

Data Acquisition Board by Using Arduino to Measure Weather Temperature and Humidity

Baraa Saad Abdulhakeem¹, Mustafa Maad Hamdi², May Abdulsamad Sadeq³

¹Department of Information System, College of Computer Science and IT, University of Anbar, Ramadi, Iraq.

²Department of Computer Science and IT, College of Computer Science and IT, University of Anbar, Ramadi, Iraq.

³College of Electrical Engineering, University of Anbar, Ramadi, Iraq.

Abstract— This research delves into exploring the Arduino platform, an open-source hardware and software paradigm recognized for its economic viability and simplified design, capturing significant attention within the technology community. The overarching objective is to construct an embedded device utilizing Arduino capable of tracking environmental variables such as temperature and humidity. The study assesses the functionality of this device across diverse temperature and humidity conditions, leveraging sensors and the Arduino microcontroller. The investigation contributes to the understanding of Arduino's versatility, underscored by its compatibility with various devices and its extensibility through code execution in the Integrated Development Environment (IDE). The research comprises two main sections: a theoretical overview elucidating the essential tools and supplies, and a practical section providing a step-by-step procedural guide for wiring and electronics. Notably, the findings reveal the successful attainment of predefined objectives, with the constructed device effectively delivering real-time information on temperature and moisture content within a building. This research enhances our comprehension of Arduino's applicability in creating cost-effective and efficient embedded devices for environmental monitoring, thereby contributing to the broader landscape of open-source technology applications

Keywords— Data Acquisition, Arduino, Temperature, Humidity.

I. INTRODUCTION

In the realm of environmental science and technology, the demand for efficient, cost-effective, and adaptable data acquisition systems has surged, particularly with the escalating impacts of climate change. In response to this imperative, this research endeavors to devise an innovative solution. a Data Acquisition Board powered by Arduino to precisely measure two pivotal weather parameters: temperature and humidity [1]. The profound implications of accurate weather data transcend various sectors, including agriculture, meteorology, and environmental monitoring. Traditional weather measurement systems often grapple with issues of exorbitant costs, limited accessibility, and inadequate adaptability to diverse environmental conditions. these challenges, the overarching objective of this research is to develop a versatile and cost-effective data acquisition solution that seamlessly measures

temperature and humidity while harnessing the accessibility, affordability, and adaptability offered by the Arduino platform [2]. This paper offers doable methods for creating a device that uses an Arduino board and sensors to detect the humidity and temperature of a building or its surroundings and display the results on an LCD (liquid crystal display) and serial monitor. The goal of this study is to monitor the ambient temperature and humidity using the Arduino control board and sensors, then display the results on the LCD screen (Liquid Crystal Display) using the C++ code in the development environment of the Arduino program. The contemporary landscape of weather monitoring systems reveals a pressing need for transformative approaches to address the shortcomings of existing technologies. Conventional systems, characterized by intricate designs and high expenses, hinder the widespread deployment of weather measurement devices. The advent of open-source hardware and software platforms, such as Arduino, has ushered in a paradigm shift by providing accessible and customizable tools for enthusiasts, researchers, and professionals alike. Arduino's cost-effectiveness, user-friendly interface, and compatibility with a myriad of sensors make it an ideal candidate for constructing a Data Acquisition Board tailored to weather measurement. The specific problem at hand involves the development of a data acquisition system that can accurately and efficiently measure temperature and humidity in real-time [3]. As climate change intensifies, understanding and predicting weather patterns become increasingly crucial for sustainable resource management and disaster preparedness. Existing solutions often fall short regarding accessibility, scalability, and cost-effectiveness. Therefore, this research aspires to contribute a novel approach that leverages the capabilities of Arduino to construct a versatile Data Acquisition Board capable of addressing these challenges [4].

The general objective of this research is twofold: first, to design a Data Acquisition Board employing Arduino as the core technology for weather monitoring, and second, to implement a system that delivers accurate, real-time data on temperature and humidity. This objective aligns with the broader aim of enhancing the accessibility and sustainability of weather data acquisition systems, thereby fostering

advancements in environmental monitoring technologies. By creating a system that is not only cost-effective but also adaptable to various environmental conditions, this research endeavors to overcome the limitations posed by current weather measurement technologies [5]. To fulfill these objectives, the research methodology involves a multi-faceted approach. The design and implementation of the Data Acquisition Board will be guided by a thorough review of existing weather monitoring technologies, with a particular emphasis on their limitations and areas for improvement. Arduino's extensive capabilities will be harnessed to integrate sensors capable of accurately measuring temperature and humidity. The research will also involve field testing to validate the performance of the developed system across diverse environmental conditions, ensuring its reliability and adaptability [6]. The theoretical framework of this research encompasses a comprehensive overview of the Arduino platform, its capabilities, and its relevance in the context of weather monitoring. Additionally, an in-depth exploration of existing weather measurement technologies will be undertaken, shedding light on their strengths and weaknesses. This theoretical foundation will inform the design and development of the Data Acquisition Board, ensuring that it addresses critical gaps in current weather monitoring systems. The practical implementation of the research involves the construction of a prototype Data Acquisition Board using Arduino. The board will be equipped with sensors capable of accurately measuring temperature and humidity. The Arduino platform's programming capabilities will be utilized to develop a user-friendly interface for data visualization and analysis. A step-by-step procedural guide will be established, providing clear instructions for the construction, programming, and deployment of the Data Acquisition Board. This practical section aims to serve as a valuable resource for researchers, enthusiasts, and practitioners interested in replicating or further developing the proposed system. The anticipated findings of this research revolve around the successful construction and implementation of a Data Acquisition Board capable of precisely measuring temperature and humidity in real time. Through field testing, the reliability and adaptability of the system across various environmental conditions will be evaluated. The research aims to demonstrate that Arduino, with its open-source nature and extensive community support, can be a game-changer in the realm of weather monitoring technologies. In addition, the expected results include real-time data outputs showcasing the accuracy of temperature and humidity measurements. The developed system is envisioned to exhibit a seamless interface, facilitating user-friendly interaction and data interpretation. Additionally, the research aims to highlight the cost-effectiveness of the proposed solution, emphasizing Arduino's role in democratizing access to advanced weather monitoring technologies. This paper, therefore, seeks to contribute not only to the academic discourse on environmental monitoring but also to the practical implementation of sustainable and accessible solutions for weather data acquisition.

DOI:<http://doi.org/10.24086/cocos2024/paper.1559>

II. RELATED WORK

In the work of [7] the Arduino-based humidity and temperature sensor was tested in two independent running tests. The study design is not explicitly mentioned in the paper. However, based on the description of the study's aim and the methodology used, it can be inferred that the study design is experimental, specifically a device development and testing study. The study is the development of a functional hardware and software system for measuring temperature and humidity, the evaluation of performance characteristics of the Arduino-based sensor through independent tests, and the successful implementation of the system with real-time data recording and storage. The limitations of the study are not explicitly stated in the paper, but based on the information provided, the inferred limitations include a lack of comparison with existing technologies, limited generalizability of findings, and potential obsolescence of the developed system. In the study [8] The temperature and humidity data are transferred through wireless data communication using RF module nRF24L01, DHT11 sensor, and Arduino Uno. The main findings of the paper are: The first one, is that wireless communication is crucial for making the system faster and easier. Second, there is a demand for smart systems with automation to monitor all environmental factors. Lastly, the proposed wireless communication system is cost-effective and has low power consumption. However, the limitations of the study include: the first one, reliance on analog data measurement with human effort may introduce errors and make it difficult to store the values of parameters. The second one, the need for a smart system with automation to monitor all environmental factors indicates that the current system may not fully meet this requirement. Lastly, emphasis on the smart system being accurate, reliable, and cost-effective suggests that there may be limitations in these aspects of the proposed monitoring system. In [9], expanded on this by incorporating multiple sensors, including the DHT22, BH1750, and Arduino Uno, to monitor temperature, humidity, and light intensity in a greenhouse. Another work in [10], used an Arduino board and DHT22 sensor to measure temperature and humidity, emphasizing the need for a functional system with hardware and software integration. These studies collectively demonstrate the potential of Arduino-based data acquisition boards for weather monitoring. In this work [11] The ability to control the climate of a greenhouse was provided for greenhouse environment monitoring. The study design involves the implementation of a system for data acquisition and monitoring in a greenhouse environment, utilizing multiple sensors and control mechanisms to enhance plant growth. The main results include the aim to acquire data from multiple sensors to enhance plant growth, the major factors affecting plant growth, and the system's ability to control the greenhouse climate. However, the limitations of the study include the inability of the system to control the climate of the greenhouse automatically and the suggestion to add an automatic control system to the greenhouse. Lastly, in [12] an

assessment of missing weather data in small data sets is introduced utilizing correlation and meta-analysis of different weather parameters like temperature, humidity, and wind speed. Autoregressive integrated moving average (ARIMA), a well-known artificial model widely used for weather prediction, is evaluated on various sets with missing data. The main findings of the study include the evaluation of the ARIMA model for weather prediction on sets with missing data, the presentation of univariate and multivariate ARIMA models, and the revelation that the imputation of missing data increases prediction accuracy by almost 50%.

I. III. MATERIALS

A. Arduino

A single-board microcontroller for creating interactive and digital things is called an Arduino. Arduino is made to sense its surroundings by using actuators to interact with its surroundings and sensors to receive input signals. A basic motor, sensors, or an LED (light emitting diode) might all be considered actuators. Depending on the paper, ethernet or other devices may be used [13]. The Arduino hardware comes in a variety of forms and designs that allow for various capabilities. Hardware wiring is the foundation of the programming. Windows, Linux, or Mac OS can all be used to run the Arduino program. [14].

Programming for the Arduino allows it to function independently of a computer or other electronic devices.; It is possible to accomplish this using the IDE (Integrated Development Environment) software for Arduino. Given that as the Arduino software and hardware are freely available, there are already a lot of Arduino hardware clones on the market with lots of interesting features, Arduino Uno, a board that is utilized in this paper, is shown in Figure 1. Arduino Uno boards are built on an 8-bit CPU. microcontroller device. Fourteen pins are used for digital input and output six of which are suitable for pulse width modulation (PWM) outputs. It has a 16 MHz quartz crystal or oscillator and six analog inputs. The Arduino Uno board can be connected to a computer via a USB (universal serial bus) cable., ICSP (In-Circuit Serial Programming) header, reset button, and power jack [1]. The Arduino Uno board was given this name since it is the first USB Arduino board in a series. In Italian, uno denotes one.



Figure 1. Arduino UNO SMD R3.

B. Sensors

An electrical signal is produced by a sensor, which is an electronic device that senses changes in physical phenomena. It can transmit data to computers and other electronic devices. As a result, it is a component of the interface that connects the electronics and the surrounding environment. [15]. A sensor's job is to take a physical signal as input and transform it into an electrical signal (voltage). Depending on the type of sensor, it is a semiconductor device made to react when its resistive or capacitive properties change. Transfer function, sensitivity, range, accuracy/uncertainty, hysteresis, linearity, noise, resolution, and bandwidth are the characteristics that define a sensor's performance. [16].

Numerous items and locations use sensors, such as touch-sensitive phone screens and motion-activated light switches, among other things. Since the invention of microcontrollers, the application of sensors has increased dramatically. Robotics, aircraft and aerospace, automobiles, and numerous other applications employ the sensors. [17].

The temperature and humidity sensor DHT11 (Digital Humidity and Temperature sensor) is the sensor used in this work. When connected and programmed appropriately, the analog DHT11 sensor can detect physical changes in temperature and moisture in the air. Due to its small size, Its inexpensive cost, low power consumption, and fast reactions make it one of the top options for a lot of users. The DHT11 sensor is used for testing and assessing consumer items and equipment related to HVAC stands for heating, ventilation, and air conditioning. It is also applicable. to construct a humidity regulator or weather station. The DHT11 sensor's application in home appliances, medicine, and numerous other fields has demonstrated its value in monitoring and regulating temperature and humidity. [18].

Figure 2 shows the DHT11 temperature and humidity sensor. The sensor DHT11 has the following performance range and accuracy presented in table 1.



Figure 2. DHT11 Temperature and Humidity Sensor.

Table 1: Set-up Parameters.

Measurement	Value
Temperature:	0 - 50°C
Humidity:	20 - 90 % RH
Accuracy:	
Temperature:	±2 %
Humidity:	±5 %
voltage range	(3-5.5) volts

The electrical resistance between Electrodes is measured by the sensor DHT11 to identify dampness in the atmosphere. It is built from a substrate that retains dampness. Ionization occurs when moisture is absorbed by the substrate, increasing the conductivity between the electrodes. Appendix A.2 provides a complete datasheet for DHT11; it is correlated with the shift in resistance between electrodes as a result of absorbing moisture.

C. Liquid Crystal Digital (LCD)

In figure 3, displays the systems block diagram. The DHT11 sensor was utilized to measure the temperature in the surrounding air and monitor the ambient temperature. An Arduino Uno microcontroller was utilized to operate the environmental temperature monitoring system. The sensor would produce results that would be shown on the 16x2 i2c LCD.



Figure 3. Block diagram of the system.

The most common type of LCD used in all electronic research to show the process status is 16 x 2. It is inexpensive., Easily configurable, capable of displaying a big character set, and appropriate to every microprocessor and microcontroller. It

features two horizontal lines with sixteen spaces for characters to be shown. [19]. It has two built-in resistors, namely, command and data resistor as shown in Figure 4.

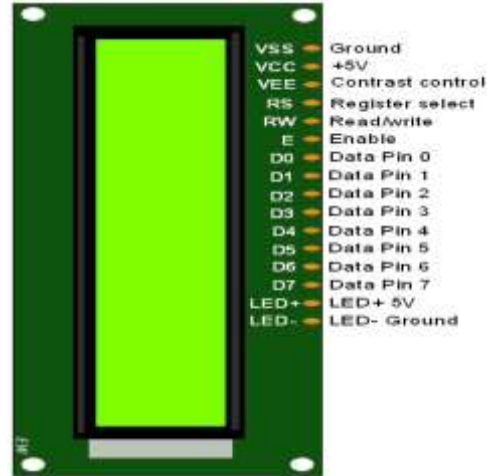


Figure 4. LCD 16x2 Liquid Crystal Display.

D. Integrated development environment (IDE)

The software development environment or program for Arduino is called the Arduino IDE (integrated development environment). where people can test and write various computer programs. The Arduino understands C and C++, so the user can create code in both languages using the IDE. What and how the system functions are determined by the program (codes) uploaded into the Arduino microcontroller using the Arduino IDE. Before transferring the typed code There is a "built-in code parser" in the Arduino IDE that verifies its genuineness. Following confirming that the codes are valid, the compilation and translation work is completed in the IDE. The Arduino microcontroller receives the program from the IDE once the code has been translated. [15].

The collection of several programs that are prepared for device testing is included in the IDE software The Arduino IDE can be expanded through the use of libraries, just as other programming platforms; installing the IDE involves installing several libraries.[18]. The Arduino code-writing software page appears as displayed in Figure 5. The "loop ()" and "setup ()" routines are its two primary functions. The codes for the program should be written in the setup and loop sections, respectively, to ensure that it runs and repeats itself until the reset or power off button is pressed It enables users to modify and program Arduino to do any function they choose. The Arduino board can communicate with the IDE via USB, depending on the features of each board. [13]. The Arduino IDE screen grab is displayed in the accompanying figure.

SETUP/CONNECTION

It is possible to program Arduino to display measurement data on an LCD as well as a serial display. The wiring and

connections are the most important parts for the proper operation of electronics. The list of materials required for this research is as follows:

- an Arduino board (Arduino Uno)
- a breadboard • a sensor (DHT11)
- a 220 Ω resistor

```

1. #include "LiquidCrystal.h"
2. LiquidCrystal lcd(3, 5, 8, 9, 10, 11);
3. #include "DHT.h"
4. DHT dht (13, DHT11);
5. int temp; int humidity;
6. void setup() {
7. // put your setup code here, to run once:
8. dht.begin();
9. lcd.begin (16,2);
10. }
11. void loop () {
12. // put your main code here, to run repeatedly:
13. lcd.setCursor (0,0);
14. lcd.setCursor (0,1);
15. lcd.print ("4. TH STAGE");
16. delay ( 3000); lcd.clear();
17. lcd.setCursor (0,0);
18. lcd.print ("AL MAAREF");
19. lcd.setCursor (0,1);
20. lcd.print ("COLLEGE");
21. delay (5000);
22. lcd.clear();
23. temp = dht.readTemperature
24. void loop() {
25. // put your main code here, to run repeatedly:
26. lcd.setCursor (0,0);
27. lcd.print ("UDAY SLAEBY");
28. lcd.setCursor (0,1);
29. lcd.print ("4. TH STAGE");
30. delay (3000);
31. lcd.clear());
32. lcd.setCursor (0,0);
33. lcd.print ("AL MAAREF");
34. lcd.setCursor (0,1);
35. lcd.print (" COLLEGE");
36. delay (5000); lcd.clear();
37. temp = dht.readTemperature;
38. humidity = dht.readHumidity;
39. lcd.setCursor (0,0);
40. lcd.print ("Temperature:");
41. lcd-print (temp);
42. lcd-print ("c" );
43. lcd.setCursor (0,1);
44. lcd.print ("Humidity:");
45. lcd-print (humidity);
46. lcd.print (" %");
47. delay (8000) ;
48. lcd.clear ();

```

potentiometer

- jumper wires
- LCD Display

The required wiring and pins to connect the LCD and the sensor with Arduino is given below [14]:

- Arduino GND Breadboard -ve power rail
- Arduino 5v Breadboard +ve power rail
- DHT11 -ve pin Breadboard -ve power rail
- DHT11 +ve pin Breadboard +ve power rail
- DHT11 S Arduino Analog pin A0
- LCD 1 Breadboard -ve power rail
- LCD 2 Breadboard +ve power rail
- LCD 3 Potentiometer centre pin
- LCD 4 Arduino Digital pin 12
- LCD 5 Breadboard -ve power rail
- LCD 6 Arduino Digital 11
- LCD 11 Arduino Digital Pin 5
- LCD 12 Arduino Digital Pin 4
- LCD 13 Arduino Digital Pin 3
- LCD 14 Arduino Digital Pin 2
- LCD 15 220 Ω (ohm) resistor and the other pole of 220 Ω resistor to Breadboard positive power rail
- LCD 16 Breadboard -ve power rail
- Potentiometer +ve Breadboard negative power rail.

In a schematic representation, the connections between the LCD, DHT11 sensor, and Arduino Uno are illustrated in Figure 5[14].

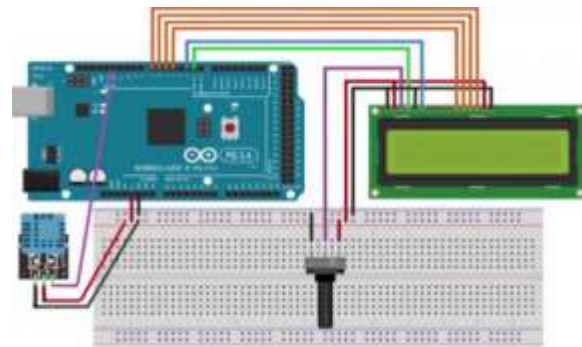


Figure 5. Arduino interfacing with Liquid crystal and DHT11 sensor.

Once all wiring and connections have been completed, The IDE is where the code should be written, and the code typed there indicates the Arduino to operate for the LCD to read the measurement that was taken from the sensor. In table 2, the required codes for showing temperature and humidity readings on an LCD panel are shown below, which, when the program runs in the IDE, displays the codes.

• a

Table 2: Pseudocode the program runs in the IDE.

Following writing, the IDE should verify the codes listed above, and once the verification is finished The Arduino program is prepared for uploading. The reading is displayed

on the LCD while the software is operating, as shown in Figure 6.



Figure 6. Humidity and Temperature.

IV. RESULT AND ANALYSIS

The initial deployment of the programmed gadget, upon the completion of wiring and coding, yielded promising results as it effectively measured temperature and humidity. Figures 9 and 10, depicting readings from the serial monitor and LCD, serve as visual representations of the gadget's immediate success. These initial outcomes lay the foundation for a comprehensive discussion on the performance attributes of the Arduino-based temperature and humidity sensor. To delve into the sensor's capabilities, two distinct testing setups were employed, providing a holistic assessment of its functionality. In figure 7, conducted within the controlled environment of the Anbar Health Department's medical oxygen laboratory, offered insights into the sensor's performance under stable and controlled conditions. In figure 8, carried out outdoors, introduced dynamic external factors to evaluate the sensor's adaptability in real-world scenarios. To induce variability in temperature and humidity for the outdoor test, the sensor was strategically placed within an insulated wooden box. This setup aimed to simulate fluctuating environmental conditions and provided a controlled yet dynamic scenario for evaluation. The measurements were captured outside the building, further enhancing the complexity of the test.

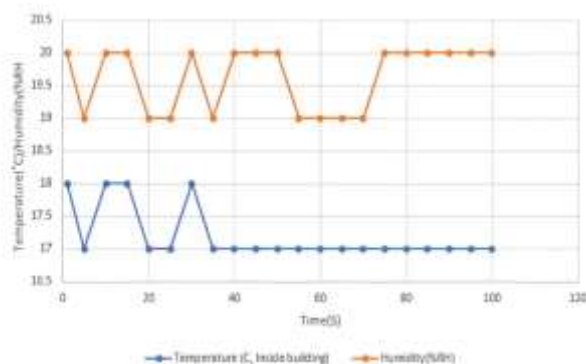


Figure 7. Analysis of measurement data.

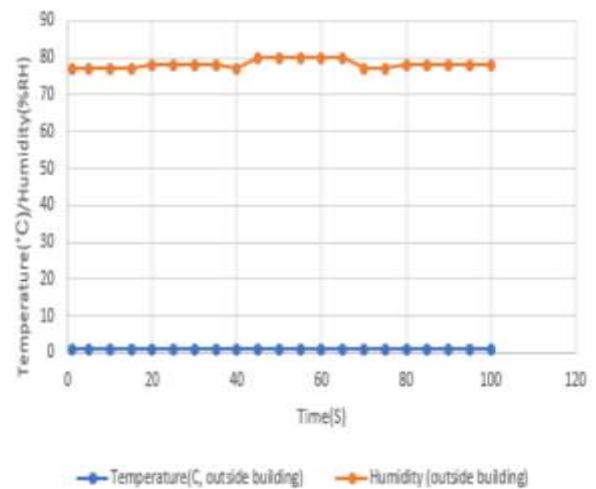


Figure 8. Sensor measurements of temperature and humidity.

The analysis of measurement data was conducted using Excel, allowing for a meticulous examination of temperature and humidity trends over time. The resulting graphs present a visual representation of the sensor's response to varying conditions, facilitating a nuanced understanding of its performance dynamics. Upon reviewing the graphical representations, several notable observations emerge. In the laboratory setting, the sensor demonstrated consistent and precise measurements, showcasing its reliability under controlled conditions. The outdoor test, with the introduction of external variables, elucidated the sensor's adaptability, capturing the nuances of temperature and humidity fluctuations. The use of an insulated wooden box as a variable inducer in the outdoor scenario further accentuated the system's robustness in responding to changing environmental factors. These results underscore the effectiveness of the Arduino-based monitoring system in measuring temperature and humidity across diverse settings. The successful deployment in both controlled and dynamic environments attest to the versatility and reliability of the system. The insights gained from this discussion pave the way for potential refinements and optimizations in future iterations, ensuring continued advancements in environmental monitoring through Arduino-based technologies.

IV. CONCLUSION

In conclusion, the construction of the monitoring gadget, designed as a thermometer to measure temperature and humidity within a structure, represents a noteworthy achievement in the realm of Arduino-based environmental monitoring systems. The system not only effectively gauges external temperature and humidity but also achieves commendable energy efficiency, offering a cost-effective

alternative to pricier sensors. Despite encountering minor challenges such as measurement inaccuracies and sensor constraints, the overall performance of the system remains reliable and precise, underscoring the potential of Arduino-based solutions in practical applications. This paper not only highlights the current capabilities of Arduino Uno technology but also acknowledges the dynamic nature of technological progress, with more advanced programs and boards available to enthusiasts. Beyond its technical aspects, engaging in this paper has proven to be both enjoyable and highly educational, providing valuable insights into the real-world applications of Arduino microcontrollers, basic electronics principles, and the C programming language. The hands-on experience gained serves as a foundational step in understanding the vast potential of microcontrollers. It contributes to the continuous improvement of research methodologies and educational practices within the field. In future work, this domain could focus on addressing the observed measurement inaccuracies and sensor constraints, refining the system to enhance its precision and reliability. Additionally, exploring the integration of more advanced Arduino boards and programs, as mentioned, could open new possibilities for expanding the functionalities of the monitoring gadget. Further research could delve into the development of user-friendly interfaces and applications to facilitate seamless interaction with the monitoring system. Investigating the potential integration of emerging technologies, such as Internet of Things (IoT) connectivity, could pave the way for a more interconnected and responsive environmental monitoring network.

REFERENCES

- [1] Bogati, S., Sharma, M., & Shrestha, R. (2021). Designing of the Wireless Digital Portable Temperature and Humidity Monitoring Device and Analyzing Data at Different Part of Kathmandu Valley. *Semiconductor Science and Information Devices*, 3(1), 41-48.
- [2] Banzhi, M., n.d. How Arduino is open-sourcing imagination. s.l.: TED Talk.
- [3] Dangi, N. (2017). Monitoring environmental parameters: humidity and temperature using Arduino based microcontroller and sensors. *Helsinki Metropolia University of Applied Sciences-2017 Disponible en: https://www.theseus.fi/bitstream/handle/10024/142235/Dangi_Nagendra.pdf*.
- [4] Vleeschouwer, K. D., Loey, Van, A. & hendrickx, M. E., 2017. The Effect of High Pressure-High Temperature Processing Conditions on Acrylamide Formation and Other maillard Reaction Compounds. *Journal of Agricultural and food chemistry*, Volume 2010, 58(22), pp. 11740-11748.
- [5] Akami, P., Oke, A. & Akpomiemie, O., 2015. Impact of environmental factors on building project. *ScienceDirect*, 11(1), pp. 91-97.
- [6] kushner, D., 2011. The Making of Arduino-IEEE Spectrum. [Online] Available at: <https://spectrum.ieee.org/geek-life/hands-on/the-making-of-arduino> [Accessed 10 11 2017].
- [7] Kuria, K. P., Robinson, O. O., & Gabriel, M. M. (2020). Monitoring temperature and humidity using Arduino Nano and Module-DHT11 sensor with real-time DS3231 data logger and LCD display. *Health Hyg*, 6(7), 8.
- [8] Karra, A., Kondi, B., & Jayaraman, R. (2020). Implementation of wireless communication to transfer temperature and humidity monitoring data using Arduino Uno. In *2020 International Conference on Communication and Signal Processing (ICCSPP)*. IEEE.
- [9] Nasution, I. S., Sundari, S., & Rifky, N. (2020). Data acquisition of multiple sensors in greenhouse using Arduino platform. *IOP Conference Series: Earth and Environmental Science*, 515(1). IOP Publishing.
- [10] Mihai, B. O. G. D. A. N. (2016). How to use the DHT22 sensor for measuring temperature and humidity with the Arduino board. *Acta Universitatis Cibiniensis–Technical Series*, 68, 22-25.
- [11] Park, D.-H., et al. (2011). A study on greenhouse automatic control system based on wireless sensor network. *Wireless Personal Communications*, 56, 117-130.
- [12] Kyrtisoglou, A., et al. (2021). Missing data imputation and meta-analysis on correlation of spatio-temporal weather series data. In *2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. IEEE.
- [13] Sandhu, R., 2016. What is Arduino? A full definition. [Online] Available at: <https://www.lifewire.com/what-is-arduino-2495652> [Accessed 12 11 2017].
- [14] Wilson, J. S., 2005. *Sensor Technology Handbook*. Oxford: Elsevier Inc.
- [15] Blaauw, F. et al., 2016. Let's get Physical-An intuitive and generic method to combine sensor technology with ecological momentary assessments. *ScienceDirect*, Volume 63, pp. 141-149.
- [16] Bogati, S., Sharma, M., & Shrestha, R. (2021). Designing of the Wireless Digital Portable Temperature and Humidity Monitoring Device and Analyzing Data at Different Part of Kathmandu Valley. *Semiconductor Science and Information Devices*, 3(1), 41-48.
- [17] Kouhia, E.-P., 2016. Development of an Arduino based Embedded system, s.l.: s.n.
- [18] Circuitbasics, 2017. How to set up the DHT11 humidity sensor. [Online] Available at: <http://www.circuitbasics.com/how-to-set-up-the-dht11-humidity-sensor-on-an-arduino/> [Accessed 25 10 2017].
- [19] Sandhu, R., 2016. What is Arduino? A full definition. [Online] Available at: <https://www.lifewire.com/what-is-arduino-2495652> [Accessed 12 11 2017].