CYBER-PHYSICAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

SMART ROOM

GROUP 10

Adam Bintang Arafah	**2206029273**
Aisyah Arifatul Alya	**2206059383**
Andrew Kristofer Jian	**2206059673**
Ivan Yuantama Pradipta	**2206824243**

PREFACE

We offer our praise and gratitude to Almighty God for His blessings, which have enabled us to complete the final laboratory report for the Cyber-Physical Systems class. This report documents our final laboratory project, titled “Smart Room.” We chose this title to reflect our goal of enhancing comfort and quality of life within a room.

This report serves as the final assignment for the Cyber-Physical Systems course. Completing this report on time would not have been possible without the support and assistance of various individuals. We are deeply grateful for their contributions in numerous forms.

The authors recognize that there may be several mistakes in this report, including errors in spelling, vocabulary, grammar, ethics, and content. Therefore, we sincerely welcome any criticisms and suggestions from readers, which we will use for future improvements. We hope this final laboratory report will be accepted as a valuable contribution to the field of computer engineering, enriching intellectual wealth. May it benefit both the readers and ourselves.

Depok, May 28, 2024

Group 10

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

The idea of this project is based on the increasing need for automated home features, particularly for individuals such as the elderly, infants, and those with limited mobility. These groups often require assistance with their daily activities, which can lead to a significant reliance on others for help. By integrating "Smart Room" features, we aim to enhance their comfort and independence, regardless of the external weather conditions or indoor air quality. Our "Smart Room" system offers automated control of the air conditioner, which adjusts based on the room's current temperature, ensuring a consistently comfortable environment. Additionally, the system includes automation of humidifiers, which activate in response to poor air quality, as detected by the DHT11 sensor. This ensures that the air remains clean and breathable, further enhancing the overall living conditions..

To bring this concept to life, the project is implemented using two Arduino Uno microcontrollers configured in a master-slave arrangement. The master Arduino Uno serves as the central hub, processing data and making decisions based on inputs from the sensor. The slave Arduino Uno assists by managing specific tasks such as controlling the relay switches for the air conditioner and humidifier. Various components are integrated into the system, including an LED for each AC and humidifier to indicate system status, a button for manual control or override functions, and relays to manage the power supply to the air conditioner and humidifier. The DHT11 sensor plays a crucial role by continuously monitoring the room's temperature and humidity levels, providing real-time data to the master Arduino Uno, which then communicates with the slave Arduino Uno to execute the necessary actions.

The primary goal of this project is to create a user-friendly automation system that minimizes the need for manual intervention, thereby supporting those who are less able to manage these tasks themselves. By leveraging the capabilities of modern microcontrollers and sensors, we aim to deliver a solution that not only improves the quality of life for the target users but also showcases the potential of smart home technology in addressing real-world challenges. This project exemplifies how thoughtful design and innovative technology can converge to create practical, impactful solutions for everyday living.

1.2 PROPOSED SOLUTION

The "Smart Room" system consists of several key components. Two Arduino Uno microcontrollers are used in a master-slave configuration and communicate using the I2C protocol. The master Arduino Uno serves as the central processing unit, gathering data from sensors, making decisions based on predefined thresholds, and coordinating actions. The slave Arduino Uno is responsible for executing specific control tasks such as operating the relays connected to the air conditioner and humidifier. The DHT11 sensor continuously monitors the room's temperature and humidity levels, providing real-time data to the master Arduino, which uses this information to maintain optimal environmental conditions.

The system includes relays that control the power supply to the air conditioner and humidifier based on commands from the slave Arduino. These relays act as switches that can turn these devices on or off as needed. The LED indicator provides visual feedback on the system's status, indicating whether the automation is active or not. Additionally, a button allows users to manually override the system, providing flexibility and control when needed.

Automated climate control is achieved by the master Arduino continuously monitoring the room temperature using the DHT11 sensor. When the temperature exceeds or falls below predefined thresholds, the master Arduino signals the slave Arduino to activate or deactivate the air conditioner via the relay, maintaining a comfortable temperature. Similarly, the DHT11 sensor also tracks humidity levels. If the humidity deviates from comfortable ranges, indicating poor air quality, the master Arduino instructs the slave Arduino to turn on the humidifier using the relay. This ensures that the air remains clean and breathable.

The implementation of this project involves several steps. The master Arduino collects data from the DHT11 sensor at regular intervals and processes this data to determine if any action is required to adjust the room's temperature or humidity. The master and slave Arduinos communicate via serial or I2C communication protocols, allowing for efficient data transfer and command execution without overloading a single microcontroller. The system includes a simple user interface with LEDs as the indicator that would tell the user whether the AC and humidifier is operating or not and a manual control button. The LEDs provide the status of the AC and humidifier, while the button allows users to intervene if necessary.

1.3 ACCEPTANCE CRITERIA

The acceptance criteria of this project are as follows:

1. The functionality of the DHT11 which allows the air conditioner to be automatically switched on when the temperature reaches 30°C and the humidifier when the humidity level falls below 40%.
2. LEDs successfully indicate when the AC and humidifier is operating.
3. If the humidity level rises above the threshold, the humidifier will be turned off and will be turned on when the humidity level is below 40%.
4. Air conditioner will be switched on when the temperature of the room exceeds 30°C.

1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Brainstorming the idea for the proposed project	Brainstorming the possible ideas that can be implemented for the Cyber-Physical System final project	• Adam Bintang Arafah • Aisyah Arifatul Alya • Ivan Yuantama Pradipta
Proteus schematic and Assembly	Builds the project using the Proteus software	• Ivan Yuantama Pradipta
Documentation and Final Project Report	Contributes in the documentation of the project and the making of the final report	• Adam Bintang Arafah • Aisyah Arifatul Alya • Andrew Kristofer Jian • Ivan Yuantama Pradipta
Assembling the real components	Take part in the making of the project	• Adam Bintang Arafah • Aisyah Arifatul Alya • Andrew Kristofer Jian • Ivan Yuantama Pradipta

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

Tasks	May 2024																	
	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Hardware Design Completion																		
Software Development																		
Integration and Testing of Hardware and Software																		
Final Product Assembly and Testing																		

a) Hardware Design completion: A milestone indicating the date when the hardware design for the embedded system is finalized, including schematic.

b) Software Development: The date when the development of the user-created assembly code (software) begins, focusing on specific tasks and functionalities.

c) Integration and Testing of Hardware and Software: A milestone indicating when the hardware and software components are integrated and tested together to ensure proper functionality.

d) Final Product Assembly and Testing: A milestone marking when the final system product is assembled, tested, and verified to meet the acceptance criteria.

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

Hardware design involves creating a functional circuit, while schematics provide a graphical representation of that design. In the smart room hardware design, several key components are included. At the core is the Arduino Uno microcontroller, which handles input signals, processes data, and manages the system. The design also includes a DHT11 sensor for monitoring temperature and humidity. A button is used to control the system's power status, relays are employed to switch the AC and/or humidifier on and off, and LEDs are used to provide visual feedback. These components collectively establish the smart room's foundation, with the Arduino Uno as the central control unit, the DHT11 sensor for temperature and humidity detection, the button for system power control, the relays for operating the AC and/or humidifier, and the LEDs for providing visual feedback.

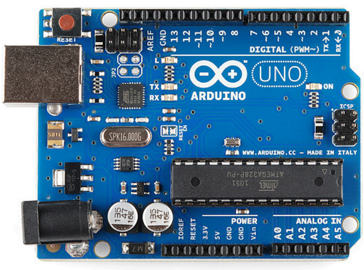


Fig 2. Microcontroller Arduino Uno

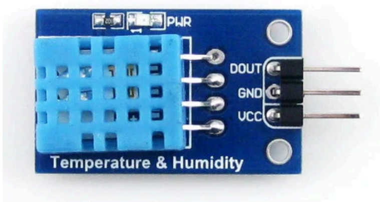


Fig 3.. DHT11 Sensor

Fig 4. Push Button

Fig 5. Relay Module

The schematic diagram illustrates the interconnections between various components in the smart room project. It features two Arduino boards, with one acting as the master and the other as the slave. The master Arduino, linked to the slave Arduino, is connected to another three devices: DHT11 sensor, push button, and relays. It monitors the temperature and humidity, controls whether the data being sent to slave is the temperature or humidity, and controls the relays to turn on/off the AC and/or humidifier. The slave Arduino is only connected to the master and also a serial monitor. The connection between slave and master Arduino is used for sending the data: temperature and/or humidity, from master to slave. The data will be used to show it on the serial monitor, providing another visual feedback. This schematic diagram visually depicts the relationships and connections between the components, emphasizing their roles within the smart room.

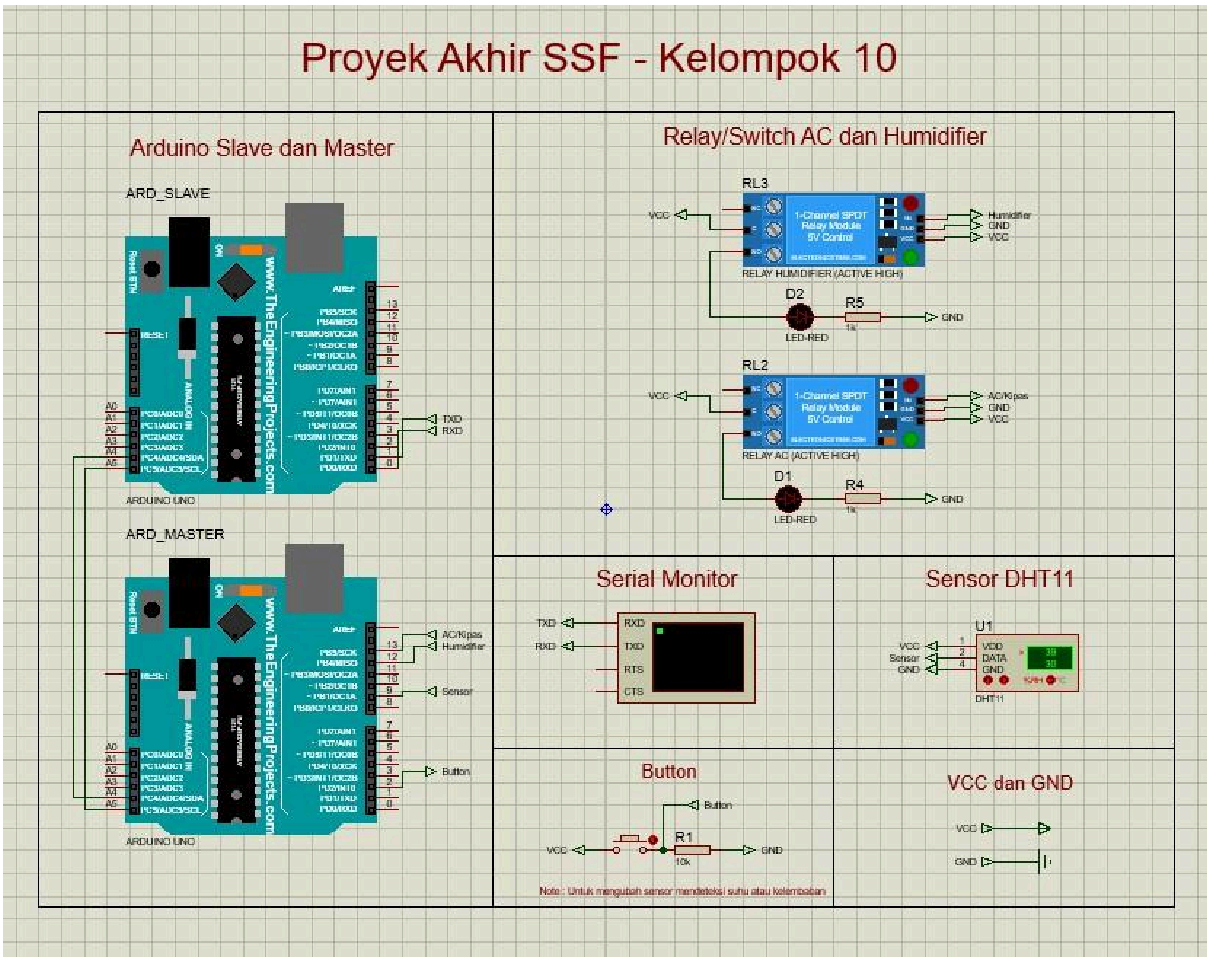


Fig 6. Schematic of System's Hardware

2.2 SOFTWARE DEVELOPMENT

The software development for the smart room entails programming Arduino boards to automate the system. The DHT11 sensor serves as an input device for detecting temperature and humidity, with the Arduino master device receiving this input. A button on the master device selects whether the data being sent to the slave is temperature or humidity data from the DHT11. The master then transmits the selected data to the Arduino slave using the I2C protocol. The slave interprets this data and displays it on the serial monitor, while the master uses the data from DHT11 (temperature and humidity) to control the relays. The development process involves setting up the Arduino boards, coding for the master and slave devices, integrating and testing their functionality, optimizing performance, and documenting the process. This ensures a reliable and efficient software solution for the smart room.

In summary, the software development for the smart room includes programming the Arduino master to receive input from the DHT11 sensor, select which data to send, and communicate it to the Arduino slave via I2C. The slave device displays the received data on the serial monitor, and the master controls the relays based on the DHT11 input. The process encompasses setting up the Arduino boards, coding for both master and slave devices, integrating and testing their functionality, optimizing performance, and documenting the process. This ensures an effective software solution for automating the curtain system. The flowchart for this software development is as follows:

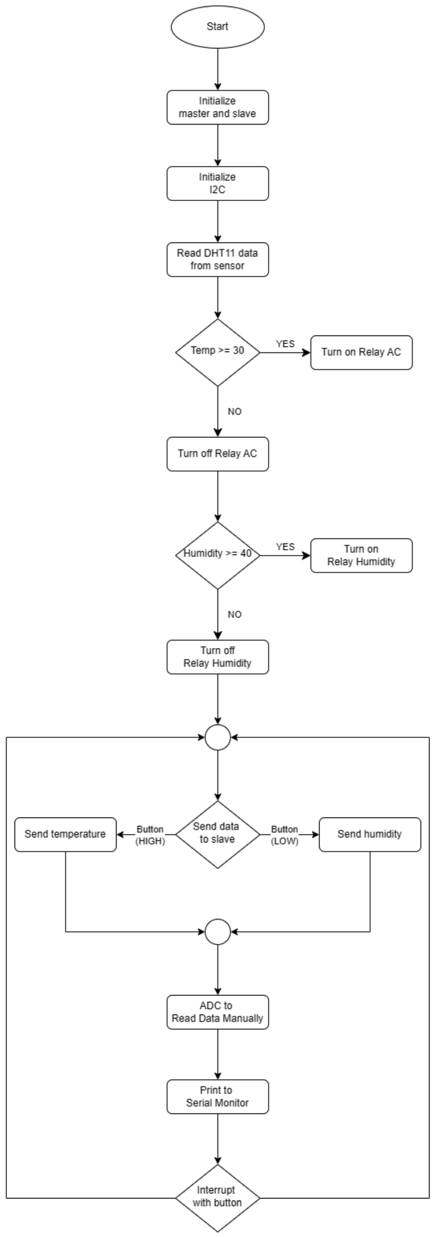


Fig 7. Flowchart

2.3 HARDWARE AND SOFTWARE INTEGRATION

During the integration phase, the initial task was to connect the hardware components to the Arduino Uno microcontroller. This required physically wiring the sensors, output devices, and LEDs to the designated pins on the Arduino board. Each hardware component was allocated a specific pin for sensor input, output device management, or LED indication.

After correctly connecting the hardware components, the next phase involved setting up communication between the software and hardware. The software code, written in assembly language, was then uploaded to the Arduino board. This code included instructions that determined the system's behavior, such as the relay's response to signals from the sensor.

In the integration phase, testing and debugging were conducted to ensure smooth interaction between the hardware and software. This process involved confirming that the sensors accurately detected the intended input signals and that the software properly controlled the output devices in response. Any problems or errors were identified and resolved by making necessary adjustments and refinements to both the hardware and software components.

The primary objective of the integration phase was to attain seamless integration between the hardware and software. This required the hardware components and software code to function together harmoniously as an unified system. The hardware needed to accurately detect environmental conditions (temperature and humidity), while the software had to precisely interpret and respond to these inputs, effectively controlling the relay's response.

Successfully integrating the hardware and software components resulted in a smart room that could respond to variations in temperature and humidity. This seamless integration allowed the system to offer convenience and an improved user experience.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

Testing an automatic temperature and humidity control system built in assembly language using an Arduino ATmega328P involves verifying the functionality and performance of various components. This system uses temperature and humidity sensors to control electronic devices such as air conditioners and humidifiers automatically based on detected environmental conditions.

First, testing is carried out on the temperature sensor which is responsible for detecting environmental temperature. This test is carried out by simulating various temperature conditions to ensure the sensor responds accurately. The test results show that the sensor successfully detects changes in temperature and triggers the relay to turn on the AC when the temperature exceeds 30°C, and turns off the AC when the temperature below 30°C.

Second, the humidity sensor is tested to ensure its ability to detect environmental humidity levels. This test also simulates various humidity conditions. The humidity sensor successfully detects changes in humidity and triggers a relay to turn on the humidifier when the humidity is below 40%, and turns off the humidifier when the humidity rises above 40%.

Next, tests were carried out on the relays that control the AC and humidifier. The relay for the AC is tested to ensure that it can control AC power properly based on input from the temperature sensor. Test results show that the relay works well, turning on the AC when the temperature is above 30°C and turning it off when the temperature drops below 30°C. Likewise, the relay for the humidifier successfully turns on the humidifier when the humidity is below 40% and turns it off when the humidity rises above 40%.

Overall, testing of automatic temperature and humidity control systems shows that all components work in harmony to provide accurate and reliable results in controlling temperature and humidity based on detected environmental conditions. This system successfully integrates temperature sensors, humidity sensors, and relays to control the AC and humidifier automatically, thus ensuring a comfortable and stable environment according to predetermined parameters.

3.2 RESULT

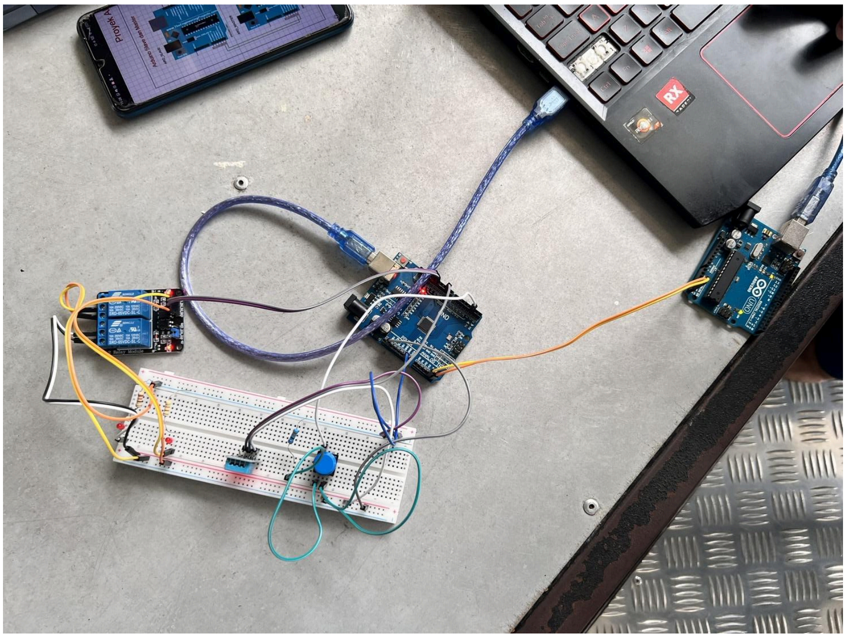


Fig 8. Testing Process

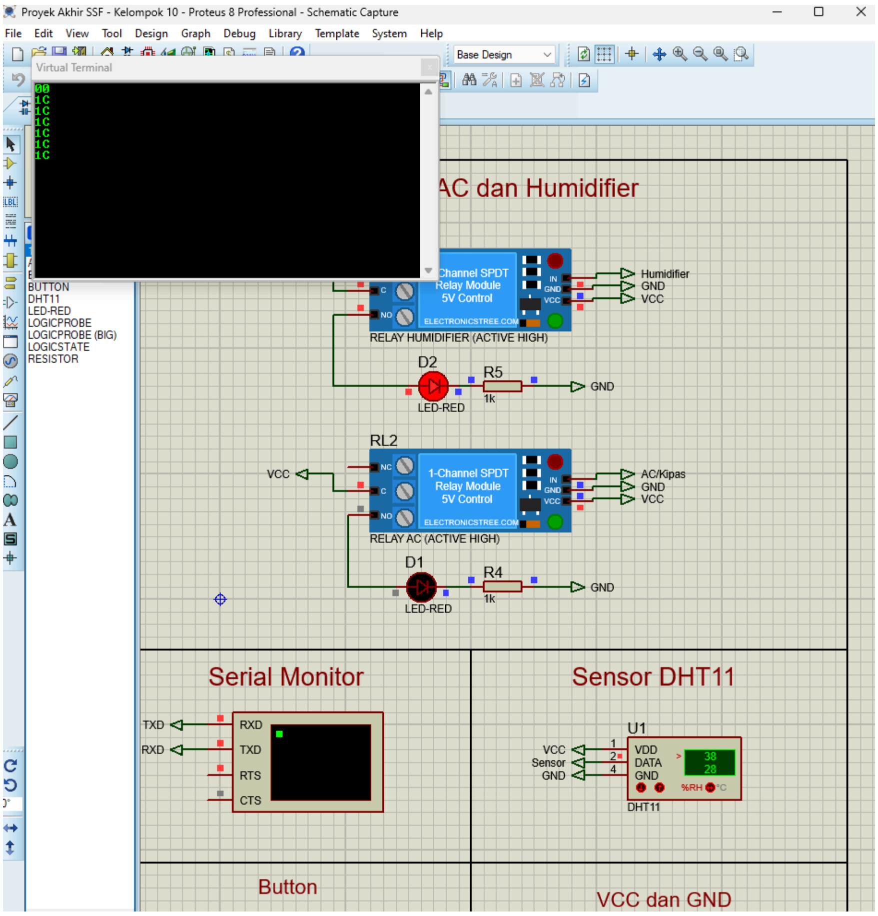


Fig 9. Testing Result when Humidity is lower than 40%

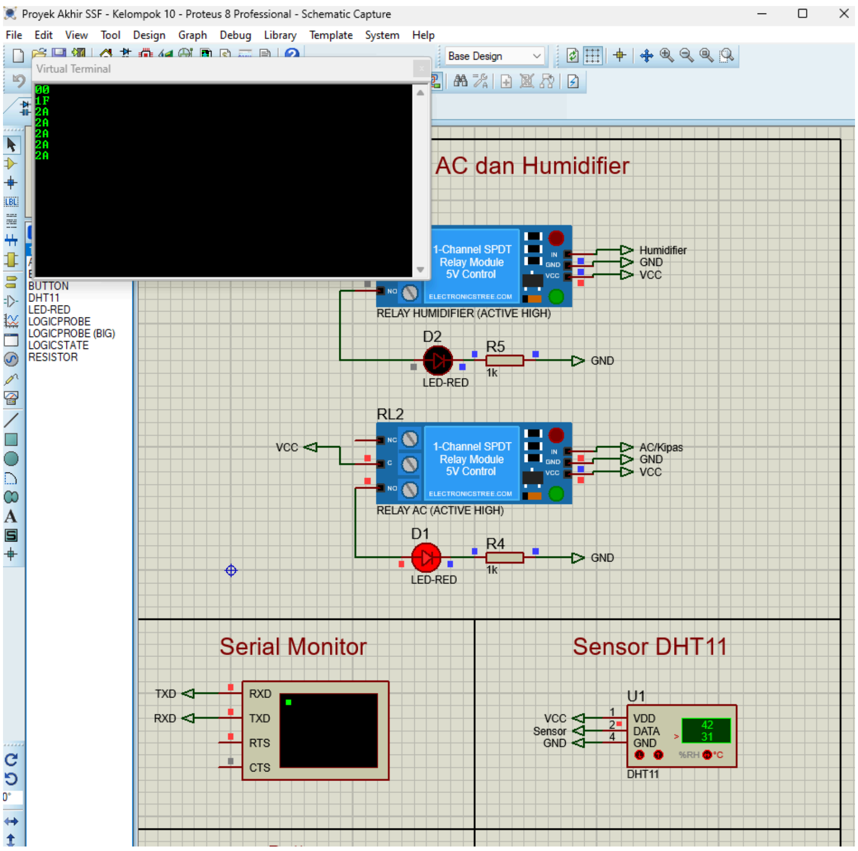


Fig 10. Testing Result when Temperature is lower than 30°C

Testing of the Arduino ATmega328P based automatic temperature and humidity control system shows satisfactory results in various environmental scenarios. In testing the temperature sensor, the sensor succeeded in detecting temperature changes accurately and triggered the relay to turn on the AC when the environmental temperature exceeded 30°C. This test involves simulating temperatures above and below a threshold of 30°C, where the sensor consistently responds appropriately by turning the AC on and off as required.

Humidity sensor testing also shows accurate results. The humidity sensor is able to detect the environmental humidity level precisely and triggers a relay to turn on the humidifier when the humidity drops below 40%. When the humidity increases above 40%, the relay successfully turns off the humidifier. Testing is carried out by simulating various humidity conditions, ensuring sensors and relays function according to design.

Additionally, testing of the relays that control the air conditioner and humidifier showed reliable performance. Relays for air conditioners and humidifiers are tested to ensure fast and precise response based on input from temperature and humidity sensors. The AC relay turns the AC on and off quickly when the temperature changes past the 30°C threshold, while the humidifier relay turns the humidifier on and off precisely when the humidity changes past the 40% threshold.

Overall, system testing shows that all components – temperature sensors, humidity sensors, and relays – work synergistically to control environmental temperature and humidity automatically and accurately. The system manages to maintain comfortable environmental conditions by activating and deactivating devices according to predefined parameters. The results of this test confirm that this Arduino-based automatic control system can be relied on for daily use in regulating room temperature and humidity.

3.3 EVALUATION

Automatic temperature and humidity control program using Arduino ATmega328P shows several main advantages. The temperature and humidity sensors function accurately, activating the air conditioner when the temperature exceeds 30°C and the humidifier when the humidity is below 40%. This system runs efficiently with fast response, thanks to the use of assembly language which maximizes microcontroller performance.

However, this program has several drawbacks. Using assembly language requires a deep understanding of the hardware, which can make maintenance and updates difficult for unfamiliar users. The program is also less flexible than high-level languages such as C or Python, which limits the addition of new features. The system also lacks a user interface for manual monitoring and control, and uses fixed thresholds that cannot be changed without modifying the code.

To upgrade the system, it is recommended to use a high-level language such as C or C++ to facilitate maintenance and development. Developing the user interface will increase comfort and flexibility of use. Adding the ability to dynamically change temperature and humidity thresholds will make the system more adaptive. Further testing under various conditions is required to ensure long-term reliability. More complete documentation will also help users and developers better understand and operate the system. With these improvements, automatic temperature and humidity control systems can become more reliable and easier to use.

CHAPTER 4

CONCLUSION

The Smart Room presents a sophisticated yet practical solution for maintaining optimal environmental conditions within a room through automation and user intervention. By employing a master-slave configuration with two Arduino Uno microcontrollers, the system efficiently distributes tasks, ensuring reliable and responsive climate control. The master Arduino acts as the brain, continuously gathering data from the DHT11 sensor and making decisions based on predefined thresholds, while the slave Arduino handles the execution of these decisions, such as operating relays to control the air conditioner and humidifier. This division of labor not only enhances the system’s efficiency but also minimizes the processing load on each microcontroller.

Key to the system's functionality is the continuous monitoring of temperature and humidity levels by the DHT11 sensor, which provides real-time data to the master Arduino. The master Arduino’s ability to analyze this data and promptly activate or deactivate the air conditioner and humidifier ensures that the room’s environment remains comfortable and healthy. The integration of relays as control mechanisms for these appliances underscores the system's reliability, allowing for precise and timely responses to environmental changes. Additionally, the inclusion of an LED indicator and a manual override button offers users immediate feedback and control, enhancing the user experience by providing a clear indication of the system's status and the flexibility to manually adjust settings when necessary.

Overall, the Smart Room exemplifies an effective blend of automation and user control, leveraging the capabilities of Arduino microcontrollers and sensors to maintain optimal living conditions. Its thoughtful design, which incorporates real-time monitoring, efficient communication protocols, and user-friendly interfaces, demonstrates a robust approach to smart home technology. By ensuring that temperature and humidity levels are constantly regulated, the system not only enhances comfort but also promotes better air quality, making it a valuable addition to any modern living space.

REFERENCES

[1] W. Storr, “Analogue to digital converter (ADC) basics,” Basic Electronics Tutorials, https://www.electronics-tutorials.ws/combination/analogue-to-digital-converter.html (accessed May 26, 2024).

[2] Nebel, “Real programmers write assembly code,” Arduino Craft Corner, https://arduino-craft-corner.de/index.php/2021/10/11/real-programmers-write-assembly-code/ (accessed May 26, 2024).

[3] “Arduino - humidity sensor,” Tutorialspoint, https://www.tutorialspoint.com/arduino/arduino_humidity_sensor.htm (accessed May 26, 2024).

[4] “Assembly via Arduino (part 19) - dht11 sensor,” YouTube, https://youtu.be/vnLpzvkCUq8 (accessed May 26, 2024).

[5] “Assembly via Arduino (part 15) - programming spi,” YouTube, https://youtu.be/D2CS5N_e3NE (accessed May 26, 2024).

[6] “Assembly via Arduino (part 17) - programming I2C,” YouTube, https://youtu.be/N0tmYhU9sN8 (accessed May 26, 2024).

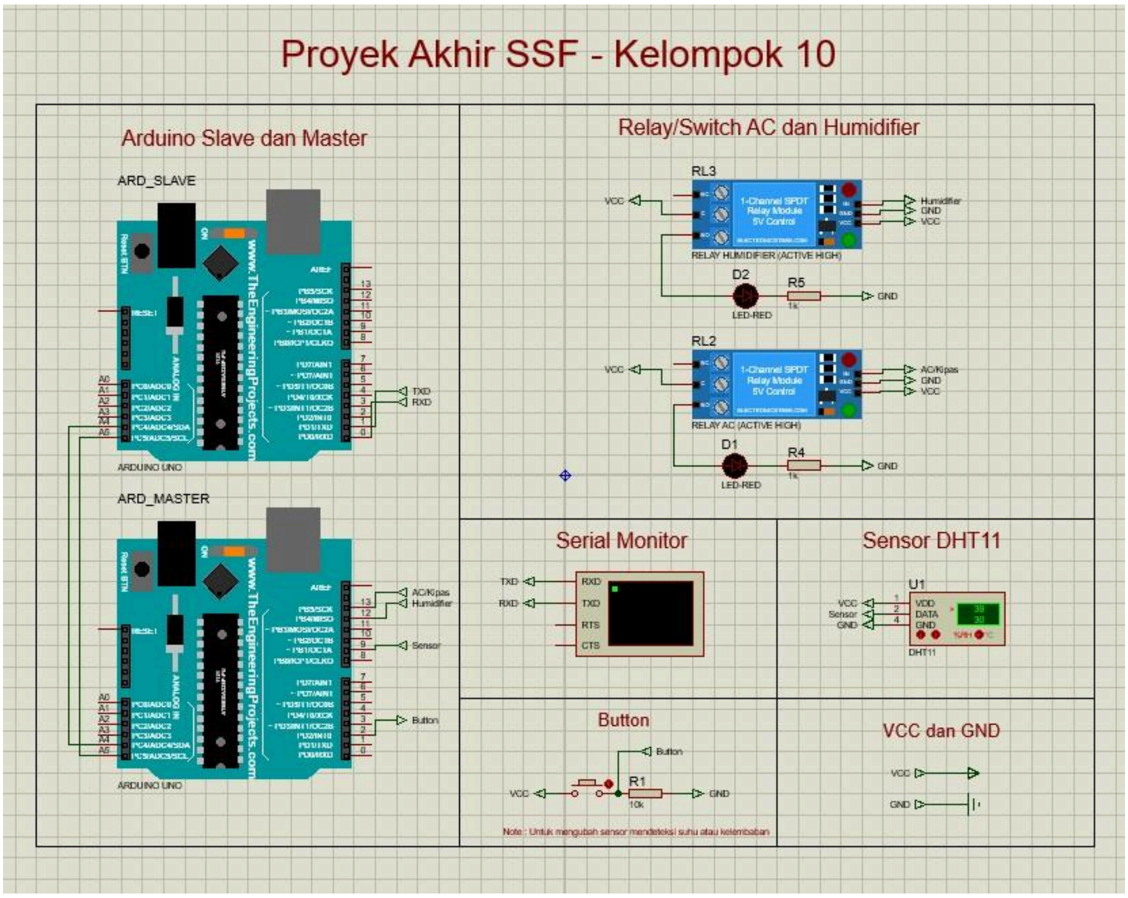
[7] “Assembly via Arduino (part 7) - programming serial port,” YouTube, https://youtu.be/a6LGLwfO154 (accessed May 26, 2024).

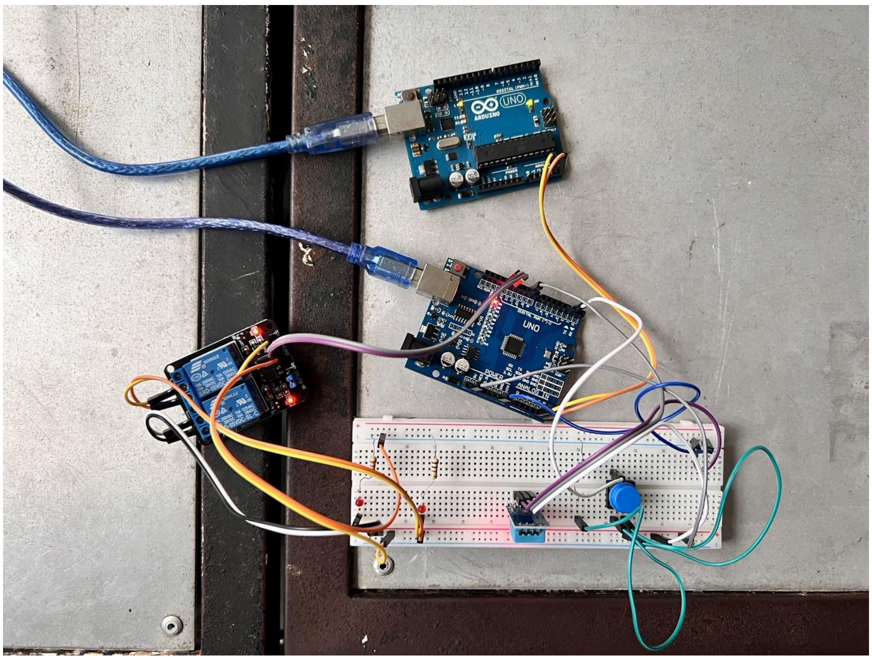
[8] “Arduino I2C,” Arduino, https://www.electronicwings.com/arduino/arduino-i2c (accessed May 26, 2024).

[9] “Arduino & Serial Peripheral Interface (SPI),” docs.arduino.cc, https://docs.arduino.cc/learn/communication/spi/ (accessed May 26, 2024).

APPENDICES

Appendix A: Project Schematic





Appendix B: Documentation