Toast-E — Revision 1

Anthony Fabian[1], Forrest Herman[2], Patrick Kosierb[3], and Vasily Kapustin[4]

[1]400135464
[2]400179149
[3]400274083
[4]400185427

March 30, 2024

Table 1: Document Revisions

Version	**Date**	**Authors**	**Description**
0.0	2024-02-20	AF, FH, PK, VK	Initial formatting of the LaTeXdocument.
0.1	2024-03-24	AF, FH, PK, VK	Requirements section. V&V Section.
0.2	2024-03-25	AF, FH, PK, VK	Hazards section. Design Section.
0.3	2024-03-26	AF, FH, PK, VK	Future revisions section.
0.4	2024-03-27	AF, FH, PK, VK	Document review.
0.5	2024-03-28	AF, FH, PK, VK	Document review (part 2).
1.0	2024-03-30	AF, FH, PK, VK	Final revision.

Contents

1 Introduction **4**
1.1 Background . . . 4
1.2 Development Process . . . 5

2 Requirements **6**
2.1 Variables and Constants . . . 6
2.2 Behaviour Overview . . . 6
2.3 Required Behaviour . . . 7
2.4 Performance Requirements . . . 8
2.5 Normal Operation . . . 8
2.6 System Requirements . . . 9

3 Hazards **10**
3.1 Critical Assumptions . . . 10
3.2 Safety Hazards - Manufacturing . . . 10
3.2.1 Electric Shock . . . 10
3.2.2 Burn . . . 10
3.2.3 Cuts . . . 10
3.2.4 Toxicity . . . 11
3.3 Safety Hazards - Operation . . . 11
3.3.1 Electric Shock . . . 11
3.3.2 Burn . . . 11
3.4 Hazards Fault Tree Analysis . . . 11
3.4.1 Safety Hazards . . . 11
3.4.2 Project Hazards . . . 11
3.5 Risk Mitigation . . . 14

4 System Design **16**
4.1 Mathematical Definition of Crispiness . . . 16
4.2 Hardware Components . . . 16
4.2.1 Raspberry Pi 4 . . . 16
4.2.2 ESP32 Camera Module . . . 16
4.2.3 Power Cable . . . 17
4.2.4 10A fuse . . . 18
4.2.5 120VAC to 5VDC Adapter . . . 18
4.2.6 Wire Rails . . . 18
4.2.7 240VAC Relay . . . 18
4.2.8 5VDC Solenoid Electromagnet . . . 19

4.2.9 Push Button . . . 19
4.2.10 Slider Mechanism . . . 19
4.2.11 Tempered Glass . . . 19
4.2.12 Nichrome Wire . . . 20
4.2.13 Mica Sheet . . . 20
4.2.14 Galvanized Steel Sheet Metal Frame . . . 20
4.2.15 Camera Mount . . . 20
4.2.16 Plastic Base . . . 20
4.3 Software Components . . . 21
4.3.1 Control Module . . . 22
4.3.2 Training Module . . . 25
4.3.3 Crispiness Classifier . . . 25
4.3.4 Camera Module . . . 26
4.3.5 Bluetooth Application Module . . . 26

5 Validation **28**
5.1 Toast-E Performance . . . 28
5.2 Real-World Usability . . . 31
5.2.1 Current Prototype . . . 31
5.2.2 Final Product . . . 31
5.3 Final Toast Comparison . . . 31
5.4 Selection User Experience . . . 32

6 Verification **33**
6.1 Traceability Matrix . . . 33
6.2 Hardware Verification . . . 35
6.2.1 Mechanical Component Verification . . . 35
6.2.2 Electrical Component Verification . . . 36
6.3 Software Verification . . . 36
6.3.1 Control Software Verification . . . 37
6.3.2 Mobile App Software Verification . . . 38
6.3.3 Integrated Software Verification . . . 39

7 Summary of Testing Results **40**
7.1 Lessons Learned . . . 40

8 Reviews **42**
8.1 Code Reviews . . . 42
8.2 Hardware Selection Overview . . . 43

9 Future Revisions **45**
9.1 Precision Manufacturing . . . 45
9.2 I^2C Communication for ESP32 Cameras . . . 46
9.3 Liquid Crystal Display (LCD) . . . 46
9.4 Custom Printed Circuit Board (PCB) . . . 46

Chapter 1

Introduction

This document acts as an overview of all major design considerations of Toast-E Revision 1. Some aspects are still subject to change throughout the remaining development process. Aspects of this document such as an overview of requirements and hazards may be more independent of implementation details, but others such as system design or verification/validation are more implementation dependent.

1.1 Background

The current cooking process in traditional toasters is implemented using open-loop control. Most toasters use two methods for specifying cooking intervals. The first allows the user to select a specific time to toast using a dial or digital touchscreen which sets a cooking timer. The second method prompts the user with preset time limits for a particular type of toastable (eg. Frozen, Bagel, Waffle, etc.) or preset, labelled, crispiness levels (eg. "Light", "Medium", "Dark").
Problems arise with both methods.

Problems with Method 1
If the user does not toast often, they may not know how long to cook their toastable before it reaches their desired crispiness level. This will result in either an overcooked or undercooked toastable.

Problems with Method 2
Preset time limits are better than the first method concerning crispiness accuracy if the user does not toast often, however, the presets are generalizations and are at best "ballpark" estimates since the toastables between the manufacturer testing and end-user may vary to a significant degree. The variance is mainly due to unique characteristics of a particular toastable such as thickness, surface area, moisture content, and what ingredients were used to create the toastable.

Solution
Toast-E aims to resolve these issues, by making toast using a closed-loop control system, where the state of the bread is known to the control system. The reason closed-loop control solves the problems described above is that there is constant feedback to the internal controller as to how crispy the toastable is at that current moment in time. This is important because in real-time, Toast-E can compare the current crispiness level calculated using computer vision merged with machine learning methods to the desired crispiness level which is selected by the user from an entire spectrum of crispiness levels, not only a few presets. There is no need to find the perfect time to cook a toastable because Toast-E does it for the user.

1.2 Development Process

Table 1.1: Revision History

Date	Description
2023-11-20	Toast-E proof of concept demo. Demonstrated the ability to interface with a COTS toaster after a minor re-design.
2024-01-22	Toast-E Revision 0. Created a one-slot toaster frame and heating elements from scratch with COTS toaster and spring mechanism. Demonstrated closed-loop control, taking image data input.
2024-02-08	Toast-E Revision 0 update. Rewired the heating elements to increase the power output. Preliminary camera mount design
2024-03-18	Toast-E Revision 1. A main focus in ensuring the heat output is sufficient and uniform and working on computer vision software.
In Progress	Toast-E Revision 2. Improvements to heating and computer vision. Improved UI/UX. Higher quality manufacturing.

Table 1.2: Timeline

Date	Deliverable
2023-12-22	Submit System Design
2023-12-26	Purchasing more sensors and continued testing for sensor fusion
2024-01-06	Begin implementation of modules and control loop
2024-01-15	High confidence that the control system works
2024-01-22	Rev 0 presentation
2024-01-23	Begin incorporating Rev 0 feedback for Rev 1
2024-02-08	Rev 0 updated heating system and begin testing
2024-02-13	Rev 0 updated camera system and begin testing
2024-02-16	Validation and Verification deliverable
2024-03-01	Consistent, even cooking of toastables
2024-03-08	Higher overall repeatability of toasts
2024-03-17	Rev 1 complete
2024-03-30	Improved Toast-E software reliability and implement CNN
2024-04-09	Re-manufacture hardware and organize wiring
2024-04-20	Custom PCB fully integrated

Table 1.3: Relevant Terms

Term	Definition
Toastable	Food permitted to be toasted, such as bread or bagels.
Crispiness	A compound metric incorporating colour, texture, and moisture content. This estimates how cooked the toastable is.
Toast Time	The period taken to fully cook a toastable from an untoasted state.
Cycle	The complete sequence of operations performed after starting Toast-E throughout the toast time.
Carriage	The slot in which toastables are placed and held.

Chapter 2

Requirements

The following chapter will outline the Toast-E requirements, terms, and any relevant diagrams.

2.1 Variables and Constants

Table 2.1: Monitored Variables

Variable Name	Description	Units	Used For
M_SLIDER	Slider status	Category (e.g., enum) Pressed Not pressed	Determining if the user is ready to start toasting cycle
M_BUTTON	Abort button status	Category (e.g., enum) Pressed Not pressed	Determining if the abort button has been pressed (GPIO signal)
M_COLOUR	Colour of the toastable	RGB or other colour space (integer vector)	Measuring toastable's average surface colour for cripiness status
M_TIME	Holds time since cycle began	Time (s)	Determining if a cycle has gone past the max time a cycle can take
M_CRISPINESS	Users crispiness level	N/A	Determining setpoint for control system

Table 2.2: Controlled Variables

Variable Name	Description	Units	Used For
C_EJECT	Ejection status	Category (e.g., enum) Eject Wait	Eject toastable at the end of the cycle
C_HEAT_LEFT	Controls left heating element	Category (e.g., enum) On Off	Turning the left heating element on of off
C_HEAT_RIGHT	Controls right heating element	Category (e.g., enum) On Off	Turning the right heating element on of off
C_SHUTDOWN	Shut down Toast-E	Category (e.g., enum) Shut down Wait	Killing power if required

2.2 Behaviour Overview

When Toast-E is idle, the controller waits for the user to insert their toastable followed by putting the slider down to signal to the controller that the user is ready to select crispiness. Once the user selects their desired crispiness level and confirms it, the toasting cycle commences. The

Table 2.3: Constants

Variable Name	Metric	Units	Justification
CONST_MAXPOW	Max output power of heating element	Watts	Given the electrical power input, only so much heat can be generated.
CONST_SSTEMP	Steady state temperature of heating element	Degrees Celsius	Aside from the initial transient behaviour, the heater's temperature will be in steady state.
CONST_WIDTH	Minimum slot width	Millimetres	Needs sufficient space to fit all toastables.
CONST_MAXTEMP	Max temperature set point	Degrees Celsius	For user safety and to ensure components are protected.
CONST_MAXTIME	Max time till setpoint	Seconds	For user safety and to ensure the control has an exit condition if set point is passed.

measured variables will be updated regularly and used to compute the values of the controlled variables operating the control loop. Figure 2.1 gives an overview of the monitored and controlled variables interaction within the system.

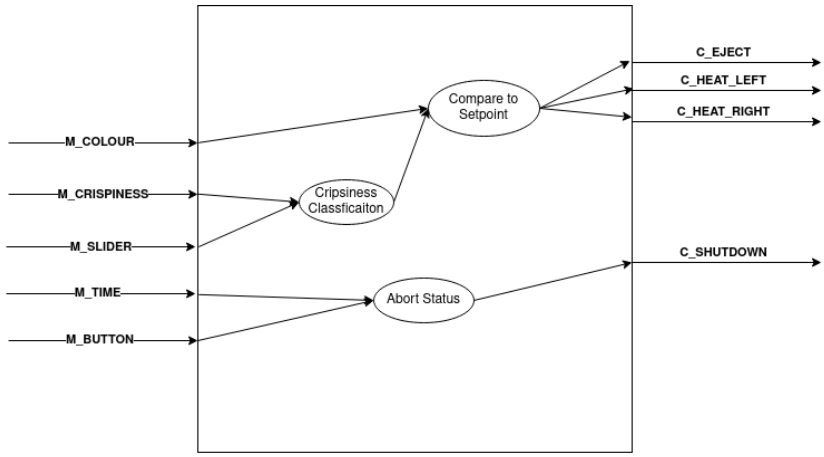


Figure 2.1: System Context Diagram

2.3 Required Behaviour

Toast-E will operate such that a toastable is cooked within one cycle within a reasonable time. The device will accept a range of toastables including sliced bread and bagels. Figure 2.2 shows a functional decomposition of Toast-E starting with the high level goal of delivering a toasted toastable.

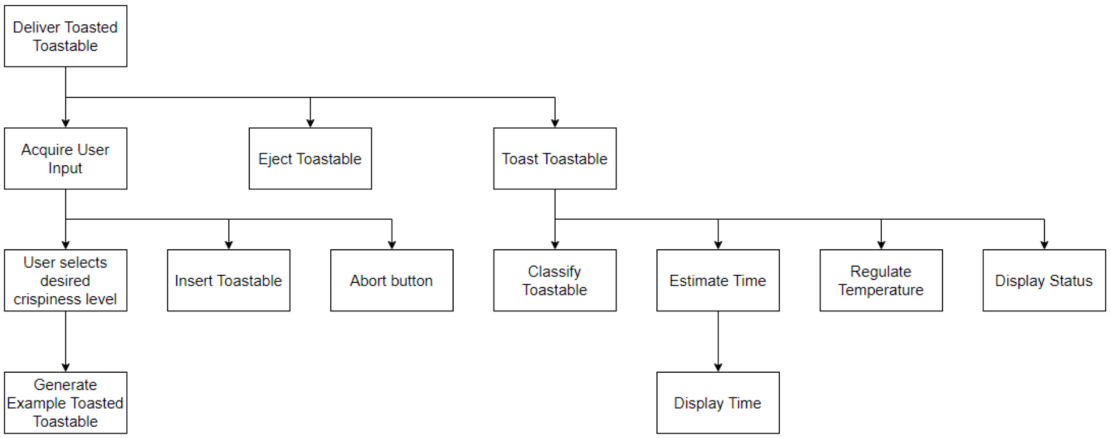


Figure 2.2: Functional Decomposition Diagram

2.4 Performance Requirements

Below is a list of performance requirements for Toast-E pertaining to deadlines and power draw.

1. **Max cycle time - 6 minutes**
   - Cycle has to complete within an amount of time expected by the user.
2. **Peak power draw - 1500W**
   - Toast-E should not trip an average 15A circuit breaker.
3. **Max avg power draw - 750W**
   - Toast-E should be relatively energy efficient compared to toasting alternatives.

2.5 Normal Operation

The normal operation of Toast-E is summarized in the following list.

1. The user inserts a toastable within Toast-E.
2. Slider is pulled to the down position.
3. The desired level of crispiness is selected.
4. The toastable is cooked until the desired level of crispiness is reached.
5. The toasting process halts.
6. Slider springs back to up position.
7. The user removes the toastable.

Table 2.4 lists possible undesired events that can occur during normal operation and how Toast-E accounts for the events.

Table 2.4: Event handling

ID	Event	Response	Rationale
EV-01	The toaster is unplugged during the toasting phase	System powers off and the state is reset.	No more input power.
EV-02	Toastable is overcooked	Toasting process immediately halts.	This is the primary feature of the product.
EV-03	Toastable is undercooked after full cycle	Do nothing. Wait for next user input.	If the cycle halts, there's no way of sensing the state of the toastable.
EV-04	User removes toastable before cycle has completed	Empty carriage so the toasting process ends.	If the user removes their toastable, there's nothing left to toast.

2.6 System Requirements

The high-level system requirements for Toast-E can be seen in Table 2.5.

Table 2.5: High-level System Requirements

Requirement ID	Description
REQ-01	Toast-E shall cook toastables to a specified crispiness.
REQ-02	Toasting to the desired amount shall occur in one cycle.
REQ-03	User shall be able to abort the toasting process at any time.
REQ-04	Toast-E shall eject toast when the user-specified crispiness level is reached, an abort is issued, or the toast time surpasses 6 minutes.
REQ-05	Toast-E shall be capable of measuring metrics associated with crispiness.
REQ-06	Toast-E shall be capable of cooking the toastable until it reaches the desired crispiness.
REQ-07	Toast-E shall offer users an input range for their desired crispiness.
REQ-08	Toast-E must have error statements for critical software components.
REQ-09	Toast-E must have a hard cut-off at a specified max temperature determined by component limitations.
REQ-10	Toast-E must warn users (with labelling) or mitigate any accessible high-temperature areas on the device.
REQ-11	Toast-E must inform users of possible electrical hazards on the device.
REQ-12	Toast-E must inform users that the toasting cycle has finished.
REQ-13	Toast-E must have a physical abort switch for users to operate in case of emergencies.
REQ-14	All Toast-E internal components must withstand cooking temperatures.

Chapter 3

Hazards

3.1 Critical Assumptions

fAs this project involves modifying a toaster, the hazards intrinsically linked with toasters are present in our design (e.g., electricity). Certain hazards can be exacerbated through particular user behaviours, for example, risks of electrical shock are greater if a user were to contact the heating elements with a metallic object. It will be assumed that the user will operate the toaster according to its intended purpose and follow all warning labels.

3.2 Safety Hazards - Manufacturing

3.2.1 Electric Shock

The risk of electrical shock is present when assembling the circuit required to power and operate Toast-E. Safety must be in mind when wiring up the heating element connections since the coils are powered directly from 120VAC at 5A which is fatal to humans. This hazard is the most prominent hazard we were faced with when creating the device. We took measures to mitigate any fatal accidents, including having proper non-conducive equipment when handling wires and terminal connections, breaker fuses in case of a short circuit, and multiple emergency power switches handled by multiple operators when there was power drawn.

3.2.2 Burn

The team was at risk of burns when performing heating element benchmark tests and regular toasting cycles. The heating elements naturally heat the metal frame making up Toast-E. This poses a risk when addressing any mechanical issues due to possible burns that can arise from manipulating the metal frame.

3.2.3 Cuts

The frame making up Toast-E is made out of galvanized steel sheet metal. This material is very thin and when cut with metal shears leaves prominent burs on the edges. Because of this, without proper safety equipment such as gloves, the team is subject to cuts when manufacturing.

3.2.4 Toxicity

The team must be aware of some materials used for manufacturing Toast-E that can be toxic to humans. A list of possible toxins to the team are listed below.

- Inhaling mica sheet dust
- Inhaling solder fumes
- Inhaling plastic fumes
- Poisoning from metal (e.g., tetanus)

3.3 Safety Hazards - Operation

3.3.1 Electric Shock

Users operating Toast-E are isolated from electrical shock with regards to the expected input sequence users shall take to toast their toastable. However, if the user does not follow the labels on the device there is a serious electrical hazard if an electrically conductive object is inserted into the toastable slot while a cycle is active.

3.3.2 Burn

Users operating Toast-E may be subject to burns if hazard labels are not followed. The nature of the device is to dissipate large amounts of energy from the nichrome coils, leaving the possibility for burns if the user touches parts of the toaster labelled hot or takes out their toastable without caution.

3.4 Hazards Fault Tree Analysis

This section will outline the safety and project FTAs conducted for Toast-E. All connections between levels represent "OR" gates.

3.4.1 Safety Hazards

Our definition for a safety hazard is any hazard that will directly cause harm to the user or surrounding environment. The device utilizes high voltage and high temperatures to achieve project goals which are traditionally seen as hazardous to people. High voltage creates the hazard of lethal electric shocks to the user if exposed to live heating elements or open circuits. Another hazard stems from the possibility of a short circuit causing a fire. A safety FTA can be seen in Figure 3.1.

3.4.2 Project Hazards

Our definition of a project hazard is any outcome that either impedes the production of Toast-E or results in project goals not being met. A project hazard FTA can be seen in Figure 3.2.

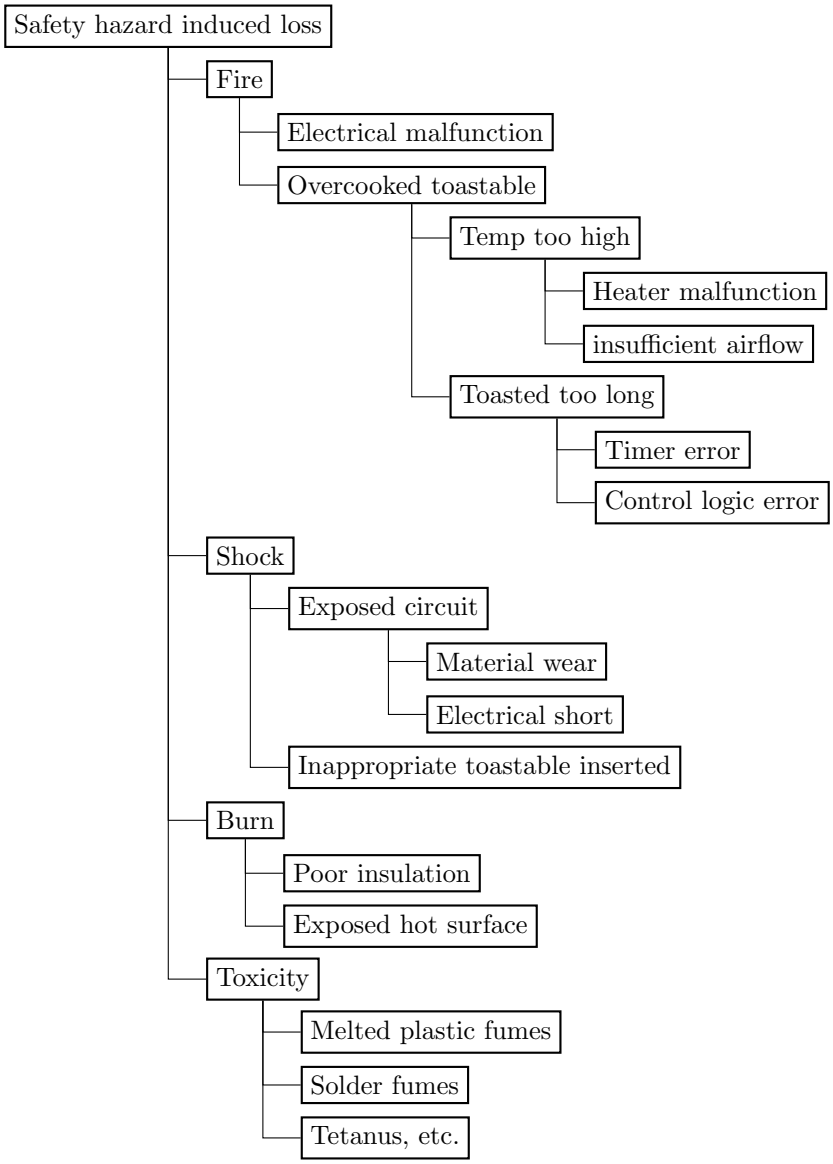


Figure 3.1: Safety Hazard FTA

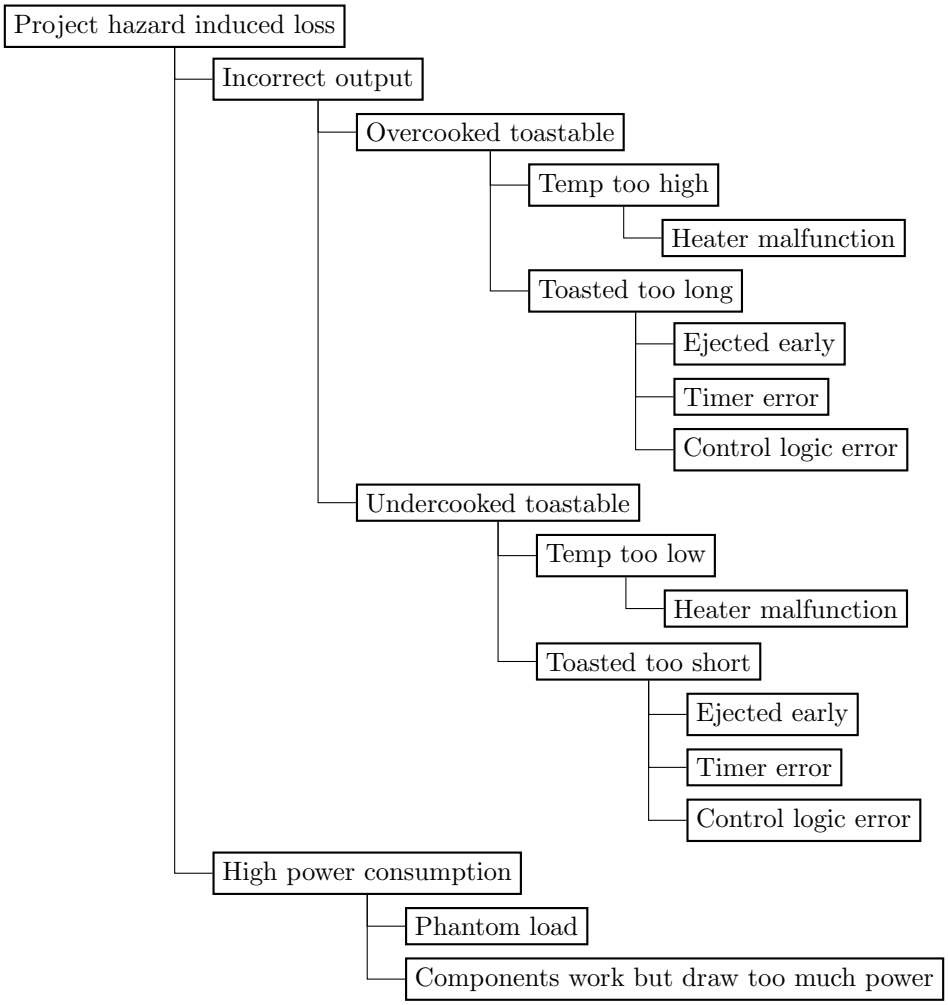


Figure 3.2: Project Hazard FTA

3.5 Risk Mitigation

This section outlines the risk assessment and mitigation strategies developed for Toast-E design, manufacturing, and usability. Table 3.1 shows the scale used for determining the likelihood of an event occurring. Table 3.2 shows the scale used for determining the impact of a particular event. Table 3.3 shows the safety hazards outlined in Figure 3.1 and their respective severity along with the response plan.

Table 3.1: Risk Probability Scale

Level	Definition	Likelihood
1	Very surprising to happen	<1%
2	Surprising to happen	1-3%
3	Neither likely nor unlikely	∼10%
4	Surprised not to happen	20-30%
5	Very surprising not to happen	>50%

Table 3.2: Risk Impact Scale

Level	Definition	Project Impact
1	Can be fixed within 2 hours.	∼10%
2	Can be fixed within 6 hours Project development is halted for 1 day	∼30%
3	Minimal re-development Project development is halted for 2 days Must re-develop a software module Lead time of 3 days.	∼50%
4	Toasting result isn't accurate. Project development is halted until solved Must re-develop many software modules Lead time more than 1 week User is harmed.	∼70%
5	Surrounding environment is damaged.	∼90%

Table 3.3: Saftey Risk Mitigation

Risk ID	Type	Description	Impact	Probability	Severity	Status Entry Date	Response Strategy	Response Plan
1	Safety	The user is burned.	5	2	High	10/25/23	Mitigate	Inform users of high temperature areas on the device
2	Safety	Device starts a fire.	5	1	Medium	10/25/23	Avoid	The device must have a hard cut off at a specified max temperature
3	Safety	The user is shocked.	5	1	Medium	10/25/23	Mitigate	Inform users of possible electrocal hazards on the device
4	Safety	Melted plastic.	5	1	Medium	10/26/23	Avoid	The device must have emergency stops for high temperatures
5	Safety	Developer is burned	3	3	Medium	3/25/24	Mitigate	Wear proper safety equipment when manufacturing
6	Safety	Developer is shocked	3	3	Medium	3/25/24	Mitigate	Wear proper safety equipment when manufacturing
7	Safety	Overtoasted bread	4	2	Medium	3/25/24	Avoid	Ensure robust control loop
8	Safety	Toasted too long	4	2	Medium	3/25/24	Avoid	Ensure max allocated time to toast
9	Safety	Temperature too high	3	2	Medium	3/25/24	Avoid	Ensure heating elements max temperatures are safe for the device
10	Safety	Logic error	2	3	Medium	3/25/24	Avoid	Hold code reviews
11	Safety	Timing Malfunction	2	1	Low	3/25/24	Avoid	Perform functional tests of toasting cycles
12	Safety	Calculation error	2	1	Low	3/25/24	Avoid	Perform functional tests of toasting cycles
13	Safety	Not enough air flow	2	1	Low	3/25/24	Avoid	Perform functional tests of toasting cycles
14	Safety	Heating element malfunction	5	1	Medium	3/25/24	Avoid	Perform inspection tests on heating elements to ensure no shorts or compromised wire
15	Safety	Electrical malfunction	5	1	Medium	3/25/24	Avoid	Perform inspection tests on circuitry
16	Safety	Poor insulation	2	1	Low	3/25/24	Avoid	Ensure cover on device is sufficient to block end user from high temperatures
17	Safety	Failure to indicate high temperatures	5	1	Medium	3/25/24	Mitigate	Add safety labels on high temperature zones on device
18	Safety	Toxic fumes	5	1	Medium	3/25/24	Avoid	Ensure all materials are rated for max heating element temperature
19	Safety	Solder joints melt	5	1	Medium	3/25/24	Mitigate	Ensure solder joints are out of range from high temperature zones

Chapter 4

System Design

4.1 Mathematical Definition of Crispiness

Crispiness is a quantity which estimates the true state of a toastable based on measurements taken throughout the toast cycle. Crispiness is calibrated such that it is a real number between 0 and 1, inclusive (quantized so a computer can handle it).

By definition, crispiness can only ever increase or remain the same (since toastables can't be uncooked within Toast-E). Due to measurement noise however, crispiness can momentarily decrease. The probability that crispiness decreases substantially due to this noise is negligible.

Ultimately, the goal of crispiness is to have a numerical way to compare the cooked status of different toastables. If two toastables have similar crispiness, they will have similar states in terms of being cooked.

1. $0 \leq \text{Crispiness} \leq 1$
2. $\text{Crispiness} = f(\text{state} + \text{noise})$
3. $\text{Crispiness}_A \approx \text{Crispiness}_B \Rightarrow \text{state}_A \approx \text{state}_B$

4.2 Hardware Components

4.2.1 Raspberry Pi 4

Responsible for performing computation as well as regulating timing and signalling the cycle to end. The Raspberry Pi 4 (RPi4) was selected as it can perform operations quickly while providing an easy-to-use environment for development and I/O options. Initialization of the RPi4 starts a script which is responsible for the Toast-E control, setting up required peripherals and start polling for user input. Being a real-time system, Toast-E must accomplish certain tasks at certain times. The RPi4 must be able to store and execute the code that performs these tasks at the appropriate times while being compact enough to fit within the Toast-E chassis. The RPi4 can be seen in Figure 4.1. Note that elements on the same row of Table 4.1 are not necessarily directly related and that multiple inputs may have effects on multiple outputs.

4.2.2 ESP32 Camera Module

The ESP32 camera module (ESP) acts as the primary sensing device for Toast-E, taking pictures periodically depending on a set sampling time. There are two located inside of the Toast-E chassis, each sends images via Wi-Fi to the RPi4 of both the left and right side of the toastable

Table 4.1: Raspberry Pi 4 I/O

Inputs	Outputs
Abort button (M_BUTTON/GPIO 24)	Toastable eject sequence (C_EJECT)
Toastable state measurements (M_COLOUR)	Left Heating element control (C_HEAT_LEFT)
User crispiness (M_CRISPINESS)	Right Heating element control (C_HEAT_RIGHT)
5V DC	Slider hold (GPIO 27)
Slider sense (M_SLIDER/GPIO 26)	Left relay (GPIO 22)
	Right relay (GPIO 23)

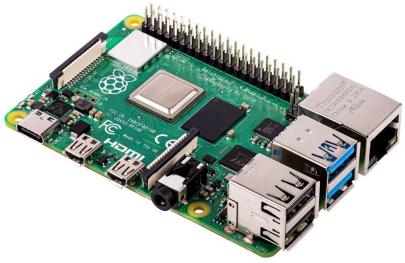


Figure 4.1: Raspberry Pi 4

for control. ESP were chosen due to the simplicity of implementation due to online support, small form factor, and low price. The ESPs also have the added advantage of having an on-board flash LED. This helps Toast-E be more robust as the flash allows sampled images to be roughly the same no matter the external lighting conditions. On top of that, the ESPs have a lot of configurations such as white balance, brightness, saturation, and contrast. Choosing ESP also opens up the possibility of using different MCUs with a smaller form factor than the RPi4 such as ESP development boards. On initialization, the ESPs connect to Wi-Fi and wait for a signal from the RPi4 to get a sample (take a picture) and send it. The ESPs must be able to handle temperatures generated by the heating elements around 5cm from the elements, along with fitting inside of the Toast-E chassis. The ESP can be seen in Figure 4.2.

Table 4.2: ESP32 Camera Module I/O

Inputs	Outputs
Sample picture signal	Byte array compressed JPEG
5V DC	

4.2.3 Power Cable

A single cable connects Toast-E to the 120VAC mains power grid via a standard 3 prong electrical outlet. This provides power for all components and allows for a grounded connection. There is a physical power switch (O = Off, I = On) connected to the cable which acts as an additional safety measure.

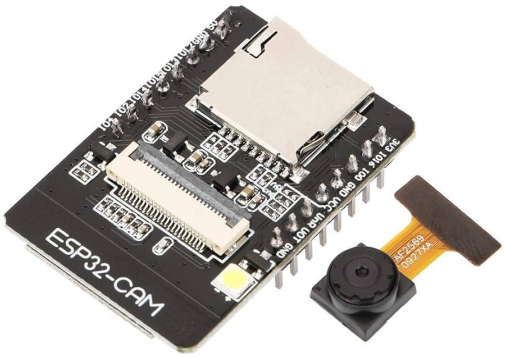


Figure 4.2: ESP32 Camera Module

4.2.4 10A fuse

This is positioned in series with one of the power wires connected to the prong of the 120VAC cable. This is easily replaceable if there were to ever be an issue leading to the fuse popping.

4.2.5 120VAC to 5VDC Adapter

Rectifies and adjusts the 120VAC input into a 5VDC output usable by all components, except the heating elements.

4.2.6 Wire Rails

The 5VDC power source and its associated ground are wired into two separate rails. This acts as a form of cable management and also streamlines electrical debugging/maintenance.

4.2.7 240VAC Relay

The relays are rated for 240VAC but will only ever be exposed to 120VAC. They're responsible for closing and opening the heating element and slider circuits. They are powered by the 5VDC rail. We chose relays due to the safety features they provide. If they're set to normally open, setting their control input to HIGH closes the mains power circuit (otherwise mains power is an open circuit). Also, using the relays proved very easy to interface with only requiring one GPIO signal from the RPi4 to trigger. Upon initialization, the slider hold, left heater, and right heater relays are set to LOW. As far as performance, the relays must be able to handle 120VAC for the heating element triggers, along with being constrained to a small form factor to be able to fit three total inside the Toast-E chassis. The relay module can be seen in Figure 4.3.

Table 4.3: 240AC Relay I/O

Inputs	Outputs
Left heating element (C_HEAT_LEFT)	Slider solenoid status (M_SLIDER)
Right heating element (C_HEAT_RIGHT)	
Slider solenoid relay (C_EJECT)	
5V DC	

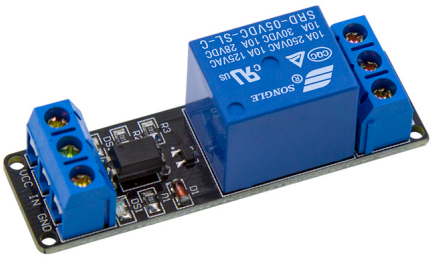


Figure 4.3: 204VAC Relay Module

4.2.8 5VDC Solenoid Electromagnet

5VDC Solenoid Electromagnet (solenoid) is responsible for holding the slider down when engaged. When the slider is pushed down, a button is clicked which sets C_EJECT to LOW and the slider is held down. When C_EJECT is set to HIGH, the solenoid releases the slider and its spring raises the slider.

4.2.9 Push Button

The push button works as an abort button for the user. This includes stopping the current toasting cycle which involves shutting off heating elements and resetting crispiness set-point. The RPi4 senses for a signal from the push button via the M_BUTTON variable. On startup, the button is pulled to HIGH to counter floating voltages. The button must be tactile for the user to sense an abort has been instantiated, along with fitting within the cutout on the Toast-E external shell.

4.2.10 Slider Mechanism

The slider mechanism is responsible for controlling the toastable position. The slider mechanism also acts as a user input to indicate to the RPi4 that the user is ready to input the desired crispiness. Once a toasting cycle is complete, the slider mechanism goes up to bring the toastable back to the start position, which allows safe and easy access for toastable insertion or retrieval.

4.2.11 Tempered Glass

The tempered glass is required for the side panels of Toast-E to provide a clear view of the toastable for the cameras to sample high-quality pictures for the control algorithm, while maintaining a thermal shield and insulator for the heating element.

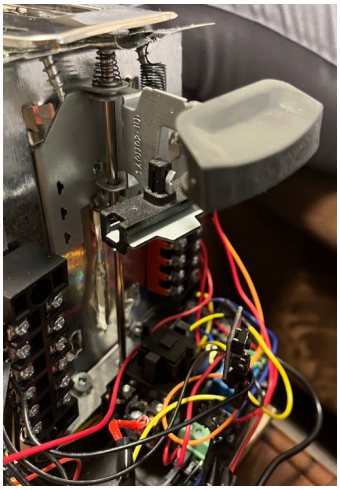


Figure 4.4: Slider mechanism

4.2.12 Nichrome Wire

Arguably the most important part of Toast-E is the nichrome wire. An array of nichrome wires on both sides of the toaster act as the heating elements, essentially acting as conductive (yet highly resitive), heat-dissipating elements with a high enough melting point to stay solid for the duration of the toasting cycle (and many successive cycles).

4.2.13 Mica Sheet

These sheets begin as 5”×5” squares which are cut to specific sizes and placed in order to insulate certain regions. Thin mica strips are also used to separate the nichrome wire and maintain spacing as there is a tendency for these wires to deform when Toast-E runs. Although exposed wires touching doesn’t affect much in terms of the electrical circuit, it does mean there is a large amount of heat generated in a concentrated area so this should be avoided. This usage of mica as a spacer is shown in Figure 4.5.

4.2.14 Galvanized Steel Sheet Metal Frame

The main structure of Toast-E comes from thin sheets of galvanized steel cut and bent into shape. These pieces are either screwed or riveted together depending on if they need to be disassembled later. Figure 4.6 shows a CAD model for the frame for clarity.

4.2.15 Camera Mount

This was 3D printed with ABS filament. These mounts hold the ESP camera a sufficient distance from the glass such that it can focus correctly and does not overheat. This is pictured in 4.7.

4.2.16 Plastic Base

This acts as an anchor for the frame to attach. Currently this base is being used from a COTS toaster but our team is actively designing a simpler one. Both versions can be seen in 4.8.

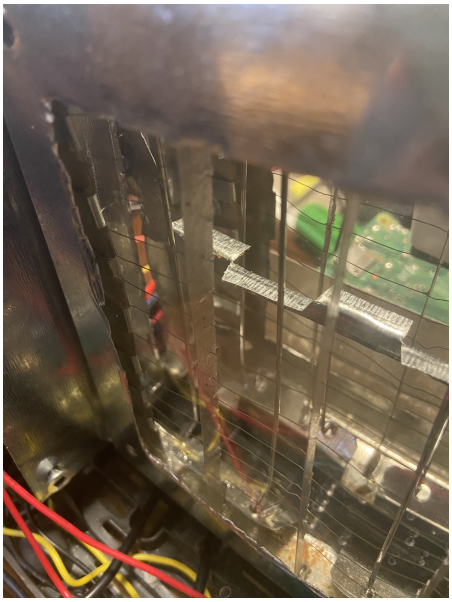
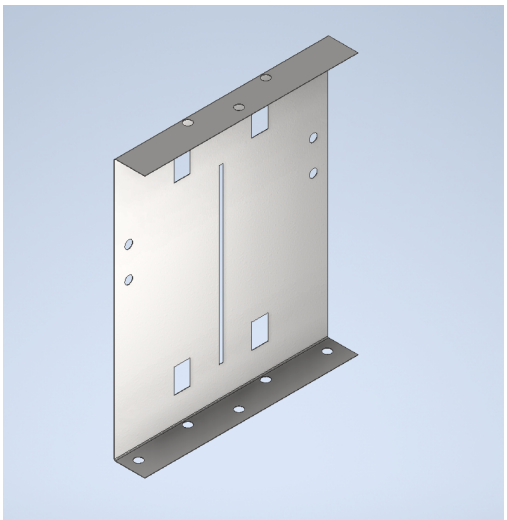
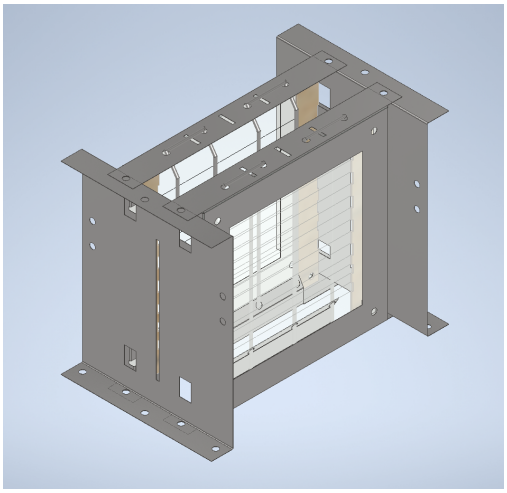


Figure 4.5: Vertical mica spacers holding nichrome wire in place



Front of frame



Complete frame

Figure 4.6: CAD render of metal frame

4.3 Software Components

The Toast-E software can be split into five key software components, the control module, Bluetooth module, camera module, training module, and crispiness classifier. The high-level software component diagram for Toast-E can be seen in Figure 4.9.

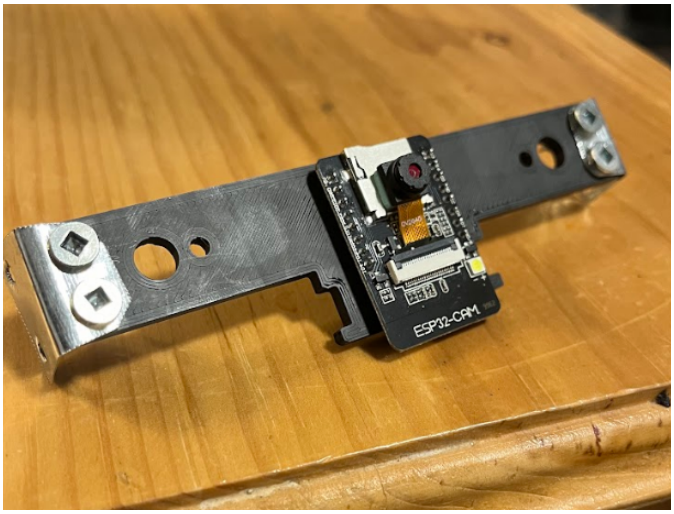
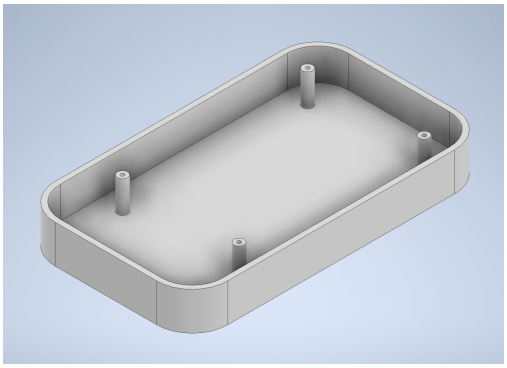
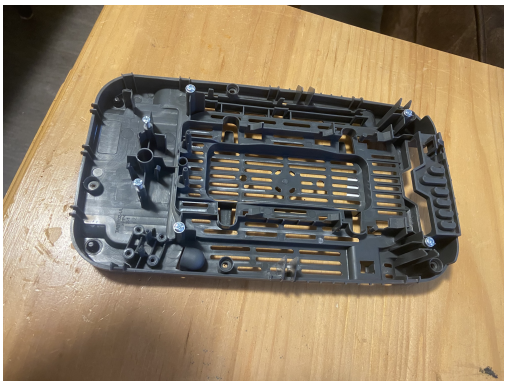


Figure 4.7: ESP32 camera mount



Render of new, simpler base



Existing COTS toaster base

Figure 4.8: Comparison of plastic bases

4.3.1 Control Module

The control module is a single Python script that runs on the RPi4. This script initializes all GPIO along with the BLE and Wi-Fi clients required. Figure 4.10 shows a control flow diagram of the main software loop.

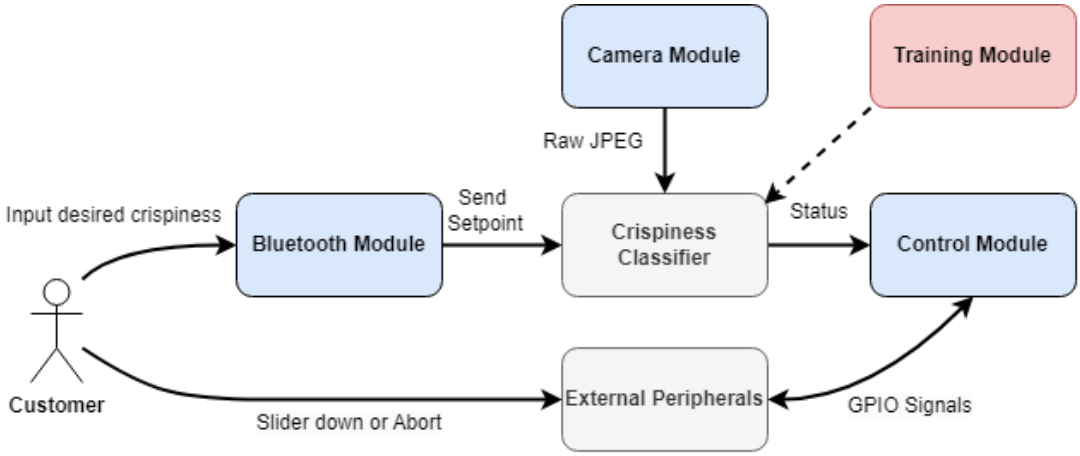


Figure 4.9: Software component diagram

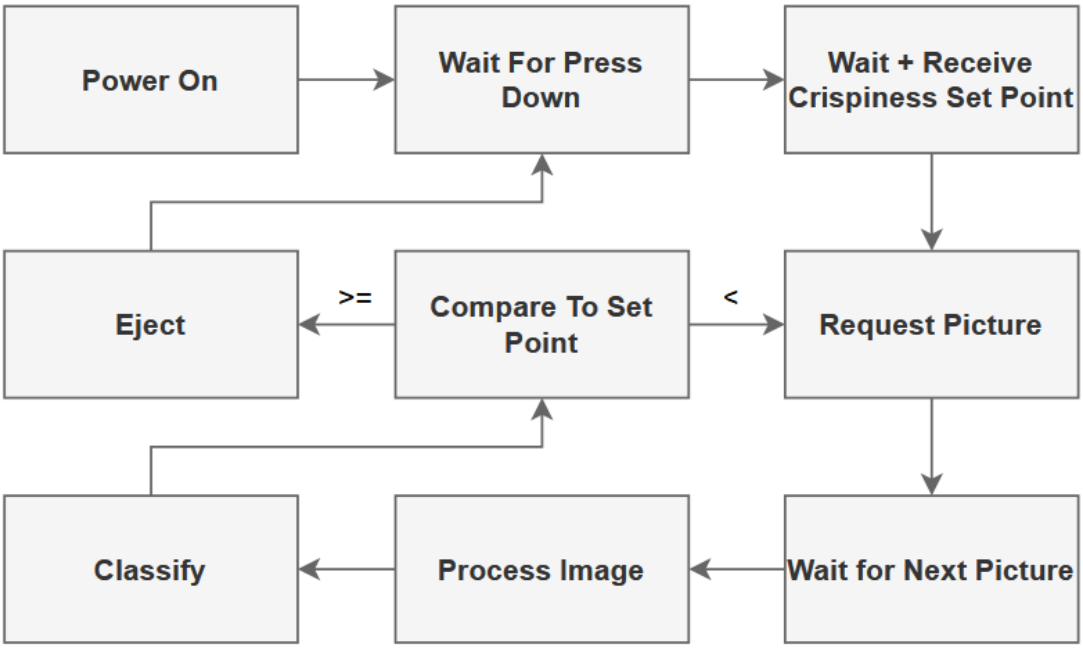


Figure 4.10: High-level Control flow

Pseudo-code for the control module initialization and main loop can be seen below.

```
Setup GPIO
Setup Wi-Fi
Setup Bluetooth

while(1):

    while(not M_SLIDER):
        wait

    // turn on solenoid to hold down slider
    C_EJECT = HIGH
    M_CRISPINESS = get_crispiness()

    setpoint = crispiness_to_colour(M_CRISPINESS)

    get start time

    while( not (left_done and right_done) and dt<CONST_MAX_TIME
       and not M_BUTTON):

        dt = current time - start time

        // sample time for capture
        if (dt % T_SAMPLE):
            tell the ESP cameras to take a picture

        if (received pictures from ESP cameras):
            data_left = left camera data
            data_right = right camera data
            left_done = compare(data_left, setpoint)
            right_done = compare(data_right, setpoint)
    eject()
    time.sleep(2)

def eject():
    C_HEATER_LEFT = LOW
    C_HEATER_RIGHT = LOW
    C_EJECT = LOW
    left_done = False
    right_done = False
    exit program
```

4.3.2 Training Module

This module consists of three main parts:

1. Capture images of toastables via the ESP32 cameras.
2. Processes these images in bulk.
3. Generate a relationship between the appearance of a toastable and its crispiness.

The first script is fairly straightforward, capturing images at regular intervals throughout the toasting process. The second one simply involves some cropping and median filtering (i.e., remove outlier pixels belonging to wires or mica) such that the images can be compressed without loss of important data. The training images are also labelled automatically with metadata such as timestamps.

This last script takes in a directory of processed images as input and outputs a function that classifies crispiness. Internally, there are many versions of this last script that use the same input and output formats as to increase the modularity and allow for rapid testing. Some versions use forms of polynomial (or more advanced) interpolation and others use learning-based methods.

Previously a 6th order polynomial regression on each colour channel (of the processed images) produced the best result. We are now using a learning-based method.

4.3.3 Crispiness Classifier

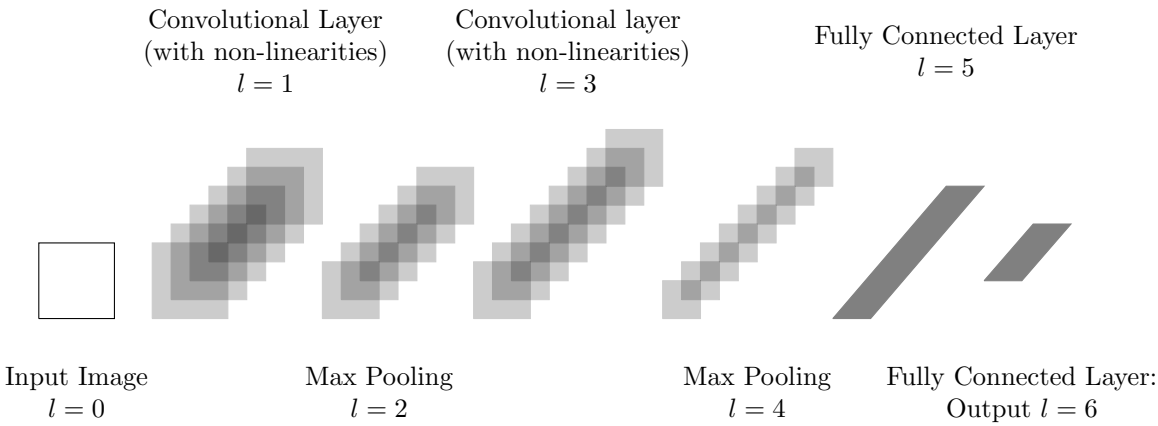


Figure 4.11: High-level structure of the CNN-based crispiness identification algorithm currently in development. This approach allows for robust feature extraction which accounts for varied lighting conditions.

Our team at Baker-E has recently implemented a CNN-based classification method that can outperform our previous interpolation-based methods to determine when a toastable is cooked correctly. Being a non-deterministic algorithm, we are aware that closed-loop control may be unstable, so there are additional control structures which may help improve reliability.

The objective moving forward would be to decrease the uncertainty via a variety of methods including more targeted training as well as re-tuning the structure of the network. The efficacy of our classification method continues to improve rapidly.

Ultimately, the function produced by the CNN and training data is known as the Crispiness Classifier. This function takes in a crispiness value in the range [0,1] (chosen by the user) and a stream of images (taken automatically by the ESP32 camera) which monitors the toastable. The output is a boolean signal which classifies if the toastable has reaches the desired level of crispiness.

4.3.4 Camera Module

The camera module is the firmware flashed on the ESP32 cameras. Upon startup, the camera is configured depending on the loaded configuration structure, adjusting various settings such as white balance, contrast, saturation, brightness, and more. Once configured the ESP connects to the same Wi-Fi that the RPi4 is connected to. Upon successful connection, the camera polls until a "take picture" request is received from the RPi4, once received the ESP camera proceeds to take a picture in compressed JPEG format (the quality of the image can also be adjusted). If a successful capture is made the camera module sends the picture as a byte array via MQTT protocol to the RPi4 for further processing.

It's important to note that pictures are taken with the heating coils off momentarily. This allows for a much cleaner looking picture with better lighting (since the coils dramatically over-expose the image). This momentary deactivation of the heating coils does not appear to impact toast time significantly.

4.3.5 Bluetooth Application Module

The bluetooth application module is responsible for all wireless control signals for Toast-E. This includes user-crispiness selection, remote stop command signals, and toasting complete notifications. Due to it's reliability and low power draw, the Bluetooth Application Module will be using Bluetooth Low Energy (BLE).

All BLE systems require at least two components:

1. A peripheral device that broadcasts a service with one or more characteristics.
2. A client that can scan for peripherals and initiate a pairing request.

BLE Peripheral

The Bluetooth Application Module is responsible primarily for running the BLE Server on the peripheral; for Toast-E, this is the Raspberry Pi microcontroller. The server is set to run in it's own thread, independent from the main control task (though both tasks do share data), and broadcasts Toast-E as a BLE peripheral ready for pairing. This peripheral offers a characteristic with a set of read, write, and notify request options, detailed below.

There are three types of native BLE commands in use: READ, WRITE, and NOTIFY. For WRITE requests, the first byte of the message indicates the command, and the second byte represents the data payload, where necessary. READ and NOTIFY requests don't require a payload, as they prompt the peripheral to return the relevant data for the current characteristic.

Table 4.4: BLE Peripheral Command Properties

Command Name	Identifier	Description	Properties
send current crispiness	0x01	Messages with this identifier require a second byte containing a numerical value. Upon receiving a message with this identifier, the peripheral will pass along the provided crispiness value to the main control loop.	WRITE
get current state	0x01	When the peripheral receives a message with this identifier, it responds with a payload object of the current state variables, structured as follows: • ENUM controller_state • INT current_crispiness • INT target_crispiness • INT time_remaining_estimate • INT time_elapsed This command also supports notification streaming, where a client can subscribe to receive state updates at 1 second intervals.	READ/NOTIFY
stop toasting	0x03	Upon receiving a message with this identifier, the peripheral will forward a stop interrupt to the main control loop. No additional data is required.	WRITE

BLE Client

The Toast-E mobile app, responsible for the majority of a user's interactions with Toast-E, offers a simple way to communicate with the Toast-E peripheral. The app features the ability to select a crispiness value, which is then transmitted via BLE message to the server and control loop. Once the toasting process has begun, the app automatically subscribes via a BLE NOTIFY command to receive regular updates on toasting progress. During toasting, the app also provides a 'cancel' button, which upon action immediately dispatches the *stop toasting* request detailed above.

Chapter 5

Validation

This chapter will outline the Toast-E Revision 1 validation process.

5.1 Toast-E Performance

It is important to validate the performance of Toast-E to ensure that it is up to par with the vision the team has put forth for a closed-loop design. One of the key relationships that can showcase the device's performance is the crispiness vs. time curve (CvT). The relationship is important because it can show the time intervals required to get to a particular crispiness level which can be compared to the time it takes for convention toasters to reach the same level, validating performance standards.

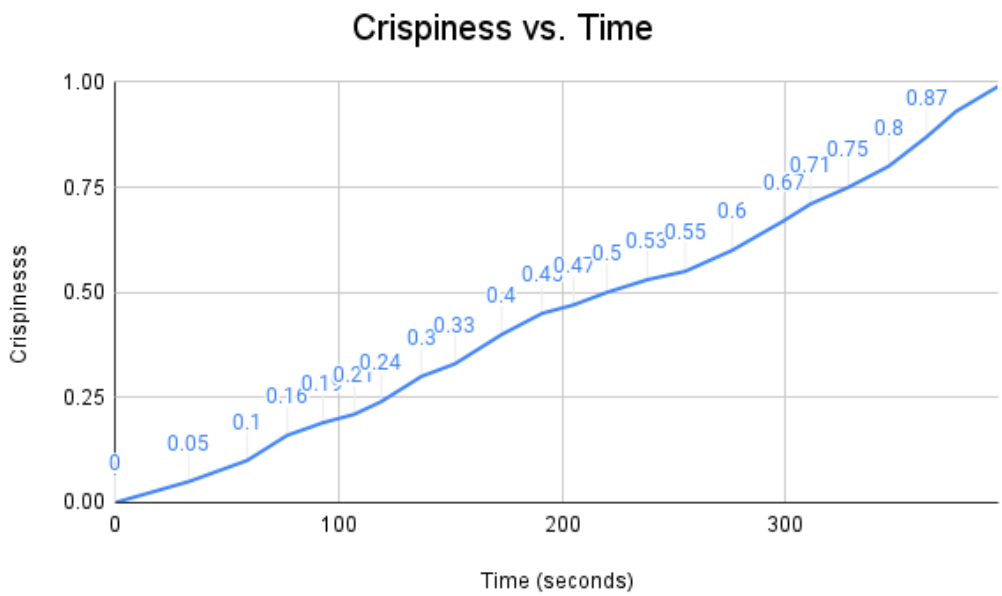


Figure 5.1: Crispiness vs. Time curve (CvT) for traditional white bread

Though the CvT curve is approximately linear (which might lead to the assumption that open-loop timers are sufficient for toasting), there are many factors that complicate the model. Firstly, this curve varies based on the toastable (type, temperature, thickness, etc.), the ambient room temperature, and the recent activity of the toaster. Furthermore, conventional toasters have

no knowledge of where they are positioned along the curve, meaning they cannot accurately determine how much farther they have until they reach the desired set-point. Additionally, there is a some degree of variance in the specifics of this curve depending on how the particular toastable was inserted because this may change the angle at which ambient light enters Toast-E. More information can be found regarding this problem in Section 9.1.

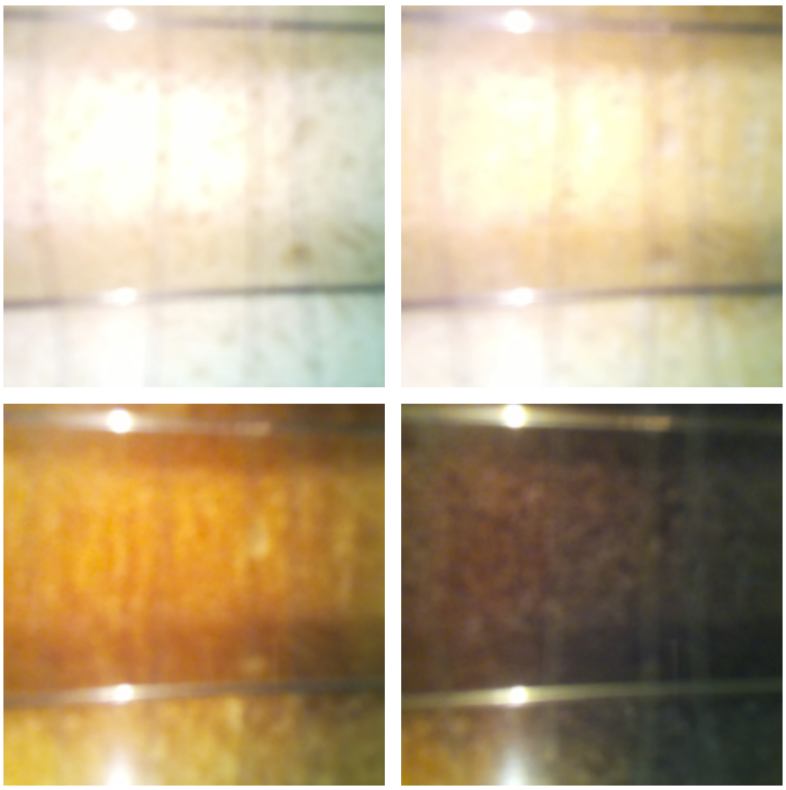


Figure 5.2: Appearance of toastable, sampled at equal time intervals (crispiness set to 1)

At various points in the toasting cycle, images were captured from the mounted cameras so the appearance of the toastable could be observed. Figure 5.2. shows 4 images captured at approximately equal time intervals. Note that the heating element was deactivated momentarily while the pictures were being captured to allow for easier interpretation.

These images represent an average white bread toastable. Differences between successive trials are suspected to be due to toaster being preheated, ambient room temperature, and temperature of the bread (room temperature or from fridge). Throughout the toast cycle, the colours of toastables typically change according to the curve shown in Figure 5.3.

Displayed in Figure 5.4 is an example of how the image captured would appear if the heating element stayed on. Due to the dominance of the heating coils, the colour of the bread is much harder to interpret. For this reason, the team decided to disable power to the heating coils momentarily while the cameras capture the current state of the bread. This ensures there is a clear change in appearance throughout the toast time.

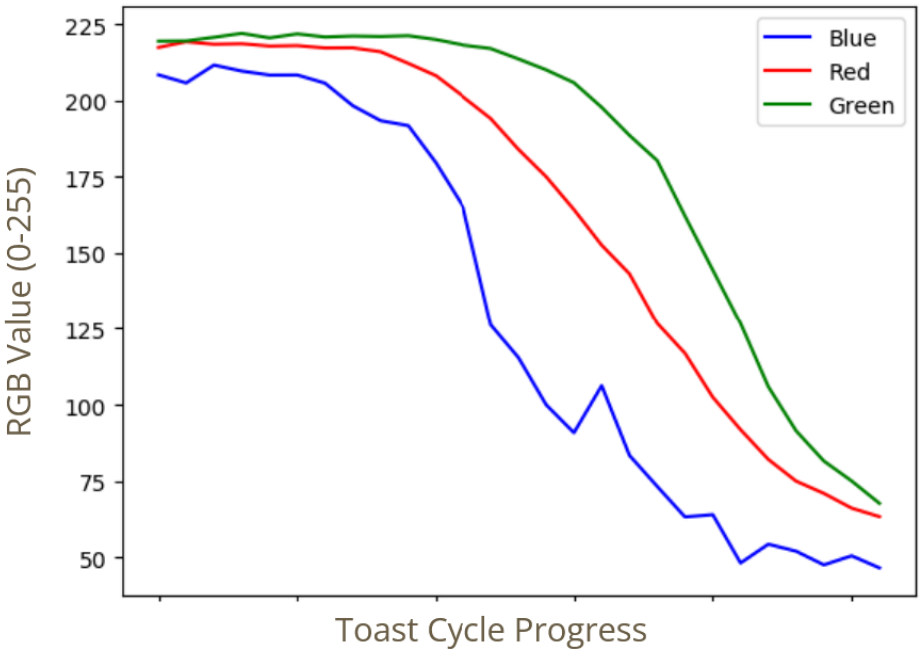


Figure 5.3: RGB vs Cycle Time

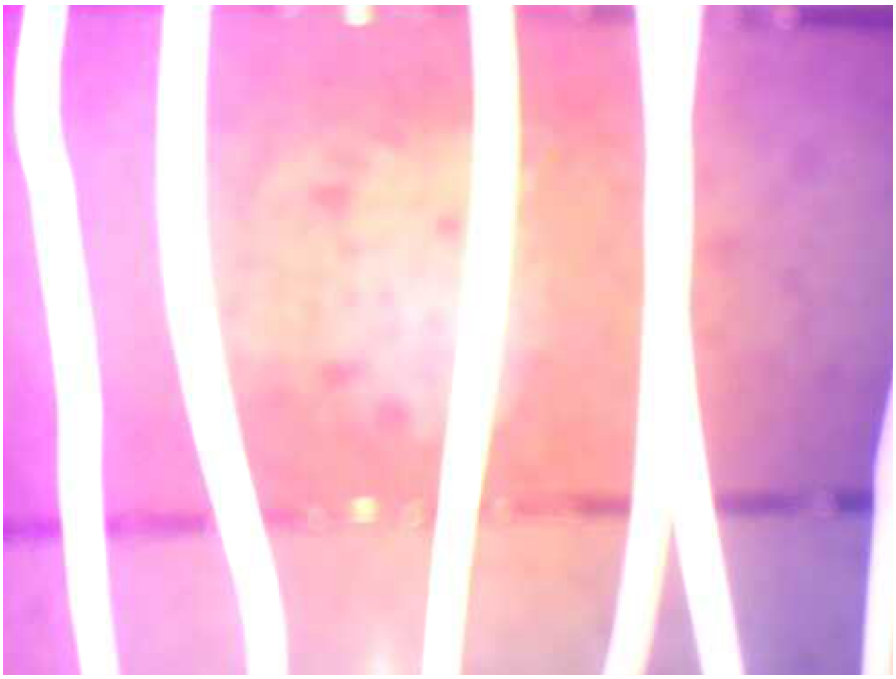


Figure 5.4: Appearance of the same toastable from Figure 5.2 with heater on

5.2 Real-World Usability

5.2.1 Current Prototype

It's important to acknowledge that Toast-E is currently a work in progress and several aspects of its current design and operation do not match "conventional" toasters. For example, there are many wires connected in ways that are easy to move around and detach/reattach. Although this drastically helps with the continuous and iterative design work being done, it can impact aspects such as user safety. We believe that this is a necessary compromise at this stage of development.

Additionally, our team is always extremely careful when transporting Toast-E as to not damage it, and although we cannot prove this, we believe the average appliance user is probably a bit more "relaxed". This of course is because commercial products (should) have extremely robust builds which can handle real-world operations. Again, because we are rapidly updating the design, having a robust build means it would be very awkward to disassemble multiple times per day.

5.2.2 Final Product

The vision for the final commercial Toast-E product would mimic current conventional toasters with a outer shell to both abstract away all internal complexity and to keep the user safe from electrical or heating hazards. All wire connections would also be minimized and a PCB would replace many of the more complex layouts to ensure the product is structurally sound. Beyond this, the glass panes included with Toast-E are isolated from impacts due to their being suspended with the sheet metal frame, and protected from scratches by the outer shell.

5.3 Final Toast Comparison

If the final output of a conventional toaster were to be compared to Toast-E with no additional information being considered, there would be very little overall difference until a high level of crispiness. Figure 5.5 shows a comparison between a traditional toaster and Toast-E for similar levels of crispiness.

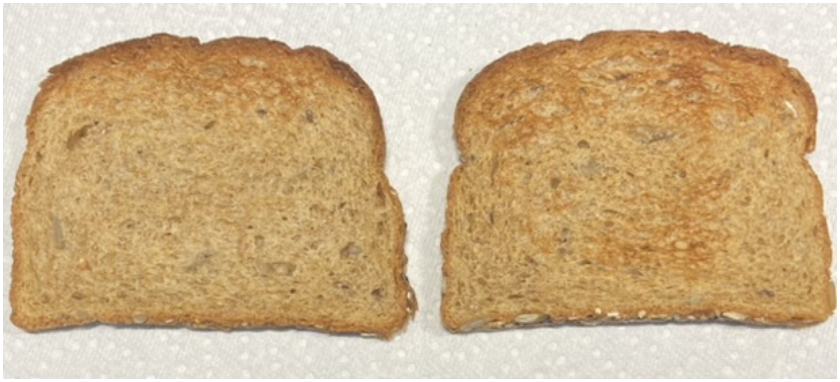


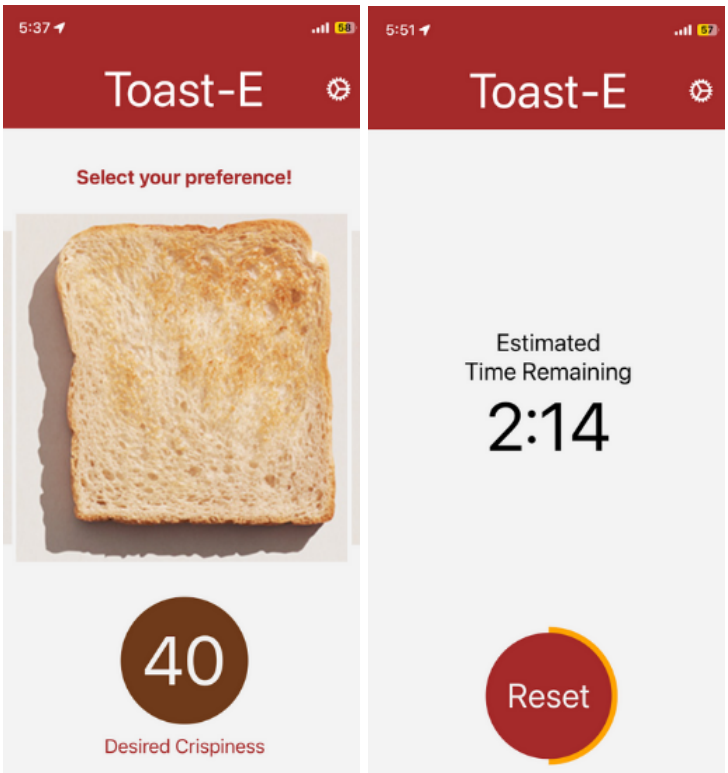
Figure 5.5: Crispiness gradient of Toast-E (left) and convention toaster (right)

The main discrepancy appears when comparing the colour gradient. Here are some potential explanations for this:

1. The nichrome wire should have a different wrapping density.
2. The geometry of Toast-E causes the airflow to behave differently.
3. The Toast-E frame is significantly worse at trapping hot air than a traditional toaster.
4. Glass allows more black body radiation through than metal layered with mica.

5.4 Selection User Experience

With the toastable's visual appearance being extremely important to the overall experience, it was important to provide a visual-based selection system for choosing the desired crispiness. Users should find a familiar and intuitive experience using Toast-E.



(a) Crispiness Selection

(b) Toasting Time Estimate

Figure 5.6: Toast-E's Accompanying App

Customers exposed to the app expressed positive feedback about the simplicity and heavy emphasis on user choice. They understand the functionality and what Toast-E offers compared to competitors. The lack of "preset" restrictions or an ambiguous dial gives the user full control over the result of their toastable of choice.

Chapter 6

Verification

The verification process for Toast-E will include individual component tests for both hardware and software components. First components will be isolated to ensure they work independently, and then integrated tests will ensure the product functions as required.

Throughout the process of developing and implementing tests, our team had the philosophy that metrics should not be arbitrary and empirical data is more important than theoretically determined quantities. For example, if components were stated to be rated for 5VDC and 100mA min but were found to require more power in practice, then simply meeting the 5VDC/100mA requirement would be insufficient.

6.1 Traceability Matrix

The traceability for Toast-E Rev.1 can be seen in Table 6.1. The tests outlined in Table are related to each of the pre-defined system requirements found in Table 6.2.

Table 6.1: Revision 1 Traceability Matrix

Req. Id. / Test Id.	REQ-01	REQ-02	REQ-03	REQ-04	REQ-05	REQ-06	REQ-07	REQ-08	REQ-09	REQ-10	REQ-11	REQ-12	REQ-13	REQ-14
T-01	X	X			X	X	X					X		X
T-02			X	X	X		X						X	
T-03	X	X			X									
T-04									X					
T-05										X	X			
T-06								X		X	X			
T-07														X
Count	2	2	1	1	3	1	2	1	1	1	1	1	1	2

Our traceability matrix successfully covers all of the high-level system requirements from Table 2.5.

Table 6.2: Test type definitions

Test type	Definition
Func	Functional test. Verifies component through testing functionality
Envir	Environmental test. Test performance of component under a set of environmental conditions
Inspec	Inspection test. Verifies a component through measurements, document reviews, or visual confirmation

Please note that when comparing two quantities which can take on a high number of values (e.g., analogue values, digital values with high precision), there is always a tolerance for error. For example, if we strive to ensure a tests yields a crispiness of 0.5, a level of tolerance may allow for

values in the interval [0.45, 0.55] to be acceptable. This may vary between tests depending on various circumstances.

Table 6.3: Test cases for High-level System Requirements

Test ID	**Test Type**	**Description**	**Inputs**	**Expected Outputs**	**Actual Outputs**	**Pass/Fail**
T-01	Func	Run an expected operating cycle.	Crispiness = 0.4, toastable = white bread, toastable temp = room temp	Toastable at desired crispiness level.	Toastable at crispiness 0.41	Pass
T-02	Func	Run an expected operating cycle but abort using UI or button sometime before cycle completion.	Crispiness = 0.4, toastable = white bread, toastable temp = room temp	The heating elements are powered off. The Toastable is ejected.	Heating elements power off and the toastable is ejected	Pass
T-03	Func	Collect time-varying crispiness variable data	Crispiness	A positive crispiness vs. time plot which increases and is within a range of target crispiness.	A positive crispiness vs. time plot	Pass
T-04	Func	Set a hard cut-off temperature to within the range of heating elements to ensure the correct abort sequence.	Crispiness	Once the internal temperature reaches hard cut-off the toastable is ejected and the heating elements are powered off.	No temperature reading	Fail
T-05	Inspec	Probe areas with warning labels to ensure the hazard is present	✗	Hazardous temperature is present (> 70°C)	Hazardous temperature is present	Pass
T-06	Inspec	Review control loop for error logs in appropriate locations.	✗	Error statements with printed status and program exits	Error statements are present	Pass
T-07	Envir	Run heating elements for 2*CONST_MAXTIME to ensure components withstand unexpected conditions	✗	Components withstand temperatures	Components withstanded temperatures	Pass

6.2 Hardware Verification

This section will detail verification done for critical hardware components that makeup Toast-E. These test suites also help verify high-level system requirements.

Note that requirements in brackets beside table names in this section refer to the those which are being tested.

6.2.1 Mechanical Component Verification

Table 6.4: Mechanical Tolerance (**REQ-14**)

Test ID	Test Type	Description	Inputs	Expected Outputs	Actual Outputs	Result
3.2.1.1	Func	Carriage can move easily up and down at least 5 cm	Movement with hand	Carriage moves down until touching electromagnet and up until toast is easy to retrieve safely	Carriage moves as expected	Pass
3.2.1.2	Envir	Toaster frame can withstand temperatures produced by heating elements	120VAC power	Toaster is structurally sound through a max toasting cycle (6 minutes)	Frame shows no signs of warping after do multiple cycles	Pass
3.2.1.3	Func	Tempered glass can withstand temperatures produced by heating elements.	120VAC power	Tempered glass is intact	Tempered glass is intact	Pass
3.2.1.4	Func	Electronic components (microcontrollers, cameras, relays, wires) are safe from melting	120VAC power	Air temperature outside the frame (containing the heating elements) below 70°C	Inspecting with manual thermal sensor shows valid temps	Pass

6.2.2 Electrical Component Verification

Table 6.5: Heating Coils (**REQ-06**)

Test ID	Test Type	Description	Inputs	Expected Outputs	Actual Outputs	Result
3.2.2.1	Func	Verify heating coils power draw power and output temperate	120VAC power	Heating coils light up and produce energy in the form of heat	Heating coils produce adequate heat to cook toastable	Pass
3.2.2.2	Inspec	No arcing between coil	120VAC power	Strands of the coil do not touch	Visual inspection shows that the coil strands don't touch each other	Pass
3.2.2.3	Inspec	Coils do not thermally expand into the case	120VAC power	Strands of the coil do not touch the frame	Visual inspection shows that the coil strands dont touch the case	Pass
3.2.2.4	Func	Nichrome wires do not overlap regardless of heating element state (wires induced EMF causes warping when powered)	120VAC power	Nichrome wires do not overlap or touch when current flowing	Nichrome wires are successfully isolated	Pass

Table 6.6: Electromagnet (**REQ-06**)

Test ID	Test Type	Description	Inputs	Expected Outputs	Actual Outputs	Result
3.2.2.5	Func	Verify that the electromagnet holds down the sliding mechanism	5VDC	Electromagnet is powered and holds the slider mechanism in place	Electromagnet holds the mechanism in until toastable reaches crispiness	Pass
3.2.2.6	Func	Verify the electromagnet releases slider when powered off	5VDC	Electromagnet releases slider mechanism	Electromagnet releases slider and toastable is ejected	Pass
3.2.2.7	Func	Verify that the rely controlling the electromagnet powers on	5VDC	Rely LED indicator is powered on	Rely LED indicator is powered on	Pass
3.2.2.8	Func	Verify that the relay controlling the electromagnet provides sufficient power	5VDC GPIO High	Audible switch is heard once signal to close the circuit is sent from GPIO and electromagnet is powered on	Audible switch is heard and a 5V voltage is read across the electromagnet	Pass

Table 6.7: Camera Hardware (**REQ-14**)

Test ID	Test Type	Description	Inputs	Expected Outputs	Actual Outputs	Result
3.2.2.9	Func	Cameras offer sufficient visual fidelity	Camera feed	Colour and shape of toastable can clearly be determined	Matches expectations	Pass
3.2.2.10	Envir	Cameras can withstand heat for entire toast cycle	Camera near heater	Cameras can withstand heat for >5 mins	Cameras can withstand heat for >7 mins	Pass
3.2.2.11	Envir	Cameras can withstand multiple cycles in a row	Camera near heater	Cameras can withstand 2 consecutive toast cycles	Cameras can withstand >5 consecutive toast cycles	Pass

6.3 Software Verification

This section will detail verification done for control and mobile app software components. These test suites also help verify high-level system requirements.

Note that requirements in brackets beside table names in this section refer to the those which are being tested.

6.3.1 Control Software Verification

Table 6.8: Normal ON-OFF Control and Configuration (**REQ-02, REQ-03, REQ-04, REQ-06**)

Test ID	Test Type	Description	Inputs	Expected Outputs	Actual Outputs	Extra Criteria	Result
3.3.1.1	Func	Cycle ends when toastable is correct colour	Toastable colour, target colour	Cycle ends when measured colour is near target	Matches expectations	Within difference of "tolerance" parameter	Pass
3.3.1.2	Func	Cycle ends when toastable is cooking for too long (emergency shut off)	Crispiness	Cycle automatically ends after 5 mins	Matches expectations	✗	Pass
3.3.1.3	Func	Cycle ends when "OFF" button is bressed	"OFF" button	Cycle ends	Matches expectations	✗	Pass
3.3.1.4	Func	Successfully configures peripherals (GPIO Pins)	5VDC	Print message upon success	Matches expectations	✗	Pass
3.3.1.5	Func	Writes to Left relay (heater)	5VDC GPIO High	Relay turns on	Matches expectations	✗	Pass
3.3.1.6	Func	Writes to Right relay (heater)	5VDC GPIO High	Relay turns on	Matches expectations	✗	Pass
3.3.1.7	Func	Writes to Solenoid relay (slider/eject)	5VDC GPIO High	Relay turns on	Matches expectations	✗	Pass

Table 6.9: Camera Software (**REQ-05**)

Test ID	Test Type	Description	Inputs	Expected Outputs	Actual Outputs	Result
3.3.1.8	Func	ESP32 Camera can communicate with the Raspberry pi 4	ESP32 Camera	Successful acknowledge byte transaction	Transaction	Pass
3.3.1.9	Func	Two ESP32 Cameras can communicate with the Raspberry pi 4 in parallel	x2 ESP32 Camera	Successful acknowledge byte transaction	Transactions successful	Pass
3.3.1.10	Func	Refresh rate is sufficient	2 Cameras	Changes in crispiness can be easily observed throughout toast cycle	Matches expectations	Pass

Table 6.10: Crispiness Identification Algorithm (**REQ-01, REQ-05**)

Test ID	Test Type	Description	Inputs	Expected Outputs	Actual Outputs	Extra Criteria	Result
3.3.1.11	Func	Average colour can be determined via cameras	Camera feed	Average colour measured matches RGB colour picker output reference	Matches expectations	Appearance in real life reasonably close to RGB value	Pass
3.3.1.12	Func	Colour to Crispiness	Training images	Toast-E can "interpolate" between colours to find corresponding crispiness value	Matches expectations	✗	Pass

6.3.2 Mobile App Software Verification

Table 6.11: Crispiness Selection Verification (**REQ-01, REQ-07**)

Test ID	Test Type	Description	Inputs	Expected Outputs	Actual Outputs	Extra Criteria	Pass/Fail
3.3.2.1	Func	Crispiness value is correctly pulled from current carousel image	✗	crispiness=0.3	crispiness=0.3	Output matches with currently displayed photo	Pass
3.3.2.2	Func	Crispiness value is correctly pulled from interpolation between two images	User slides carousel to the midpoint between image 3 and 4	crispiness=0.7	crispiness=0.7	Output matches fraction of crispiness to the nearest percent.	Pass
3.3.2.3	Func	Crispiness value is correctly limited to 0-1 scale.	User over-scrolls carousel past the last image	crispiness=1.0	crispiness=1.0	Output matches fraction of crispiness to the nearest percent.	Pass
3.3.2.4	Func	Selecting crispiness sends chosen value over BLE and sends a request for subscribing to future BLE messages	Target crispiness	BLE write, BLE notification request	Both BLE commands are sent.	No race conditions	Pass

Table 6.12: BLE Commands Verification

Test ID	Test Type	Description	Inputs	Expected Behaviour	Actual Behaviour	Extra Criteria	Result
3.3.2.5	Func	Current selected crispiness is sent over BLE	Crispiness value of 0.45 is selected	0.45 is encoded in a byte array where the first byte is command identifying this as a selected crispiness value, and the second byte is 0.45 and passed to the BLE library methods for writing	Matches expected behaviour	Values are accurately encoded: proper command is passed in correct order	Pass
3.3.2.6	Func	Toaster state data payload arrives at even time intervals once subscribed	Byte array containing state payload parameters (controllerState, currentCrispiness, timeRemaining, elapsedTime) is received	Decoded into Object for easy access	Successfully decodes object	Values are accurately decoded into Object with keys	Pass
3.3.2.7	Func	User remote cancel toasting from the app	Cancel command triggered by app user	Cancel command identifier is encoded as the first byte of the message.	BLE library write method receives byte array with first byte as cancel command identifier	Values are accurately encoded: proper command is passed in correct order	Pass

6.3.3 Integrated Software Verification

Table 6.13: Toast Controller Commands and Logic (**REQ-01, REQ-07**)

Test ID	Test Type	Description	Inputs	Expected Behaviour	Actual Behaviour	Extra Criteria	Result
3.3.3.1	Func	Insert toastable, choose desired crispiness in the visual mobile app, lower toaster carriage, and await completed cycle.	Crispiness from mobile app, Trained image dataset	The toaster triggers a shutoff command when the toastable reaches the desired crispiness.	Toaster eventually shuts off, but not always at the perfect crispiness	Activated within 0.05 of target crispiness	Pass
3.3.3.2	Func	During an ongoing cycle, cancel the cycle through the mobile app interface.	Cancel command from app	Toaster cancels the current cycle	Toaster is shut off	Action completed	Pass
3.3.3.3	Func	Open app no Toast-E available	Mobile App No Toast-E	App notifies the user that no Toast-E is present	App opens to splash screen and prompts user to connect to Toast-E	✗	Pass

Chapter 7

Summary of Testing Results

This chapter summarizes the results our team found from three core test suites, along with lessons learned from validation and verification that will be applied in the future.

Table 7.1: Summary

Testing Suite	Purpose	Summary
High-Level System Requirements Verification	To verify Toast-E requirements	• T-04 failed due to the lack of onboard temperature sensor therefore REQ-09 was not fully verified. It should be noted that all components appear to behave properly during all tests though. • All other test cases passed
Hardware Verification	To verify the functionality of Toast-E hardware	• Hardware functioned as expected, all tests passed
Software Verification	To verify the functionality of Toast-E software	• Some edge cases and more complex states need more polishing, such as race conditions for multiple BLE commands and repeated toasting cycles • All other test cases passed

7.1 Lessons Learned

Many of the challenges our team faced were small and nuanced. Examples include minor mechanical tolerance issues (i.e., play) and securing wires. We did not properly account for how much of the engineering process would need to be dedicated to these types of problems in relation to the "expected" problems such as computer vision accuracy. The process of cutting and bending metal safely and accurately has taken much more time than anticipated.

Additionally, a lot of the recent design/assembly work has been directed at adjusting the power level and thermal output. This has called for the design and construction of a new power supply. This was also so that we could power all the 5V components while toasting with sufficient power.

The testing has uncovered some of the most pressing design limitations and necessary improvements moving forwards. The requirements that are still not fully fulfilled are more clear and development moving forward can be more streamlined towards those areas. This includes race

conditions in the command module and Bluetooth messages, more specific error handling and identification, and priority of interrupts for the control loop, where safety critical interrupts should be prioritized above all else.

Past iterations of Toast-E have used Wi-Fi, bluetooth, and even solely GPIO for communication. There are obviously pros and cons to different methods and we are currently exploring I^2C as it appears to solve some minor problems we've encountered. More information is available in Section 9.2.

Chapter 8

Reviews

This chapter highlights the software and hardware reviews conducted by the team.

8.1 Code Reviews

Table 8.1 outlines the series of code reviews conducted by team members on critical software modules.

Table 8.1: Code Review

Module	Reviewer	Comments	Suggested Fixes	Date
CV training module	AF	There are some "quality of life" improvements that can be made to allow the training to run smoother.	Make the file names generated very simple and intuitive. Consider adding meta-information like the date.	02/10/24
Control Module	FH	Some of the if conditions, (the ones checking if the toast's average colour is within bounds) aren't readable (e.g., `if left_done == False and blue_range[0] < average1[0] < blue_range[1] and green_range[0] < average1[1] < green_range[1] and red_range[0] < average1[2] < red_range[1]`) Also, these types of conditions may not successfully terminate the control loop since there are multiple `and` operations	Rewrite the boolean logic into descriptive functions for better legibility. Ex: `def is_left_done() -> bool` Use a more robust control strategy.	02/10/24
Bluetooth Send/Receive	PK	Ensure that incoming bluetooth messages are more easily accessible from the main control task. Rather than check all variables each cycle, there are more efficient ways. Address the "TODO:" comments in `ble.py` line 42.	Use a message queue.	02/11/24
Control Module	VK	Update main loop to include blocking statements when waiting for a condition to be met.	if(*condition*) to while(not *condition*)	03/10/24
Camera Module	VK	Camera configuration missing.	*sensor_t * s = esp_camera_sensor_get();* will allow for camera config.	03/10/24

8.2 Hardware Selection Overview

Table 8.2 outlines each physical component used to design Toast-E revision 1 and their respective relevance.

Table 8.2: Toast-E Electrical Components

Hardware Component	Quantity	Description	Relevance
32 AWG Nichrome wire	2	A resistance wire made of nickel-chromium alloy used in heating elements	• A part of the heating elements • Works as the "heating coil" for the device due to it's high-temperature tolerance and mechanical strength
18 AWG copper wire	4	Copper wire used to pass current	• Used to connect GPIO pin-outs and 5V DC
30 AWG copper wire	2	Copper wire used to pass current	• Used to connect heating elements to 120V AC
5V one-channel relay	3	Electromechanical switch	• Used as switches for x2 120V AC inputs (heating elements) and x1 5V DC input (power to relays)
Raspberry Pi 4	1	Development Board	• The controller used for the device • Controls all relays • Works as the server for application
ESP-32 Camera	2	8MP camera	• The main sensor for control of the heating elements
120VAC-5VDC Adapter	1	Converts AC to DC	• Allows for the main 120VAC cable to power 5VDC components.
Power Cable	1	Connects Toast-E to an outlet	• Allows for a single cable to power all components • Has built-in On/Off switch

Table 8.3: Toast-E Mechanical Components

Hardware Component	**Quantity**	**Description**	**Relevance**
Galvanized Steel sheet metal side panel	2	Galvanized Steel panel which withstands high temperatures	• Holds glass, nichrome wire, and mica sheets in place • Encloses heat generated by heating elements • Structural support
Galvanized Steel sheet metal front panel	1	Galvanized Steel panel which withstands high temperatures	• Holds side panels • Encloses heat generated by heating elements • Holds slider mechanism
Galvanized Steel sheet metal back panel	1	Galvanized Steel which withstands high temperatures	• Holds side panels • Encloses heat generated by heating elements • structural support
Aluminum top panel	1	Aluminum panel which withstands high temperatures	• Encloses internal structure • Encloses heat generated by heating elements • Blocks electrical hazards from users • Slot for toastable
High-temperature tempered glass	2	Tempered glass which can withstand high temperatures	• Provides a "pass-through" for cameras to see the toastable and collect relevant data • encloses heat generated by heating elements • Structural support
Mica sheet	2	A sheet of mica, a silicate mineral	• A part of the heating elements • Works as a base for nichrome wire to wrap around due to its electrically non-conductive properties and high heat tolerance
Spring	2	Stores energy	• Used for damping slider mechanism when toastable is ejected
Plastic base	1	Grey plastic base with slots and screw holes for mounting components	• Provides a base for the internals of the device including the metal frame, heating elements and glass • Encloses heat generated by heating elements, Structural support
Plastic cover	1	Thin white plastic shell	• Hides internal structure from the end user • Encloses heat generated by heating elements • Ensures lighting is consistent inside of the device which helps mitigate noise associated with computer vision data

Chapter 9

Future Revisions

9.1 Precision Manufacturing

Although our team at Baker-E is very proud of the parts manufactured to assemble Toast-E, we strive for even better. We currently have an appointment to precision cut and bend steel at the machine shop, and we wish to create our final revision using these tools.

Figure 9.1 shows some previous designs, which have gotten the job done so far but have minor imperfections (e.g., burrs, warping, sub-optimal tolerance). Not only is it safer for the end user to handle a "nicely" manufactured frame, but it also is safer for our team since manufacturing steel parts is quite intensive to say the least. The initial stage of transforming a piece of sheet metal into the final design with a stencil is shown in Figure 9.2.

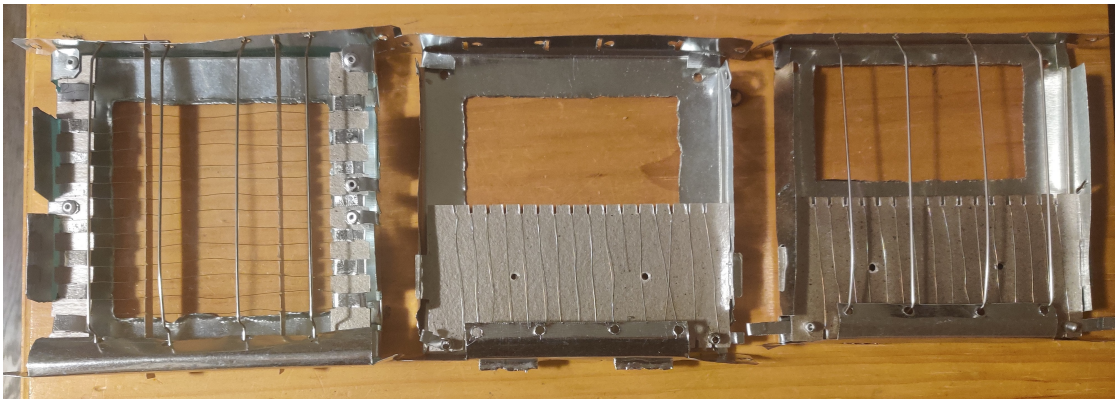


Figure 9.1: Hand-manufactured designs from previous revisions

With better tolerance on the manufacturing, our team hopes to ensure that toastables will be inserted within Toast-E with greater consistency in position and slant. This will allow for improved lighting conditions which will increase the consistency of the pictures taken and improve crispiness classification.

Figure 9.2: Stencil and bare sheet metal which will soon be processed.

9.2 I^{2}C Communication for ESP32 Cameras

Currently, the ESP32 cameras communicate to the RPi4 through Wi-Fi using MQTT protocol. While this approach has its benefits, such as reduced wiring compared to wired protocols, the downsides are prominent since the Toast-E requires a Wi-Fi connection for the control to work bottleneck the versatility of Toast-E and hinders convenience. The OTA nature of Wi-Fi communication also has a higher latency than a wired communication protocol. One prominent communication method used all over industry is I^{2}C protocol, This simple protocol is supported by both ESP32 Camera module and RPi4 allowing for a shared data line and a clock line through various GPIO pins. A packet containing a capture from an ESP32 can be over the SDA line and checked for corruption using an error-detecting code such as a cyclic redundancy check (CRC).

9.3 Liquid Crystal Display (LCD)

Currently, Toast-E users need to use the BLE app to input crispiness using an intuitive menu. However, users who don't have access to a smartphone need a way to input their desired crispiness. The team is currently designing an LCD touchscreen to serve as another input method.

9.4 Custom Printed Circuit Board (PCB)

The wiring shown in Figure 9.4 is less organized than what could be achieved using a custom PCB. The 120VAC wires are quite simple to arrange within the toaster but all of the 5VDC wires crowd the inside of Toast-E. This makes the other case much harder to fit on top and raises the likelihood of a wire becoming detached. Figure 9.4 shows the a PCB with an equivalent circuit to that of Figure 9.4. We hope to have this printed shortly.

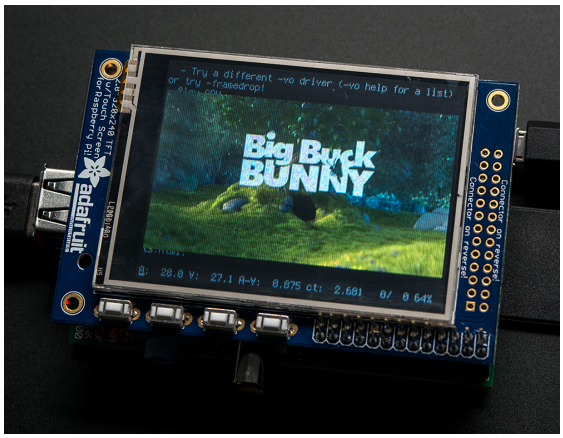


Figure 9.3: Liquid Crystal Display (LCD)

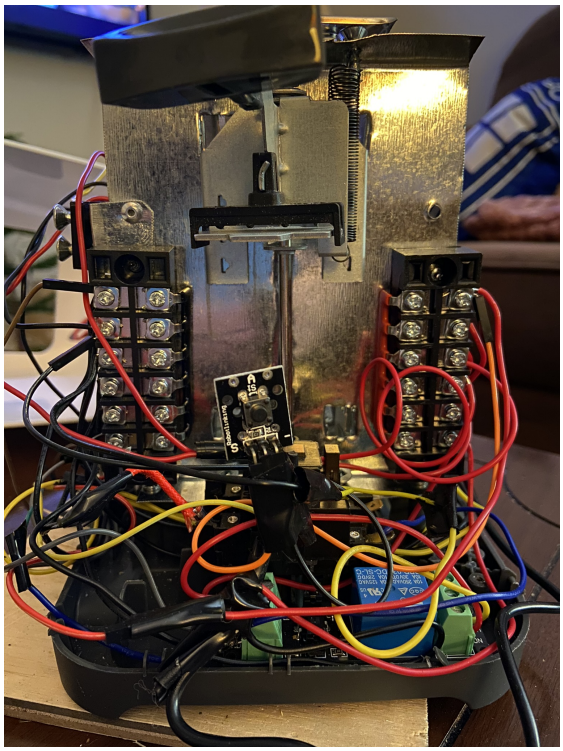


Figure 9.4: The ground (left) and 5VDC (right) rails among many wires.

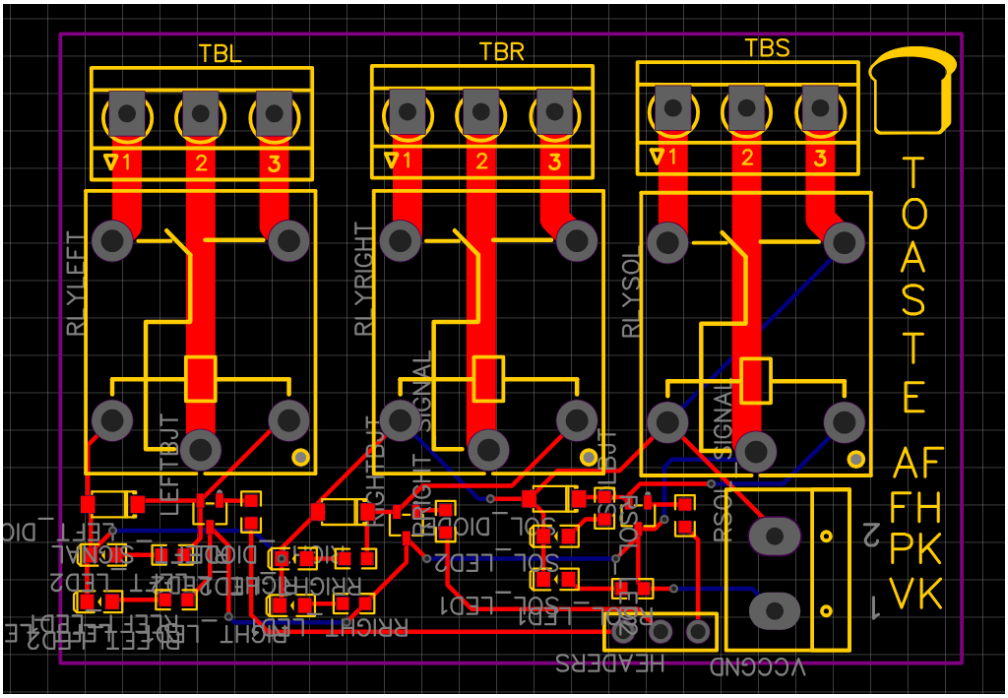


Figure 9.5: Custom PCB for 5V components