

Design and Simulation of a Microcontroller Based Weather Station

Sarmeje C. P.¹, Afolabi Olusegun A.¹ and K. G. Gaya²

¹Department of Electrical/Electronic Engineering, Federal Polytechnic P.M.B 35, Mubi, Adamawa State, Nigeria. ²Department of Physics, Adamawa State University, Mubi, Nigeria.

Corresponding author: cysarmeje@yahoo.com

Abstract

The motivation for this work is centered on prevailing shortage of weather-related data in Nigeria. The evidence of this fact could be seen in poor weather forecasting in the country, poor response to weather related natural disasters because of lack of foreknowledge, and unnecessary disruption of flights and flood disasters that destroy lives and properties. This is as a result of the unavailability of a network of weather stations dedicated for national weather service and partly due to the high cost of imported weather stations. This work aims to solve the problem of unavailable weather stations due to its cost and complex technology. Block approach was used in the designing of the system to simply the realization of the objectives. The system was designed to comprise data sensors, microcontroller and display sections. The system was simulated in Proteus 8. Simulation environment. Results from the simulation show that the speed is from between 1.8Km/hr. and 117.3Km/hr. Temperature and humidity range depended on the sensor used, in this case the DHT11.

Keywords: Weather, Windspeed, Arduino NANO Microcontroller, Proteus, DHT11.

Introduction

Weather is the state of the atmosphere of a given place at a particular time (Yates, 1947). In order to describe the atmospheric conditions, certain key weather elements must be known and quantified. Some of those important elements are temperature, relative humidity, atmospheric pressure, wind speed, wind direction, cloud cover, precipitation, etc.

Weather monitoring and forecasting play a crucial role in numerous industries, including agriculture, aviation, energy, and disaster management (Smith & Johnson, 2022). In recent years, the emergence of microcontroller technology has revolutionized the way weather stations are designed and operated, offering advanced capabilities for data collection, processing, and transmission (Chen et al, 2023).

This paper presents the comprehensive design and simulation of a state-of-the-art microcontroller-based weather station, aimed at providing real-time weather data with increased accuracy and reliability. Leveraging the capabilities of microcontrollers, this weather station offers a cost-effective and scalable solution for weather monitoring in both urban and remote areas.

The weather station is equipped with a range of sensors, including temperature sensors for measuring ambient temperature, humidity sensors for monitoring moisture levels, atmospheric pressure sensors for gauging air pressure, wind speed and direction sensors for wind analysis, and rain gauges for precipitation measurement. These sensors collectively gather critical meteorological parameters from the surrounding environment (Rodriguez et al, 2023).

A key advantage of microcontroller technology is its ability to process data locally. The weather station employs powerful microcontrollers to process the collected sensor data and execute advanced algorithms for real-time weather analysis. This ensures that weather information is promptly available, facilitating quicker response times to changing weather conditions and potential weather-related hazards (Li et al, 2022).

Moreover, the microcontroller-based weather station is designed to be energy-efficient, extending the operational life of the station when powered by batteries or renewable energy sources. This feature makes the station particularly suitable for remote and off-grid installations, where access to continuous power may be limited.

To validate the performance and reliability of the proposed weather station, extensive simulations have been conducted. These simulations involve subjecting the station to various weather scenarios, including rapid temperature changes and varying wind speeds). The results obtained from the simulations provide critical insights into the station's ability to deliver accurate and reliable weather data.

Weather stations also provide data archives over a long period of time (Guo & Song 2010). Such archives could be used by the academia or research institutions. The motivation for this work is centered on prevailing shortage of weather-related data in Nigeria. The evidence of this fact could be seen in poor weather forecasting in the country, poor response to weather related natural disasters because of lack of foreknowledge, and unnecessary disruption of flights and flood disasters that destroy lives and properties. This is as a result of the unavailability of a network of weather stations dedicated for national weather service and partly due to the high cost of imported weather stations. Also, operating and managing imported weather stations requires much technical expertise making their use difficult for private users like small-scale agriculturists, industrialists and schools. This work would make weather related data readily available to small-scale farmers, institutions and others that may need it without huge financial implications. Simplicity of operation was also factored into the design and as such a very high technical know-how is not necessary to operate the system.

Problem Statement

Weather monitoring and forecasting play a crucial role in numerous industries, including agriculture, aviation, energy, and disaster management (Smith & Johnson, 2022). In recent years, the emergence of microcontroller technology has revolutionized the way weather stations are designed and operated, offering advanced capabilities for data collection,

processing, and transmission. This paper presents the comprehensive design and simulation of a state-of-the-art microcontroller-based weather station, aimed at providing real-time weather data with increased accuracy and reliability. Leveraging the capabilities of microcontrollers, this weather station offers a cost-effective and scalable solution for weather monitoring in both urban and remote areas.

Methodology

The method adopted would be by first converting wind speed/direction detected by a mechanical wind vane, decoded by magnetic sensors, and converted to digital signal by analog to digital converters in an Arduino NANO microcontroller. Temperature and humidity sensors will be used to acquire data for processing by the microcontroller on an LCD screen.

The block diagram of the mini digital weather station is shown in figure 1.0. It consists of the power supply unit, the sensors, microcontroller unit and the display unit.

The power supply section serves to provide power to the circuit as well as the sensors. For this system the power supply is a battery to make it less dependent on mains electricity.

The sensors act as the input to the system, while the microcontroller is the heart of the system serving to receive the three sensor readings and convert to appropriate units for display on the 16 x 2 LCD display which is the output of the system.

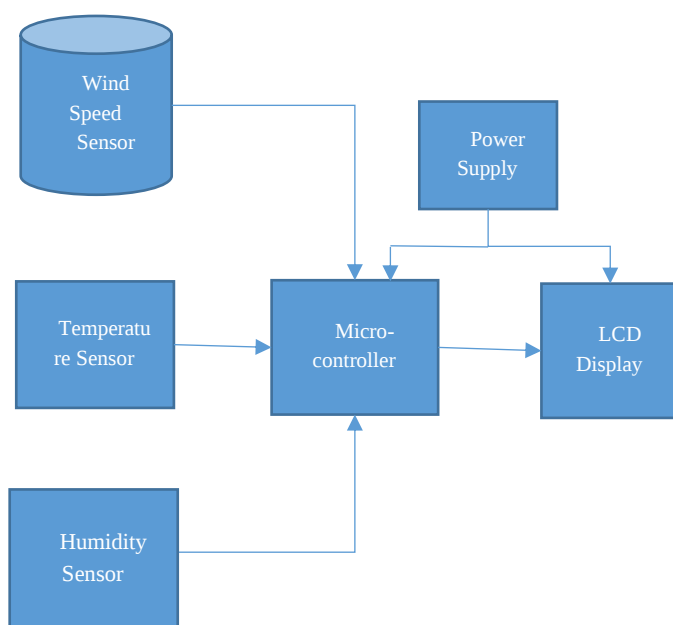


Figure 1: Block diagram of the system

Method of Design

The individual blocks that make up the system are carefully designed step by step from first principles. The system is made of two sections hardware and software sections.

Hardware Design

Wind Speed Sensor Design

Wind speed sensing is done by counting the pulses produced by a type CYL3503 Hall effect sensor when a rotating magnet is made to pass the sensor at a rate proportional to the speed of the wind. CYL3503 linear Hall-effect integrated circuit includes a voltage regulator, Hall-voltage generator, linear amplifier, and emitter-follower output stage. The output of the ICs changes linearly with the magnetic flux density of the input. This sensor was chosen because of the following qualities Small Size, High Accuracy, High Sensitivity, Excellent Reliability and High Linearity (Chentang Tech, 2023).

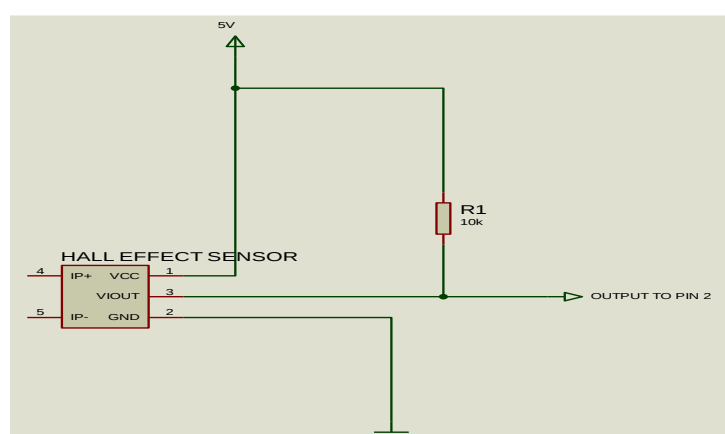


Figure 2: The wind speed sensor connection

Output from the wind speed sensor is sent to a 4047 digital IC to serve as an inverter and buffer, this is to ensure that pure digital signals get to the microcontroller. Figure 2.0 show the wind speed sensor connection.

Temperature and Humidity Sensor Selection

The DHT11 is a commonly used Temperature and humidity sensor that comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers.

The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of $\pm 1^\circ\text{C}$ and $\pm 1\%$. So, if you are looking to measure in this range then this sensor might be the right choice for you. The DHT11 Sensor is factory calibrated and outputs serial data and hence it is highly easy to set it up. The connection diagram for this sensor is shown in figure 3 below (Mouser Electronics, 2023).

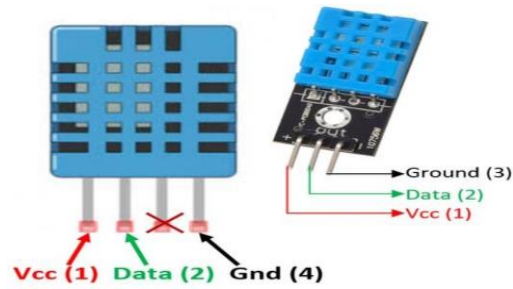


Figure 3: The DH11 Sensor (Mouser Electronics, 2023)

The sensor has the following characteristics (Mouser Electronics, 2023).

- Operating Voltage: 3.5V to 5.5V
- Operating current: 0.3mA (measuring) 60uA (standby)
- Output: Serial data
- Temperature Range: 0°C to 50°C
- Humidity Range: 20% to 90%
- Resolution: Temperature and Humidity both are 16-bit
- Accuracy: $\pm 1^\circ\text{C}$ and $\pm 1\%$

The Microcontroller Selection

The main component in this design as shown in figure 4 is the microcontroller with the coordination and control function. The microcontroller chosen for the work is the Arduino Nano. Arduino Nano is a small, complete, and breadboard-friendly board based on the ATmega328 (Arduino, 2023).

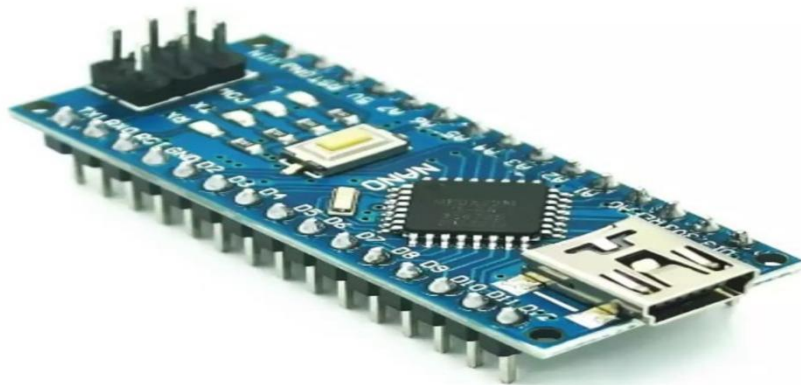


Figure 4: The Arduino Nano (<https://www.konga.com/product/arduino-nano>)

The Display Selection

LCD modules are very commonly used in most embedded projects, the reason being its cheap price, availability and programmer friendly. The appearance and the pinouts are as

shown in figure 5. (Shenzhen Hi-Tech Co (2023). The 16×2 LCD is named so because; it has 16 Columns and 2 Rows.



Figure 5: The LCD display connections (<https://www.konga.com/product/16x2-lcd>)

The Power Supply

The entire system is powered from a 9.0V battery. The battery is chosen to last long and to deliver the required current. For this work a PP3 type 9.0V battery is chosen. The nine-volt battery, or 9-volt battery, is an electric battery that supplies a nominal voltage of 9 volts. Actual voltage measures 7.2 to 9.6 volts, depending on battery chemistry.

Complete Circuit Diagram

The complete circuit diagram is as shown in figure 6.o. The parts of the circuit are as listed in table I

Table I: Part List

S/N	Component	Label	Type	Quantity
1	Microcontroller	UC	Arduino Nano	1
2	Fixed resistor	R1 – R3	10K	3
3	Temp & Humidity Sensor	S1	DH11	1
4	Infra-red Pair	IR1	TCST1103	1
5	LCD Display	LCD	16 x 2	1
6	Hall Sensor	H1	CYL3503	1
8	Switch	SW1	Single pole	1
9	Battery	9V	PP3	1
10	Fixed resistor	R4	1kΩ	1

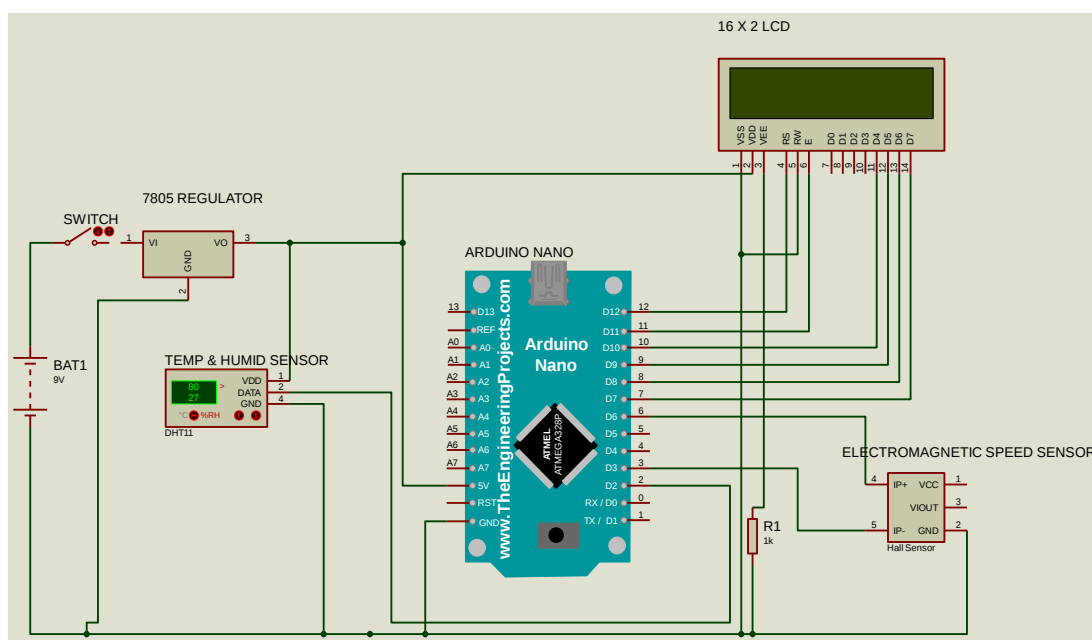


Figure 6: The Complete Circuit Diagram (field survey)

Software Design

The microcontroller needs to be programmed for it to perform the repeated tasks assigned to it. The software is written in C++ language for the Arduino IDE.

The software is designed following the flow chart as shown in figure 7 made from its intended mode of operation

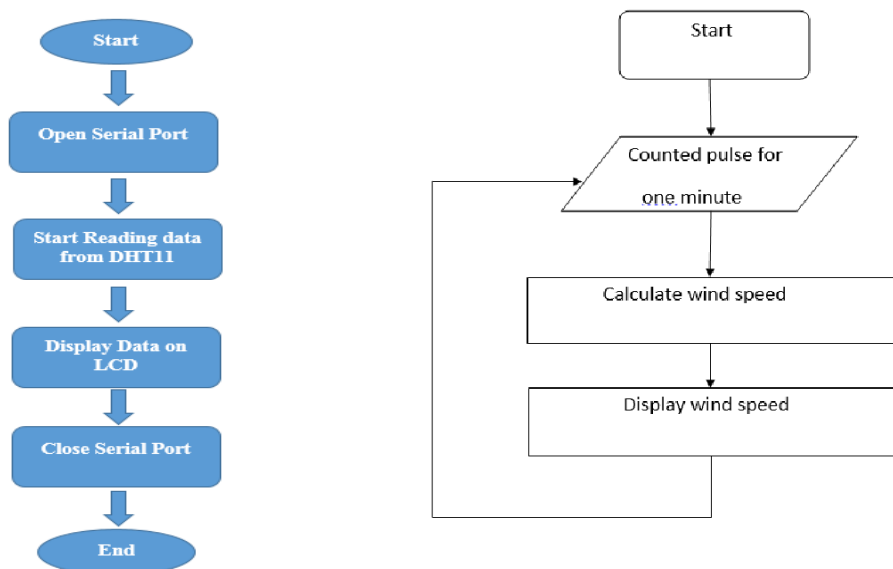


Figure 7: The flow Chart of the Software Design

The Arduino Nano can be programmed with the Arduino software (IDE 1.8.5). The ATmega328 on the Arduino Nano comes pre burned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. (Arduino, 2023).

Simulation

In this design, the Arduino Nano was simulated in the Proteus environment using Arduino virtual library for Proteus. To achieve the simulation programs for the Arduino has to be written in the C programming language. These programs were written using the Arduino IDE. After writing and compiling the codes, the hex file was copied to the virtual Arduino library in Proteus. Figure 8.o shows the Arduino IDE.

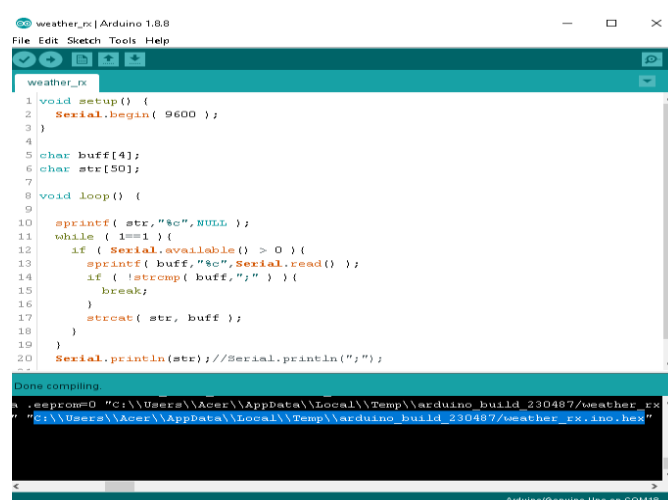


Figure 8: Arduino IDE showing codes (field survey)

Proteus was used in this work to, not only to design the circuit diagram, but to also simulate the entire system.

The designed circuit was drawn in Proteus 8.0 software in windows 10 platform. All component and parameters were carefully chosen from the database. Simulation was done under five different settings for the wind speed/direction sensors.

- W1 Wind speed at 20%
- W2 wind speed at 50%
- W3 Wind speed at mid-75%
- T1 Temperature 20 °C
- T2 Temperature 25 °C
- T3 Temperature 30 °C
- H1 Humidity at 30 °C
- H2 Humidity at 50 °C
- H3 Humidity at 80 °C

Simulation starts when the following sequence of instructions are followed:

- i. Proteus is started
- ii. Arduino IDE is started
- iii. The codes are processed
- iv. The Hex file directory exported from IDE to Proteus
- v. Simulation started

With simulation running, the sensors which are represented by variable resistors are at default positions. The positions can be altered using the mouse.

Results and Discussion

Wind Speed Simulation Result

For the first condition(W₁) where wind speed is at 20%, the setting and result obtained is shown in figure 9.0 and figure 10.0 respectively.



Figure 9: Simulation showing wind speed condition W₁ setting

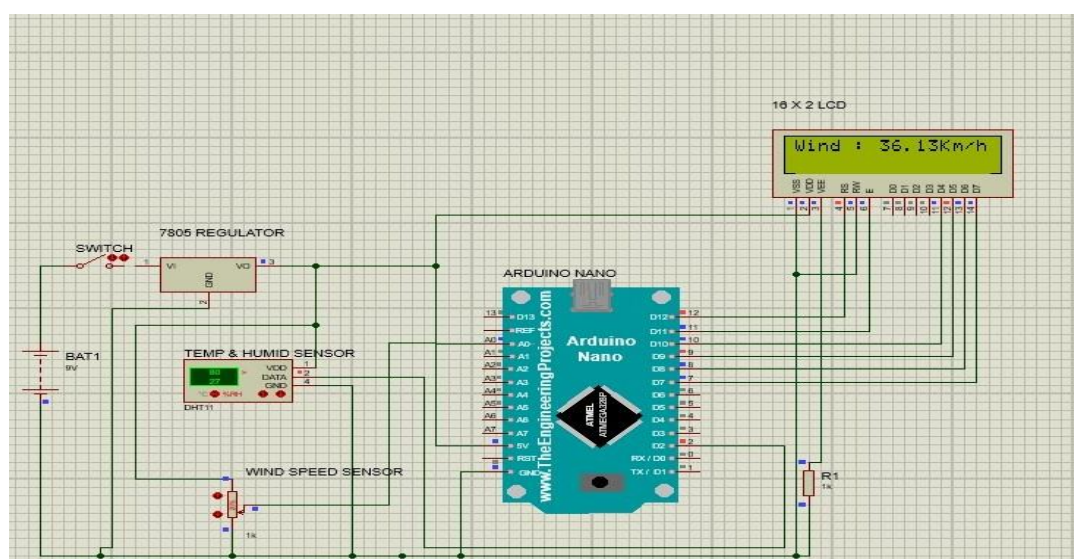


Figure 10: Simulation showing result for wind speed condition W₁

For second condition (W₂) where wind speed is at 50%, the setting and result obtained is shown in figure 11.0 and figure 12.0 respectively.

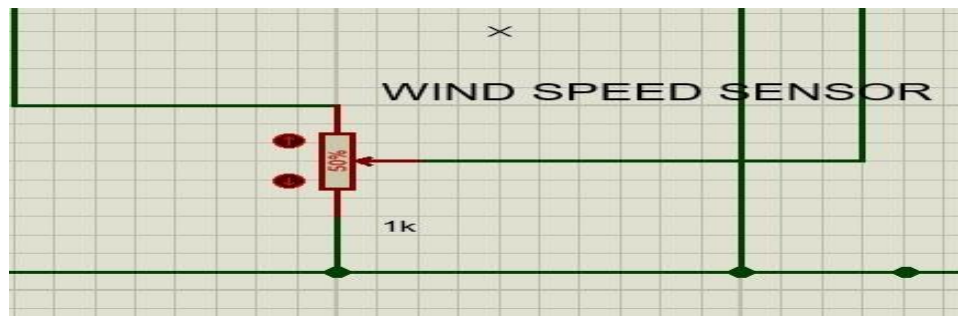


Figure 11: Simulation showing wind speed condition W₂ setting

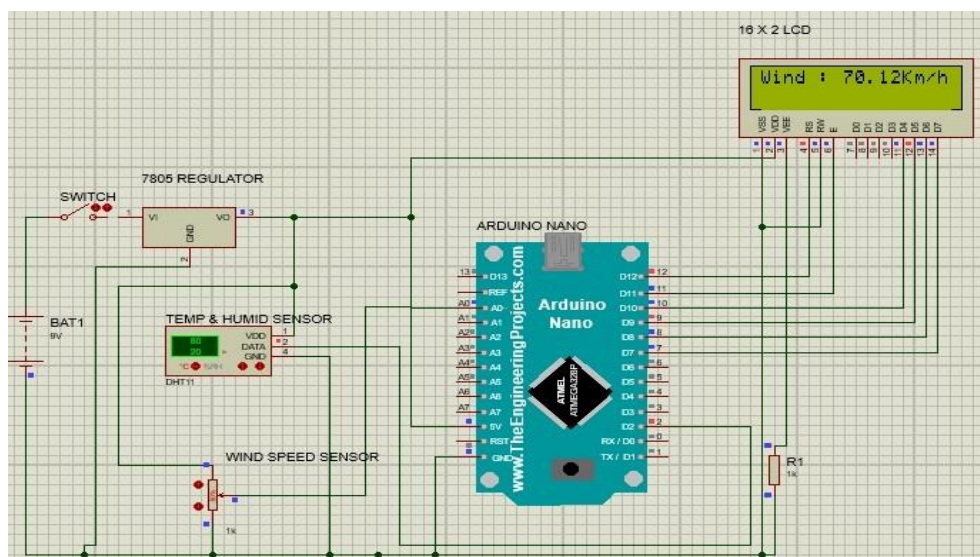


Figure 12: Simulation showing result for wind speed condition W₂



Figure 13: Simulation showing wind speed condition W₃ setting

For the third condition (W₃) where wind speed is at 75% the setting and result obtained is shown in figure 13.0 and figure 14.0 respectively.

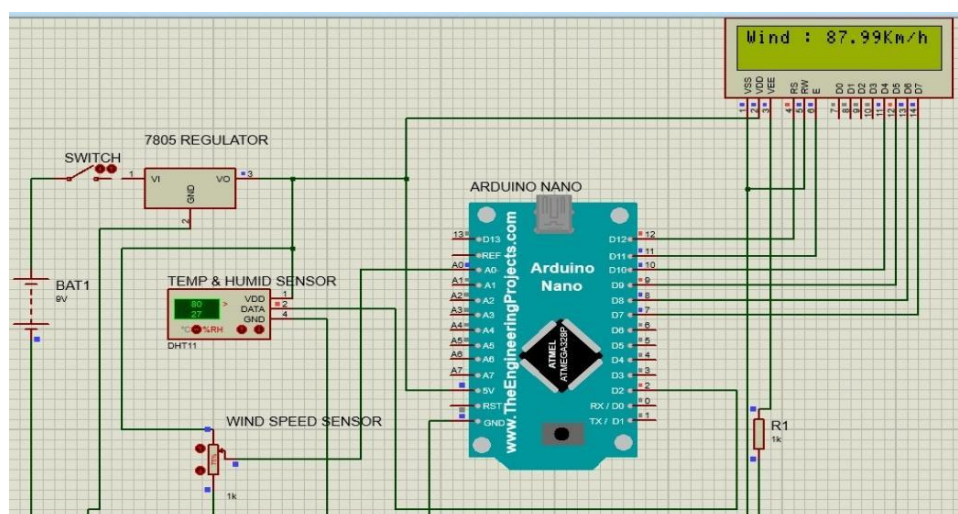


Figure 14: Simulation showing result for wind speed condition W_3

Temperature Simulation Result

For the first temperature condition, (T_1) where simulation setting is at 20% the setting and result obtained is shown in figure 15.0 and figure 16.0 respectively.

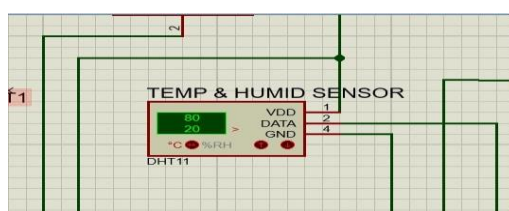


Figure 15.0: Simulation showing temperature condition T_1 setting

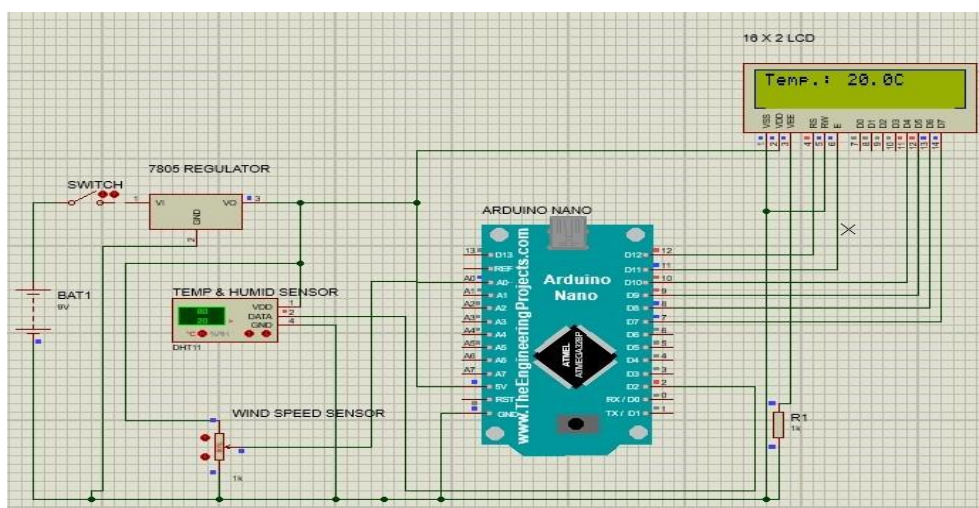


Figure 16: Simulation result for temperature condition T_1

For the second temperature condition (T_2) where simulation setting is at 25% the setting and result obtained is shown in figure 17.0 and figure 18.0 respectively.

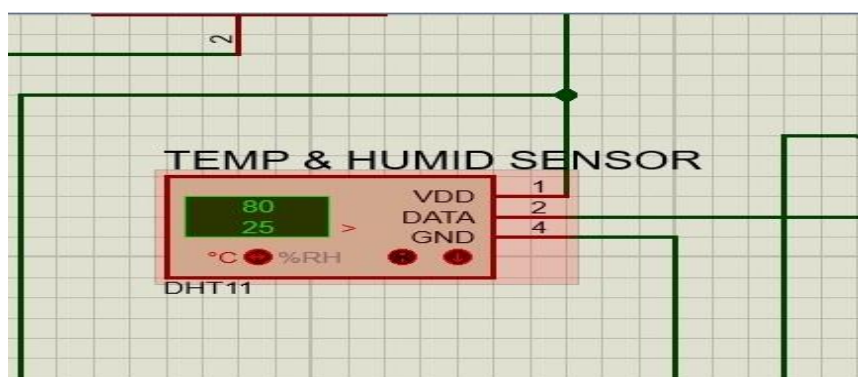


Figure 17: Simulation showing temperature condition T_2 setting

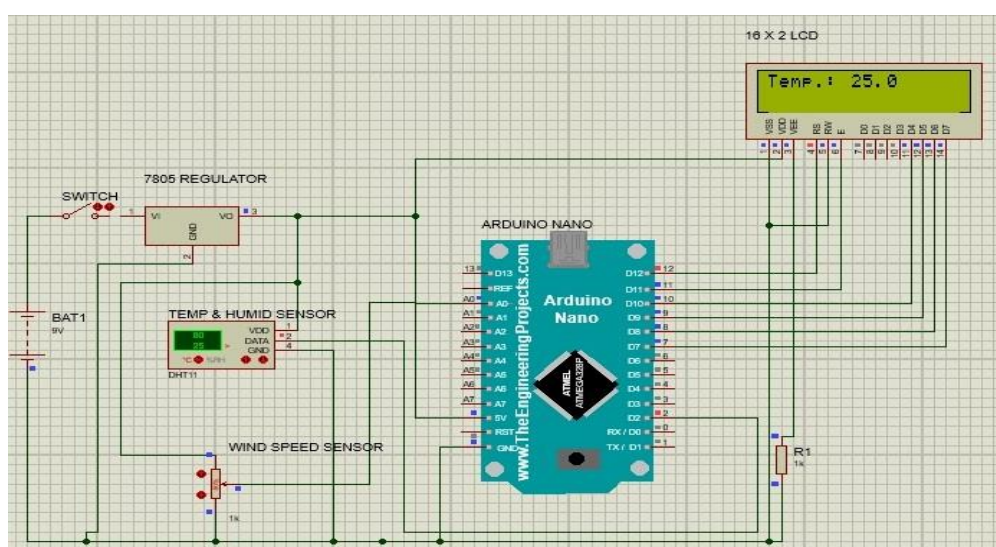


Figure 18: Simulation result for temperature condition T_2

For the third temperature condition (T_3) where the setting is at 30%, the setting and result obtained is shown in figure 19.0 and figure 20.0 respectively.

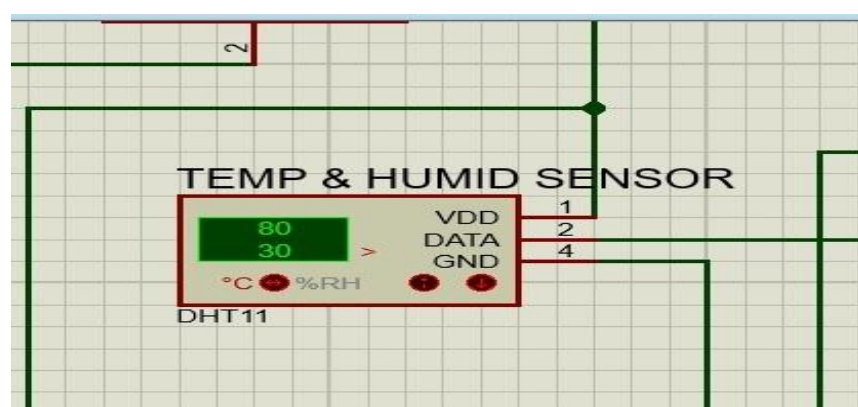


Figure 19: Simulation showing temperature condition T_3 setting

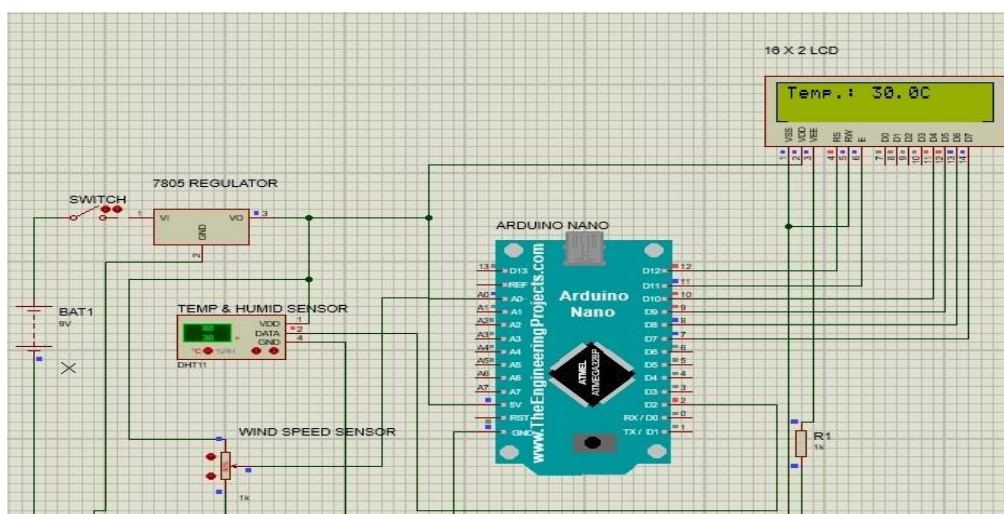


Figure 20: Simulation result for temperature condition T2

Humidity Simulation Result

For the first condition (H1) where humidity simulation setting is at 30%, the setting and result obtained is shown in figure 21.0 and figure 22.0.

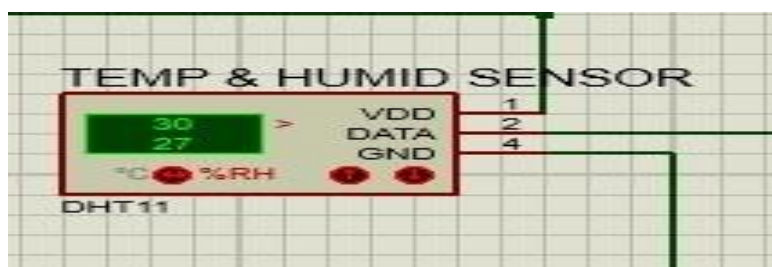


Figure 21: Simulation showing humidity condition H1 setting

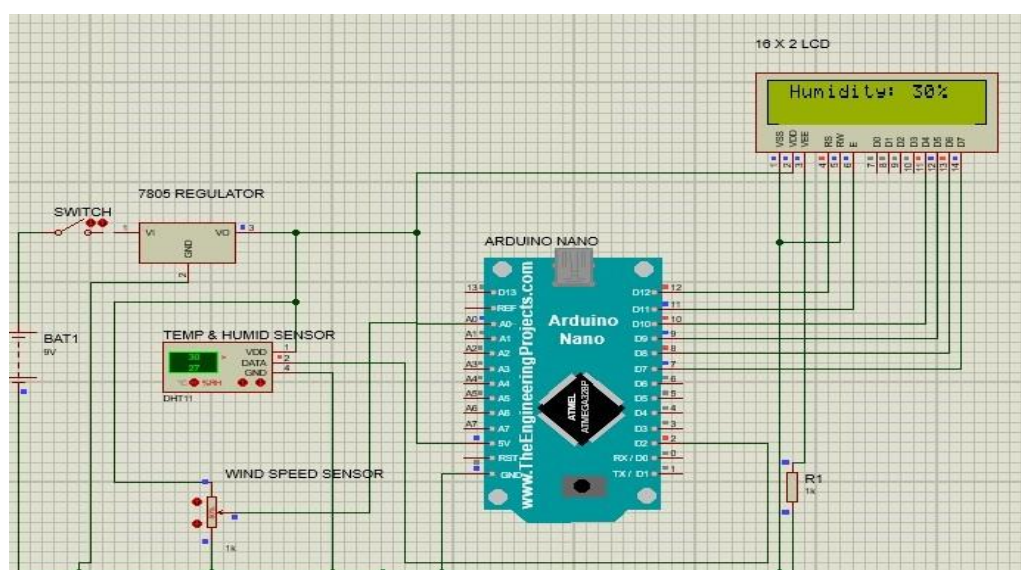


Figure 22: Simulation result for Humidity condition H1

For the second humidity condition (H2) where humidity simulation setting is at 50%, the setting and result obtained is shown in figure 23.0 and figure 24.0.

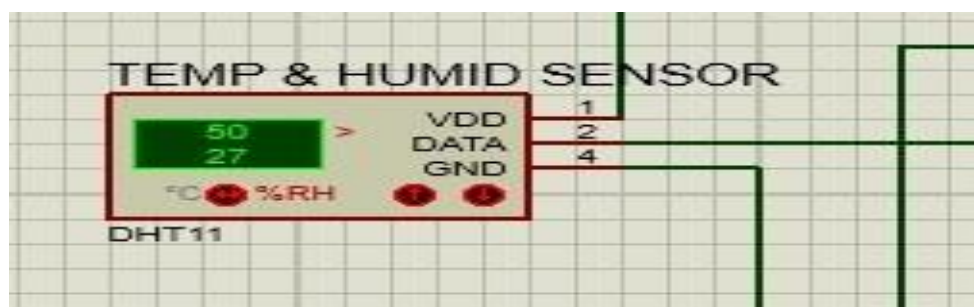


Figure 23: Simulation showing humidity condition H2 setting

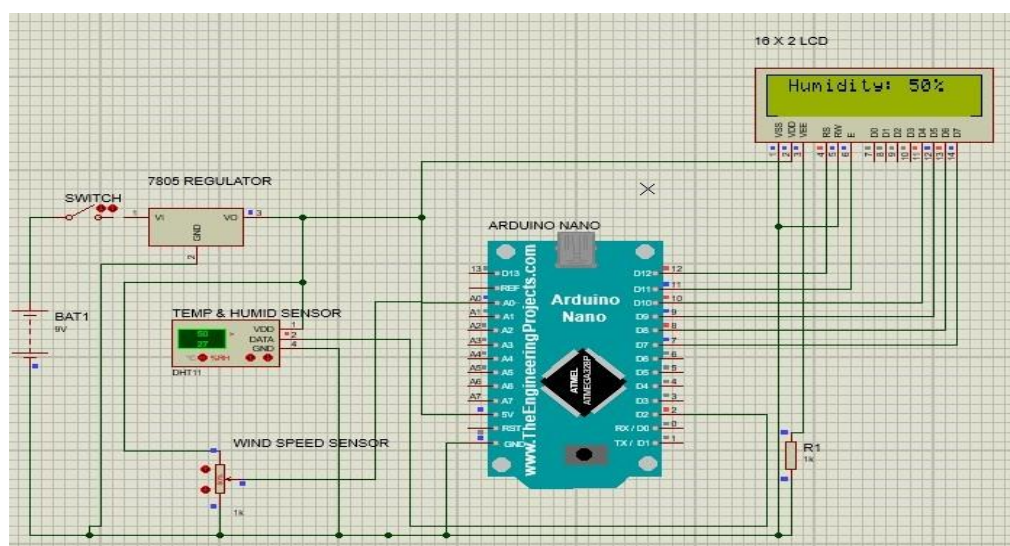


Figure 24: Simulation result for humidity condition H1

For the third humidity condition (H3) where humidity simulation setting is at 80%, the setting and result obtained is shown in figure 25.0 and figure 26.0.

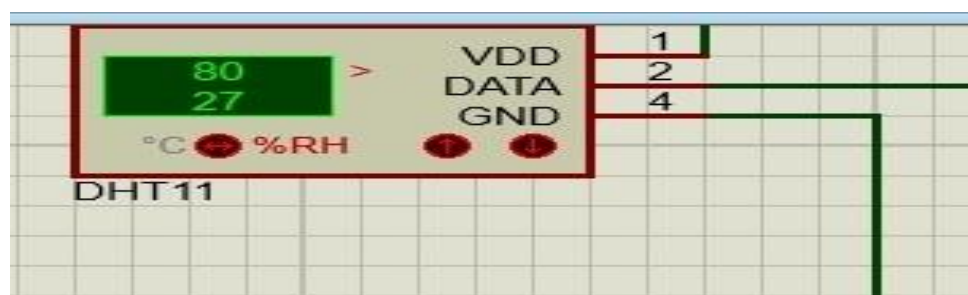


Figure 25: Simulation showing humidity condition H3 setting

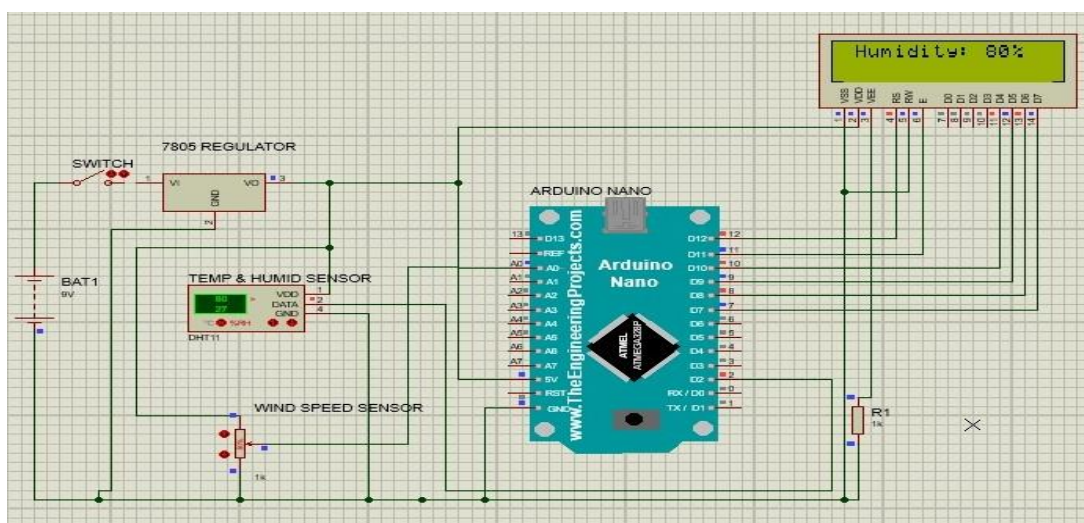


Figure 26: Simulation result for Humidity condition H₃

Summary of Simulation Results

Table II: Summary of simulation result

S/N	Condition	Simulation Setting	Displayed Output Result
1	W ₁	20%	36.15Km/hr
2	W ₂	50%	70.12 Km/hr
3	W ₃	75%	87.99 Km/hr
4	T ₁	20 °C	20°C
5	T ₂	25 °C	25°C
6	T ₂	30 °C	30°C
7	H ₁	30%	30%
8	H ₂	50%	50%
9	H ₃	80%	80%

Result Discussion

From the wind speed simulation result obtained from figure 9.0 to figure 14.0 the readings of wind speed and the displayed direction tally with the varying sensor positions. Though not accurate because the resistor is not varied accurately with the mouse. That is bringing the resistance to 0% does not necessarily mean the resistance the microcontroller is seeing under simulation is 0. These are summarized in table II, from the summary, it can be deduced that the range of speed is range between 1.8Km/hr and 117.3Km/hr. Temperature and humidity ranges given in figure 16.0 to figure 26.0 are dependent on the range of the DHT11 sensor. The temperature and moisture levels were almost the same because the sensors used for detection are standard and calibrated for use with microcontroller.

Conclusion

This work aims to solve the problem of unavailable weather stations due to its cost and complex technology. Block approach was used in the designing of the system to simply the realization of the objectives. The system was designed to comprise data sensors, microcontroller, and display sections. The system was simulated in Proteus 8. Simulation environment. Results from the simulation show that the speed is range between 1.8Km/hr and 117.3Km/hr. Temperature and humidity range depended on the sensor used, in this case the DHT11.

Recommendations

Based on the design presented in this paper, the following are recommended for future work:

- Wind direction detecting should be included in the design.
- Wireless data streaming to a monitoring station or via the web should be included.
- Considering the global economic meltdown, Nigeria and other third world nations should take up the challenge of sponsoring indigenous designs of systems that have potential of promoting agriculture and mitigating against natural disasters like flood and famine.

References

- Arduino Nano (2023). Arduino Nano. Retrieved from (<https://store.arduino.cc/products/arduino-nanoURL>)
- Chen, C., Lee, D., Wang, S. (2023). *Design and Implementation of a Low-Power Weather Station for Remote Areas. * IEEE Transactions on Instrumentation and Measurement, 68(5), 1718-1726.
- Chenyang Technologies (2023). CYL3503 Datasheet. Retrieved from <https://www.chenyang-gmbh.com/en/cat/index/sCategory/25>
- CYL Corporation. (2023). CYL Hall Sensor [Datasheet]. Retrieved from [<https://www.example.com/datasheet>]
- Guo, X.; and Song, Y. 2010. Design of automatic weather station based on GSM module. Proc. Int. Conf. on Computer, Mechatronics, Control and Electronic Engineering (CMCE), Changchun, China, 24-26 August 2010, Vol. 5, pp. 80-2.
- Jackson, R.M. 1993. Weather station. Electronics Now 64(10): 31-8.
- Li, J., Zhang, R., Kim, S. (2022). *Real-Time Weather Analysis using Microcontrollers and Machine Learning. International Conference on Embedded Systems and Applications (ESA), Proceedings, 211-218.
- Mouser Electronics, (2023). DHT11 Datasheet. Retrieved from <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>
- Rodriguez, M., Garcia, L., Gonzalez, P. (2023). *Wireless Communication in Weather Monitoring: A Review. * Sensors, 23(2), 345-362.
- Shenzhen Hi-Tech Co (2023). 16 x 2 LCD Display (Blue). Retrieved from (<https://www.made-in-china.com/manufacturers/lcd-1602.html>)
- Smith, J., Johnson, A. (2022). *Advancements in Microcontroller-Based Weather Monitoring Systems. Journal of Meteorological Technology, 48(2), 183-198.

Yates, R.F. 1947. The Weather for a Hobby: A Guide to the Construction and Use of Weather Instruments Intended for Amateurs. Dodd, Mead & Co., New York, NY, USA.