ECE 445 SP24

Design Document

Team 3: Dough Monitor

Team

Jake Hayes (jhayes@illinois.edu)

Abhitya Krishnaraj (abhitya2@illinois.edu)

Alec Thompson (alect3@illinois.edu)

TA

Tianxiang Zheng (tz32@illinois.edu)

February 22, 2024

Contents

Introduction

Problem

Making bread at home, especially sourdough has become very popular because it is an affordable way to get fresh-baked bread that's free of preservatives and other ingredients that many people are not comfortable with. Sourdough also has other health benefits such as a lower glycemic index and greater bioavailability of nutrients.

However, the bulk fermentation process (letting the dough rise) can be tricky and requires a lot of attention, which leads to many people giving up on making sourdough. Ideally, the dough should be kept at around 80 degrees F, which is warmer than most people keep in their homes, so many people try to find a warm place in their home such as in an oven with a light on, but it's hard to know if the dough is kept at a good temperature. Other steps need to be taken when the dough has risen enough, but rise time varies greatly, so you can't just set a timer, and if you wait too long the dough can start to shrink again. In the case of activating dehydrated sourdough starter, this rise and fall is normal and must happen several times; and its peak volume is what tells you when it's ready to use.

Solution

Our solution is to design a device with a distance sensor and temperature sensor that can be attached to the top and underside of most types of lids, using magnets. The sensors will be controlled with a microcontroller; and an LCD display will show the minimum, current, and maximum heights of the dough along with the temperature. This way the user can see at a glance how much the dough has risen, whether it has already peaked and started to shrink, and whether the temperature is acceptable or not. There is no need to remove it from its warm place and uncover it, introducing cold air; and there is no need to puncture it to measure its height or use some other awkward method. A typical use case will proceed as follows:

1. The device is attached to a container's lid with the sensor enclosure on the underside of the lid and with the display and buttons on the topside, and the lid is placed on the container.
2. The button to set container height is pressed. The device measures the distance to the bottom of the container and temporarily displays the result so the user can confirm.

3. A ball of dough that needs to rise is placed in the container, and the button to set dough height is pressed. The device records and displays the current height as both starting height and maximum height.
4. The container is placed in a warm location so the dough can rise. The device periodically updates the temperature, current height, and peak height on the display. The starting height continues to be displayed unchanged.
5. When the dough reaches the desired height, or when the user sees that the current height is lower than the peak height, then the dough is removed.

A similar process can be followed for sourdough starter, except that the starter will peak and fall after every feeding, and the user can optionally press the set dough height button each time. The user will decide when the starter is ready based on the peak height rather than the current height. The device will not attempt to tell the user when the dough/starter is ready because that is a judgment call that depends on the type of bread, the recipe, and the baker's personal preference. Instead, the device makes it easy for the user to see the information they need to make their decision.

Visual Aid

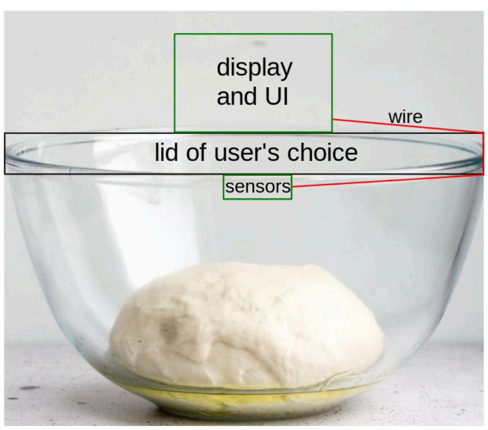

Figure 1: Device on regular dough

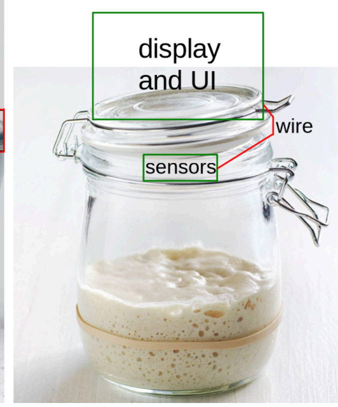

Figure 2: Device on starter

Our device is meant to sit on top of whatever container the dough or starter is in. With the sensors inside the container and the display and other components on the top of the container, we can gather valuable information while easily informing the user what that information is (rise distance and temperature).

High-level Requirements List

- Charge the battery and operate on battery power for at least 10 hours, but ideally a few days for wider use cases and convenience.
- Accurately read and store distance values (+-10% of total height, minimum height 3 centimeters) and temperature values (within 3°C).
- Display the minimum height, maximum height, current height, and current temperature values on a display, updated at least once every five minutes.

Design

Physical Design

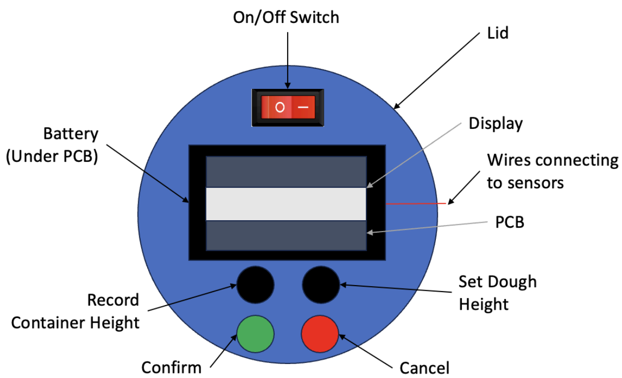

Figure 3: Top View of device

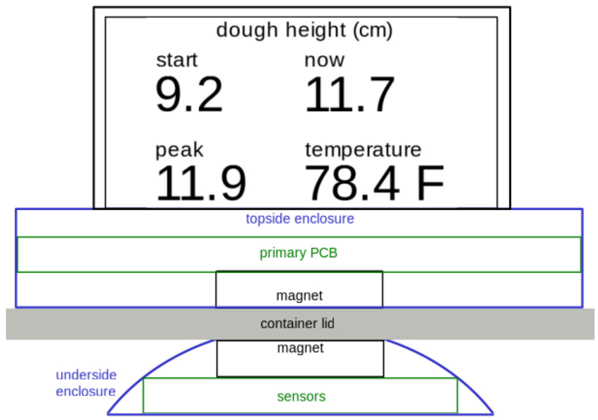

Figure 4: Front View of Device

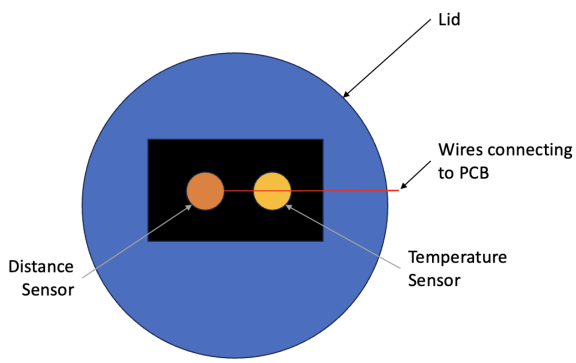

Figure 5: Bottom View of Device

Block Diagram

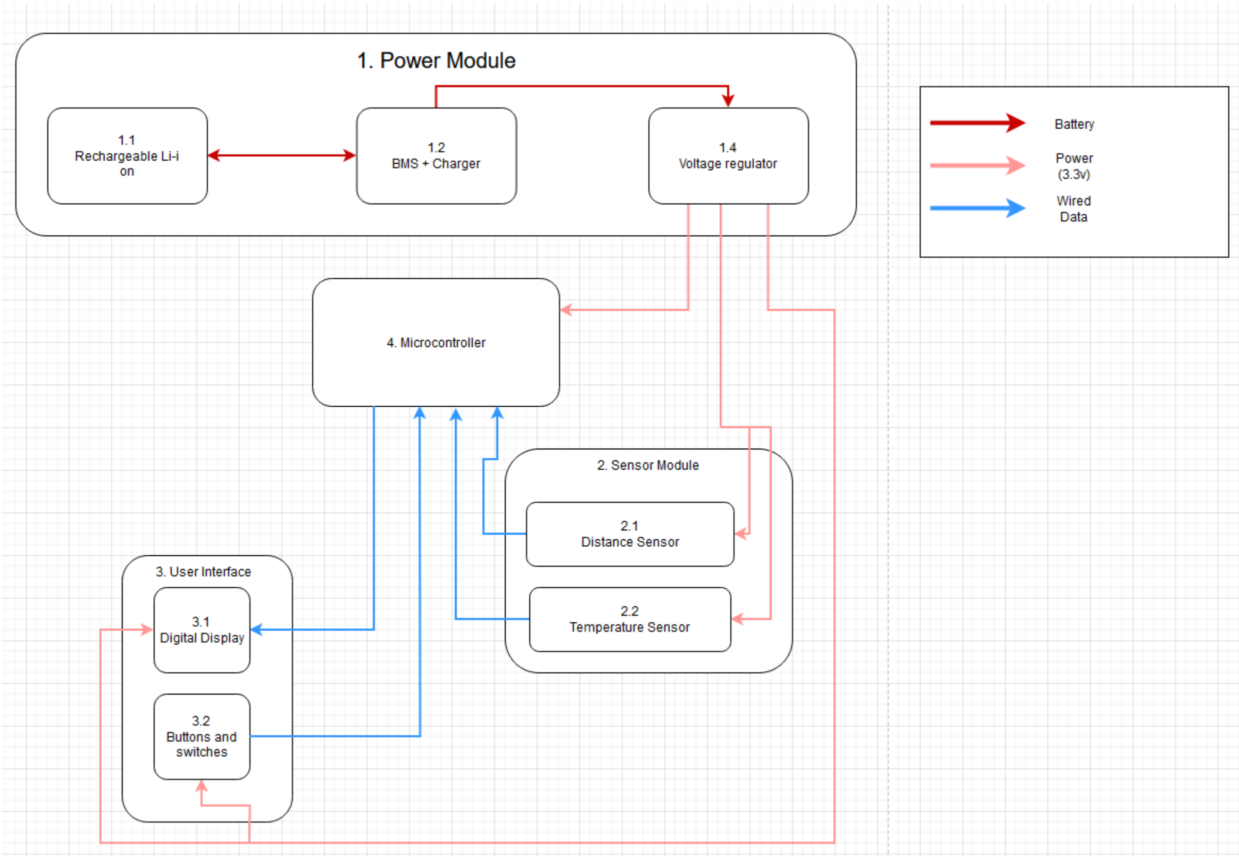

Figure 6: Block Diagram

Power Subsystem

The power subsystem is responsible for powering all other subsystems, even those on the underside of the container lid, and will consist of three main components: a small and rechargeable lithium-ion battery, a BMS + charger module, and a voltage regulator. The Li-ion battery will be relatively small since our design does not have high power consumption. In order to ensure that our components do not take up too much space while having enough capability, we chose a 420 mAh battery (about the size of a quarter) for the current design. We expect this capacity to be enough that it does not need to be recharged during a dough rise. The BMS + charge module will get power input via USB-C and connect directly via the micro-lipo to our Li-ion battery. Our 3.3v voltage regulator will then send power out to the rest of the circuit. For convenience during development and testing, the power subsystem will also include an external power connector and a power multiplexer that allows us to switch between battery and external power. Schottky diodes will protect the device from power being connected with reverse polarity, and a TVS (Zener) diode will prevent excessive voltages from static discharge or other sources.

This subsystem will be located on the main PCB on the top side of whatever lid is on the dough container (see **Figure 3**).

Table 1: Power Subsystem Requirements and Verification

Requirements	Verifications
The battery must last for at least 10 hours.	- Apply a 30mA load to the battery and record the time it takes for a full battery (4.2V ± .05V) to drop down to 3.4V ±.05V using a multimeter. Check the time elapsed and ensure it is at least 10 hours.
Recharge the battery through a 5V USB-C port that is exposed to the user in less than two hours.	- The USB-C port must be exposed enough and in a location where one can plug a USB-C cable into the port. - The CHG LED on the BMS + Charging module needs to light up to indicate the battery is being charged when connected to power (if it is not fully charged). - Take a discharged battery (3.4V±.05V) and connect it to the charging module. Power the USB-C charging module. If the CHG light goes off and the DONE light comes on within two hours, check the battery voltage with a multimeter and verify it is 4.2V ± .05V. If the CHG LED stays on for the full two hours, check the battery voltage at two hours and verify it is 4.2V ± .05V.

Sensor Subsystem

Sensors will be placed on the part of the device that attaches to the underside of a lid. A temperature sensor will measure the ambient temperature near the dough to ensure the dough is kept at an acceptable temperature. A proximity sensor or sensors will first measure the height of the container, and then begin measuring the height of the dough periodically. We plan on using one VCSEL sensor as the height sensor and believe that to be sufficient based on our research since VCSEL has a narrower beam that would not be as prone to potential interference from the container walls as other sensors (see **Figure 7**). This would also allow the user to put the sensor module wherever they want to get a reading (ex: maybe they want to get a reading of dough height near the edge of a pan instead of in the middle. This sensor would allow that). The distance sensor will communicate with the MCU via a wired I2C interface.

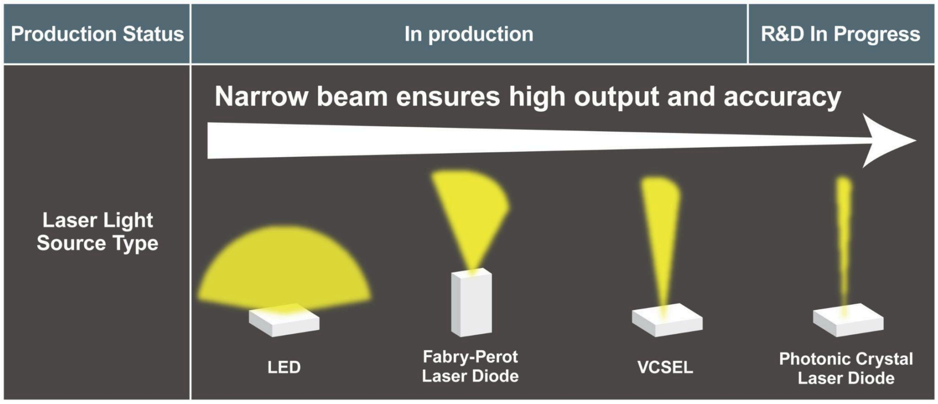

Figure 7: Difference in distance sensor beams

We will be using IC as the type of temperature sensor, specifically the LM235AZ from Texas Instruments. This is a low-cost analog temperature sensor with measurement tolerances that are just fine for this project since extremely accurate temperature readings are not necessary for the user to determine if they are close to the temperature they would like to be at (see **Figure 8**). Since our accuracy does not need to be extremely high, and operating temperatures will typically be at or above room temperature, the IC sensor is very attractive with its low cost, power consumption, and complexity. The MCU will read the temperature as the voltage across the temperature sensor with a conversion factor of 10 mV/°K.

Criteria	RTD	Thermistor	Thermocouple	IC sensor
Temperature range	-250°C to +750°C	-100°C to +500°C	-267°C to +2316°C	-55°C to +200°C
Accuracy	Best	Depends on calibration	Good	Good
Linearity	Good	Worst	Good	Best
Sensitivity	Less	Best	Worst	Good
Circuitry	Complex	Depends on accuracy/power requirements	Complex	Simplest
Power consumption	High when taking measurement		Low-high	Lowest
Relative system cost	$$-$$$	$-$$$	$$-$$$	$

Figure 8: Different types of temperature sensors

Table 2: Sensor Subsystem Requirements and Verification

Requirements	Verifications
Measure the ambient	- Place our device in a container along with a known

temperature inside a container with an accuracy of ±3°C and transmit the data to the MCU at a rate of at least 1 reading/second.	accurate thermometer. The temperature from the board's microcontroller via serial debugging must be ±3°C of the temperature from the thermometer. - Output the time between readings after each reading to the serial debugging terminal and confirm that the MCU is getting a temperature value at least once a second
Measure the distance from the sensor to the bottom of an empty container and the distance to an object inside the container with an accuracy of of ±1 cm at distances below 10 cm and ±10% at distances from 10-20 cm and transmit the data to the MCU at a rate of at least 1 reading/second.	- Measure the height of a container with a ruler. Attach the sensor to the underside of a lid placed on the container and read the distance from the board's microcontroller via serial debugging. Repeat the above with an object inside the container, after measuring the object's height with a ruler. The distance read from the sensor must be ±1 cm of the ground-truthed value at distances below 10 cm and ±10% at distances from 10-20 cm - Output the time between readings after each reading to the serial debugging terminal and confirm that the MCU is getting a distance value at least once a second
Taking the lid off the container and putting it back on does not significantly change the distance value if it should be the same (nothing added to the inside of the container).	- The readings from the serial debugging terminal after replacing the lid must be ±.5 cm of the initial value if the initial value was under 10 cm total or ±5% if the initial value was between 10-20 cm.

User Interface Subsystem

The UI will attach to the top of the lid and consist of several simple push buttons, a switch, and a display to control and monitor the device. The switch will turn the device on and off, a button will measure the height of the container, a button will set/reset the minimum dough height, two buttons will allow the user to confirm or cancel a choice, and a display will show important data to the user (current height, max height, current, temperature), etc.

The user interface will include the following buttons, LEDs, switch, and USB port:

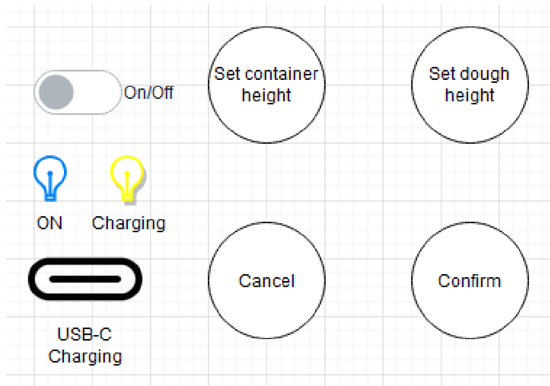

Figure 9: Example UI

The device and MCU will not have power when the switch is in the off position. Once powered on, the device will be waiting on user input for the container height and the starting dough height. To set the container height, the user presses "Set Container Height" and then "Confirm." This tells the MCU to store the distance value it gets when "Confirm" is pressed as the distance to the bottom of the container. A similar process can be done for the min/max height, but once "Confirm" is pressed, both the minimum and maximum height are recorded by the MCU. The minimum height will stay the same until set again, but the maximum height will change based on what the sensor records periodically, now. Should the user accidentally press a set button when not meaning to, they can press cancel to go back to the prior state. The user will be able to tell the device is on by an "ON" LED (the MCU has power) and that it is charging by a 'Charging" LED (the USB charging module is being supplied power).

Table 3: User Interface Subsystem Requirements and Verification

Requirements	Verifications
Turn the device on and off with a switch on top of the device.	- Use a voltmeter to measure a 3.3V ± .3V. output from the voltage regulator whenever the switch is moved from the off to the on position - Verify that the display has blank outputs for the readings or prompts the user to set the height values when the switch is moved from off to on - When the switch is moved from on to off, use a voltmeter to measure 0V ± .1V from the voltage regulator
Measure or reset the starting height of the dough with a button.	- See **Figure 9**. When the device is on, pressing the "Set dough height" button and then confirming with "Confirm" should reset the minimum and maximum dough heights shown on the

	display - Verify the minimum height does not change when the maximum height is updated via the sensor data unless it is reset or the device is power cycled
Measure or reset the container height with a button.	- See **Figure 9**. When the device is on, pressing the "Set container height" button and then confirming with "Confirm" should reset the minimum and maximum dough heights output to the serial console - Verify this value displayed is not modified unless it is reset or the device is power cycled
Temporarily display the container height when measured, just for confirmation.	- When the "Set container height" button is pressed, display current height measured and the current stored container height, even before the "Confirm" is pressed - Pressing confirm or cancel needs to go back to the normal screen displays all of the dough statistics
Display the minimum height, maximum height, current height, and current temperature values on a display, updated at least once every five minutes.	- Use a serial console to verify the time between updates sent to the display is at least once every five minutes - The maximum height should be taken as the maximum reading from the distance sensor since the last reset or power cycle. Verify there is always a value shown on the display when the display is showing dough statistics - The minimum height will need to be set if it hasn't been since the last power cycle. If it has been set, verify there is always a value shown on the display when the display is showing dough statistics. If it hasn't been set yet, the display should indicate that to the user in its value field - Verify there is always a current height value shown on the display when the display is showing dough statistics. - Verify there is always a temperature value shown on the display when the display is showing dough statistics.
Pressing the "Cancel" button should not update, even with a subsequent push of the "Confirm" button	- Ensure the current dough height is at least a 1 cm difference from the minimum height. Press the "Set dough height" button. Press the "Cancel" button. Press the "Confirm" button. Verify the value on the display for minimum dough height has not changed.

Microcontroller Subsystem

The microcontroller (MCU) subsystem will interact with the other subsystems at least once every five minutes. Both sensors are connected to the MCU's PCB by wires, and the MCU will get the required data by reading the voltage at the output of the temperature sensor and reading the data sent by the distance sensor via the I2C protocol. The MCU will take 10 readings within 10 seconds for both sensors, calculate the average, store the data, and send it to the display using the SPI protocol with a wired connection and a development kit for the E-ink display. The MCU will also take input from the buttons and switches in the user interface subsystem, which will be essential on the startup of the device to make sure the output on the display is accurate.

The state machine below is how our buttons interfacing with the MCU will work. Important to note: we must press one of the "set" buttons before pressing "Confirm" to change any values, and pressing "Cancel" in any state will take it back to the base state. The modification states are simply to show that we should modify the stored values at that point. User input would have no impact when we are in these "states", and we automatically go back to the base state once the value has been modified.

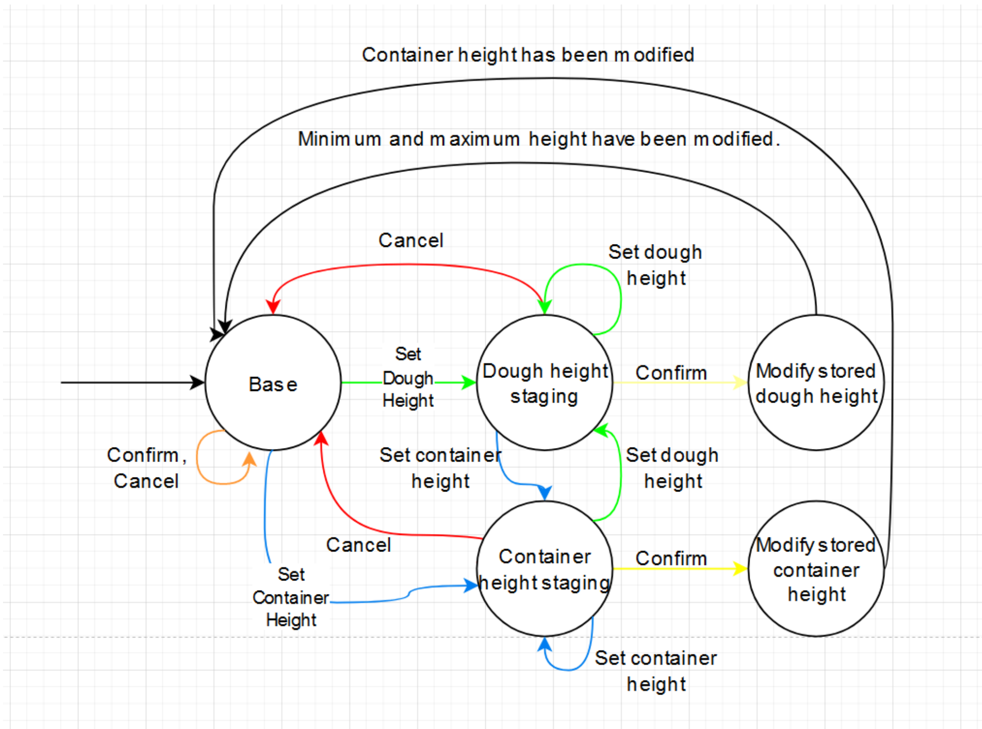

Figure 10: Button Input State Machine

Table 4: Microcontroller Subsystem Requirements and Verification

Requirements	Verifications
Store the recorded values from the sensor subsystem for the current "rise", specifically distances (container, dough starting, dough peak, and dough current) and the temperature.	- Print out ten values every ten seconds for each sensor to the serial console during testing
Compute the average of several consecutive readings to reduce the effect of noise.	- Print ten values taken in a ten-second span for each sensor. Take the average of the ten values for both of the sensors. Print out the stored temperature value that is used to update the display. Verify the average of these ten values is the value used to update the display for each sensor (within ±0.01 for rounding differences)
Use the recorded distance values to calculate dough height (container floor distance minus dough distance).	- Ensure values exist already for dough height and container height by printing those values to the serial console. Verify the {container distance} - {dough distance} = {dough height stored for

	display}
Send an error message to the display if the dough reaches zero distance from the sensor.	- Place an object against the sensor and press the "set dough height" button. Verify that an error message is displayed.
Update the values on the display at least once every five minutes or within 5 seconds of the user pressing a measure/reset button	- Ensure the current distance is substantially different than the minimum (at least 1cm difference). Press the "Set dough height" button and then the "Confirm" button. The time it takes for the value on the display to update needs to be less than five seconds. - When no user interaction has occurred with the buttons or the switches, time the interval between the display updates and print this value to the serial console once the display is updated, ensuring it is less than 300 seconds (5 minutes).

Tolerance Analysis

One important aspect of the project that poses a risk to successful completion is power, especially power dissipated by the voltage regulator, and ensuring a safe operating temperature. Overheating and causing damage to the components, the dough (or starter), or increasing the overall temperature of the dough beyond what is desired would be detrimental to the project and the safety of the device. We will be looking at the upper bound on the power consumption to determine if our voltage regulator will be within a safe operating range.

The AP2112K-3.3TRG1 is our linear voltage regulator.

Table 5: Max current draws for components at 3.3V

Part (Operating at 3.3V)	**Max Current Draw at 3.3V**	**Comments**
STM32F303K8	160 mA [8]	Average case is much lower than the worst case.
VCNL36826S	20 mA [4]	VCSEL Sensor
LM335AZ/NOPB	5 mA [5]	Temperature Sensor
E2266PS0C2	2.94 mA [6]	E-Ink TFT Display

CTL0603FGR1T	20 mA [7]	LED indicator
Total	207.94 mA	

This current draw is less than the 600mA max for the AP2112K-3.3TRG1.

Using $T_j = i_{out}(v_{in} - v_{out})(\Theta_{jc} + \Theta_{ca}) + T_a$ to estimate the junction temperature of the voltage regulator:

Table 6: Variables, values, and comments for tolerance equation

Variable	**Value**	**Comments**
max(T_j)	150 C	Max operating temperature of AP2112K-3.3TRG1
i_{out}	207.94 mA	Max current draw @ 3.3V
v_{in}	5.5 V	Max input voltage from USB charging port
v_{out}	3.3 V	Operating Voltage
Θ_{jc}	95° C/W	Junction-to-case thermal resistance of AP2112K-3.3TRG1
Θ_{ca}	89° C/W	Max Junction-to-ambient was 184° C/W for AP2112K-3.3TRG1
T_a	38 C	Assume around 100F if a really warm dough environment

With the above values, our T_j = 122.17 C, which is less than the max of 150 C, meaning temperature and power risks are not high for our circuit.

Cost and Schedule

Bill of Materials

The average starting salary of ECE grads can be found on the ECE Illinois website [3] showing that the average starting salary of a member of our group is about $102k (two CompEs and one EE). Assuming a regular work week and two weeks of paid vacation per year, that total comes out to about $50 an hour. We expect 45 hours per team member, bringing our cost per team member to $50/hr * 2.5 * 45 hours = $5625 for each team member, or $16875 in total labor costs.

Table 7: Itemized list of components and costs

Description	Manufacturer	Part Number	Quantity	Total Cost	Link
STM32 MCU	STMicroelectronics	STM32F303K8T6TR	1	$6.29	Link
Analog Temperature Sensor	Texas Instruments	LM335AZ/NOPB	1	$1.40	Link
I2C IR Distance Sensor	Vishay Semiconductors	VCNL3040	1	$2.07	Link
On-off switch	E-Switch	RA1113112R	1	$0.67	Link
LED indicator	Venkel	CTL0603FGR1T	2	$0.04	Link
Tactile Switch Buttons	CUI Devices	TS02-66-50-BK-160-LCR-D	4	$0.40	Link
Linear Voltage Regulator	Diodes Incorporated	AP2112K-3.3TRG1	1	$0.35	Link
420 mAh Li-Ion Battery	Adafruit	4236	1	$6.95	Link
Battery charger	Adafruit	4410	1	$5.95	Link
External power connector	TE Connectivity AMP Connectors	1-2213629-2	1	$0.14	Link
Power multiplexer	GLF Integrated Power	GLF4028	1	$0.48	Link
Schottky diode	Toshiba	CUS08F30,H3F	2	$0.64	Link
TVS diode	Diodes Incorporated	SD05-7	1	$0.36	Link
2.66" EPD (display)	Pervasive Displays	E2266QS0F1	1	$7.18	Link
EPD dev kit	Pervasive Displays	B3000MS044	1	$16.00	Link
1x10 EPD header	Adam Tech	PH1-10-UA	1	$0.17	Link
1x6 UART header	Sullins Connector Solutions	PPTC061LFBN-RC	1	$0.52	Link

2x5 debug header	On Share Technology	302-S101	1	$0.33	Link
1x3 header	Adam Tech	PH1-03-UA	4	$0.40	Link
1x2 jumper	Sullins Connector Solutions	SPC02SYAN	4	$0.40	Link
Neodymium magnets	MAGXCENE	n/a	2	$2.00	Link
Capacitor 10n	Samsung Electro-Mechanics	CL21B103KBANNNC	1	$0.10	Link
Capacitor 100n	Samsung Electro-Mechanics	CL21B104KBCNNNC	6	$0.50	Link
Capacitor 1µ	Samsung Electro-Mechanics	CL21B105KAFNNNE	6	$0.20	Link
Capacitor 4.7µ	Samsung Electro-Mechanics	CL21A106KOQNNNE	1	$0.10	Link
Resistor 22	Cal-Chip Electronics	RM10J220CT	1	$0.002	Link
Resistor 60.4	Cal-Chip Electronics	RM10F60R4CT	2	$0.006	Link
Resistor 270	Cal-Chip Electronics	RM10J271CT	1	$0.002	Link
Resistor 2.2k	Cal-Chip Electronics	RM10J222CT	2	$0.004	Link
Resistor 10k	Cal-Chip Electronics	RM10J103CT	5	$0.008	Link
Resistor 100k	Cal-Chip Electronics	RM10J104CT	3	$0.006	Link

Total Cost: $16,929.15

Parts total: $54.15 **Labor Cost**: $16,875

Schedule

Table 8: Schedule for Final Project

Week	Task	Person
February 19 - February 25	Review and Update Proposal	Everyone
	Complete Design Document	Everyone
	Start General Circuit/Board Design	Jake
	Find Temperature and Distance Sensors	Jake
February 26 - March 3	Design Review (2/27 3:30 PM)	Everyone
	Begin Board Design	Jake
	PCB Review	Jake
	Order sensors and other parts for general testing/prototyping	Everyone

March 4 - March 10	Finalize initial PCB design for the first order	Jake
	Teamwork Evaluation (3/6 11:59 PM)	Everyone
	First Round PCB Orders (3/5 4:45 PM)	Jake
	Start developing the code to program the microcontroller to connect the sensors and the display	Abhitya/Alec
March 11 - March 17	Spring Break	Everyone
March 18 - March 24	Make revisions to PCB design	Jake
	Continue revising microcontroller program	Abhitya/Alec
	Second Round PCB Orders (3/19 4:45 PM)	Jake
	Assemble the first prototype board	Jake/Abhitya
	Program the first board and test the sensor, button, and display connectivity	Everyone
	Begin designing the 3D printed enclosure for device	Abhitya/Alec
March 25 - March 31	Make revisions to PCB design and code for microcontroller	Everyone
	Print 3D enclosure and test with current prototype	Abhitya/Alec
	Third Round PCB Orders (3/26 4:45 PM)	Jake
	Create 2nd prototype and test functionality of buttons, sensors, and display and place the device within the enclosure	Everyone
April 1 - April 7	Finalize PCB design based on testing	Jake
	Finalize microcontroller programming to ensure connectivity within the device	Abhitya/Alec
	Finalize enclosure design based on previous testing	Abhitya/Alec
	Fourth Round PCB Orders (4/2 4:45 PM)	Jake
	Test prototype in actual dough rising situation and monitor results	Everyone
April 8 - April 14	Create final prototype with all final pieces	Everyone
	Test the final prototype for full functionality	Everyone
	Fifth Round PCB Orders (4/9 4:45 PM)	Jake
April 15 - April 21	Mock Demo (4/16 3:00 PM)	Everyone
	Record testing of device with actual dough	Everyone
	Final assembly and presentation of device for demo	Everyone

April 22 - April 28	Final Demo w/ Instructor and TAs	Everyone
	Mock presentation w/ TAs	Everyone
	Work on final paper	Everyone
April 29 - April 31	Final presentation	Everyone
	Finish writing the final paper	Everyone

Ethics and Safety

IEEE Code of Ethics

In our project, we aim to create a product that can be used by people to make their own bread at home. This requires us to create a product that "hold[s] paramount the safety, health, and welfare of the public" [2] who are its users. This also holds for creating an ethical and sustainable design that does not endanger the user or environment. These standards are further expanded upon in the food safety requirements section.

We commit to upholding the standards of ethical and professional conduct throughout this project as listed in the IEEE Code of Ethics. We commit ourselves to avoiding conflict, offering honest criticism, and communicating with each other if any problems arise. We will treat each other fairly and respectfully, being non-discriminatory, and refraining from harassment and injury of group members.

We strive to support each other throughout this project, maintaining and improving our technical competence while holding ourselves to the highest standards for the delivery of this endeavor.

Food Safety Requirements

Our product falls under the category of food storage, as the dough may come into contact with the system when rising. The NSF/ANSI 2 documentation [1] establishes food protection and sanitation standards for any equipment that comes into contact with food. There are 4 primary standards that we will need to follow to ensure compliance with these standards.

- Material Safety: The parts of the system that can potentially come into contact with the dough must be made of materials that are non-toxic and can't migrate chemicals into food upon contact

- Cleanability/Sanitization: The product should be designed to be easily cleaned and sanitized to reduce possible contamination
- Durability: The product should be durable enough to withstand deep cleaning and continuous use without affecting the food involved
- Documentation: The materials used in the product should be properly documented and to ensure compliance with state and federal standards

Safety Concerns

The major safety concerns with our project concern food contamination, the materials we utilize throughout the exterior of the product cannot have any toxic materials or have a possibility of contamination. The product must also be able to withstand a general cleaning and sanitation process if any contaminants do come in contact with it or if any non-food product is found on its surface. There are also some electrical safety concerns regarding the wiring between the top and bottom halves of the solution which will need to withstand some external factors and pressures so that they pose no harm to the user.

The battery and live electrical components that are utilized within the device must not be exposed to the user as that could cause harm to anyone who comes in contact with it. We must ensure that the device as a whole is safe for long-term use and operation and that the components will both individually and collectively not pose a risk to the user.

We understand the overall risk associated with creating an electronic product to be utilized with food and maintain the utmost safety regulations to create the most effective and safe product possible.

Sources

[1] B. Kelechava, "NSF/ANSI 2-2022: Food Handling Equipment - ANSI Blog," *The ANSI Blog*, Aug. 16, 2023. https://blog.ansi.org/nsf-ansi-2-2022-food-equipment-standard/ (accessed Feb. 22, 2024).

[2] IEEE, "IEEE Code of Ethics," *ieee.org*, Jun. 2020. https://www.ieee.org/about/corporate/governance/p7-8.html (accessed Feb. 22, 2024).

[3] "Salary Averages," *ece.illinois.edu*. https://ece.illinois.edu/admissions/why-ece/salary-averages (accessed Feb. 22, 2024).

[4] "Proximity Sensor With Interrupt, VCSEL, and I 2 C Interface," Vishay Semiconductors, Feb. 2022. Accessed: Feb. 22, 2024. [Online]. Available: https://www.vishay.com/docs/84964/vcnl36826s.pdf

[5] "LM135, LM135A, LM235, LM235A, LM335, LM335A LMx35, LMx35A Precision Temperature Sensors 1 Features 3 Description Basic Temperature Sensor Simplified Schematic Calibrated Sensor," Texas Instruments, Feb. 2015. Accessed: Feb. 22, 2024. [Online]. Available: https://www.ti.com/lit/ds/symlink/lm335.pdf?ts=1708655741439&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FLM335%252Fpart-details%252FLM335AZ%252FNOPB

[6] "E2266PS0C1 E2266PS0C2 Product Specifications," Pervasive Displays, Aug. 2022. Accessed: Feb. 22, 2024. [Online]. Available: https://www.mouser.com/datasheet/2/1089/Pervasive_displays_8_24_2022_1P332_00_01_E2266PS0C-3009734.pdf

[7] "CTL0603FGR1T DATASHEET," Venkel LTD. Accessed: Feb. 22, 2024. [Online]. Available: https://page.venkel.com/hubfs/Resources/Datasheets/LEDs/CTL0603FGR1T.pdf

[8] "STM32F303x6/x8," STMicroelectronics, Jul. 2018. Accessed: Feb. 22, 2024. [Online]. Available: STM32F303x6/x8