6-2014

Controllable Home Coffee Roaster from a Modified Popcorn Popper

Nicholas Brenn
Union College - Schenectady, NY

Controllable Home Coffee Roaster from a Modified Popcorn Popper
ECE-499 Capstone Design
March 30, 2014

Submitted to:

Union College, Department of Electrical and Computer Engineering
Professor Traver

Prepared by:

Nicholas A. Brenn

Executive Summary

Foreward

This Senior Capstone Project is the culmination of over three years of undergraduate electrical engineering instruction along with supporting courses in mathematics and physics. The diligent study of these concepts has allowed for the application into a project that combines a passion for both electrical engineering and coffee.

The extraction of the complex aromatic and flavor characteristics of a coffee bean is dictated by a roaster's ability to control a variety of variables that act on a green coffee bean while it is roasting. Not only must one understand certain audible and visual cues from the beans during the course of the roast, but the roasting apparatus must also react to adjustments made by the roaster user to the fan speed and heater power in order to adjust the temperature inside of the roasting chamber. At the same time, the roast logging software must be displaying a graphical summary of the temperature in the chamber versus time, also known as the roasting profile.

This feat often necessitates the purchase of an expensive programmable roaster; but, a modified popcorn popper can rival the functionality of a high-end programmable roaster if adapted with internal temperature sensors, a reactive control system, and the ability for the software to save roasting profiles for later use. This modified popcorn popper will automatically control the temperature and total roast time based on industry roasting profiles; component costs will remain under $360; and a taste comparison between beans roasted in a modified and unmodified roaster will be completed by an experienced coffee taster.

Summary

Upon submission of this technical report, the roaster has the capabilities of controlling both the power to the heater coil and the speed of the universal AC/DC motor that is used to blow air over the heating coil. This air is blown into the roasting chamber where thermocouples are sensing and sending temperature information to a microcontroller which is interfaced with a roast logging software called Artisan. This roast logging software displays the temperature versus time inside of the roasting chamber while also providing functionality to the roaster user to adjust the percentage of power going to the heater coil and the percentage of speed that the fan is spinning.

Table of Contents

List of Tables and Figures

Introduction

The goal of this project is to modify a home popcorn popper with computer aided logging of the roasting profile, supported by solid state switching of the heating coil and fan to control the temperature inside of the roasting chamber, and finally, automatic control of the roasting chamber temperature through a PID algorithm. With this control over the roast of the coffee bean, an experienced roaster user is able to identify the stages of the roast where control over the temperature inside the chamber is most critical. As compared to roasting green beans in an unmodified home popcorn popper, this project gives a coffee drinking connoisseur a hands-on appreciation for the development of the flavor profile that is unique to each brewed cup.

100 million Americans drink a cup of coffee each day[13]. Many of these coffee drinkers are unaware of the path that a coffee bean takes before it ends up in the mug. Of these steps, the roasting process may have the greatest impact on how the cup of coffee ultimately tastes. The quality of the coffee bean and the origin of growth also have significant impacts on the roasted bean’s flavor, but the roasting process transforms the undrinkable and rock-hard green coffee bean into one of the many types of roasts which will be described in the background.

This paper will begin with an in-depth background on the coffee bean and the roasting process. Following the coffee background, the design requirements of the roaster will be outlined in detail. This paper will conclude with a detailed description of design implementation and an evaluation of how the popper’s roasted beans compare to professionally roasted beans of the same roast and origin.

Background

Professionally certified coffee tasters have trained their palates to identify a variety of flavors and aromas in a cup of coffee that has been brewed with freshly roasted beans. The unique flavor profile of a coffee bean is influenced by the regional origin of the coffee bean, the method of bean removal from the coffee cherry, the roasting process, and the brewing of the coffee bean into the liquid gold plays such an important role in the morning routines of people around the world.

Coffee Origins

In addition to price and freshness being perks of roasting at home, many coffee connoisseurs also enjoy exploring the intricacies of beans grown in different parts of the world. The majority of the world's coffee is grown in warm and moist climates at elevations between from 1000ft to 6000ft. The primary growing regions are in Africa (Ethiopia and Kenya), Southeast Asia (Indonesia), and in many countries in Latin America. The *arabica* coffee species is often grown at higher elevations while the less appreciated and cheaper *robusta* species is grown at a lower elevation. (Davids 72) Arabica beans are typically of higher quality and they are often labeled as "specialty" coffee beans or as "single-origin". Single origin beans are beans distinguished by their specific region or by the coffee cooperative that they are grown in. When arabica beans are not sold as *single-origin*, they are distinguished by the generalized country of origin or as a blend of beans from various regions. Robusta beans, however, are often much cheaper and are the trademark beans for brands like Maxwell House® and Folgers. Robusta beans lack complex flavors, but the heavy mouth feel and bold taste appeals to many everyday coffee drinkers.

Due to the breadth of different flavors and aromas, this paper will not cover the specific complexities of coffees grown around the world. But to name a few of the commonly compared features, there are differences in acidity, mouth-feel, fruit tastes, chocolate undertones, aftertastes and bitterness.

Coffee Cherry Processing

The coffee bean is the seed of the fruit known as a coffee cherry. In order to retrieve the green coffee bean, it must first be separated from the coffee fruit. The separation of the fruit from the bean is known as *processing*. (Davids 76) Davids explains, “Processing is one of the most important influences on coffee quality and taste.” (76) The two most widely used methods to separate the bean from the fruit are by using the *wet* and *dry* processes.

In the wet process, “the various layers of skin and fruit around the bean are stripped off gently and gradually, layer by layer, before the bean is dried.” (Davids 76) Wet processed coffees typically taste cleaner with a bit more acidity than dry-processed beans. Dry processed beans “are dried with the coffee fruit still adhered to the bean.” (76) These coffees tend to taste fruitier and have a thicker mouth feel than wet-processed beans. Davids explains that the drying process also affects the quality and flavor of the bean. “As a rule, sundried coffees are considered preferable to machine-dried coffees.” (77)

Coffee Bean Roasting

Kenneth Davids, author of Home Coffee Roasting, states, “Nothing influences the taste of coffee more than roasting.” (Davids 51) Before the 1900’s, roasting coffee at home was common for working class families. Europeans would roast a week’s batch of beans on their balconies using an *abbrustulaturo*, which is the Neopolitan word for a coffee roaster. The smell of roasted coffee would fill the streets of Naples as families roasted their beans in a rotatable cylinder which sat atop a charcoal grill[15]. In the 1800's, pre-roasted coffee became more commercially available in the United States and home roasting slowly became the practice of hobbyists and connoisseurs (5).

Why is it attractive for hobbyists to roast small batches of coffee at home? As a coffee drinker looks to get more out of their cup of coffee, they venture towards higher quality coffee beans. Since freshly roasted coffee beans are more expensive than the average pound of beans from the supermarket, home roasters save between 25 and 50 percent of the cost per pound of green coffee beans (Davids 7).

Early home roasters were simply modified pans which were placed over a fireplace. The beans would be placed in the pan and they would roast by conduction. Stove roaster development began in mid to late 19th century, where a cylindrical roasting chamber had a handle to twirl over the heat source to evenly distribute the heat to the beans. Commercial coffee roasters have primarily retained the same concept of applying heat to a rotating drum, but of course, temperature controls and byproduct management has kept these roasters technologically modern. Back home however, a convection method of coffee roasting emerged in the form of electric fluid-bed roasters. Davids explains, "A column of hot air rises from the base of the device, roasting and seething the beans inside the narrow cylinder that protrudes necklike from the base" (Davids 42). These roasters contain just a heating element and a fan that pushes the hot air into the roasting chamber. A coffee hobbyist can own the most expensive roaster on the market, but they must understand the science behind the roasting process in order to get the most out of their investment.

A scientific understanding of the roasting process is necessary. Davids rhetorically asked what happens to coffee when it is roasted. His response was that, "In fact, no one knows – exactly" (Davids 9). Visually, we see that the coffee bean changes from green to varying shades of brown, it increases in size, and smoke begins to increase as the roast progresses. A chemical process known as *pyrolysis* "carmelizes a portion of the bean's sugars and transforms some into what are popularly called the coffee's flavor oils" (Davids 10). A green coffee bean also loses water weight and expands in size as the roast progresses. Aside from the known chemical reactions taking place when heat is applied to an organic substance, "700 to 850 substances have been identified as possible contributors to the flavor of roasted coffee." (Davids 9)

Over the course of the roast, the bean turns from green to varying degrees of brown. The first noticeable change is a smell that roasters relate to a loaf of fresh baked bread. At this stage, the beans are slowly changing to a yellowish color. The color change during the roast is known as *Maillard Reaction*. This is the same reaction that explains why our food turns brown when we apply heat to it. Within at least two to three minutes, "the first popping or crackling sound[1]" can be heard from the roasting chamber. In the roasting industry, this is known as the *first crack*. Davids describes the first crack at the "definite start of the roast transformation or pyrolysis." Davids 11) At this stage, the sugars begin to carmelize and more roasting smoke starts to mildly bellow from the roasting chamber.

The bean temperature, classified by home roasters by the acronym, *BT*, can range from "350°F/175°C at the beginning of pyrolysis to about 435°F/225°C for a medium roast, to as high as 475°F/245°C in a very dark roast. Any time after the first crack is an acceptable stage to cut the heat to the roasting chamber. A second crack can be heard if the roaster keeps the heat on a few minutes past the first crack. Roasts that are halted after the second crack are very dark, producing a very bold tasting cup. Giving the roaster the ability to react to the audible and visual variables of the roast with control of the temperature inside of the roasting chamber is the motivation of this design project.

Roasted Coffee Classification

The taste and aroma are critical classification qualities of a roasted coffee bean. An extensive list of terms can be used to describe a cup of coffee, including descriptors for acidity, body, aroma and finish. Professional coffee tasters use the flavor wheel in Figure 1[14] to describe the flavors and aromas of the coffees they taste. Davids describes, "slower-roasted coffee tends to be fuller in body and more complex in taste." Roasted coffee is also classified by the color of the bean, ranging from light brown to very dark brown, which can also be precisely classified by a near-infrared spectrophotometer[1].

Figure 1: Coffee Wheel for Tastes and Aromas

Coffee bean classification depends not only on the bean origin, but also on the manner in which it is roasted. The manner in which a bean is roasted is quantified by a roasting profile. Renowned green coffee bean distributor, Sweet Marias, describes the roasting profile as a graphical summary of the temperature adjustments made during the entire roasting process[2]. Roast profiles are commonly used as a tool for consistency; therefore it will be important to be able to save different roasting profiles for future implementation with a specific bean.

Brewing and Extraction

Brewing a cup of coffee is the culmination of the long journey of the coffee bean. From the tree to your supermarket, the coffee bean has been exposed to many different environments that affect the quality of the extracted cup of coffee. Since the coffee has traveled such a distance, coffee connoisseurs see the brewing process as the preservation of the experiences of the coffee bean. In other words, it would be a shame to take a high quality batch of coffee and use a brewing method that doesn't maximize the potential of the coffee bean.

For lower quality coffee, like the robusta beans found in Folger's or Maxwell House tins, is most often brewed in the typical home coffee maker like the one shown below in Figure 2. For freshly roasted and higher quality coffee, coffee enthusiasts prefer brewing coffee using the pour-over method. The pour-over method uses a filter and a filter cone that allows for the barista to precisely control how the water is poured over the beans in the filter. An example of this method is shown below in Figure 3.

Figure 2: A typical home coffee maker[16]

Figure 3: A common pour-over brewing setup[27]

Design Specifications

Project Objectives

Manual Control

The main design goal was to implement manual control of the temperature inside of the roasting chamber. By manipulating the speed of the fan and the power going to the heater, the temperature can be adjusted during the course of the roast. The adjustment of these two variables and the visualization of the temperature versus time curve (the roasting profile) should be accomplished using one of the freely downloadable roast logging applications. The roaster should also be able to manually control the temperature using two potentiometers, with one controlling the fan speed and one controlling the heater power.

PID Control

Another design goal was to to implement a PID algorithm to automatically control the temperature inside of the roasting chamber. This PID algorithm would adjust the power to the heater based on temperature set-points which vary with time in an uploaded roasting profile. The user of the roaster should be able to upload a roasting profile that contains the temperature set-points at different times during the roast. The user of the roaster should also be able to save roasting profiles…

Byproduct Management

During the roasting process, the innermost skin of the coffee fruit falls off of the coffee bean. The little brown flakes called, *chaff*, will naturally fly out of the roasting chamber inside of the hot-air roaster[1]. Chaff collection is an important component of any high-end roaster, because it will float around the room that the beans are being roasted in. Another by-product of the roasting process is roasting smoke. Davids explains, "Improvising home roasters who use gas ovens and built-in convection ovens are in luck." Since the popcorn poppers do not roast large quantities, smoke will not be as much of an issue as a large batch roaster[1].

As the coffee bean dries inside of the roasting chamber, it begins to lose mass and the beans begin to circulate more freely inside of the roasting chamber. As this happens, the beans are at risk of flying out of the chamber. In order to prevent losing beans during the roast, a chimney must be fitted into the popper in order to allow the beans to circulate freely without risk of flying out of the chamber. A chimney can be fitted out of an aluminum soup can or also out of a glass chimney from an antique camping lamp.

Cost Objectives

A student research grant supplied $360 to fund the components for the development of this project. The initial design goal was to remain under $500 in total component cost. By remaining under $500, the coffee roaster will be more expensive than uncontrollable home roasters on the market but also less expensive than home roasters with temperature control and roast profiling abilities.

Technology Scope

Roaster Selection

There are many options for roasting coffee at home, but these devices differ greatly in usability, controllability, and price. In order to remain within my original budget of $500, I chose to modify a hot-air roaster. Hot-air roasters are classified as fluid-bed roasters because the stream of hot air not only acts as a heat source for the beans, but it also acts as a bean agitator to keep the beans in constant motion in the roasting chamber[1]. A diagram of fluid-bed roasting is shown below in Figure 4.

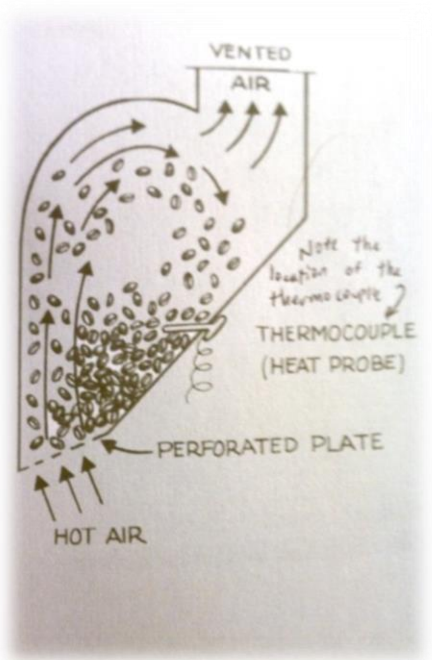

Figure 4: Fluid-bed roasting diagram[1]

As shown in Table 1, there are a variety of fluid-bed roasters on the market. Determining the best popper to modify was based on variables such as the type of time and heat control as well as the price of the device itself.

Model	Time Control	Heat / Fan Control	Price
Fresh Roast SR300	Yes	None	$150.00
Fresh Roast SR500	Yes	Low, Medium, High Settings	$169.00
Nesco Home Roaster	Yes	None	$150.00
HotTop KN-8828P-2K	Yes	Roasting Temperature, Fan Speed, Heater Power	$1,000.00
West Bend Poppery I	ON/OFF Switch	None	$32.99 (or less)

Table 1: Decision Matrix for Popcorn Popper Choice

The Fresh Roast roasters and the Nesco Home roasters have a basic level of time control. This time control is digital, opposed to the manual time control of the West Bend Poppery I through the use of the ON/OFF Switch. The Fresh Roast SR500 model has low, medium, and high settings that control the amount of heat applied to the bean. There is no way to set a specific temperature, which severely inhibits one's control over the roast. Before modification, the West Bend Poppery I would have no control over the temperature inside of the chamber. The third column outlines the average price for these roasters. For under $50, the West Bend model frees up funding opportunities to add control capabilities that surpass the more expensive models, while remaining less expensive than the HotTop roaster which has many desirable controls.

Poppery II vs. Poppery I

The West Bend line of air popcorn poppers were introduced to the home appliance market in the late 1970's[11]. The Poppery I is their 1500 Watt model and the Poppery II is their 1200 Watt model. The Poppery I uses a single-phase AC induction motor. The Poppery II uses a DC motor with a bridge rectifier connected to the motor inputs, which converts the AC into DC for the motor. In the home coffee roasting community, the Poppery I is one of the most highly regarded poppers because it provides 300 more watts than the Poppery II. It is also more durable than the and the roasting chamber can reach higher temperatures than that of the Poppery II. The two models are shown below in Figure 5.

Figure 5: West Bend® Poppery Models (I left, II right)[17]

Component Selection

Solid State Relays

In order to control the temperature inside of the roasting chamber, the speed of the universal AC/DC motor and the power going to the heating coil must be controlled. Using pulsed width modulation (PWM), the computer triggers the solid state relays to control the fan speed and heater power. Solid state relays (SSR) allow the designer to control high-voltage AC loads from low voltage DC signals. SSRs also allow for fast switching, since there are no mechanical contacts inside of the component. Instead of a moving contact, SSRs are switched by photo-coupling using an infrared LED and a phototriac which are located in the SSR housing.[18] For my project, I will be using a Zero Voltage SSR to control the heater and an instantaneous turn-on SSR to control the speed of the fan.

Zero Voltage SSR

The power output to the nichrome heating coil in the popcorn popper will be controlled using a zero voltage solid state relay. A zero voltage SSR only switches the power to the load when the AC mains voltage is at the zero crossing of the sinusoidal waveform. Since the common outlet in the United States outputs 60Hz/120VAC, the waveform cycles between -120V and 120V, 60 times per second. This project utilized a 25A relay with an operating voltage between 3 and 32 VDC. This allows the relay to be driven from the output of the Arduino microcontroller.

Integral cycle control fires the solid state relay for X number of half cycles and then the relay remains off for a remaining number of integer half cycles. For an electric heating coil, integral cycle control is electrically quieter and also extends the heater's life.

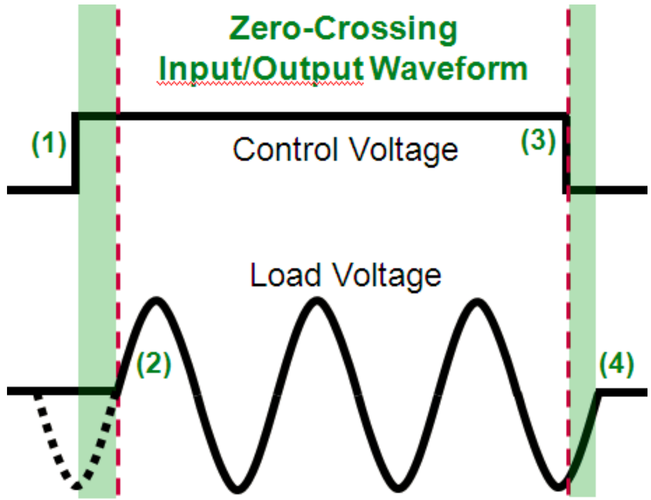

Figure 6: Zero-Cross Relay Conduction[19]

Instantaneous Turn-On SSR

Instantaneous turn-on relays, also known as random turn-on relays, are commonly used for phase angle control of an inductive load, like a universal AC/DC motor. An instantaneous turn-on relay turns on immediately upon receiving the control signal from the Arduino microprocessor. Crydom explains that this type of relay is commonly used in phase angle control applications where precise control of the power to the load is important. Figure 5 shows the control voltage signal and the point on the AC waveform where the relay conducts.

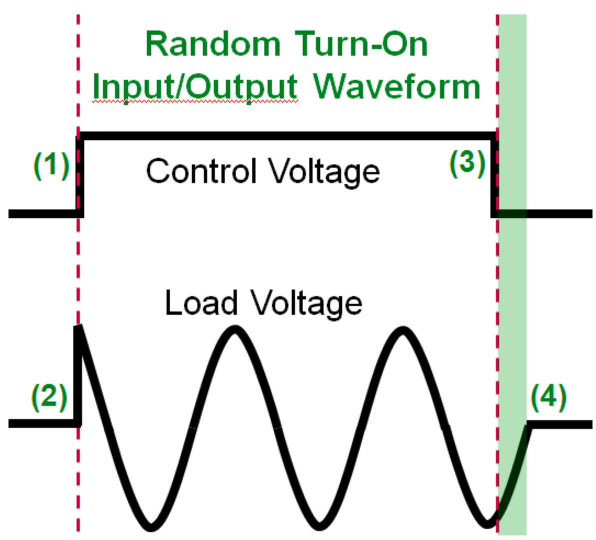

Figure 7: Instantaneous Turn-On Relay Conduction[19]

Heat Dissipation

Solid state relays develop heat during operation. Mounting a solid state relay onto a heat sink will allow the heat created by the switching SSR to dissipate into the ambient environment. In the case of this roaster project, the project enclosure was made of wood. It would not have been safe to mount the SSRs directly to the wooden project enclosure because of the small risk of burning, so the heat sinks were required. Heat sinks also ensure long term reliability of the SSRs. The proper heat sink model was determined using Crydom's thermal model for calculating a heat sink.[20]

Zero Cross Detector

The zero cross detector (ZCD) is a circuit that provides phase information of the high voltage AC waveform to the low voltage Arduino microcontroller. The ZCD produces a symmetric square wave pulse at the zero crossing of the AC waveform. This pulse is outputted through the isolated side of the 4N35 optocoupler. The phase angle control algorithm requires this signal from the ZCD in order to precisely scale the power to the universal AC/DC motor inside of the Poppery. The ZCD was connected in line with the mains voltage, while the output of the ZCD was connected to the Arduino microcontroller. A schematic of the ZCD is shown below in Figure 8.

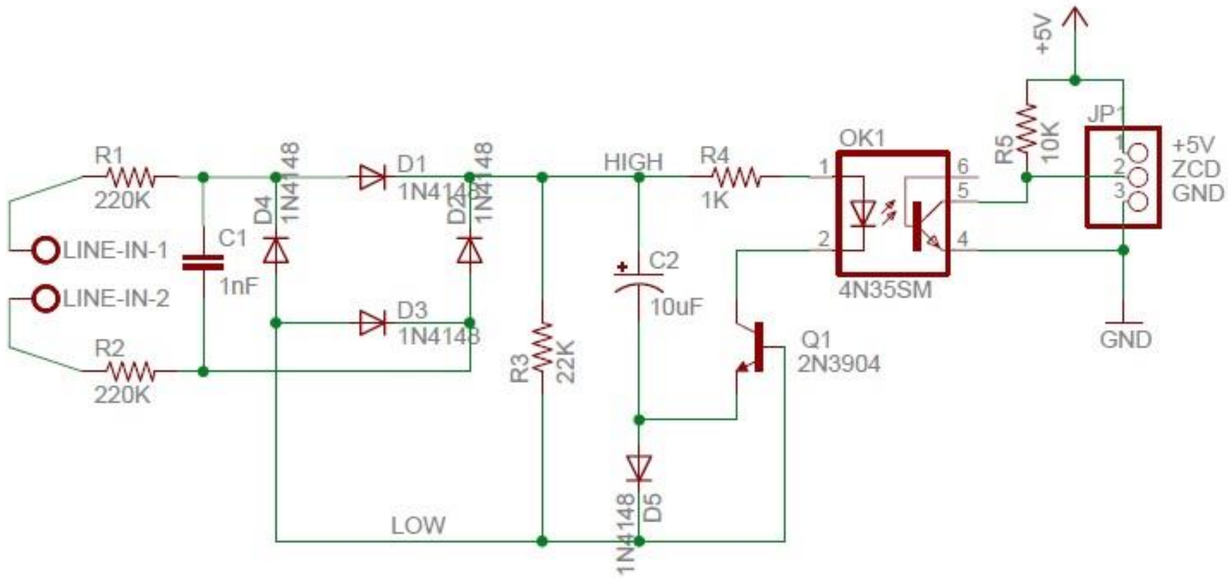

Figure 8: Schematic of the Zero Cross Detector[21]

Thermocouple Selection

The selection of the thermocouple type was not difficult because of the vastly different temperature ranges of each of the types of thermocouples. Green bean supplier, Sweet Marias, notes that the roast begins around 75 degrees Fahrenheit and ends around 450 degrees Fahrenheit. With these minimum and maximum temperature parameters, I chose the K-type thermocouple which has a temperature range between -454 degrees Fahrenheit and 2501 degrees Fahrenheit. The temperature range for coffee roasting is safely covered with the use of a K-type thermocouple. K-Type thermocouples are typically accurate to around 3.6 degrees Fahrenheit.

The placement of the thermocouple inside of the roasting chamber is important to the development of the roasting profile. In the home coffee roasting industry, thermocouples are distinguished as either “BT” or “ET” thermocouples. “BT” thermocouples represent the bean temperature, while the “ET” thermocouples represent the temperature coming directly from the heater.

Microcontroller and Software Selection

The Arduino microcontroller was chosen as the controller for the roasting program because it is designed to interface with the TC4 roasting shield and the leading roast logging application, called Artisan.

The TC4 is a shield for the Arduino microcontroller which is designed to process the data from the thermocouple. A kit to build the TC4 shield was purchased and through-hole and surface mount components were soldered. An image of the TC4 shield is shown below in Figure 9.

Artisan is a freely downloadable roast logging application that displays the temperature of the two thermocouples over time. Artisan also provides the manual control functions called "event buttons" and "sliders." The event buttons allow the roaster user to create a clickable button for specific power or speed percentages. The sliders allow the roaster user to control the power or speed percentages by clicking and moving the slider dial to a specific value.

The same developers of the Artisan program also developed the program that is loaded onto the Arduino microcontroller. This program consists of the algorithm for integral cycle control and the algorithm for the phase angle control. Since the Arduino program is designed to work for any type of coffee roaster, the program had to be customized to fit the specifications of the West Bend Poppery I. Variables such as the serial baud rate, mains voltage frequency, thermocouple type and pulse width of the solid state relays. The Arduino algorithm has been included in the Appendix to this report.

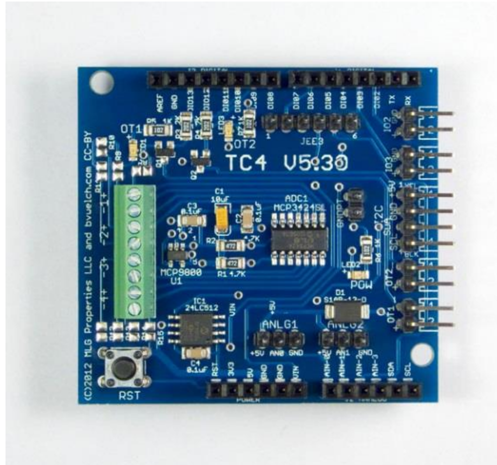

Figure 9: TC4 Arduino Shield[8]

System Diagram

A high level system schematic is shown below in Figure 10. The voltage from a standard wall outlet arrives at 120VAC at 60Hz. One lead of the heater coil is connected to the hot voltage line. The other lead of the heater coil is connected to Pin 1 of the zero-voltage turn on solid state relay (Crydom D2425). Pin 2 of this relay is connected to the neutral line. One lead to the universal AC/DC motor is connected to the hot voltage line. The other lead to the universal AC/DC motor is connected to Pin 1 of the instantaneous turn on solid state relay (Crydom D2425-10). Pins 3 and 4 of each solid state relay are connected to the outputs of the Arduino's TC4 shield, OT1 and OT2. The yellow and red leads of the thermocouples were connected to the positive and negative thermocouple inputs on the TC4 shield, respectively. The hot and neutral lines were connected to the input of the zero cross detector and the output of the zero cross detector was connected to an input/output pin of the TC4 shield. A serial cable connected the Arduino to the computer and a 9V power supply is also connected to the Arduino.

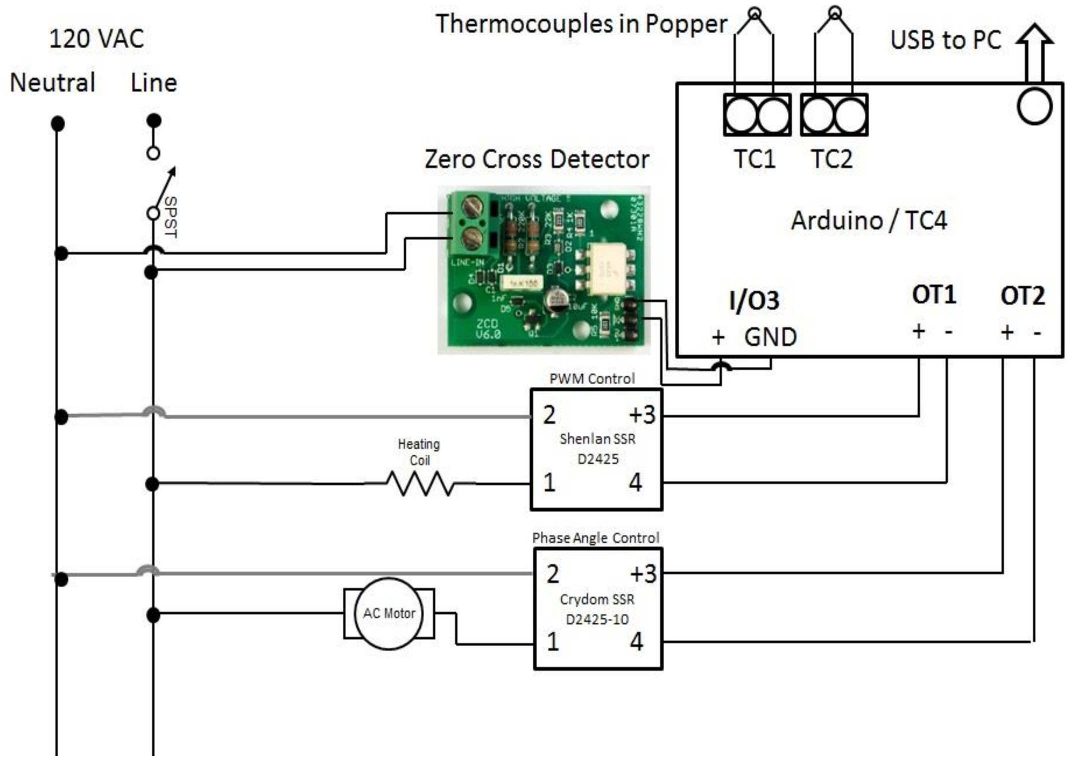

Figure 10: High Level System Diagram

Design Alternatives

Fuji PID vs. Arduino TC4

Control of the popper can also be accomplished using a standalone PID controller. The PID controller accepts a thermocouple input with single or dual control outputs. The Fuji Electric PXR series controller, for example, can hold up to 16 ramping and soaking segments. These ramping and soaking segments represent the amount of time that the temperature is held at a certain temperature. The controller also has programmable alarms with a burnout alarm in case there is not enough air coming from the fan[23].

The developers of the aArtisan Arduino code have implemented a PID algorithm. With this PID algorithm for the existing aArtisan code, the Fuji PID controller was discounted as a possible design alternative.

Light Dimmer Control

Home coffee roasters on a tighter budget may not have the funds to purchase a PID Controller or an Arduino/TC4 configuration. Another alternative to these two options is to implement strict manual control using a light dimmer. Since an unmodified popper roasts coffee too quickly, using a light dimmer to control the heater power or fan speed can drastically improve the roast. A typical light dimmer controls the output power by controlling the firing angle of the mains voltage.

One disadvantage of light dimmer control is that the control is not performed using a computer program that is also logging the temperature. Additional temperature logging circuitry would have to be implemented in order to achieve the all-in one control of the Arduino/TC4 package. Another disadvantage of the light dimmer option is that many of the cheap dimmers are not suited for high power loads, like the popcorn popper[26]. A more expensive light dimmer would have to be purchased in order to suit the power requirement of the popper. An alternative to the light dimmer is to use a variac in line with the heater. Similar to the light dimmer, a variac that is rated for currents around 12A are often more expensive.

Final Design and Implementation

Initial Popper Modifications

Upon the arrival of the West Bend Poppery I, the hot air popcorn popper was turned over and four of the bottom housing’s screws were removed. The first stages of the disassembly are shown below in Figure 11. The removal of the bottom plastic housing revealed a similarly shaped piece of metal, which then revealed the internal circuitry of the popper. The circuitry is shown below in Figure 12.

Figure 11: First Step of the Popper Disassembly

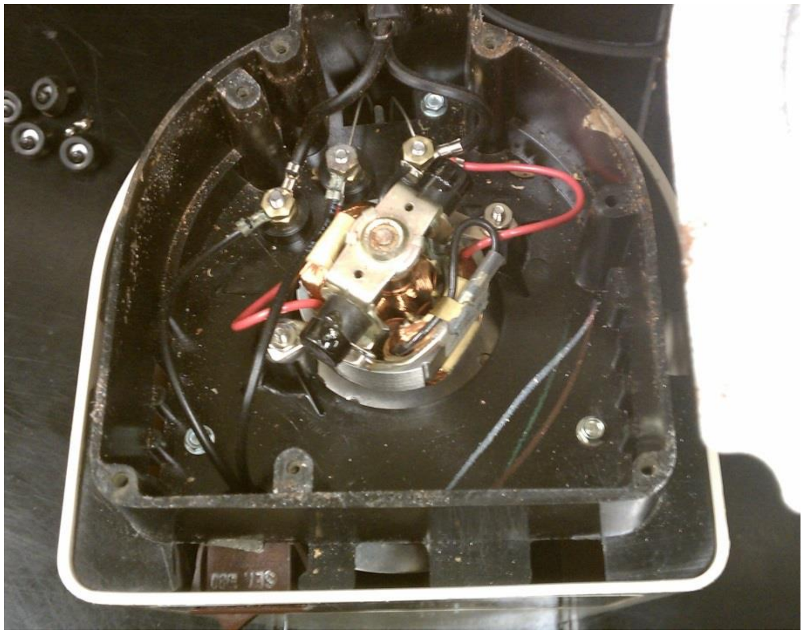

Figure 12: Unmodified Circuitry of the Popcorn Popper

As shown in Figure 12, there are two wires coming from the power cord. One wire is the line wire and the other is the neutral. The line wire is then connected to toggle switch on the front of the popcorn popper. From this toggle switch, a wire is connecting to one lead of the heating coil and one lead on the universal AC/DC motor which acts as the fan. When the switch is turned on, the heater is receiving 100% power and the fan is receiving 100% power. With this circuit configuration, the fan and the heater will be on at all times. One of the first modifications that home roasters make to the popcorn popper, is the separation of the heater and the fan circuit. By separating the heater and the fan circuit, the fan can be turned on without power going to the heater. Once these two components are isolated from one another, the control components can then be connected to each one individually.

Project Enclosure Creation

The creation of the enclosure for all of the electrical components was a critical step in the implementation process. The project enclosure was constructed during the 6-week winter break between the fall and winter terms. The construction took place in between making the initial modifications and the steps that included the final component wiring. Without a proper project enclosure, the placement of the solid state relays, Arduino, Zero Cross Detector, and terminal block would have been difficult. It would have been difficult to gauge the length of hookup wire needed between the popper that is mounted to the enclosure and the components that were placed inside of the enclosure.

A wooden box was purchased from an arts and crafts supply store. The outline of the bottom of the popcorn popper was traced onto the top of the wooden box. This was then cut out using a handheld saw and the popper was placed into the cut out portion. Figure 13, below shows the steps taken to construct the enclosure.

Figure 13: Project Enclosure Construction Steps

The enclosure was then sanded and stained brown. Wood shavings from the saw were removed and wood glue was used to fill in any chips that came off of the wooden enclosure when the popper outline was being sawed.

Thermocouple Placement

The “BT” thermocouple is an insulated glass braided thermocouple which is unshielded. The “ET” thermocouple has a flexible probe that was fixtured into one of the openings from the fan in the chamber. Both of these thermocouples were inserted into the chamber through the same hole, with the flexible thermocouple acting as a support for the flimsier unshielded thermocouple. An image of the placement of the thermocouple is shown below in Figure 14.

Figure 14: Thermocouple Placement into the Roasting Chamber

The leads for the unshielded thermocouple were secured to the “BT” thermocouple input on the Arduino/TC4 shield. Since the shielded, “ET” thermocouple had two prongs at the output, a piece of hookup wire was soldered to each prong in order for the other end to connect into the “ET” thermocouple input on the Arduino/TC4 shield.

Component Wiring

Once all of the components were purchased, the final component wiring could be completed. Black and red pieces of 12-gauge hookup wire were used to complete the connections shown in Figure 10. The hookup wire was attached to ring terminals in order for them to connect to the screw terminals on the solid state relay inputs and the terminal block. These were crimped for a secure connection. The Serial USB cable was connected to the USB port on the Arduino. A 9V DC Power Supply was also connected to the power supply port on the Arduino. Circuit board standoffs were used to secure the Arduino, terminal block, and zero cross detector to the wooden enclosure. The wires were tied together with electrical tape in order for the enclosure to be closed easier. In order to protect against electrical shock, the exposed terminals were covered in electrical tape.

Figure 15 shows the final electrical connections inside of the project enclosure. The enclosure is closed when the roast is in progress, but it remained open for initial testing purposes. A variable transformer was connected in line with the roaster power in order to test the electrical connections without supplying 100% power to the circuit. The popper was connected with 50% power and a multimeter was used to ensure that all of the connections were secure without any short circuits.

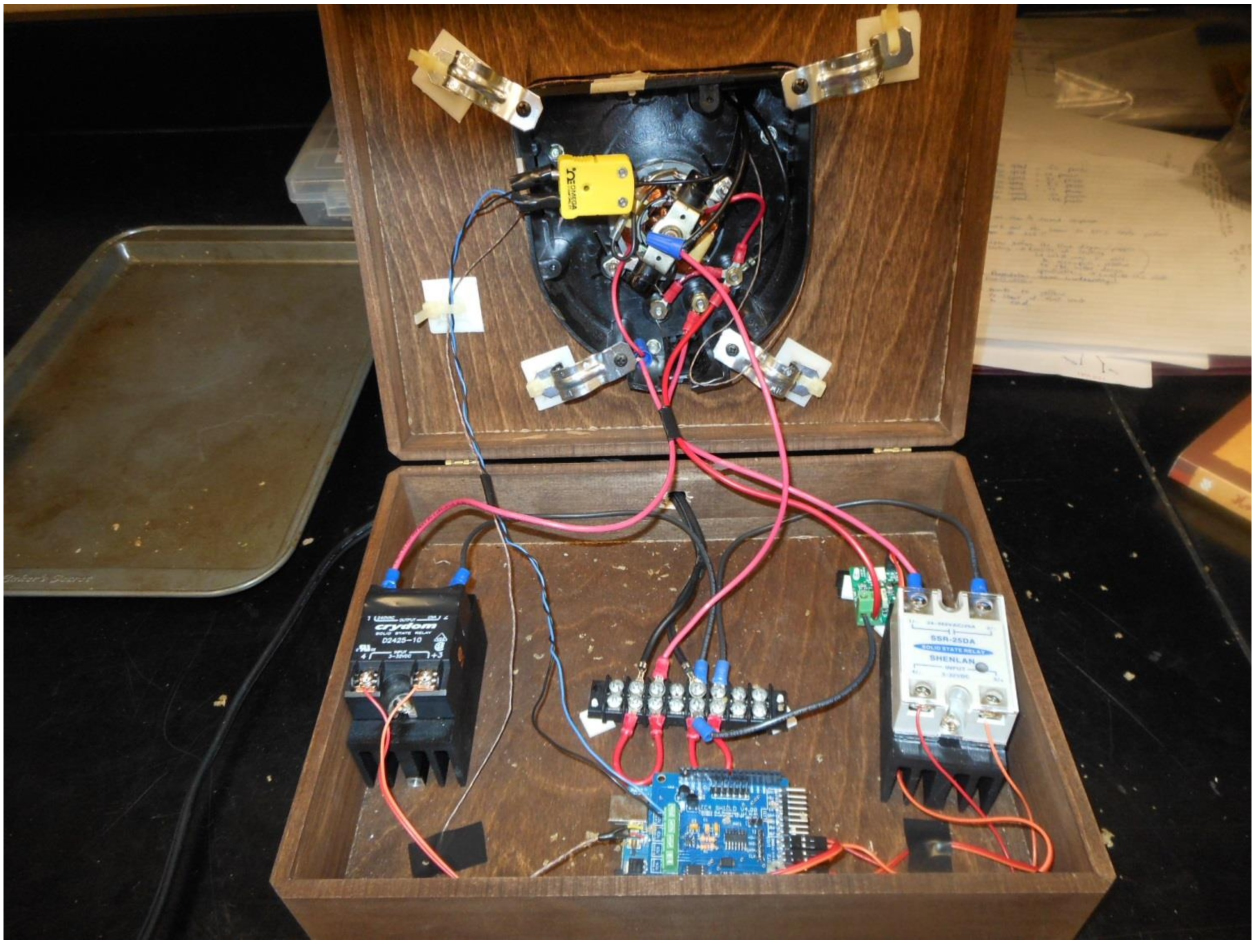

Figure 15: Final Project Connections

Performance Results

Unmodified Popcorn Popper Roast

On October 13, 2013, I performed a coffee roast in the unmodified popcorn popper. I used approximately 4 ounces of green coffee beans from the Huehuetenango region of Guatemala. The coffee was roasted to a *medium* classification in 7:00 minutes. The beans placed in a metal container with the lid slightly exposed to the air to allow for the carbon dioxide from the roasted beans to escape. They were allowed to sit for approximately 12 hours before they were tasted for the first time. The beans were brewed using a Hario V60 Pour Over cone.

Jeffrey Wettstein, a student with professional coffee tasting experience at Green Mountain Coffee Roasters, tasted the roasted beans and recorded his opinion. His notes are as follows: "smooth, not too bold, tea-like finish, bright, complete, juicy and salivating". One of the notable variables of this roast is the total roast time. Since the unmodified popper provides 100% power and 100% fan speed for the entire roast, the green beans will roast at a quicker pace as opposed to a roast that has controlled power to the heater and to the fan.

Modified Popcorn Popper Roast

On March 2, 2014, the same green beans from Guatemala and the same amount of beans were roasted in the completed coffee roaster. Although the green beans sat for under 5 months in the same plastic bag, there is no noticeable difference in the quality of the green beans within the first year[24]. The coffee was also roasted to a *medium* classification in 15:04 minutes. The beans were placed in the same metal container and were allowed to rest for 12 hours before another tasting by Jeffrey Wettstein. His notes were as follows: "Juicy first sip, a pear with sour plum-skin taste, tea-like finish, very light and clean cup of coffee." Although the art of picking out flavors in coffee is arguably a subjective practice, Jeff was able to pick out flavors in the coffee that the green bean supplier, Sweet Marias, describes as expected flavors in the roasted bean. Sweet Marias describes the Guatemala as follows, "The dry fragrance of El Turbante has vanilla and toasted sugar, along with a very sweet smell of raw honey. This is a nice, sweet coffee with a range of acidity from tartaric (green grape) to malic (red pear). Fruits like raisin and plum are subtle and pleasing. The finish is sweet and with a nice bittering allusion to citrus rind.[25]"

Another critical outcome of this controlled roast is the total roast time. The roaster was able to more than double the total time to complete the roast. As stated in the previous paragraph, coffee tasting is a subjective art and there is nobody that can definitively say that a 15 minute roast yields a better cup of coffee than a 7 minute roast. The ability to extend the roast to 15 minutes is the proof that the control electronics can influence the final outcome of the roasted coffee.

Total roasting time, however, is not the final delineator of roasting quality. The roasting profile, or the curve that represents the temperature versus time, can give the roaster user far more information about the ultimate quality of the coffee when it is brewed. An optimal roasting profile for beans from Guatemala is shown below in Figure 16.

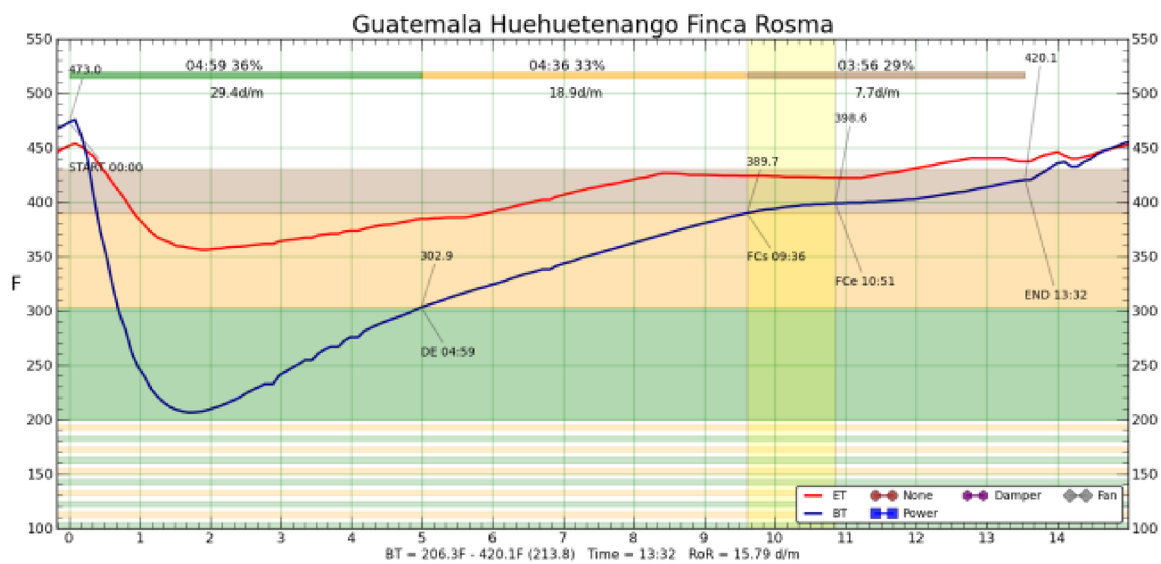

Figure 16: Example Roasting Profile of Guatemalan Beans

The roasting profile from one of my first roasts of the Guatemalan beans is shown below in Figure 17. It is certainly apparent that I was unable to obtain a consistent roasting curve of the bean temperature (blue curve) compared to the blue curve shown in Figure 16. The reason for such discrepancy between these two roasting curves is that I am not yet an experienced roaster. I am still trying to understand optimal rate of rise parameters, ramp and soak times, and I am still working to understand the audible and visual cues from the roast in order to react accordingly with the temperature control.

Figure 17: Roast Profile of Guatemala Huehuetenango

Cost Analysis

Date	Expenditure Description	Cost
9/21/13	TC4 Arduino Coffee Roasting Shield	$32.99
9/26/13	West Bend Poppery I	$59.00
10/21/13	Crydom D2425 Solid State Relay	$17.00
10/22/13	K-Type Thermocouple	$17.64
10/21/13	Crydom D2425-10 Solid State Relay	$60.86
11/28/13	Two-Pin Headers	$25.61
11/30/13	Brass Lamp with Glass Lantern	$15.90
12/5/13	Wooden Project Box	$9.99
12/5/13	Wire Connectors	$1.72
12/7/13	Various Gauge Hook-Up Wire	$8.06
12/16/13	Digikey Corporation - Electronic supplies	$8.27
12/23/13	Various Electrical Parts	$9.24
1/20/14	Zero Cross Detector Circuit	$25.00
1/15/14	Heat Sinks for Solid State Relays	$47.41
	Total:	$338.69

Table 2: Project Expenditures

Cost Analysis

The initial budget for this project was to remain under $500. The total component cost for this project was $338.69, which is approximately 38% under budget. Although the total component cost is greater than the basic hot air coffee roasters on the market, as shown in Table 1. But the total component cost remains below the price of the HotTop roasters which are either $820 or $1000.

From a financial standpoint, this roaster has good financial marketability. It has more control than the cheaper hot air coffee roasters and is cheaper than the more robust, HotTop roaster with similar temperature control as the HotTop line. Purchasing control components in bulk, improving roaster robustness, and sourcing inexpensive fans and heaters can reduce the total component cost to below $300.

Discussion and Conclusions

The successful development of a controllable hot air coffee roaster from a popcorn popper was the result of three terms of brainstorming, planning, research, and implementation. Transforming a popcorn popper without any form of control over the fan speed and heater power and implementing the software and sensors to visualize the roasting process was a daunting thought at the end of the first term of the capstone design course. With many options for temperature control, a variety of possible roasters to modify, and multiple options for roast logging software, designing the optimal system for my budget and roasting goals was a rewarding experience.

Through the capstone design process, I learned valuable lessons in time management, budgeting of available funds, and setting weekly, attainable goals. Taking advantage of the 6 week winter break was critical to the success of the project. Being able to easily shop for electronic components, constructing the project enclosure, and doing research for the project without the stress of other classes were benefits of working on the project at home. Once the Winter term started, diligently working on the project each week was the only way that I was able to meet the Week 8 presentation deadline.

I intend on completing robustness improvements to the roaster itself during the course of the spring term. For example, the solid state relays are not secured to the project enclosure. The zero cross detector and the Arduino/TC4 combination is not permanently secured to the inside of the project enclosure. I would also like to drill holes for the USB cable and the power supply cable. At the moment, the cables do not have a designated hole to enter the project enclosure and as a result, the enclosure is partly open during the roast. With the enclosure partly open, the chaff from the roast flies into the enclosure. In addition to these roaster improvements, I will also be improving my coffee roasting ability itself.

Roasting coffee is a completely different animal compared to the art of brewing coffee. The art of roasting is a skill that requires extensive practice. I will be working to develop this skill over the next term in an effort to create a more consistent roasting profile curve. As I roast each batch of beans, I will be able to better identify the different stages of the roast and react accordingly with the proper heater power and fan speed. My goal is to be able to tailor the roast to fit the specific flavor profile that I am looking for.

The entire development and construction of this roaster has been documented on Instructables.com. In an effort to give other home coffee roasting enthusiasts the opportunity to follow my instructions and documentation, step by step instructions were written for others to follow. An observation during my home coffee roasting research was that there was no central location to learn about how to modify a roaster, install the roasting software, run a roast, or source electronic components. The information was spread across many different forums, websites, and books, and I wrote the Instructable to allow for easier access into the home coffee roasting passion.

The project design experience allowed me to dive into many aspects of product design and electrical engineering applications. I was able to dive into software applications, control algorithms, power distribution circuitry, coffee roasting chemistry, and robust product design practices. Identifying safety issues such as electrical shock risk and coffee burning were also important to take into consideration during the design process.

The culmination of close to four years of education at Union College could not have been accomplished without the help and support from the entire ECE department, student research grant funding, and the unwavering support from both of my parents.

References

1. Davids, Kenneth. Home Coffee Roasting: Romance and Revival. St. Martin's Griffin; revised edition, November 2003. ISBN 978-0-312-31219-0. p. 7.
2. "Roast Profiling | Sweet Maria's Coffee Library." *Roast Profiling | Sweet Maria's Coffee Library*. N.p., n.d. Web. 08 June 2013.
3. "RoastLogger Overview." *RoastLogger Overview*. N.p., n.d. Web. 08 June 2013. http://homepage.ntlworld.com/green_bean/coffee/roastlogger/overview.html
4. T*hermocouple Type-K Glass Braid Insulated*. N.p., n.d. Web. 08 June 2013. https://www.sparkfun.com/products/251
5. "Arduino - HomePage." *Arduino - HomePage*. N.p., n.d. Web. 08 June 2013. http://arduino.cc
6. "How-To: Make a Popcorn Popper Coffee Roaster." *Engadget*. N.p., n.d. Web. 08 June 2013. http://www.engadget.com/2006/02/28/how-to-make-a-popcorn-popper-coffee-roaster/
7. "Miniature Accelerometer for High Temperature ESS and Vibration Testing." *Miniature High Temperature Accelerometer*. N.p., n.d. Web. 08 June 2013. http://www.dytran.com/go.cfm/en-us/content/etd3316c/x
8. *TC4 Wiring*. N.d. Photograph. *MLGP-LLC*. Web. 8 June 2013. http://www.mlgp-llc.com/arduino/public/tc4-wiring-small-20111120.jpg
9. "Thermocouple Response Time." *Thermocouple Response Time*. N.p., n.d. Web. 10 June 2013. <http://www.omega.com/temperature/Z/ThermocoupleResponseTime.html>.

10. "Hottop USA." *Hottop USA*. N.p., n.d. Web. 16 Oct. 2013. <http://hottopusa.com/Merchant2/merchant.mvc?Screen=PROD&Store_Code=HU&Product_Code=KN-8828B-2K&Category_Code=R>
11. "Cooking Up Traditions." *West Bend®*. N.p., n.d. Web. 24 Nov. 2013.
12. "Thermocouple." *Wikipedia*. Wikimedia Foundation, 22 Nov. 2013. Web. 25 Nov. 2013.
13. "Coffee Drinking Statistics." *Statistic Brain RSS*. N.p., n.d. Web. 29 Mar. 2014
14. "Specialty Coffee Association of America." *Specialty Coffee Association of America*. N.p., n.d. Web. 30 Mar. 2014.
15. "Urban Legends." *Cellini Caff*. N.p., n.d. Web. 30 Mar. 2014.
16. "Kitchen Appliances." *: Coffee Maker*. N.p., n.d. Web. 30 Mar. 2014.
17. "Homeroasters.org - Discussion Forum: Modifying the Poppery 1." *Homeroasters.org - Discussion Forum: Modifying the Poppery 1*. N.p., n.d. Web. 30 Mar. 2014.
18. http://www.crydom.com/en/Tech/Newsletters/Solid%20Statements%20-%20SSRs%20vs%20EMRs.pdf
19. http://www.crydom.com/en/Tech/Newsletters/Solid%20Statements%20-%20SSRs%20switching%20types.pdf
20. http://www.crydom.com/en/Tech/Whitepapers/HS_WP_HS.pdf
21. http://www.mlgp-llc.com/arduino/public/zcd-sch-006.pdf
22. "Thermocouples." *Thermocouples*. N.p., n.d. Web. 30 Mar. 2014.
23. "Fuji PXR3 Temperature Controller." *Test & Measurement Instruments with Engineering Support*. N.p., n.d. Web. 30 Mar. 2014.
24. "Green Coffee Freshness: How Old Is Too Old?" *Green Coffee Freshness: How Old Is Too Old?* N.p., n.d. Web. 30 Mar. 2014.

25. "Sweet Maria's Coffee." *Sweet Maria's Coffee*. N.p., n.d. Web. 30 Mar. 2014.

26. "Controllable Coffee Roasting Using Light Dimmers." *Too Much Coffee*. N.p., n.d. Web. 30 Mar. 2014.

27. "CHI(dot)INTELLI." *In Store Offering: The Pour Over Brew Bar at*. N.p., n.d. Web. 30 Mar. 2014.

Appendix

The following are excerpts of open source code that were written for the Artisan roasting software. This software was downloaded from: *https://code.google.com/p/tc4-shield/downloads/list* and the code was adapted tofit the correct baud rate and outlet frequency. The other code involved with the Artisan software is too long to fit into this appendix, so the following examples are shown.

phase_cntrl.cpp

```
// phase_ctrl.h
//
// Digital phase angle control on OT2 (random fire SSR drive)
// Connect zero cross detector to D3 (logic low indicates zero cross)
// Connect OT2 to random fire SSR
//
// ICC control on OT1.  Connect standard zero cross SSR to OT1.
// Period skipping (a.k.a. integral cycle control) method of AC
// control using zero crossing SSR's.  Most suitable for control
// of resistive loads, like heaters.
// Uses modified Bresenham algorithm (N in M) for ICC control.
// inspired by post on arduino.cc forum by jwatte on 10-12-2011 -- Thanks!

// created 14-October-2011

// *** BSD License ***
// ------------------------------------------------------------------------------------------
// Copyright (c) 2011, MLG Properties, LLC
// All rights reserved.
//
// Contributor:  Jim Gallt
//
// Redistribution and use in source and binary forms, with or without modification, are
// permitted provided that the following conditions are met:
//
//   Redistributions of source code must retain the above copyright notice, this list of
//   conditions and the following disclaimer.
//
//   Redistributions in binary form must reproduce the above copyright notice, this list
//   of conditions and the following disclaimer in the documentation and/or other materials
//   provided with the distribution.
//
//   Neither the name of the copyright holder nor the names of the contributors may be
//   used to endorse or promote products derived from this software without specific prior
//   written permission.
//
```

```
// THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND
CONTRIBUTORS "AS IS" AND ANY EXPRESS
// OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF
// MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
DISCLAIMED. IN NO EVENT SHALL
// THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
INDIRECT, INCIDENTAL,
// SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
LIMITED TO, PROCUREMENT OF
// SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR
BUSINESS INTERRUPTION)
// HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
CONTRACT, STRICT LIABILITY,
// OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
OF THE USE OF THIS
// SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
// ------------------------------------------------------------------------------------------

// July 1, 2012 -- Arduino 1.0 compatibility added by Jim Gallt

#ifndef _phase_ctrl_h
#define _phase_ctrl_h

#if defined(ARDUINO) && ARDUINO >= 100
#include <Arduino.h>
#else
#include <WProgram.h>
#endif

#include "timer1defs.h"
#include "user.h"

// define the pulse width for firing TRIAC (phase angle control)
#define TRIAC_PULSE_WIDTH 1000 // 500 uS default
#ifdef TRIAC_MOTOR
 #undef TRIAC_PULSE_WIDTH
 #define TRIAC_PULSE_WIDTH 4000 // 2000 uS needed for popper motor -- why?
#else ifdef TRIAC_HEATER
 #undef TRIAC_PULSE_WIDTH
 #define TRIAC_PULSE_WIDTH 1000 // 500 uS works for heaters
#endif

#define ZC_LEAD 1000 // zero cross signal leads the actual crossing by approx 500us

#define AC_TIMEOUT_MS 100 // 0.1 second
```

```
// for integral cycle control
#define RATIO_M 100 // resolution of quantization of output levels

// call when output levels need to change
void output_level_icc( uint8_t icc_level ); // call this to set output level, 0 to 100
void output_level_pac( uint8_t pac_level ); // call this to set output level, 0 to 100

// call to initialize integral cycle control
void init_control();

void setupTimer1();

// called at each zero cross by interrupt handler
void ISR_ZCD();

// detects the presence of AC
boolean ACdetect();

#endif
```

Phase_ctrl.cpp

```
// phase_ctrl.cpp
//
// Digital phase angle control on OT2 (SSR drive)
// Connect zero cross detector to D3 (logic low indicates zero cross)
// Connect OT2 to random fire SSR for small motor control.

// ICC (modified Bresenham) control on OT1.
// Connect OT1 to standard SSR.  Suitable for heater control.

// created 14-October-2011

// *** BSD License ***
// ------------------------------------------------------------------------------------------
// Copyright (c) 2011, MLG Properties, LLC
// All rights reserved.
//
// Contributor:  Jim Gallt
//
// Redistribution and use in source and binary forms, with or without modification, are
// permitted provided that the following conditions are met:
//
//   Redistributions of source code must retain the above copyright notice, this list of
//   conditions and the following disclaimer.
//
```

```
// -------------------------------------------------------------------------------------

#include "phase_ctrl.h"

extern int levelOT1;
extern int levelOT2;

// for phase angle control
enum output_state {delaying, pulse_on, disabled};
volatile output_state triac_state = disabled;
uint8_t pac_output; // output level, 0 to 100, for phase angle control

// for N:M quantization used by ICC control
int8_t ratioN; // ICC output level, 0 to 100
volatile boolean newN = true; // singles that output level has changed
volatile int8_t curr; // current value in N:M sequence
volatile uint32_t lastCross = 0;  // timer0 value at most recent zero cross
volatile boolean outputEnable = false;

// lookup table (index = rounded % output, 0 to 100)
// based on 0.5 uS per count
```

```
// FIXME: put this in PROGMEM
#ifdef FREQ60
uint16_t phase_delay[101] = { // 60Hz values based on linearizing power output
          /* 0      1      2      3      4      5      6     7      8      9 */
/* 00 */ 16667, 14733, 14218, 13851, 13554, 13301, 13076, 12874, 12687, 12514,
/* 10 */ 12352, 12198, 12052, 11912, 11778, 11649, 11525, 11404, 11287, 11173,
/* 20 */ 11061, 10953, 10846, 10742, 10640, 10540, 10441, 10343, 10248, 10153,
/* 30 */ 10060,  9967,  9876,  9786,  9696,  9608, 9520,  9432,  9346,  9259,
/* 40 */  9174,  9088,  9004,  8919,  8835,  8751, 8667,  8584,  8500,  8417,
/* 50 */  8333,  8250,  8167,  8083,  8000,  7916, 7832,  7748,  7663,  7578,
/* 60 */  7493,  7407,  7321,  7234,  7147,  7059, 6970,  6881,  6791,  6699,
/* 70 */  6607,  6514,  6419,  6323,  6226,  6127, 6027,  5924,  5820,  5714,
/* 80 */  5605,  5494,  5380,  5263,  5142,  5017, 4888,  4754,  4615,  4469,
/* 90 */  4315,  4153,  3979,  3793,  3590,  3366, 3113,  2816,  2449,  1933,
/* 100 */ 0
};
#else ifdef FREQ50
uint16_t phase_delay[101] = { // 50Hz values based on linearizing power output
          /* 0      1      2      3      4      5      6     7      8      9 */
/* 00 */ 20000, 17680, 17061, 16621, 16265, 15961, 15692, 15448, 15225, 15017,
/* 10 */ 14822, 14638, 14462, 14295, 14134, 13979, 13830, 13685, 13544, 13407,
/* 20 */ 13274, 13143, 13016, 12891, 12768, 12647, 12529, 12412, 12297, 12184,
/* 30 */ 12072, 11961, 11851, 11743, 11636, 11529, 11423, 11319, 11215, 11111,
/* 40 */ 11008, 10906, 10804, 10703, 10602, 10501, 10401, 10300, 10200, 10100,
/* 50 */ 10000,  9900,  9800,  9700,  9599,  9499,  9398,  9297,  9196,  9094,
/* 60 */  8992,  8889,  8785,  8681,  8577,  8471,  8364,  8257,  8149,  8039,
/* 70 */  7928,  7816,  7703,  7588,  7471,  7353,  7232,  7109,  6984,  6857,
/* 80 */  6726,  6593,  6456,  6315,  6170,  6021,  5866,  5705,  5538,  5362,
/* 90 */  5178,  4983,  4775,  4552,  4308,  4039,  3735,  3379,  2939,  2320,
/* 100 */ 0
};
#endif

// timer1 is used for both phase delay and for TRIAC pulse width timing
void setupTimer1() {
  TIMSK1 = 0; // disable all interrupts
  TCCR1A =  0; // put timer1 in normal mode; output pins under sketch control
  TCCR1B = _BV(TCCR1B_CS11); // set prescaler to clk/8 (1 count = 0.5 uS)
  OCR1A = 0xFFFF; // initialize output compare register A to max value
  TIMSK1 = _BV(TIMSK1_OCIE1A); // enable interrupt on output compare A match
  TCNT1 = 0; // set the timer to zero
}

// ------------------------- ISR for external zero cross detect
void ISR_ZCD() {
  TCNT1 = 0;  // reset timer1 counter
```

```
  // perform AC monitoring
  lastCross = millis(); // timer0
  // first, handle the phase angle control output
  triac_state = delaying;
  if( outputEnable )
    digitalWrite( OT_PAC, LOW ); // force output off
  // set output compare register A for delay time
  OCR1A = phase_delay[pac_output] + uint16_t(ZC_LEAD);

  // next, handle the integral cycle control output using modified Bresenham's algorithm
  // (inspired by post on arduino.cc forum by jwatte on 10-12-2011 -- Thanks!)
  if( newN ) {
    // restart sequence if new output level
    curr = int8_t ( - ( int16_t( int16_t(ratioN) + int16_t(RATIO_M) ) ) / 2 );
    newN = false;
  }
  curr += ratioN;
  if( curr >= 0 ) {
    curr -= RATIO_M;
    if( outputEnable )
      digitalWrite( OT_ICC, HIGH );
  }
  else {
    digitalWrite( OT_ICC, LOW );
  }
}

// ------------------------ ISR for comparator A match
ISR( TIMER1_COMPA_vect ) { // this gets called every time there is a match on A
  // if triac output is delaying, then
  if( triac_state == delaying ) {
    triac_state = pulse_on; // indicate output pulse is active
    if( outputEnable )
      digitalWrite( OT_PAC, HIGH );
    TCNT1 = 0; // reset timer count
    OCR1A = TRIAC_PULSE_WIDTH; // start counting for pulse
  }
  else if( triac_state == pulse_on ){  // if triac output is on, turn it off because pulse is done
    triac_state = disabled;
    digitalWrite( OT_PAC, LOW );
    TCNT1 = 0;  // reset timer count
    OCR1A = 0xFFFF; // keep triac output off until next zero cross
  }
}

// initialize ICC and PAC control
```

```
void init_control() {
  output_level_icc( 0 );
  output_level_pac( 0 );
  pinMode( OT_ICC, OUTPUT );
  pinMode( OT_PAC, OUTPUT );
  pinMode( INT_PIN, INPUT ); // enable input on the interrupt pin
  digitalWrite( INT_PIN, HIGH );  // enable internal pullup on the int pin
  triac_state = disabled;
  setupTimer1();
  attachInterrupt( EXT_INT, ISR_ZCD, FALLING );
}

// call this to set phase angle control output levels, 0 to 100
void output_level_pac( uint8_t pac_level ) {
  if( pac_level < OT1_CUTOFF ) { // if new levelOT2 < cutoff value then turn off OT1
    output_level_icc( 0 );
  }
  else {  // turn OT1 back on again if levelOT2 is above cutoff value. Might be a better way to
handle this??
    output_level_icc( levelOT1 );
  }
  if( pac_level > 100 ) // trap error condition
    pac_output = 0;
  else
    pac_output = pac_level;
}

// call this to set integral cycle control output levels, 0 to 100
void output_level_icc( uint8_t icc_level ) {
  if( levelOT2 < OT1_CUTOFF ) icc_level = 0;
  if( icc_level > 100 )
    ratioN = 0;
  else
    ratioN = icc_level;
  newN = true;  // tell the interrupt routine to restart sequence
}

// detects the presence of AC
boolean ACdetect() {
  return outputEnable = ! ( ( millis() - lastCross ) > AC_TIMEOUT_MS ) ;
}
```