Task 1 – Introduction

Pre Setup

Alert: you must do this first.

Many Arduino has previously uploaded code. We never know who uploaded what code on it.

Connect USB to Arduino without any jump wires. Long-press RESET.

Run this code first to erase the previously uploaded code.

After completion, unplug the USB.

Arduino Code

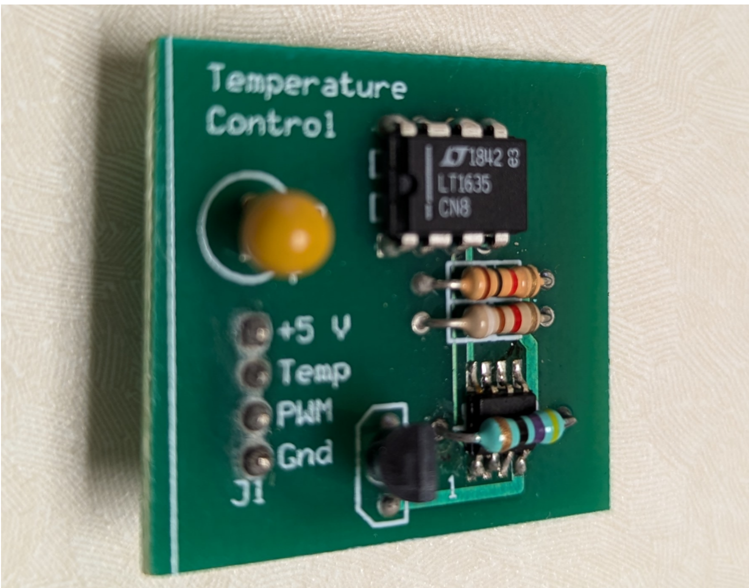
```
const int PWMPin = 10;
// Suppose use Pin 10.
void setup() {
  analogWrite(PWMPin, 0);
}

void loop() {
}
```

Hardware Introduction

In this lab, we will study a resistor-based heater system.

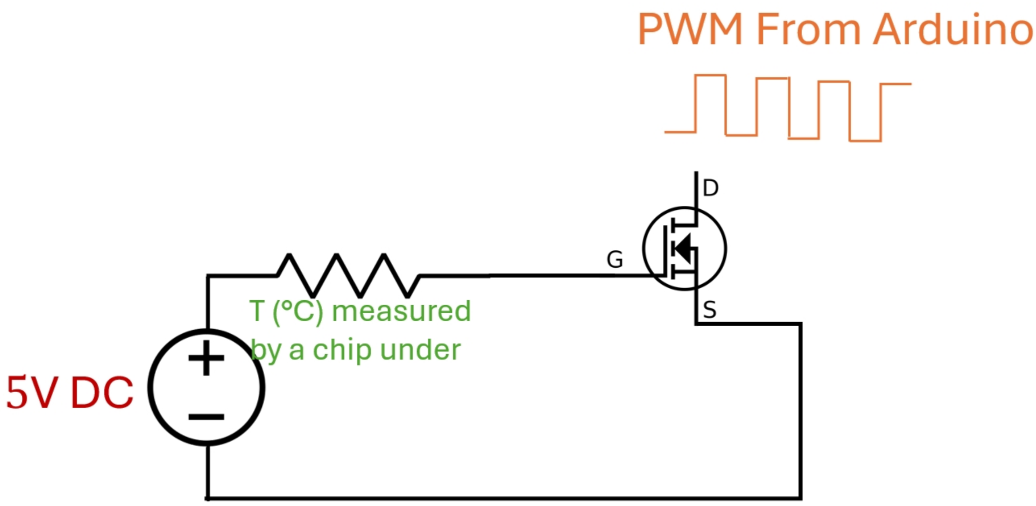
The actual system is this board:



- The green resistor is the load to be heated.
- The chip under the green resistor is a temperature sensor.
- The half-cylinder item is a 2N7000 MOSFET and acts as a logic switch. The MOSFET will be fully turned on if a gate signal is above 3 V.
  For quick intro: https://youtu.be/GrvvkYTW_0k?t=139

  In this lab, we use a 3.3 V PWM signal as the gate signal to turn on MOSFET.
  So the switch will be on when PWM signal is High, will be off when PWM is Zero.
- The +5 V DC is the voltage of the main circuit loop.

Overall, the board can be simply considered as such connection:



- The + **5 V** will be drawn from the Arduino (ItsyBitsy M0 Express). Detailed ItsyBitsy M0 pinout can be found on https://learn.adafruit.com/introducing-itsy-bitsy-m0/pinouts. This 5V DC heats up the resistor.
- The **PWM** will be programmed by the Arduino. It controls the heat-power input towards the heated resistor.
- The resistor's temperature is measured by a chip under it. The chip outputs a voltage to the **Temp** Pin:

$$\text{Celsius temperature} = \text{Chip voltage} \times 10 \tag{5.1}$$

  For example, if temperature is 23 °C. The chip outputs 2.3 V voltage.
  We use Analog Discovery to measure this sensor voltage.

The detailed pin connection is given in the table. The pins in each rows must be connected. When you have multiple connections to GND, it will be better to use the blue rail on the breadboard.

the Board	Analog Discovery	Arduino
+ 5 V		Pin Vhi
Temp	Pin 1+	
PWM		Pin ?
Gnd	Pin ↓, Pin 1-	Pin G

System Model Introduction

This system can be modeled as:

- System input: the electrical power $P_{in}(t)$ (Unit: Watt) delivered to the resistor, which is converted into heat.

  The delivered power is controlled by the PWM switch. The amount of power is proportional to Duty Cycle of the PWM signal.

  When the +5V DC starts to delivery power to the resistor, it can be considered as a step input $P_{in}(t) = Au(t)$.

- System output: relative temperature $T_{rel}(t)$ (unit: °C) of the resistor. It is the difference between the actual temperature, $T(t)$, and the room temperature T_{room} ("0-power" temperature).

  For example, if the actual temperature is $T(t) = 30$°C, the room temperature is $T_{room} = 20$°C, then the relative temperature is $T_{rel}(t) = T(t) - T_{room} = 30 - 20 = 10$ °C.

- System model: approximated as a first-order system, $\frac{T_{rel}(t)}{P_{in}(t)} = \frac{b}{s+a}$.

Note, A, a, b, T_{room} are unknown constants.

Using Laplace transform → inverse Laplace transform, we can get the output in time domain:

$$T_{rel}(t) = \frac{bA}{a}(1 - e^{-at}) \tag{5.2}$$

Then we can express the actual temperature as:

$$T(t) = T_{rel}(t) + T_{room} \tag{5.3}$$
$$= \frac{bA}{a}(1 - e^{-at}) + T_{room} \tag{5.4}$$

** Note:

- The system with $P_{in}(t)$ input, $T_{rel}(t)$ output is a linear system.
- The system with $P_{in}(t)$ input, $T(t)$ output is NOT a linear system.

Report Item 1-a

Answer based on Equation (5.2).

What is the Steady-State Value of the relative temperature $T_{rel}(t)$ as time approaches ∞?
What is the Time Constant?

Report Item 1-b

Answer based on Equation (5.4).

What is the expression of the actual temperature $T(t)$ when time approaches ∞?

** Note: final result should be in terms of A, a, b, T_{room}

Task 2 – Heat using Full Power

tIn the prelab, you studied the relationship between Power and Duty Cycle: They are proportional!

In this task, we will test the **heating power using Full Power**, in other words, **PWM Duty Cycle is 100%**

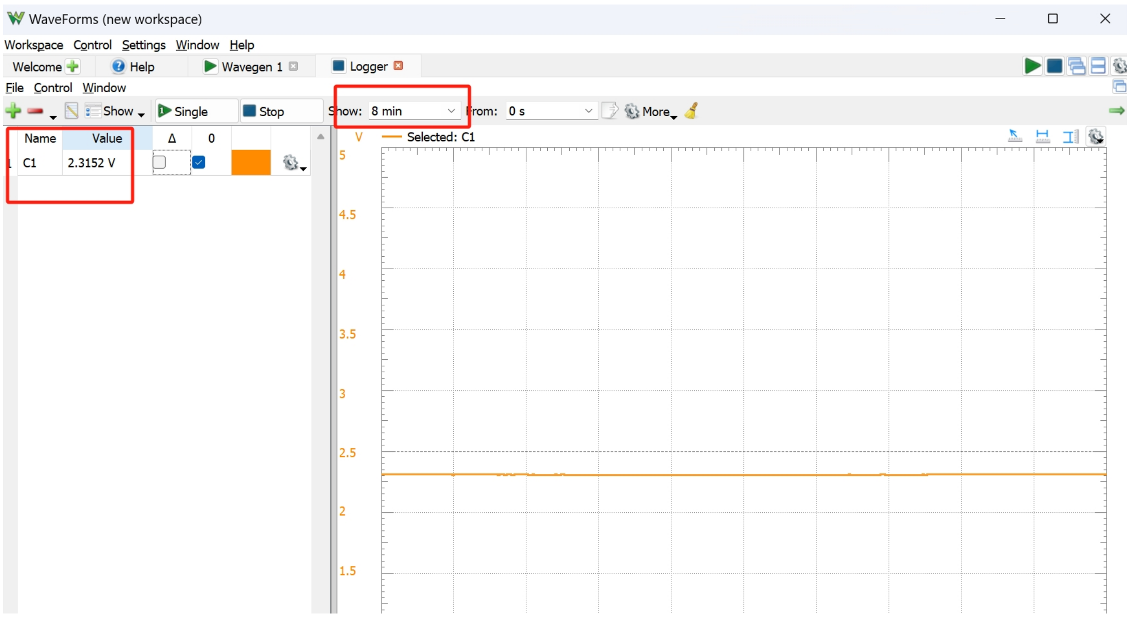
Wire everyhing up. Re-run the Arduino code given in the Pre Setup.

In Analog Discovery, open “Logger”. This is a new tool to you. It is mainly used to record signals in very long time period.

Make sure only “C1” Channel is selected. It corresponds to the TEMP readings from the board sensor.

Adjust the “Show” time to be large. We recommend 10 minutes.

Run the Logger, make sure that you can see a flat line. The voltage reading should be reasonable value that corresponds to the room temperature based on Equation (5.1).

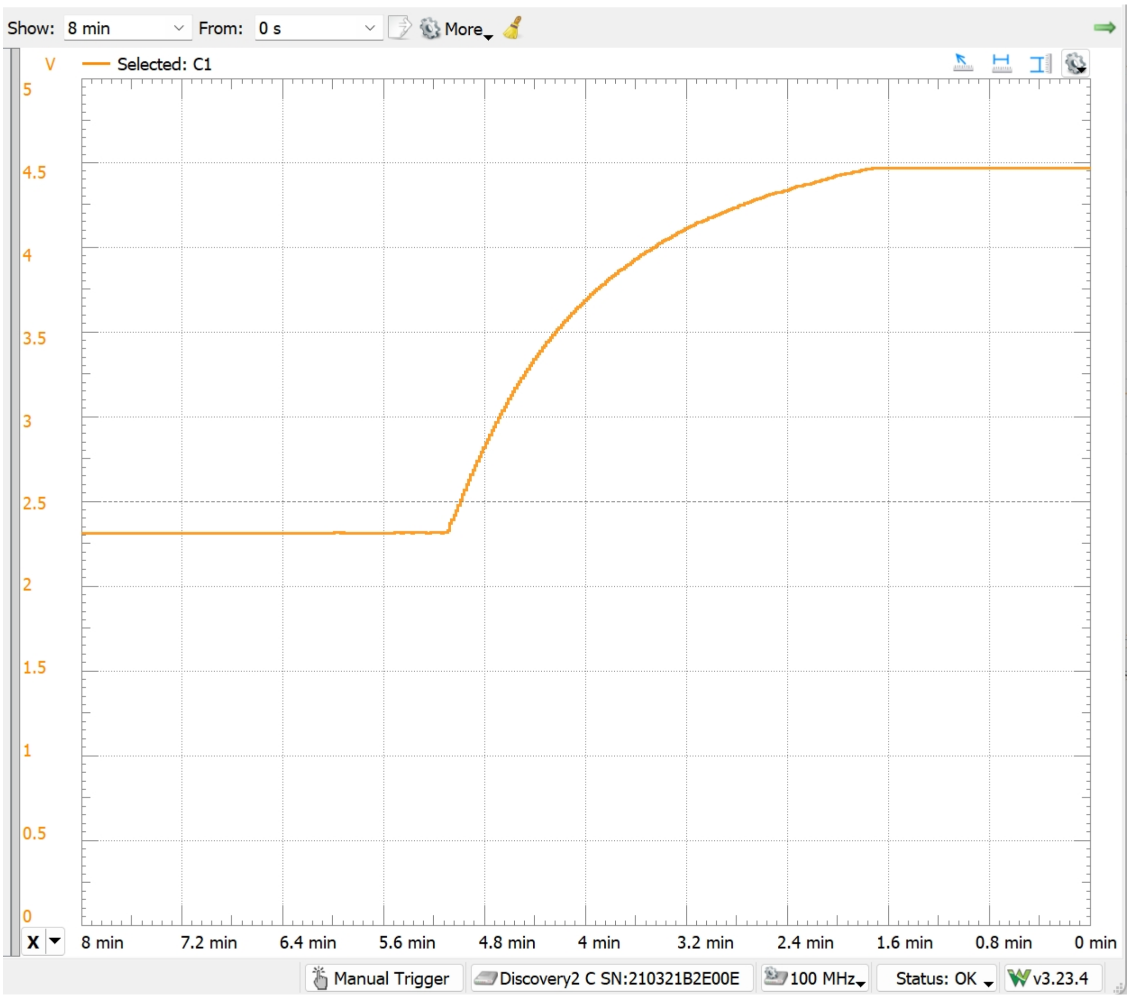


Keep the Logger running. Don't stop it.

Now, go to Arduino, adjust the value in `analogWrite(PWMPin, ?)`. So that make the PWM to be 100% Duty Cycle (fully HIGH).

Upload the Arduino code. Patiently wait for the response on Logger.

On the logger, you should observe the signal gradually increasing in an exponential manner, eventually reaching a steady state.



** This is the graph that I got. The graph must include: 1, initial state; 2, steady state.

Report Item 2-a

Save the Logger display to .csv and plot it in Jupyter Notebook.

Note: your original timestamp in .csv is reversed.

Check Point 1 – Plot in Python

Once you checked, re-set the Arduino to be `analogWrite(PWMPin, 0);`.

This changes to be 0 Power input. So the resistor load will begin cooling down via air. You should see the signal in Logger is dropping down.

Wait until you see a flat line as room temperature. Then start Task 3.

Be very patient. It will take over $\geq$ **8 minutes.**

Report Item 2-b (Individual)

Using pandas to analyze the data. Show your code and fill the table.

Min voltage	
Max voltage	
Room temperature	
Steady-state temperature	

Report Item 2-c (Individual)

Using pandas to analyze the Time Constant. Show your code and final result.

Don't use any curve fitting approach (ChatGPT may tell you this way!). Instead, you should use the 0.632-point approach, the one you learned in Lab 3.

Pandas `.iloc` and `.loc` may be helpful.

Task 3 – Heat using Half Power

Once your resistor is fully cooled down, you can start this Task.

Keep the same wiring. **Keep the Logger running. Don't stop it.**

Go to Arduino, adjust the value in `analogWrite(PWMPin, ?)`. So that make the PWM to be 50% Duty Cycle (half HIGH).

Repeat the same procedures in Task 2 to log the heating signal.

Report Item 3-a

Save the Logger display to .csv and plot it in Jupyter Notebook.

Note: your original timestamp in .csv is reversed.

Check Point 2 – Plot in Python

Once you checked, re-set the Arduino to be `analogWrite(PWMPin, 0);`.

Report Item 3-b (Individual)

Using pandas to analyze the data. Show your code and fill the table.

Min voltage	
Max voltage	
Room temperature	
Steady-state temperature	

Report Item 3-c (Individual)

Using pandas to analyze the Time Constant. Show your code and final result.

Don't use any curve fitting approach (ChatGPT may tell you this way!). Instead, you should use the 0.632-point approach, the one you learned in Lab 3.

Pandas `.iloc` and `.loc` may be helpful.

Lab Report

Please complete the report individually.

Write the report in Jupyter Notebook. Then export it as .html. Finally convert .html to .pdf. The submission in BlackBoard must be in .pdf.

The total score for the report is 20 points.

- Format. (3 points)
- Item 1-a, 1-b (2x1 points)
- Item 2-a, 2-b, 2-c (3x2 points)
- Item 3-a, 3-b, 3-c (3x2 points)
- Conclusion. (3 points)