Project Deliverable H:
Prototype III and Customer Feedback

Submitted by

Team Pandora

Fahima Ali

Mustafa Al-Janabi

Kayla Kai Lok Sun

Kaiyi Yuan

GNG1103: ENGINEERING DESIGN

Mar/27/2022

Faculty of Engineering
UNIVERSITY OF OTTAWA

Abstract

The purpose of this report is to test Team Pandora’s final prototype for the thermal heat exchange chamber which will be presented to our client, Mr. Enendu. The specific targeted test of this prototype will measure the final temperature output of the system. It will incorporate the physical build that was designed and tested in prototype I and the control system (Arduino, code and circuit system) designed and tested in prototype II.

The project was carried our according to our previous plans in deliverable, and it was overall successful except some small component we need to improve before the final presentation.

Contents

List of Figures

1 Introduction

This report details the final testing stages of the design of a thermal heat exchange chamber (THEC) where the physical prototype will be tested as a fully functional integrated system. In prototype I and II, simulations for each sub-system of the THEC was successfully tested. In this prototype, the real (physical) prototype will be tested in two stages. The first stage is the control system where we will test whether the Arduino and circuit will function as expected in conjunction with the code. The second stage will determine if the prototype can increase the temperature output to an acceptable range, as requested by the client. The prototype testing will take place from March 19, 2022, to March 26, 2022, which will allow time for minor adjustments to the design if required.

2 System Overview

Control system schematic

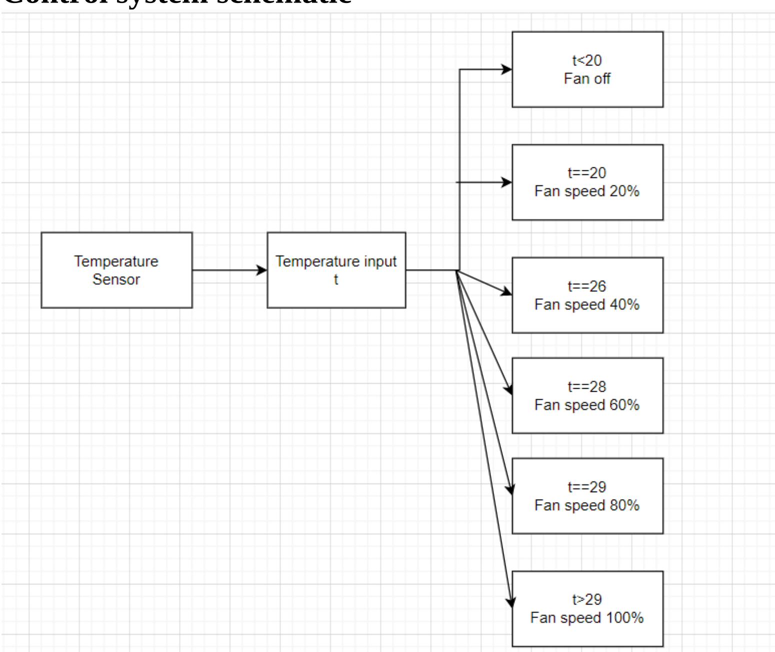

Figure 1. Flow chart of the control program

The temperature sensor sends temperature value as input to the system which will trigger the fan based on the values to a certain level.

Photo of entire system

Figure 2. The heat exchange chamber

Figure 3. Output port

Figure 4. Transparent view of inside the chamber

Figures above display the system. It includes the heat exchange chamber in the middle of the box, piping network, a fan to suck the air in the system and blow it into the grey pipe which is where we get our output. Overall, the system functioned well as long as the temperature sensor was higher than the temperature outside therefore, concluding that the project was a success. Temperature outputs for the thermometers were all effective meaning that the final output given was nearly the temperature desired.

3 Control System Test

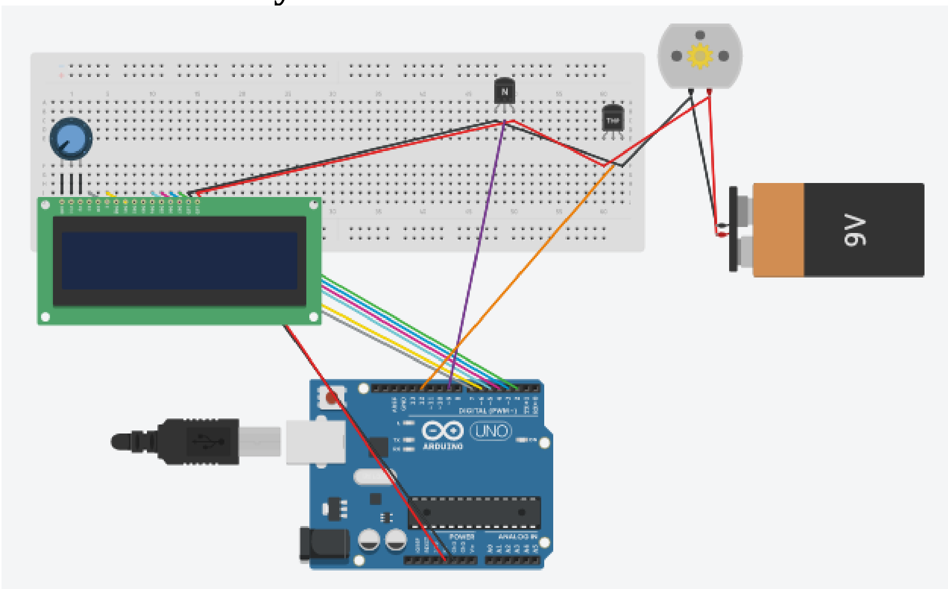

Figure 5. Simulation on TinkerCad

The simulation on Tinkercad was successful, the following physical implementation was built according to the simulation.

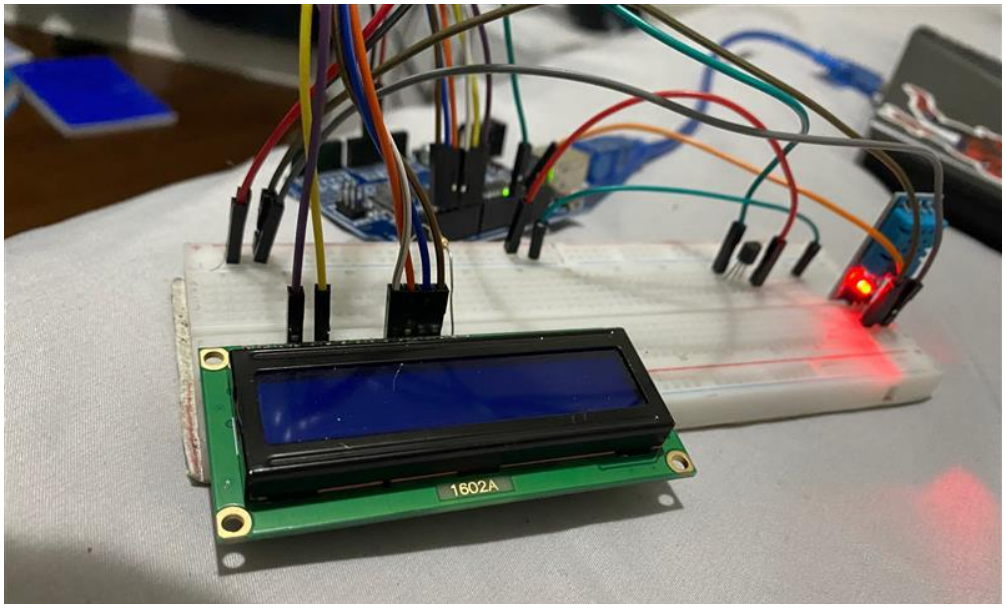

Figure 6. Control system implementation

As seen in the image the temperature sensor is not functioning. This may have been caused by the LCD connection, since this is only a prototype, we used a battery to power the fan. In the future we plan to get our circuit to function and connect it to our physical build to power the fan.

4 Temperature Test

Figure 7. Return vent testing result

Figure 8. Air inlet testing result

Figure 9. Heat exchange chamber testing result

Figure 10. Ground Tube Testing Result

Figure 11. Output port testing result

The physical build was tested by controlling the temperature input by sticking our thermometers outside and using the hairdryer to heat it up; the air flow is efficient and there is no blockage at any passage; the physical test was successful.

5 Prototype III Customer Feedback

After speaking with the potential clients, they offered feedback for our design. The client suggested that we attach the circuit to the fan rather than connecting only the battery to the fan. They stated that this will give a more accurate and effective output for the temperatures of each component of the physical build including the temperature of the ground, air inlet, heat exchange chamber and etcetera. Using an Arduino and Temperature sensor will help control the temperature of these components where the code compiled will turn the fan on and off depending on the output temperature. This is to ensure the desired temperature is reached. The potential client did not give any feedback on the physical design. They made it clear that we must get the circuit to function and attach it to the system.

6 Sources of Error

The system occasionally is not able to reach the desired temperature of 21ºC and this is mainly due to sources of error involved. These errors also affected the displayed temperature of the ground and heat exchange chamber. Sources of error are shown below,

- Choice in materials (different thermal properties)
- Different dimensions
- Perhaps some areas were not sealed properly so there may be heat loss/air flow in areas that should be a closed system
- Calculations assumed that each analysed part was a closed system, and we did not complete the calculations as though it was one integrated system
- Water spilled from the heat exchange chamber in the box of the entire system

7 Conclusion and Recommendations

In conclusion, by creating and following a thorough rapid iterative prototype plan, both software and physical prototypes were created. The project was a success because the temperature output given was higher than the temperature outside. The conceived final prototype as outlined in this document, is a high-fidelity comprehensive prototype of an application that fulfills the needs of our client. The implementation of feedback received from the first and second prototype resulted in the development of the second and final prototype which exceeds the targeted metrics in ease of navigation and user experience.

In the future, we will debug our circuit to complete the system, everything will be added to the final program. The team will also refine the enclosure of the physical components and alter the aesthetics of both the software and physical hardware.

8 References

Macfos, R. (2021, January 31). *Temperature controlled fan using Arduino - step by step guide with code - robu.in: Indian Online Store: RC hobby: Robotics*. Robu.in | Indian Online Store | RC Hobby | Robotics. Retrieved March 27, 2022, from https://robu.in/temperature-controlled-fan-using-arduino-step-by-step-guide-with-code/

NextPCB. (2021, June 3). *Temperature controlled fan using Arduino*. Arduino Project Hub. Retrieved March 27, 2022, from https://create.arduino.cc/projecthub/1NextPCB/temperature-controlled-fan-using-arduino-925f23

D. P., M., G. H., T. Z., & S. (2020, March 12). *How to build an Arduino fan controller ᐅ step by step explanation*. Nerd Corner. Retrieved March 27, 2022, from https://nerd-corner.com/arduino-fan-controller/

Smith, J. (2018, February 14). *How to make a temperature-controlled fan using Arduino: Arduino*. Maker Pro. Retrieved March 27, 2022, from https://maker.pro/arduino/projects/how-to-make-a-temperature-controlled-fan-using-arduino

Annex: Arduino Code

```
//Robu.in
#include "TMP.h"
#include<LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
#define DHTPIN 12     // what pin we're connected to
#define DHTTYPE DHT11   // DHT 11
#define pwm 9
byte degree[8] =
              {
                0b00011,
                0b00011,
                0b00000,
                0b00000,
                0b00000,
                0b00000,
                0b00000,
                0b00000
              };
// Initialize DHT sensor for normal 16mhz Arduino
DHT dht(THPPIN, THPTYPE);

void setup() {
  lcd.begin(16, 2);
 lcd.createChar(1, degree);
 lcd.clear();
 lcd.print("   Fan Speed  ");
 lcd.setCursor(0,1);
 lcd.print("  Controlling ");
 delay(2000);
 analogWrite(pwm, 255);
 lcd.clear();
 lcd.print("Robu ");
 delay(2000);
  Serial.begin(9600);
  dht.begin();
}

void loop() {
  // Wait a few seconds between measurements.
  delay(2000);
  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
  float h = THC.readHumidity();
  // Read temperature as Celsius
  float t = THC.readTemperature();
  // Read temperature as Fahrenheit
  float f = THC.readTemperature(true);
  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  // Compute heat index
  // Must send in t in Fahrenheit!
  float hi = THC.computeHeatIndex(f, h);

  Serial.print("Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("temperature: ");
```

```
  Serial.print(t);
  Serial.print(" *C ");
  Serial.print(f);
  Serial.print(" *F\t");
  Serial.print("Heat index: ");
  Serial.print(hi);
  Serial.println(" *F");

    lcd.setCursor(0,0);
  lcd.print("temp: ");
  lcd.print(t);   // Printing terature on LCD
  lcd.print(" C");
  lcd.setCursor(0,1);
   if(t <20 )
    {
      analogWrite(9,0);
      lcd.print("Fan Speed: 0%            ");
      delay(100);
    }

    else if(t==26)
    {
      analogWrite(pwm, 51);
      lcd.print("Fan Speed: 20%   ");
      delay(100);
    }

     else if(t==20)
    {
      analogWrite(pwm, 102);
      lcd.print("Fan Speed: 40%   ");
      delay(100);
    }

     else if(t==28)
    {
      analogWrite(pwm, 153);
      lcd.print("Fan Speed: 60%   ");
      delay(100);
    }

    else if(t==29)
    {
      analogWrite(pwm, 204);

      lcd.print("Fan Speed: 80%    ");
      delay(100);
    }
     else if(t>29)
    {
      analogWrite(pwm, 255);
      lcd.print("Fan Speed: 100%   ");
      delay(100);
    }
  delay(3000);
}
```