AN ARDUINO-CONTROLLED SMART SYSTEM FOR EFFICIENT CARPET MOISTURE REDUCTION

[1] Narendra Ahmad, [2] Muhammad Wicaksono

[1,] Master of Electrical Engineering, Faculty of Engineering, Diponegoro University, Semarang, Indonesia

[2,]Electrical Engineering, Faculty of Engineering, Diponegoro University, Semarang, Indonesia

Email: Narendra21356@std.undip.ac.id

Abstract—In various sectors, the increasing demand for efficiency and innovation has made the development of technology to address daily challenges highly relevant. This study aims to design and construct an automated and eco-friendly carpet dryer prototype controlled by an Arduino Mega 2560 microcontroller. The system is specifically engineered to overcome the common issue of slow and inefficient carpet drying, particularly in densely populated urban environments.

The research methodology involves both hardware design and software development. The hardware components include heating elements, a fan motor, and sensors for detecting temperature and humidity. The software was developed using the C++ programming language. The drying process is initiated by a DHT11 sensor that detects carpet moisture levels. If the humidity surpasses a predefined threshold, the system automatically activates the heating element and fan to accelerate the evaporation process. Once the desired moisture level is reached, the system will automatically shut down.

Experimental results indicate that the prototype effectively reduces carpet humidity. The use of Arduino as the central controller enables the system to operate autonomously and respond dynamically to environmental conditions. Furthermore, its compact and energy-efficient design positions this prototype as a viable and sustainable solution. This research demonstrates that microcontroller technology can be effectively applied to create practical, efficient, and automated solutions for daily life.

Keywords—Arduino, SHT11 Sensor, Automated System, Carpet Dryer, IoT (Internet of Things).

1. INTRODUCTION

Carpet drying is an essential process in both domestic and commercial cleaning maintenance. However, conventional drying methods that rely on sunlight or natural ventilation are often inefficient, time-consuming, and highly dependent on weather conditions. These limitations can lead to the growth of mold and bacteria due to residual moisture, which can ultimately cause health issues and unpleasant odors. Furthermore, in urban areas with limited space, the drying process becomes even more difficult and impractical.

Technological innovation, particularly in the fields of electronics and microcontrollers, offers a potential solution to these challenges.

ISSN xxxx-xxxx (online) DOI 10.30659/pulse-jeib

Microcontrollers, such as the Arduino series, have proven effective in accurately and efficiently controlling various automated systems. The utilization of this technology enables the creation of devices that not only accelerate the drying process but also operate automatically without continuous manual intervention, all while minimizing energy consumption.

This study aims to design and develop a smart carpet drying prototype using the Arduino Mega 2560 microcontroller. The system integrates a heating element, fan motor, and DHT11 humidity sensor into an automated device to improve drying efficiency compared to traditional methods, with attention to energy use and eco-friendliness. The research covers hardware and software design, system testing, and performance evaluation under different humidity conditions, contributing to smart home technology and microcontroller-based automation.

Indoor carpet drying is traditionally manual, slow, and inconsistent, leading to prolonged moisture retention and hygiene risks. This work presents a low-cost, Arduino-controlled dryer that automates the process using closed-loop feedback from an SHT11 sensor inside the drying chamber. Calibration shows the SHT11 provides reliable inputs for control (average error ≈ 0.97% for temperature and 10% for humidity), enabling threshold-based actuation of a heater and two fans that alternate to maintain airflow and stabilize chamber temperature. Bench tests demonstrate that the prototype dries a very wet carpet to about 42% humidity in roughly 60 minutes, indicating practical effectiveness for small facilities, home services, and lab teaching environments.

II. METHOD

The research employed a quantitative approach with an experimental method to design, develop, and evaluate the performance of an automated carpet dryer prototype. The entire research process was divided into several key stages: system design, hardware development, software programming, and prototype performance testing. The overall research flow is illustrated in Figure 2.1.

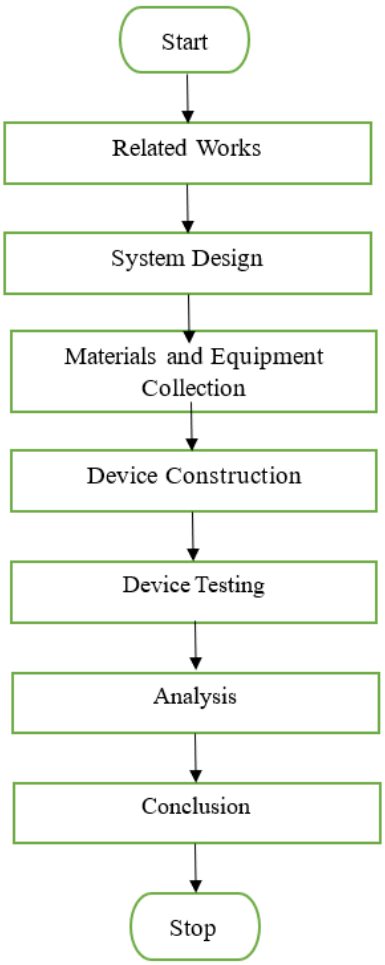


Figure 2.1. Research Flowchart

Figure 3.1. Research Flow Diagram. This diagram illustrates the logical

ISSN xxxx-xxxx (online) DOI 10.30659/pulse-jeib

sequence of steps followed throughout the study, from initial literature review to final conclusions.

System Design

The system design began with an analysis of the functional requirements for an efficient and automated carpet drying system. Based on this analysis, the system architecture was designed as a block diagram. The system consists of three main components: a data acquisition unit, a processing unit, and an actuator unit. The interconnection of these components is shown in Figure 2.2.

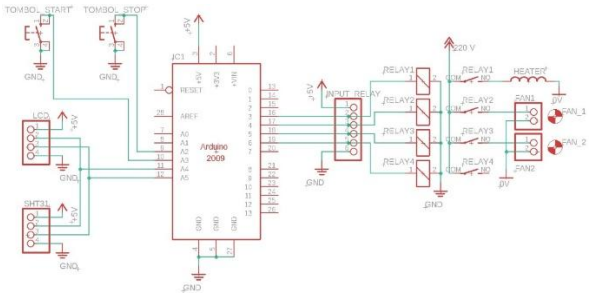


Figure 2.2. Hardware Wiring Design

The main functions of each block are detailed below:

- **Input:** The system receives input from a push button to start the process and from the SHT11 sensor to monitor temperature and humidity. A 5V DC power supply powers the microcontroller and related components, while a 220V AC power supply is used for the heater.
- **Processor:** An Arduino Uno (Atmega 328) microcontroller acts as the central control unit, managing all inputs and outputs.
- **Output:** An LCD provides real-time monitoring of temperature and humidity. A relay functions as a switch to activate or deactivate the heater and fan, which are the main actuators for the drying process.

Hardware Development

Hardware development involved the assembly of the selected components according to the system design. The wiring and connections were designed to ensure a stable and safe circuit. The wiring diagram, as shown in Figure 2.3, details the connections between the microcontroller and all other components.

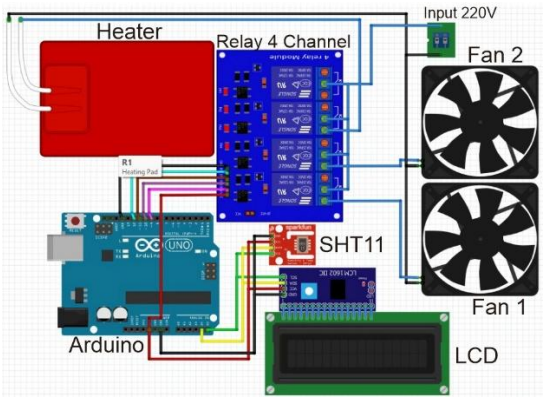


Figure 2.3. Hardware Wiring Diagram

Figure 2.3. Hardware Wiring Diagram. This diagram provides a detailed view of the electrical connections between the microcontroller and all peripheral components. The key components used in the prototype are summarized in Table 2.1.

Table 2.1. List of Main Hardware Components.

Component	Function
Arduino Uno Microcontroller	Serves as the core processor for controlling the system.
LCD 16x2 with I2C Module	Provides the user interface to display system status and sensor data.
Relay 5V	Functions as an electronic switch to control the high-voltage heater and fan.

ISSN xxxx-xxxx (online) DOI 10.30659/pulse-jeib

Component	Function
SHT11 Sensor	Measures temperature and humidity, providing essential input for control.
Fan	Circulates air to accelerate the drying process.
Heater	Generates heat to reduce carpet moisture.
Power Supply	Supplies the required power (5V for logic and 220V for actuators).

This architecture ensures a clear and reliable flow of data and control signals, enabling the automated and efficient operation of the carpet dryer prototype.

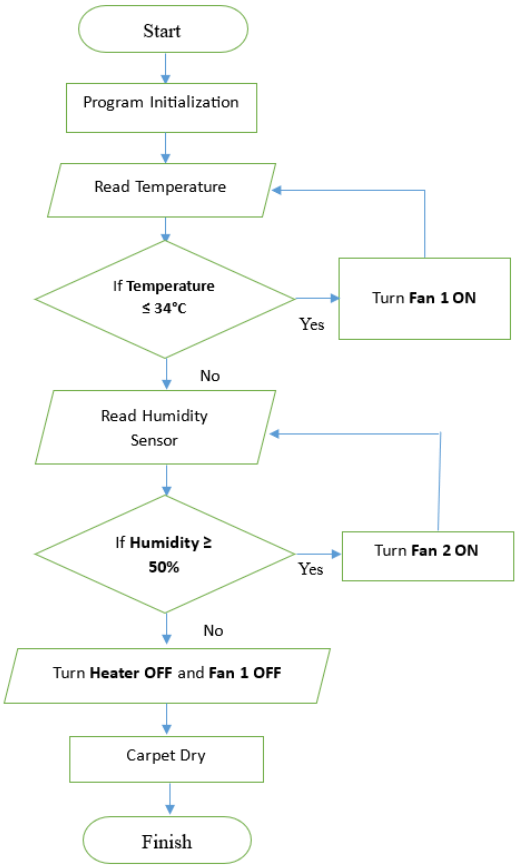


Figure 2.4. Software System Flowchart.

This flowchart illustrates the control logic of the program, detailing how sensor data is processed to activate and deactivate the actuators.

Software Development

The system's software was developed using the C++ programming language within the Arduino IDE. The program's logic is structured to manage the drying process automatically based on sensor readings. A detailed flowchart of the software logic is presented in Figure 2.4.

The software also includes routines to display real-time sensor data and system status on the LCD, providing visual feedback to the user. An example of the simplified control logic is shown in the following pseudo-code:

```
void loop() {
  float humidity = sht.readHumidity();
  float temperature = sht.readTemperature();

  if (humidity > 50) {
    if (temperature < 34) {
      turnOnHeaterAndFan();
    }
  } else if (humidity <= 30) {
    turnOffHeaterAndFan();
  }
}
```

The provided pseudo-code snippet illustrates the core **control logic** of the software. The “loop()” function, which executes continuously, begins by reading real-time **humidity** and **temperature** data

ISSN xxxx-xxxx (online) DOI 10.30659/pulse-jeib

from the SHT sensor. A primary conditional statement checks if the humidity is above 50%, which acts as the main trigger for the drying process. Nested within this is a secondary condition that ensures the heater is only activated if the temperature is below a specific threshold of 34°C, preventing overheating and **ensuring energy efficiency**. If both conditions are met, the system turns on the heater and fan. Conversely, an "else if"statement monitors the humidity level, and if it drops to 30% or below, the system automatically deactivates both the heater and fan, indicating the **drying process is complete**. This logic ensures the system operates in a fully **automated** and responsive manner.

Testing and Analysis

The prototype's performance was evaluated through a series of tests conducted in a controlled environment to ensure the validity of the results. The testing procedure was divided into several key phases: a functional test to verify the operational integrity of each component, a performance test to measure the drying time from a high initial moisture level to a desired target, and an energy efficiency test to analyze the system's power consumption. The data collected from these tests were then statistically analyzed to determine the effectiveness and efficiency of the developed prototype. The results and findings from this analysis will be presented in detail in the subsequent chapter.

The prototype integrates an Arduino microcontroller, an SHT11 sensor positioned within the chamber, a resistive heater, and two DC fans driven via relay modules. The control loop samples temperature and humidity, compares them to preset thresholds with simple hysteresis, and then (i) energizes the heater to elevate chamber temperature and (ii) alternates Fan 1/Fan 2 to sustain circulation and avoid stagnant zones. The system runs autonomously once started, requiring no manual switching during operation. Sensor calibration preceded testing to minimize bias in the feedback signal. Performance evaluation used repeated drying runs from a saturated starting condition; humidity and temperature were logged throughout each run, and time-to-target (≈42% RH) was recorded as the primary effectiveness indicator, alongside qualitative checks for thermal stability and airflow consistency

III. RESULTS AND DISCUSSION

This section presents the results and analysis from the prototype testing, covering the power supply, relay, sensor calibration, and the overall performance of the automated drying system.

Power Supply Circuit Test

The power supply circuit was tested to evaluate its output voltage and ensure it met the microcontroller's requirements. The test aimed to confirm that the generated voltage was stable and suitable for the system. The results of the power supply voltage test are shown in Table 3.1. A visual representation of the test procedure is provided in Figure 3.1.

Table 4.1. Power Supply Voltage Test Results.

No.	Input Voltage	Output Voltage
1	235 V (AC)	6 V (DC)
2	235 V (AC)	5 V (DC)

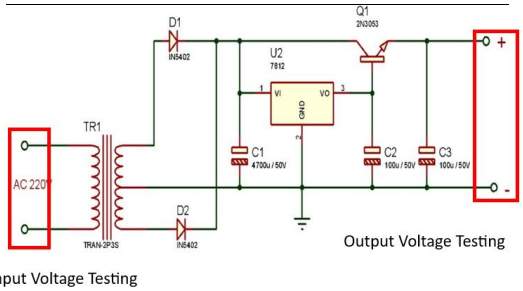


Figure 3.1. Power Supply Voltage Test.

ISSN xxxx-xxxx (online) DOI 10.30659/pulse-jeib

The test confirmed that the power supply circuit reliably produced the required 5V (DC) output, which is essential for the stable operation of the microcontroller and other low-voltage components.

4.1.Relay Test

The relay test was conducted to validate its functionality as a logical AND gate, controlled by the microcontroller. The relay is triggered by a 5V DC signal from the microcontroller, which in turn switches the 220V AC power to the heater and fan. The results of the relay voltage test are shown in Table 3.2 and depicted in Figure 3.2.

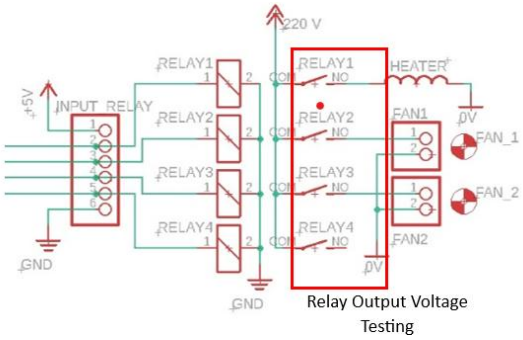


Figure 3.2. Relay Voltage Test.

Figure 3.2 displays the physical test configuration, where an input signal from the microcontroller controls the relay. Meanwhile, Table 3.2 provides the detailed results of that test. When the DC input to the relay's control pin is **0V** (low logic), the relay's Normally Closed (NC) output is **220V AC**, and the Normally Open (NO) output is **0V AC**. Conversely, when the DC input is **5V** (high logic), the relay switches, causing the NC output to become **0V AC** and the NO output to become **220V AC**.

This effectively demonstrates that the relay functions as a switch controlled by a digital signal from the microcontroller, successfully switching the high-voltage (220V AC) circuit to control the heater and fan.

Table 3.. Relay Voltage Measurement Results.

Input (V DC)	NC (Normally Closed) Output (V AC)	NO (Normally Open) Output (V AC)
0	220 V (AC)	0 V (AC)
1	0 V (AC)	220 V (AC)

The results confirm that the relay operates as intended, successfully controlling the high-voltage 220V AC circuit for the heater and fan using a low-voltage 5V DC signal from the microcon.

Sensor Calibration

The SHT11 sensor was calibrated by comparing its readings with a digital room hygrometer. The data collected included temperature (in Celsius) and humidity (in percentage). The test aimed to determine the accuracy of the sensor. The results of the temperature and humidity tests are presented in Table 3.3 and Table 3.4, respectively. The data acquisition process is shown in Figure 3.3 and Figure 3.4.

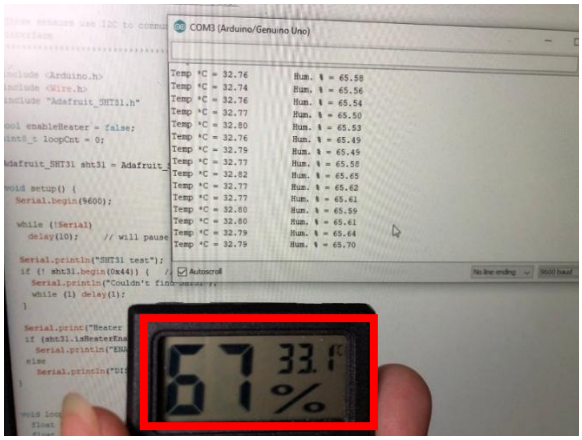


Figure 3.3. Temperature Sensor Test.

Figure 3.3. visually depicts the data acquisition process, providing a physical representation of the test setup. The detailed results from this process are presented in Table 3.3. This table shows a minute-by-minute comparison of the temperature readings from the SHT11 sensor against the reference digital hygrometer, confirming the sensor's accuracy and reliability.

ISSN xxxx-xxxx (online) DOI 10.30659/pulse-jeib

Table 3.3. Temperature Sensor Test Results.

Minute	SHT11 (°C)	Hygrometer (°C)	Difference (°C)	Error (%)
1	32.76	33.10	0.34	1.03
5	32.80	33.10	0.31	0.91
9	32.82	33.10	0.28	0.85
12	32.80	33.10	0.31	0.91
15	32.79	33.10	0.31	0.94
Average	–	–	–	**0.97**

Following the temperature calibration, the humidity readings of the SHT11 sensor were also tested against the digital hygrometer. The results, as shown in Table 3.4, indicate that the sensor provides readings with an average error of 10%, which is considered acceptable for the intended application of the automated drying system. Taken together, both tables validate the SHT11 sensor's suitability as a reliable and accurate component for the prototype.

Table 3.4. Humidity Sensor Test Results.

Minute	SHT11 (%)	Hygrometer (%)	Difference (%)	Error (%)
1	58.89	64	5.11	7.98
5	57.62	64	6.38	9.97
10	56.85	64	7.15	11.17
12	56.67	64	7.33	11.45
15	56.42	64	7.58	11.84
Average	–	–	–	**10.00**

The humidity sensor test showed an average reading of 57.29% compared to the digital hygrometer's 64% average. The calculated error value for the humidity sensor was 10%, which is within an acceptable range for the system's intended application.

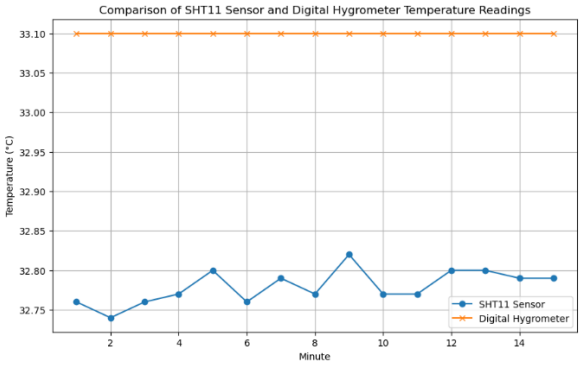


Figure 3.4. Comparison of SHT11 Sensor and Digital Hygrometer Humidity Readings

This graph compares the humidity data read by the SHT11 sensor with the reference data from a digital hygrometer. The blue line represents the SHT11 sensor readings, which gradually decrease over time, while the straight orange line shows the constant value from the reference hygrometer. The gradual decrease in the SHT11 sensor's humidity readings indicates that drying is indeed occurring. Although there is a constant offset between the two lines, their change patterns remain similar, confirming that the sensor is functioning properly, is accurate, and can be relied upon to detect changes in humidity.

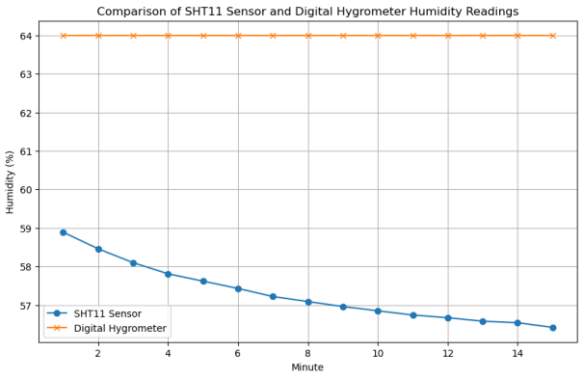


Figure 3.5. Temperature and Humidity of the Drying Chamber during Automated Process

This graph illustrates how the temperature and humidity of the drying chamber change over time during the

ISSN xxxx-xxxx (online) DOI 10.30659/pulse-jeib

automated process. The blue line (left Y-axis) tracks the humidity, which gradually decreases from approximately 55% to 42%. The red line (right Y-axis) shows the temperature, which fluctuates between 33°C and 36°C. These temperature fluctuations correspond to the program's logic, where the heater turns on to increase the temperature and dry the air, then turns off when the temperature reaches the threshold, indicated by the repeating peaks and valleys. This graph visually demonstrates that the automated control system works effectively.

Drying Process Test

The automated drying process was tested to verify its functionality based on the predefined temperature and humidity thresholds. The control logic was set to activate the heater and Fan 1 when humidity exceeded 50% and the temperature was below 33°C. Fan 2 was designed to activate when the temperature reached 36°C, indicating a need for cooling and air circulation. The system would run continuously until the humidity dropped to 42%, at which point the entire system would shut down. The results of the automated drying test are presented in Table 3.5.

Table 4.5. Automated Drying Test Results.

Minute	Temp. (°C)	Humidity (%)	Fan 1 Status	Heater Status	Fan 2 Status	Carpet Condition
1–8	33–35.5	54–55	ON	ON	OFF	Very Wet
9–14	35–36	53–54	OFF	OFF	ON	Very Wet
15–20	34.5–35	52–53	ON	ON	OFF	Wet
21–24	36	51–52	OFF	OFF	ON	Wet
25–31	34–35.5	50–51	ON	ON	OFF	Damp
32–35	36	49–50	OFF	OFF	ON	Damp
36–40	35–35.5	48–49	ON	ON	OFF	Slightly Dry
41–45	34–35	47–48	ON	ON	OFF	Slightly Dry
46–48	35–36	46	ON	ON	OFF	Slightly Dry
49–51	36	45–46	OFF	OFF	ON	Slightly Dry
52–58	34–35.5	43–45	ON	ON	OFF	Dry
59–60	36	42	OFF	OFF	OFF	Dry

The results in Table 3.5 demonstrate the effectiveness of the automated control system. The heater and Fan 1 were activated when the humidity was high and temperature was within the operational range, successfully initiating the drying process. As the temperature rose to the threshold of 36°C, the system intelligently switched off the heater and Fan 1 and activated Fan 2 for air circulation, which helps to stabilize the temperature and prepare for the next drying cycle. This intermittent operation is a key feature of the system, ensuring both efficiency and safety. The entire drying process, from a very wet state to a dry state (42% humidity), took approximately 60 minutes. This demonstrates that the prototype can effectively dry a carpet and is responsive to changes in humidity and temperature. The system's automated control logic, based on sensor readings, eliminates the need for manual intervention and provides a consistent and efficient drying solution.

ISSN xxxx-xxxx (online) DOI 10.30659/pulse-jeib

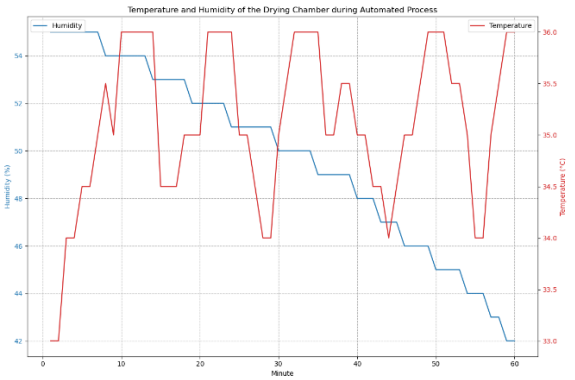


Figure 3.6. Comparison of SHT11 Sensor and Digital Hygrometer Temperature Readings

Figure 3.6 compares the temperature data read by the SHT11 sensor with the reference data from a digital thermometer. The blue line represents the temperature readings from the SHT11 sensor, which show slight fluctuations but remain around the average value of 32.7 °C. The flat orange line indicates the constant reference value of 33.1 °C from the digital thermometer. The consistency shown by both lines, despite the small difference, proves that the SHT11 sensor functions properly and has high accuracy (an average error value of 0.97%) for detecting temperature changes within the drying chamber.

IV. CONCLUSION

The prototype automated carpet dryer was successfully designed and works as intended. Calibration confirmed the SHT11 provides reliable inputs for control—showing an average error of 0.97% for temperature and 10% for humidity. With threshold-based logic, the system automatically drives the heater and alternates fan 1 with fan 2 to sustain airflow and stabilize chamber temperature, operating without manual intervention. Tests show it dries a very wet carpet to ~42% humidity in about 60 minutes, demonstrating effective performance.

For next iterations, improve humidity sensing either by tighter calibration or adopting a higher-accuracy sensor to reduce the ~10% RH error and sharpen control responsiveness. Revisit the chamber layout and fan placement to optimize hot-air circulation, which could cut drying time and energy use. Finally, add a simple UI (touchscreen or Wi-Fi/Bluetooth app) for remote monitoring/control and to log drying histories for users.

The automated carpet dryer achieved its design goals: accurate sensing for control, fully automated heater/fan operation, and effective moisture reduction to ~42% RH in ~60 minutes. These results confirm that a simple, closed-loop architecture can deliver reliable drying without manual intervention. Remaining gaps include the ~10% humidity error (suggesting tighter calibration or a higher-accuracy sensor), chamber aerodynamics that could be refined for faster, more energy-efficient drying, and the absence of a user-friendly interface. Future work should focus on improved RH sensing, airflow optimization, and a basic UI (touchscreen or wireless) for monitoring, control, and run-history logging.

ACKNOWLEDGMENT

The authors thank the Electrical Engineering Program, Faculty of Engineering, Diponegoro University for laboratory access and technical support during the SHT11 calibration and prototype testing. We are grateful to the lab assistants for their help with hardware assembly and power/relay verification, and to the anonymous reviewers for constructive feedback that improved the manuscript. This work received no external funding; any

ISSN xxxx-xxxx (online) DOI 10.30659/pulse-jeib

remaining errors are the authors' responsibility.

REFERENCES

1] F. Agustia Arini, "Prototype Penjemur Pakaian Otomatis Berbsis Arduino Menggunakan Sensor Raindrop Dan Sensor Dht11," Jurnal_Fadhillah Progr. Stud. Tek. Elektro, Fak. Teknol. Inf. Elektro Univ. Teknol. Yogykarta, vol. 1, no. 1, pp. 1–6, 2019.

[2] R. Aulia, R. Fauzan Aulia, and I. Lubis, "Pengendalian_Suhu_Ruangan_Menggunakan_Menggunakan_," J. Tek. Inform. Univ. harapan medan, vol. 6, no. 2502–7131, pp. 1–9, 2021.

[3] T. Andriani, I. Darmawan, A. Jaya, and P. A. Topan, "Rancang Bangun Alat Pengering Ikan Bage Otomatis Menggunakan Sensor SHT11 dan Real Time Clock," Dielektrika, vol. 8, no. 2, pp. 126–130, 2021.

[4] A. Kurniawan, A. Titi, I. Darmawan, and M. Desi, "Journal Homepage : http://jurnal.uts.ac.id/index.php/Altron APLIKASI SENSOR SHT11 DAN SOFT START PADA ALAT PENGERING IKAN BAGE OTOMATIS," vol. 1, no. 1, pp. 18–24, 2022.

[5] F. K. Setyawan, B. Arifin, and M. Ismail, "Prototype Pengatur Kecepatan Blower Prototype of Speed Controller for Ink Roll Dryer Using Bluetooth From Smartphones," Konf. Ilm. Mhs. Unissula 2, pp. 335–343, 2019.

[6] R. Rosmanila, T. Radillah, and A. Sofiyan, "Prototype Lemari Pengering Pakaian Otomatis," I N F O R M a T I K a, vol. 10, no. 1, p. 32, 2018, doi: 10.36723/juri.v10i1.90.

[7] M. I. Hafidhin, A. Saputra, Y. Rahmanto, and S. Samsugi, "Alat Penjemuran Ikan Asin Berbasis Mikrokontroler Arduino UNO," J. Tek. dan Sist. Komput., vol. 1, no. 2, pp. 59–66, 2020, doi: 10.33365/jtikom.v1i2.210.

[8] P. Purwadi and W. Kusbandono, "Peningkatan Waktu Pengeringan dan Laju Pengeringan Pada Mesin Pengering Pakaian Energi Listrik," Pros. Semin. Nas. XI, p. 6, 2017.

[9] S. Samsugi, Z. Mardiyansyah, and A. Nurkholis, "Sistem Pengontrol Irigasi Otomatis Menggunakan Mikrokontroler Arduino Uno," J. Teknol. dan Sist. Tertanam, vol. 1, no. 1, p. 17, 2020, doi: 10.33365/jtst.v1i1.719.

[10] S. Sumanto, S. Achmadi, and A. Mahmudi, "Perancangan Lemari Pengering Pakaian Yang Ergonomis Di Laundry Si Doel Batu," Ind. Inov. J. Tek. Ind., vol. 11, no. 1, pp. 14–21, 2021, doi: 10.36040/industri.v11i1.3409.

[11] I. G. M. N. Desnanjaya and I. B. A. I. Iswara, "Trainer Atmega32 Sebagai Media Pelatihan Mikrokontroler Dan Arduino," J. Resist. (Rekayasa Sist. Komputer), vol. 1, no. 1, pp. 55–64, 2018, doi: 10.31598/jurnalresistor.v1i1.266.

[12] M. Ardi and H. Amri, "Analisa Rancang Bangun Alat Pengering Pakaian Otomatis," JEECAE (Journal Electr. Electron. Control. Automot.

ISSN xxxx-xxxx (online) DOI 10.30659/pulse-jeib

Eng., vol. 4, no. 1, pp. 253–256, 2019, doi: 10.32486/jeecae.v4i1.326.

[13] A. Ubaidillah, K. Handoko, and S. Ida, “OPTIMALISASI KINERJA REPEATER SINYAL GLOBAL SYSTEM FOR MOBILE DENGAN METODE HIGH-MULTI SERIAL (STUDI KASUS : PEGUNUNGAN PEGANTENAN , PAMEKASAN),” no. 4, pp. 140–148, 2023.

[14] Y. M. A. Othman and A. A. A. Eissa, "Design and Development of an Automatic Drying Machine for Agricultural Products Using Microcontroller-Based System," 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Cairo, Egypt, 2021, pp. 1-6.

[15] Al-Sarayreh and F. O. H. Al-Qassem, "Design and Implementation of a Smart Clothes Dryer System Based on IoT," International Journal of Electrical and Computer Engineering (IJECE), vol. 11, no. 5, pp. 4452-4460, 2021.

ISSN xxxx-xxxx (online) DOI 10.30659/pulse-jeib