**SIT225 – DATA CAPTURE TECHNOLOGIES**

**9.1P: USE CASE DESIGN**

**HARRIET RAWSON – S218291057**

**INTRODUCTION:**

**Objective:**

In my house, the only room that experiences condensation is the ensuite. The condensation in this room appears as droplets and this dampness can lead to mould. As this room is secluded and not often inhabited, it’s important that I know when the conditions are conducive to condensation appearing. I would like to use the Arduino Nano and the DHT22 Humidity Sensor to measure the humidity and the temperature of this room, and when it reaches the dew point, it should send me an alert that tells me to go to the ensuite and turn on the heat lights and the fan.

**LITERATURE REVIEW:**

**Existing solutions:**

There are already existing devices that can monitor temperature and humidity, with app notifications. Examples of these include the [‘MySpool Temperature and Humidity Alert](https://myspool.com/content/temperature-and-humidity-alert-text-message-and-email-notifications)’ and the ‘[Govee WiFi Thermometer Hygrometer](https://www.amazon.com.au/Govee-Thermometer-Hygrometer-Temperature-Notification/dp/B09CKZGBC7)’. These are often used for greenhouses or wine cellars, but most of them do not take into consideration the dew point, only the standalone temperature and humidity. However, using these would defeat the purpose of what I have learned in this unit.

The dew point, according to media.bom.gov.au, is the temperature at which condensation is produced, based on humidity. To find the dew point at any given time, we calculate

A math equation with numbers

Description automatically generated

in which td is the dew point in degrees, t is the temperature, and RH is the relative humidity (Lawrence, 2005). These latter variables are easily found by many sensors. In this case, we will look at three Arduino-compatible sensors, the DHT11, DHT22, and DS18B20. In 2023 Yulizar et al. conducted accuracy tests by way of laboratory environments and field tests between these three sensors. The DS18B20 was found to have only 0.17°C, compared to 0.18°C for the two DHT sensors. Compared to an Automatic Weather Station in the field, the DS18B20 again performed the best, but the DHT22 came in still with 98.15% accuracy. For the purposes of humidity and dewpoint within the home, I believe that this difference is negligible.

In an article for Electropeak, Hosseini compares 15 different sensors compatible with Arduino for this purpose. Below is a summary of the differences between just some of them

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sensor | Accuracy | Sampling rate | Power consumption | Price | Suitable for this project? | Verdict |
| DHT22 | Temperature ±0.5°C  Humidity  ±2% | 0.5Hz | Operates between 3-6 volts | $10.10 | Yes | Selected, best ratio of accuracy:price |
| AM2320 | Temperature ±0.5°C  Humidity  ±3% | 0.5Hz | 3.3 or 5 volts | $7.55 | Yes |  |
| LMX35 | Temperature ±3°C |  | 4-30 volts | $3.95 | No | No humidity sensor |
| SHT71 | Temperature ±0.4°C  Humidity  ±3% |  | 2.4-5.5volts | $7.87 | Yes |  |

In general, it seems that the DHT modules are the most widely used with Arduino, and one of the few that include both humidity and temperature to a relatively accurate degree. It is understood that the DHT22 outperforms the DHT11, so we will use this in our project. Ahmad et al. conclude that the DHT22, while not suitable for an environment of rapid temperature change, is a good selection for a system of gradual temperature change and monitoring ambient temperature. It is also stated that the low cost is an advantage for large-scale monitoring.

**METHOD:**

**Materials:**

* [Arduino Nano](https://au.rs-online.com/web/p/arduino/1927589?bid=51580031&cid=rscDM746&cm_mmc=AU-EM-_-undefined_12%2Ful%2FFri%20-_-rscDM746-_-Product_Description_Link&dtm_em=e07f0de5f85b008a571fcce7ac7d0c6c2a4523c74f46fb4559095d219b3afaa9)
* [DHT22 Temperature and Humidity Sensor](https://core-electronics.com.au/dht22-module-temperature-and-humidity.html?gad_source=1&gclid=Cj0KCQjwjNS3BhChARIsAOxBM6qpGwQCfzxUWKJOsx-b7OK5_vkaDP7lj1bqBXqan5hghfqVOhYVfNAaAmDGEALw_wcB)
* [Generic Jumper Wires](https://core-electronics.com.au/female-to-female-dupont-line-40-pin-10cm-24awg.html)
* Vessel (this could be 3D printed, or made out of cardboard even)
* [Powerbank](https://www.pakronics.com.au/products/usb-powerbank-2000ma-5v-pakr-a0308?srsltid=AfmBOopuwOqUx1QOlzVTIscMuHATrC4KxF-CSoEoGmbYMiUBh0u_88yO)

**Arduino implementation:**

In order to read the temperature and humidity, the DHT22 must be connected to the Arduino Nano and a Sketch needs to contain the following integral parts:

* #include <DHT.h>
* #include “thingProperties.h”
* define pins
* a clause to catch a failed reading
  + The pseudocode for this would look something like this:

if (no humidity value or no temperature value) {  
print(“No reading from DHT22”);  
return;

* delay(300000) – check readings every 5 minutes.
* float humidity and temperature = dht.readHumidity or readTemperature
* float dewPoint = temperature – ((100-humidity)/5) <https://iridl.ldeo.columbia.edu/dochelp/QA/Basic/dewpoint.html>
* update Cloud variables (temp, hum, dew\_point)
* if (temperature <= dewPoint), send\_alert = true – if the temperature is equal to or below the calculated dew point, I want the Remote App to receive an alert.
* defer notifications. If the readings are every 5 seconds, I don’t want to receive a notification every 5 seconds should the conditions be met. To do this, I would implement an interval (900000 (15 minutes)). I would introduce variables called currentTime and passedTime. currentTime = millis() using Arduino’s built-in millis() function. This gives me the time since the current program started running and will be updated as the loop cycles. passedTime will record when the last alert was sent. My if statement will then become if (temperature <= dewPoint && (currentMillis - previousMillis >= interval)), send\_alert = true. This means that if a notification has been sent within the past 15 minutes, another will not be sent. Then I will reset the passedTime to equal currentTime.
* The final step is else {send\_alert = false;}

**Python implementation:**

I would like, for the sake of demonstrating this unit’s teachings, for each day’s data to be recorded in a CSV using Python. Python will connect to the serial port, and create a new CSV for each day.

To do this, each file will be created named ‘dewpoint\_{current\_date}’.

* import serial, import csv, from datetime import datetime
* current\_date = datetime.now().strftime("%d-%m-%Y")
* csv\_file = open(f'data\_{current\_date}.csv', 'w', newline='')
* csv\_writer = csv.writer(csv\_file)
* csv\_writer.writerow(['Timestamp', 'Temperature', 'Humidity', 'Dew Point'])

From here, Python will read the data from the serial port and record in the CSV a Timestamp, Temperature, Humidity, and Dew Point on one line for each recording.

**Arduino Thing Setup:**

In the Arduino IoT Cloud site, I will create a new Thing, adding the properties float temperature, float humidity, float dewPoint and Boolean alert. I will need to add a trigger (which means upgrading my Arduino package to ‘Maker’ for $6 a month). The trigger will be set such that when send\_alert is equal to true, I will receive a custom notification that says “Ensuite at dew point. Turn on heat lights and fan.”

**Arduino Dashboard:**

The Arduino Dashboard will also have a line graph simultaneously showing temperature, humidity, and dew point, so that trends can be viewed over time, or the user could see when the temperature and humidity are satisfactory.

**Gantt Chart Plan:**

A graph with text and colorful rectangles

Description automatically generated

The first three stages are self-explanatory, but the fourth stage, interface and physical design, is put in place to allow time to create a vessel for the Arduino to be housed in. This could be 3D printed, or just simply put in a nice cardboard box with a hole for the sensor. At this point, the Arduino Dashboard should be finalised and laid out in the most user-friendly manner.

**BIBLIOGRAPHY:**

Ahmad, Y.A., Surya Gunawan, T., Mansor, H., Hamida, B.A., Fikri Hishamudin, A. and Arifin, F. (2021). On the Evaluation of DHT22 Temperature Sensor for IoT Application. *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*.Available at: <https://doi.org/10.1109/iccce50029.2021.9467147>.

Hosseini, A.A. (2023). *15 Best Arduino Compatible Temperature Sensors [2024]*. [online] Electropeak. Available at: https://electropeak.com/learn/15-arduino-compatible-temperature-sensors-for-your-project/.

Intan Kusuma Wardani *et al.* (2023) ‘The feasibility study: Accuracy and precision of DHT 22 in measuring the temperature and humidity in the greenhouse’, *IOP conference series*, 1230(1), pp. 012146–012146. Available at: <https://doi.org/10.1088/1755-1315/1230/1/012146>.

Lawrence, M.G. (2005). The Relationship between Relative Humidity and the Dewpoint Temperature in Moist Air: A Simple Conversion and Applications. *Bulletin of the American Meteorological Society*, 86(2), pp.225–234. doi:https://doi.org/10.1175/bams-86-2-225.

media.bom.gov.au. (2017). *Feeling hot and bothered? It’s not the humidity, it’s the dew point - Social Media Blog - Bureau of Meteorology*. [online] Available at: https://media.bom.gov.au/social/blog/1324/feeling-hot-and-bothered-its-not-the-humidity-its-the-dew-point/.

Yulizar, D., Soekirno, S., Ananda, N., Prabowo, M.A., Perdana, I.F.P. and Aofany, D. (2023). Performance Analysis Comparison of DHT11, DHT22 and DS18B20 as Temperature Measurement. *Advances in Physics Research*, pp.37–45. Available at: https://doi.org/10.2991/978-94-6463-232-3\_5.

‌