ECE 445 Design Documentation

<u>StoveSense</u>

Nikil Nambiar (nikiln2)

Aryan Gupta (aryang4)

Dinal Gunaratne (dinalg2)

TA: Stasiu Chyczewski

Professor: Arne Fliflet

Date: September 15, 2023

Contents

1. Introduction

1.1. Problem:

In recent years, there has been a concerning rise in the number of house fires attributed to stoves being left unattended. Nearly 50% of house fires are caused by burners being left on and unattended. In addition, being able to control a stove away from the knobs allows for more control while cooking. As a result, there should be an easy solution where a user can remotely control and turn off any burner that is on.

1.2. Solution:

Our solution involves having heat sensors on each of the stoves to determine which burners are on and relay this information to the user via an app. The user will be able to see which stoves are on, and control each stove remotely. Each stove knob will have a pulley system that will rotate the knob to the desired level. This will work by having a motor that will serve as a tensioner and tension a belt around the shaft of the knob. When the tensioner isn't activated the user will be able to freely rotate the knob. Through this design we will be able to automatically rotate the stove knob, while also allowing the user to control the knob without our mechanisms affecting them.

All communication between the app and the microprocessor will be done over a network connection. This microprocessor will let us control the knob turner remotely and with precision.

This solution will also have automated features. We plan to add a water-proof thermocouple the user can manually put in pots with soups or other liquids to add boil over protection. This device will monitor the temperature and automatically turn down the temperature of the stove once there is a risk of a boil over (temperature rising significantly above boiling point).

Our app will contain a visual interface which allows users to see which exact burner is on, and change the burner intensity to whatever is desired, including off. Additionally, we will add push notifications to notify the user if a burner is on or if boil over was detected and handled. Finally, our app will allow users to set a timer for how long they want the stove on. After the timer ends our app will relay this information to the microcontroller telling it to turn off the stove.

1.3. Visual Aid:

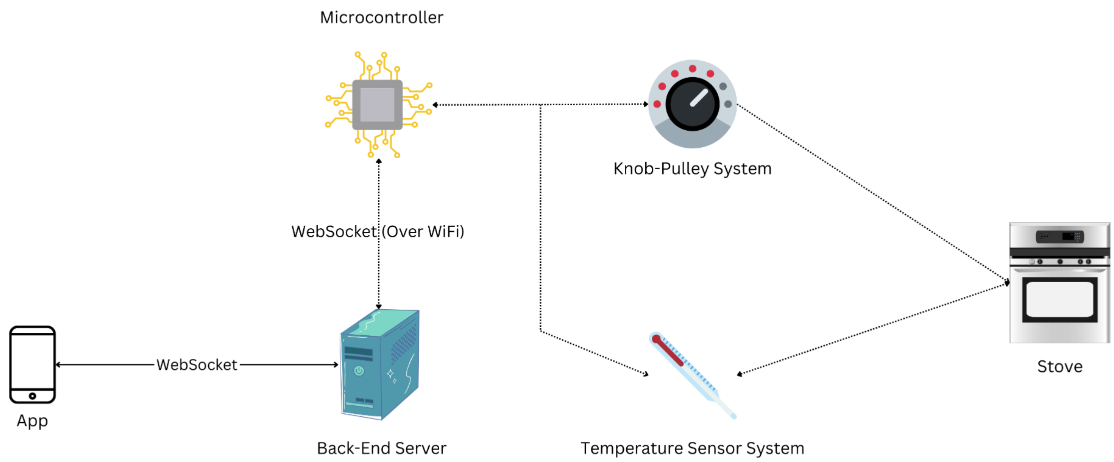

Figure 1: The scope of StoveSense from the physical product to the application for user control

1.4. High-Level Requirements:

In order for StoveSense to be considered successful, it must fulfill the following:

1. Product must be capable of maintaining water temperature at 100 degrees celsius to prevent over-boiling and spilling out of the pot. This will be made possible by constant monitoring of the data relayed by the LM35DZ in the pot of water. When the temperature surpasses 100 degrees celsius, the stove knob turner will slightly dial back the knob to lower the temperature back down.

2. Product must be able to turn corresponding knob kX degrees to turn knob to desired level, where X corresponds to the amount of degrees required to turn the knob a level, and k corresponds with the amount of levels the user desires to change by.

3. Users must be able to communicate with the microcontroller via the app with a latency of less than 1 second. The physical system should be able to complete the operation (e.g turn down stove, turn up stove) in less than 10 seconds. In total, end-to-end operation time should be 11 seconds or less.

2. Design

2.1. Diagrams:

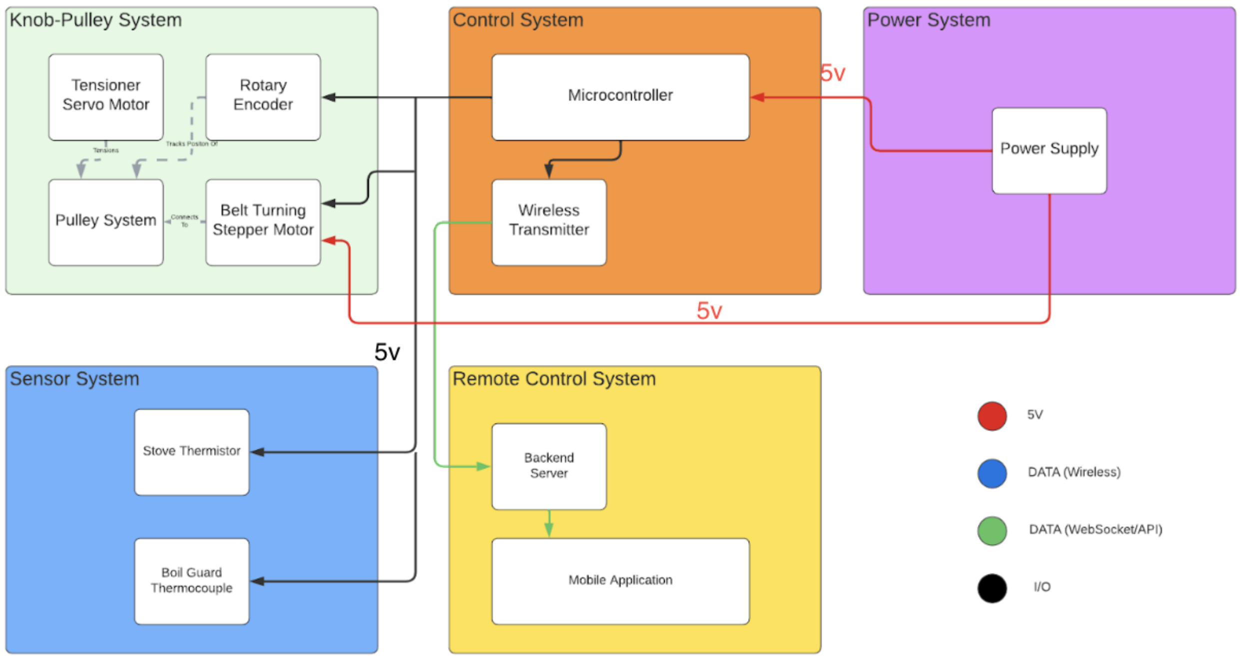

Figure 2.1: StoveSense high-level block diagram

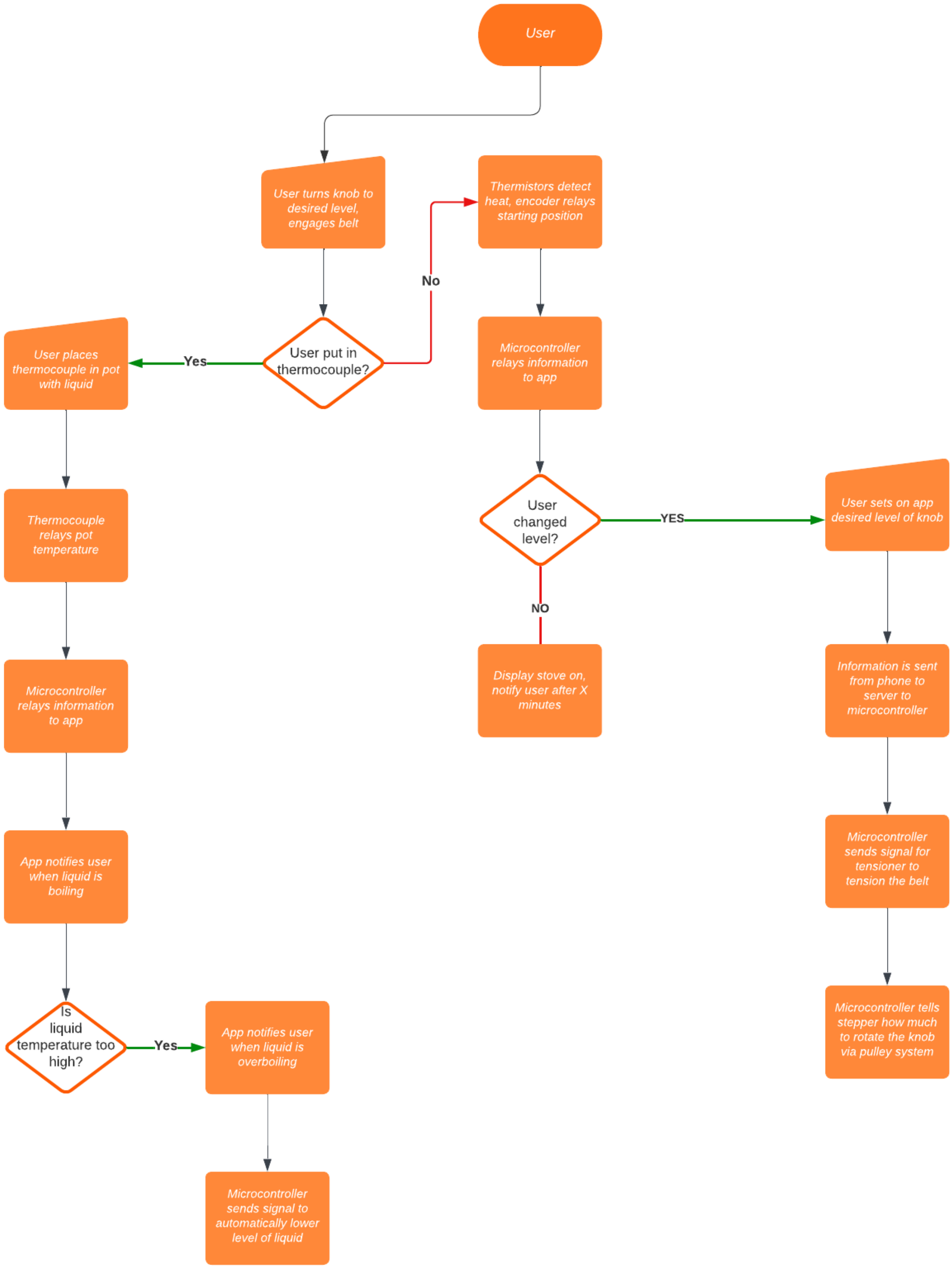

Figure 2.2: StoveSense high-level flowchart

2.2. Physical Design:

For the autonomous turning of the stove knob, we will be implementing the design shown below in Figure 3. The design consists of a pulley and belt system, the stove knob/shaft, an incremental rotary encoder, one servo-motor, and one stepper-motor.

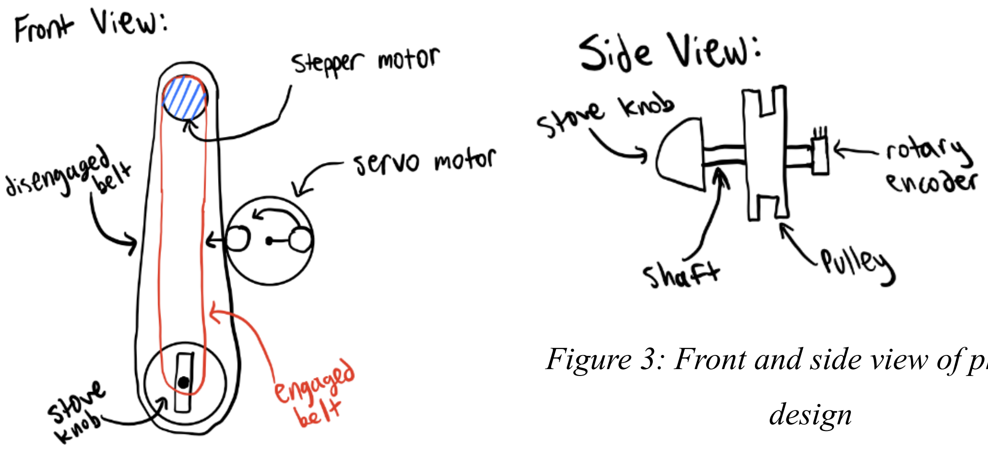

Figure 3: Front and side view of physical design

The stepper motor will be used to drive the belt along the pulley, which will only turn the stove knob when the belt is engaged. The belt is engaged by the servo motor positioned to the side of the knob. We are implementing this design choice to allow the user to have resistanceless control over their stove knob when they do not want StoveSense active. As soon as the user prompts the application to adjust the stove knob, the servo motor will push against the disengaged belt until it is taught against the pulley system and is therefore engaged. The rotary encoder will be mounted behind the shaft that the stove knob rests on, so that whether the belt is engaged or not, the microcontroller will always have an indication of the position of the stove knob and will be able to accurately adjust to the setting that the user has specified.

2.3. Subsystem Overview & Requirements/Verification:

2.3.1. Sensor Subsystem:

Our sensor subsystem consists of three sensors, a thermistor, LM35 Temperature Sensor, and a Incremental Rotary Encoder. Our thermistors will be connected to the PCB and attached near each stove knob to check the status of the stove. The circuitry will be encased in a heat resistant material such as thermal pads while the ends of the thermistors will be open and exposed to the heat of the stove tops. We can get the temperature for the thermistor through calculating the rise or drop of its resistance. “The thermistor comes at a given resistance, we can use a voltage divider circuit with known resistance that is connected in series with a thermistor, and apply a voltage reference to one end of the resistor and connect the other end of the thermistor to ground, the voltage across the thermistor is read to determine the temperature based on the voltage reading" (Chueng, 2019).

$$\frac{V_{out}}{R_t} = \frac{V_s}{R_1 + R_t}$$

$$V_{out} = \frac{V_s \cdot R_t}{R_1 + R_t}$$

$$R_t = \frac{R_1 \cdot V_{out}}{V_s - V_{out}}$$

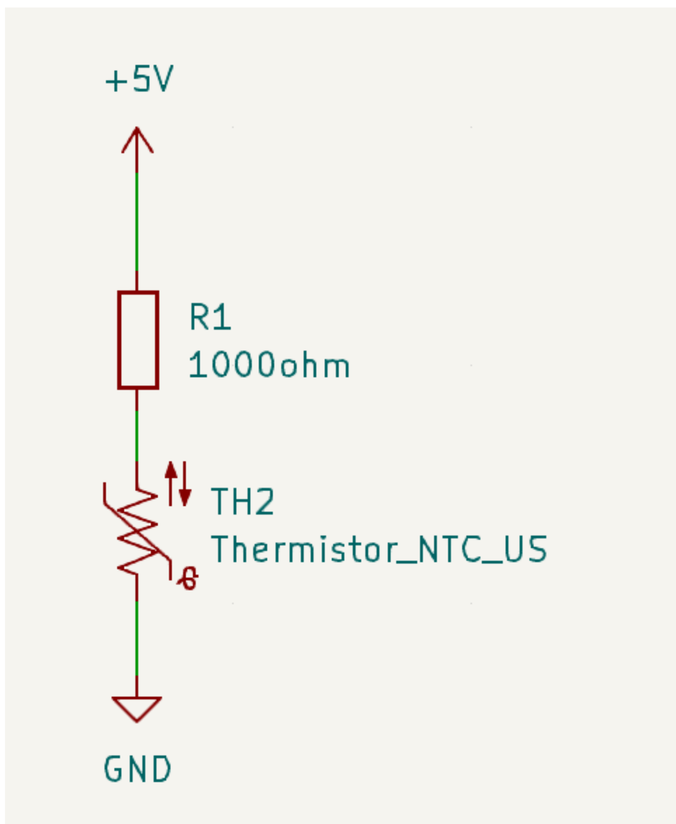

Figure 4: Formula to get R_T and schematic for Thermistor

Our LM35 Water Resistant Temperature will also be connected to the PCB and held in a pot of a liquid. Users will be able to manually put the LM35 in desired items (e.g soup, water). The LM35 will constantly relay its data to the microcontroller so that the stove knob turner can regulate the temperature of the desired liquid. We will use analog to digital converter software to convert the data voltage highs of lows from our LM35 to digital temperature values. See Figure 6 in the control subsystem to see it wired up to the ESP32 MCU.

The incremental rotary encoder will be used to determine the positioning of the stove knob. The incremental rotary encoder keeps track of relative position. For every movement counterclockwise or counterclockwise it'll send a signal high or low to the MCU and that can be used to determine how much the knob has moved. There are two outputs for the incremental rotary encoder A and B. On the rising edge of A, if B is high then the knob is being turned counterclockwise and if B is low then the knob is being turned clockwise. By knowing the initial position we can track the position of the stove knob through performing a series of calculations based on the signals received from A and B. We wish to measure every 18 degrees of rotation. This will give our rotary encoder's resolution as 360 / 18 = 20 degrees of pulses per rotation. This can be tracked by having a counter that increments by 1 for every 18 degrees of rotation and decreases by 1 for every 18 degrees of rotation.

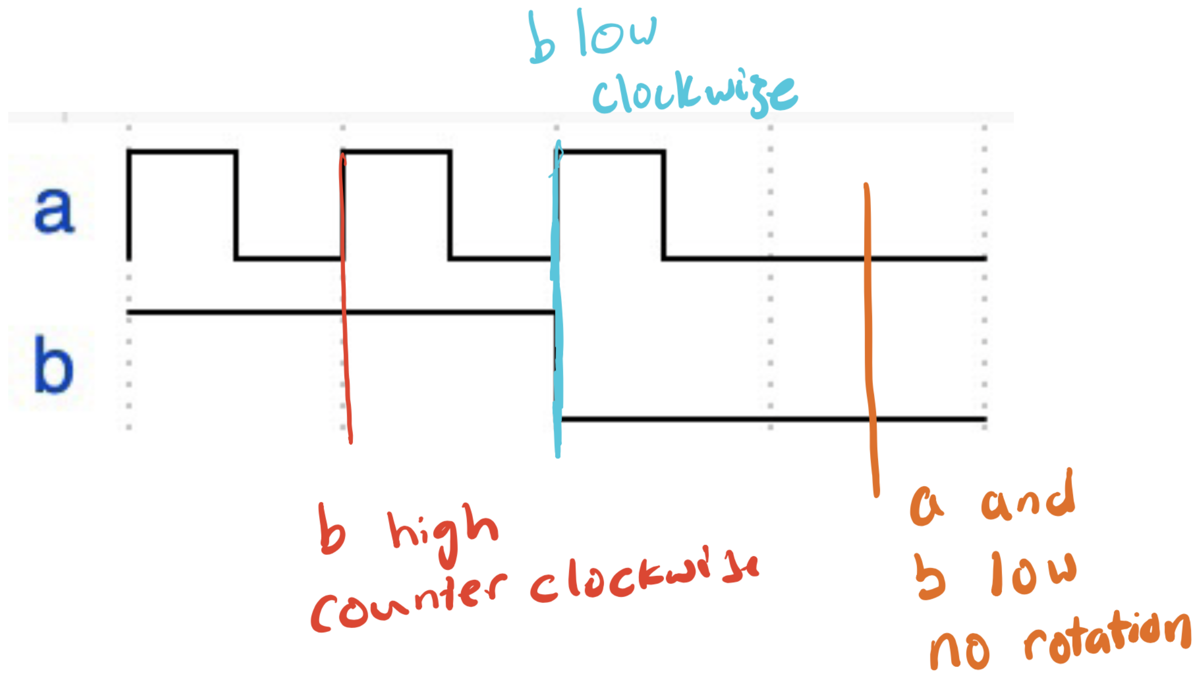

Figure 5: Timing diagram for rotary encoder

Requirements	**Verification**
LM35 Waterproof sensor must accurately be able to read the temperature of liquid to within 3°C.	• Set a pot of water to boil. • Place an electronic thermometer into a pot of water. • Temperature reflected on our backend server from the waterproof sensor should be within 3°C of the thermometer.
Thermistor must accurately be able to read the temperature of the stove to within 5°C.	• Turn on one stove. • Use an electronic thermometer to measure the temperature of the stove. • Confirm that the temperature read from the stove is within 5°C of the

	actual temperature read by the thermometer on our backend server.
Incremental Rotary Encoder must accurately be able to relay the position of the stove knob to the backend server. There should be an update relating to the level of the stove for every 18 degrees of rotation.	● Turn the stove knob 19 degrees using a protractor or an angle measuring app. ● The rotary actuator encoder should relay to the MCU the relative position of the knob which should precisely determine that the stove knob is at level 1. ● This number should be portrayed on our backend server.

2.3.2. Tensioner Subsystem:

The tensioner subsystem consists of a servo motor that is being used to engage and disengage the pulley subsystem. The purpose of having a tensioner is to ensure that the user can turn their stove knob without any resistance when they are not using the assistance of StoveSense. The servo motor will have an extension from its shaft that will come in contact with the belt in the pulley subsystem. In order for the belt to be kept taught against the pulley when engaged, the servo motor must exert between 22.2 and 66.7 Newtons (5-15 pounds) of force on the belt. The tension on the belt needs to remain that high throughout the duration of the pulley subsystem's functioning. To make sure that this is the case, we will shut off the servo motor and lock it in place when it is maximally tensioning the belt and pulley system. After the stove knob has been autonomously adjusted, the servo will be reactivated and turned 180° to be fully disengaged. The servo motor we have selected is rated for anywhere between 4.8 - 6.0 volts of power, so we will be able to feed it 5V directly from our PCB.

Requirements	**Verification**
The servo motor must be able to	● Use a force gauge to

exert 22.2 - 66.7 Newtons of force against the belt throughout the duration of the pulley subsystem's functioning time. The servo must be able to exert this much force to allow the pulley to be taught enough to drive the rotation of the stove knob smoothly.	measure the force exerted by the servo motor. ● Make sure the gauge is reading somewhere between 22.2 Newtons and 66.7 Newtons of force. ● This is the force needed to safely rotate a stove knob. ● See that there is no slack between the belt and the pulley on either side.
The servo motor must keep the belt engaged with the pulley for 5 +/- 1 second.	● Activate the servo motor and turn it 180° to the engaged position. ● Stop feeding power to the motor and start a timer. ● Make sure the timer reaches between 4-6 seconds before feeding power back to the servo motor and allowing another 180° rotation.
The servo motor should be receiving between 4.8 and 6.0 volts of power while activated.	● Engage the servo motor to turn 180° to a set position. ● While the motor is turning, use a voltmeter to measure the power being supplied to the motor during this process. ● Make sure the power reading stays within 4.8 volts and 6.0 volts the whole time.

2.3.3. Knob-Pulley Subsystem:

The pulley subsystem is responsible for the actual rotation of the stove knob. This system will function with the use of a belt, pulley, and a stepper motor. The pulley subsystem is only active when the tensioner subsystem has removed all slack between the pulley and the belt, and when the user initiates the stepper motor via control on the mobile application. When the belt is taught and the user has specified the stove setting they wish to adjust to, the stepper motor will begin to drive the belt along the pulley, which is attached directly to the shaft that the stove knob is on. As the pulley begins to rotate, the shaft will as well, which in turn causes the stove knob to turn to the user's desired position. The adjustment period should be relatively brief, taking roughly 5, +/- 1 second to reach the desired position. After the knob has been rotated accordingly, the stepper motor stops driving the belt along the pulley and the tensioner subsystem disengages the belt from the pulley until notified otherwise by the user. The stepper motor we have selected is rated for 3 volts of power, so we will use a voltage transformer to convert 5V of power to 3V to power the motor.

Requirements	**Verification**
Stove knob should be able to reach the desired position in 5, +/- 1 second(s).	● Select a specific setting for the stove on the StoveSense app. ● Start a timer as soon as it has been selected. ● Stop the timer once the stove knob stops turning. ● Make sure the recorded time is within 4-6 seconds.
Stepper motor should be getting at least 3 volts of power throughout the duration of a knob adjustment.	● Select a specific setting for the stove on the StoveSense app. ● Use a voltmeter to measure the power being supplied to the stepper motor during the rotation period. ● Make sure the voltage reading does not drop below 3 volts in that duration.

2.3.4. Control Subsystem:

Our control system will primarily be run through a microcontroller. We decided to go with an ESP32 microcontroller unit with Wi-Fi enabled communication. The primary communication protocol that we decided to go with is a web socket connection between the board and our app. By establishing a web socket connection we can ensure bidirectional communication between our app and the microcontroller. The microcontroller will be relaying information regarding whether the stove is on or not and the temperature of the liquid on the stove to the app. The ESP32 microcontroller can only take between 2.2V to 3.6V so we will need a voltage stepper sitting between the power supply and microcontroller to step down the voltage from 5V to 3.3V. The schematic below does not include this voltage stepper. Additionally, the microcontroller will consume information from the LM35 Temperature Sensor and a Thermistor and then send them to the Remote Control Subsystem. Additionally, the ESP32 MCU will be responsible for receiving information from the rotary encoder to determine the position of the stove knob. Furthermore, the microcontroller will be responsible for relaying information to the motors based on the Remote Control Subsystem. The two motors that are going to be controlled are stepper and servo motors. The stepper motor will be responsible for tensioning the belt in the pulley system, while the servo motor will be the driving force for the pulley system.

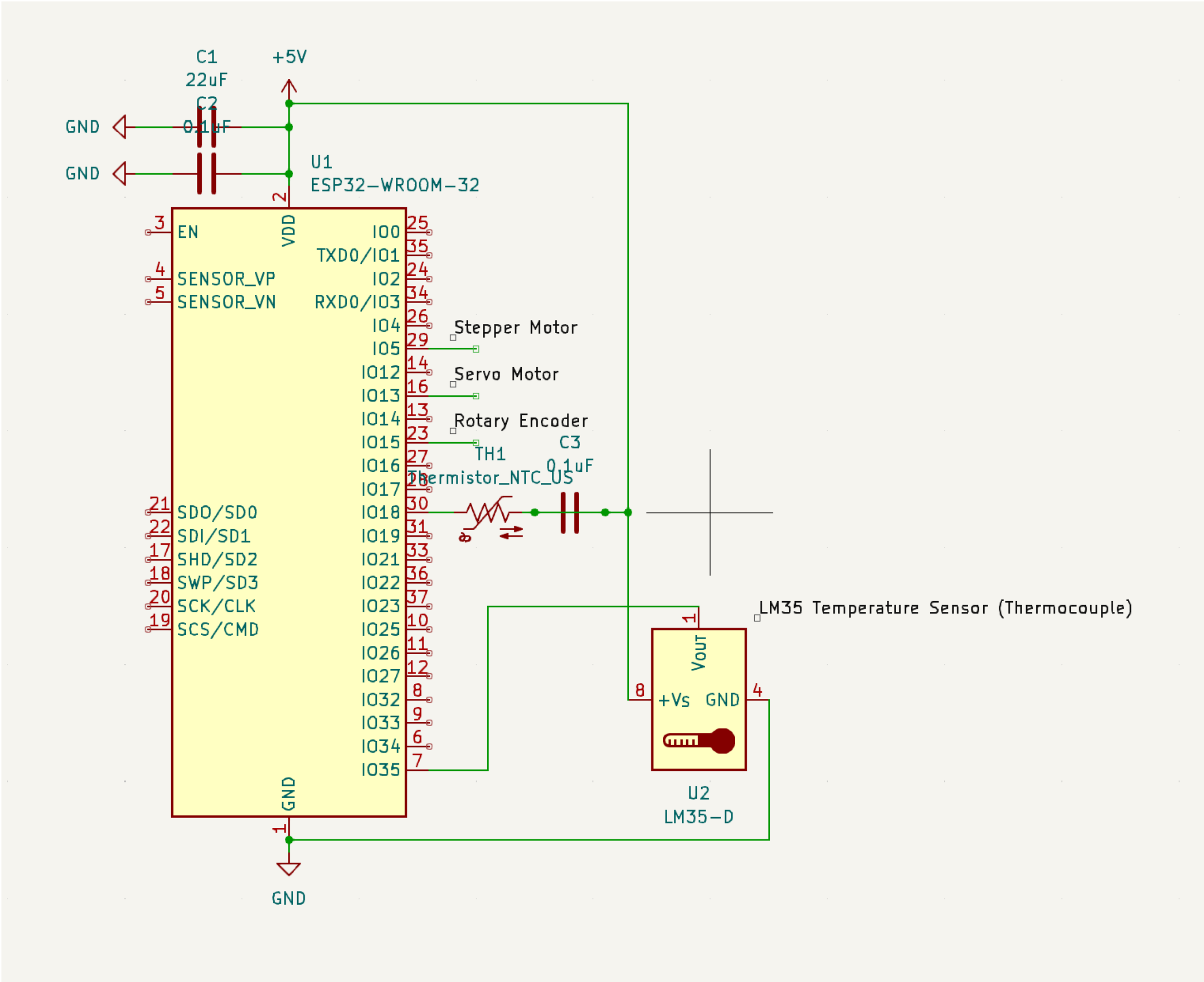

Figure 6: Schematic of the ESP32 microcontroller and LM35 Thermocouple

Requirements	Verification
The microcontroller should be able to give an accurate temperature reading from the thermistor to the backend server within 5sec +/- 1sec.	• Open the StoveSense app on the device. • Turn the stove on the knob. Start a timer on the stopwatch. • The temperature reading on the app should be updated within 6 seconds.
The microcontroller should be able to start the tensioner and knob-turning process within 5sec +/- 1sec based on the user's input.	• Open the StoveSense app on the device. • Select a command to update a stove knob's position. • The tensioner's servo motor should begin rotating within 6 seconds.

The microcontroller should be taking in 3.3V of input.	● Use a voltmeter to measure the voltage across Vdd to ensure that voltage is at 3.3V.

2.3.5. Remote Control Subsystem:

Our mobile application will be primarily controlled through a backend server. The backend server will be responsible for facilitating communication between the ESP32 MCU and the app itself. The backend server should be able to properly take in user's input and relay them to the app. We will set up a back-end stack using Python's Flask library, and connect it to a MySQL server via MySQL Connection Cursor. The MySQL database will be used to store all of the temperature data along with data from the rotary encoder, so that the microcontroller can process the data and decide how to operate the pulley system accordingly. The front-end of the app will display a knob that is supposed to mirror the knob on the physical stove. Additionally, if the LM35 is in use the app should display the temperature of the liquid that is currently on the stove. The primary input that our app should take is allowing the user to rotate a specific knob to any level that it desires. The user will not be able to turn on the stove remotely, but will be able to decrease and turn off the stove via the app. We envision the user rotating a virtual knob on our app that will correspond to the knob on the stove. When the user is freely using the stove (without the use of the tensioner or pulley subsystems), the application should reflect any changes to the stove setting on the virtual mirrored knob.

Requirements	**Verification**
Turning the knob to a desired setting on the app is reflected on the physical stove top as well.	● Use the StoveSense mobile application to turn the virtual stove knob to various positions. ● Watch the physical stove knob and make sure it also moves to the same position as indicated on the app.
When the LM35 is dropped into a pot of liquid, the app should be	● Place the LM35 sensor into a pot of liquid that is

displaying the temperature of that liquid.	warming up. ● Open the StoveSense app and make sure a temperature is being displayed and that it is rising on the app.
The app should reflect what level the stove knob is currently at, even when the user manually turns the knob.	● Turn the stove knob by hand to a specific level. ● Open the StoveSense app and make sure that the virtual stove knob is displaying that it is at that same level. ● Adjust the level of the stove knob by hand again and ensure that the StoveSense app changes the level of the virtual knob to the same level.

2.3.6. Power Subsystem:

Our power system will simply consist of a power supply. This power supply will deliver power from an outlet to a microcontroller. Since we are using voltage from a wall which is AC we will need to use a rectifier to convert AC to DC. Typically, voltage from a wall is around 120 V so there will need to be a rectifier that converts this 120 AC voltage to DC. Our power supply will deliver 5V to a voltage stepper which should bring down the voltage to 3.3V for the ESP32 MCU. The power supply will need to deliver 5V to the LM35 Waterproof sensors and the motors. In order to ensure that we are getting 5V from the power supply we will need to use a voltage regulator. This voltage regulator will bring down a higher voltage to a lower one and ensure that we are dealing with a known voltage. The figure below depicts how we can use a power supply and voltage regulator to appropriately bring down the voltage to 5V.

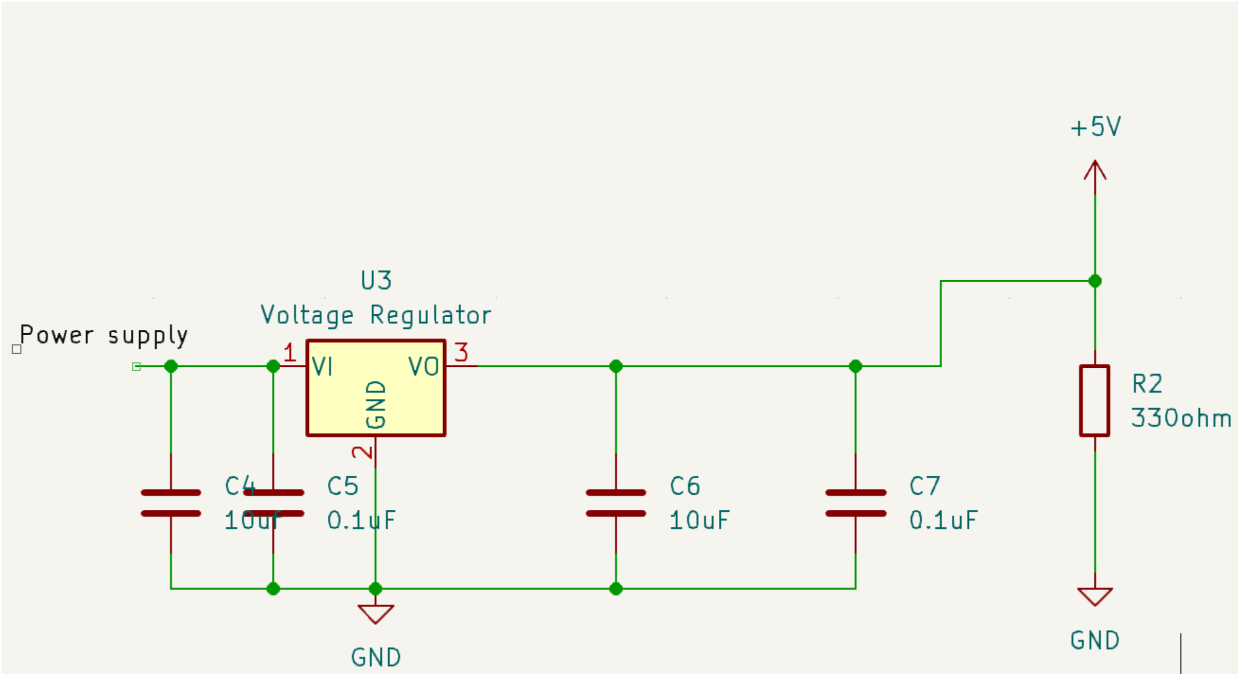

Figure 7: Schematic of voltage regulator to drop input voltage to 5 volts

We can use a voltage divider circuit which takes the input from the voltage regulator to step the voltage down from 5V down to 3.3V as seen below.

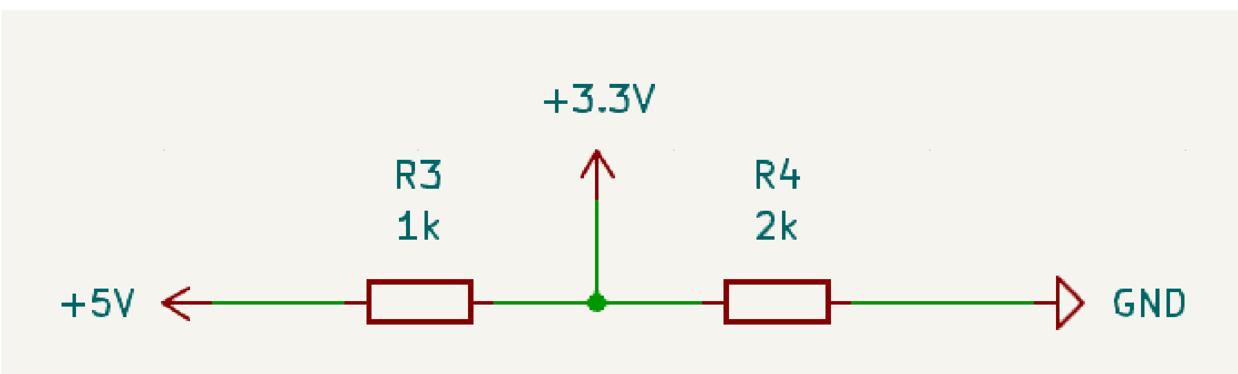

Figure 8: Schematic of voltage divider to drop 5 volts to 3.3 volts

Requirements	**Verification**
The voltage coming from the power supply and through the voltage regulator must be 5V.	• We can measure the voltage coming from the voltage regulator by using a voltmeter. • The voltmeter must measure 5V for the power supply to be functional

The voltage coming out of the voltage divider circuit must be 3.3V.	• We can measure the voltage coming from the voltage divider circuit by using a voltmeter. • The voltmeter must measure 3.3V for the voltage divider circuit to be valid
The voltage regulator must take in more than 2V over 5V for it to regulate the voltage down to 5V. This means that the voltage coming from the power supply must be greater or equal to 7V. It also needs the voltage to be less than 25V for it to be functional.	• Use a voltmeter to measure the voltage coming out of the power supply. • Confirm that the voltage coming from the power supply is between 5V to 7V.

2.4. Tolerance Analysis:

Required Torque for Motor in Pulley System:

Stove knobs are designed to provide mechanical advantage, which makes them easier to turn, opposed to directly rotating the shaft. We can also achieve a mechanical advantage with our pulley system. In our case, we have two fixed pulleys having force exerted on them via a stepper motor. This would allow us to have a mechanical advantage of 2 [10]. Essentially, we can use half the force necessary to turn the shaft compared to a knob of a similar size. The problem is, how do we estimate the amount of torque needed to rotate the shaft? An average plastic stove knob is between 1-3 oz. Assuming the higher end, to rotate a 3 oz knob with a 1 inch diameter we will need roughly 3oz-in of torque. Using our pulley system we only need 1.5oz-in of torque due to the mechanical advantage. The stepper motor we are utilizing has a holding torque of 67.97 oz-in and detent torque of 3.11 oz-in accordance with the datasheet [11]. To calculate the torque at low speeds, we can use this equation [12]:

$$Holding\ Torque\ -\ 2(Detent\ Torque)\ =\ Available\ Torque$$

This gives us an available torque of 61.742 oz-in, which is significantly more than the force to turn the knob, leaving us with ample torque to overcome friction.

3. Cost and Schedule

3.1. Cost Analysis:

Labor:

The hourly wage for a research project is estimated to be $50 an hour, and our total time commitment is 12 hours/week for 12 weeks. We estimate a labor cost of $21,600 per person. Therefore, our labor cost is estimated to be $64800. This is broken out in table X. Additionally, we will be utilizing the machine shop, thus we also need to account for their labor costs.

Table 8: Labor Cost

Name	Weekly Hours	Hourly Pay	Scaling Factor	Weeks	Cost (USD)
Aryan	12	50	2.5	12	21,600
Nikil	12	50	2.5	12	21,600
Dinal	12	50	2.5	12	21,600
Machine Shop	20	56.12	1	1	1122.4
Total					**$65,922.40**

Parts:

The cost of our parts are also a significant percentage in our total costs. These costs are directly pulled from Digikey if available or the cheapest viable option. Pricing will include shipping if necessary.

Table 9: Parts Cost

Part Name or Number	Quantity	Cost (USD)	Link
LM35DZ Waterproof Thermocouple	1	3.66	Link
Stepper Motor - 68 oz.in (400 steps/rev)	1	19.50	Link

Servo - Generic High Torque (Standard Size)	1	13.95	Link
ESP32-WROOM-32 (16MB) Microcontroller	1	4.50	Link
RL1004-65.6-59-D1 Thermistor	1	2.65	Link
QFWC-05-05 Power Supply	1	4.82	Link
EN11-VSM1AF20 Rotary Encoder	1	2.84	Link
RC0402FR-071KL Resistors	30	3.00	Link
A4988 Stepper Motor Driver	1	6.95	Link
AC DC Converter Module Universal 110V 120V 220V 230V to DC 5V 12V	1	11.99	Link
Total			**$73.86**

3.2. Schedule:

Week	Task	Assignee(s)
October 2 - 6	Begin PCB design, attend design reviews with TA, start ordering parts	All
October 9 - 13	Wrap up initial PCB design	Nikil

	Work on creating app + backend	Aryan + Dinal
	Read up and practice coding with ESP32, First PCBway order, Teamwork Evaluation 1	All
October 16 - 20	Wrap up remaining orders, Second round PCBway order	All
	Start designing the motor + pulley system	Nikil + Dinal
	Have ESP32 communicate with Backend via WebSocket	Aryan
October 23 - 27	Wire sensors to ESP32 and send signals to Backend	Aryan + Nikil
	Finish designing motor + pulley system	Dinal
October 30 - Nov 10	Work on PCB board, wire everything up and unit test each component	All
	Integration test all components together	
Nov 13 - 24	Mock demo + fix issues	All
Nov 27 - Dec 1	Final demo	All
	Work on presentation	
Dec 4th - Dec 8th	Final presentation + wrap up loose ends	All

4. Ethics and Safety:

The main safety risk associated with this project is a potential malfunction of the stove knob turner. If this component were to malfunction and turn the heat up too high it could potentially cause a fire. Additionally, having remote control of the stove's level could pose a risk to safety as the user might accidentally turn the stove up too high or even have someone steal their device and turn it up on purpose. One way to remedy this problem is to require the user's face ID or fingerprint ID on their mobile device to confirm that they want to operate the stove. This would make it difficult for someone to forcibly mess with the stove and also for the user to accidentally interfere when they don't mean to. An issue that could arise during the development of this project is if the boil-over protection feature were to fail and cause liquid from the boiling pot to spill across the stove top. Since we will be using a wall outlet to provide power to this system, we need to be able to step-down the typical 120V AC current to a manageable 5V of DC current. Doing so requires four major components; a step-down transformer, a full-wave rectifier, a filter to smooth out the current, and a voltage regulator. The step-down transformer will drop the 120V from the wall to about 6V-12V. The full-wave rectifier will then convert the AC voltage to a pulsating DC voltage. The filter is a capacitor that smoothes out the pulsating DC voltage into a continuous DC voltage. Finally, the voltage regulator will be used to bring the 6V-12V down to a consistent 5V to power the PCB and external components [9].

The potential risk with the boil-over protection feature can be remedied by keeping a close watch on the pot while testing this feature and ensuring that it never reaches this critical point. Section 1.2 of the ACM Code of Ethics states that unjustified damage to property should be avoided [5]. This is an ethical issue that we must also take into account, as our product has the capability to severely damage property if misused or malfunctioned. This ethical breach can be mitigated by holding the leaders accountable for knowing when to "pull the plug" on the operation. "If leaders do not act to curtail or mitigate such risks, it may be necessary to "blow the whistle" to reduce potential harm" [5]. Section 1.5 of the ACM Code of Ethics states to "Respect the work required to produce new ideas, inventions, creative works, and computing artifacts." [5] Smart stove knobs already exist as a product and we will respect this invention by crediting the creators and also identifying how our product differs from the existing ones.

5. Citations:

[1] "ESP32 Wireless Communication Protocols | Random Nerd Tutorials," Oct. 28, 2022. https://randomnerdtutorials.com/esp32-wireless-communication-protocols/#wifi

[2] "CALLIOPE | Measuring the temperature of liquids," *calliope.cc*. https://calliope.cc/en/examples/temperatur-von-fluessigkeiten-messen (accessed Sep. 15, 2023).

[3] "Torque Calculator," *www.omnicalculator.com*. https://www.omnicalculator.com/physics/torque#:~:text=Measure%20the%20distance%2C%20r%20%2C%20between

[4] IEEE, "IEEE Code of Ethics," *ieee.org*, 2020. https://www.ieee.org/about/corporate/governance/p7-8.html

[5] ACM, "ACM Code of Ethics and Professional Conduct," *Association for Computing Machinery*, Jun. 22, 2018. https://www.acm.org/code-of-ethics

[6] "ESP32 - Rotary Encoder," ESP32 Tutorial. https://esp32io.com/tutorials/esp32-rotary-encoder (accessed Sep. 27, 2023).

[7] "ESP32 - LM35 Temperature Sensor," ESP32 Tutorial. https://esp32io.com/tutorials/esp32-lm35-temperature-sensor

[8] Dynapar, "Incremental Encoders | Fast Lead Times | Dynapar," www.dynapar.com. https://www.dynapar.com/products_and_solutions/rotary-encoders/incremental/ (accessed Sep. 27, 2023).

[9] "Bridge Rectifier Circuit Theory with Working Operation," ElProCus - Electronic Projects for Engineering Students, Nov. 09, 2018. https://www.elprocus.com/bridge-rectifier-circuit-theory-with-working-operation/

[10] "Powerful Pulleys - Lesson," *TeachEngineering.org*, Jun. 30, 2019. https://www.teachengineering.org/lessons/view/cub_simple_lesson05

[11] "Stepper Motor - 68 oz.in (400 steps/rev) - ROB-10846 - SparkFun Electronics," *www.sparkfun.com*. https://www.sparkfun.com/products/10846 (accessed Sep. 27, 2023).

[12] "What's the holding torque and detent torque of the stepper motor? – AKT Motor and Drive." https://aktmotor.com/whats-the-holding-torque-and-detent-torque-of-the-stepper-motor/

[13] "How to Build a 5 Volt Power Supply | Jaycon Systems," Jaycon Systems | Product Design, PCB & Injection Molding, Aug. 26, 2013. https://jayconsystems.com/blog/basic-5v-power-supply#:~:text=For%20proper%20operation%2C%20we%20need (accessed Sep. 27, 2023).

[14] "How to Level Shift 5V to 3.3V | Random Nerd Tutorials," Sep. 24, 2015. https://randomnerdtutorials.com/how-to-level-shift-5v-to-3-3v/

[15] “Analog to Digital Converter (ADC) - ESP32 - — ESP-IDF Programming Guide v4.4 documentation,” docs.espressif.com. https://docs.espressif.com/projects/esp-idf/en/v4.4/esp32/api-reference/peripherals/adc.html

[16] “ESP32WROOM32E & ESP32WROOM32UE Datasheet.” Available: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf

[17] “ESP32 Series Datasheet Including.” Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[18] “ESP32: K-Type Thermocouple with MAX6675 Amplifier | Random Nerd Tutorials,” Mar. 15, 2022. https://randomnerdtutorials.com/esp32-k-type-thermocouple-max6675/ (accessed Sep. 27, 2023).