SUBTERRANEAN THERMAL ENERGY STORAGE

USING THERMOSIPHON ARRAYS

by

Harvest Media Luna Montemayor

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

The University of Utah

August 2015

The University of Utah Graduate School

STATEMENT OF THESIS APPROVAL

The thesis of **Harvest Media Luna Montemayor**

has been approved by the following supervisory committee members:

Kent S. Udell	, Chair	**5/20/2015** Date Approved
Amanda D. Smith	, Member	**5/20/2015** Date Approved
Sivaraman Guruswamy	, Member	**5/20/2015** Date Approved

and by **Timothy Ameel** , Chair/Dean of

the Department/College/School of **Mechanical Engineering**

and by David B. Kieda, Dean of The Graduate School.

ABSTRACT

Increasing evidence of ecosystem damage caused by burning fossil fuels has sparked an interest in finding new methods of storing and utilizing energy that do not contribute to release of carbon dioxide. There is untapped potential for storing cold energy within the ground in climates with extreme winters and summers. This stored energy can be employed as a method of cooling buildings if effective heat transfer to and from the storage media and the building is obtained. A thermosiphon array is considered a means of enhancing that heat transfer. A test site has been studied at the University of Utah Sill Center consisting of two concentric thermosiphon arrays. It was designed to operate such that its performance is indistinguishable from traditional cooling systems.

The research objective of this paper is to document the initial stages of the installation and operation, as they have spanned many years. The data collected from the thermosiphons are presented as a baseline for performance of the system. Ground temperature data are analyzed by finding the thermal diffusivity of the soil at the site. The scope of this work includes implementing the data acquisition system, ensuring the thermosiphon array was operational and leak proof, and presenting preliminary performance data as a background for future work.

Results include the preliminary performance data for two charged thermosiphons as well as ground temperature data. The thermal diffusivity of the ground was calculated using iterative methods to match the actual temperature distribution found in the ground

from February to April 2015 to a mathematical model. The mild temperatures occurring late in the winter of 2015 stifled the freezing in the ground near the thermosiphons but still allowed detailed measurements of the thermal response to various atmospheric temperatures. It was also discovered that the thermosiphons were not receiving sufficient refrigerant, and that the addition of a flow meter and a pump would be beneficial. However, this project is to be continued for seasons to come in an effort develop a cost-effective method to reduce carbon dioxide emissions from cooling of commercial and residential buildings.

TABLE OF CONTENTS

ABSTRACT iii

ACKNOWLEDGEMENTS vii

CHAPTERS

1. INTRODUCTION 1

    1.1 Thermosiphon Heat Pipes 3
    1.2 Pump Assisted Thermosiphons 5
    1.3 Thermal Energy Storage 6
    1.4 Research Objectives 7

2. BUILD OF THE SYSTEM 9

    2.1 Site Choice and Construction 9
    2.2 Thermosiphon Array and Above-ground System 11
        2.2.1 Leak Detection and Prevention 13
        2.2.2 Charging the Thermosiphons 15
    2.3 Data Acquisition Hardware and Software 17

3. DATA COLLECTION AND ANALYSIS 19

    3.1 Thermosiphon Data 19
    3.2 Soil Temperature Data 34
    3.3 Periodic Heating 36

4. DISCUSSION AND FUTURE WORK 41

    4.1 Discussion 41
    4.2 Future Work 43

5. CONCLUSIONS 44

    5.1 Conclusions 44

APPENDICES

A: MATLAB CODE FOR THERMOSIPHON VERTICAL DEVIATION 46

B: THERMOSIPHON DEVIATION ANGLES 47

C: ARDUINO CODE 48

D: TEMPERATURE LINE THERMISTOR ADDRESSES 58

REFERENCES 60

ACKNOWLEDGEMENTS

I would like to thank my grandfather Michael Daily, for his continued support of my education and interest in my field of study, and his mother Enid for originally ensuring my educational success. I also thank my cousin Dean Daily for your help in steering me in the right direction toward higher education and facilitating my chosen path. My parents Drusilla and Timothy Montemayor have always given me the support I needed to get through tough times and lend an ear when I've felt overwhelmed. Thank you Kent Udell for providing such an incredible opportunity to work with this project. Thank you Bidzina Kekelia for your indispensable advice and guidance. I would like to mention Mike Beeman for all his hard work on this project including designing and compiling the DAQ. And finally, thanks to my fiancé Greg Scott, who is my best friend, giving me support, inspiration and motivation in all aspects of my life.

CHAPTER 1

INTRODUCTION

Global climate change has become a top concern for both leading and developing countries across the globe. Worsening weather conditions and record-setting natural disasters have prompted investigations into the possible cause and involvement of mankind's contribution to climate change. The concern comes from a desire to continue to shelter and feed Earth's exploding population in a way that allows the maintenance of a certain standard of living. Emerging technologies that can provide this standard of living without excessive pollution and production of greenhouse gases will only take center stage if they can be proven economically viable in competition with traditional use of fossil fuels.

Carbon dioxide is a powerful greenhouse gas released into the atmosphere when coal, natural gas and oil are used for fuel. The excess production of carbon dioxide since the Industrial Revolution has caused the planet's mean temperature to rise. The oceans contain 60% of the world's carbon dioxide including 30% of new carbon dioxide produced by industrious humans [1]. Any further increase threatens ecosystems and surpasses the ocean's capability to process carbon dioxide. Recent data from the National Oceanic and Atmospheric Administration (NOAA) Mauna Loa Observatory report mean carbon dioxide levels for the month of April, 2015 as 403 parts per million [2]. This powerful greenhouse gas has not been observed at this level since prehistoric times, when the Earth

was still forming into the life-giving planet it is today. Scientists have been working on many different methods to mitigate the release of carbon dioxide, whether by improving efficiencies, storing it underground [3], injecting it into oceanic crust [4], or capturing it from flue gas in power plants and factories for later use or sequestration [5].

The U.S. Energy Information Administration (EIA) provides electricity use information for residential and commercial sectors. In 2009 the residential use of electricity for space conditioning attributed to 47.7 % of electricity use, totaling 486 trillion Btu [6]. The same year, commercial buildings used 32% of their total electrical energy for heating ventilation and air conditioning (HVAC) systems, [7] totaling 648 trillion Btu for space heating and cooling [8]. The Department of Energy reports 38% of total U.S. carbon dioxide emissions are from residential and commercial building's energy usage [9]. These large percentages are presented here to underscore the potential for reducing carbon dioxide emissions and fuel use for the space conditioning of homes and commercial buildings.

Arguably one of the best methods to reduce the production of carbon dioxide and other harmful gases would be to seek renewable sources for energy production. Solar, wind and geothermal energy often come to mind when examining renewables. However, regions with extreme winters and summers offer free energy as a pure consequence of already occurring weather conditions. The harnessing of weather is given the term 'seasonal energy storage', and can offer a locally sourced, low-cost and pollution-free alternative to fossil fuels.

1.1 Thermosiphon Heat Pipes

Thermosiphon heat pipes are used at the University of Utah's Sill Center site to facilitate heat transfer between the underground storage and above-ground HVAC system. They are, as the name implies, a combination of a heat pipe and a thermosiphon. Gravity-induced, high-rate heat transfer occurs by vaporizing a liquid on a heated end and condensing the vapor at the cooled end. A heat pipe is a sealed tube, commonly made of copper with a high rate of heat conduction and protection against large temperatures. For this application, the pipe contains a heat transfer fluid that has a low boiling point and a low freezing point. The pipe can also contain a wicking material. One end of the heat pipe is introduced to high temperatures and a pressure gradient is formed as the fluid reaches boiling. The fluid quickly evaporates and travels to the cooler end of the pipe where heat is exchanged and the fluid condenses. The wick facilitates a capillary action that transports the condensed liquid back to the other end of the pipe so that the cycle can continue (Fig. 1.1). Heat pipes have many advantages that make them attractive for use in heat exchange systems. Only a small amount of working fluid is needed inside the pipe, and it is fully contained and sealed under vacuum. The potential for heat transfer is very high due to the vacuum inside the tube reducing the boiling point of the liquid, allowing even small temperature differences to be utilized. Additionally, heat pipes do not rely on gravity to function [10].

A thermosiphon is traditionally used for heating and works via the forces of gravity using natural convection and a working fluid contained inside, whose circulation is considered passive. As the fluid absorbs heat from its surroundings, the liquid phase of the working fluid vaporizes. The heated vapor will rise to the top of the thermosiphon where

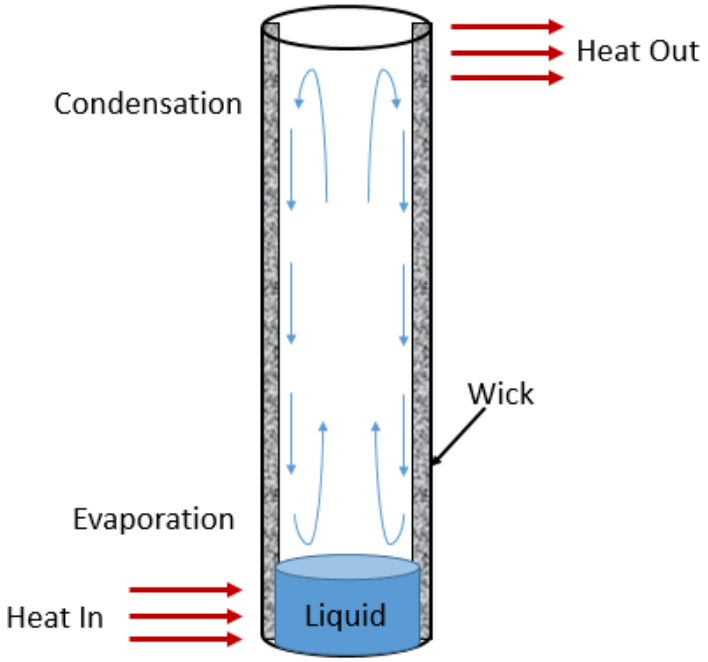

Figure 1.1. Heat pipe schematic.

it will transfer its heat and condense to fall to the bottom under the forces of gravity to begin the cycle again (Fig. 1.2, with added pump). A known advantage of thermosiphons has been that they do not require a pump to operate, removing heat from below and transferring it to a cooled zone above. Since the liquid phase is not present at the top of the thermosiphon if the temperature there is higher than at the bottom, the thermosiphon naturally operates as a thermal diode, only transferring energy upwards. However, a pump is required inside the thermosiphon if energy is to be transferred to a cooled region below the space where cooling is desired [11].

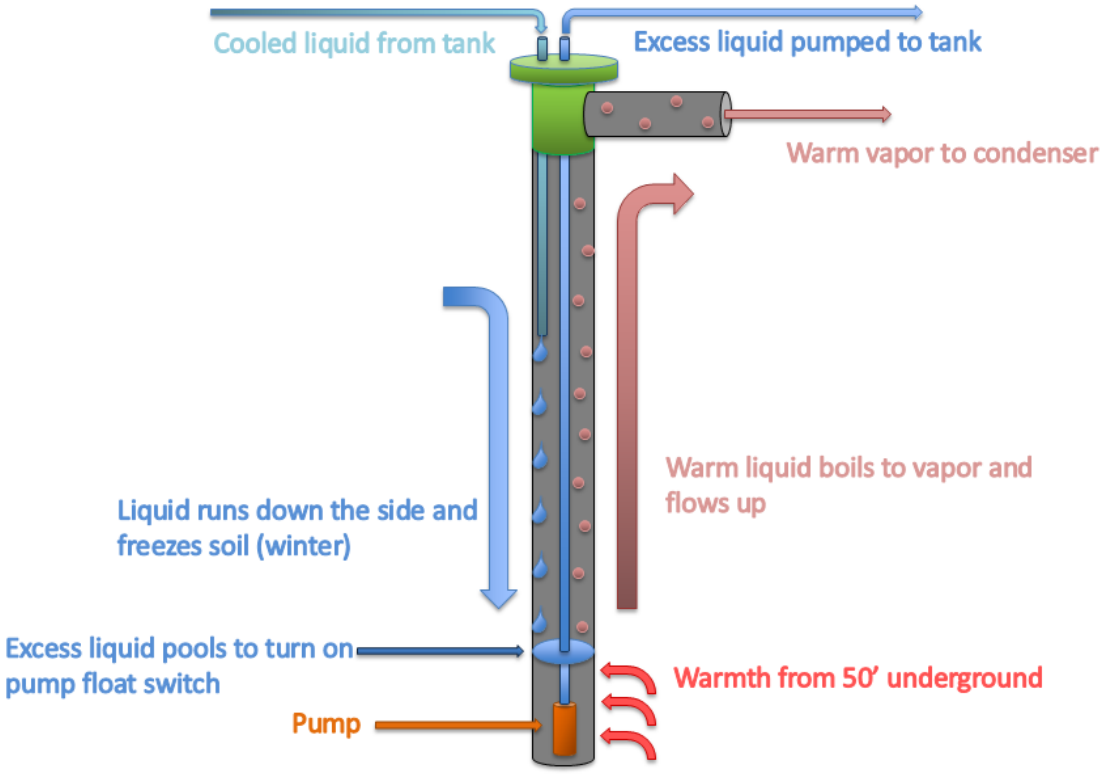

Figure 1.2. Pump assisted smart thermosiphon.

1.2 Pump Assisted Thermosiphons

The combination heat pipe and thermosiphon give the benefits of both objects and have been employed at the Sill Center site. A pump has been added to the bottom of each thermosiphon so it may be used for space cooling. The pump is used to transport the condensed vapor of the working fluid that has traveled to the bottom of the thermosiphon back up to the evaporator that is located above-ground in the HVAC system. The pump does require a small amount of electricity, but that power could easily be supplied by a 10 Watt solar panel. This overall combination of the pump assisted thermosiphon heat pipe is a novel portion of this work, and has been coined under the term 'smart thermosiphon'.

When the thermosiphons are grouped into an array, they are called 'smart thermosiphon array', or STA [11] [12].

1.3 Thermal Energy Storage

Underground thermal energy storage uses soil as the heat depository medium, and has many advantages for storing hot and cold energy, the foremost reasons being that the soil is free of cost and available almost everywhere. Soil naturally has a low thermal conductivity and suitable thermal capacity as long as an adequate volume is available. This volume must be determined by the heating or cooling load required by the nearby building. It follows that drilling or digging of trenches is mandatory to utilize soil as the storage medium, and this can be of considerable cost given the specific nature of the chosen site, and its relative resistance to penetration. Drilling would result in vertical bore holes, whose depth is again determined by the heat load required. Drilling and boring, although more costly than alternative horizontal trenches, offers a space-saving configuration as the majority of the distance is vertical, hiding the size deep underground. This vertical orientation is the configuration at the Sill Center site, where the bore holes are up to 63 feet deep. It is known that year-round temperature fluctuations for soil have a much lower amplitude than that of the ambient air temperatures outside. This is due to the high volume and low thermal conductivity of the soil. It was found in this research that soil temperatures at the University of Utah ranged from 5 °C to 13 °C (Fig. 1.3), with standard deviations from the mean for all dates between February 20 and March 9, 2015. The soil temperature is dependent on the depth, and temperature fluctuations greatly decrease as depth is increased. Specifically, at a depth below 25 feet, the temperature becomes stabilized.

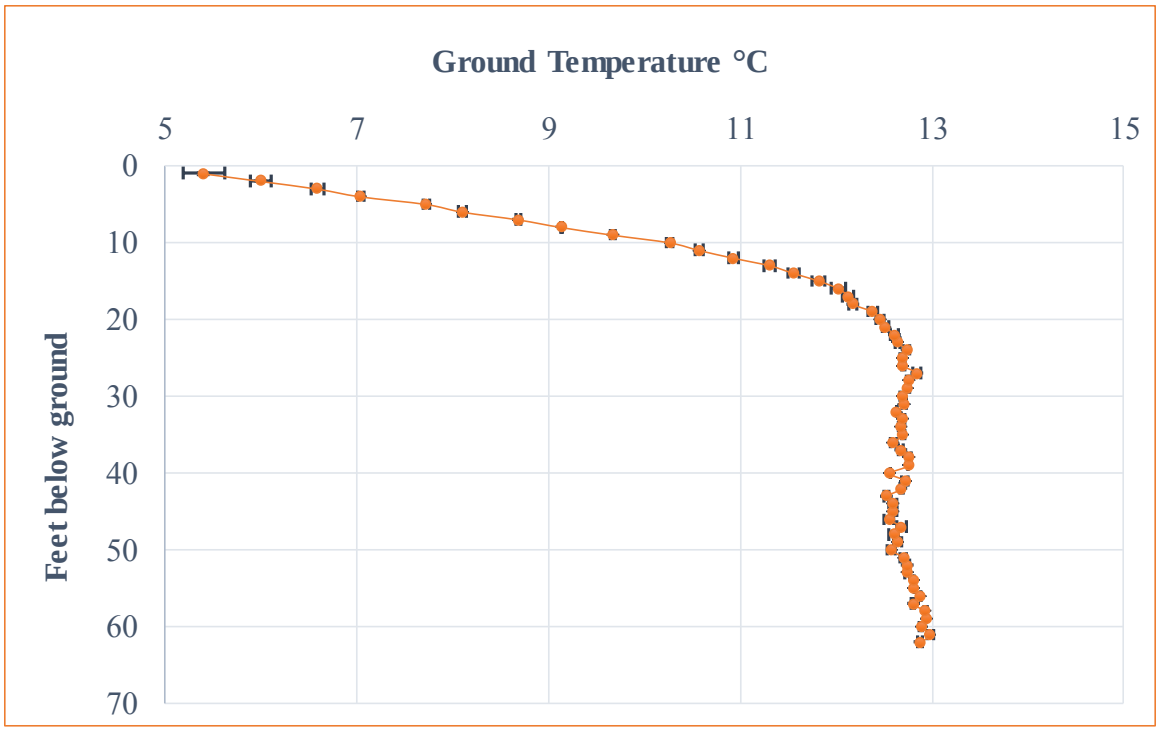

Figure 1.3. Ground temperatures across depth for winter 2015 at the Sill Center.

1.4 Research Objectives

As the project has spanned a period over 5 years and has involved handfuls of students, there exists no concise record of the progress made with this project thus far. This work documents the history of the project and identifies improvements to the system to aid in future work. Temperature data for the system are presented over a 3-month period from February to April 2015. Implementation of the data acquisitions system was completed in addition to ensuring operations of the system by leak detection and prevention. Necessary improvements for the system were found during the course of this work and include the installation of flow meters and above-ground pumps for monitoring and distributing fluid back into the thermosiphon field. Heat transfer analysis was performed using thermal diffusivity, ground temperatures and preliminary thermosiphon temperature data. The

results of this analysis gave good indication and reason to continue this work, while helping the Sill Center to operate with less emissions.

CHAPTER 2

BUILD OF THE SYSTEM

2.1 Site Choice and Construction

It was by invitation that the Sterling Sill Center hosts the smart thermosiphon array. The center is known for its sustainable projects including an array of solar panels and an organic garden. The soil bank chosen for the site located behind the Sill Center was in good proximity to the building and adequate for drilling deep boreholes. The boreholes are not perfectly vertical and their deviation is calculated with Matlab for visualization (Fig. 2.1). The code and information used to create the graph in Fig. 2.1 are included in Appendices A and B. Each deviation is denoted by a blue line in units of feet and the top of the thermosiphon is shown with a red circle. Construction began in the spring of 2010 at the Sill Center, and after the holes were drilled, two circular trenches were dug by hand to lay the 1 inch high density poly ethylene (HDPE) piping connections for the vapor lines. Each array is a separate entity; the inner array of 7 thermosiphons is connected together by a circular ring of the 1 inch HDPE pipe, and similarly for the outer array of 12 thermosiphons.

Altogether, 20 holes were drilled into the ground behind the Sill Center and they are grouped in two concentric hexagons 1.52 meters (5 feet) apart (Fig. 2.2). The first 19 holes, one for each thermosiphon, are all 50 feet in depth save number 11, which could only be drilled to 35 feet due to an obstruction. The last and 20^{th} hole is drilled at a depth

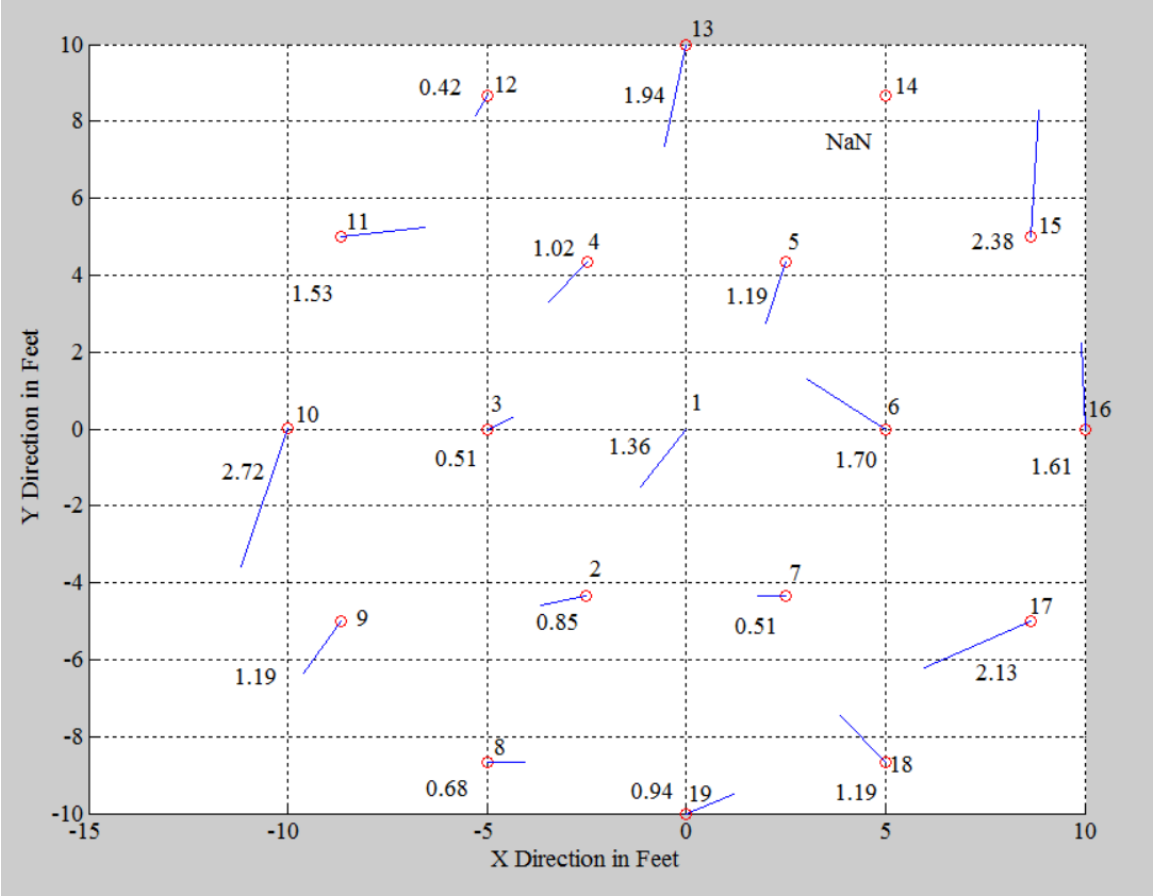

Figure 2.1. Deviation from the vertical for each thermosiphon.

of 63 feet and contains a line of 63 temperature sensors for monitoring ground temperature and is situated between the two hexagons (Fig. 2.2). The thermosiphon heads themselves are 1 to 3 feet below ground, accessed by green irrigation control valve boxes. One additional green control valve box was added near the outer array to house the wiring center for the pumps and their float switches coming from the array and connecting to the data acquisition center.

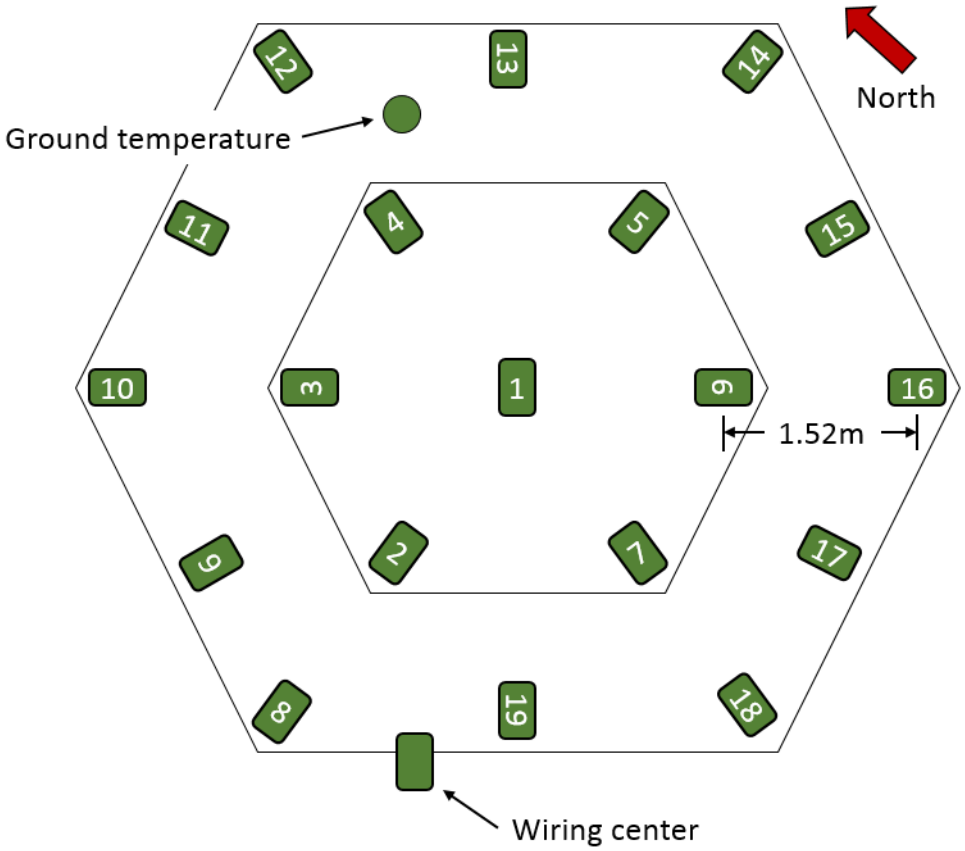

Figure 2.2. Top view of thermosiphon array.

2.2 Thermosiphon Array and Above-Ground System

Each thermosiphon has the connection to the circular, 1 inch HDPE vapor pipe ring with a large valve that is located slightly below the thermosiphon head. Additionally, each thermosiphon has liquid and vapor lines protruding from the top of the head made of ¼ inch copper pipe with brass valves. These brass valves were installed to throttle the amount of liquid or vapor coolant to and from the system. These copper lines lead back to the tank and condenser, respectively. Again at the top or head of each thermosiphon, a brass wire feed-through allows the passage of wires from the pump and float switch. There are three

thermosiphons with temperature sensor lines, numbers 1, 4 and 13, which also come through the brass wire feed-through (see Fig. 2.2 for numbering scheme).

Two HVAC systems in place next to the Sill Center building are connected to the thermosiphon arrays (Fig. 2.3). The inner thermosiphon array belongs to the system on the left, and the outer array is connected to the system on the right. Each has identical components; a blue liquid tank for refrigerant storage and overflow, condenser and evaporator, and two fans. Liquid sight glasses can be seen just to the left of each liquid tank. A system of solenoid valves are brazed in place behind each evaporator, and covered with a large galvanized steel sheet metal housing. The grey control box between the two setups houses the electrical switches to control the solenoid valves for the inner array. A schematic of the switch configuration is included in Fig. 2.4, and there are also power switches for each of the two fans, the compressor and a metered pump that has yet to be

Figure 2.3. HVAC systems for the two thermosiphon arrays.

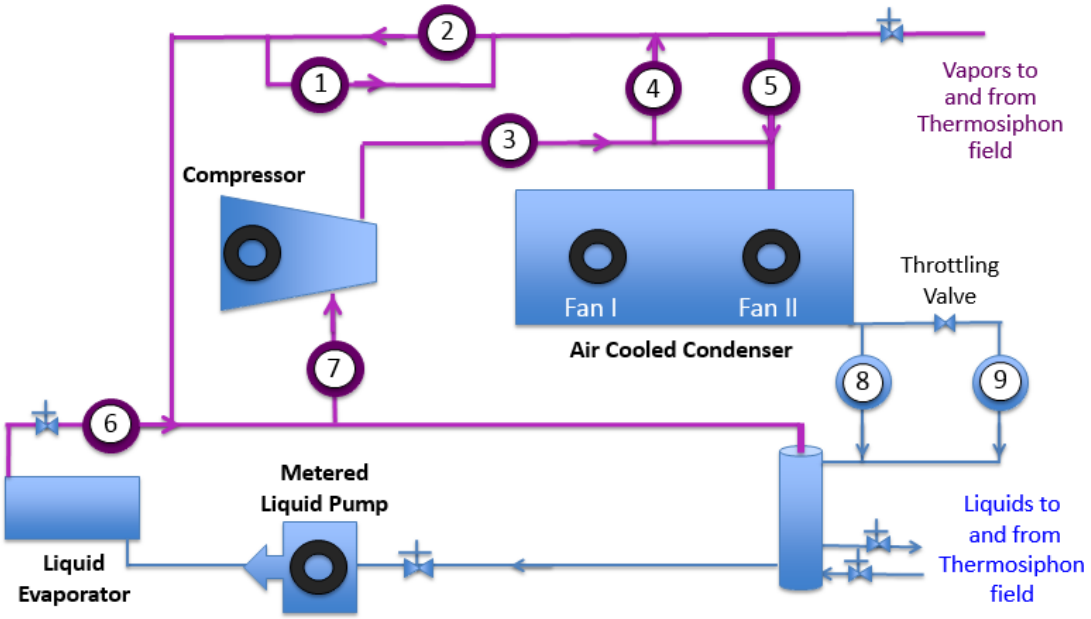

Figure 2.4. Solenoid valve numbering 1-9 for inner ring thermosiphon array.

purchased and installed. The grey control box on the very left contains the data acquisition system.

2.2.1 Leak Detection and Prevention

As the system must operate under vacuum in addition to containing refrigerant at pressure near 60 psi, leaks in the piping would not be acceptable. As in conventional refrigeration systems, connections must be copper or brass and brazed when possible. Leaks were first detected in the summer of 2013 originating from the thermosiphon head assembly. It was discovered that the metal threaded components were unable to contain pressurized air indefinitely. It was decided to replace as many threaded metal components as possible with HDPE plastic weld-able components (Fig. 2.5). These components were

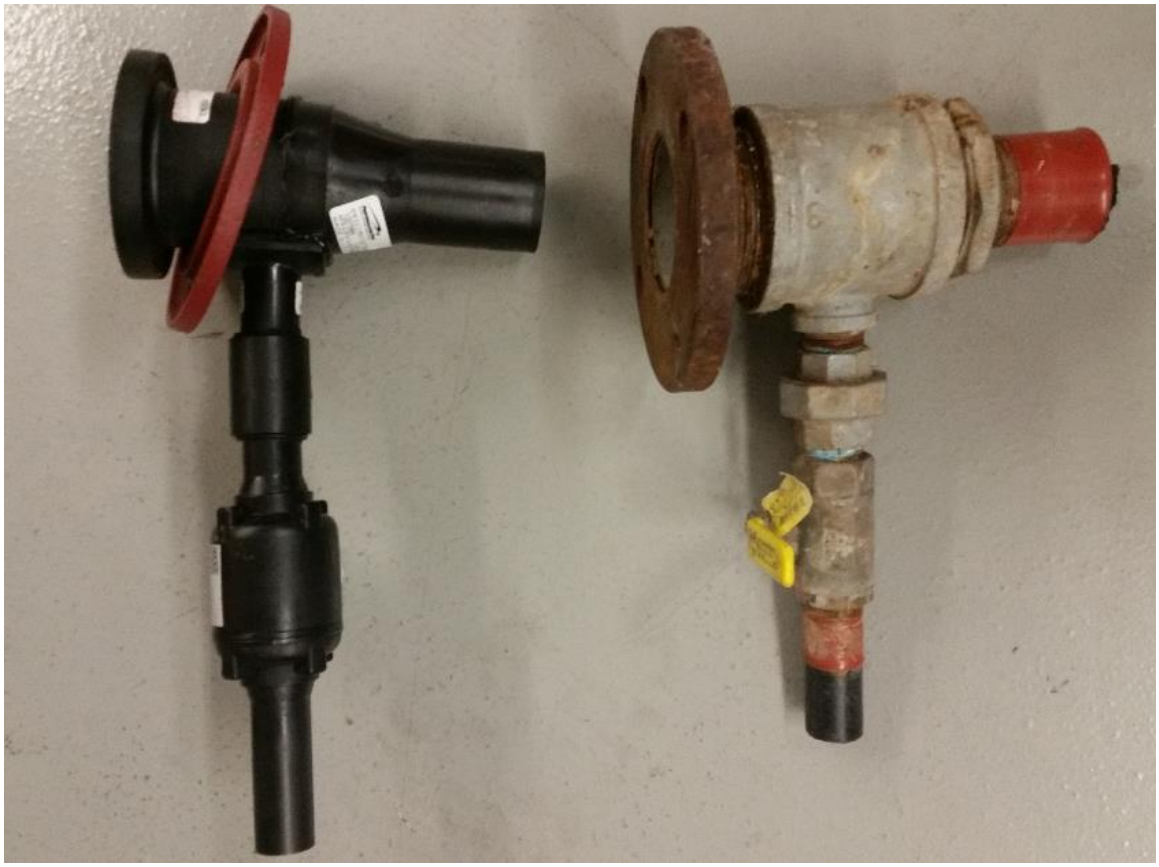

Figure 2.5. Thermosiphon heads. LEFT: HDPE thermosiphon head. RIGHT: Old thermosiphon head.

purchased from High Country Fusion located in Salt Lake City. Beginning in the fall of 2014, there were dozens of leaks detected with a soap and water solution. Bubbles could be seen between the brazed copper joints of the piping system, and between most threaded joints. All threaded joints were replaced with copper slip fittings and brazed until no additional leaks were detected. Threads could not be replaced where the tanks are connected to the system, therefore a special brazing rod was purchased to marry steel to copper. Leaks detected between the connections to the thermosiphon heads and the vapor and liquid lines were eliminated by placing silicone sealant inside the brass fittings.

2.2.2 Charging the Thermosiphons

The system was charged with R134a on February 20, 2015. Before charging, the system was determined to be free of leaks by pressure testing with pure nitrogen. A vacuum pump was carefully connected to the system and left to run overnight to fully evacuate the system of any entrained air or nitrogen (Fig. 2.6). At 28 inches Hg, or 0.032 bar, the vacuum pump was removed and a 30 kg tank of R-134a refrigerant was placed on a scale upside down (Fig. 2.7). The tank was then connected to the liquid return line leading through the thermosiphon cap. The liquid return line terminates at the top of a fabric mesh 3.05 meters (10 feet) below the ground surface. By monitoring the electrical resistance across the wires leading to the down-hole pump, it was found that the thermosiphons filled to the location of the float switch with about 6 kg of R134a. However, the switch was not activated after draining over 10 kg into thermosiphon number 1 (TS1). Due to concerns of a switch failure

Figure 2.6. Vacuum pump used to evacuate the system.

Figure 2.7. Charging the thermosiphons with R-134a.

in TS1 that could allow the thermosiphon to fill with liquid R134a, valves on both the return liquid line and the vapor line were turned off. Each of the other 6 thermosiphons in the inner array received 6 kg of refrigerant, and 10 kg was deposited into the HVAC storage tank. An electronic leak detector was used on each of the seven thermosiphons to ensure that no R-134a would be released to the atmosphere. A leak detected in thermosiphon number 6 was unable to be repaired due to a cracked brass nut on the liquid return line, and that thermosiphon was isolated by turning all three of its valves to the off positon.

2.3 Data Acquisition Hardware and Software

The data acquisition system used for this project is an Arduino Mega and a Hydrogen Wi-Fi shield with mini SD card storage (Fig. 2.8). The Arduino program used in this study is included in Appendix C. The program uses open source code to collect temperature and pump power status and is aided by a real-time clock backed up by a 3V battery. The Arduino is run by its own power supply that can be replaced by a solar panel and battery. There are a total of 19 pumps with on/off status recorded onto the SD card in time intervals chosen by the user. None of the pumps responded to the liquid in the thermosiphons, and no pump data were ever recorded.

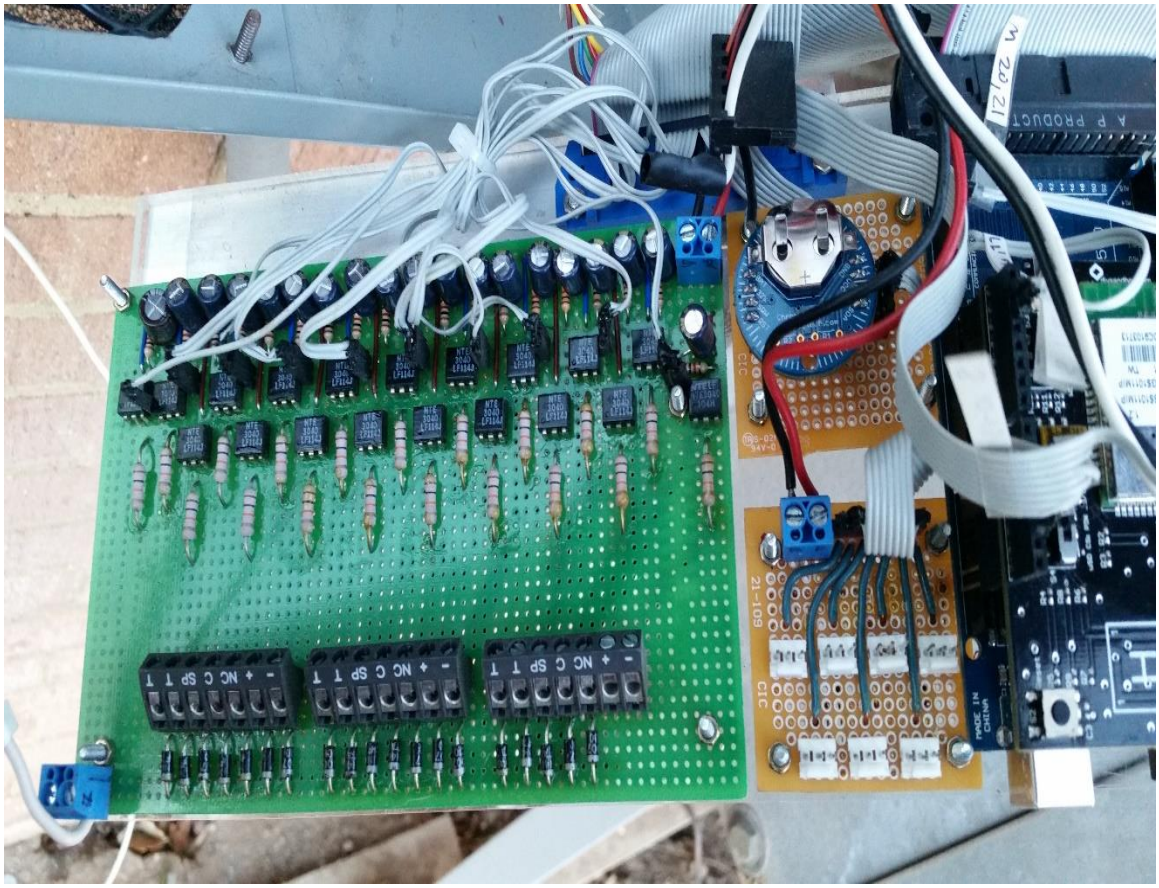

Figure 2.8. Data acquisition system.

Five separate temperature lines are recorded within the Arduino program and can be accessed by opening the SD card's logger files. There are 11 temperature sensors on a 50 foot line inside thermosiphon numbers 1, 4 and 13, (TS1, TS4 and TS13). No temperature data were recorded for TS13, as the sensor line malfunctioned. Circuit testing indicated that there was a short circuit in the subterranean line. The ground temperature line is 63 feet long with a temperature sensor for every one foot. The last and 5th line is ambient air temperature, monitored by one temperature sensor, and this line protrudes from the data acquisition box hung on the building. All temperature sensors used are Dallas DS18B20 thermistors on 3-core wire (Fig. 2.9). These temperature sensors have a programmable resolution and each sensor is identified by a unique 64-bit, 16 digit code for identification. The addresses for each thermistor for all temperature lines are included in Appendix D and are used to interpret the location of the sensors inside the thermosiphons.

Figure 2.9. Dallas DS18B20 temperature sensor line with two thermistors.

CHAPTER 3

DATA COLLECTION AND ANALYSIS

3.1 Thermosiphon Data

The thermosiphons were designed to operate over winter such that liquid refrigerant was contained in the wick at 35 to 50 feet below ground. This liquid would vaporize due to the warmer, deep underground temperatures and flow upwards (Fig. 3.1). Wintertime soil temperatures near the surface create a temperature gradient so the rising vapor will either condense in the top of the thermosiphon or travel to the condenser if the vapor pressures are lower there. In that case, the vapors from the hot bottom regions in the soil can condense in the cold upper soils from 0 to 25 feet while liquid condensate runs down the sides of the pipe towards the wick where it should be vaporized again. This cold energy would build up over the winter and be redistributed above-ground for air conditioning to the Sill Center during the following summer. While the thermosiphons are operated over winter, temperatures from 0 to 10 feet should be the coldest recorded inside the thermosiphons and temperatures everywhere else should reflect the response to changing ambient temperatures. For the data collected and reported in this work, the response of the thermosiphons was observed under three conditions:

1. low ambient temperatures reflected in low condenser pressures
2. transfer of heat from high temperature locations deep in the ground

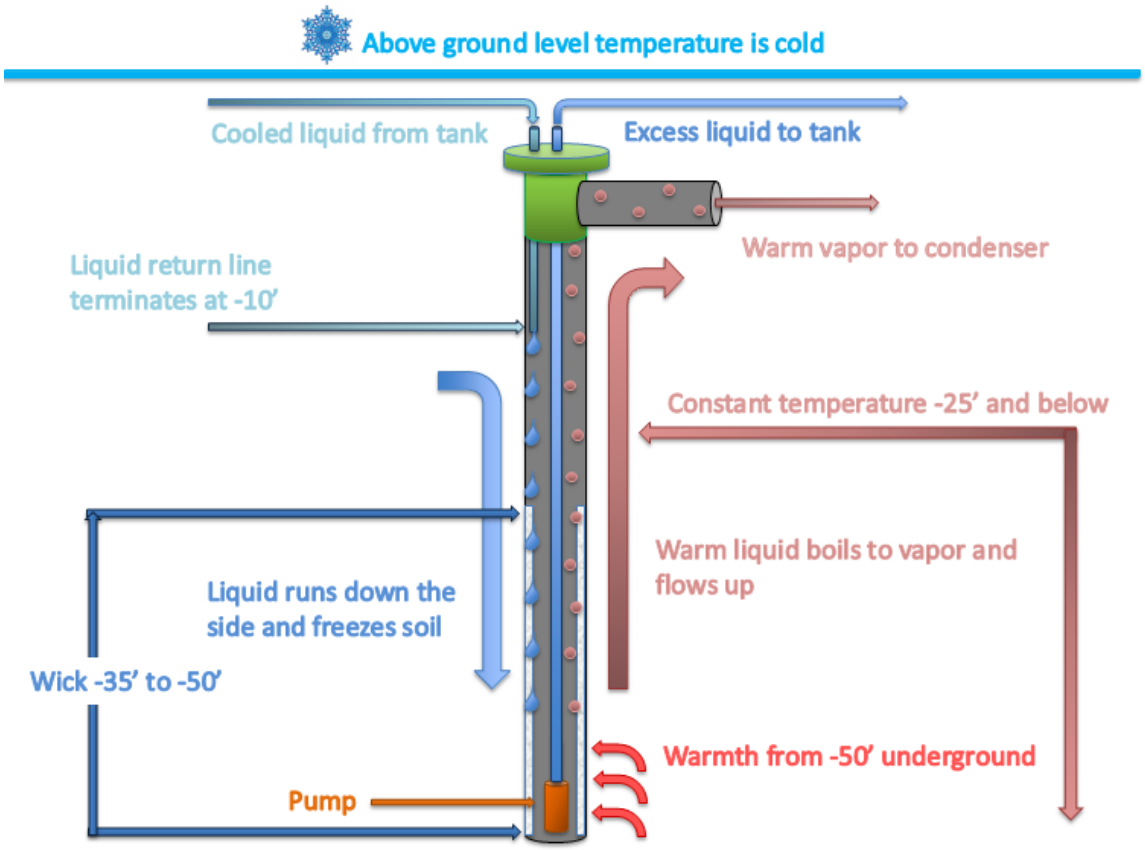

Figure 3.1. Schematic of pump assisted thermosiphon heat pipe.

3. low temperature regions near the soil surface

The temperature measurements within TS1 and TS2 provide insight into their operation during various situations. In subsequent figures, each colored line is either a single temperature sensor defined in the legend by its depth of 0 to 50 feet below ground level, or the ambient temperature. Most temperature data are presented in 30-minute intervals except two that are presented in 15-minute intervals. Although data were actually collected once a minute, that level of detail was not needed for interpretation. Each operational change, referenced by Table 3.1, is indicated by the event number and correlated observed responses are designated alphabetically in all figures.

Table 3.1 Timeline of operational events.

#	Date	Change	Reasoning
1	20-Feb	Initial Charge	
		Opened TS4 valves 1/4 turn	Intent on throttling liquid and vapor
2	21-Feb	Opened TS1 valves fully	Insufficient fluid returning to TS1
		Turned fans on	Initiate condenser and return chilled liquid
3	24-Feb	No data for 24th, took Arduino out and loaded known file	Dates were discovered to be wrong
4	4-Mar	Large vapor line for TS1 was previously closed, now open	Allow complete circulation of vapor
5		Pump data was not found after testing	No pump data was able to be collected
6	6-Mar	TS1 and TS4 vapor line closed	Attempt to trap liquid inside TS1 and TS4
7		Re-charged TS1 3:36 pm	Insufficient fluid in TS1
8		TS4 liquid line opened the rest of the way from 1/4 open	Insufficient fluid returning to TS4
9		TS1 temp lost, reads 85	
10		TS4 temp sensor 0, 25 & 50 lost, reads -127	
11	10-Mar	All liquid valves closed except TS1 and TS4	Insufficient fluid Prevent liquids to TS2,3,5,6,7
12	20-Mar	Fans off 4:46 pm	Nearing end of experiment, see what occurs

After the thermosiphons were initially charged, their liquid return line valves were opened ¼ turn (Fig. 3.2 event #1) with the intent of throttling the liquid flow to each thermosiphon to provide uniform liquid returns rates. Liquid and vapor line valves were both closed on TS1 until February 21 (Fig. 3.2 event #2). The first evidence TS1 was operating as intended begins after event #2. This can best be seen by the sudden drop in temperature for sensor numbers 35 to 50 in the presence of the liquid condensate. There is then an exponential increase in temperature in these sensors, as well as a relationship

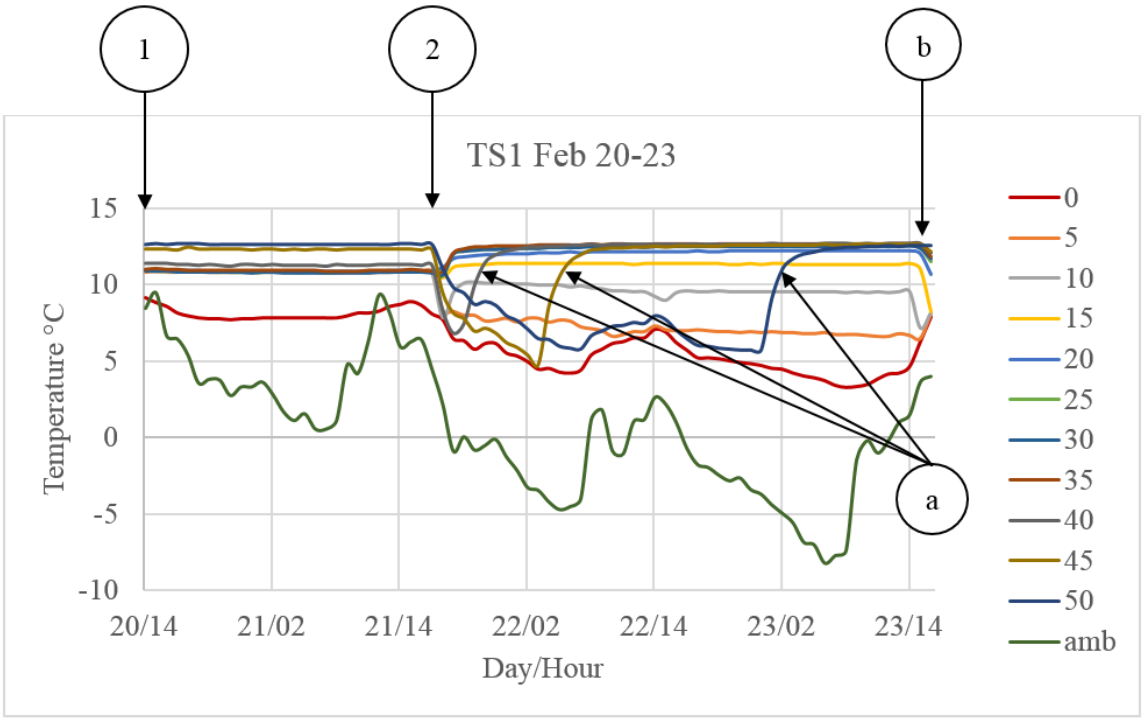

Figure 3.2. TS1 operation February 20-23. #1: Thermosiphons charged, TS4 return valves opened ¼ turn. #2: TS1 liquid and vapor return valves opened fully, turned fans on. a: Sudden decrease in temperature then exponential rise in temperature as liquid runs out and vaporization ceases. b: Expected decrease in temperature with condensation at the top of TS1.

between sensor depth and how quickly vaporization and subsequent condensation ceases while TS1 begins to run dry of fluid (Fig. 3.2 a). Condensation at the top of TS1 reduces the temperatures for sensors 10 to 25 during the warmest ambient temperatures for February 23, even as sensors 0 and 5 increase (Fig. 3.2 b) giving the expected behavior of attempted equalization of temperatures.

Activity for TS4 began when the vapor and liquid lines were opened right after charging (Fig. 3.3 event #1) with a more rapid exponential dry-out as compared to TS1 (Fig. 3.3 a) and a more complicated overall reaction. The refrigerant was caught in the wick and took many hours to drain to the location of the 50 foot temperature sensor. As the deep

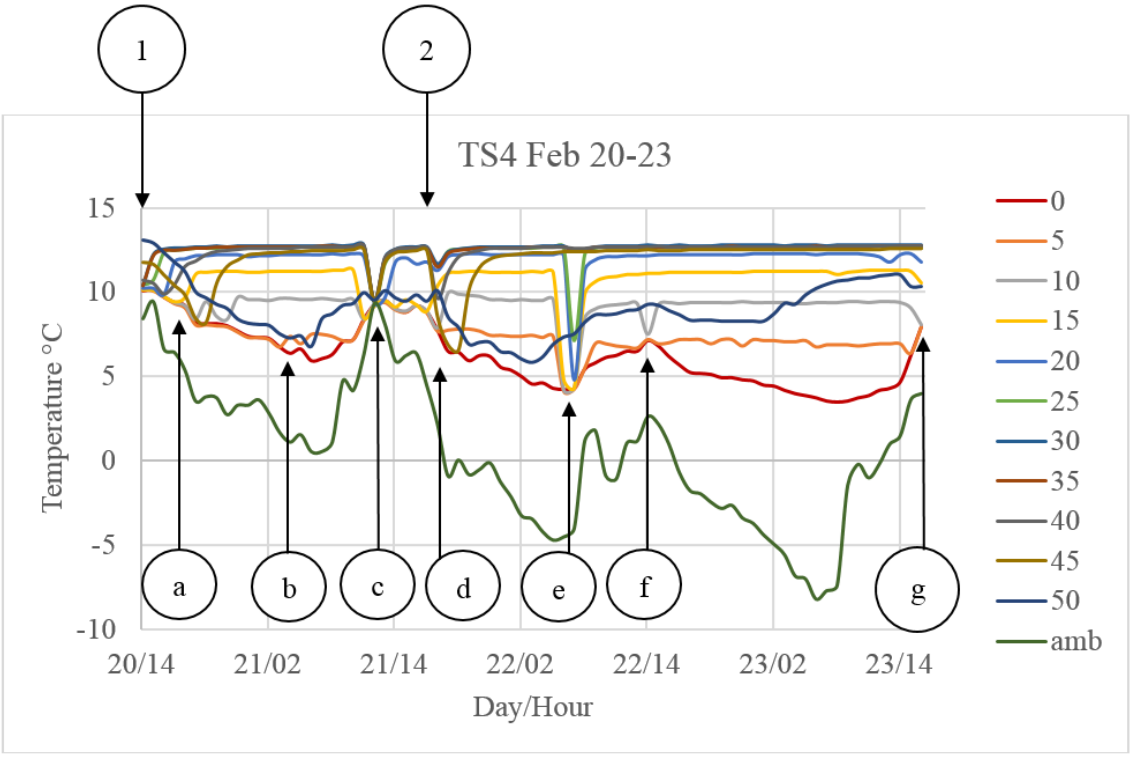

Figure 3.3. TS4 operation February 20-23. #1: Thermosiphons charged, all liquid and vapor return valves opened ¼ turn. #2: TS1 liquid and vapor return valves opened fully, turned fans on. a: Sudden decrease in temperatures then exponential rise in temperatures as liquid runs out and vaporization ceases. b: Liquid at 50 feet equalizing temperature with 0 and 5 foot sensors. c: First occurrence of complete convergence for all sensors. d: Event # 2 causing reaction in TS4, curiously most prominent in sensor 45. e: Lowest ambient temperatures for that day, fluid drains back from above-ground and again rebounds. f: Sensor 10 dropping a few degrees from return line. g: Sensors 10-20 attempting to equalize with sensors 0,5 and ambient.

sensor 50 attempted to match ambient temperature, it equalized with sensors 0 and 5 (Fig. 3.3 b), as was the expected behavior. The first complete temperature convergence is seen in TS4 (Fig. 3.3 c) where the ambient temperature rises to its highest value and all other temperatures attempt to equalize as there is vaporization all throughout the lower length of the pipe and condensation at the top. Condensation at the top of the pipe begins to slow and vaporization then begins to increase as temperatures again exponentially rebound

upward. However, evidence of condensation cooling that would chill the surrounding soil can be seen in the drop in temperature of sensors 10 and 15 (Fig. 3.3 c). As the liquid and vapor return lines from TS1 were fully opened and the fans turned on (Fig. 3.3 event #2), there was a response in TS4 (Fig. 3.3 d) resulting in decreased temperatures most drastic near 45 feet. This is an indication that opening the vapor return lines for TS1 either decreased the pressure in the other thermosiphons or caused an increase in the liquid flow into TS4. However, the minimum temperatures in TS1 and TS4 at depth (>30ft) were similar implying similar pressures. The fans to the condenser were also turned on at the same time that TS1 was brought online. Thus, this change is attributed to turning on the fans, which increased the condensation rate of the vapors above-ground, and an accompanied drainage of liquid back into the thermosiphon. Similarly, as ambient temperatures drop to the lowest levels overnight, increased liquid condensation above-ground led to increased liquid flow back into the thermosiphon (Fig. 3.3 e) seen by the dramatic decrease in temperature in sensors 5, 10 and 15, of which 10 is the location of termination in the return liquid line where the fluid would first encounter the pipe as it travels back to the field from the tank. There is a slight time lag for liquid to drain down, cooling sensors 20 and 25, but condensation quickly stops and evaporation of the fluid again exponentially raises the temperature at that location (Fig. 3.3 e). Fluid can again be seen draining from the return liquid lines dropping sensor 10 a few degrees (Fig. 3.3 f) but temperatures again soon rebound, although at a lower temperature than the previous rebound (Fig. 3.3 d). This lower rebound also shows a potential for chilling the ground outside the thermosiphon at 10 and 15 feet. Condensation at the top of TS4 reduces the temperatures for sensors 10 to 20 at the warmest ambient temperatures for February 23,

even as sensors 0 and 5 increase (Fig. 3.3 g) emulating the expected behavior of attempted equalization of temperatures also seen in TS1.

No temperature data are available for February 24 as the Arduino was off-line for software updates for that time (Fig. 3.4 event #3). For TS1, the first occurrence of spiked decreases in temperature occur (Fig. 3.4 a). It is hypothesized that liquid is condensing in the 1-inch HDPE vapor lines that circle the array in shallow soils and falling back into the thermosiphon causing the transient spikes. Partial convergence between the 0, 5 and 10 foot sensors was observed but it appears that there was not enough fluid to stay cold (Fig.

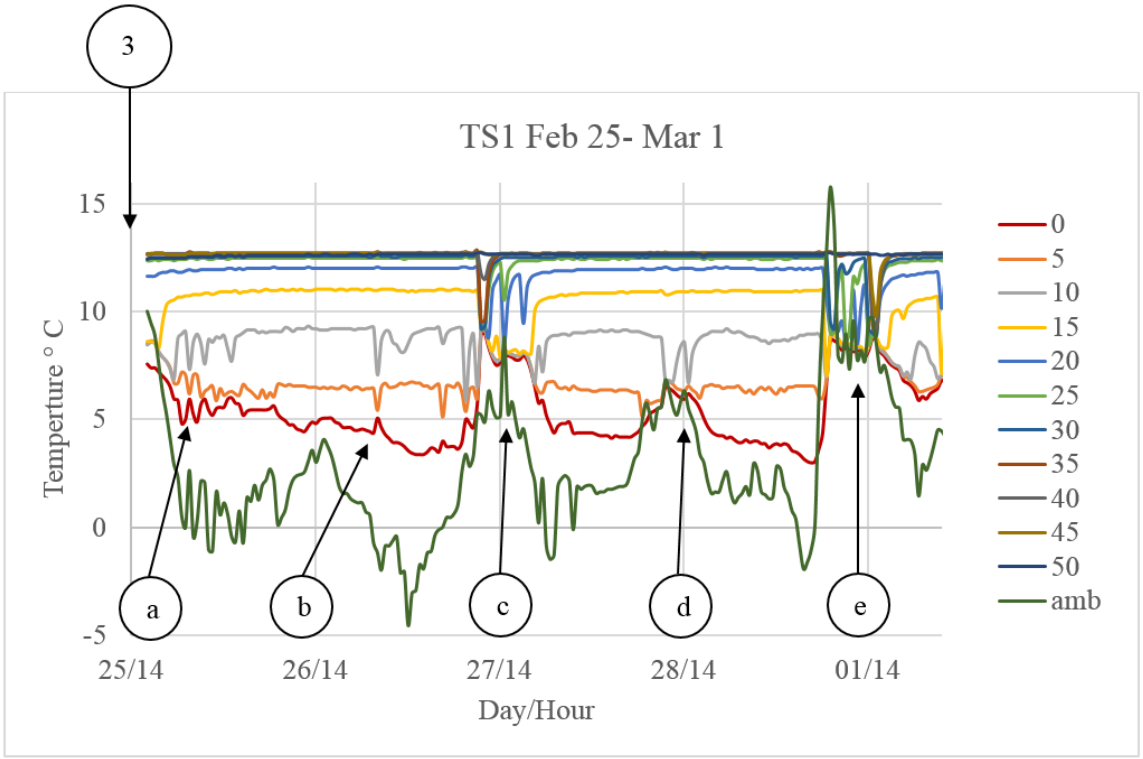

Figure 3.4. TS1 operation February 25-March 1. #3: No data for Feb 24 is available. a: Transient spikes in temperature decrease are first seen, attributed to vapor line condensate. b: Expected temperature convergence between sensors 0, 5 and 10 does not hold. c: Convergence for Feb 27 is incomplete and sensor 15 diverges from sensor 0 as vapor is lost. d: Convergence for sensors 0, 5 and 10 again dry out with decreasing ambient temperature. e: See Fig. 3.4 for close-up view.

3.4 b). Incomplete convergence is seen at the peak temperature for February 27, adequately reducing sensor 15 for soil chilling but without sufficient liquid to overcome nighttime ambient temperatures as 15 diverges from sensor 0 (Fig. 3.4 c). For February 28, low ambient temperatures converge with sensors 0 to10 and quickly dry out (Fig. 3.4 d). A close-up view for March 1, TS1 is shown in Fig. 3.5 (Fig. 3.4 e).

Fig. 3.5 shows a detail for TS1, March 1. Desired behavior for sensors 0 to 15 shows potential for temperatures to equalize, condense and cool the soil (Fig. 3.5 a). Shortly after, the ambient temperature exceeds the highest temperatures inside the thermosiphon for the first time and incomplete convergence ensues for sensors 20 to 30. Then oscillations with ambient temperature occur while sensors 0 to 15 hold steady for roughly 7 hours (Fig. 3.5

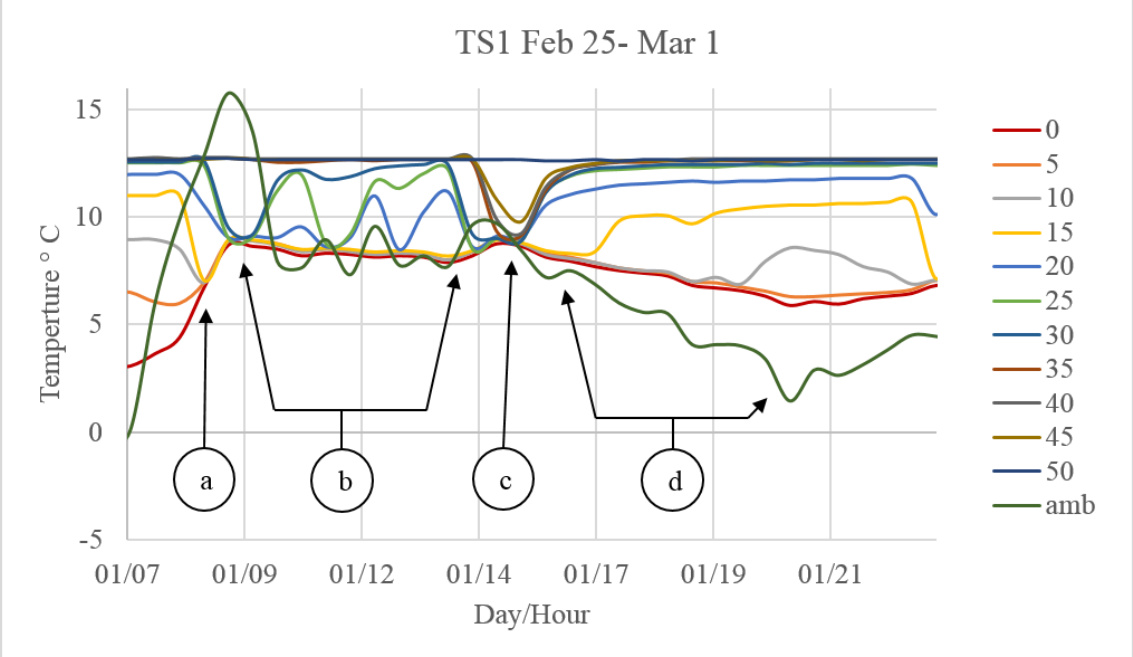

Figure 3.5. Detail view of TS1, March 1. a: Equalization for sensors 0-15 initiate desired behavior. b: Ambient temperature exceeds the warmest temperature inside the thermosiphon for the first time and temperatures converge, sensors 20 and 30 oscillate with ambient while 0-10 hold steady for 7 hours. c: Incomplete convergence. d: Divergence with cooling ambient.

b). This behavior would be ideal had it held consistently for sensors 20-30 as well. An encouraging subsequent cooling of sensors 20 to 30 was followed by a time lag reaction with sensors 35 to 45, again attempting convergence with ambient temperatures (Fig. 3.5 c). The characteristic dry-out is inferred by a rapid increase in temperature and an exponential decay to local soil temperature, and sensor temperatures diverge with cooling nighttime temperatures; however, sensors 15 and 20 have in fact cooled overall on this interval (Fig. 3.5 d).

Fig. 3.6 for TS4 shows the same behavior as TS1 in Fig. 3.4 a and c. The short transient spikes are again attributed to condensate congregating in the vapor lines and

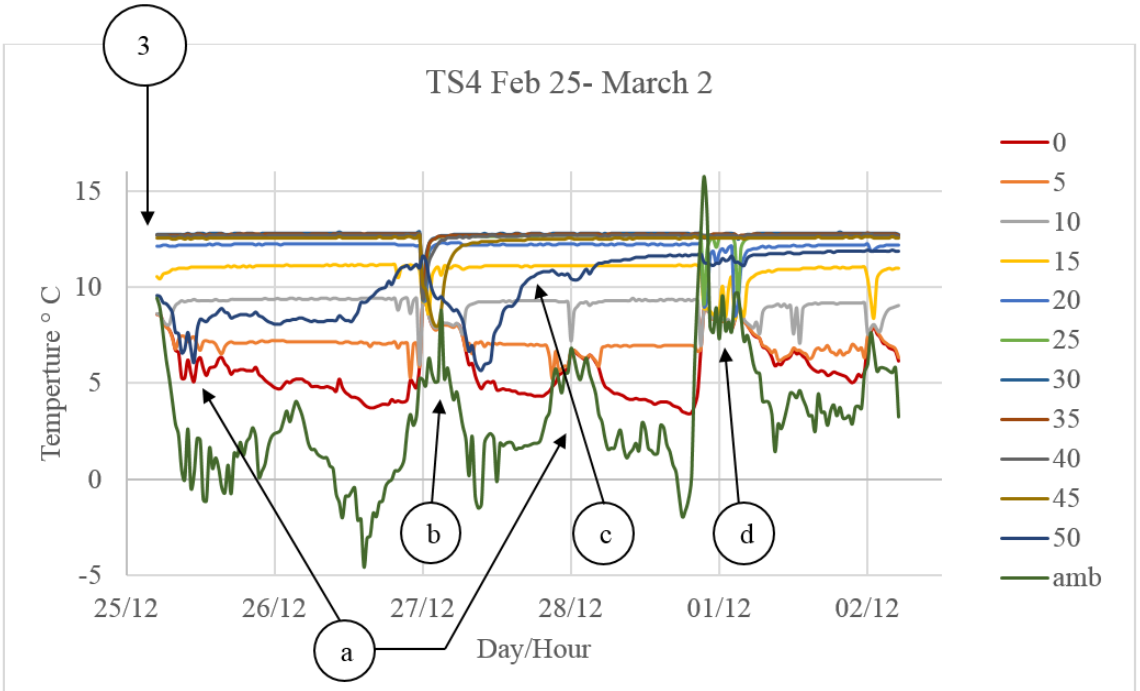

Figure 3.6. TS4 operation February 25–March 2. a: TS4 shows similarities to TS1. b: A stronger response for TS4 since there is fluid left at the bottom indicated by the low temperature of the 50 foot sensor. c: Liquid at the bottom begins to dry out shown by the exponential increase in temperature for sensor 50. d: See Fig. 3.7 for close-up view of March 1.

flowing back into the thermosiphon (Fig. 3.6 a). However, a stronger cooling response is seen for TS4 on February 27 since there is fluid left at the bottom indicated by the low temperature of the 50 foot sensor (Fig. 3.6 b). Liquid at the bottom begins to dry out shown by the eventual increase in temperature for sensor 50 (Fig. 3.6 c). Another close-up view for March 1, TS4 (Fig. 3.6 d) is shown in Fig. 3.7.

Fig. 3.7 shows a detail for TS4, March 1. Convergence for sensors 0 to 25 shows simultaneous equalization performance for the hottest part of the day (Fig. 3.7 a). However, better performance at a lower depth was seen in TS1, down to 25 feet while TS4 oscillates at only 15 feet below ground; the deepest sensor to drop temperature was 20 instead (Fig. 3.7 b). Divergence begins as the temperatures return to deep below ground temperature

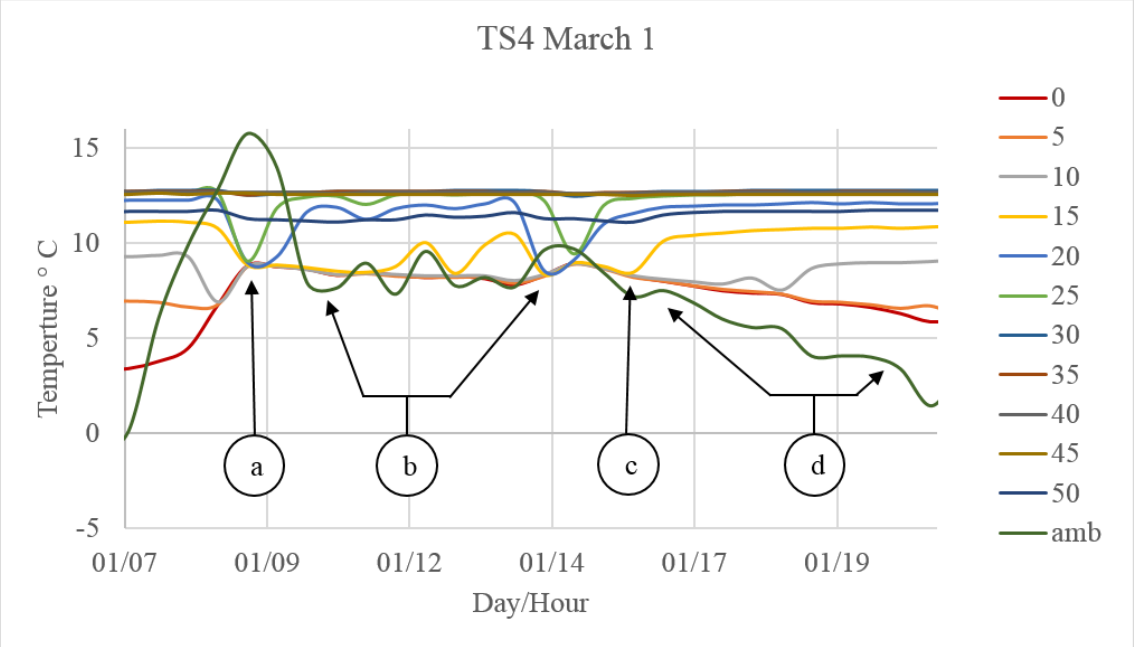

Figure 3.7. Detail view of TS4, March 1. a: Convergence for sensors 0 to 25 shows simultaneous equalization performance for the hottest part of the day. b: TS4 oscillates at only 15 feet below ground and the lowest sensor to drop was 20. c: Divergence begins as the temperatures return to deep below ground temperature. d: Sensor lines 15 and 10 both lowering overall temperature slightly over this time interval.

with sensor lines 15 and 10 both lowering overall temperature slightly over this time interval (Fig. 3.7 c and d).

A long period of inactivity can be seen for both thermosiphons March 3 to 4, during which data are presented in 15-minute intervals to describe activity (Fig. 3.8 a and 3.9 a). It was discovered the large underground vapor line for TS1 was closed, and it was opened to allow better vapor circulation (Fig. 3.8 and Fig. 3.9 event #4). Sharp spikes of activity can be seen in the 10 foot sensor for both thermosiphons (Fig. 3.8 b), although less so for TS4 (Fig. 3.9 b), in which the vapor valve had been opened initially. These transient spikes of cold could be condensate in the buried 1-inch HDPE vapor pipes falling back into the

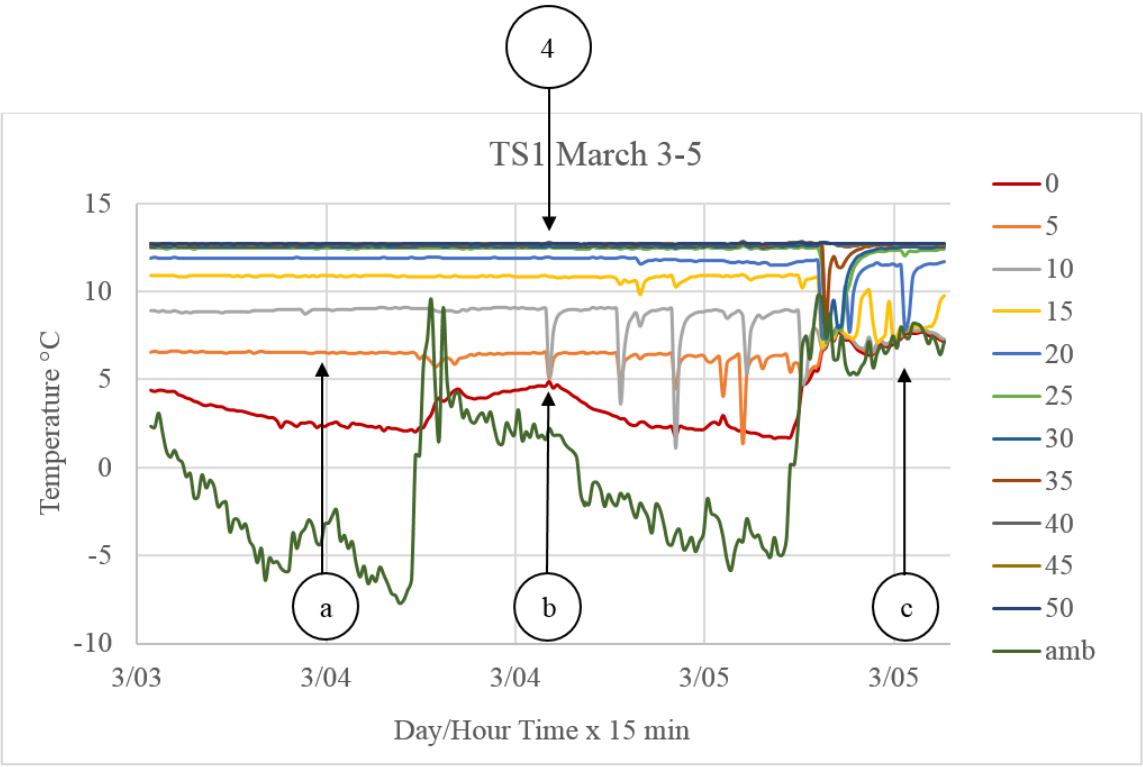

Figure 3.8. TS1 operation March 3–5. #4 Vapor line for TS1 is opened to initiate circulation. a: Long period of inactivity in TS1. b: Transient spikes of cold for sensor 10 could be condensate from vapor pipes. c: Characteristic activity restarts with daytime temperatures for March 5.

Figure 3.9. TS4 operation March 3–5. #4 Vapor line for TS1 is opened to initiate circulation and condensate runs into TS4 as well. a: Long period of inactivity in TS4. b: Sharp spike of cold for sensor 10 could be condensate from vapor pipes. c: Characteristic activity restarts with daytime temperatures for March 5.

thermosiphons. It is hypothesized that pressure gradients allow activity from TS1 to be shared with TS4 through the opened vapor pipes. Both thermosiphons regain characteristic convergence activity during the warmest part of the day for March 5 (Fig. 3.8 c and Fig. 3.9 c).

All previous data collected from the Arduino SD card had never shown any activity for the pumps, which always read as 'off'. On March 4, the Arduino program was changed to display data every second to see if the pumps were ever read as 'on', in-between the one minute intervals in which data were collected. No response was seen with this change (Table 3.1 event #5) and it was believed the two diodes between the pumps and the data

acquisition may be preventing the pumps from achieving a full signal. Again on March 4, the pump fuse for TS4 was discovered to be broken but not burnt out, and it was removed along with the diode connected to it. Additional tests were performed to investigate whether the pump for TS4 would then turn on without the second diode, but no activity for the pump was found in the program data. It is believed that the pumps were working without communicating to the program, since there was response for the pump float switches on the day the thermosiphons were originally charged, and again on March 4 when tested.

On March 6, the vapor lines for TS1 and TS4 were both closed in order to contain vapor (Table 3.1 and Fig. 3.10 event #6), and in preparation to re-charge TS1 (Table 3.1 and Fig. 3.10 event #7). The return liquid line for TS4 was then opened fully from the previous ¼ turn since it showed lack of fluid (Table 3.1 and Fig. 3.10 event #8). An immediate response was seen for TS4, and characteristic activity restarted with sharp drops in temperature between 45 and 50 feet (Fig. 3.10 a) indicating liquid pooling at the bottom. Complete convergence is seen for TS4 for the next three days as TS4 continues to operate with similar performance, allowing sensors 15 and 10 to cool slightly overall (Fig. 3.10 b and d). Condensation activity is seen for sensors 0 and 5 (Fig 3.10 c). Unfortunately, all temperature sensors for TS1 malfunctioned after March 6, and no additional temperature data from that thermosiphon was ever collected (Table 3.1 event #9). In addition, temperature sensors 0, 25 and 50 for TS4 also malfunctioned, reading -127 °C for the successive duration of data collection (Table 3.1 event #10). Data for all following graphs was changed from -127 °C from sensors 0, 25 and 50 to averaged values of neighboring sensors giving decreased accuracy in the readings after March 6 up through April 22.

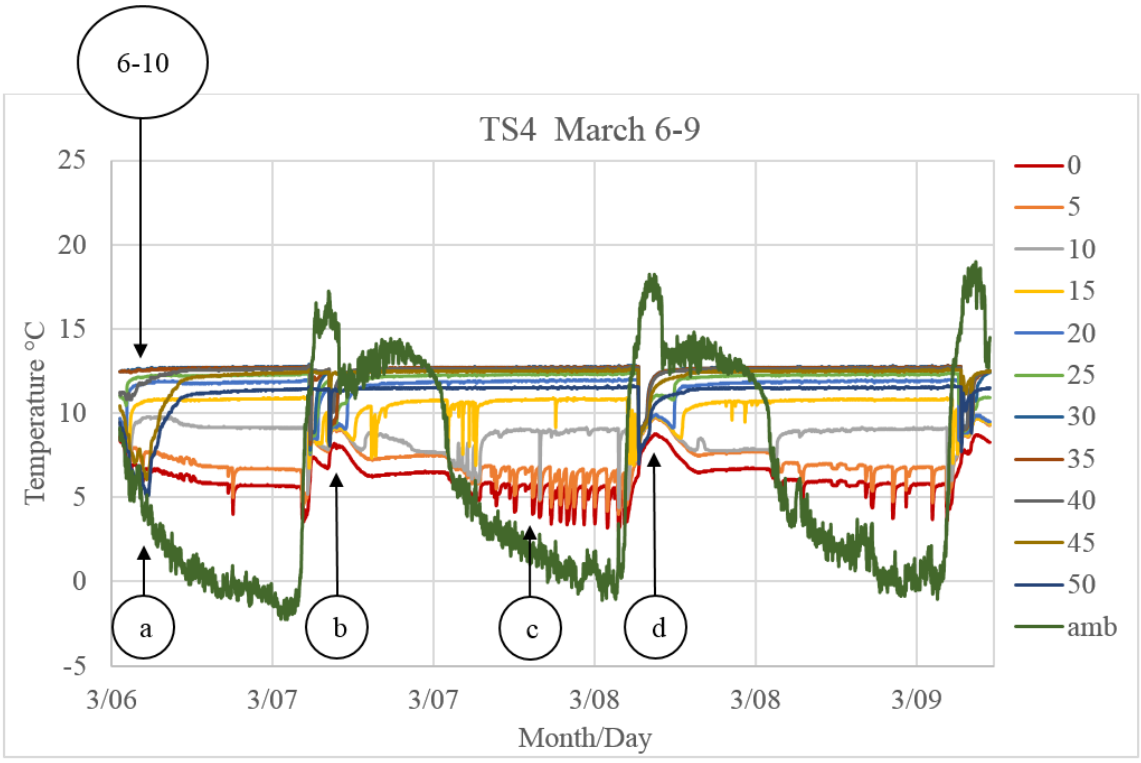

Figure 3.10. TS4 operation March 6 – 9. #6 TS1 and TS4 vapor lines closed. #7. Re-charge TS1. #8. TS4 liquid line opened fully from ¼ turn. #9. TS1 temperature sensor line malfunctioned. #10. TS4 sensors 0, 25 and 50 lost. a: Characteristic activity continues in TS4 after re-charging TS1. b: Characteristic partial convergence during daytime. c: Condensation occurring at sensors 0 and 5. d: Complete convergence continues.

A revival of convergence activity occurred on March 10 when all thermosiphon liquid valves except TS1 and TS4 were closed in an effort to isolate fluid movement to these thermosiphons (Fig 3.11 event #11). No significant changes from previous activity were observed for TS4 for the duration of the experiment until March 15 when the fluid appears o dry out and temperatures stabilize with fluctuations terminating (Fig 3.11 a). After March 15, all sensor fluctuation activity terminates (Fig. 3.11 a). The fans were turned off on March 20 and no changes were observed in the data (Fig. 3.12 event #12). Fig. 3.12 shows the temperature sensors reaction without the presence of fluid. There are three possibilities for the state of the fluid in the experiment shown in Fig. 3.12:

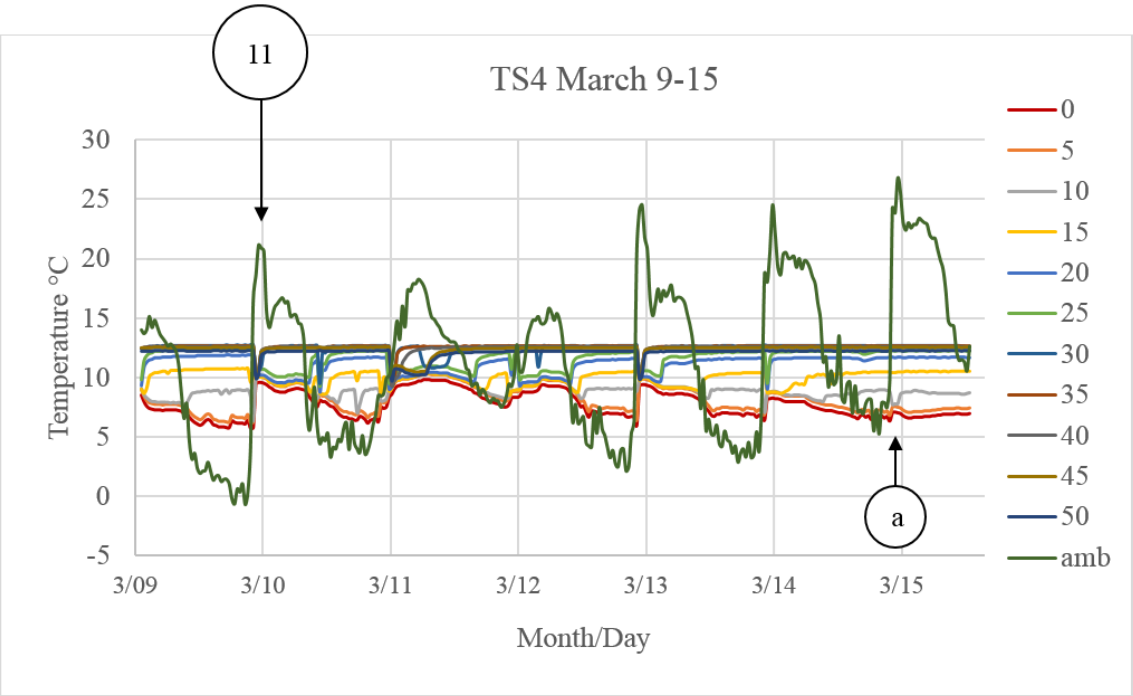

Figure 3.11. #11. TS4 operation March 9 - 15. All TS liquid lines closed except for TS1 and TS4 to isolate liquid to these thermosiphons. a: TS4 continues to function as before with decreased performance until March 15.

1. the fluid has completely evaporated
2. the fluid is trapped in the above-ground tank
3. the fluid is trapped inside the other 5 thermosiphons that do not have temperature sensors

Possibility number 1 would not be ideal for the operation of the system, with possibility number 2 being the best outcome.

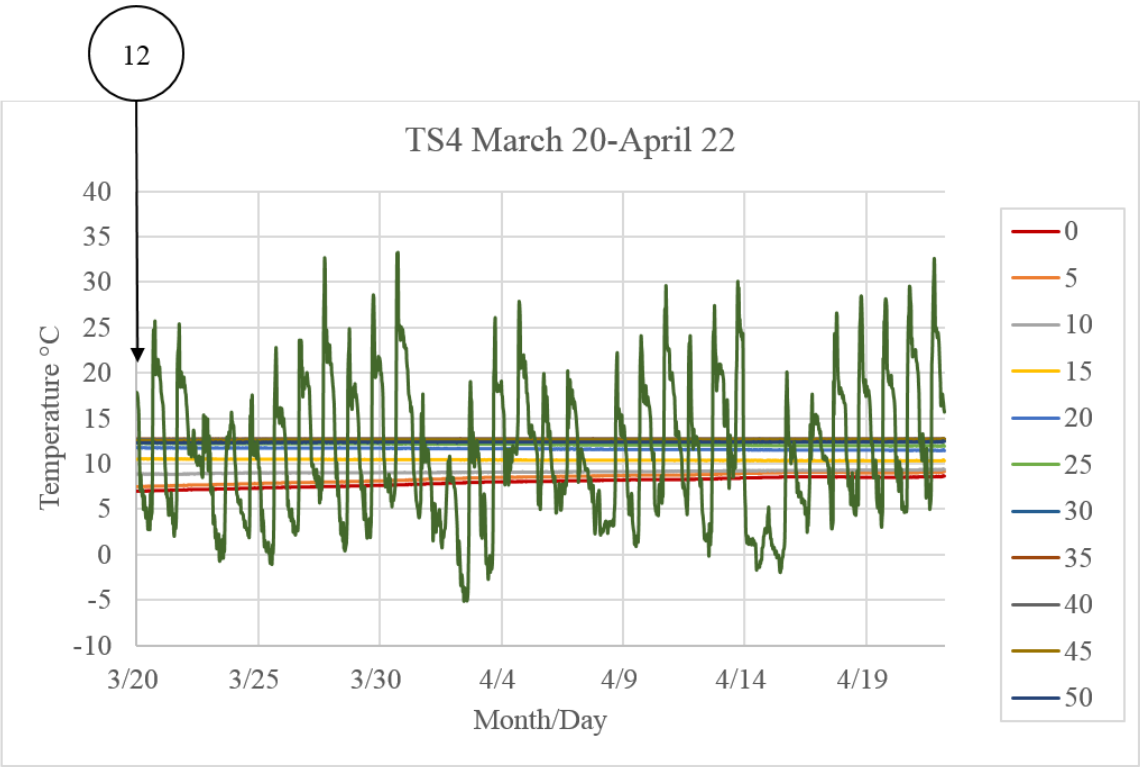

Figure 3.12. TS4 operation March 20 – April 22. #12: Fans are turned off, no reaction is observed. Temperature variation for sensors terminates permanently.

3.2 Soil Temperature Data

Near-surface soil temperatures fluctuate due to solar radiation, convective heat transfer from the air, and latent energies associated with snow melting and soil water vaporization. The complex transport processes that govern the thermal properties of soil are affected by moisture and organic content. Radiation is an important factor for regions at ground level, convection is applicable to surface soils, but conduction is the dominant process for all soils [13].

Ground temperature data proved to be valuable for finding the thermal diffusivity of the soil at the Sill Center site. Measured ground temperatures for three months were compiled into one chart for subsequent calculations in Fig. 3.13. There is a cooling effect

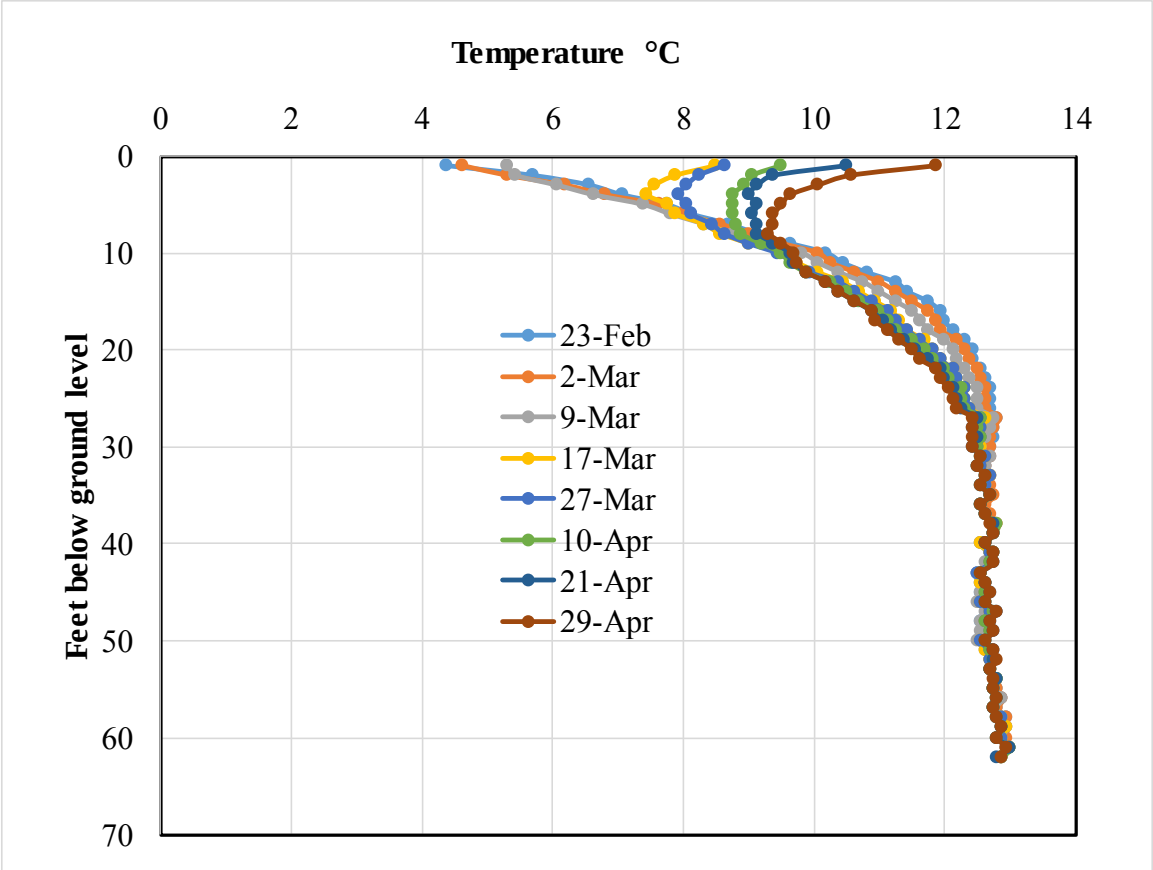

Figure 3.13. Measured ground temperature profiles.

shown in the figure for the soil at a depth of 10 to 30 feet below ground level. This is believed to be due to conduction in the soil driven by surface convective cooling.

Salt Lake City Airport data were used to estimate the average high and low temperatures over one year (Fig. 3.14) [15]. The average high was taken at 80°F (26.9°C) while the average low temperature was 29°F (-1.9°C). The airport temperature data were used because they were consistent data during the specific time period needed, and the elevation difference was considered negligible.

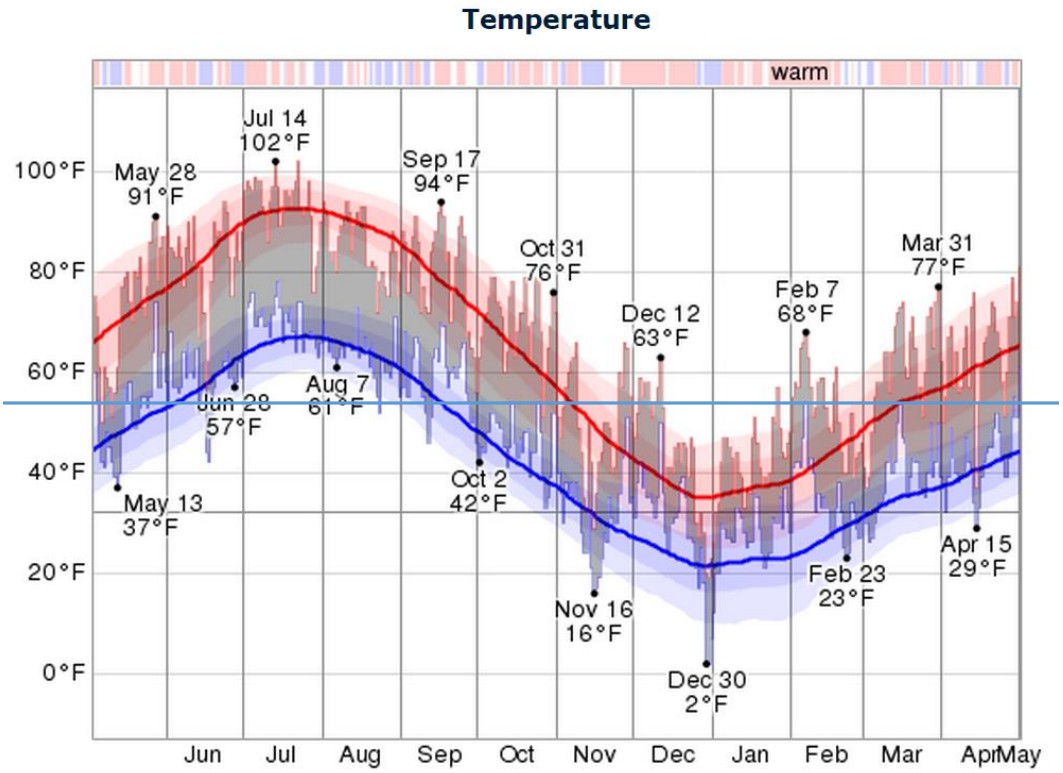

Figure 3.14. Salt Lake City Airport temperatures.

3.3 Periodic Heating

The concept of periodic heating was used to calculate the thermal diffusivity. Ambient temperatures shown in Fig. 3.14 are a good example of periodic heating as the temperature varies about a mean value. Air temperatures have an annual and diurnal cycle and are effected by seasons and by meteorological events [13]. Cyclic solar radiation is largely responsible for both the annual and diurnal temperature changes observed. Irradiation first hits the soil and its heat must then diffuse through the layers of soil to change the ground temperature. This causes amplitude damping and results in a reduction of heat flux as depth is increased. A time lag is also observed when measuring temperatures at any certain depth in the soil, and becomes more pronounced with increased travel

downwards. Thus, soil temperature may be treated as a time-dependent sinusoidal function that oscillates around the mean temperature. This mean soil temperature is equated to the average air temperature over a year, and also happens to be the year-round actual ground temperature below 25 to 30 feet.

The subsurface temperature, as a function of both time and depth, can be obtained by solving the heat equation below:

$$\frac{1}{\alpha}\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial z^2} \tag{3.1}$$

with:

$$\alpha = \frac{k}{c_p \rho} \tag{3.2}$$

where:

c_p is the soil specific heat capacity [J/kg K]

ρ is the soil density [kg/m^3]

α is the thermal diffusivity [m^2/s]

If the ground is considered to be a semi-infinite solid, the surface temperature history may be described as [14, 16]:

$$T(0,t) = T_i + \Delta T sin\omega t \tag{3.3}$$

With the boundary condition:

$$T(x \rightarrow \infty, t) = T_i \tag{3.4}$$

Then the solution to Eq. 3.1 above is:

$$\frac{T(x,t)-T_i}{\Delta T} = \exp\left[-z\sqrt{\omega/2\alpha}\right]\sin[\omega t - z\sqrt{\omega/2\alpha}] \tag{3.5}$$

Given the frequency of:

$$\omega = 2\pi/P \tag{3.6}$$

where the period, P is 365 days or 3.1536 x 10^7 seconds.

The calculated temperatures of Fig. 3.15 were obtained from $T(x,t)$ in Eq. 3.5 above. The time was set to zero at April 29 to produce the graph in Fig 3.15. The best-fit diffusivity was 5.0 E-7 m^2/s, slightly above the values found in the literature [13]. With this value and tabulated values for the density and specific heat capacity from the literature, [16] it is possible to solve for the thermal conduction of the soil, k in units of [W/mK] with Eq. 3.2 and also the heat flux at the surface by applying Fourier's law:

$$q"(t) = k\Delta T\sqrt{\omega/2\alpha}\,\sin(\omega t + \pi/4) \tag{3.7}$$

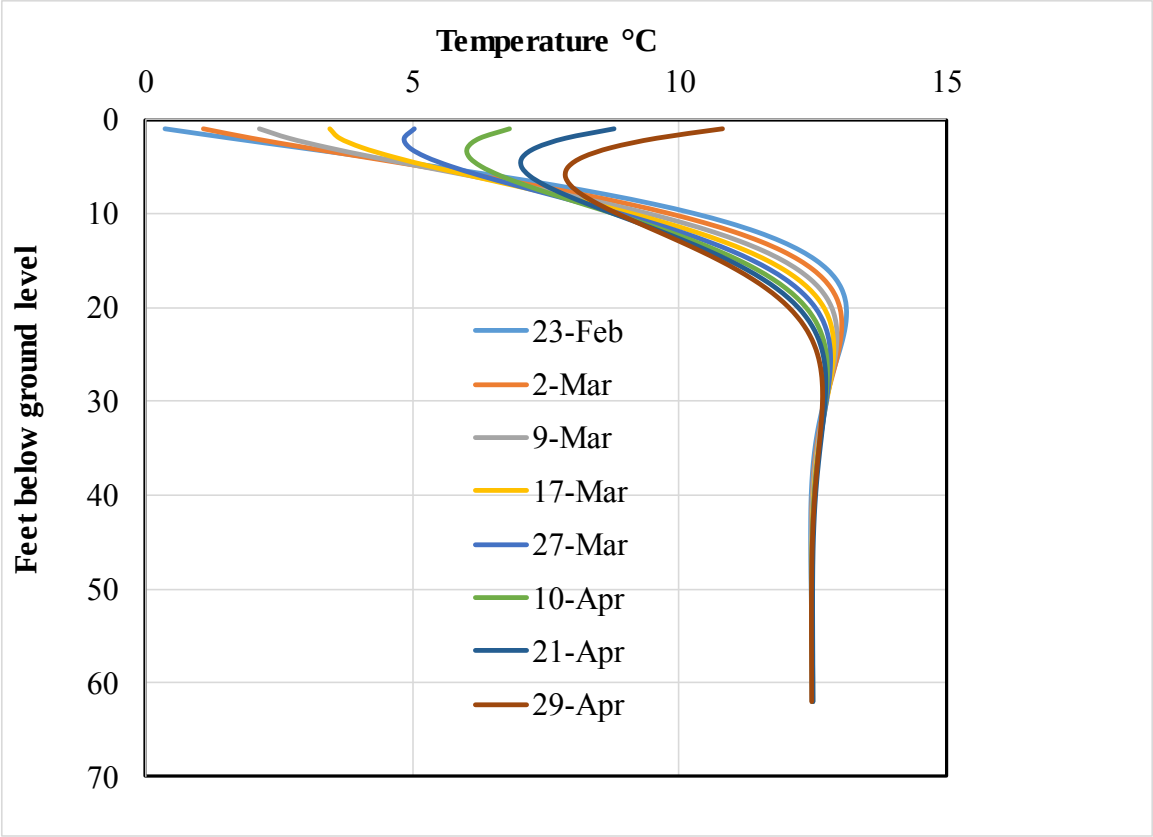

Figure 3.15. Calculated ground temperatures versus depth using Eq.3.5.

In comparing Figs. 3.13 and 3.15, the analytical solution produces a very good prediction of the subsurface temperatures for most days. The exception is the 17-Mar comparison when measured surface temperatures were much larger than those predicted. The reason for this discrepancy is clear with a closer examination of Fig. 3.14; the ambient temperatures during the month of February were the warmest on record in Salt Lake City. And after a short term cooling, the end of February and beginning of March, the temperatures again reached values above historical highs in the days leading to the 17-Mar temperature log.

Finally, predictions of the temperature profiles at times throughout the full year are of interest as the baseline thermal field is defined. Using the solution of Eq. 3.5, the thermal

diffusivity found to best match the first few profiles presented in this work is calculated. With the sinusoidal function of Eq. 3.5 representing the average surface temperature throughout the year, the profiles shown in Fig. 3.16 are obtained. The profiles are essentially monthly distributions. Validation of this prediction by field data will allow further definition of the baseline conditions of the site. These predictions also give good indication of appropriate times to start and stop different modes of operation.

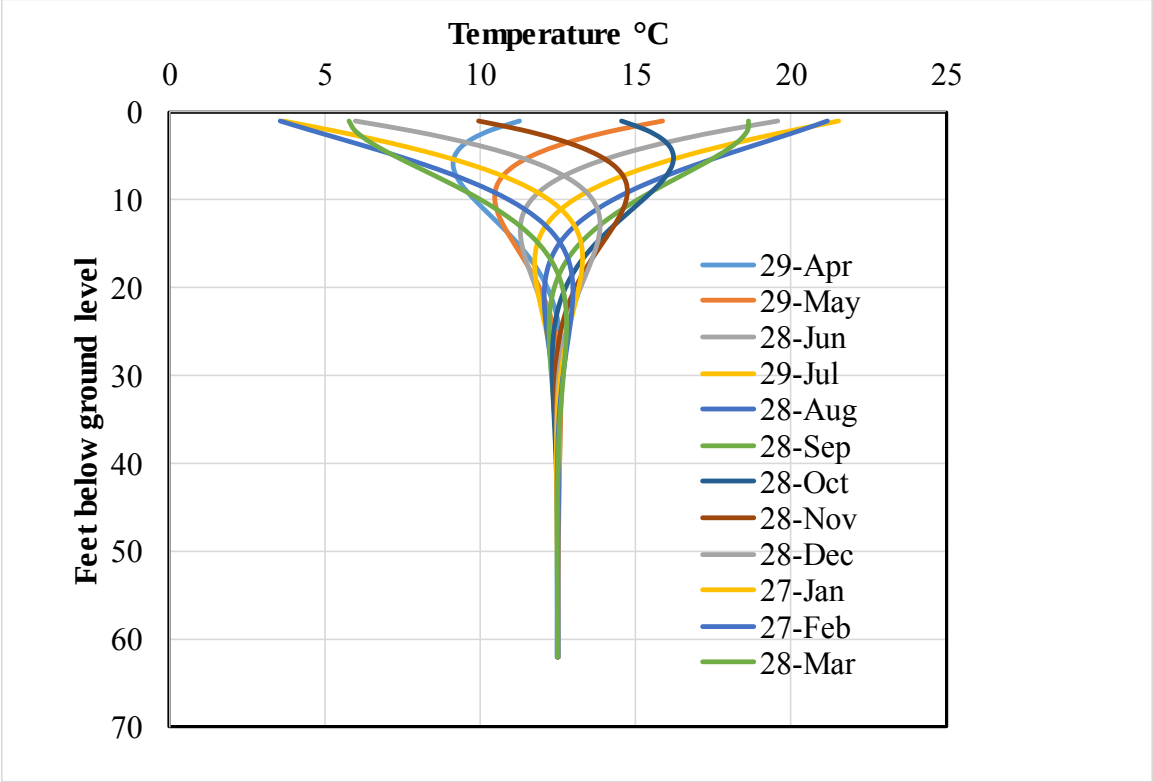

Figure 3.16. Calculated monthly ground temperatures versus depth using Eq. 3.5.

CHAPTER 4

DISCUSSION AND FUTURE WORK

4.1 Discussion

Overall the thermosiphons did show an ability to function as expected, however only for short periods of time. Much of the data showed a sudden cooling for the sensors but a subsequent increase in temperature as liquid flows decreased and vaporization slowed. Convergence of all thermosiphon temperatures, indicating thermodynamic equilibrium and the presence of both liquid and vapor phases at a location, was observed. Liquid scarcity only intermittently allowed for the expected behavior for all sensors to converge to the same temperature and hold a cooling temperature that could potentially chill the soil surrounding the thermosiphons. This evidence of condensation cooling can be seen in Fig. 3.7 d when sensors 10 and 15 rebound to a lower temperature on that time interval; however, the decrease is only a few degrees. Convergence or equalization of most temperature sensors was often seen when ambient temperatures approached or exceeded the constant deep underground temperature. This semi-beneficial behavior would not be expected during a cold winter as shown by the data for cold days. Complete convergence where all temperature sensors equalized to the same temperature was only seen when there was refrigerant at the bottom of the pipe. As seen in the figures with sharp spikes of decreased temperature, it is hypothesized that vapor was condensing in the 1-inch HDPE

vapor lines that circle the array in the shallow soils, with the condensate draining back into the thermosiphon causing the transient spikes of decreased temperature. From the data it is clear there was insufficient fluid inside the thermosiphons to distribute liquids in the wick and to fill the bottom to the float switch level. For the piping configuration in place, a pump should be added above-ground to increase the refrigerant flow back to the field from the storage tank. It is believed that the head initially calculated to be satisfactory for passive flow back to the field was insufficient due to the vapor back-pressure in the thermosiphon inhibiting return liquid flow. In addition, better control of the fluid and vapor may be attained if the data from this work are used to optimize the throttling of both the liquid and vapor return lines as well as the subterranean vapor pipes.

The value found for the thermal diffusivity, 5.0 E-07, was twice that expected. This value resulted in a thermal conductivity of 1.9 W/m-K if a density of 2050 kg/m^3 and a specific heat of 1840 J/kg-K are accurate. This value of thermal conductivity is almost four times the size of 0.52 W/m-K reported in [16]. There are many variations that lead to this discrepancy. Since the diffusivity was estimated from temperature measurements that matched the analytical solution, the validity of the analytical solution might be questioned. Ambient temperatures were abnormally high during the winter of 2015, and therefore the calculated average surface temperatures were much warmer than the temperatures predicted by Eq. 3.5. Also, for this comparison, the radiative flux was considered negligible and the diffusivity was based on the penetration depth of the surface cooling during periods before the time of abnormally high temperatures. Abnormal surface temperatures during late winter would not account for observed deep penetration depth of the cooling. Thus, the estimated thermal diffusivity is a reasonable starting value.

4.2 Future Work

Opportunities for future work are abundant in this project due to its complicated nature and various modes of operation. The winter season during this study did not provide enough consecutive cold days to produce ample freezing in the ground. As such, a colder winter would give new results. Many improvements to the systems were identified in the course of this project. The pumps at the bottom of the thermosiphons never indicated they were working from the Arduino program and therefore need inspection. The temperature sensors in all thermosiphons began to malfunction and must now be replaced. It is possible that these sensors are not compatible with the R-134a refrigerant or that they malfunctioned due to moisture or a short in the wires. While replacing the temperature sensors it is suggested they be accompanied by pressure transducers so the vaporization and condensation activity may be accurately studied and verified. Flow meters on the vapor lines connecting the above-ground equipment to the gas phase in the thermosiphon should be installed for additional thermodynamic data. It was discovered that insufficient fluid was returning from the above-ground storage tank. With this, the installation of a metered pump is suggested to increase control of the liquid to and from the field. Finally, it is suggested to replace the R-134a refrigerant with a more environmentally safe fluid.

Identification of the problems encountered during operation of this complex, field-sized demonstration are highly valuable. Once these minor improvements to the data acquisition and fluid delivery systems are incorporated, this project can bring many future students an opportunity to learn about thermodynamics, heat transfer and renewable energy.

CHAPTER 5

CONCLUSIONS

5.1 Conclusions

The history and construction of the Sill Center test site spanning many years is presented in this work. The Arduino data acquisition system is functional and ready to be fitted with new temperature sensors. The above-ground thermosiphon condensers and tanks at the Sill Center were tested and proven to be free of leaks. In addition to these foundational conclusions, specific conclusions were also drawn:

1. The *in situ* thermal diffusivity is approximately 5.0 E-07 m^2/s.
2. Heat transfer from the soil to the thermosiphon occurred as expected. Liquids vaporized in the wick and condensed in either the above-ground condenser or the upper zones of the system within the chilled shallow soils.
3. Condensates created in the buried vapor lines drain back to the vertical thermosiphons causing transient drops in temperature.
4. An installed pump would provide additional head needed to allow liquid condensate to flow back to the individual thermosiphons.
5. The DS18B20 sensors used to measure temperatures in the thermosiphons fail in the thermosiphon environment.

6. Flow meters installed between the vapor lines and thermosiphons would augment the measurement of the system.
7. Investigations into the use of superior working fluids would benefit the use of the system.

With climate change awareness reaching unprecedented levels at the present time, it is clear that efforts should be made to reduce the production of greenhouse gasses. Given the high mass of carbon dioxide produced by commercial and residential cooling systems, it is clear that therein lies an opportunity to reduce emissions in these systems. Many unexploited opportunities exist in subterranean seasonal energy storage, especially with the use of smart thermosiphon arrays that can reduce carbon dioxide emissions.

APPENDIX A

MATLAB CODE FOR THERMOSIPHON VERTICAL DEVIATION

```
% Thermosiphon_3d_Plot.m :
% use this program to see the 19 thermosiphon's deviation from the
vertical.
% by Mike Beeman 2012
% Edited by Harvest Montemayor 2014
clc
close all
[Filename, Pathname, FilterIndex] = uigetfile('*.xlsx','Choose the file
named Thermosiphon Angles';
FilenamePath = [Pathname,Filename];
[num,Headers,raw] = xlsread(FilenamePath, 'Sheet1');
for i = 1:length(num(:,1))
  [xx,yy] = pol2cart(num(i,6)*pi/180,num(i,7));

%  plot(x(1),y(1),'ro')% plots 1-18 bottom location in red (viewed from
TOP plane)
  x(1) = xx;
  y(1) = yy;
  z(1) = 0;

 plot(x(1),y(1),'ro')% plots 1-19 top location in red (viewed from TOP
plane)

  [x(2),y(2),z(2)] = sph2cart(num(i,4)*pi/180,atan(-36/num(i,5)),50);
  x(2) = x(2) + xx;
  y(2) = y(2) + yy;

  plot3(x,y,z);
  % print to graph the thermosiphon #s
  labels = num2str((num(i,1))','%d');
  text(x(1), y(1), labels, 'horizontal','left', 'vertical','bottom')
  % print to graph the deviation distance
  labels2 = num2str((num(i,5))','%G');
  text(x(1), y(1), labels2, 'horizontal','right', 'vertical','top')
  hold on

end
grid on
xlabel('X')
ylabel('Y')
zlabel('Z')
```

APPENDIX B

THERMOSIPHON DEVIATION ANGLES

TS #	Deviation over 36"	Mag Angle from North	True Angle from North	Deviation at bottom (ft)	Ground True Angle Position	Ground True Radial Position
1	1	220	232	1.36	0	0
2	0.875	240	252	1.19	0	5
3	0.75	215	227	1.02	51.4285714	5
4	0.375	12	24	0.51	102.857143	5
5	0.625	180	192	0.85	154.285714	5
6	0.375	170	182	0.51	205.714286	5
7	1.25	135	147	1.70	257.142857	5
8	1.1875	80	92	1.62	308.571429	5
9	1.75	75	87	2.38	0	10
10					32.7272727	10
11	2	246	258	1.94	65.4545455	10
12	0.3125	230	242	0.43	98.1818182	10
13	1.125	355	367	1.53	130.909091	10
14	2	240	252	2.72	163.636364	10
15	0.875	223	235	1.19	196.363636	10
16	0.5	348	360	0.68	229.090909	10
17	0.6875	10	22	0.94	261.818182	10
18	0.875	122	134	1.19	294.545455	10
19	1.5625	192	204	2.13	327.272727	10

APPENDIX C

ARDUINO CODE

```
// Thermosiphon_Data_Logger
// Compiled by Mike Beeman 2012
// Edited by Harvest Montemayor 2014

#include <OneWire.h>               //Open OneWire Library
#include <DallasTemperature.h>     //Open DallasTemperature Library
#include <SD.h>                    //Open SD Library
#include <Wire.h>                  //Open Wire Library
#include <RTClib.h>                //Open Rtc_DS3231 Library
#include <SPI.h>
// how many milliseconds between grabbing data and logging it. 1000 ms
// is once a second
#define LOG_INTERVAL  60000 // mills between entries (reduce to take
// more/faster data)

// echo data to serial port
#define ECHO_TO_SERIAL   1

// Data wire is plugged into port X, X, X, X, X on the Arduino
//++++Need to determine.++++
#define ONE_WIRE_BUS_A  41
#define ONE_WIRE_BUS_B  42
#define ONE_WIRE_BUS_C  43
#define ONE_WIRE_BUS_D  44
#define ONE_WIRE_BUS_E  45
#define ONE_WIRE_BUS_F  46
#define ONE_WIRE_BUS_G  47                      //Not   currently used

// MISO = 50;                                   // SPI Pins as a reminder
// MOSI = 51;
// SCK = 52;
// SS = 53;
// SDA = 20;                                    // I2C Pins as a reminder
// SCL = 21;

// how many milliseconds before writing the logged data permanently to
// disk
// set it to the LOG_INTERVAL to write each time (safest)
// set it to 10*LOG_INTERVAL to write all data every 10 datareads, you
// could lose up to
// the last 10 reads if power is lost but it uses less power and is
```

```
// much faster!
#define SYNC_INTERVAL 1000 // mills between calls to flush() - to write
// data to the card
uint32_t syncTime = 0; // time of last sync()

//init the real time clock
RTC_DS1307 RTC;

// the logging file
File logfile;

// Setup a oneWire instance to communicate with any OneWire devices
//(not just Maxim/Dallas temperature ICs)
OneWire oneWireA(ONE_WIRE_BUS_A);
OneWire oneWireB(ONE_WIRE_BUS_B);
OneWire oneWireC(ONE_WIRE_BUS_C);
OneWire oneWireD(ONE_WIRE_BUS_D);
OneWire oneWireE(ONE_WIRE_BUS_E);
OneWire oneWireF(ONE_WIRE_BUS_F);

// Pass our oneWire reference to Dallas Temperature. (Data Type
// DallasTemperature)
DallasTemperature sensorsA(&oneWireA);
DallasTemperature sensorsB(&oneWireB);
DallasTemperature sensorsC(&oneWireC);
DallasTemperature sensorsD(&oneWireD);
DallasTemperature sensorsE(&oneWireE);
DallasTemperature sensorsF(&oneWireF);

// arrays to hold device addresses
uint8_t devicesA [60][8];
uint8_t devicesB [60][8];
uint8_t devicesC [60][8];
uint8_t devicesD [60][8];
uint8_t devicesE [60][8];
uint8_t devicesF [60][8];

// for the Arduino Mega, we use digital pin 10 for the SD cs line
// const int chipSelect = 53;  //++++Need to determine.++++

// Pump Pin number
const int pump[19] =
{22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40};

// SETUP
////////////////////////////////////////////////////////////////////////

void setup(void)
{

//store the number of sensors to the variable numSensors
sensorsA.begin();
const int numSensorsA = sensorsA.getDeviceCount();
sensorsB.begin();
const int numSensorsB = sensorsB.getDeviceCount();
```

```
sensorsC.begin();
const int numSensorsC = sensorsC.getDeviceCount();
sensorsD.begin();
const int numSensorsD = sensorsD.getDeviceCount();
sensorsE.begin();
const int numSensorsE = sensorsE.getDeviceCount();
sensorsF.begin();
const int numSensorsF = sensorsF.getDeviceCount();
const int totalnumSensors = numSensorsA + numSensorsB + numSensorsC +
numSensorsD + numSensorsE + numSensorsF;

//Set pin mode for pump on/off
for (uint8_t i=0; i < 19; i++) {
pinMode(pump[i], INPUT);
}

  pinMode(chipSelect, OUTPUT);

/*++++++++++++++++++++++++++
Real Time Clock
++++++++++++++++++++++++++
*/
  // start serial port
  Serial.begin(9600);
  // connect to RTC
  Wire.begin();
  RTC.begin();

  // Set time to computer time
  RTC.adjust(DateTime(__DATE__, __TIME__));

/*++++++++++++++++++++++++++++++++
  Initialize Thermometers Sensors
  ++++++++++++++++++++++++++++++++
*/
  // Start up the library
  sensorsA.begin();
  sensorsB.begin();
  sensorsC.begin();
  sensorsD.begin();
  sensorsE.begin();
  sensorsF.begin();

  // locate devices on the buses
  Serial.print("Locating devices...");
  Serial.print("Found ");
  Serial.print(numSensorsA, DEC);
  Serial.print(",");
  Serial.print(numSensorsB, DEC);
  Serial.print(",");
  Serial.print(numSensorsC, DEC);
  Serial.print(",");
  Serial.print(numSensorsD, DEC);
  Serial.print(",");
  Serial.print(numSensorsE, DEC);
  Serial.print(",");
```

```
  Serial.print(numSensorsF, DEC);
  Serial.println(" devices on Thermosiphon 1, 5, 6, 14 and Ground 1 and
2.");

  // report parasite power requirements
  Serial.print("Parasite power is: ");
  if (sensorsA.isParasitePowerMode()) Serial.println("ON");
  else Serial.println("OFF");

//Check to see if devices can be accessed
 for (uint8_t i = 0; i < numSensorsA; i++) {
  if (!sensorsA.getAddress(devicesA[i], i)) Serial.println("Unable to
find address for Device on array A "); Serial.println(i);
}
for (uint8_t i = 0; i < numSensorsB; i++) {
  if (!sensorsB.getAddress(devicesB[i], i)) Serial.println("Unable to
find address for Device on array B"); Serial.println(i);
}
for (uint8_t i = 0; i < numSensorsC; i++) {
  if (!sensorsC.getAddress(devicesC[i], i)) Serial.println("Unable to
find address for Device on array C"); Serial.println(i);
}
for (uint8_t i = 0; i < numSensorsD; i++) {
  if (!sensorsD.getAddress(devicesD[i], i)) Serial.println("Unable to
find address for Device on array D"); Serial.println(i);
}
for (uint8_t i = 0; i < numSensorsE; i++) {
  if (!sensorsE.getAddress(devicesE[i], i)) Serial.println("Unable to
find address for Device on array E"); Serial.println(i);
}
for (uint8_t i = 0; i < numSensorsF; i++) {
  if (!sensorsF.getAddress(devicesF[i], i)) Serial.println("Unable to
find address for Device on array F"); Serial.println(i);
}

  // show the addresses we found on the bus
  for (uint8_t i = 0; i < numSensorsA; i++) {
    Serial.print("Device Array A");
    Serial.print( i+1 );
    Serial.print(" Address: ");
    printAddress(devicesA[i]);
    Serial.println();
  }
  for (uint8_t i = 0; i < numSensorsB; i++) {
     Serial.print("Device Array B");
    Serial.print( i+1 );
    Serial.print(" Address: ");
    printAddress(devicesB[i]);
    Serial.println();
  }
  for (uint8_t i = 0; i < numSensorsC; i++) {
    Serial.print("Device Array C");
    Serial.print( i+1 );
    Serial.print(" Address: ");
    printAddress(devicesC[i]);
    Serial.println();
```

```
  }
  for (uint8_t i = 0; i < numSensorsD; i++) {
    Serial.print("Device Array D");
    Serial.print( i+1 );
    Serial.print(" Address: ");
    printAddress(devicesD[i]);
    Serial.println();
  }
  for (uint8_t i = 0; i < numSensorsE; i++) {
    Serial.print("Device Array E");
    Serial.print( i+1 );
    Serial.print(" Address: ");
    printAddress(devicesE[i]);
    Serial.println();
  }
  for (uint8_t i = 0; i < numSensorsF; i++) {
    Serial.print("Device Array F");
    Serial.print( i+1 );
    Serial.print(" Address: ");
    printAddress(devicesF[i]);
    Serial.println();
  }

/*++++++++++++++++++++++++++++++++++
  Initialize SD Card Files
  ++++++++++++++++++++++++++++++++++
*/
  // initialize the SD card
  Serial.print("Initializing SD card...");
  // make sure that the default chip select pin is set to
  // output, even if you don't use it:


    // see if the card is present and can be initialized:
  if (!SD.begin(chipSelect)) {
    error("Card failed, or not present");
  }
  Serial.println("card initialized.");

//  File logfile = SD.open("datalog.txt");
//
//  // if the file is available, write to it:
//  if (logfile) {
//    while (logfile.available()) {
//      Serial.write(logfile.read());
//    }
//    logfile.close();
//  }
//  // if the file isn't open, pop up an error:
//  else {
//    Serial.println("error opening datalog.txt");
//  }

// create a new file or create an error message if not found
char filename[] = "LOGGER00.CSV";
  for (uint8_t i = 0; i < 100; i++) {
```

```
    filename[6] = i/10 + '0';
    filename[7] = i%10 + '0';
    if (! SD.exists(filename)) {
      // only open a new file if it doesn't exist
      logfile = SD.open(filename, FILE_WRITE);
      break;  // leave the loop!
    }
  }
  if (! logfile) {
    error("Could not create file");
  }

  //Write headers to logfile
  Serial.print("Logging to: ");
  Serial.println(filename);
  logfile.print("Millis,Month,Day,Year,Hour,Minute,Second,Pump 1
On/Off,Pump 2 On/Off,Pump 3 On/Off,Pump 4 On/Off,Pump 5 On/Off,Pump 6
On/Off,Pump 7 On/Off,Pump 8 On/Off,Pump 9 On/Off,Pump 10 On/Off,Pump 11
On/Off,Pump 12 On/Off,Pump 13 On/Off,Pump 14 On/Off,Pump 15 On/Off,Pump
16 On/Off,Pump 17 On/Off,Pump 18 On/Off,Pump 19 On/Off,");
  for (uint8_t i = 1; i <= numSensorsA; i++) {
    logfile.print("Temperature (deg C) Array A");
    logfile.print(i);
    logfile.print(", ");
  }
  for (uint8_t i = 1; i <= numSensorsB; i++) {
    logfile.print("Temperature (deg C) Array B");
    logfile.print(i);
    logfile.print(", ");
  }
  for (uint8_t i = 1; i <= numSensorsC; i++) {
    logfile.print("Temperature (deg C)  Array C");
    logfile.print(i);
    logfile.print(", ");
  }
  for (uint8_t i = 1; i <= numSensorsD; i++) {
    logfile.print("Temperature (deg C) Array D");
    logfile.print(i);
    logfile.print(", ");
  }
  for (uint8_t i = 1; i <= numSensorsE; i++) {
    logfile.print("Temperature (deg C)  Array E");
    logfile.print(i);
    logfile.print(", ");
  }
  for (uint8_t i = 1; i <= numSensorsF; i++) {
    logfile.print("Temperature (deg C)  Array F");
    logfile.print(i);
    logfile.print(", ");
  }

  logfile.println();
  logfile.flush();

  if (ECHO_TO_SERIAL)
```

```
  Serial.println("Time Stamp,Pump 1 On/Off,Pump 2 On/Off,Pump 3
On/Off,Pump 4 On/Off,Pump 5 On/Off,Pump 6 On/Off,Pump 7 On/Off,Pump 8
On/Off,Pump 9 On/Off,Pump 10 On/Off,Pump 11 On/Off,Pump 12 On/Off,Pump
13 On/Off,Pump 14 On/Off,Pump 15 On/Off,Pump 16 On/Off,Pump 17
On/Off,Pump 18 On/Off,Pump 19 On/Off");

}

// LOOP
//////////////////////////////////////////////////////////////////////

void loop(void)
{
//store the number of sensors to the variable numSensors
sensorsA.begin();
const int numSensorsA = sensorsA.getDeviceCount();
sensorsB.begin();
const int numSensorsB = sensorsB.getDeviceCount();
sensorsC.begin();
const int numSensorsC = sensorsC.getDeviceCount();
sensorsD.begin();
const int numSensorsD = sensorsD.getDeviceCount();
sensorsE.begin();
const int numSensorsE = sensorsE.getDeviceCount();
sensorsF.begin();
const int numSensorsF = sensorsF.getDeviceCount();
const int totalnumSensors = numSensorsA + numSensorsB + numSensorsC +
numSensorsD + numSensorsE + numSensorsF;

DateTime now;

  // delay for the amount of time we want between readings
  delay((LOG_INTERVAL -1) - (millis() % LOG_INTERVAL));

// fetch the time
  now = RTC.now();                        // Current time
    logfile.print(millis());                      // Write to file...
    logfile.print(",");
    logfile.print(now.month(), DEC);
    logfile.print(",");
    logfile.print(now.day(), DEC);
    logfile.print(",");
    logfile.print(now.year(), DEC);
    logfile.print(",");
    logfile.print(now.hour(), DEC);
    logfile.print(",");
    logfile.print(now.minute(), DEC);
    logfile.print(",");
    logfile.print(now.second(), DEC);
    logfile.print(",");
    logfile.flush();


 if (ECHO_TO_SERIAL) {
  Serial.print(now.year(), DEC);
```

```
  Serial.print("/");
  Serial.print(now.month(), DEC);
  Serial.print("/");
  Serial.print(now.day(), DEC);
  Serial.print(" ");
  Serial.print(now.hour(), DEC);
  Serial.print(":");
  Serial.print(now.minute(), DEC);
  Serial.print(":");
  Serial.print(now.second(), DEC);
  Serial.print('"');
 }

// Pump on/off Status
for (uint8_t i=0; i < 19; i++) {
  if (digitalRead(pump[i]) == HIGH){
  Serial.print(",");
  logfile.print(" On");
  logfile.print(",");
  logfile.flush();
  Serial.print(" On ");
  }
  if (digitalRead(pump[i]) == LOW){
  Serial.print(",");
  logfile.print(" Off");
  logfile.print(",");
  logfile.flush();
  Serial.print(" Off");
  }
}

// Temperature Status
 // call sensors.requestTemperatures() to issue a global temperature
  // request to all devices on the bus

  //Serial.print("Requesting temperatures...");
  sensorsA.requestTemperatures();
    for (uint8_t i = 0; i < numSensorsA; i++) {
    float tempA = sensorsA.getTempC(devicesA[i]);
//    delay(750);
    logfile.print(tempA);
    logfile.print(", ");
    Serial.print("Temp C: ");
    Serial.print(tempA);
//  // print the device information
//   printData(devicesC[i]);;
 }

  sensorsB.requestTemperatures();
  for (uint8_t i = 0; i < numSensorsB; i++) {
      float tempB = sensorsB.getTempC(devicesB[i]);
    logfile.print(tempB);
    logfile.print(", ");
    logfile.flush();
    Serial.print(", ");
    Serial.print("Temp C: ");
```

```
    Serial.print(", ");
    Serial.print(tempB);
//  // print the device information
//   printData(devicesC[i]);
 }

  sensorsC.requestTemperatures();
  for (uint8_t i = 0; i < numSensorsC; i++) {
    float tempC = sensorsC.getTempC(devicesC[i]);

    logfile.print(tempC);
    logfile.print(", ");
    logfile.flush();
    Serial.print(", ");
    Serial.print("Temp C: ");
    Serial.print(", ");
    Serial.print(tempC);
//  // print the device information
//   printData(devicesC[i]);
 }
  sensorsD.requestTemperatures();
  for (uint8_t i = 0; i < numSensorsD; i++) {
      float tempD = sensorsD.getTempC(devicesD[i]);
    logfile.print(tempD);
    logfile.print(", ");
    logfile.flush();
    Serial.print(", ");
    Serial.print("Temp C: ");
    Serial.print(", ");
    Serial.print(tempD);
//  // print the device information
//   printData(devicesC[i]);
 }
  sensorsE.requestTemperatures();
  for (uint8_t i = 0; i < numSensorsE; i++) {
     float tempE = sensorsE.getTempC(devicesE[i]);
    logfile.print(tempE);
    logfile.print(", ");
    logfile.flush();
    Serial.print(", ");
    Serial.print("Temp C: ");
    Serial.print(", ");
    Serial.print(tempE);
//  // print the device information
//   printData(devicesC[i]);
 }
  sensorsF.requestTemperatures();
  for (uint8_t i = 0; i < numSensorsF; i++) {
      float tempF = sensorsF.getTempC(devicesF[i]);
    logfile.print(tempF);
    logfile.print(", ");
    logfile.flush();
    Serial.print(", ");
    Serial.print("Temp C: ");
    Serial.print(", ");
    Serial.print(tempF);
```

```
//  // print the device information
//   printData(devicesC[i]);
 }
  //Serial.println("DONE");
  Serial.println();
  logfile.println();
  logfile.flush();
}

//FUNCTIONS
/////////////////////////////////////////////////////////////////////////
//// main function to print information about a device
//void printData(DeviceAddress deviceAddress)
//{
//  Serial.print("Device Address: ");
//  printAddress(deviceAddress);
//  Serial.print(" ");
//  printTemperature(deviceAddress);
//  Serial.println();
//}
//
// function to print a device address
void printAddress(DeviceAddress deviceAddress)
{
  for (uint8_t i = 0; i < 8; i++)
  {
    // zero pad the address if necessary
    if (deviceAddress[i] < 16) Serial.print("0");
    Serial.print(deviceAddress[i], HEX);
  }
}
//// function to print the temperature for a device
//void printTemperature(DeviceAddress deviceAddress)
//{
//  float tempC = sensors.getTempC(deviceAddress);
//  logfile.print(", ");
//  logfile.print(tempC);
//  Serial.print("Temp C: ");
//  Serial.print(tempC);
//  Serial.print(" Temp F: ");
//  Serial.print(DallasTemperature::toFahrenheit(tempC));
//}
//// function to print a device's resolution
//void printResolution(DeviceAddress deviceAddress)
//{
//  Serial.print("Resolution: ");
//  Serial.print(sensors.getResolution(deviceAddress));
//  Serial.println();
//}
void error(char *str)
{
Serial.print("error: ");
Serial.println(str);
//// red LED indicates error
//digitalWrite(redLEDpin, HIGH);
while(1);  //halts on error
```

APPENDIX D

TEMPERATURE LINE THERMISTOR ADDRESSES

Temperature sensor locations and serial numbers					
Loc.	**Ground line**		**TS1**		**TS4**
1	10DD472F00080051	D39	1010512F00080064	B1	101A462F00080086
2	10DE4D2F000800B5	D26	1006502F00080040	B5	1015442F00080021
3	106D512F000800B6	D38	105F542F000800E7	B10	10DB4B2F000800C2
4	101F432F000800BF	D62	10BB492F00080082	B8	10E1442F0008008F
5	10A1532F000800D7	D29	104F492F0008002C	B9	10C34E2F000800EA
6	10FB452F000800D6	D48	103F432F00080009	B11	10AE542F000800A2
7	109A522F0008000F	D17	10D6442F000800E7	B6	10CF4B2F00080045
8	105F452F00080056	D63	10F64E2F000800EC	B7	10A5402F00080039
9	10A0472F00080083	D1	10DA4A2F00080038	B4	10ED4A2F00080050
10	10C8422F00080033	D4	1024422F00080067	B2	10EF4E2F00080021
11	1090502F00080043	D2	10E4472F0008002A	B3	1006412F000800F1
12	1023472F00080030	D40			
13	1006442F00080023	D18			
14	1047412F000800B3	D49			
15	10B34D2F0008003C	D41			
16	1037472F000800B7	D56			
17	102B4A2F0008007D	D44			
18	10EF482F000800BD	D61			
19	10C2502F00080003	D14			
20	106C432F0008007E	D12			
21	1059432F00080078	D35			
22	1049502F00080011	D32			
23	10D1472F0008002C	D31			
24	10F9412F000800A7	D36			
25	10D8512F0008005A	D6			
26	10A7522F000800A8	D52			
27	1037502F0008009A	D53			
28	1011522F0008001D	D30			
29	10444B2F00080057	D8			

Loc.	**Ground line**				
30	10274E2F0008001F	D51			
31	108B4A2F00080021	D42			
32	107E432F0008004B	D27			
33	10C7432F000800DA	D50			
34	1058422F00080082	D5			
35	10664C2F000800DE	D20			
36	102E412F000800E6	D24			
37	10224D2F000800BA	D15			
38	1037442F000800F9	D54			
39	10FE522F00080010	D28			
40	102B442F000800DF	D43			
41	1042442F0008008A	D13			
42	103B422F00080018	D46			
43	10CF422F000800B6	D59			
44	10DE4E2F000800FB	D25			
45	1066502F00080083	D19			
46	10BB4C2F00080050	D47			
47	10D5492F00080052	D37			
48	10544B2F0008000C	D9			
49	10344A2F00080002	D10			
50	1069472F0008008A	D34			
51	1066462F00080063	D21			
52	102E442F00080034	D23			
53	108C502F00080065	D11			
54	1076472F000800F5	D22			
55	1049472F0008003C	D33			
56	1084452F0008006A	D7			
57	10B0452F0008005B	D3			
58	10CF542F00080056	D58			
59	1037522F00080019	D55			
60	100F512F0008001B	D57			
61	106F462F000800F5	D60			
62	100A4D2F000800AD	D16			
63	106B4E2F00080017	D45			
	Ambient				
1	10504D2F0008004C	E1			

REFERENCES

[1] Bassam Z. Shakhashiri, et al. "Climate change and our responsibilities as chemists," *Arabian Journal of Chemistry,* vol. 7, no. 1, pp. 5-9, 2014.

[2] Dr. Pieter Tans, et al. "Trends in Atmospheric Carbon Dioxide," Earth System Research Laboratory, 19 April 2015. [Online]. Available: http://www.esrl.noaa.gov/gmd/ccgg/trends/. [Accessed 25 April 2015].

[3] Maroto-Valer, Developments and Innovation in Carbon Dioxide (CO2) Capture and Storage Technology, Volume 2 - Carbon Dioxide (CO2) Storage and Utilisation, Woodhead Publishing, 2010.

[4] Chiara Marieni, et al. "Geological storage of CO2 within the oceanic crust by gravitational trapping," *An Agu Journal, Geophysical Research Letters,* vol. 40, no. 23, p. 6219–6224, 2013.

[5] Yeo II Yoon, et al. "Characteristics of CO2 capture system using KIERSOL in the LNG flue gas," *Energy Porcidia ,* vol. 63, pp. 1745-1750, 2014.

[6] U.S. Energy Information Administration, "Today in Energy," EIA, 7 March 2013. [Online]. Available: http://www.eia.gov/todayinenergy/detail.cfm?id=10271&src. [Accessed 1 April 2015].

[7] Department of Energy , "Buildings Overview," DOE, 2008. [Online]. Available: http://www.c2es.org/technology/overview/buildings. [Accessed 1 April 2015].

[8] U.S. Energy Information Administration, "Total Energy," EIA, 27 September 2012. [Online]. Available: http://www.eia.gov/totalenergy/data/annual/showtext.cfm?t=ptb0211. [Accessed 1 April 2015].

[9] D&R International, 2008 Buildings Energy Data Book, Department of Energy , 2008.

[10] A. Reay, et al. Heat Pipes, Theory, Design and Applications, Butterworth_Heinemann, 2006.

[11] Kent S. Udell, et al. "Net Zero Energy Air Conditioning Using," *ASHRAE Transactions,* vol. 11, no. 22, pp. 892-898, 2011.

[12] P. M. Jankovich, Seasonal underground thermal energy storage using smart thermosiphon arrays, Salt Lake City: University of Utah, 2012.

[13] D. Hillel, Fundamentals of Soil Physics, New York: Academic Press, 1980.

[14] J. J. H.s. Carslaw, Conduciton of Heat in Solids, Clarendon Press, 1959.

[15] "Historical Weather," Weather Spark, 28 4 2015. [Online]. Available: https://weatherspark.com/history/31629/2015/Salt-Lake-City-Utah-United-States. [Accessed 5 May 2015].

[16] F. P. Incropera, Fundamentals of Heat and Mass Transfer, John Wiley & Sons, 2007.