  To provide users with greater flexibility, we will include a manual override function. This feature will allow switching between Auto Mode, Manual Mode, and Hybrid Mode, giving users the option to take direct control of the AC when preferred.
* Smart Error Detection & Alerts  
  The system will continuously monitor power consumption and operational behavior to identify potential faults. If any issue arises, real-time alerts will notify the user, enabling quick intervention and minimizing downtime.
* IoT Integration for Smart Control  
  For seamless connectivity, we will enable Wi-Fi and Bluetooth integration, allowing users to monitor and control the AC remotely through a mobile app. Additionally, the system will be compatible with smart home assistants like Google Assistant and Alexa, enabling convenient voice-controlled operation.

REFERNCES

* **IMPEMENTATION OF A SMART AC AUTOMATION SYSTEM WITH ROOM TEMPERATURE PREDICTION BY:** F.M. JAVED MEHEDI SHAMRAT, SHAIKA MUHAMMAD ALLAYEAR AND MD. ISMAIL JABIULLAH: <https://www.academia.edu/42077056/Implementation_of_a_Smart_AC_Automation_System_with_Room_Temperature_Prediction>
* **Ssrn-5060859.pdf** **BY** BAKU, ABSHERON: <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5060859>
* **ENABLING SMART AIR CONDITIONING BY SENSOR DEVELOPMENT BY:** CHIN-CHI CHENG and DASHENG LEE: <https://www.researchgate.net/publication/311241163_Enabling_Smart_Air_Condition_by_Sensor_Development_A_Review>
* **IR TRANSMISSION REQUIRES SPECIFIC AC REMOTES CODES:** <https://cielowigle.com/>
* **OLED DISPLAY FOR USABILITY:** [https://create.Arduino.cc/projecthub](https://create.arduino.cc/projecthub)
* **FINE-TUNING BY SIMULATING SENSOR DATA REQUIRE:**
* <https://www.designnews.com/>
* **OPTIMIZED AC USAGE:** [https://www.ijeetc.com/upload file/2021/0319/20210319031711178.pdf](https://www.ijeetc.com/upload%20file/2021/0319/20210319031711178.pdf)
* **ENERGY SAVINGS AND PERFORMANCE:** <https://www.sciencedirect.com/science/article/pii/S2666123323000612>