Toaster Reflow Oven (Easy Bake PCBs)

ECE 445 – Final Paper

Bhaven Shah, Zak Kaminski, and Raghav Narasimhan

TA: Prannoy Kathiresan, Professor Arne Fliflet

3 May 2023

Team 51

Abstract

The reflow oven is a tool to help with soldering PCBs as soldering is a time-consuming process that requires a high level of skill to do properly. The commercial reflow ovens available can cost upwards of $250 dollars. These products come with rudimentary features such as buttons and are very difficult to create new profiles, if it even allows for custom profiles. The other issue with the commercially available reflow ovens is that they can be too expensive for the generic PCB enthusiasts. Thus, we decided to make reflow ovens more accessible to hobbyists by creating a controller to pair with a basic toaster oven. To take it a step further, we wanted to implement more safety features such as a camera system that will be able to determine when a component moves of the pad during the reflow process.

TABLE OF CONTENTS

1 Introduction

1.1 Problem

Surface Mount Devices (SMD) are electrical components used on Printed Circuit Boards (PCBs). These tiny components can be as small as 0.016in by 0.008in and are attached to the PCBs via a process called soldering. Hand soldering, the most common soldering process, is a very time-consuming task that can produce many errors or bad connections as this is a skill that requires practice and knowledge of the subject. The size of some of the more common SMD components, “0805, …, 0603, or 0402” compound the difficulty of the soldering process as precision is required to successfully solder the component without damaging anything [1]. As a result, testing and debugging often require many hours due to imperfect solder joints, which decreases efficiency and productivity. Thus, a tool called the reflow oven was created to improve the efficiency and reduce the errors of hand soldering. A new technique, reflow soldering, was invented alongside the reflow oven and allows for one to hundreds of SMD components to be attached to the contact pads on a PCB using controlled heat. This process uses thermal profiles to accurately melt the solder and create near perfect joints. Although it is an excellent technology, it is one that has very limited access. Most reflow ovens are commercial units with large price tags that individuals or small business owners have capital for. While there are consumer products, these too have high prices that do not provide results expected at such a price. Do-it-Yourself (DIY) kits to convert a toaster oven into a reflow oven exist as well, but they are never in stock due to part shortages and come with their own set of issues. Additionally, the consumer models and DIY kits do not offer any sort of advance features such as stopping the reflow process if the components move during the reflow process causing a failed connection.

1.2 Solution

As a solution to the issues of availability, price, decreasing the rate of error and increasing efficiency we are creating our own DIY kit that will address the concerns. We will design our kit using the parts that are commonly in stock as to best avoid having issues with part shortages. Next, we will develop the kit to be used on almost any toaster oven if it meets a basic requirement of providing 1000 W or greater. This wattage is necessary as a requirement for the reflow process is for the heating chamber to achieve a temperature of over 250 °C to ensure that the solder will reach a liquified state. Finally, our biggest improvement will be adding a camera with a top-down view to oversee the reflow process. This camera paired with some image recognition software will determine if components move too far off the contact pads and then alert the user and stop the reflow process as to not create poor solder joints.

1.3 High Level Requirements

Our DIY reflow oven has three requirements for it to be considered a success. These requirements are:

1. The reflow oven must be able to solder PCBs with a margin of error of 10%.
2. The camera system of the reflow oven must be able to detect when a component covers less than 50% of the pad.
3. The reflow oven must be able to reach temperatures of 270 °C with a tolerance of ± 10 °C.

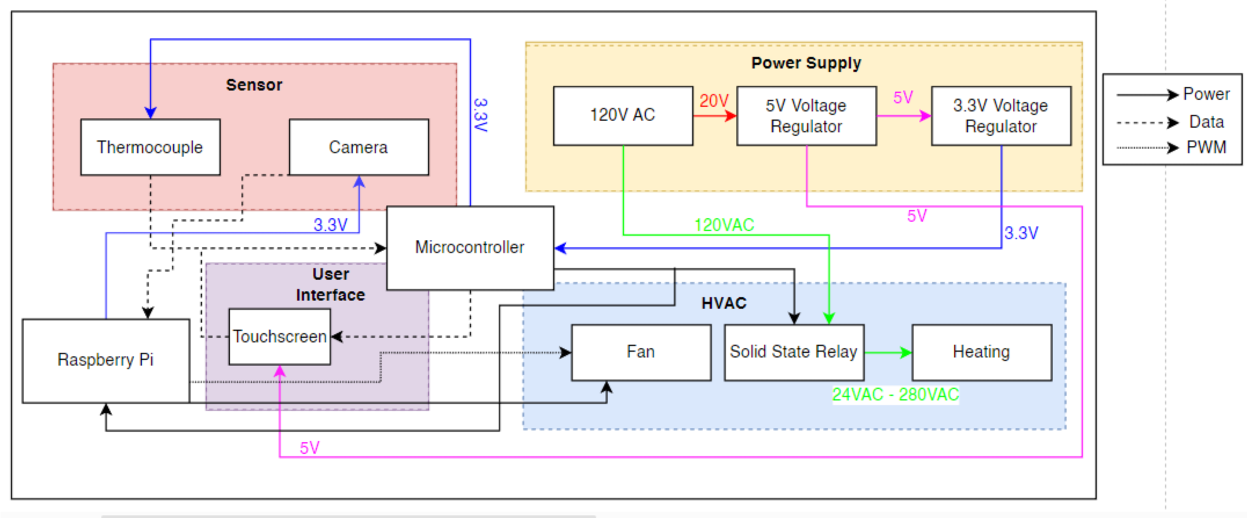

Figure 1: DIY Reflow Oven Block Diagram

2 Design

2.1 Introduction

The Easy Bake PCBs consists of four major subsystems as seen in Figure 1. These are the sensor, HVAC, user interface, and power supply subsystems. The sensor subsystem is dedicated to measuring temperature and visual data by relaying it to the microcontroller. The HVAC subsystem deals with the heating and cooling of the reflow oven in order to correctly follow solder profiles needed to appropriately solder surface mount components. The user interface subsystem oversees taking in user inputs to show visual feedback to the user and allow the user to determine what profiles to use and when to start and stop the reflow process. Finally, the power supply subsystem is simply needed to provide power at the required voltage levels to the various other components used across the design.

2.2 Design

2.2.1 Sensors Subsystem

The sensor subsystem consists of the thermocouple circuit, camera, and microcontroller. The thermocouple and camera communicate data to the microcontroller, which then uses the supplied data to update

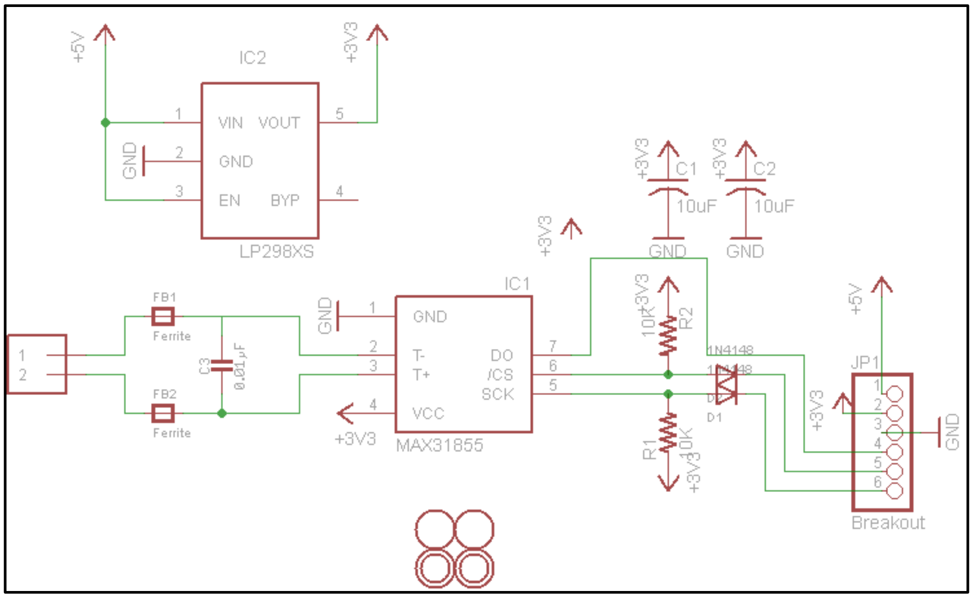

Figure 2: Thermocouple Circuit Design

the status of the touchscreen, fan, and solid-state relay. The microcontroller relies on these sensors to understand whether our product is functioning as intended. It is imperative that these sensors provide accurate and meaningful information to our microcontroller. The microcontroller is the brain of the operation and must communicate effectively with the other components. The thermocouple circuit shown in Figure 2 [2] takes the readings acquired by the thermocouple and converts them into meaningful data to be sent to the microcontroller via SPI. This circuit uses the MAX31855 Cold-Junction Compensated Thermocouple-to-Digital Converter, which enables the conversion of the thermocouple data into readable SPI data for the microcontroller [3]. For a K-type thermocouple, we want to connect the chromel wire to T+ and the alumel wire to T-. This circuit also requires ferrite beads labeled as FB1 and FB2 to reduce the electromagnetic interference between the leads and the rest of the circuit.

The next sensor is the camera. We originally chose the ArduCAM-M-2MP camera as seen in Figure 3 [4] which uses an SPI interface to communicate with the microcontroller and I2C for sensor configuration [5], but later pivoted to the ArduCAM RPI-CAM-V2. The reason for this change was the camera we had originally chosen was deprecated and lacked support for the rest of our system. Additionally,the resoloution of this camera is 8 MP which is four times the resolution of the previous camera. This camera communicated with our Raspberry Pi via USB interface and was powered through the Raspberry Pi.

Figure 3: SPI Camera

The microcontroller is the final part of the sensor subsystem. We have decided to use the ATSAMD51J20 Cortex M4 microcontroller which was the microcontroller that was on the touchscreen board. This microcontroller has dedicated SPI and I2C pins, but I2C pins were available. Thus, we had to use a software trick to override the I2C pins to become GPIO pins that would work with our SPI thermocouple. Using the SDA pin as our input from the MAX31855, SCL was connected to SCLK on the MAX31855, and a GPIO pin was connected to the chip select pin on the MAX31855. The touchscreen was connected to the microcontroller using dedicated touch pins that was 8-bits. This uses eight data lines and four to five control lines. Another GPIO pin will be used as a data signal to open or close the Solid-State Relay and by extension power the heating coils.

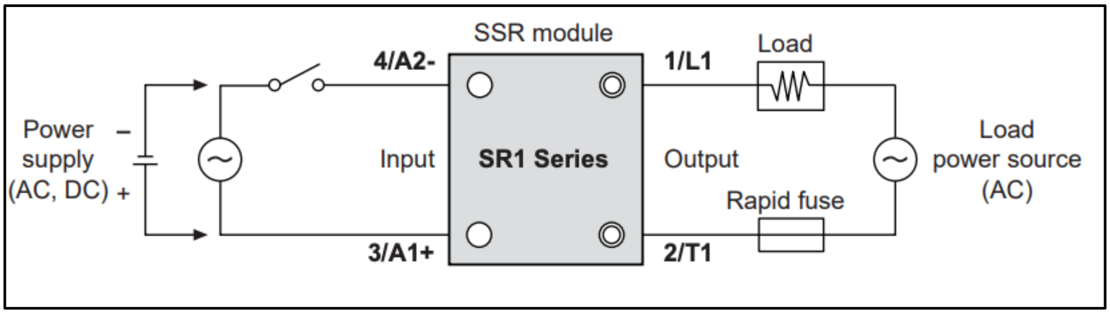

Figure 4: Solid State Relay Circuit

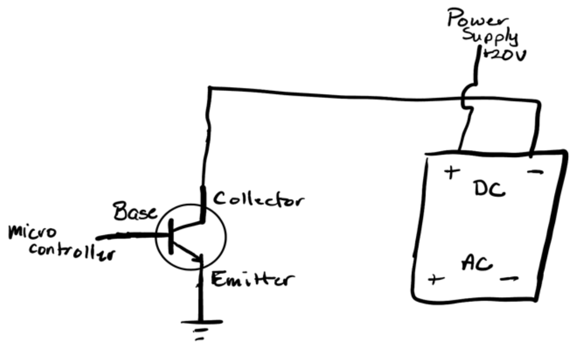

Figure 5: BJT Circuit

2.2.2 HVAC Subsystem

The HVAC subsystem is responsible for heating and cooling the chamber of the toaster oven based on data received from the microcontroller. The microcontroller uses the temperature reading from the thermocouple and the current phase of the reflow process to determine whether to turn on the fan or the heating system.

2.2.2.2 Heating System

The heating system consists of an SR1-1210 Solid State Relay, as shown in Figure 4, which cuts the one-phase power from the wall to the toaster oven. It takes a GPIO input from the ATSAMD51J20 Cortex M4 which then runs through a TL431 BJT transistor as we ran into issues with not having enough current to power our Solid State Relay. As show in Figure 5, we had the GPIO pin connected to the Base, the Emitter to ground from the power supply, and +20 VDC (Voltage Direct Current) coming from our external power supply. This solution would provide the necessary 0.7 mA (milliamps) to properly control the Solid State Relay.

Table 1: JEDEC Classification Reflow Profile

Profile Feature	Sn-Pb Eutectic Assembly	Pb-Free Assembly
Average ramp-up rate (T_{smax} to Tp)	3° C/second max.	3° C/second max.
Preheat - Temperature Min (Ts_{min}) - Temperature Max (Ts_{max}) - Time (Ts_{min} to Ts_{max}) (ts)	 100 °C 150 °C 60-120 seconds	 150 °C 200 °C 60-180 seconds
Time maintained above: - Temperature (T_L) - Time (t_L)	 183 °C 60-150 seconds	 217 °C 60-150 seconds
Peak Temperature (Tp)	See Table 4.1	See Table 4.2
Time within 5°C of actual Peak Temperature (tp)[2]	10-30 seconds	20-40 seconds
Ramp-down Rate	6 °C/second max.	6 °C/second max.
Time 25°C to Peak Temperature	6 minutes max.	8 minutes max.

This GPIO serves as an input for the solid-state relay, which provides the toaster oven heating coils with AC voltage. This would then heat the coils and increase the temperature of the chamber based on the JEDEC specified profile which can be seen in Table 1 or the solder past specified profile.

2.2.3 User Interface Subsystem

The User Interface Subsystem is responsible for serving as a liaison between the user and the Reflow Oven. With the User Interface subsystem, the user can communicate directly with the Reflow Oven and perform functions such as starting the process, stopping the process, and selecting a profile. Users can also visualize the soldering process take place through the graph that is displayed when a process is started.

2.2.4 Power Supply Subsystem

The power supply subsystem is responsible for providing power to all the electronics on our board, as well as for the heating and cooling of the toaster oven. This subsystem includes the toaster oven's 120 V ac power supply, an off-the-shelf 24 V adjustable power supply, a 5 V regulator, and a 3.3 V regulator. The power supply is powered by the 120 V ac wall outlet and converts that to a 24 V dc output that we can adjust to be 20 V. The 5 V regulator is powered by the 20 V output of the power supply to obtain a 5 V DC output. The 5 V output is necessary to power the fan and the solid-state relay of the HVAC subsystem. The 3.3 V regulator is supplied by the 5 V output of the 5 V regulator and is used to power the remaining electronics, such as the microcontroller, thermocouple,

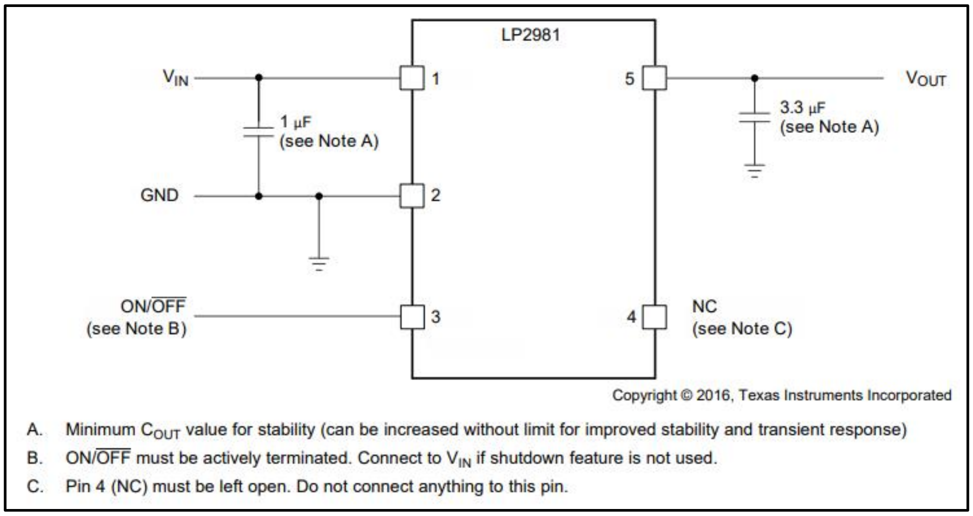

Figure 6: LP2981 Common Application Setup

Table 2: LP2981 Parameters

PARAMETER	DESIGN REQUIREMENT
Input voltage	5 V ±10%, provided by an external regulator
Output voltage	3.3 V ±5%
Output current	100 mA (maximum), 1 mA (minimum)
RMS noise, 300 Hz to 50kHz	< 1 mV_{RMS}
PSRR at 1kHz	> 40 dB

and camera. For this, we will use the LP2981 shown in Figure 6 from its datasheet [6] while following the parameters as described in Table 2. The requirement table for these components can be found in Table 6 in Appendix A.

2.3 Software Design

2.3.1 Touchscreen User Interface

The Adafruit PyPortal CircuitPython Powered Internet Display will be programmed using the Python programming language. We will use the Mu Editor to develop a user interface as it allows us to quickly save and test our code on the display. When the interface starts, it will display the default profile. On this screen, it will show the Alloy, Profile, and current temperature. From here, the user has the option to switch profiles by selecting the change profiles button. From here, another screen is displayed with a list of available profiles. The user can select their desired profile here and start the process. As the process begins, three things on the display will change: the time, the temperature, and the graph. The temperature will change depending on the profile and these changes can be visualized on the graph.

2.3.2 Component Slip Detection

The component slip detection system consists of a camera that will send data to the microcontroller and will determine if the soldering is going as intended. We will be programming this such that the difference between the image where the component is perfectly aligned (the beginning of the process) and any subsequent picture is analyzed. These pictures will be captured in 0.5-second intervals. We used OpenCV as well as the Scikit-image library to compute the numerical difference between the two pictures. The algorithm would return a score between –1 and 1 where –1 indicates two completely dissimilar images and 1 indicates identical images. If the score falls below a certain threshold, the user is notified that a slip has occurred and that the process should halt.

2.4 Commercial Component Selection

2.4.1 Toaster Oven

For our toaster oven we are using an old toaster oven that a group member had in their apartment. It fit the minimum requirements of 1000W. It is also on the smaller size for toaster ovens, meaning that it would be better for quicker and even heating of the internal chamber. We are going this route to show that expensive components are not necessary and that old used up toaster oven can be repurposed into another functional tool instead of ending up in a landfill.

2.4.2 Touchscreen

For our touchscreen, we pivoted from using the TFT LCD 3.2’ 240x320 RGB SPI Display with Touchscreen and instead used the Adafruit PyPortal. The reason we made this change was because the touchscreen we originally had purchased was deprecated and lacked any formal documentation. The Adafruit PyPortal featured a very intuitive user interface and could also connect to the Internet. It also has a microSD slot as well as a speaker which was used for notifying the user when the phases of the reflow cycle changed. Making this change in our touchscreen was incredibly beneficial to the success of our project due to the fact that we utilized the microcontroller on the touchscreen as our main microcontroller.

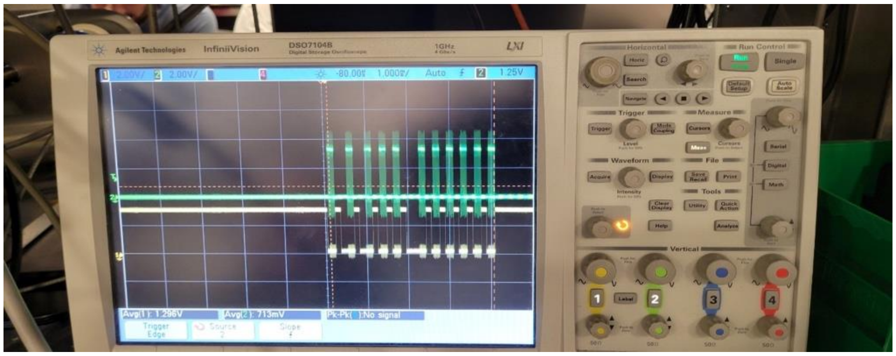

Figure 7: Time Difference Between Consecutive Temperature Readings

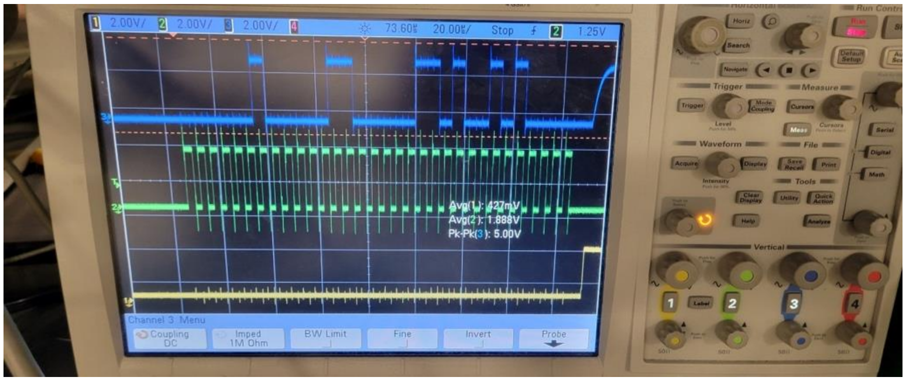

Figure 8: Example Temperature SPI Signal

3 Design Verification

All requirement and verification tables can be found in Appendix A.

3.1 Sensor Subsystem

We will begin by looking at the thermocouple circuit. This portion of the design must be able to send our temperature data within 0.4 to 0.6 seconds and accurately measure our temperature within a degree Celsius. We can see in Figure 7 that the time it takes to send five temperature readings is less than 2 ms (milliseconds). This is well under the desired maximum time given for a temperature reading. We can also see in Figure 8 an example of the temperature reading in blue and we can convert this SPI data into the temperature reading and find that it accurately gives us the room temperature of 25 degrees Celsius.

The next sensor is the camera. For the camera, our requirements were to ensure that the camera can send a frame once every 0.5 ± 0.1 second. To verify this requirement, we wrote a Python script that communicated with our camera and captured a picture every 0.5 seconds. This picture was then saved to the local file storage of our Raspberry Pi. For our second requirement, we used OpenCV to calculate the difference between two images.

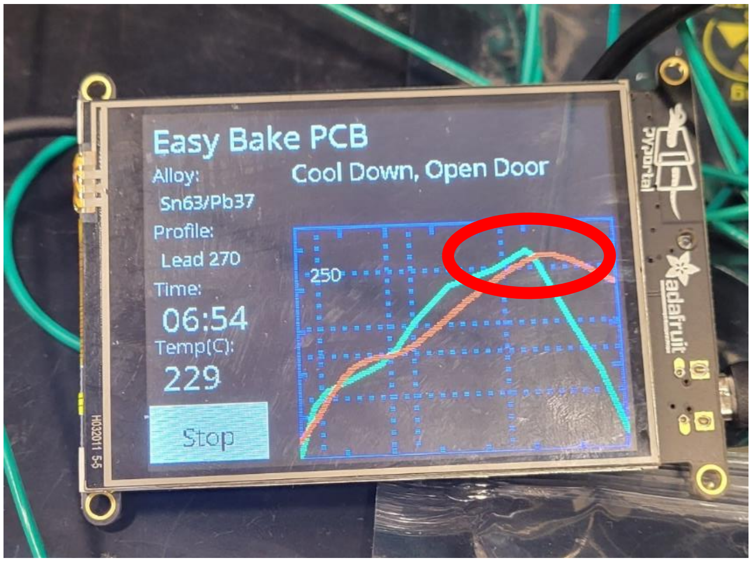

Figure 9: Thermocouple reaching 270 Celsius

We took three pictures, one where a test component was placed perfectly on the PCB, one where the component was slightly misaligned but still within our margin of error, and one where the component completely slipped. From here, the difference between the images was computed and assigned a value between –1 and 1 where –1 signifies two completely different pictures and 1 signifies identical images. The algorithm was able to detect when the component had slipped above our threshold and output that a slip had occurred.

Finally, we look at the microcontroller. This just required the sending and reading of the appropriate digital values of zero or one for the corresponding voltage levels of below 0.2 V and above 3.0 V respectively. These are easily verified by the fact that we were able to correctly read our thermocouple SPI signal as this is only possible if the microcontroller can output and read digital ones and zeros at the voltage levels required. We ended up not requiring a PWM signal as we could utilize a regular GPIO pin to do the same functionality instead.

3.2 HVAC Subsystem

The HVAC subsystem as we progressed through the semester was continually modified and eventually was changed into only having the Heating section of the HVAC subsystem. This was because through testing, we learned that a PWM cooling fan was unnecessary and added a level of complexity that was not needed. We were able to reach our requirement of reaching 270 °C with a tolerance of 10 °C. This is shown in Figure 9 where the peak of the red line is about approximately 268 °C. The other requirements for our HVAC subsystem required our

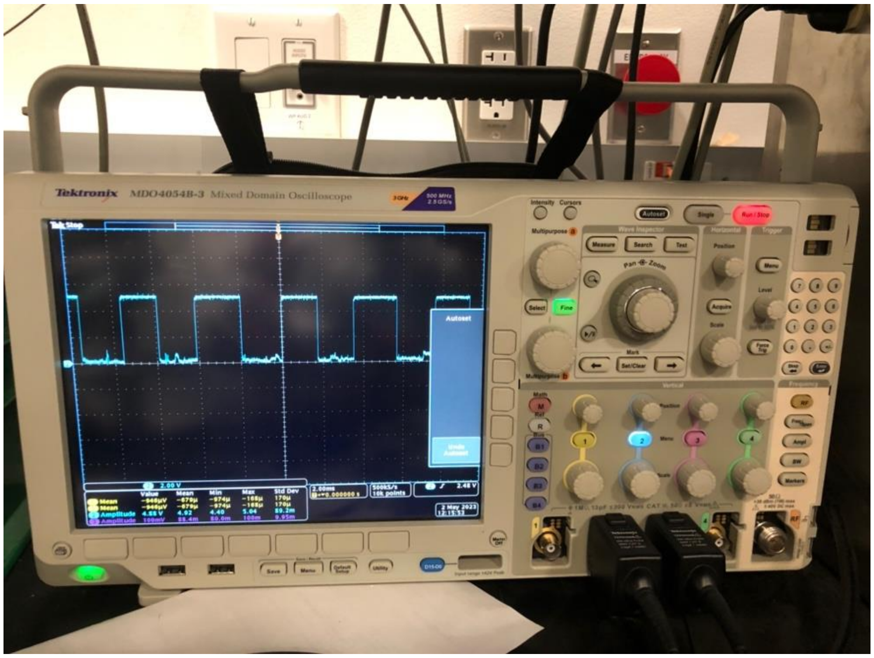

Figure 10: Fan Tachometer readout at 90% Duty Cycle

fan to reach certain RPMs (Revolutions per Minute) by controlling the PWM signal that was sent to the fan. By adjusting the duty cycle and reading the tachometer pin we could calculate the RPMs. Using the waveform generator, we were able to find the required duty cycles, with 5 V, to run the fan at necessary speeds. With the frequency at 29 kHz and Amplitude at 5 V we found that to run at 7600 RPM a 90% duty cycle was needed. To reach 8300 RPMs, we used the same settings, but now with a 99% duty cycle. Using the equation, $RPM = 60 * \frac{f}{2}$, we can take the frequency in Figure 10 which is 90% duty cycle we will get an RPM of approximately 7680 RPM. Similarly for Figure 11 with a duty cycle of 99% and a frequency of approximately 267.5 Hertz, we get an RPM of about 8010.

3.3 User Interface Subsystem

For the User Interface Subsystem, our requirements were that the system should be able to register 99% ± 5% of user inputs, verify that there are minimal 'dead zones' in the touch screen that would affect user inputs, and should be able to Create, Read, and Update preset soldering profiles with a maximum of three profiles. Verifying the first requirement was very simple. As we interacted with the user interface, we ensured that all our

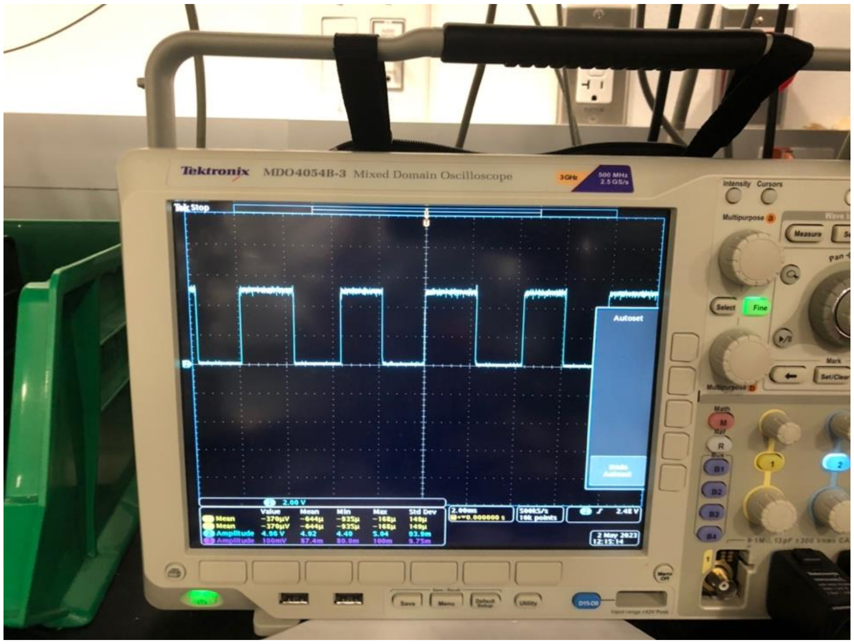

Figure 11: Fan Tachometer readout at 100% Duty Cycle

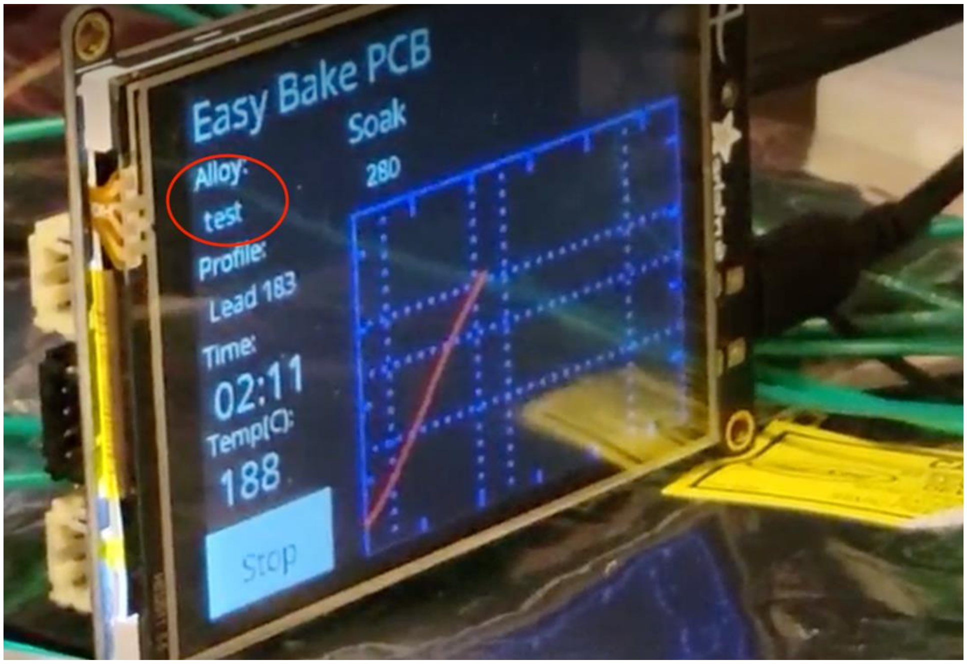

Figure 12: Added Profile Example

inputs were getting registered, and the appropriate action would take place. In our user interface, the two main inputs were the start/stop button and the profile button. When these buttons were pressed, the inputs were registered 100% of the time. The second requirement was verified in the same manner. For the third requirement, we created a test profile and ensured that we were able to view this profile from the touchscreen as well as update it. This added profile can be seen in Figure 12.

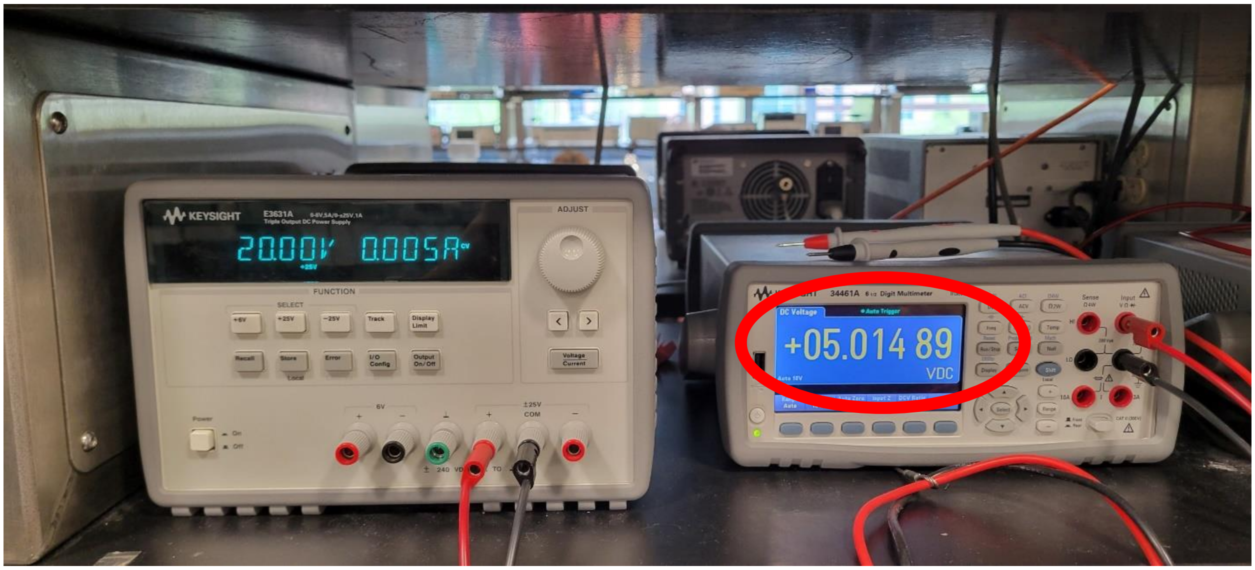

Figure 13: 5 V Regulator Output Given a 20 V Input

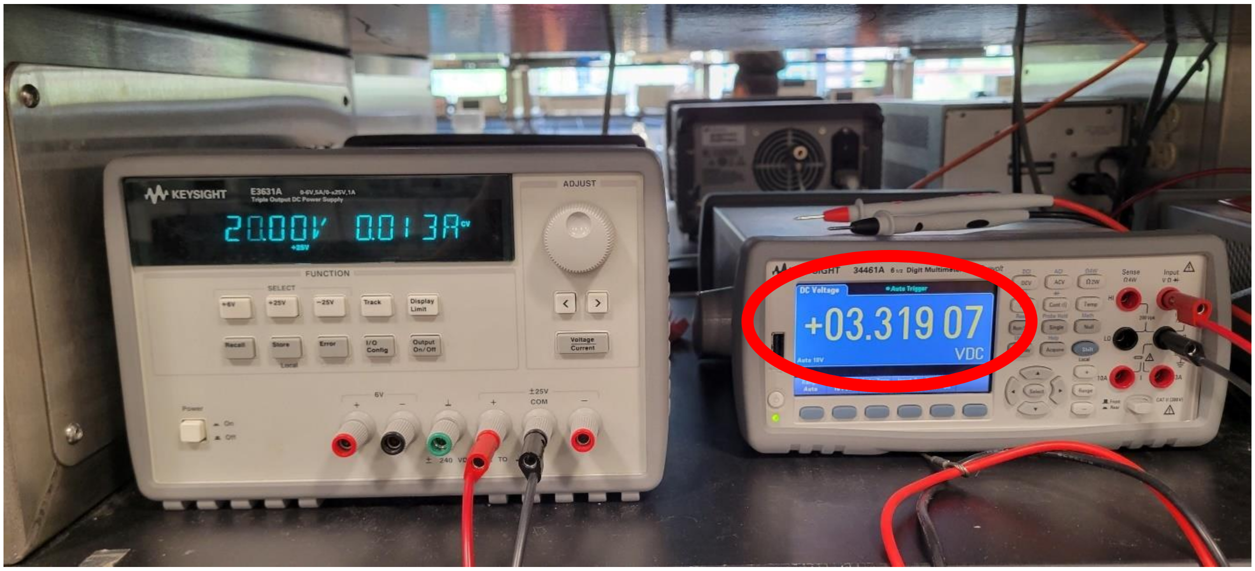

Figure 14: 3.3 V Regulator Output Given a 5 V Input

3.4 Power Subsystem

The power subsystem is very straightforward in its verification procedure. Given the input of the 20 V power supply, we can measure the voltage at each regulator which the 5 V regulator output can be seen in Figure 13 and the 3.3 V regulator is Figure 14. Both values are well within the tolerances specified in our requirement and verification table.

4 Cost and Schedule

4.1 Parts Cost

The total list of parts can be found in Table 8 in Appendix B. The total for all the electrical components is $139.50. If we include the MSRP price for the toaster oven the total becomes $184.49. The MSRP price for the toaster oven has been included but was purchased for a cheaper price on 3rd party sites such as Craigslist and Facebook Marketplace. It is recommended and expected that the toaster oven will be purchased on such sites to save on costs and recycle old machines. We will also need to include a 9% sales tax and an 5.5% shipping cost estimate to bring the total up to $199.87.

$$(184.49) + (184.49 \cdot .09) + (184.49 \cdot .055) = \$211.24 \quad (1)$$

4.2 Labor Cost

Given that the average salary of a computer engineer and an electrical engineer that graduated from the ECE department from UIUC in 2021 was $105,352 and $80,296 respectively, the average salary is $92,824 [7]. Then, given that we work 40 hours a week and 52 weeks in a year, the hourly salary becomes $44.63/hour.

$$\frac{10{,}5352+80{,}296}{2} = 92{,}824 \qquad \frac{92{,}824}{40*52} = 44.63 \quad (2)$$

Then, the total cost for one team member at a rate of $44.63/hour would be equal to $21,422.40. Then the total labor cost for three team members will be $64,267.20.

$$\frac{\$44.63}{hr} * \frac{30hrs}{week} * 16\ weeks = \$21{,}422.40 \quad (3)$$

4.3 Total Cost

The total cost for this project will include the Labor Costs for all three team members—$64,267.20— and the cost for the parts—$211.24—which will total $64,478.44 for the whole project.

4.4 Schedule

The schedule for all team members can be found in Table 9 in Appendix B.

5 Conclusion

5.1 Accomplishments

We accomplished almost all the parts of the project minus the camera system which we had working individually, but still needed to be fully integrated into our project. We were able to successfully program our new microcontroller to control the reflow process, we were able to successfully heat to the required temperature while following the specified reflow profile, and we were able to get accurate temperature readings from the thermocouple. The camera system was more of a proof of concept that was working by itself on the Raspberry Pi, but still needed to be integrated into the full system.

5.2 Uncertainties

While we were able to get the main functionality of the project working, there were parts that we were unable to get fully working. This includes the camera system, the original PCB and microcontroller we were going to use, and the link switch part of the PCB. One of the biggest issues with our PCB was that we forgot to provide power to the microcontroller thus making controlling anything on our PCB impossible without some bodge wires or making and ordering a new PCB. We also ran into some communication protocol issues with our thermocouple and its connection to the PyPortal. Our thermocouple communicates via SPI and the only available pins on the microcontroller were I2C. So, we had to manipulate the I2C pins via the software to trick them into being SPI pins so that it would work with our thermocouple. We had some inconsistencies with the output of the thermocouple, but we figured out that the issue was withing our bodge wires on our PCB that was creating this issue.

5.3 Ethical Considerations

From looking at our block diagram, we recognized that the block that poses the most danger to our user's safety is the HVAC system. As stated previously, this block is responsible for getting the reflow oven to reach very high temperatures in order to perform the solder properly. In reaching these high temperatures, a lot of safety issues could arise such as the entire reflow oven combusting or the user getting burned by accidentally touching a component that gets very hot. Additionally, powering this system consists of a 120V AC power supply. This can pose a lot of risk if not properly handled.

In addition to this, there are various health risks with soldering. One of these risks includes rosin exposure [8]. Rosin exposure comes from the solder flux and when soldering occurs, the flux produces smoke which can cause irritation in the eyes, throat, and lungs. It can also cause other issues such as nose bleeds and headaches. Our user's safety is our highest priority, and we will take the appropriate actions to ensure that we will "hold paramount the safety, health, and welfare of the public" [9]. In order to prevent our users from getting exposed to these dangerous fumes we will ensure that any fumes are contained, and the user is not directly exposed to any of these fumes. Additionally, we will heavily emphasize the use of proper lab ware when operating our reflow oven. These include masks, gloves, and goggles.

5.4 Future Work

Our future work would be to start with creating a PID controller to follow more closely the reflow profile as we noticed that it would tend to loosely follow the profile curve. The next step would be to modify the PCB to better fit the functionality that we need. This new PCB would be redesigned using the ATSAMD51J20 Cortex M4 as our microcontroller and then include a connector for a camera, connector for the touchscreen, input for our thermocouple, an output to control our Solid-State Relay, and a speaker to signify the changing of stages and a fault in the reflow process. This would be the high-level design of the PCB, and there would be more included in the PCB design. Once we have this new PCB made, the next step would be to create an enclosure to contain everything to make it into a nice package. This would make the product nice to use and take up less space.

References

[1] "Resistors," *Resistors - SparkFun Learn*. [Online]. Available: https://learn.sparkfun.com/tutorials/resistors/types-of-resistors. [Accessed: 16-Feb-2023].

[2] L. Ada, "Max31855 thermocouple," Adafruit Learning System, 29-Jul-2012. [Online]. Available: https://learn.adafruit.com/thermocouple/downloads. [Accessed: 18-Feb-2023].

[3] "MAX31855 Cold-Junction Compensated Thermocouple-to-Digital Converter" Feb-2012. [Online]. Available: https://cdn-shop.adafruit.com/datasheets/MAX31855.pdf. [Accessed: 18-Feb-2023].

[4] "Arducam mini 2MP plus - OV2640 SPI camera module for Arduino Uno MEGA2560 Board & Raspberry Pi Pico," Arducam, 30-Oct-2022. [Online]. Available: https://www.arducam.com/product/arducam-2mp-spi-camera-b0067-arduino/. [Accessed: 20-Feb-2023].

[5] "ArduCAM-M-2MP Camera Shield," Feb-2015. [Online]. Available: https://www.uctronics.com/download/Amazon/ArduCAM_Mini_2MP_Camera_Shield_DS.pdf. [Accessed: 22-Feb-2023].

[6] LP2981 100-mA Ultra-Low Dropout Regulators With Shutdown (2004). Available at: https://www.ti.com/lit/ds/symlink/lp2981.pdf [Accessed: 18-Feb-2023].

[7] Grainger Engineering Office of Marketing and Communications, "Salary averages," Electrical & Computer Engineering | UIUC. [Online]. Available: https://ece.illinois.edu/admissions/why-ece/salary-averages. [Accessed: 23-Feb-2023].

[8] I. Slack, "Soldering safety," Department of Engineering Health & Safety, 27-Mar-2018. [Online]. Available: https://safety.eng.cam.ac.uk/safe-working/copy_of_soldering-safety#:~:text=Rosin%20exposure&text=Flux%20generates%20the%20visible%20fumes,sensitisation%2C%20causing%20and%20aggravating%20asthma. [Accessed: 22-Feb-2023].

[9] “IEEE Code of Ethics,” IEEE, Jun-2020. [Online]. Available:

https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 22-Feb-2023].

Appendix A Requirement and Verification Tables

Table 3: Requirement Verification Table for Sensors and Microcontroller

Requirement	Verification
1. Thermocouple must be accurate within ± 2°C	1. Assemble the thermocouple circuit and make the appropriate connections to the microcontroller. 2. Place the thermocouple in an environment with a known temperature (recommend room temperature). 3. Power the thermocouple circuit and the microcontroller with 3.3V 4. Print the measured values on the console and check they are within 2°C of the known temperature
1. Thermocouple must be able to give a temperature every 0.5 ± 0.1 second	1. Assemble the thermocouple circuit and make the appropriate connections to the microcontroller. 2. Power the thermocouple circuit and the microcontroller with 3.3V 3. Take ten consecutive measurements from the thermocouple and calculate the time difference between each consecutive measurement. 4. Print the largest time difference to the console and record it. 5. Make sure this value is less than 0.6 seconds.
1. Camera must be able to detect when a component is covering less than 70% of its corresponding pad	1. Connect the appropriate pins from the camera to the microcontroller. 2. Place a model or real PCB with a component ≤70% covering a pad. 3. Power the camera with 5V and the microcontroller with 3.3V. 4. Take a picture of the board and analyze the data on the microcontroller. 5. Verify that the microcontroller can see that the component is covering less than 70% of its pad. 6. Repeat with a component covering more than 70% of its pad and verify that the microcontroller sees that the component is covering more than 70% of the pad.
1. Camera must be able to send a frame once every 0.5 ± 0.1 second	1. Connect the appropriate pins from the camera to the microcontroller. 2. Power the camera with 5V and the microcontroller with 3.3V. 3. Take ten consecutive pictures from the camera and calculate the time between pictures. 4. Print to console the largest of these time gaps. 5. Verify that this time gap is less than 0.6 seconds.
1. Microcontroller digital output pins must correctly output a 1 or 0 for a corresponding voltage of ≥ 3.0V and ≤ 0.2V respectively.	1. Power on the microcontroller using 3.3V. 2. Set the output of all the pins to be 0.

	3. Probe each pin with a voltmeter and verify that the voltage is less than 0.2V. 4. Repeat steps 2 and 3 by setting all the pins to 1 and verifying the voltage on each pin is greater than 3.0V.
1. Microcontroller digital input pins must correctly read a 1 or 0 for a corresponding voltage of ≥ 3.0V and ≤ 0.2V respectively	1. Power on the microcontroller using 3.3V. 2. Attach to all pins a voltage of 0.2V. 3. Set all digital pins to be inputs, print the values to the console, and verify that all values are 0. 4. Repeat steps 2 and 3 by attaching a voltage of 3.0V across all the pins and verifying that all pins read a 1.
1. Microcontroller can produce a variable PWM signal at a frequency of 100kHz	1. Power on the microcontroller using 3.3V. 2. Attach an oscilloscope probe to the output of the PWM in use. 3. Set the duty cycle to 50%. 4. Verify that the frequency and duty cycle are correct. 5. Repeat steps 3 and 4 for varying duty cycles.

Table 4: Requirement Verification Table for HVAC

Requirements	Verification
1. When the user begins the reflow process, the coils should begin to heat the chamber and follow the specified profile that was chosen within ±10°C	1. Verify that the correct PWM signal is coming from the microcontroller by attaching one wire from the oscilloscope to the output pin of the microcontroller and then the other wire to ground. 2. Verify that the input to the boost circuit is a 3.3V PWM signal with the specified duty cycle by measuring the input pin of the MOSFET against ground. 3. Check the output voltage from the boost circuit and ensure that the value is a PWM signal with a minimum voltage of 4VDC or a maximum voltage of 40VDC by placing the leads of the oscillator on the output of the diode and ground. 4. Next, measure the output value of the solid-state relay by placing the positive lead of a multimeter on the output of the solid-state relay and then the ground lead of the multimeter to a ground or metal surface. The voltage should be within the range of 24V-240V AC. 5. Finally, measure the temperature that is relayed by the thermocouple in the toaster oven and compare it with what the profile

	states it should be at. This value should be within ± 10°C.
1. When the reflow process reaches ± 10°C peak temperature the fan will turn on at 7600 ± 10% RPM.	1. When the thermocouple returns a temperature that is ± 10°C of the peak temperature of the profile, the microcontroller will send a PWM signal to the level shifter. Verify that the correct PWM signal is coming from the microcontroller by attaching one wire from the oscilloscope to the output pin of the microcontroller and then the other wire to ground. 2. Check that the voltage coming from the level shifter is 5V ± 15% by placing the positive lead of a multimeter on the output of the lever shifter and the ground lead on a ground. This is because the fan has a minimum voltage of 4VDC and maximum of 5.75VDC. 3. Next, verify that the fan is spinning within the expected range (7600 RPM ± 10%) by placing one lead of an oscilloscope on the Tachometer or 3rd pin of the fan and then getting the frequency and dividing by 3600 to get the RPM.
1. 10 – 30 seconds after reaching the peak temperature ± 5°C the fan will run at 8300 ± 10% RPM.	1. When the thermocouple returns a temperature that is ± 5°C of the peak temperature of the profile, the microcontroller will send a PWM signal to the level shifter. Verify that the correct PWM signal is coming from the microcontroller by attaching one wire from the oscilloscope to the output pin of the microcontroller and then the other wire to ground. 2. Check that the voltage coming from the level shifter is 5V ± 5% by placing the positive lead of a multimeter on the output of the lever shifter and the ground lead on a ground. 3. Next, verify that the fan is spinning within the expected range (8300 ± 10% RPM) by placing one lead of an oscilloscope on the Tachometer or 3rd pin of the fan and then getting the frequency and dividing by 3600 to get the RPM.

Table 5: Requirements Verification Table for User Interface Subsystems

Requirements	Verification

1. Register 95% ± 5% of user inputs	1. Build a simple 'touch screen test' application that just serves as a drawable blank canvas. 2. Use stylus to draw on the touchscreen and visually ensure that all inputs are getting registered
1. Verify there are minimal 'dead zones' in the touch screen module to affect user inputs	1. Utilize the same 'touch screen test' application. 2. Draw over the entire screen to make sure no spots cannot take in user inputs
1. Have the ability to Create, Read, and Update preset soldering profiles with a maximum of three profiles	1. Store some soldering profiles on the SD card. 2. Ensure that the display can read these preexisting profiles. 3. Edit an existing profile and ensure that the changes are permanent. 4. Create new soldering profile and ensure that the profile gets stored and remains in the UI.

Table 6: Requirement and Verification Table for the Power Supply

Requirements	**Verification**
1. 5V regulator circuit outputs 5V ± 10% for a 120V AC power supply.	1. Attach a 100Ω resistor across the output of the 5V regulator circuit. 2. Connect the 5V regulator circuit to the 120V AC power supply. 3. Measure the voltage magnitude across the output with an oscilloscope and verify it is within the range of 4.5V to 5.5V.
1. 3.3V regulator outputs 3.3V ± 0.3V given a 5V ±10% input.	1. Attach a 100Ω resistor across the output of the 3.3V regulator. 2. Apply a 4.5V DC voltage source to the input. 3. Measure the output voltage with an oscilloscope and verify that the voltage is between 3.0V and 3.6V. 4. Repeat steps 2 and 3 applying a 5.5V DC voltage source to the input.

Table 7: Requirement and Verification Table for the Component Slip Detection

Requirements	**Verification**
1. Camera component can successfully take a picture and send data back to the microcontroller.	1. Connect camera to microcontroller 2. Take a picture 3. Ensure the picture has been sent to the microcontroller and can be viewed

2. Slip Detection algorithm successfully measures distances given a known distance	1. Input a given known distance in the frame 2. Have the algorithm return a measure of another known component (not inputted into the algorithm) 3. Verify that the measurement from the algorithm matches with the true measurement
3. Slip Detection successfully halts the process when in a ‘slipped’ state	1. Manually put the system into a failed state 2. Ensure that the algorithm detects that this is a failed state and halts the process.

Appendix B Parts and Schedule

Table 8: Bill of Materials (BOM) for Project

Description	Manufacturer	Part #	Qty	Price/Unit	Subtotal	Link
PYPORTAL - CIRCUITPYTHON POWERED	Adafruit Industries LLC	1528-4116-ND	1	54.95	54.95	Link
SSR RELAY SPST-NO 10A 24V-240V	AUTONICS	SR1-1210	1	17.00	17.00	Link
IC REG LINEAR 5V 1A TO220-3	Texas Instruments	LM7805CT/NOPB	1	1.91	1.91	Link
Toaster Oven	Bella	SO-312093_14413_BELLA-4-slice-toaster-oven	1	44.99	44.99	Link
Temperature Sensor Development Tools Thermocouple Type-K Glass Braid Insulated Stainless Steel Tip	Adafruit	3245	1	9.95	9.95	Link
IC CONV THERMOCOUPLE-DGTL SOIC	Analog Devices Inc./Maxim Integrated	MAX31855KASA+T	1	8.31	8.31	Link
IC REG LINEAR 3.3V 100MA SOT23-5	Texas Instruments	LP2981-33DBVR	1	0.56	0.56	Link
RES 10K OHM 1% 1/10W 0603	YAGEO	RC0603FR-1010KL	2	0.10	0.20	Link
CAP CER 0.1UF 50V X7R 0603	Samsung Electro-Mechanics	CL10B104KB8NNNL	2	0.10	0.20	Link
FERRITE BEAD 750 OHM 0603 1LN	Laird-Signal Integrity Products	HZ0603B751R-10	2	0.12	0.24	Link
TERM BLK 2POS SIDE ENTRY 5MM PCB	Würth Elektronik	691137710002	3	0.41	1.23	Link
Raspberry Pi Zero W	Raspberry Pi	3400	1	15.00	15.00	Link
Raspberry Pi Camera Board v2 - 8 Megapixels	Raspberry Pi	3099	1	29.95	29.95	Link

Table 9: Schedule for Assigned Work for the Project

Week	Bhaven Shah	Zak Kaminski	Raghav Narasimhan
2/27	Design Review Order Components Design PCB	Design Review Design PCB	Design Review Understand how to setup and use the microcontroller
3/6	Test modules Design Enclosure for PCB	Pass PCB Audit Test modules	Code interface with thermocouple, camera, and touchscreen Program PWMs

3/13	Assemble PCB Debug	Assemble PCB Debug	Program Touchscreen
3/20	Design Updated PCB	Design Updated PCB	Program Profiles and HVAC control
3/27	Debug	Pass PCB Audit Debug	Debug
4/3	Debug	Debug	Debug
4/10	Team Contract Assemble PCB Create Demo	Team Contract Assemble PCB Create Demo	Team Contract Create Demo
4/17	Mock Demo, Final Demo, Final Presentation, Final Paper	Mock Demo, Final Demo, Final Presentation, Final Paper	Mock Demo, Final Demo, Final Presentation, Final Paper
4/24	Final Demo, Final Presentation Slides, Final Paper	Final Demo, Final Presentation Slides, Final Paper	Final Demo, Final Presentation Slides, Final Paper
5/1	Final Presentation & Final Paper	Final Presentation & Final Paper	Final Presentation & Final Paper