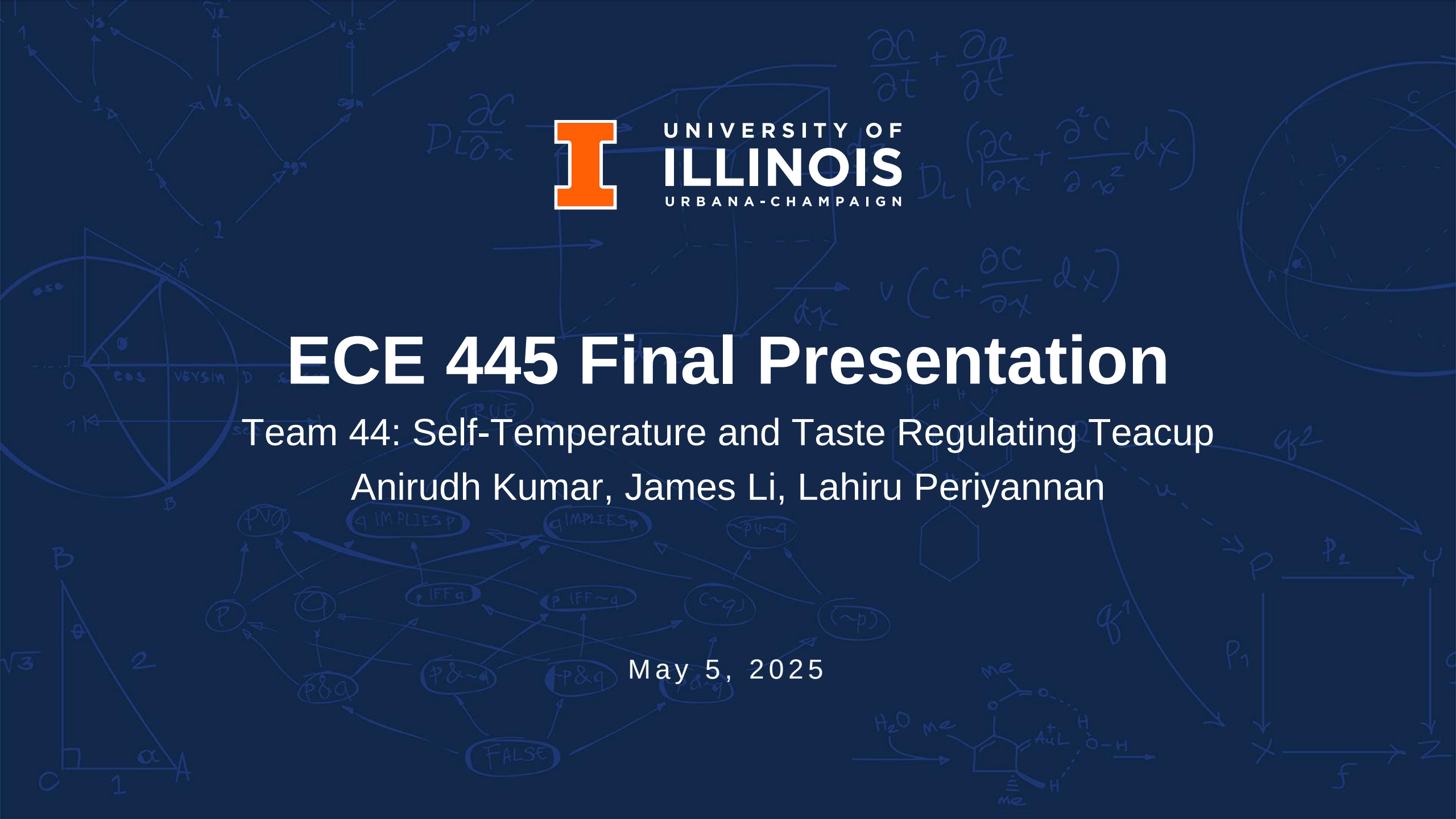


UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

ECE 445 Final Presentation

Team 44: Self-Temperature and Taste Regulating Teacup

Anirudh Kumar, James Li, Lahiru Periyannan

May 5, 2025

Introduction

- A cup that could heat water to the optimal temperature for a given tea leaf for best flavor and aroma extraction

Popularity of Tea

- US market: $25.1 Billion in 2024
- Around 80% US households drink tea
- Green tea lowers blood pressure and cholesterol

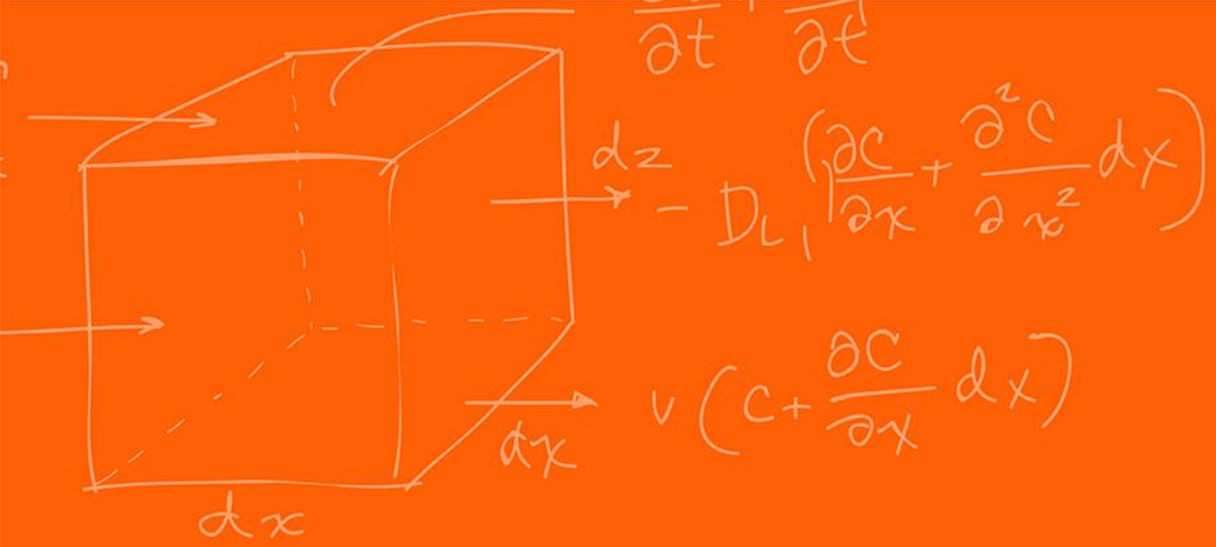
Why our Product?

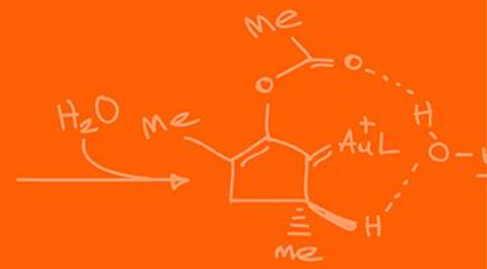
- Precise, consistent temperature control
- Unlocks maximum flavor and aroma from any tea leaf

High Level Requirements

We believe our project must display these high-level characteristics to successfully solve the problem

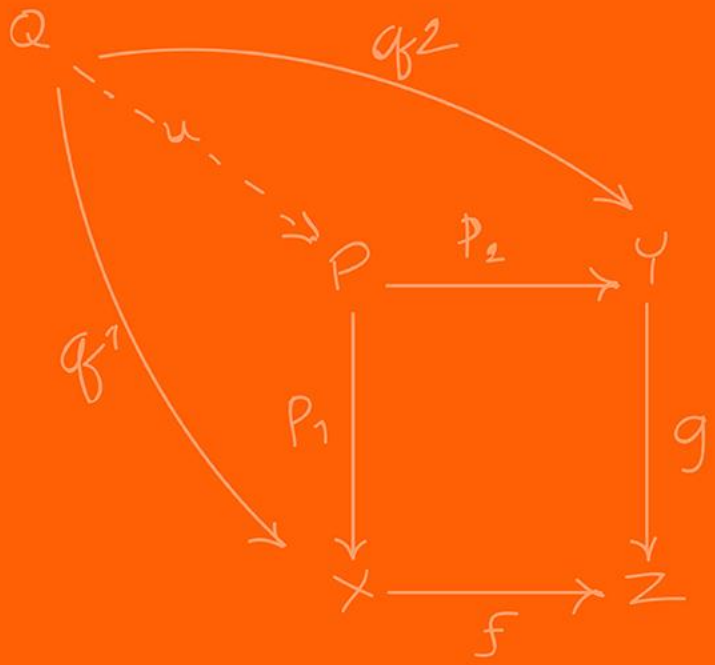
- The cup is intended for daily usage, so the desired temperature is to be maintained for at least six hours.
- The cup should be able to maintain the temperature with 1-2°C of the desired user temperature
- Ideally the cup should not be too heavy; according to KitchenJournal, empty water bottles tend to weigh around 0.3 kg.



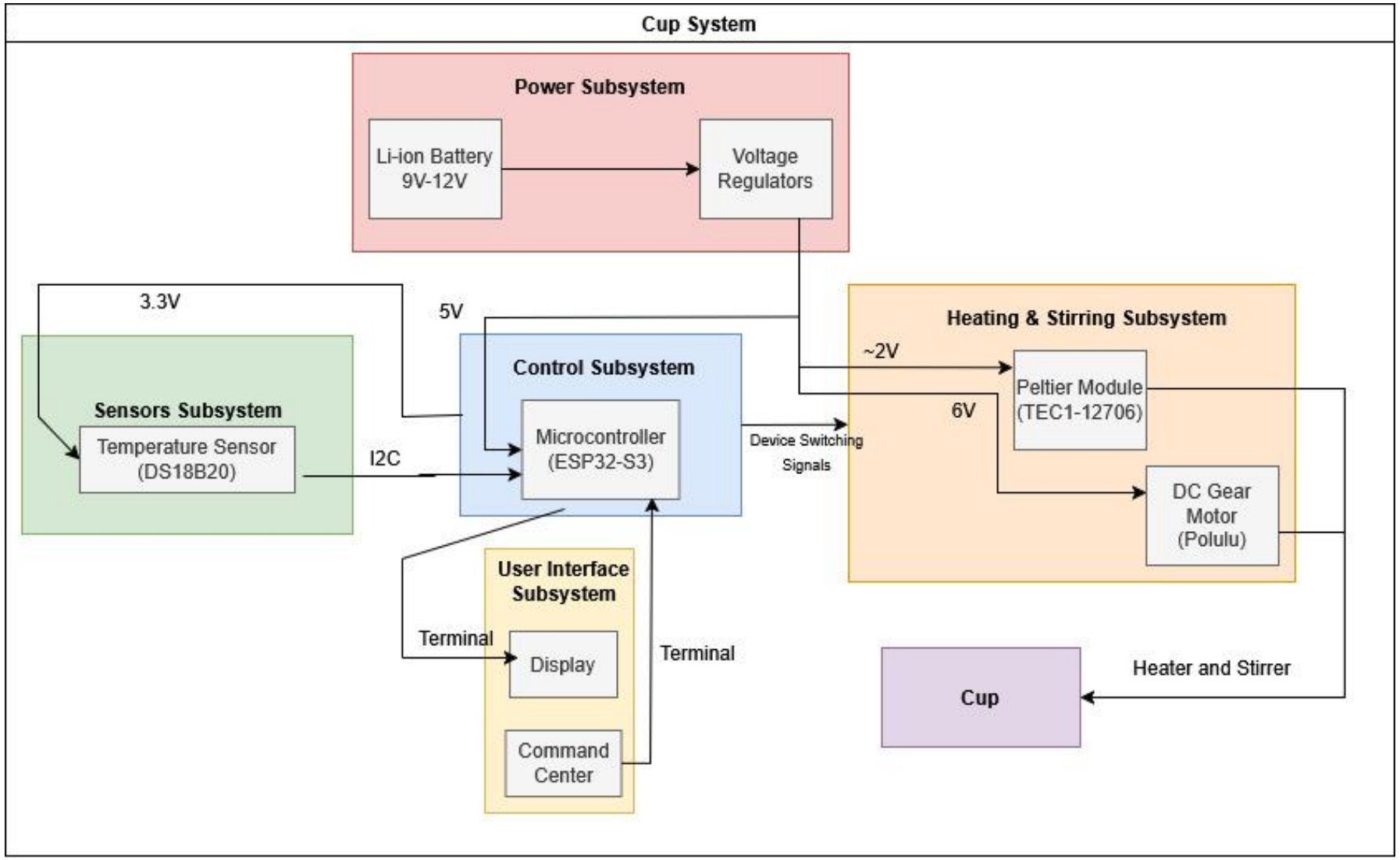


Design

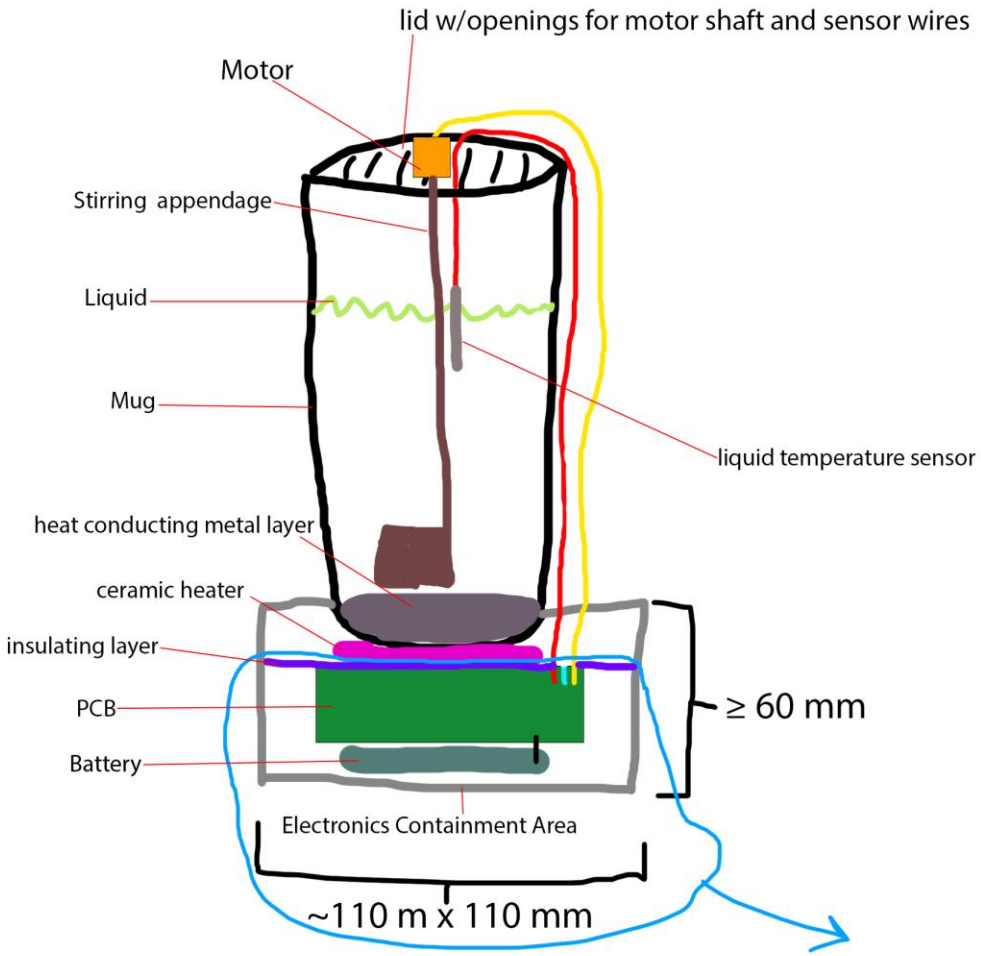
Discussion of the Various Subsystems



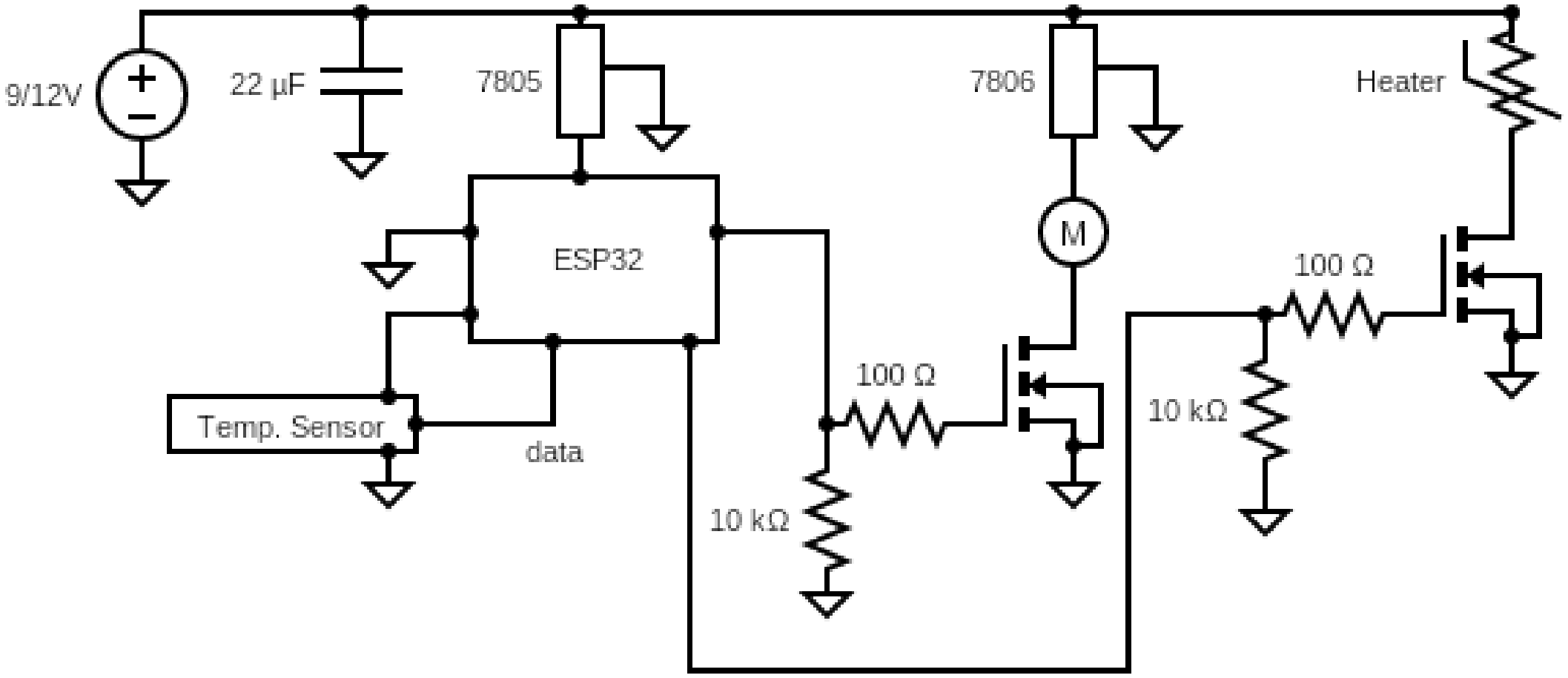
Block Diagram of the System



Hand-drawn Schematic vs Physical Result



Circuit

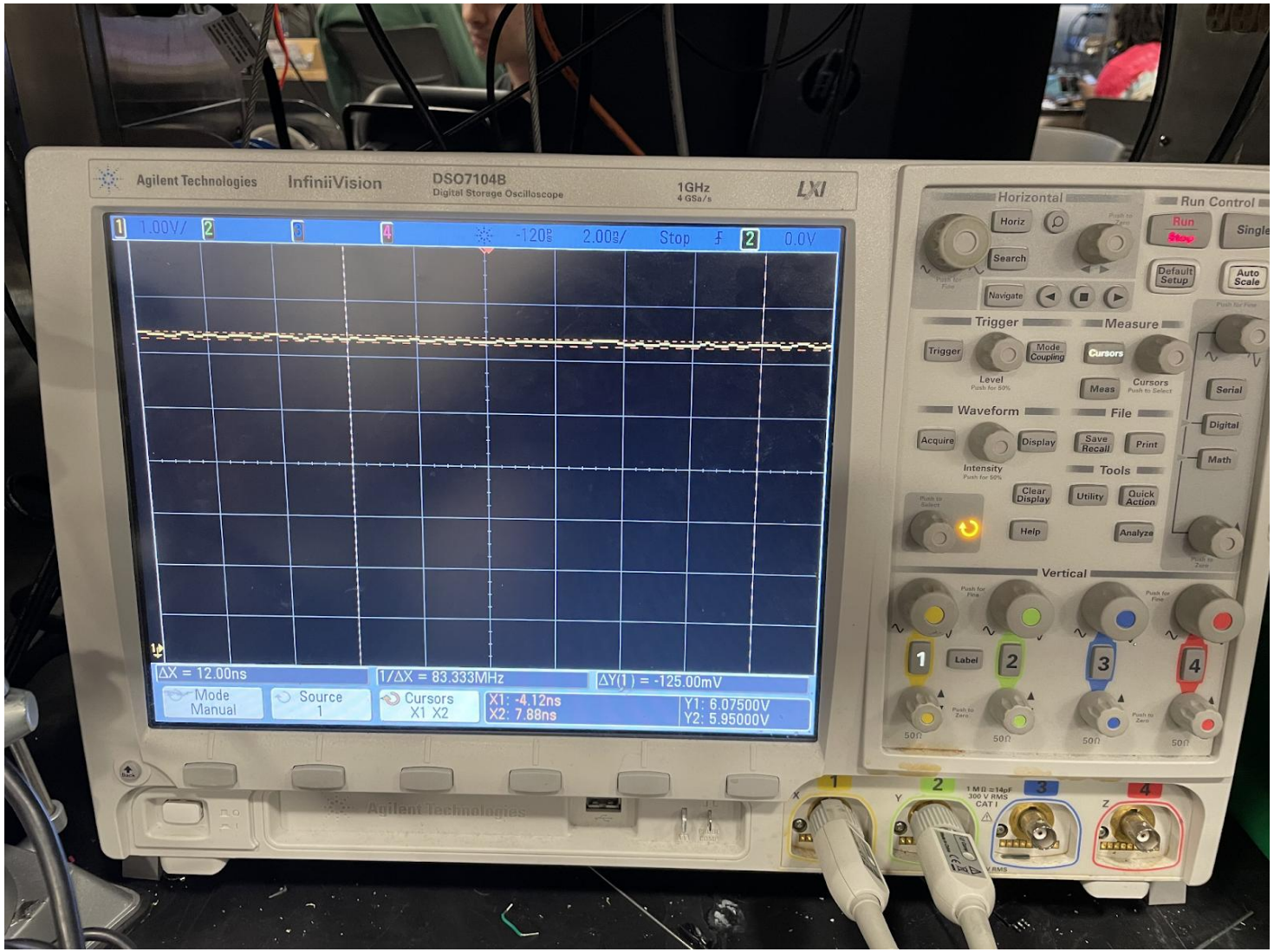

9/12V
22 µF
7805
7806
Heater
ESP32
M
100 Ω
100 Ω
Temp. Sensor
data
10 kΩ
10 kΩ

Subsystem #1: Power

This subsystem is responsible for supplying power to the sensors, motor, heating element, and ESP32

- 9V and 12V (8/16 AA) battery options available.
- Tradeoffs in efficiency, longevity, and safety
  - PCB current support
- 7805 and 7806 voltage regulators used to drop voltage to motor and ESP32's respective levels, tested for ripple/tolerance.
- Sensor receives power from ESP32 3.3V pins

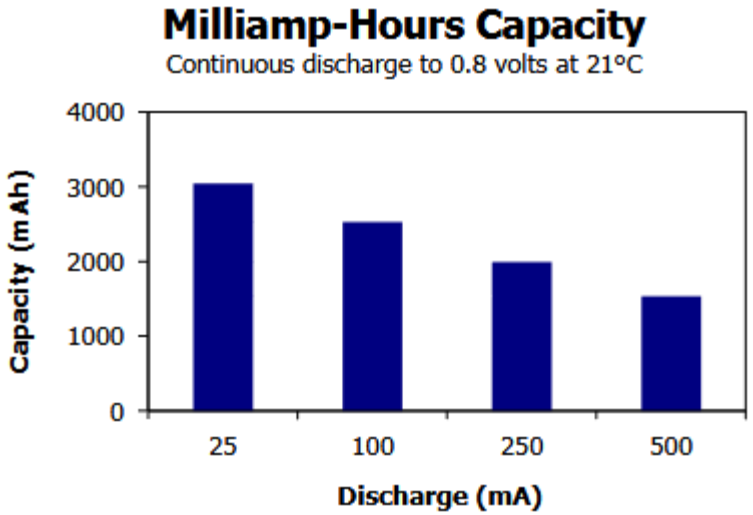
Voltage Regulator Verification



- 125 mV ripple
- ~2.1% ripple
- Tolerance of 5%

Battery Life Verification

- ~500 mA avg current draw when actively heating
- 1600 mAh at avg. current
- Double battery pack option offers longest longevity, but weight + safety risks
- Ideally supplied with rechargeable Lithium-ion battery



Subsystem #2: Control and Communication

This subsystem acts as a bridge between the user interface subsystem and the other subsystems.

- ESP32-S3-WROOM microcontroller collects data from the sensors, to provide the user and heating control
- Based on the sensor data and the user-provided settings, it then gives instructions to the other subsystems
- The microcontroller controls signals to different devices' MOSFET switches, i.e. the Peltier module or the motor

Requirements and Verification to Ensure Prompt Directions

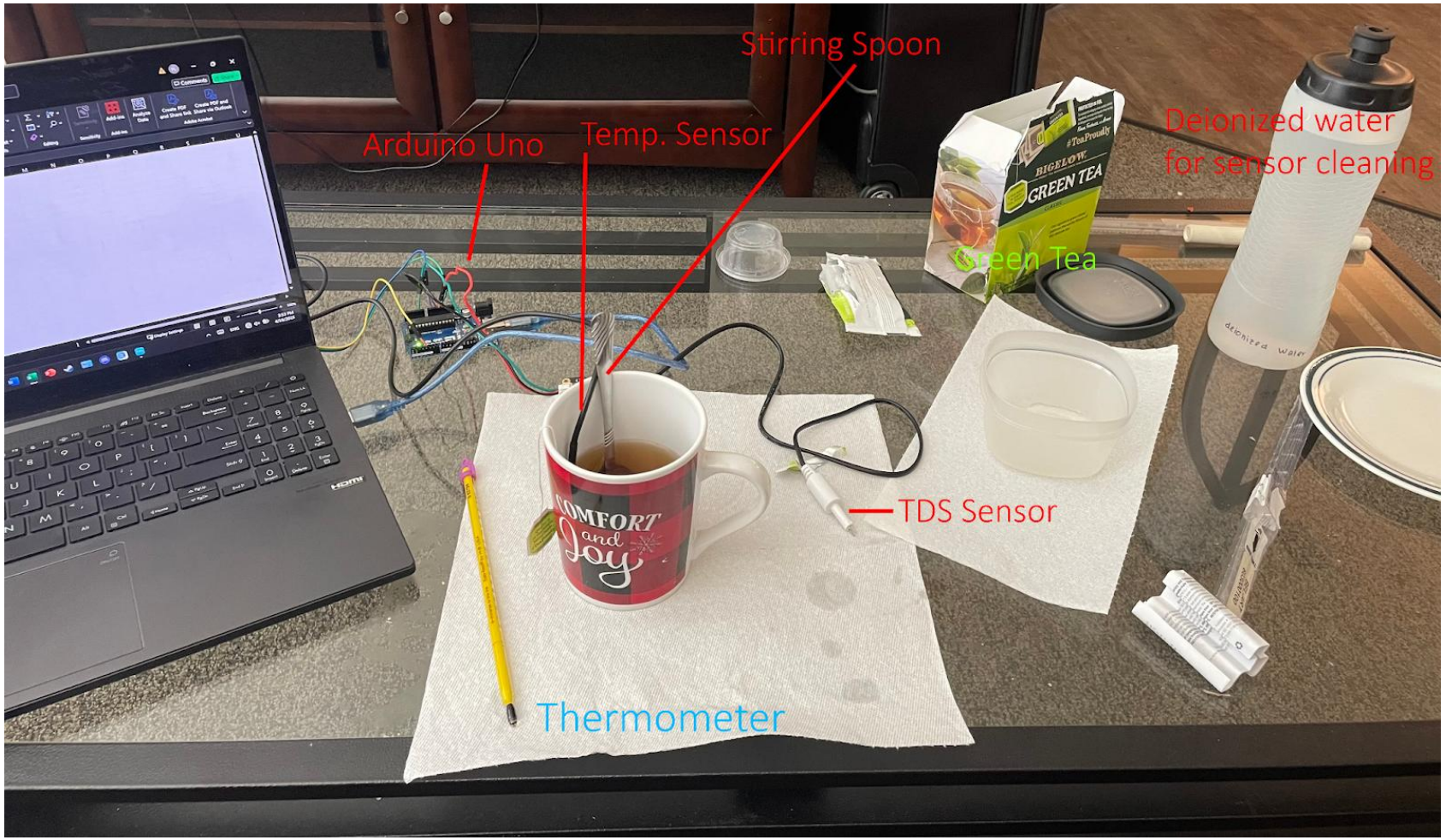
- The heating device should be turned on and off as needed, depending on the water temperature, which would be measured regularly
- A thermometer was used to verify this functionality
- The stirring should be done at a regular interval, which only turns off when the user sends that command; it should continue regardless of the status of the connection between the user and the microcontroller

Subsystem #3: Sensors

The sensor subsystem is responsible for reading the necessary data from the water to maintain tea quality and temperature.

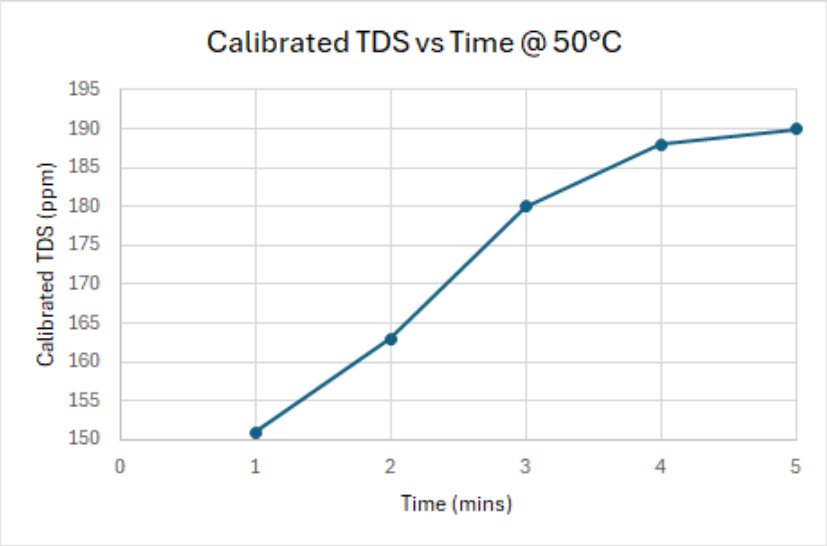
- Temperature sensor needs to be within 1-2°C accuracy to ensure temperature setpoints are met
- Total Dissolved Solids sensor used in external experimental testing to determine tea concentration in ppm
  - Operates as a conductivity sensor, effectively sensing ions introduced by tea diffusion
- Light weight and low power consumption sensors such as DS18B20 to minimize cup weight and battery depletion

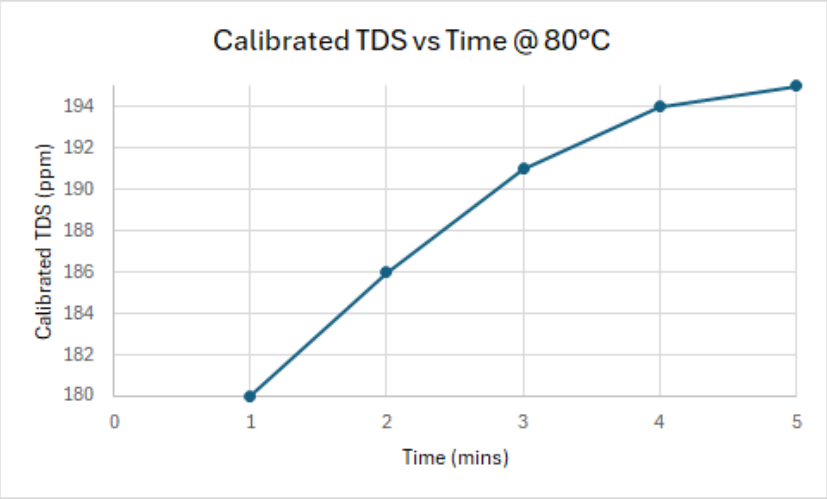
Sensor and Tea Data Collection



Experimental setup for testing sensors and collecting tea diffusion data

Test Results





- One tea bag per 250 mL
- Used Total Dissolved Solids sensor to measure tea particle diffusion levels
- 217 innate tap water TDS
- Used for tea strength feedback

Temperature Sensor Measurement	Thermometer Measurement
78.5°C	79.5°C
25.94°C	25°C

Subsystem #4: User Interface

This subsystem lets the user change the settings and also receive information about tea temperature and strength

- We use computer terminal for the user interaction
- The user can monitor temperature and tea strength (TDS values were mapped to qualitative descriptors) in real time
- Can set a desired heating temperature as well as manually turn the stirring element on or off

Initial UI with Bluetooth

- Initially wrote code using Espressidf framework
- Had to incorporate Arduino framework to use the DS18B20 temperature sensor (using Platformio)

```
if "arduino" in frameworks:
    # Downgrade the IDF version for mixed Arduino+IDF projects
    self.packages["framework-espidf"]["version"] = "~3.40407.0"
    # Delete the latest toolchain packages from config
    self.packages.pop("toolchain-xtensa-esp-elf", None)
```

Switching to the Laptop Terminal

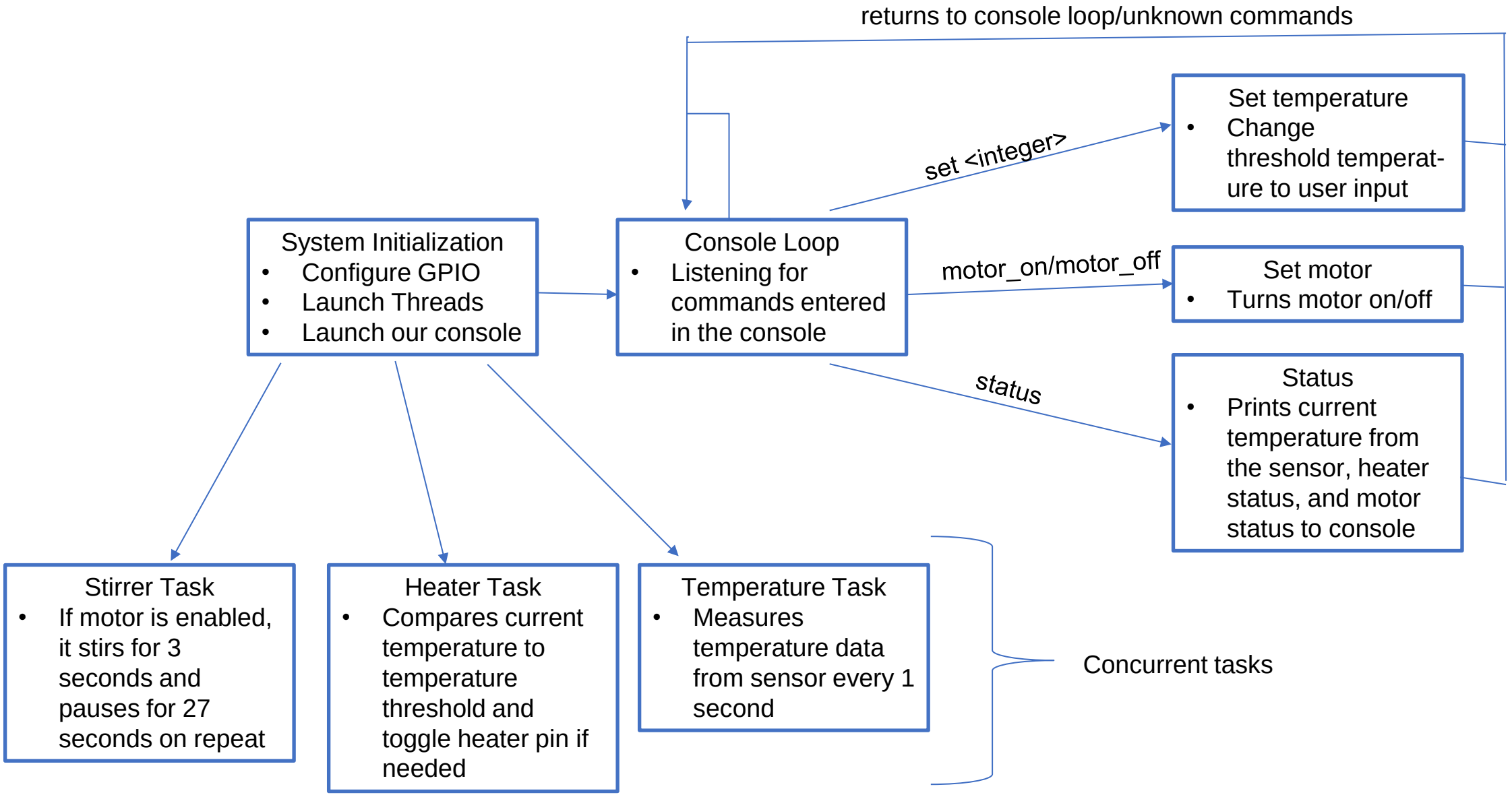
- Does not require advanced versions of Espressidf
- Can perform exact same operations as with the original Bluetooth-based user interface

```
T = 24.94 °C, set-point = 25.00 °C, heater ON, motor ON
Elapsed Time: 128 sec  |  Strength: Mild
cmd>
cmd> status
T = 25.00 °C, set-point = 25.00 °C, heater ON, motor ON
Elapsed Time: 149 sec  |  Strength: Mild
cmd>
cmd> status
T = 25.00 °C, set-point = 25.00 °C, heater ON, motor ON
Elapsed Time: 167 sec  |  Strength: Medium
cmd>
cmd> status
T = 24.94 °C, set-point = 25.00 °C, heater ON, motor ON
Elapsed Time: 253 sec  |  Strength: Strong
```

Requirements and Verification to Ensure Real-Time Communication

- Ensure that the water temperature matches the current temperature in the cup (used a thermometer again for this)
- When the user sets a new temperature, the heater should respond immediately (checked by setting a higher and lower desired temp)
- Notify the user when the tea is finished brewing (shown in the status messages in the laptop terminal)

Code Flow


returns to console loop/unknown commands
System Initialization
• Configure GPIO
• Launch Threads
• Launch our console
Console Loop
• Listening for commands entered in the console
set <integer>
Set temperature
• Change threshold temperat-ure to user input
motor_on/motor_off
Set motor
• Turns motor on/off
status
Status
• Prints current temperature from the sensor, heater status, and motor status to console
Stirrer Task
• If motor is enabled, it stirs for 3 seconds and pauses for 27 seconds on repeat
Heater Task
• Compares current temperature to temperature threshold and toggle heater pin if needed
Temperature Task
• Measures temperature data from sensor every 1 second
Concurrent tasks

Subsystem 5: Heating and Stirring

- **This subsystem maintains the tea at the desired user temperature.**

The Peltier heating module. Lower voltages, higher current, high power consumption

The metal-ceramic ring heater. Very high voltage, medium current, similar power consumption

The motor placed at the lid. It is attached to an appendage that goes into the liquid and stirs it.

Requirements and Verification to Ensure Correct Stirring and Heating

- The heating subsystem should start and stop within a few seconds of receiving those requests from the control subsystem
- A thermometer was used to continuously measure the water and ensure the Peltier module was working as expected
- The purpose the stirring device was used to ensure uniform temperature and concentration distribution

Subsystem 6: Cup

- Cup needed to be less than 500g
  - When measured without liquid, 260g
- Outside surface temperature should be safe to hold
  - Double steel layer + handle
- Electronics should not be at risk of contacting liquid

Ethical and Safety Considerations

- Warning labels near heating elements
- Safe liquid handling
- Better electrical protection from liquids
- Internet of Things considerations

Successes and Failures

- Learned PCB design, soldering and microcontroller operation
- Built a functional cup that can brew tea as the user likes it
- Had a good method for UI but were unable to implement Bluetooth
- Short circuited microcontroller + ring heater and had to quickly adapt

Future Steps

- Implement Bluetooth for user communication
- Make a more robust base to store all the electronics so that the cup can be more easily handheld
- Enhance the heating efficiency
- Improve the aesthetic look of the device and UI

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN