ECS 327 LAB REPORT 2021

Sensor Integration and Analysis for Smart Applications

Rita Abani

19244

This report is presented for the Lab Course ECS 327 that was carried out under the esteemed guidance of Dr Mitradip Bhattacharjee, Department of Electrical Engineering and Computer Science, IISER Bhopal

Sensor Integration and Analysis for Smart Applications

Abstract

Industrial progress is quintessential for the economic development of any country. India especially, being a developing country values industries as its economic barometer. Associated with Industrial acumen are factors like technical infrastructure, scientific task forces etc. Such set-ups make it indispensable to monitor the ambience quantified by temperature and humidity in order to ensure smooth functioning. Improper monitoring of these parameters have led to several tragedies in the past like minor explosions in factory establishments or wastage of factory produce, transportation issues etc. In this project, we explore the use of sensor integration using a DHT11 sensor and Arduino microcontroller board to determine the temperature and relative humidity of a given setting.

Keywords

a) DHT11: Temperature and Humidity sensor
b) Breadboard: Board for making experimental model of electric circuits
c) Arduino: Micro-controller Board

Introduction

The term IoT (Internet of things) is an umbrella term used to describe physical objects or groups of such objects that are embedded with sensors, processing ability, software and other technologies that connect and exchange data with other devices and systems over the Internet or other communication networks. Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things. IoT is most closely associated with "Smart sensing and its applications" off-late owing to a surge in the consumer market. IoT is versatile in the sense that it finds applications in a wide array of fields ranging from Thermodynamics, Healthcare, High energy physics, Astronomy, Telecommunications etc to name a few. In this project, the Arduino , which is an open source hardware, has been used. The Arduino UNO is a microcontroller board that functions like a mini computer to develop necessary interaction with programming software and IDEs. We integrate the DHT11 Temperature and Humidity Sensor with the Arduino Uno to obtain the relative humidity and temperature of the surroundings.

Temperature is one of the most important measurement parameters that is used for monitoring and control in various industries.Temperature sensors are employed for a broad variety of practical purposes across many industries throughout the world. Essentially, these sensors provide input to a system in order to either approximately or accurately determine the temperature of a particular object or environment. From food processing to medical applications, to petrochemical handling and automotive monitoring, to biological research and geological studies, to HVAC systems and other consumer electronics, temperature sensors are a crucial tool in countless fieldsLikewise, humidity is also an important parameter.

Humidity is defined as the amount of water vapor in the air. The most commonly used term is “relative humidity” (RH) as this quantity is relative to the temperature. Throughout the history of relative humidity measurement, there have been numerous approaches to the task. Early hygrometers would measure the physical changes of different materials that would absorb water vapor, and examples included animal hair and paper coils. These methods proved to be inaccurate, with limited measurement range and high maintenance requirements. Other methods of calculating the relative humidity were then developed and some are still used today, such as the sling psychrometer and the chilled mirror hygrometer. These methods use evaporation and condensation to measure and calculate the humidity levels indirectly, but they are not always ideal. As demand grew for more accurate and full-range relative humidity measurements in weather and research applications, a more accurate and low-maintenance solution was needed.   After the invention of the transistor, the first electrical humidity sensors were then developed. These sensors were based on measuring electrical resistance, where a material would absorb water vapor and their resistance changed accordingly. These sensors also had their drawbacks, such as poor stability, limited accuracy, and difficulty responding to decreasing humidity levels, also known as hysteresis.

This project hence aims to build a subtle prototype of a temperature and humidity sensor that can help monitor these parameters. The proposed set-up is cost-effective and is easy to implement considering its ease of integration with the IDE.

Objectives

The objective of this project is to understand the working and underpinnings of the Internet of Things, specifically the use of Arduino Uno and its vast capabilities that make it easily integrable with various sensors and other softwares. This project also explores the use of the Arduino Programming Language which is a framework built on top of C++ (we have used the Arduino IDE in this project). The aim is to develop a conducive framework that outputs the relative humidity and temperature of the surrounding environment.

Materials and Methods

a) Arduino UNO

The Arduino UNO is a a microcontroller board with 14 digital input/output pins, 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started

The figure below shows an Arduino UNO Micro-controller board

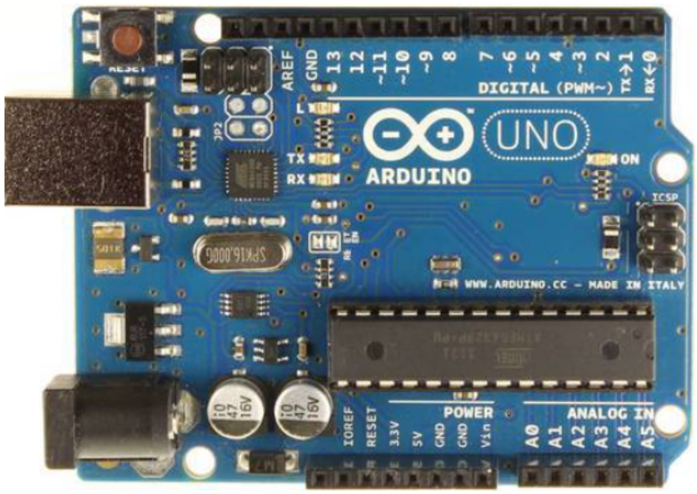

Figure 1 : Arduino UNO. Image Source : [here]

b) DHT11

The DHT11 is a commonly used temperature and humidity sensor. The sensor comes with a dedicated Negative Temperature Coefficient to measure temperature and an 8-bit microcontroller to output 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 0ºC to 50ºC and humidity from 20% to 90% with an accuracy of ±1ºC and ±1%. This range of temperature and humidity is usually used in hospitals and industries where the temperature and humidity need to be controlled. The DHT11 sensor is factory calibrated and outputs serial data, and hence it is highly easy to set it up. The DHT11 is a basic, ultra low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air and spits out a digital signal on the data pin (no analog input pins are needed). The only real downside of this sensor is you can only get new data from it once every 2 seconds, so when using our library, sensor readings can be up to 2 seconds old. The figure below shows a DHT11 sensor.

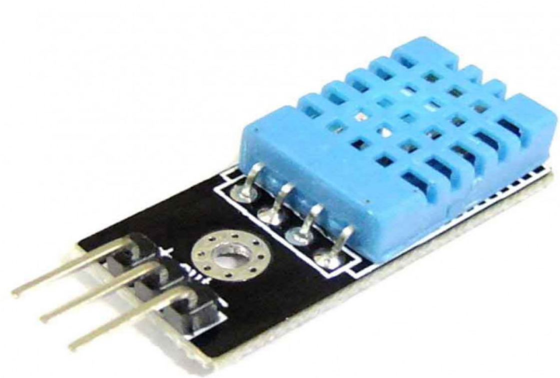

Figure 2: DHT11 sensor . Image source: here

c) Jumper Wires

Jumper wires are simply wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed.Jumper wires typically come in three versions: male-to-male, male-to-female and female-to-female. The difference between each is in the end point of the wire. Male ends have a pin protruding and can plug into things, while female ends do not and are used to plug things into. We have used male to male Jumper wires for this project.

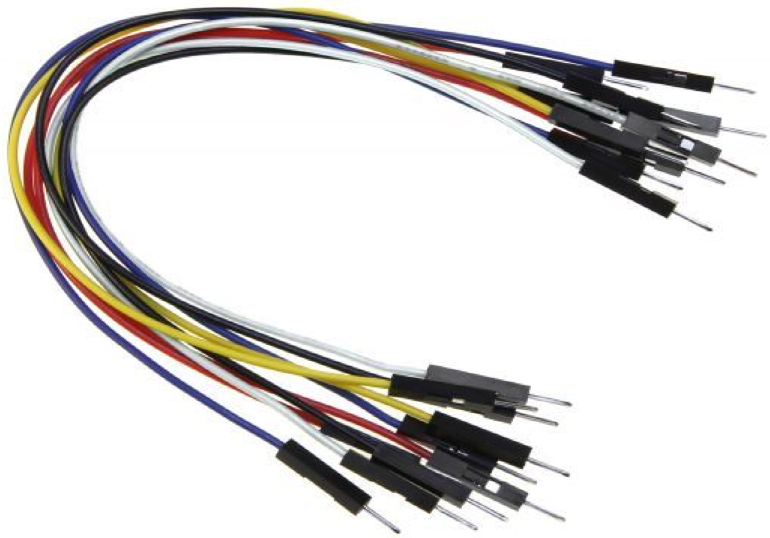

Figure 3: Jumper Wires (Male to Male) . Image Source: [here]

d) Breadboard

A breadboard, or protoboard, is a construction base for prototyping of electronics. Because the solderless breadboard does not require soldering, it is reusable. This makes it easy to use for creating temporary prototypes and experimenting with circuit design. Compared to more permanent circuit connection methods, modern breadboards have high parasitic capacitance, relatively high resistance, and less reliable connections, which are subject to jostle and physical degradation. Signaling is limited to about 10 MHz, and not everything works properly even well below that frequency.

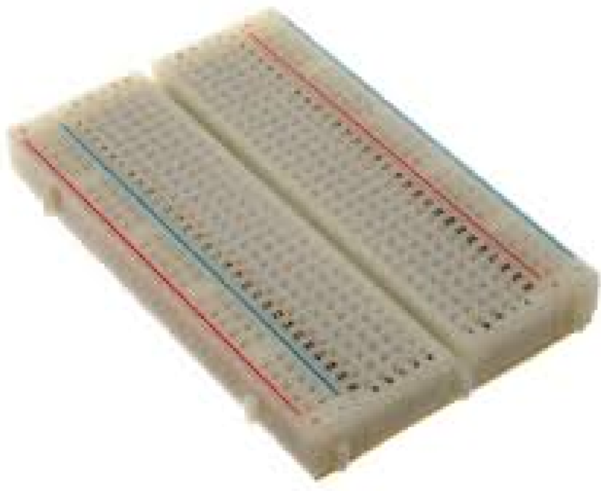

Figure 4: Breadboard. Image Source : [here]

Circuit Diagram and Image of the prototype

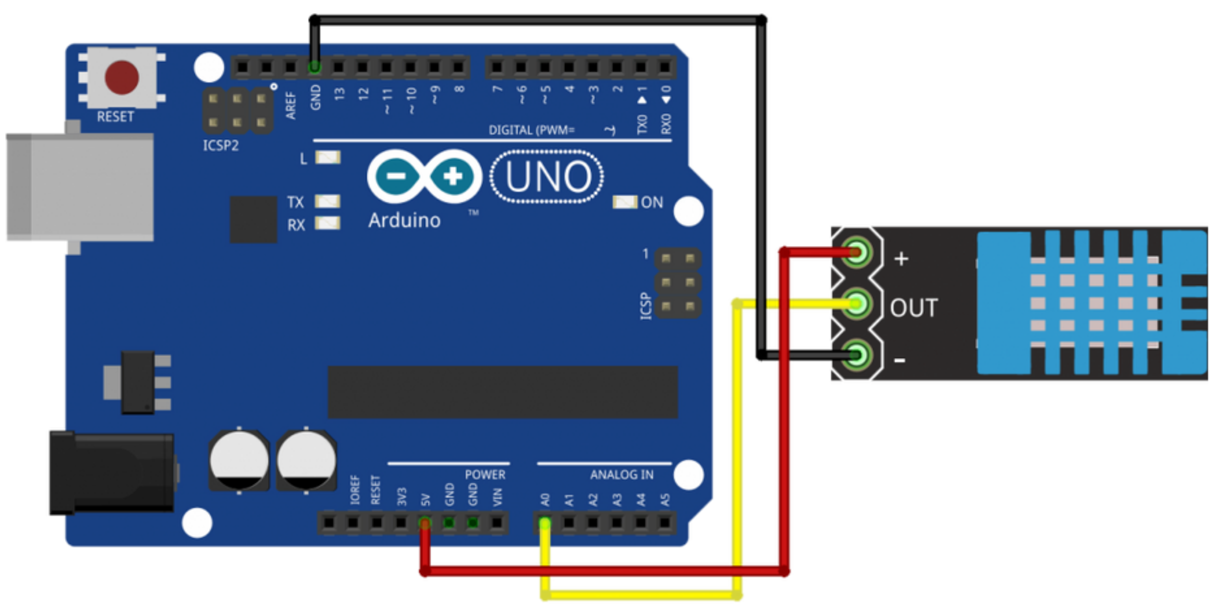

Figure 5: Circuit Diagram

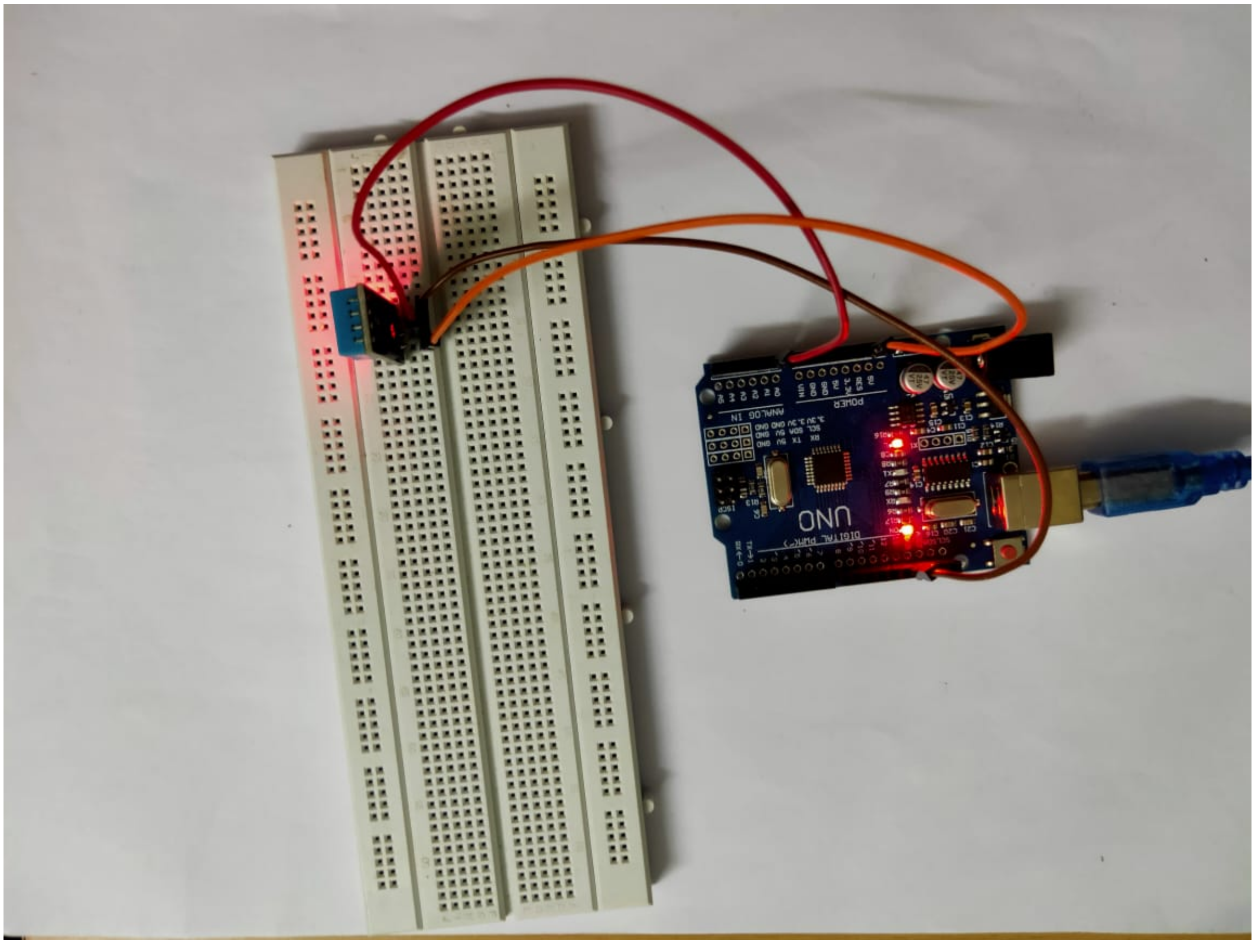

Figure 6: Prototype

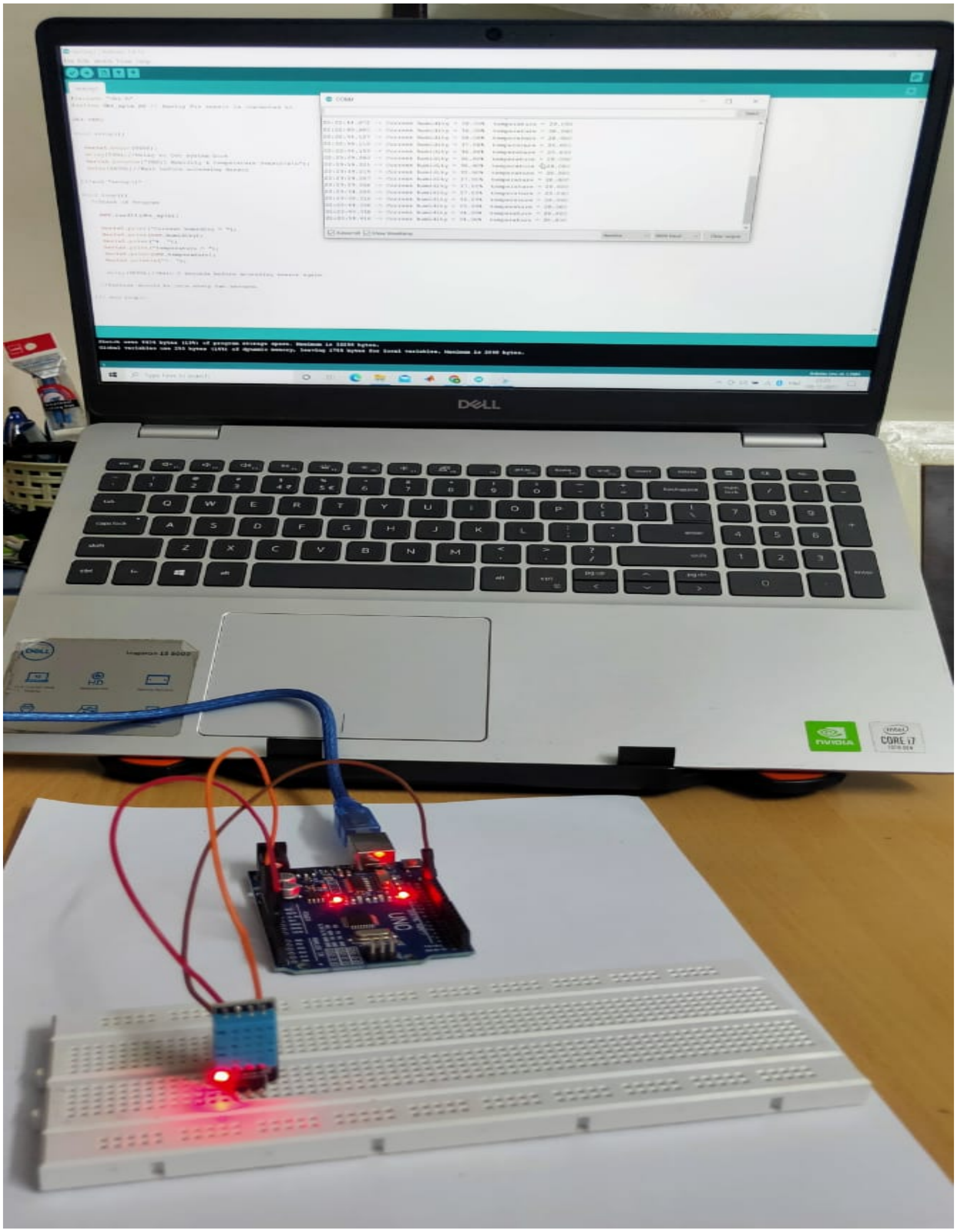

Figure 7: The complete set-up of the circuit. An LCD could also have been used in place of the serial monitor to display the results (but the former involved soldering and few other technical difficulties).

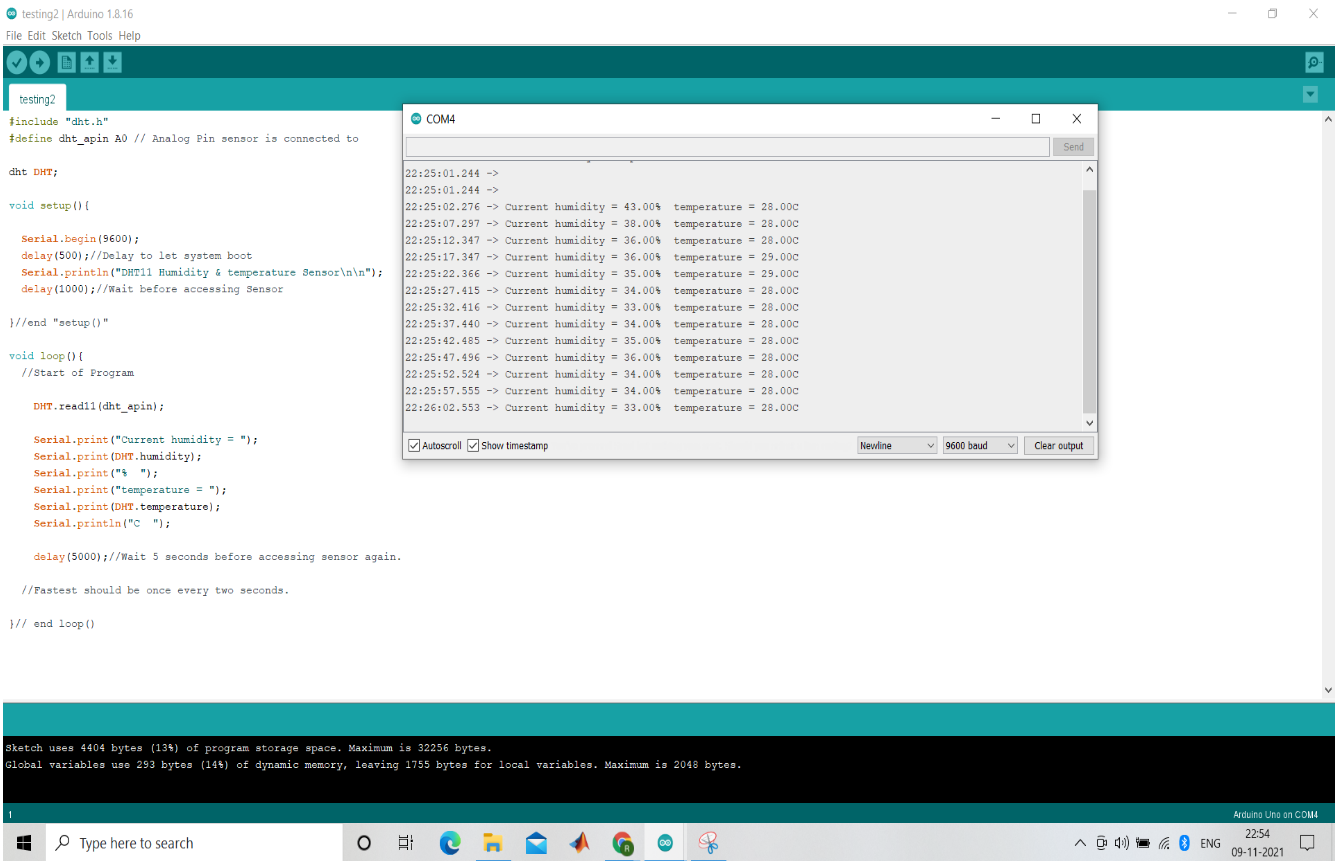

Figure 8: A screenshot of the output screen that shows the temperature and relative humidity with the respective timestamps

Major Connections:

Negative pin of the sensor is connected to the ground terminal of the Arduino.

The centre pin of the sensor is connected to the A_0 or Analog 0 pin of the Arduino.

The positive pin of the sensor is connected to the 5V voltage terminal of the Arduino.

The code makes use of an already available standard library 'DHT' that was installed (alternatively, it can also be downloaded from GitHub and then extracted to the libraries folder of the Arduino IDE).

The following is the code pertaining to the project (written in the Arduino Programming Language):

```
dht DHT;


void setup()  {


  Serial.begin(9600);

  delay(500);//Delay to let system boot

  Serial.println("DHT11 Humidity & temperature Sensor\n\n");

  delay(1000);//Wait before accessing Sensor
```

```
                  }

void loop()    {
 //Start of Program

   DHT.read11(dht_apin);

   Serial.print("Current humidity = ");
   Serial.print(DHT.humidity);
   Serial.print("%  ");
   Serial.print("temperature = ");
   Serial.print(DHT.temperature);
   Serial.println("C  ");

   delay(5000);  //This is to ensure that we wait 5 seconds before accessing the sensor again.

                  }
```

Results and Discussion:

The results seemed very close to the actual readings with a baud rate of 9600 and a short interval timestamp. There were very few fluctuations observed in the temperature readings. The slight change in temperature was observed when as an experiment, the door was opened for a while to allow inlet of warm air (it was a sunny day). Owing to convection currents, that increased the temperature of the room by a degree. A subsequent drop was observed after closing the door. The temperature readings recorded were in the range of 28-29^{o}C which was close to 28.6^{o}C noted using a mobile device in the same room. There were frequent fluctuations seen in the case of relative humidity ranging from 33% to 43% which was close to the recorded average relative humidity of 38.5% for that day at Bhopal.

Overall, the set-up was in decent working condition as could be gauged in comparison with the actual values of relative humidity and temperature monitored regularly via alternative sources. Owing to technical difficulties encountered during soldering and pinning, the Serial Monitor was used to display the readings instead of the LCD screen. Though the LCD screen might seem more sophisticated and cogent, the advantage of using the serial monitor was the timestamp list that saved the readings over the entire recording interval while the LCD only showed instantaneous readings.

The set-up could however have been improved or made more resilient by making stronger connections via soldering and making the system more sensitive to surrounding temperature conditions (making use of a potentiometer or other variable voltage sources). Incorporating more techniques like air quality sensors (to determine the dust or spm content), NFC (Near Field Communication Technology) , other app based interfaces could add to the level of sophistication and cogency of the set-up. Buzzers or alarms could be added to beep at a certain threshold or nominal value of humidity or temperature (such a set-up is generally sought after in bakeries, other food factories where ambience is very crucial).

Conclusion

The set-up employed in our system does provide a potent way for effective monitoring of temperature and relative humidity in real time. The advantages of this set-up lie in its simplicity, cost-effectiveness, ease of use in terms of hardware and software dependence and portability or compactness. Juxtaposed to the sensors used in the pharmaceutical and healthcare industries, the size, cost and complexity of this set-up is lesser. For sensing applications that allow marginal error rates, this set-up could be considered as a conducive and reliable way for real-time monitoring of temperature and humidity.

References

[1]https://www.brainy-bits.com/post/how-to-use-the-dht11-temperature-and-humidity-sensor-with-an-arduino

[2]https://www.hackster.io/HilfePlus/temperature-and-humidity-lcd-dht11-c73c5f#schematics

[3]https://www.esmagazine.com/articles/100617-room-sensors-belimo

[4]https://flaviocopes.com/arduino-programming-language/

[5]https://community.blynk.cc/t/correct-dht-library/30774

[6]https://components101.com/sensors/dht11-temperature-sensor

[7]https://www.vaisala.com/en/blog/2020-06/importance-humidity-sensors

[8]https://www.pyrosales.com.au/blog/thermocouple-information/what-are-temperature-sensors-used-for/