Weather-Tek

by

Emmanuel Molina
Dedi Nuragung
Bassam Dakhel
Alaa Rahime

Table of Contents

1. Introduction

1.1 Problem Statement

The goal is to design a weather station which can measure and display temperature, humidity levels, and wind speed to the client. It must be eco-friendly and safe to use. This would mean we have to find a power source that meets this requirement. It would also have to be made with minimal material and not produce any waste.

1.2 Review of Literature

Cactus.io et. al [1] created a weather station project that uses an Arduino board with an Ethernet for communication. The station measures both external and internal temperature as well as humidity, pressure, wind speed, wind direction and includes a rain gauge. This project uses a total of 4 sensors: a DS18B20, a BME280, a Hydreon RG-II and a Davis Anemometer. It has two LED lights; one to show power is being supplied to the control box and another to show if data is being transmitted to the remote server. This project makes their container out of a 100mm PVC pipe. This project required a 12V input which was converted to 9V then outputted into the Arduino board. Dejan et. al [2] created this weather station that has wireless communication between it's indoor and outdoor parts using NRF24L01 transceiver modules. Having two parts creates the need for two Arduino controllers. The indoor part includes an 0.96" OLED screen to display information. The indoor part is powered by a 9V-12V power jack while the outdoor part is powered by two 3.7V batteries. The outdoor part only has one sensor module which is a DHT22 sensor. This sensor measures both temperature and humidity.

The indoor part of the project also has a DHT22 sensor which gives the user the ability to compare the indoor and outdoor temperature and humidity. This project also uses a DS3231 Real Time Clock to provide the time. The clock module has the ability to keep track of time when the Arduino is out of power. The case that holds everything together is made from a 8mm thick MDF board cut using a circular saw and put together using wood glue, clamps and screws. Open green energy et. al [3] made a project that uses a DHT11 sensor which measures temperature, humidity, rainfall and pressure. It uses a transmitter receiver pair to send and receive information. It is powered by solar panels that charge 2 AA batteries.

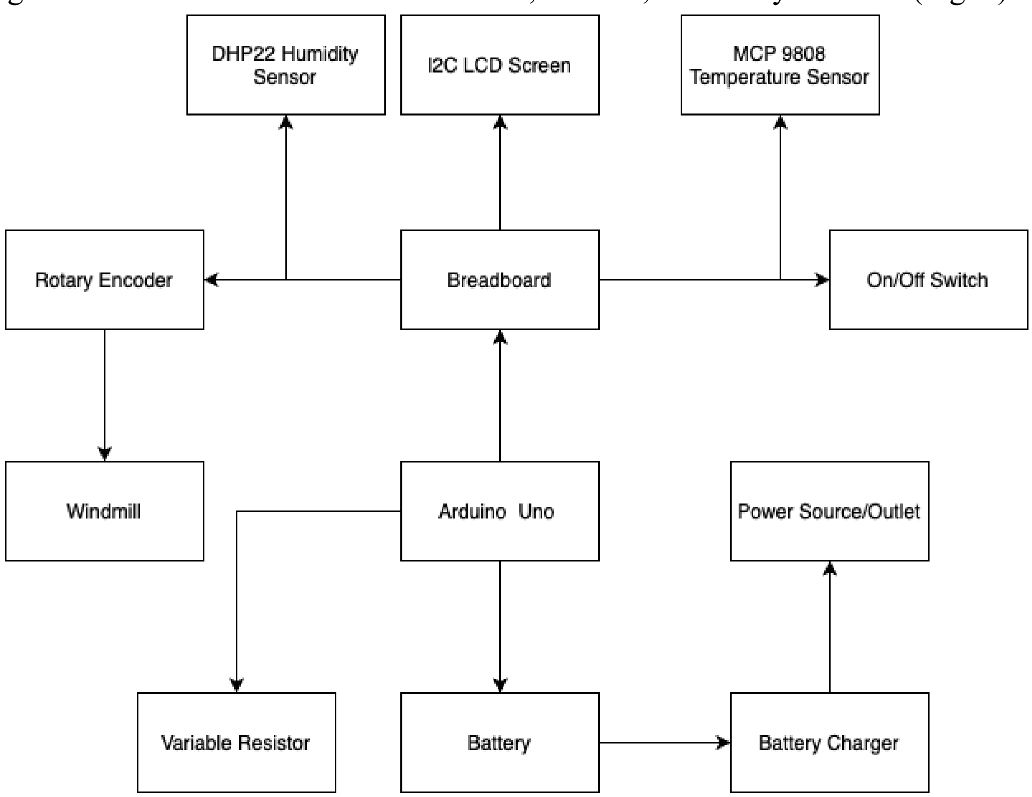
To power the arduino the voltage required is 5V which means a boost converter is required for this project. There are two Arduino boards used in this project; an Arduino Nano and an Arduino uno. The display is an LCD that displays four rows of 20 characters. The display part is made out of a cardboard box while the outdoor part is made out of a plastic container. The reason that this project uses solar panels is that it is impractical to use a cable to power the outdoor part. The indoor display module however uses non-solar generated energy. Dan Fein et. al [4] created a project which has an interesting design that looks like a gamebot. It's powered by Arduino UNO. The DHT22 sensor is used for temperature and the BMP085 sensor is used to measure pressure. It displays the output on a ks0108 LCD screen. This project also uses a 10k ohm resistor as well as a 10k trimpot.

The main difference between this project and the other projects researched is that this one has a unique but simple design. The “robot” is modeled on BMO. Amandaghassaei et. al [5] created a project that explains how to power an Arduino with a battery. It explains the number of volts required to power an Arduino and how to use different voltages. In the author's opinion a 9V battery “works well”. When using a 9V battery it is a good idea to use a battery snap connector and a toggle switch. While a 9V battery is ideal, anywhere between six and twenty volts works. The recommended range is between seven and twelve volts.

2. Design

2.0 Overview;

The electronics are controlled by an Arduino with the breadboard as a base to connect all the wiring to. The sensors included are MCP9808, DHT22, and rotary encoder. (Fig. 1)



(Figure 1)

3D Printed rectangular shaped box with dimensions 19.8 x 10.9 x 10.0 cm(L x W x H). our goal is to create an eco friendly design that minimizes the use of materials though still achieved essential and necessary elements of the weather station. Below is a Completed ¾ view sketch of the overall design. (Fig 2)

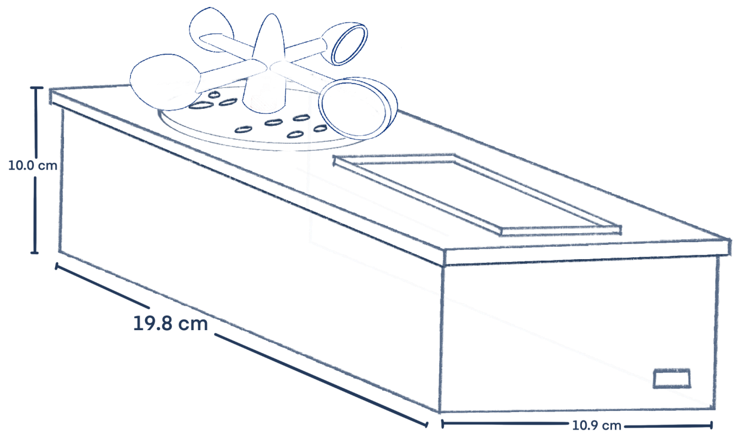


Figure 2

2.1 Structural

The structural design is non-conventional. It is also meant to be portable as it does not require an external power source since it contains a rechargeable, internal battery.

The base of the structure measures 19.8cm in length, 10.9 cm in length, and 10.0 cm in height. The layers are 3mm in thickness all around. This should provide enough stability and integrity. The turbine measures 13.75 cm in diameter from the end of one cup to the opposite. Its height measures 6.25cm from its base to the tip. The base diameter measures 2.4 cm.

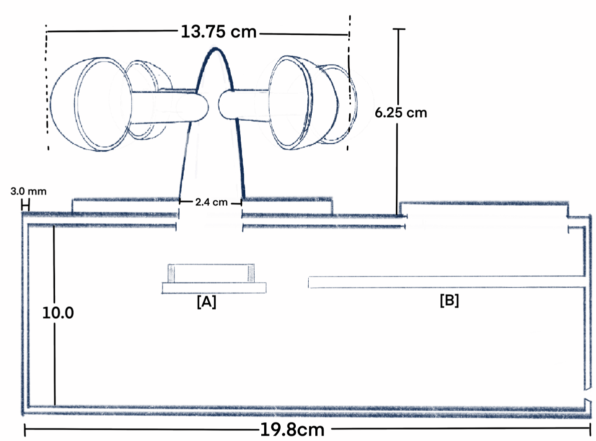


Figure 3

Brackets [A] (Fig. 4)and [B] (Fig.5) allow the encoder and screen to be propped higher within the box. This will make so that the encoder and screen can be level with the top lid.(Fig. 3) .

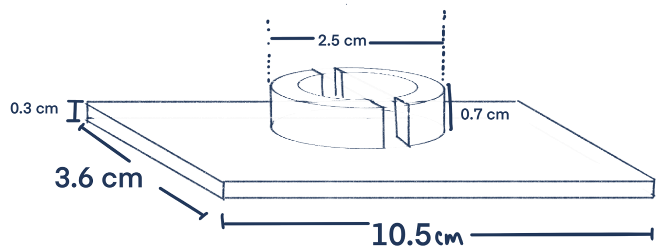


Figure 4 [A]

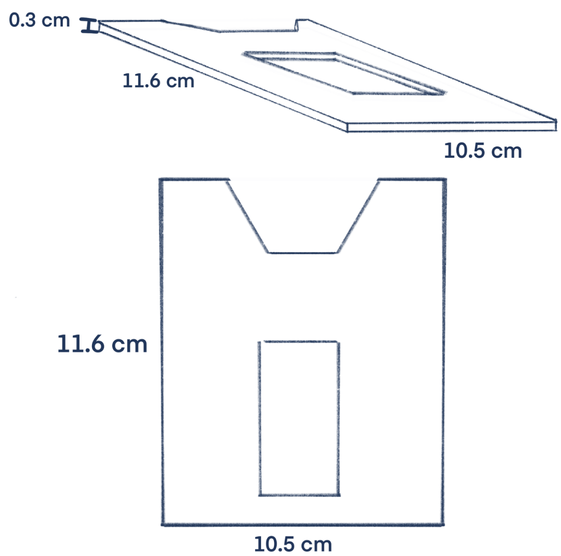


Figure 5 [B]

Incorporating the brackets to raise the encoder and screen will allow for room to hide the other electrical components on the bed of the box. It is designed so that the breadboard containing the temperature and humidity sensors to be placed underneath the pinholled section of the lid. This will make it so that the environment can come in contact with the sensors giving more accurate results. (Fig. 6)

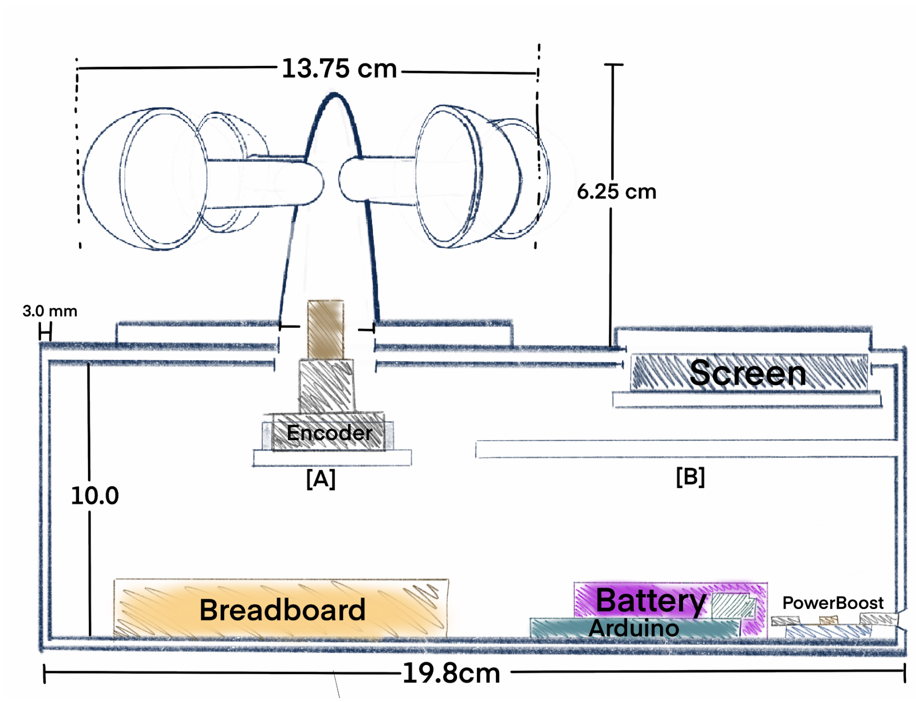


Figure. 6

2.2 Mechanics

The turbine is the biggest part of the design. It takes up the most space as it should have no trouble in accurately measuring wind speed without any interference from the physical design. There is a cutout within the base of the turbine which houses the rotary encoder allowing it to spin while keeping it stable. There are 4 cups to ensure that the turbine can efficiently catch enough wind in order to spin. (Fig 7 .)

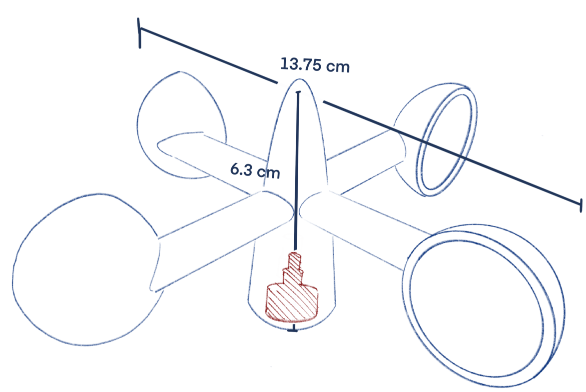


Figure 7

It is designed so that the breadboard containing the temperature and humidity sensors to be placed underneath the pinholled section of the lid. This will make it so that the environment can come in contact with the sensors giving more accurate results (Fig.8)

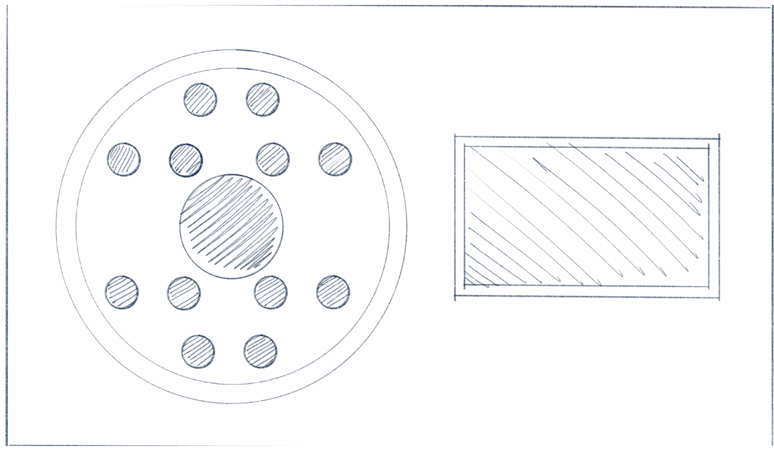


Figure 8

The battery is a 3.5V lithium ion battery which can be recharged with the PowerBoost LiPo charger. This will allow for less electronic waste. On the side of the design, there will be an acrylic indicator to show the battery status using the integrated LEDs. It can be charged through a simple outlet adapter.

2.3 Software

The sketch that we run for all the sensors based of our original sketch,
What it does to the encoder is that we initialize a position and wait couple of seconds and get a new position as the encoder rotates, which works exact same thing as to get the initial time and the new time, but the only difference with time is that we initialize its counter to be zero to start with compare to Position.

When it comes to the temperature and humidity sensor, the sensor will delay every 2 seconds to get new information surrounding it over and over again, and will display an error message when there's something wrong with the wiring or the sensor itself.

With the combined sketch, each sensor will display to the screen every 2 seconds, 2 seconds for humidity and temperature and 2 seconds for wind speed.

-

- Flowchart for DHT22

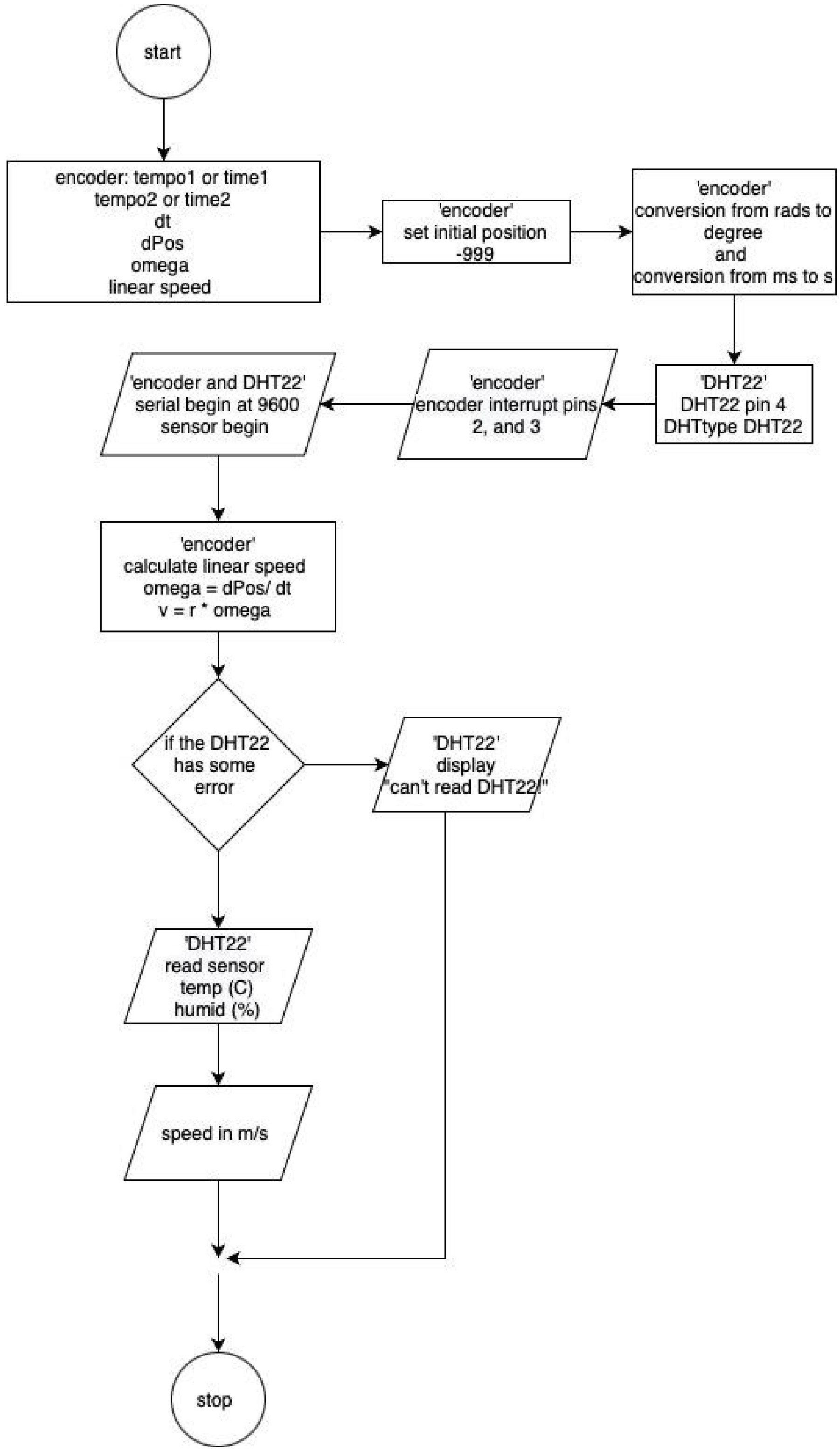


Figure 9

2.4 Electrical

Wiring for all the sensors

- I2C LCD to Arduino

  The hardware of I2C LCD display consists of a HD44780 based character LCD display and an I2C LCD adapter. It is a 16 x 2 character meaning it consists of two rows and 16 columns that can display ASCII characters with mounted LED backlight.

  For the wiring of I2C LCD :
    - we plug the VSS which is pin number one to ground (GND).
    - the pin two of the LCD connected to the 5V of Arduino.
    - The pin 3 of LCD which control the brightness of the LCD connected to pin 2 of the potentiometer that is attached to the Breadboard.
    - The pin 4 (RS) of LCD connected to the pin 12 on the Arduino.
    - The pin 5 (RW) of LCD connected to the Ground (GND) pin.
    - The pin 6 (EN) of LCD connected to the pin 11 of the Arduino.
    - The pin 11 of the LCD connected to pin 9 of the Arduino.
    - The pin 12 of the LCD connected to pin 8 of the arduino.
    - The pin 13 of the LCD connected to pin 7 of the Arduino.
    - The pin 14 and 15 of the LCD connected to the vss and ground respectively.

- Encoder to Arduino

  We use an incremental encoder which generates a pulse for each incremental step in it's rotation. The type that we use produces about 200 pulses per rotation, Which we can convert that to an angle everytime the encoder rotates to get angular speed which we can again convert that to angular speed to measure the wind speed data.

  For the wiring of the Encoder to Arduino :

    - Pin 1 VCC connected to 5V Arduino pin.
    - Pin 2 and 3 which is Signal A and B connected to pin 2 and 3 which has interrupt on the arduino board, An Interrupt job is to make sure that the processor responds quickly to important events and ignore the rest of the task to prioritize said task.
    - Pin 4 GND connected to the GND Arduino pin. DHT22 to Arduino.

- DHT22 to Arduino

DHT22 consumes about 2 mA when it's reading the temperature, with a max current 2.5 mA and operating power 3 to 5V.

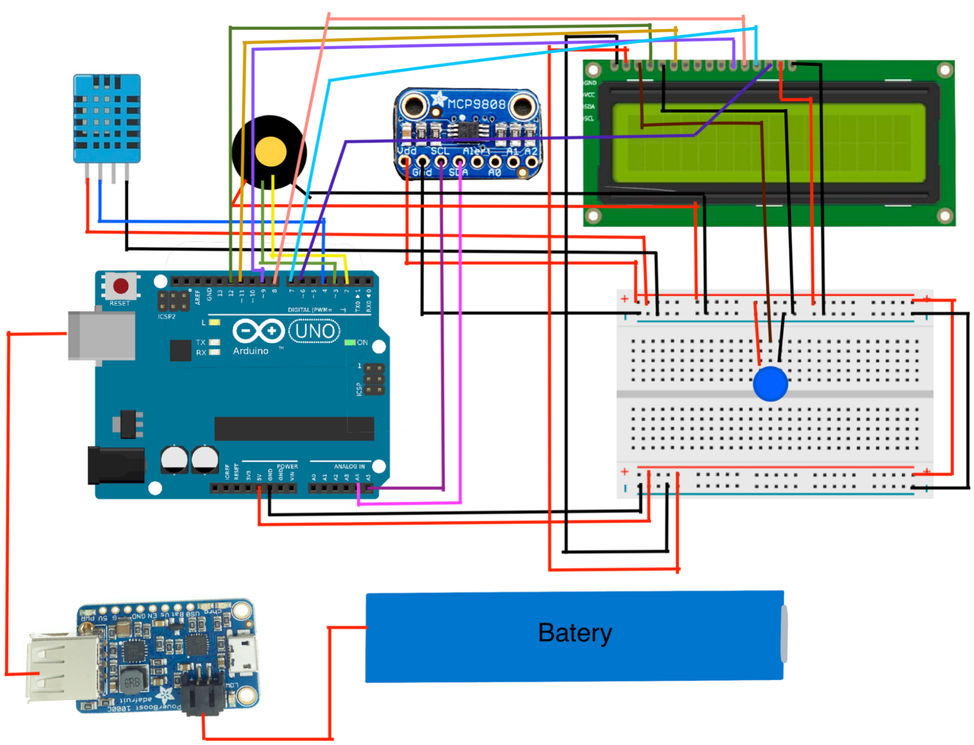
For the wiring of I2C LCD :

- Pin 1 of DHT22 connected to 5V pin of arduino.
- Pin 2 of DHT22 connected to pin 4 of the arduino.
- We don't use the pin 3 from the DHT22.
- Pin 5 connected to the GND pin of the arduino.

- MCP9808 to Arduino

For the wiring of I2C LCD :

- VDD pin of MCP9808 connected to the 5V Arduino pin.
- GND pin of MCP9808 connected to the GND Arduino pin.
- SCL pin of MCP9808 connected to the A5 Arduino pin.
- SDA pin of MCP9808 connected to the A4 Arduino pin.



The wiring for all sensors, Arduino, Powerboost, and battery.

3. **Testing**

MCP9808

We tested the MCP9808 indoor which reads normal room temperature, and as soon as we placed my finger on the sensor for a minute, the temperature rises up because the body temperature tends to be hotter compared to room temperature, and that concludes the testing of the MCP9808.

Due to the box case being less ventilated, the temperature drops or rises tend to get slower compared to testing it without the box case.

The picture below shows us the final product with the case and working LCD that displays the temperature and humidity.

Final product.

DHT22

We had a couple testing that works, and few testing that fails due to the Arduino UNO that wont power. We got the replacement of the Arduino UNO as soon as possible and rewired it again and never had an error since then.

For the DHT22 sensor we tested both inside and outside for the accurate measurements. The humidity reading and temperature reading drops down quite drastically from being placed inside room temperature to the outside temperature, which as it says on the data sheet of the DHT22 that the sensor has a humidity range between 2 to 5% and accuracy of the temperature 0.5 degrees celsius.

Temperature testing

The top picture represents outdoor temperature and the bottom picture represents indoor temperature. The top photo reads Tmp: 27.00 C, Hmd: 23.00%. The bottom photo reads: Tmp: 28.00 C, Hmd: 27.30% This is after being exposed to the outside weather for 2 minutes. The temperature changes slightly due to the sensors being inside the majority of the time. It may also stem from the enclosure being tight. However, if exposed for a long period of time , it will stabilize and produce an accurate result.

I2C LCD

We tested the LCD screen with a simple display that outputs to the LCD screen, we tested with an open library found on Arduino library. We were able to know how does the screen works and how should we placed all outputted reading sensors that nice to look at when it displays to the LCD screen.

The below picture shows that the temperature displays on row 0 column 0, and the humidity that displays on row 1 column 0, we then delay it to show the wind speed that encoder produces when we rotate it which places it on row 0, and column zero.

Top picture temperature column 0 row 0, humidity column 0 row 1.
The bottom image wind speed 0 column 0 row.

RES20D Incremental Encoder

The tested the encoder countless times, it keeps outputting zero, then we display everything using the serial print to see where it went wrong and we learn our lesson that to calculate the rational number, the conversion from radian to degree is a rational number, we have to use function `floats` rather than function `long` since function floats only calculate an integer.

The picture below shows that when the turbine is spinning, it outputs the wind speed in meters per second units to the I2C LCD 16x2.

The turbine looks blurry because it spins, and outputs 1.58 m/s linear speed.

4. Summary & Conclusions

Our design includes 3 sensors, one to measure humidity, another for temperature and a third for wind speed. It has a lithium battery that can be charged with a PowerBoost Lipo charger. It also includes an integrated LED battery indicator. The shell's material is to be made of plastic. Our design can be improved by using more eco-friendly material. It could also have been made as a larger design with a larger and more detailed display. Multiple more sensors could be added. The design could also be made to look more aesthetic and not as boxy. Having a green power source such as solar panels would help

The design is missing one more sensor. It senses wind speed, humidity and temperature. The temperature and humidity, however, are one in the same. If we were to add another sensor, we would have added a pressure sensor. This would remind the client about atmospheric conditions before leaving for his day.

When 3D printing and modeling we have learned that while making a model, we have to take into account of print nozzles. This ended up making our prints 1mm smaller than intended resulting in most of our components not fitting.

To avoid wiring problems for future projects we need to take into an account of how much wiring we need to plug to all the sensors and arduino, and how much space we need for all sensors and wiring that can fit into the box.

Reference

[1] cactus.io, "Arduino Weather Station Project - Overview," *Design - Make - Code - Deploy*. [Online]. Available: http://cactus.io/projects/weather/arduino-weather-station. [Accessed: 20-Jan-2021].

[2] Dejan, "Arduino Wireless Weather Station Project," *HowToMechatronics*, 05-Feb-2021. [Online]. Available: https://howtomechatronics.com/tutorials/arduino/arduino-wireless-weather-station-project/. [Accessed: 20-Jan-2021].

[3] Opengreenenergy and Instructables, "SOLAR POWERED ARDUINO WEATHER STATION," *Instructables*, 02-Jun-2019. [Online]. Available: https://www.instructables.com/SOLAR-POWERED-ARDUINO-WEATHER-STATION/. [Accessed: 20-Jan-2021].

[4] Dan Fein. "Weather Robot," *Arduino Project Hub*. [Online]. Available: https://create.arduino.cc/projecthub/4244/weather-robot-c5a2f2?ref=tag&ref_id=weather&offset=62 . [Accessed: 20-Jan-2021].

[5] Amandaghassaei, & Instructables. (2017, October 28). Powering Arduino with a battery. Retrieved March 09, 2021, from https://www.instructables.com/Powering-Arduino-with-a-Battery/

Appendix

Appendix 1: Combine code for DHT22, RES20D50 Encoder, and I2C LCD

```
#include <DHT.h>
#include <DHT_U.h>
#include <Adafruit_LiquidCrystal.h>
#include <Encoder.h>

#define DHTPIN 4
#define DHTTYPE DHT22
Encoder myEnc(2, 3);

DHT dht(DHTPIN, DHTTYPE);
Adafruit_LiquidCrystal lcd(12, 11, 9, 8, 7, 6);

// speed stuff
float tempo1;
float tempo2;
float oldPosition = -999;
const float theta = 0.031415;
const long conversion = 1000;

float dt;
float dPos;
float omega;
float linearSpd;



void setup() {
 Serial.begin(9600);
```

```
dht.begin();
lcd.begin(16, 2);

lcd.setCursor(0,0);
lcd.print("hmd:");        // print humidity
lcd.print("tmp:");        // print temperature

lcd.setCursor(0,1);       // move to the next line
lcd.print("wspd:");       // print wind speed

}

void loop() {
  enc_reading();     // wind speed measurement using arduino
(encoder)
  dht_reading();     // temperature humidity reading from DHT22
sensor

}


void enc_reading() {
 long newPosition = myEnc.read();
 if (newPosition != oldPosition) {
   tempo2 = millis();
   dPos   = abs(newPosition - oldPosition); //  this absolute
value will help with the CCW issue
   dt     = float(tempo2 - tempo1) / conversion;
   oldPosition = newPosition;
   tempo1      = tempo2;
```

```
   omega     =  (float)(dPos * theta) / (float)(dt); //
conversion based of rotation of per revolution
   linearSpd =  0.14* omega;                               //
radius of 0.14m

   lcd.setCursor(0,0);
   lcd.print("WindSpd ");
   lcd.print(linearSpd);
   lcd.print(" m/s");
   delay(1500);
   lcd.clear();
   Serial.println("WindSpd: ");
   Serial.println(linearSpd);
   Serial.println(" m/s");
 }
}

void dht_reading() {
 delay(2000);
float h = dht.readHumidity();
float t = dht.readTemperature();

if (isnan(h) || isnan(t) ){
 Serial.println("Failed to read from DHT sensor!");
 return;
}

lcd.setCursor(0,1);
lcd.print("Hmd: ");
lcd.print(h);
lcd.print(" %");
```

```
lcd.setCursor(0, 0);
lcd.print("Tmp : ");
lcd.print(t);
lcd.print(" C");
delay(2000);
lcd.clear();
Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(t);
Serial.println(" *C ");
}
```

Appendix 2: Detailed Drawings

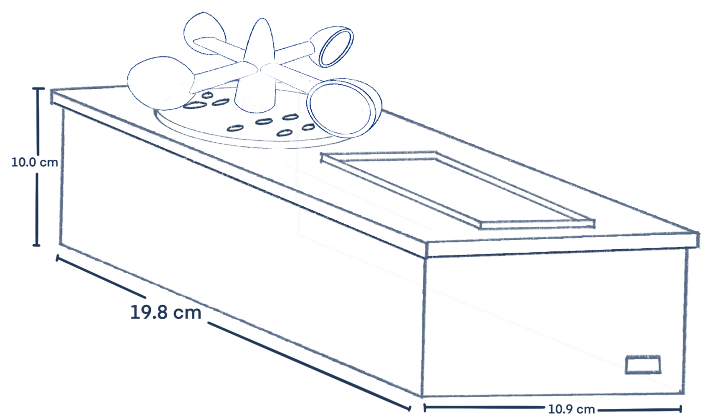


Figure 2-¾ view of structure

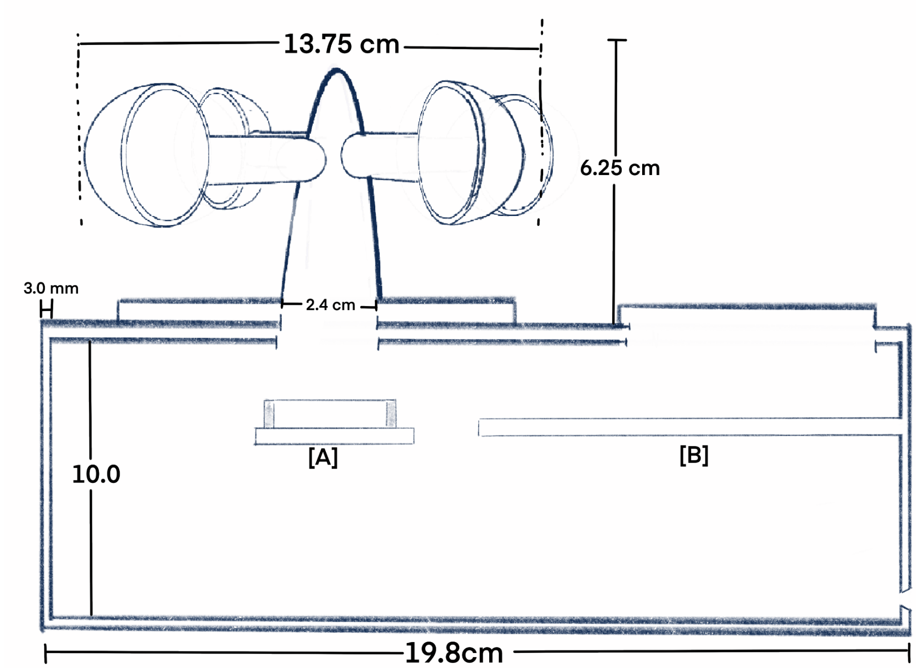


Figure 3- Main body and supports

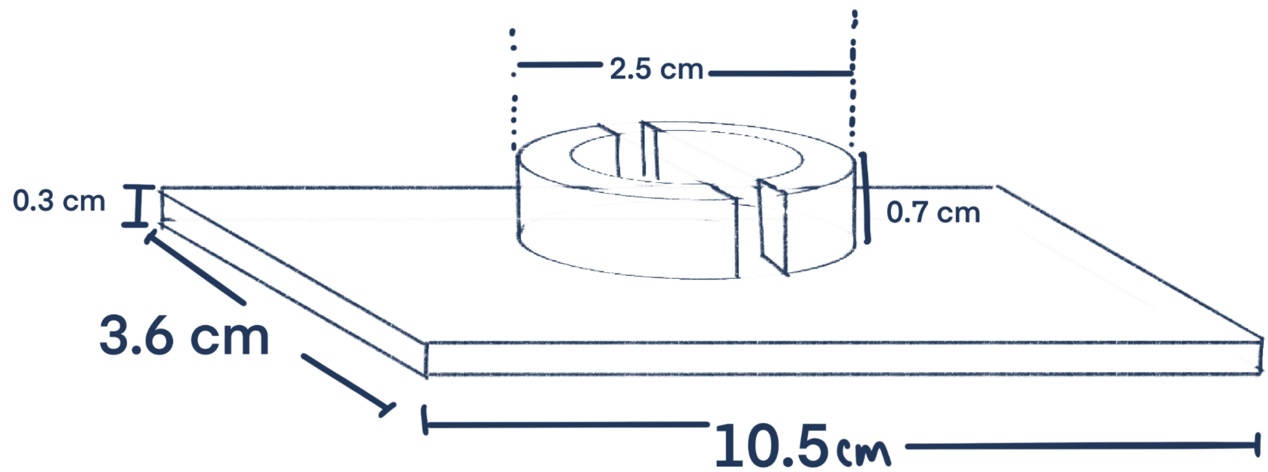


Figure 4[A]-Bracket for the encoder

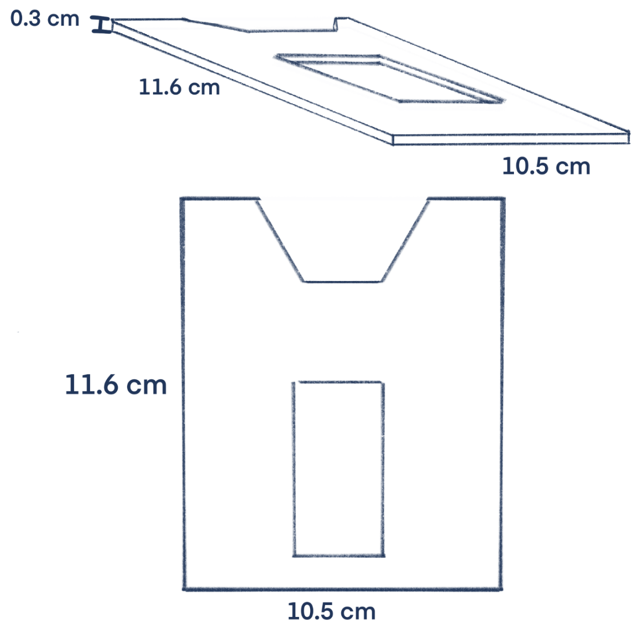


Figure 5[B]-Bracket for the LCD screen

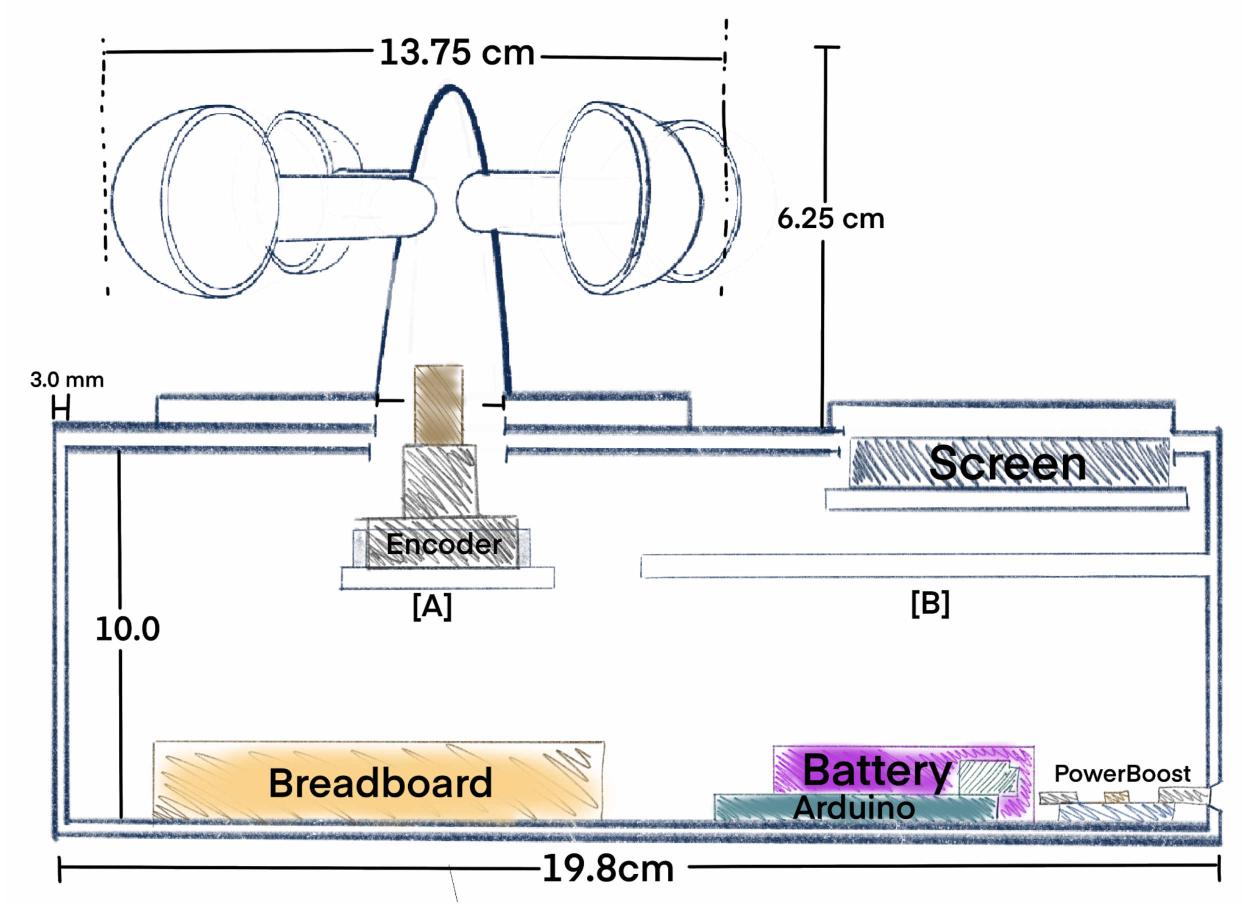


Figure 6-Internal View of structure

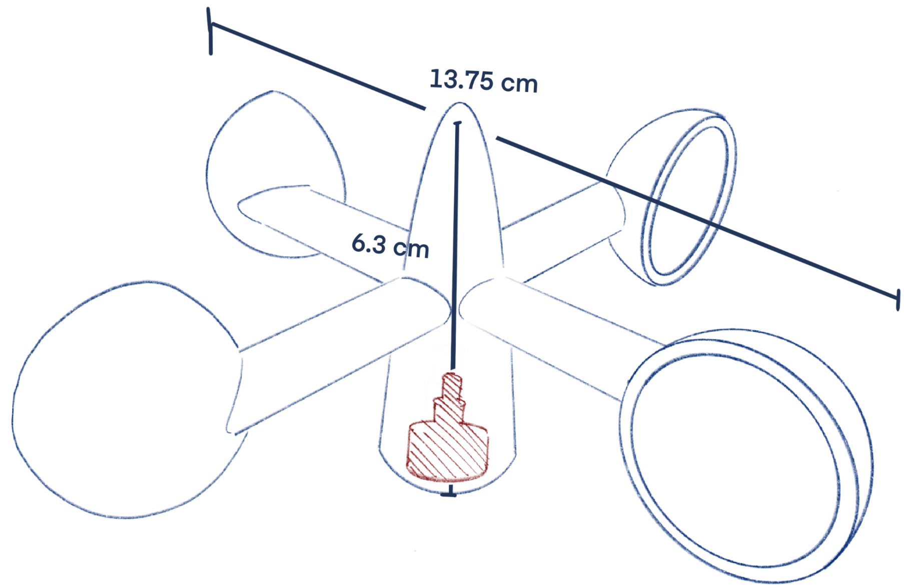


Figure 7-Turbine

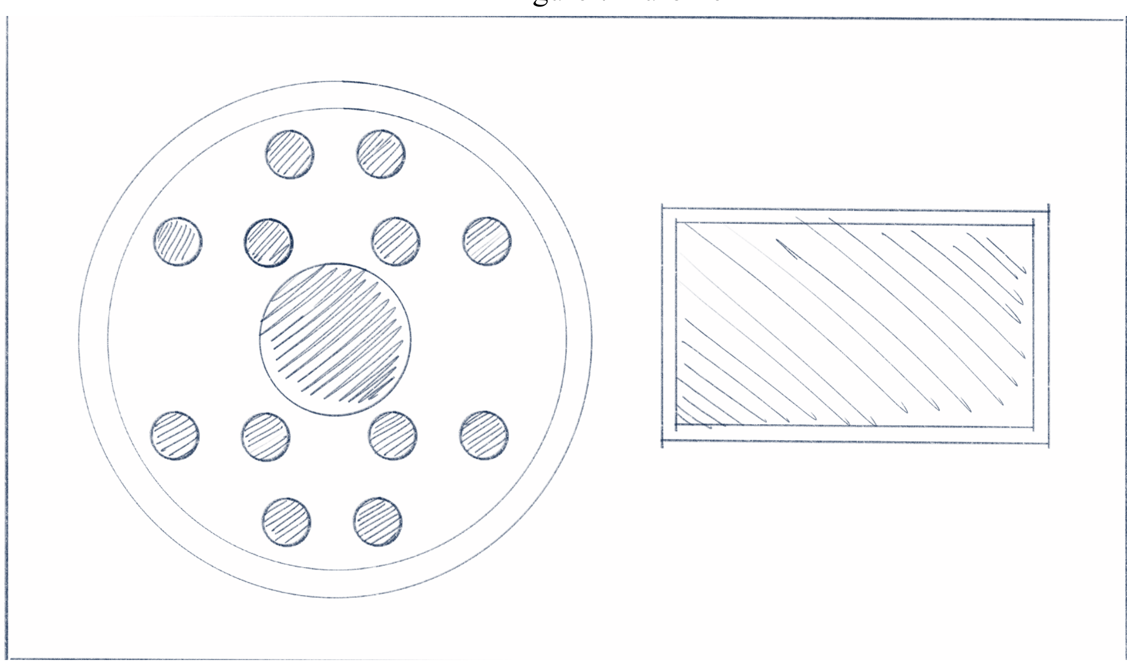


Figure 8- Top lid