Equipment Cabinet Corrosion Prevention System

By

Andrew Miller

Andrew Schmitt

Kevin Lee

Final Report for ECE 445, Senior Design, Fall 2020

TA: Shaoyu Meng

9 December 2020

Project No. 17

Abstract

The Equipment Cabinet Corrosion Prevention System is an autonomous device that regulates the cabinet environment to prevent condensation (which can cause corrosion). By taking in humidity temperature data both inside and outside the cabinet, our system decides whether to run a dehumidifier or heater. Additionally, it stores the data to a microsd card which can be translated to a MATLAB graphical format on the computer so that cabinet conditions can be monitored over time. We found that the system accurately powers the correct device given the input sensor data, and is able to write to a microsd card that translates properly to a graph.With regard to long term testing, the system was able to remain autonomous for one week and would be able to continue for longer.

Contents

1. Introduction

1.1 Problem and Background Overview

The utility Ameren Illinois is interested in reducing corrosion, specifically on spare electrical cabinets in isolated areas. These cabinets are not in climate controlled environments, and they do not have access to a low voltage power supply. The central need is to have an off grid, autonomous system that regulates the environment of these cabinets to prevent corrosion on exposed electrical terminals.

Electrical cabinets are an inherent part of any substation, and they are prone to corrosion. Specifically, oxide corrosion, or rust, is an electrochemical reaction that occurs when water is exposed to certain kinds of metal [1]. Most metals can experience surface corrosion at humidies above 80%[2]. This commonplace process puts a heavy financial burden on electrical utilities. A study done by the National Association of Corrosion Engineers estimated that in 2013, utilities spent nearly $27.7 billion to correct corrosion related maintenance [3]. The process of corrosion is accelerated by humid conditions, when water can more easily condense on electrical equipment. Preventing corrosion is important from both financial and reliability standpoints, as the Electrical Power Research Institute of the United States estimates that "more than half of all unplanned power outages are due to corrosion" [3]. Therefore, preventing corrosion is a worthwhile endeavor, even on these spares.

Electric heaters prevent corrosion by decreasing the relative humidity, meaning that the air can hold more moisture without condensing. However, their ability to do this is hampered at higher temperatures in the summer when hot, humid air cannot be effectively made hotter. Therefore, the incorporation of a dehumidifying system that exchanges control with the heater is necessary to be able to take moisture out of the air at higher temperatures.

1.2 High-Level Block Diagram

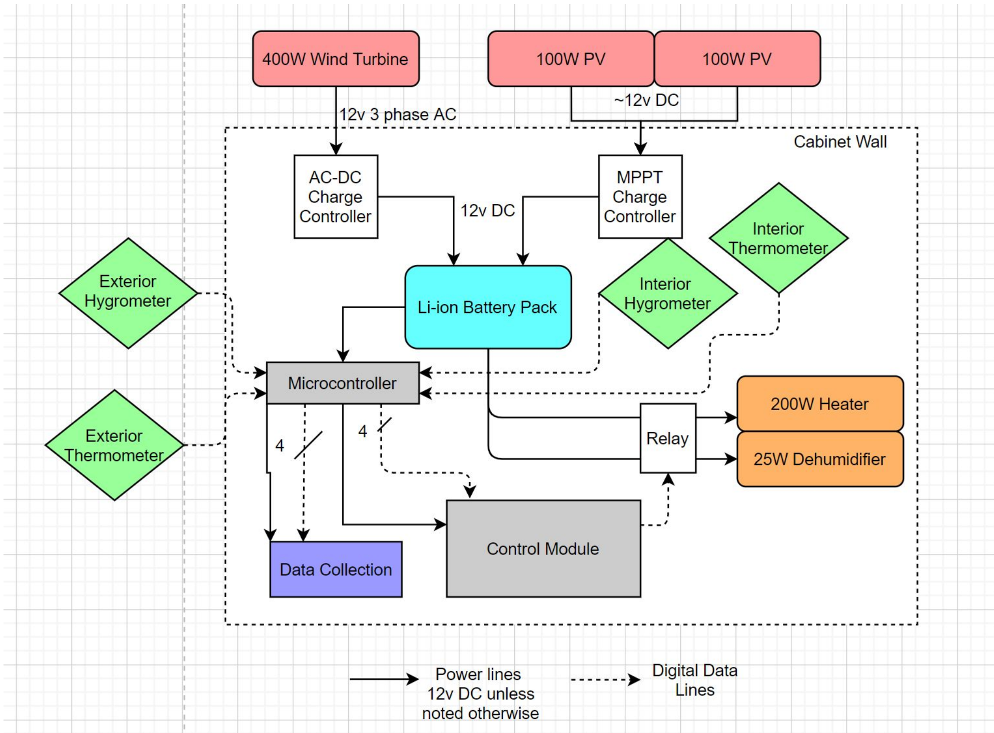

Figure 1: Corrosion prevention system high-level block diagram

The high-level block diagram in Figure 1 shows an overview of our solution. Input power from the sources shaded in red charge a Li-ion battery pack. This battery pack powers both the heater and dehumidifier as well as our microcontroller. The microcontroller takes in humidity and temperature sensor data from inside and outside the cabinet to send to our control module. The control module is a finite state machine that determines whether to run the dehumidifier, heater, or neither given the conditions. Finally, the data collection block in purple represents the microSD card which functions to record sensor data periodically, enabling an Ameren worker to be able to monitor the conditions over time.

1.3 High-Level Requirements List

- The electric heater must keep all condensation off of exposed electrical terminals for at least a five-day period (ideally indefinitely), keeping humidity below 80%.
- The battery must have the capacity to run the dehumidifier and electric heater each for three hours a day for two days to account for a hypothetical situation of no wind and no sun for two days. This is 1350 Wh or 112.5 Ah on a 12 V battery.
- The data collection process must obtain and store temperature and humidity data both inside and outside of the cabinet to the precision of the tenths place once per hour (to be viewed by an Ameren worker at a later time).

2. Design Procedure

2.1 Sensor Inputs and Logic

A total of five sensors were utilized in this design. The following sensors provide data to our microcontroller to process:

1) Voltage sensor — measures voltage of battery.
2) Thermometer (T_in) — measures ambient temperature inside of the electrical cabinet.
3) Thermometer (T_out) — measures ambient temperature outside of the electrical cabinet.
4) Hygrometer (H_in) — measures ambient relative humidity inside of the electrical cabinet.
5) Hygrometer (H_out) — measures ambient relative humidity outside of the electrical cabinet.

The thermometer and hygrometer came fitted into one physical device, the AM2303 (also known as DHT22). Two AM2303s were used, one for outside the electrical cabinet and one for inside the electrical cabinet. The sensor devices were able to measure temperatures from -40 ℃ to 80 ℃ within 0.5 ℃ accuracy and relative humidity from 0% to 100% at 2% to 5% accuracy.The voltage sensor used was the GR0430X10. This device measures DC voltage from 0.02445 V to 25 V.

These sensors were chosen for their compactness, efficiency, and compatibility with our microcontroller. Figure 2 below depicts the thermometer and hygrometer. Figure 3 below depicts the voltage sensor.

Figure 2: DHT22/AM2302 temperature and humidity sensor

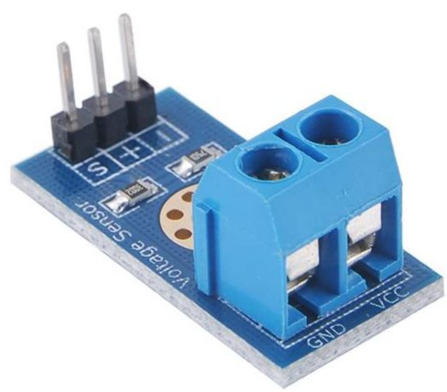

Figure 3: GR0430X10 *Voltage sensor*

2.2 Control Module

For the main control system, we started by creating a flow chart of what decisions needed to be made, as shown in Figure 4. This includes checking for battery power, the comparative temperature, the relative humidity, and the operating condition for the dehumidifier (surrounding temperature above 60).

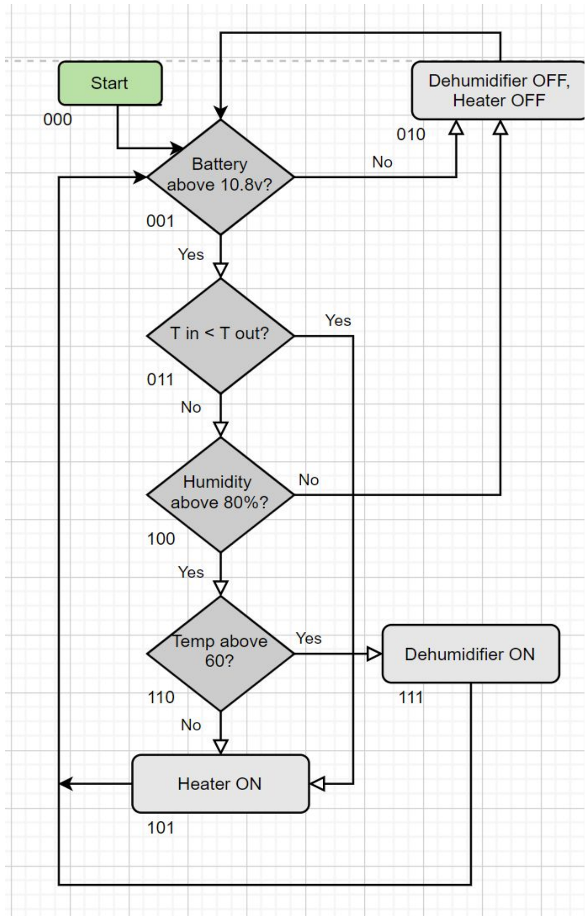

Figure 4: Control Module Flow Chart

The flow chart lent itself well to the creation of a finite state machine; the bubbles just required state numbers, as shown in Figure 4. Special consideration was made to these numbers such that the first two bits of every decision stage was unique. This allowed the inputs B, C, H and T, which are the decision making factors at various stages, to be prefiltered by a 4:1 mux. The mux takes in the first two bits from the current state, "x" and "y", and uses them to select one of the input signals as the general input "I" bit to be used by the finite state machine. This consideration reduced the number of inputs in the Karneugh maps during design, and therefore significantly simplified the next state logic expressions. These were then implemented in hardware using NAND gates and verified to work on a breadboard circuit before soldering to PCB.

2.3 Outputs

The microcontroller will control two electrical relays that connect power from the Lithium-ion battery to the heater and dehumidifier. The relay used for our project was the HiLetgo SRD-05CDC-SL-C. It was chosen for its space efficiency and compatibility with our system and microcontroller. It has a rating of DC 30 V at 10 A making it capable of powering the heater and dehumidifier with the Lithium-ion battery. For testing of this project though, we used a variable DC voltage source in a controlled lab environment. Pictured below in Figure 5 is the relay.

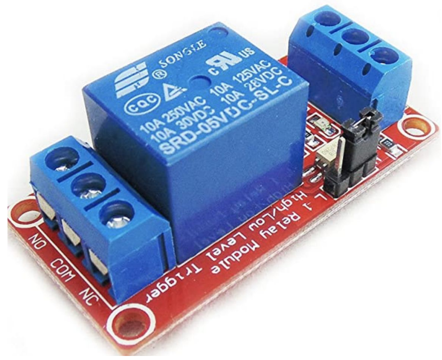

Figure 5: Electrical relay

2.4 Reset Switches

We chose to include a reset switch for the finite state machine in case an issue arose with it, especially during testing, where we would like to ground the current state variables to bring it back to the start of the control module. Additionally, we included a reset switch for the microcontroller. This was necessary because after taking the microsd card out of the system and putting it back in, the system needs to restart the serial connection between the microcontroller and microsd card module.

2.5 Data Storage

We incorporated the HiLetgo six pin microsd card module because of its included 5 V to 3.3 V converter that is necessary for using the microsd card. Additionally, the slot is helpful to be able to easily insert and remove the card from the system.

Figure 6: Microsd card module

We chose to store the data on the microsd card in the following way for each "capture" of data (without commas between values).

inside temperature, *outside temperature*, *inside humidity*, *outside humidity*

This was done such that our MATLAB program can take in the data and convert it to a graphical representation.

3. Design Details

3.1 PCB

Pictured in Figure 7 is our PCB with all components labeled.

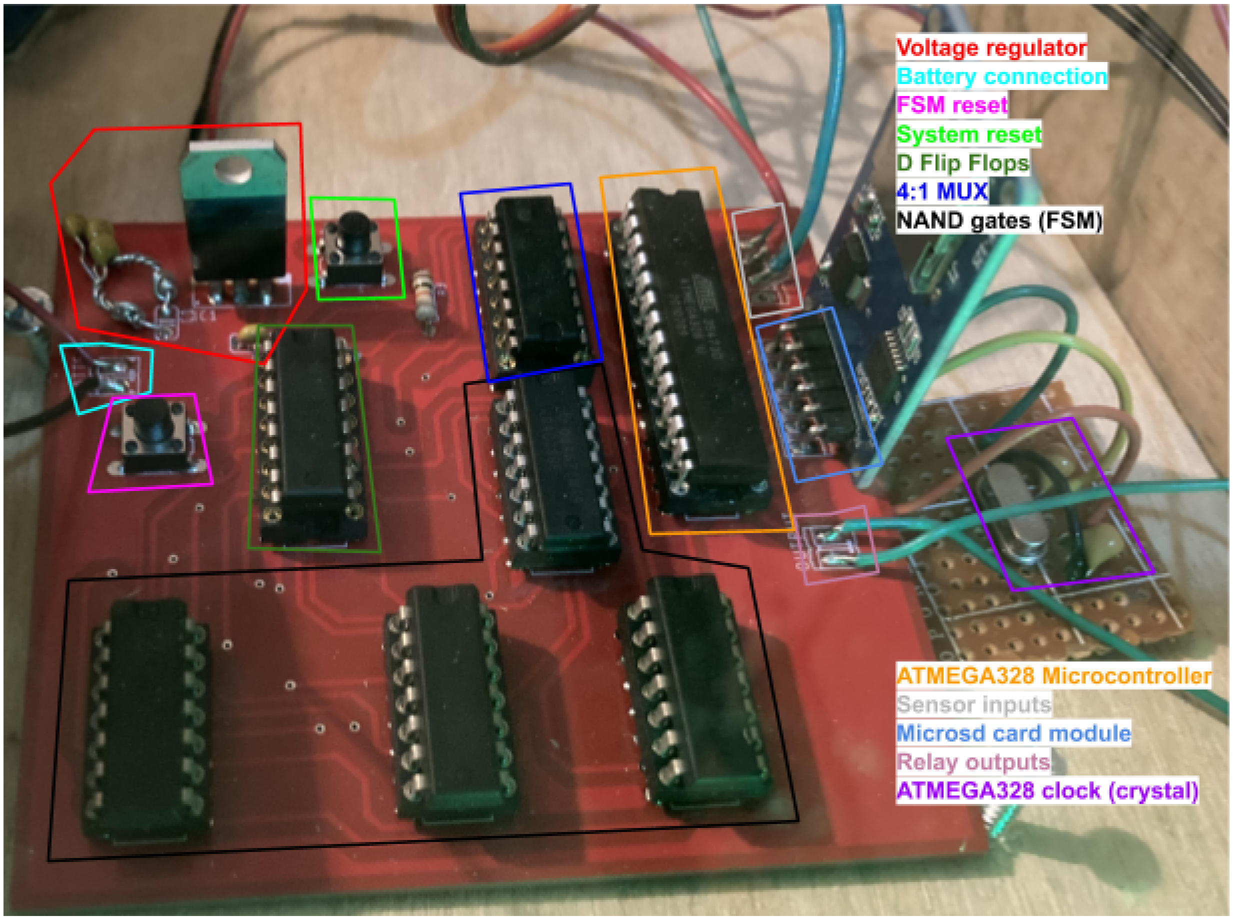

Figure 7: Corrosion prevention system PCB layout

3.2 Power Considerations

For the finite machine state machine, we calculated the value of 10.8 V as the battery voltage threshold for which we will allow the dehumidifier or heater to run. Below this voltage level, the system will be in “conserve power” mode, meaning that the system will only power the bare minimum to keep itself alive (and retain autonomy). The calculations we performed are below.

To begin, we wanted to determine the power needed to run the bare minimum of our system. This includes the CMOS chips, microcontroller, microsd card module, voltage regulator and sensors. For the CMOS chips, we relied on equations 1, 2, and 3 to determine their dissipated power [4].

$$P_{total} = P_{static} + P_{dynamic} \quad (1)$$

$$P_{static} = I_{static} * V_{cc} \quad (2)$$

$$P_{dynamic} = (C_L + C)V_{cc}^2 fN^3 \quad (3)$$

In eq. 3, C_L is the assumed load capacitance (which we assume a high value of 70 pF), C is the internal capacitance, f is the switching frequency of 1 Hz for our design, and N is the number of switching bits (assumed to be the number of transistors). Also noting that $V_{cc} = 5\ V$, we calculated the total power consumption of the three CD4023BE ICs using their known parameters on the datasheet [5]. .

Part	# of chips	I_{static}	C	N	P_{total}
CD4023BE (3 input NAND)	3	10^{-5} uA	50 pF	18	0.0525 mW

Table 1: CD4023BE power consumption calculation

Because of such a small value, we did not do the calculations for the 4:1 MUX and D flip flops and assumed that in total the IC chips use a generous 0.5 mW. The other calculations of dissipated power are in Table 1.

Part	Estimated Power Consumption
ATMEGA328 microcontroller	60 mW (running at 5 V, low clock speed, internal crystal oscillator frequency of 8MHz) [17]
MicroSD card module	$P_{dissipated} = 5\ V * 80\ mA\ = 400\,\text{mW}$ [5]
L7805CV Voltage Regulator	$P_{dissipated} = (12-5)V * (\frac{.060}{5} + .080)\ A = 644\,\text{mW}$
Two DHT22/AM2302 Sensors	$P_{dissipated} = 2 * 5\ V * 0.0025\ A = 25\,\text{mW}$ [6]
All of the IC chips	0.5 mW

Table 2: Power consumption calculations for the system (excluding dehumidifier and heater)

After summing the estimated values, we found the system power consumption to be approximately $P_{system} = 1.2\ W$. To be on the safe side and account for an unrealistic scenario that there is no sun or wind power for two days, we estimated a spare 57.6 Wh is needed to keep the system alive. This is approximately 3.46% of the 85% capacity battery. Looking at an approximate discharge curve for a 12 V Lithium Iron Phosphate battery at 25 ℃ in Figure 1, we determined a cut-off [7]. The quadratic dropoff begins roughly at 12.5 V at 20% capacity and goes down to 10 V at 1% capacity. This as a quadratic equation is:

$$y = -69.3(x-0.2)^2 + 12.5 \qquad (4)$$

Therefore, at 4.46% percent capacity, we found the voltage to be 10.8 V, which should be the cut-off for which the system conserves power without running the heater or dehumidifier.

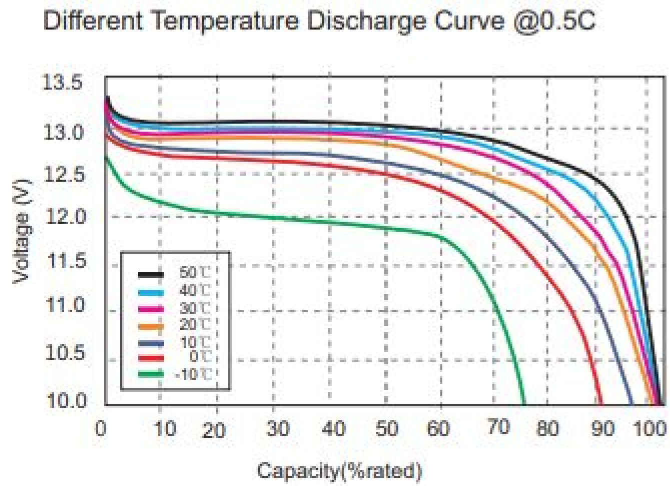

Figure 8: Discharge curve for a 12 V lithium iron phosphate battery [7]

3.3 Sensor Inputs and Logic

The microcontroller (ATMEGA328) processes the signals from our five sensors to output four bits that will henceforth be referred to as B, C, H, and T. These bits B, C, H, and T feed back into the finite state machine control module as inputs to transition from one state to another. Code regarding the data processing was first prototyped on an Arduino UNO before being flashed to a standalone ATMEGA chip.

Measurements from the sensors were read at a sampling rate of 0.5 Hz. As such, the clock cycle for the finite state machine was set to 0.5 Hz. Note that the outdoor hygrometer serves no purpose regarding device functionality. The outdoor hygrometer is solely for data collection purposes.

Table 3 below outlines the conditions for output bits B, C, H, and T.

Sensor Data Condition	Resulting Variable Logic
Battery voltage above 10.8 V	B = 1
Inside temperature less than outside temperature (by at least 3℉)	C = 1
Inside humidity greater than 80%	H = 1
Inside temperature greater than 60 ℉	T = 1

Table 3: Sensor data interpretation

The code snippet that reads data from the sensors can be seen in Figure 9 below. The 0.5Hz clock cycle can also be seen.

```
void loop() {
  //set clock cycle

  digitalWrite(CLOCK_PIN, HIGH);
  delay(1000);
  digitalWrite(CLOCK_PIN, LOW);
  delay(1000);

  //read battery voltage
  float voltageSensorVal = analogRead(VOLTAGE_SENSOR_PIN);
  float v = (voltageSensorVal / 1024) * 25;

  //read internal thermo/hygro
  float h_in = sensor_in.readHumidity();
  float t_in = sensor_in.readTemperature(true); //true == farenheit

  //read external thermo/hygro
  float t_out = sensor_out.readTemperature(true);
  float h_out = sensor_out.readHumidity();
```

Figure 9: Sensor data processing codeAppendix

An oscilloscope in a lab environment was used to calibrate and verify the accuracy and precision of the battery voltage sensor. A variable DC voltage source was used to simulate various battery conditions. Ice cubes and body temperature were used to ensure the microcontroller correctly processed various changes in ambient climate. To verify bits B, C, H, and T, LEDs were connected to the microcontroller pins in order to verify correct output of either 1 (high) or 0 (low) for each output bit. Details outlining the requirements and verification procedures can be found in Appendix A.

3.4 Outputs

The microcontroller was also used to control the outputs turning on and off the heater and dehumidifier. These "out" bits Q_0 and Q_1 were used to represent the status of the dehumidifier and heater being on and off. "1" designating the respective device being turned on and "0" designating it being turned off. In various states of the finite state machine, Q_0 and Q_1 may be "xx" meaning that Q_0 and Q_1 can be either a one or a zero without affecting functionality of switching states in the state machine. However, in reality we do not want these Q_0 and Q_1 bits to be shifting. We want to maintain the Q_0 and Q_1 bit status until the finite state machine reaches a state that explicitly sets Q_0 and Q_1 otherwise. Because of this requirement, the microcontroller is coded to ignore states from the finite state machine in which Q_0 and Q_1 correspond to "xx."

The "Out" bits Q_0 and Q_1, as indicated in Appendix B, represent whether the dehumidifier and heater are turned on or off. In a typical finite state machine, "xx" represents "don't care bits" meaning these bits can be either a 1 or a 0 without affecting state machine functionality. However, in our design we need the "Out" states to remain consistent when switching to a state where the out bits are "xx." To accomplish this, the three bits representing the current state will be connected to the microcontroller that will provide a digital output to a relay turning the dehumidifier and heater on and off. Based on the current state, the microcontroller will be programmed to ignore the current states corresponding to “xx.” This way, the state of the dehumidifier and heater will not change when changing states unless explicitly needed.

In order to test the microcontroller was able to detect current state bits x, y, and z from the finite state machine, system print statements were outputted to a terminal. LEDs built into the relay indicated whether the relay was receiving power and when the relay switched signals. Using these signal LEDs, we were able to verify that the relays were able to switch on and off the heater and dehumidifier. In order to test the relay subsystem as a standalone, buttons were used to simulate different states form the finite state machines. Details outlining verification procedures can be found in Appendix A.

Below in Figure 10 is a code snippet illustrating functionality of outputs.

```
int x,y,z;
x = digitalRead(BIT_X_PIN);
y = digitalRead(BIT_Y_PIN);
z = digitalRead(BIT_Z_PIN);

//decide what device to turn on based on xyz bits
//000 & 010 -> 00
if ( ((x == LOW) && (y == LOW) && (z == LOW)) || ((x == LOW) && (y == HIGH) && (z == LOW)) ){
  digitalWrite(RELAY_DEHUM_Q0_PIN, LOW);
  digitalWrite(RELAY_HEATER_Q1_PIN, LOW);

}

//101 -> 01
if((x == HIGH) && (y == LOW) && (z == HIGH)){
  digitalWrite(RELAY_DEHUM_Q0_PIN, LOW);
  digitalWrite(RELAY_HEATER_Q1_PIN, HIGH);

}

//111 -> 10
if((x == HIGH) && (y == HIGH) && (z == HIGH)){
  digitalWrite(RELAY_DEHUM_Q0_PIN, HIGH);
  digitalWrite(RELAY_HEATER_Q1_PIN, LOW);

}
```

Figure 10: Relay output code

3.5 Data Plotting

The code in Figure 11 is what MATLAB executes to graph our data file. It interprets each column as the appropriate variable and then graphs them on a two axis graph.

```
close
clear
clc

filename = '\\client\d$\data.txt'; %assumes "D" drive for microsd
data = dlmread(filename, '', 0, 0);
Tin = data(:,1);
Tout = data(:,2);
Hin = data(:,3);
Hout = data(:,4);

yyaxis left
plot(Tin, '-b');
hold on
plot(Tout, '-g');
hold on
ylabel ('Temperature F')
xlabel ('Data Points')
ylim([-5 100])
yyaxis right
ylabel ('Humidity %')
plot(Hin, '-r');
hold on
plot(Hout, '-k');
hold on
title('Temperature and Humidity over time')
legend('Temperature Inside','Temperature Outside','Humidity Inside','Humidity Outside')
```

Figure 11: MATLAB code for interpreting sensor data

3.6 Schematics

We include the schematics in Figures 12 and 13 to highlight the specific connections on the PCB. As a brief overview, battery voltage is input at 12 V and converted down to 5 V to power the microcontroller and IC chips. The microcontroller takes in sensor data and outputs the digital variables B, C, H, and T to a 4:1 MUX which decides which variable to choose from based on the current state as the select bits. This bit, “I,” is an input to our finite state machine consisting of 3 D flip flops and NAND gates. The current state is fed back into the microcontroller which then outputs to the relays and also writes to the microsd card.

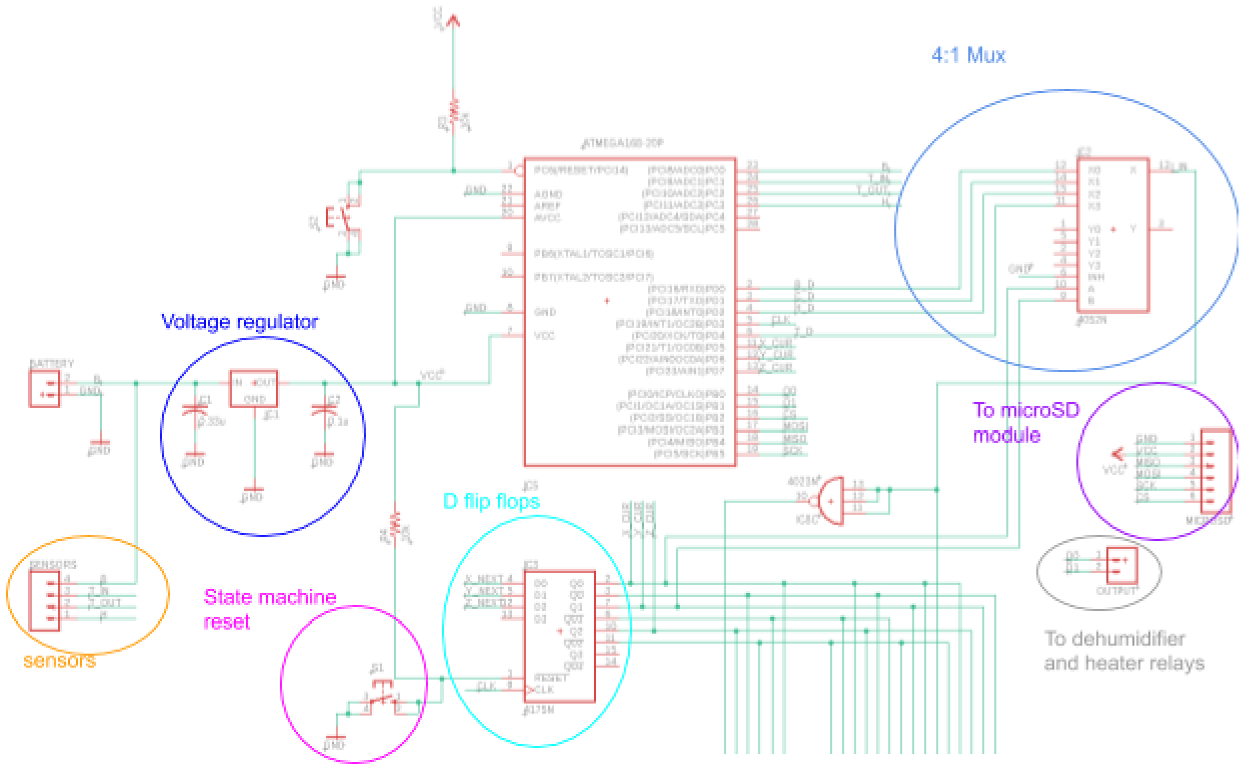

Figure 12: Schematic showing the inputs and outputs of the microcontroller

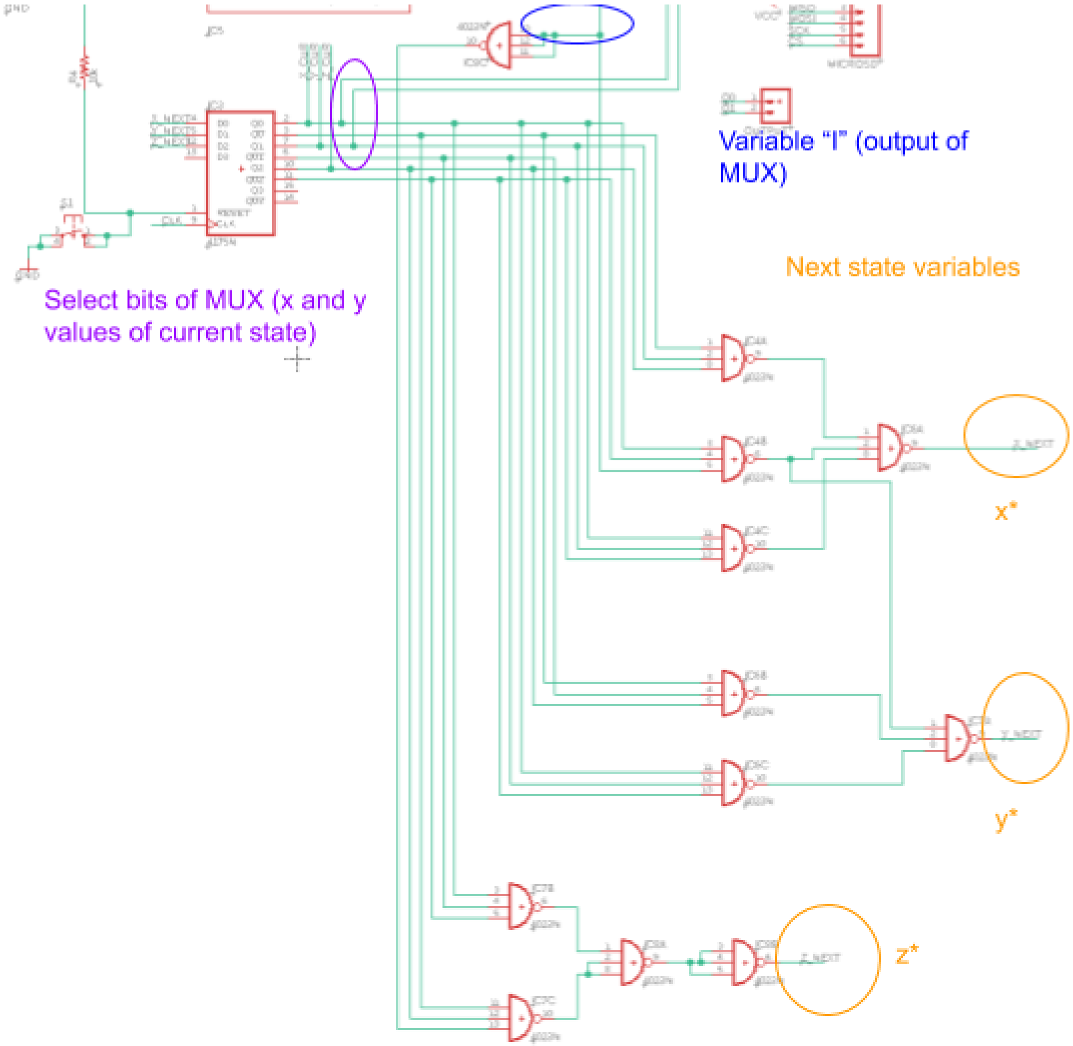

Figure 13: Schematic showing the control module state machine

4. Verification

4.1 Data Plotting

Presented in Figure 14 is a sample data file obtained from our microsd card and its corresponding MATLAB graph. The left axis of the graph is for temperature (the green and blue curves) and the right axid is for plotting humidity (the red and black curve). It is important to note that this data was sampled at a high rate of 0.5 Hz where in reality we would only sample twice per hour. Additionally, the curves on the right were obtained during extreme testing conditions of rapid cooling or humidifying and would in reality we would not expect the curves to be as drastic.

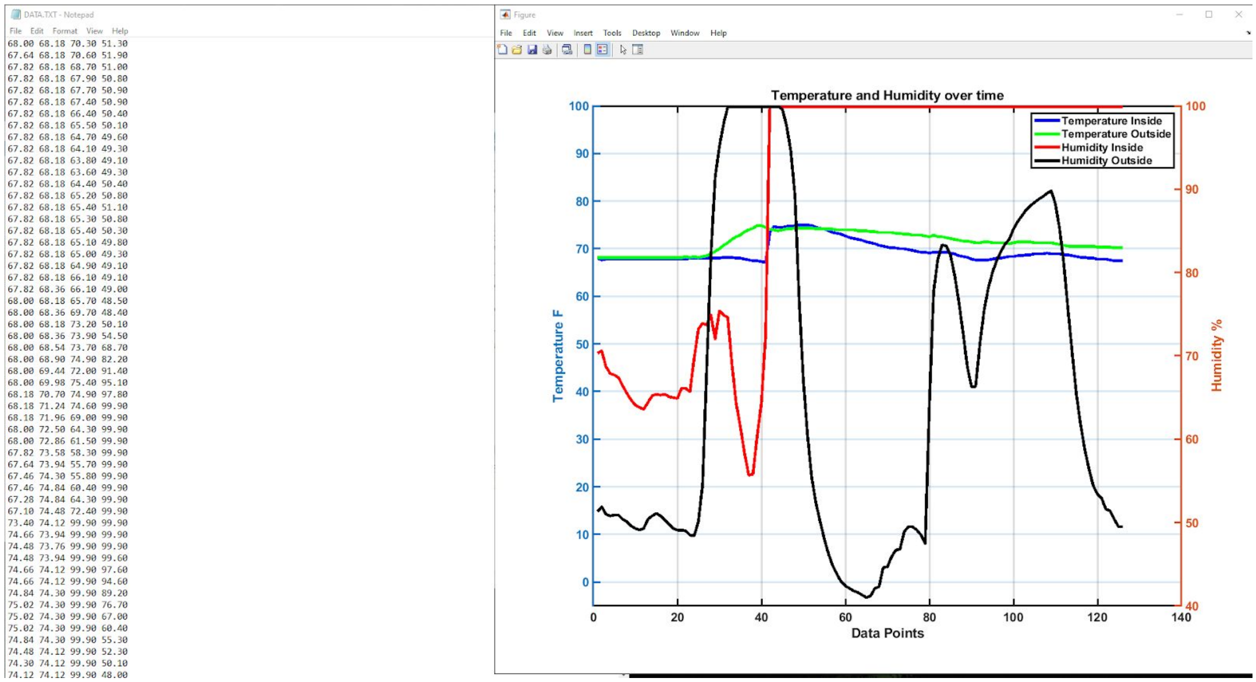

Figure 14: Data in MATLAB example plot

4.1 Finite State Machine

Table 4 is a portion of data obtained from our testing that highlights part of the completely working finite state machine. A screenshot of a larger part of this data.txt file is found in Appendix B. We can verify in the first two rows that B, C, H, and T variables are correct in value based on the logic outlined in Table 3. Additionally, we note a correct state transition from "011" to "101" and the appropriate powering of the heater relay. The black bar signifies the skipping of some rows in the file to then focus on the last four. These again display the correct B, C, H, and T values and have correct state transitions. Because C is now "0," the humidity condition of greater or less than 80% and temperature above or below 60℉ are checked and a result the heater is appropriately switched off and the dehumidifier on.

BCHT	Current state	Dehumidifier	Heater	Battery [V]	T_{in} , T_{out} [℉]	H_{in} , H_{out} [%]
1101	011	OFF	OFF	11.94	68.18, 71.24	74.60, 99.90
1101	101	OFF	ON	11.94	68.18, 71.96	69.00, 99.90
1011	100	OFF	ON	12.08	74.48, 73.94	99.90, 97.60
1011	110	OFF	ON	12.08	74.66, 74.12	99.90, 97.60
1011	111	ON	OFF	12.08	74.66, 74.12	99.90, 94.60
1011	001	ON	OFF	12.08	74.84, 74.30	99.90, 89.20

Table 4: Data portion from testing

5. Costs

For calculating our labor cost, we assume an hourly rate of $35/hr per engineer. This project involved three engineers working around 10 hours per week for 10 weeks from when the project was proposed to its demonstration. Therefore, we calculate:

$$\frac{\$35}{hour} * \frac{8\ hours}{week} * 10\ weeks * 3\ engineers = \$8{,}400$$

Additionally, we anticipate the machine shop taking 10 hours to create the two revisions of our PCB.

$$\frac{\$35}{hour} * 10\ hours = \$350$$

Part #	Description	Manufacturer	Quantity	Cost
ATMEGA328-PU-ND	ATMEGA328 Microcontroller	Microchip Technology	1	$1.90
DHT22/AM2302	Temperature + Humidity Sensor	HiLetgo	2	$11.99
GR0430X10	Voltage Sensor	Geekstory	1	$0.86
SRD-05CDC-SL-C	Relay	HiLetgo	2	$5.97
887-2015-ND	16MHz Crystal	TXC CORPORATION	1	$0.30
COM-15539	20pF Capacitor	Sparkfun	2	$0.20
10k Ohm Resistor	N/A	Sparkfun	2	$0.10
6 Pin MicroSD Card Module	MicroSD breakout board with 5V to 3.3V converter included	HiLetgo	1	$2.00
CD4023BE	3 input NAND	Texas Instruments	3	$2.00
CD40175B	quad D flip flop	Texas Instruments	1	$0.50
CD4052BC	dual 4:1 MUX	Texas Instruments	1	$0.50
Tact Button	Momentary tact button	FUNMANY	2	$0.20
COM-15515	0.33uF Ceramic Capacitor	Sparkfun	1	$0.10
COM-15507	0.1uF Ceramic Capacitor	Sparkfun	1	$0.10
L7805CV Linear Voltage Regulator	511-L7805CV	STMicroelectronics	1	$0.50
Hauture 600ml	Mini Dehumidifier	Hauture	1	$31.99

Rover Elite 20A MPPT	Solar Charge Controller	Renogy *note: may not actually buy	1	$99.99
SANDISK 4GB Micro SDHC Memory Card	MicroSD card	SANDISK	1	$8.37

PARTS TOTAL:	Electronics: $34.37 Whole: $175.09

GRAND TOTAL (parts and labor):	$8,925.09

Table 5: Parts and total cost

6. Conclusion

6.1 Summary

The Equipment Cabinet Corrosion Prevention System accurately regulates the internal humidity and temperature of a cabinet, thereby preventing corrosion. The sensor data is digitized and fed into a working finite state machine that decides whether to power a corresponding relay to a heater or dehumidifier. Finally, the sensor data is stored properly on a microsd card and translates well to a graphical format in MATLAB.

6.2 Ethical Considerations

Our primary ethical consideration in our project is ensuring transparency with regards to the specification of our product to Ameren. In accordance with section 7.8.1 and section 7.8.9 of the IEEE Code of Ethics [8] as well as section 1.2 of the ACM Code of Ethics [9], we present a clear depiction and description of our project and its primary purpose. Given the full development of this project beyond its prototyping proof-of-concept stage, this product very well may be used on spare electrical cabinets that serve as backup equipment for maintaining critical infrastructure. As such, these components may be relevant in maintaining public safety, hence the need to communicate product specifications and operational capabilities precisely and accurately.

The primary safety factor in developing our corrosion prevention system was interacting with high powered electrical devices, namely the lithium ion battery pack. To adhere with section 7.8.9 and section 7.8.6 of the IEEE Code of Ethics [8], two persons were required to be on site at all times when the risk of electrical shock was present. When possible, sources of power were disconnected when modifying devices and components within the electrical system.

In regard to initial concerns regarding malicious cyber attacks that may arise with the remote transmission of data, this project ultimately works only in offline conditions. Data recording only occurs locally and requires physical on-site interaction in order to collect. This decision was made to be in accordance with section 7.8.1 and 7.8.9 of the IEEE Code of Ethics [8] and section 2.9 of the ACM Code of Ethics [9]. Data storage system has zero effect on the functionality of our device. Therefore, in the event of an error with data recording or a malicious cyber attack, the functionality of our device will not be compromised.

6.3 Improvements

This project was only ever tested in a lab environment with forced simulated weather conditions. There is sure to be a great deal of improvements to be made in optimizing the system. For example fine tuning the dead battery level voltage which would shut everything off, the temperature difference buffer between the two sensors which would trigger the heater, and the humidity level at which the dehumidifier is really needed. Also, much more research could be done on how much climate control is really needed to keep the cabinets good enough; the goal being to reduce power demand and therefore expenses in battery capacity and generation devices.

Another improvement that would make our product significantly more user friendly is wireless real time data transfer. Then a field worker would not need to view the system in person to read the micro SD chip, and labor expenses would reduce. Given that the places this system would be used will not be in wifi range, this would require some sort of cellular or satellite form of communication. This could prove to be more expensive than it is worth, but we will not know unless we draft a design for it.

References

[1] Uphoff, 2020. *Corrosion Resistant Coatings For Various Types Of Corrosion*. [online] Thomasnet.com. Available at: <https://www.thomasnet.com/articles/chemicals/corrosion-resistant-coatings/> [Accessed 17 September 2020].

[2] Workshopcompanion.com. 2009. *Preventing And Removing Rust*. [online] Available at: <http://workshopcompanion.com/Demos/Sawdust_Sessions/Rust_Removal_Classroom/Rust_Removal_Classroom_files/Shop_Notes_Rust_Prevention.pdf> [Accessed 17 September 2020].

[3] Taylor, J., 2017. *Corrosion Reduction In Power Distribution.*. [online] Eaton.com. Available at: <https://www.eaton.com/content/dam/eaton/markets/oil-and-gas/knowledge-center/whitepaper/Corrosion-reduction-in-power-distribution.pdf> [Accessed 15 September 2020].

[4] Texas Instruments. 2012. *CMOS Power Consumption And Cpd Calculation*. [online] Available at: <https://www.ti.com/lit/an/scaa035b/scaa035b.pdf> [Accessed 30 September 2020].

[5] Texas Instruments. 2003. *CD4023B*. [online] Available at: <https://www.ti.com/lit/ds/symlink/cd4011b.pdf> [Accessed 30 September 2020].

[6] Cdn-learn.adafruit.com. 2020. *DHT11, DHT22 And AM2302 Sensors*. [online] Available at: <https://cdn-learn.adafruit.com/downloads/pdf/dht.pdf> [Accessed 30 September 2020].

[7] Batteryspace.com. n.d. *Lithium Iron Phosphate (Lifepo4) Battery*. [online] Available at: <https://www.batteryspace.com/prod-specs/9055.pdf> [Accessed 30 September 2020].

[8] IEEE, "IEEE Code of Ethics," IEEE, 2020. [Online]. Available:https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed 28 September 2020].

[9] ACM, "ACM Code of Ethics," ACM, 2018. [Online]. Available:https://www.acm.org/code-of-ethics. [Accessed 28 September 2020].

Appendix A Requirements and Verification

Requirements	Verification	Result
1. Voltage sensor accurately measures battery voltage to within 0.1 V. 2. Thermometers accurately measure ambient temperature to within one degree Fahrenheit at a rate no slower than once every two seconds. 3. Hygrometers accurately measure relative humidity to within 5% at a rate no slower than once every two seconds. 4. Code for data processing correctly produces desired digital outputs B, C, H, and T every two seconds.	1. Use an oscilloscope to apply voltage to the sensor and print readout to a computer screen. Check voltage of battery with separate voltmeter at 100% charge, 50% charge, and 5% charge. 2. Set data sampling rate to two seconds, and use a heating element and ice cube to change ambient temperature and print recording to screen. 3. Set data sampling rate to two seconds, and use a humidifier to change relative humidity and print recording to screen. 4. Use LED to verify the correct output of either 1 (high) or 0 (low) for each output bit.	1. Yes 2. Yes 3. Yes 4. Yes

Table 6: Requirements and verifications for sensor input logic

Requirements	Verification	
1. Follows the flow chart and properly maps all 8 states. 2. Properly responds to all inputs. 3. Can translate current state into output. 4. Has the availability of a manual reset.	1. Using an oscilloscope, the FSM states were tracked and verified to follow the cycle correctly through all states. 2. The output of the mux was measured with an oscilloscope side by side the current states and verified to output the correct input signals 3. The microcontroller receives three unique current states signifying different output commands. 4. A push-button switch which sets the flip-flop inputs to ground resets the FSM back into the start state.	1. Yes 2. Yes 3. Yes 4. Yes

Table 7: Requirements and verification for finite state machine

Requirements	Verification	Result
1. Microcontroller is able to detect the current state. 2. Microcontroller is able to control the on/off status of the heater and dehumidifier. 3. Microcontroller turns on and off the heater and dehumidifier at states 000, 010, 101, and 111 based on Table 5. 4. Microcontroller maintains previous status of the heater and dehumidifier in all other states with bits Q_0 Q_1 at xx. 5. Microcontroller has reset switch	1. Print outputs to a computer screen to verify microchip is able to detect high/low signals. 2. Connect relays to microcontroller and heater and dehumidifier and verify that a high and low output signal from microcontroller turns on and off respectively the heater and dehumidifier. 3. Use switches or buttons to simulate different states. Use LEDs to represent digital outputs. 4. Use switches or buttons to simulate different current states. Use LEDs to represent digital outputs. 5. Take microsd card out of system. Insert and press reset switch. Data should be written to data.txt if reset worked properly	1. Yes 2. Yes 3. Yes 4. Yes 5. Yes

Table 8: Requirements and verifications for output logic

Requirements	Verification	Result
1. Maintains an output voltage of 5 V +/- 0.2 V 2. Doesn't exceed temperature of 125 ℃, causing component to fail	1. Probe the voltage with battery at 100% charge and 50% charge with input power and without, making sure voltage stays within bounds 2. Temperature checked after 1 hour of normal operation in system and noting any signs of component failure (burning smell, smoke)	1. Yes 2. Yes

Table 9: Requirements and verification for voltage regulator

Requirements	Verification	Result
1. Each sensor data "capture" is stored as a space separated string. 2. Sensor data is correct to the tenths place at the time of writing to the microSD card.	1. Change frequency of "capture" from once per hour to once per ten seconds to verify the printing of strings 2. a. Isolate the writing to the microSD card for one sensor. b. Change the frequency of "capture" from once per hour to once per ten seconds. c. As a side-by-side comparison, record what the sensor reads. Then, open the SD card text file to check that the value written matches. d. Repeat steps a.-c. for each sensor.	1. Yes 2. Yes

Table 10: Requirements and verification for data collection

Appendix B Testing Data

```
                                                              Dehumidifier Heater Battery Voltage T in  T out H in  H out
B is high C is low H is low T is high x is low y is high z is low Q0 = 0 Q1 = 0 11.94 68.00 69.98 75.40 95.10
B is high C is low H is low T is high x is low y is low z is high Q0 = 0 Q1 = 0 11.94 68.18 70.70 74.90 97.80
B is high C is high H is low T is high x is low y is high z is high Q0 = 0 Q1 = 0 11.94 68.18 71.24 74.60 99.90
B is high C is high H is low T is high x is high y is low z is high Q0 = 0 Q1 = 1 11.94 68.18 71.96 69.00 99.90
B is high C is high H is low T is high x is low y is low z is high Q0 = 0 Q1 = 1 12.08 68.00 72.50 64.30 99.90
B is high C is high H is low T is high x is low y is high z is high Q0 = 0 Q1 = 1 12.08 68.00 72.86 61.50 99.90
B is high C is high H is low T is high x is high y is low z is high Q0 = 0 Q1 = 1 12.08 67.82 73.58 58.30 99.90
B is high C is high H is low T is high x is low y is low z is high Q0 = 0 Q1 = 1 12.08 67.64 73.94 55.70 99.90
B is high C is high H is low T is high x is low y is high z is high Q0 = 0 Q1 = 1 12.08 67.46 74.30 55.80 99.90
B is high C is high H is low T is high x is high y is low z is high Q0 = 0 Q1 = 1 12.08 67.46 74.84 60.40 99.90
B is high C is high H is low T is high x is low y is low z is high Q0 = 0 Q1 = 1 12.08 67.28 74.84 64.30 99.90
B is high C is high H is low T is high x is low y is high z is high Q0 = 0 Q1 = 1 12.08 67.10 74.48 72.40 99.90
B is high C is low H is high T is high x is high y is low z is high Q0 = 0 Q1 = 1 12.08 73.40 74.12 99.90 99.90
B is high C is low H is high T is high x is low y is low z is high Q0 = 0 Q1 = 1 12.06 74.66 73.94 99.90 99.90
B is high C is low H is high T is high x is low y is high z is high Q0 = 0 Q1 = 1 12.08 74.48 73.76 99.90 99.90
B is high C is low H is high T is high x is high y is low z is low Q0 = 0 Q1 = 1 12.08 74.48 73.94 99.90 99.60
B is high C is low H is high T is high x is high y is high z is low Q0 = 0 Q1 = 1 12.08 74.66 74.12 99.90 97.60
B is high C is low H is high T is high x is high y is high z is high Q0 = 1 Q1 = 0 12.08 74.66 74.12 99.90 94.60
B is high C is low H is high T is high x is low y is low z is high Q0 = 1 Q1 = 0 12.08 74.84 74.30 99.90 89.20
```

Figure 15: Test file stored on microsd card

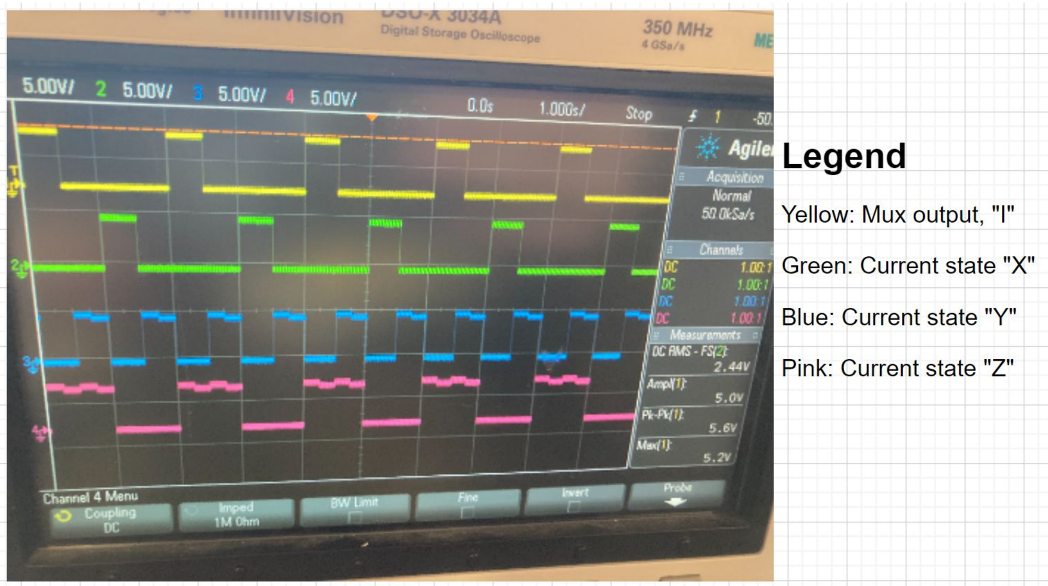

Figure 16: Finite state machine oscilloscope reading sample

Appendix C: Finished prototype

Figure 17: Finished prototype