Title

Protocol for building a user-friendly temperature control system to support microscopes, microfluidic chambers, and custom incubators.

Journal
Star Protocols, 5(1)

Authors
Liu, Ziteng
Nguyen, Thao
Ding, Fangyuan

Publication Date
2024-01-30

DOI
10.1016/j.xpro.2024.102862

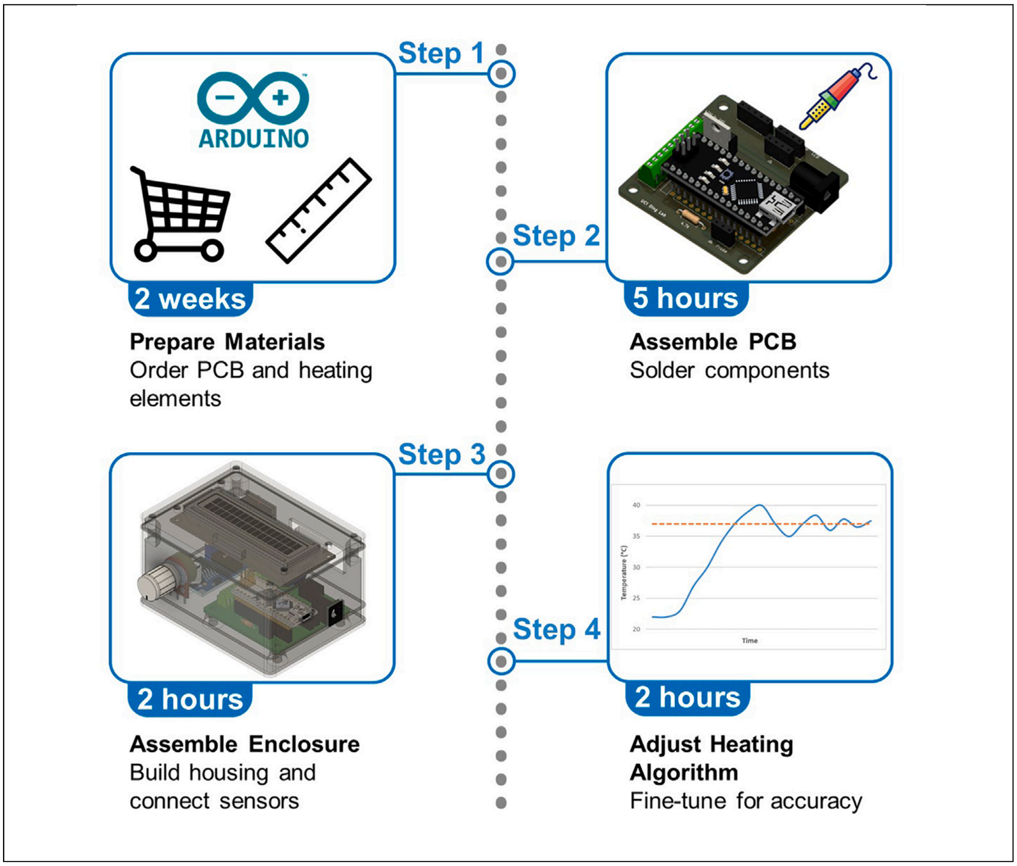
Peer reviewed

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for building a user-friendly temperature control system to support microscopes, microfluidic chambers, and custom incubators



Step 1

2 weeks

Prepare Materials
Order PCB and heating elements

Step 2

5 hours

Assemble PCB
Solder components

Step 3

2 hours

Assemble Enclosure
Build housing and connect sensors

Step 4

2 hours

Adjust Heating Algorithm
Fine-tune for accuracy

Biological experiments require precise temperature control, necessitating an integrated adjustable temperature system for instruments such as microscopes, microfluidic chambers, or custom incubators. We present a protocol for building a user-friendly temperature control system suitable for both *in vitro* and *in vivo* assays. We describe steps for preparing materials, assembling the printed circuit board and enclosure, and fine-tuning the heating algorithm for accuracy. This system can maintain a stable temperature of up to 60°C with stabilities under 0.06°C.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Ziteng Liu, Thao Thi Thu Nguyen, Fangyuan Ding

zitenl1@uci.edu (Z.L.)
dingfy@uci.edu (F.D.)

Highlights

A protocol for building an intuitive and versatile temperature control system

Detailed steps to solder the temperature control circuit board assembly

Steps to assemble the 3D-printed electronics enclosure for temperature adjustment

Approach to increase temperature stability by fine-tuning PID heating algorithm

Liu et al., STAR Protocols 5, 102862
March 15, 2024 © 2024 The Author(s).
https://doi.org/10.1016/j.xpro.2024.102862

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Protocol for building a user-friendly temperature control system to support microscopes, microfluidic chambers, and custom incubators

Ziteng Liu,[1,4,*] Thao Thi Thu Nguyen,[2] and Fangyuan Ding[2,3,5,*]

[1]Department of Electrical Engineering and Computer Science, University of California Irvine, Irvine, CA 92697, USA

[2]Center for Complex Biological Systems, University of California Irvine, Irvine, CA 92697, USA

[3]Department of Biomedical Engineering, Center for Synthetic Biology, Chao Family Comprehensive Cancer Center, Department of Developmental and Cell Biology, and Department of Pharmaceutical Sciences, University of California Irvine, Irvine, CA 92697, USA

[4]Technical contact

[5]Lead contact

*Correspondence: zitenl1@uci.edu (Z.L.), dingfy@uci.edu (F.D.)
https://doi.org/10.1016/j.xpro.2024.102862

SUMMARY

Biological experiments require precise temperature control, necessitating an integrated adjustable temperature system for instruments such as microscopes, microfluidic chambers, or custom incubators. We present a protocol for building a user-friendly temperature control system suitable for both *in vitro* and *in vivo* assays. We describe steps for preparing materials, assembling the printed circuit board and enclosure, and fine-tuning the heating algorithm for accuracy. This system can maintain a stable temperature of up to 60°C with stabilities under 0.06°C.

BEFORE YOU BEGIN

Temperature influences physiological and pathological processes,[1] so experiments studying biological systems require precise temperature environments and the ability to test various conditions. Therefore, an adjustable temperature control system integrated into the experimental instrument is essential. This protocol is tailored to enhance commercial or in-house devices without a built-in temperature system, including microscopes,[2] microfluidic devices,[3,4] and self-built incubation chambers.[5,6] The system allows researchers to perform *in vitro*[7] or *in vivo*[8,9] assays at a desired temperature on their existing instruments, such as characterizing biochemical reactions[10,11] or live cell tracking.[12]

In detail, the temperature control system consists of two main sections: an electronics enclosure and a heating block (Figure 1A). The enclosure contains a custom circuit board connecting the microcontroller, Arduino Nano, to user interface components such as the temperature display and the target temperature control knob (Figure 1B). On the heating block, two flexible heating elements surround the magnetic tweezer camera objective to heat the block evenly (Figure 1C). Two temperature sensors are attached to the front of the heating block: one located within the block, acting as a probe, and another affixed to the block's exterior surface (Figure 1D).

The Arduino Nano first reads the user input of the target temperature and the current measured temperature. Then, with the Proportional-Integral-Derivative (PID) algorithm, it determines the exact percentage of power needed to maintain the temperature without overshoots and oscillation.[13,14] Finally, it sends the calculated output to the Metal Oxide Silicon Field Effect Transistor (MOSFET), which acts like an analog switch to control the heating elements.[15] As electricity flows through the

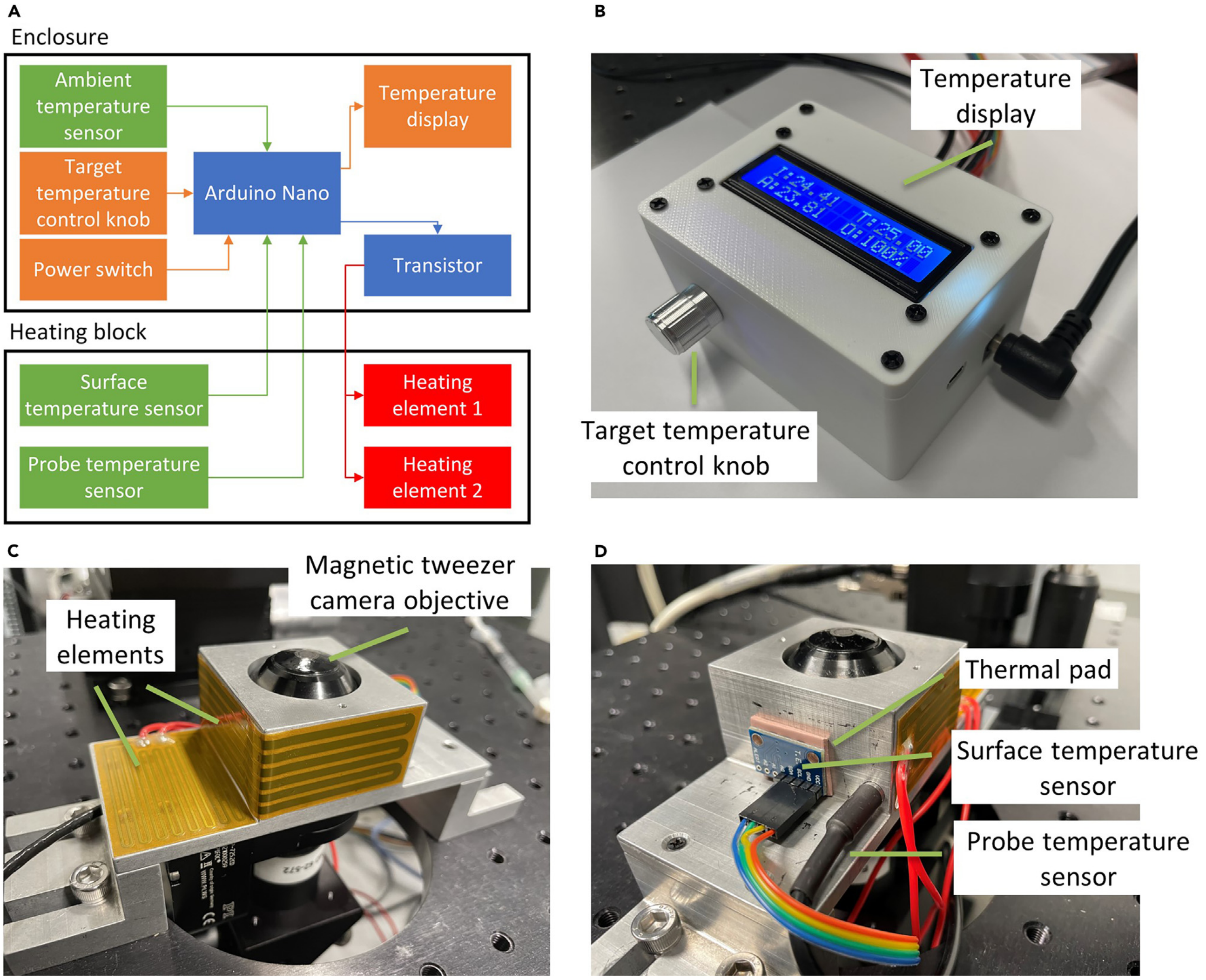


Figure 1. The temperature control system consists of two main sections: an enclosure with target temperature adjustment and display, and a heating block with temperature sensors and heating elements

(A) The block diagram of the temperature control system shows the major components, including three temperature sensors (green), two heaters (red), a heating controller (blue), and a user interface (orange). The upper half of the components in the diagram are placed inside the 3D printed enclosure shown in Figure 1B. The lower half of the components are placed on the heating block, as shown in Figures 1C and 1D.

(B) The enclosure has a user-friendly interface with a temperature adjustment knob and a temperature display showing the internal average temperature (I), target temperature (T), ambient temperature (A), and heating power output (O) in real time.

(C) To heat the block evenly, two pieces of polyimide heating elements cover the entire back side of the magnetic tweezer objective block.

(D) The surface temperature sensor attaches to the thermal pad in front of the heating block to measure the surface temperature, while the probe sensor measures the temperature inside the heating block.

flexible polyimide Nichrome heating elements, the heating block gradually reaches and maintains the desired temperature.[16]

Here, as an example platform, we designed the temperature control system to integrate with our magnetic tweezers.[17] However, the system can be modified to accommodate varying user applications including those in need of higher heating power.

Software installation

Timing: 30 min

1. Install the latest version of Arduino IDE: https://www.arduino.cc/en/software.
2. In the Library Manager of Arduino IDE, download the software libraries listed in the key resources table.

Note: The Arduino Nano will use these libraries as a part of the code in the future steps.

Prepare temperature control materials

Timing: 2 weeks

3. Prepare the heating elements.
   a. Determine the shape and dimension of the polyimide heating elements.

Note: In our setup, we attached a 25 × 138.5 mm piece around the heating block and a 49 × 34 mm piece to the backside (Figure 1C).

      i. Locate your heating block and its surfaces to attach the polyimide heating elements.

Note: The polyimide heating element is thin and flexible, offering a range of possible shapes, including rectangles and circles.

      ii. Measure out the desired dimension and shape of the heating elements.
   b. Determine the heating power.

Note: In our setup, the 25 × 138.5 mm piece is rated for 40 W at 24 V and the 49 × 34 mm piece is rated for 15 W at 24 V. Therefore, the total power is 55 W at 24 V. Turning the heating elements on at full power heats the block to a maximum temperature of 60°C.

      i. The heating power will vary depending on your temperature target, heating block volume, and material.

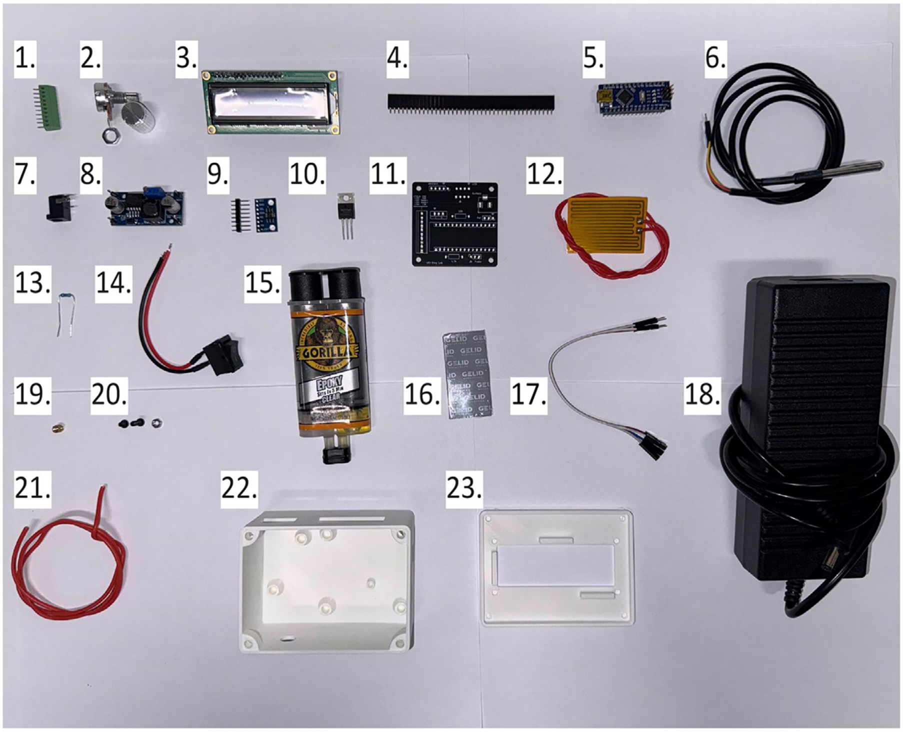
Optional: To determine the exact heating power needed, measure the power usage with commercially available polyimide heating elements. Install the heating elements and connect them to a lab power supply. Turn on the power supply and vary the voltage until the desired equilibrium temperature is reached. The power usage shown on the power supply is the total power needed for the custom polyimide heating elements.

   c. Order the customized heating elements.
      i. Contact polyimide heating element manufacturers with the desired size and power. The elements should have adhesive on one side and lead wires long enough to reach the electronics enclosure.
4. Prepare the temperature sensors.

Note: We applied two sensors in our setup (Figure 1D). The blue MCP9808 surface sensor in the front is a high-accuracy temperature sensor. The black DS18B20 probe sensor is an additional sensor inserted into the objective block as a secondary measurement. Notably, using the latter probe sensor requires drilling a hole in the target block.

   a. Depending on the space of the setup, either one or both sensors can be used.

Note: If only one is used, you will need to adjust the Arduino code accordingly. For example, if the probe sensor is not used, delete all references to the probe sensor in the code, such as functions that request temperature readings or print its data on the display.



1. 10 pin screw terminal
2. Potentiometer with knob
3. I2C LCD
4. 40-pin female header
5. Arduino Nano
6. Probe temperature sensor
7. DC jack barrel
8. Buck converter
9. Surface temperature sensor
10. MOSFET
11. Printed Circuit Board (PCB)
12. Polyimide heating element
13. Resistor
14. Power switch
15. Epoxy
16. Thermal pad
17. Jumper wires
18. Power supply
19. Threaded insert
20. M3 screws and nut
21. Electrical wire
22. Enclosure bottom

Figure 2. The list of components in the key resources table and the custom enclosure

5. Prepare the Printed Circuit Board (PCB).
   a. The KiCad design file zip and manufacturer Gerber files are provided here: https://github.com/UCI-Ding-Lab/Temperature-Control/tree/main/PCB.
   b. PCB can be ordered from any manufacturer. We recommend ordering with a lead-free surface finish for safety.
6. Prepare the electronics enclosure.
   a. The STL files for the top and bottom pieces are provided here: https://github.com/UCI-Ding-Lab/Temperature-Control/tree/main/Enclosure.
   b. The enclosure can be printed on any 3D printer without support material. We printed with white PLA on an Ender 3 V2 printer. Parts can also be outsourced to a 3D printing service.
7. Prepare other commercially available items listed in the key resources table.
   a. Figure 2 depicts all materials used in this protocol in the order of the key resources table.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
PCB KiCad files	This paper	https://github.com/UCI-Ding-Lab/Temperature-Control/tree/main/PCB
PCB enclosure	This paper	https://github.com/UCI-Ding-Lab/Temperature-Control/tree/main/Enclosure
Example heating curve data	This paper	https://github.com/UCI-Ding-Lab/Temperature-Control/tree/main/Example%20Data

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
Adafruit MCP9808 library	Adafruit	https://www.arduino.cc/reference/en/libraries/adafruit-mcp9808-library/
Arduino IDE	Arduino	https://www.arduino.cc/en/software
DallasTemperature library	Miles Burton	https://www.arduino.cc/reference/en/libraries/dallastemperature/
LiquidCrystal I2C library	Frank de Brabander	https://www.arduino.cc/reference/en/libraries/liquidcrystal-i2c/
OneWire library	Jim Studt et al.	https://www.arduino.cc/reference/en/libraries/onewire/
QuickPID library	David Lloyd	https://www.arduino.cc/reference/en/libraries/quickpid/
Temperature control Arduino code	This paper	https://github.com/UCI-Ding-Lab/Temperature-Control/tree/main/Arduino%20Code
Other		
10 pin 2.54 mm screw terminal	Amazon	Cat#B07WQV2PDD
10 k Ohm potentiometer with knob	Amazon	Cat#B09897HR3C
16x2 I2C LCD	Amazon	Cat#B07S7PJYM6
2.54 mm female header, 40 pins	Amazon	Cat#B00VVI1L1W
Arduino Nano with headers	Amazon	Cat#B07G99NNXL
DS18B20 probe temperature sensor	Amazon	Cat#B09NVFJYPS
DC jack barrel	Amazon	Cat#B081DYQSC9
LM2596 buck converter	Amazon	Cat#B08NV3JCBC
MCP9808 temperature sensor	Amazon	Cat#B07WXPGZKZ
N-Channel MOSFET	Amazon	Cat#B07CTF1JVD
PCB	JLCPCB	https://jlcpcb.com/
Polyimide heating element	Topright Industrial Technology	https://cntopright.en.made-in-china.com/product/edxmhrJjYnVS/China-Topright-Heating-Film-Customize-Strip-Pet-Film-Heater.html
Resistor, 4.7 k and 10 k Ohm	Amazon	Cat#B06WRQS97C
Rocker power switch	Amazon	Cat#B07SPY9H7Y
Epoxy	Amazon	Cat#B01M7VD07W
Thermal pad	Amazon	Cat#B07BSPZTNV
Jumper wires, 15 cm male to female	Amazon	Cat#B07GD2869Z
24 V power supply	Amazon	Cat#B01GC6VS8I
M3 threaded inserts	Amazon	Cat#B09TNK8GD6
M3 8 mm, 12 mm screws, and nuts	Amazon	Cat#B08N5XDHMW
18-gauge solid electrical wires	Amazon	Cat#B08KFS5342
Soldering kit	Amazon	Cat#B0B3D96MN6
Dupont wire crimp toolkit	Amazon	Cat#B07ZK5F8HP
Wire cutter	Amazon	Cat#B00FZPDG1K
Wire striper	Amazon	Cat#B073YG65N2
Phillips head screwdriver	Amazon	Cat#B08H5VJHJY
Mini flathead screwdriver	Amazon	Cat#B0C1JXWY8N
Multimeter	Amazon	Cat#B00KXX2OYY
Electrical tape	Amazon	Cat#B0000AZ9HG
Lab power supply (optional)	Amazon	Cat#B08DJ1FDXV
Standard polyimide heating element (optional)	Amazon	Cat#B074SXKPZL
3D printer (optional)	Amazon	Cat#B088GTBQWD

STEP-BY-STEP METHOD DETAILS

Build the PCB assembly

Timing: 5 h

In this step, you will solder all components to the PCB. The placement of the components is printed on the PCB and shown in the layout (Figure 3A). Components are placed on the top or front side of the PCB. Then, the component leads pass through the holes and are soldered on the backside. Make sure the component is aligned and flush against the PCB before soldering.

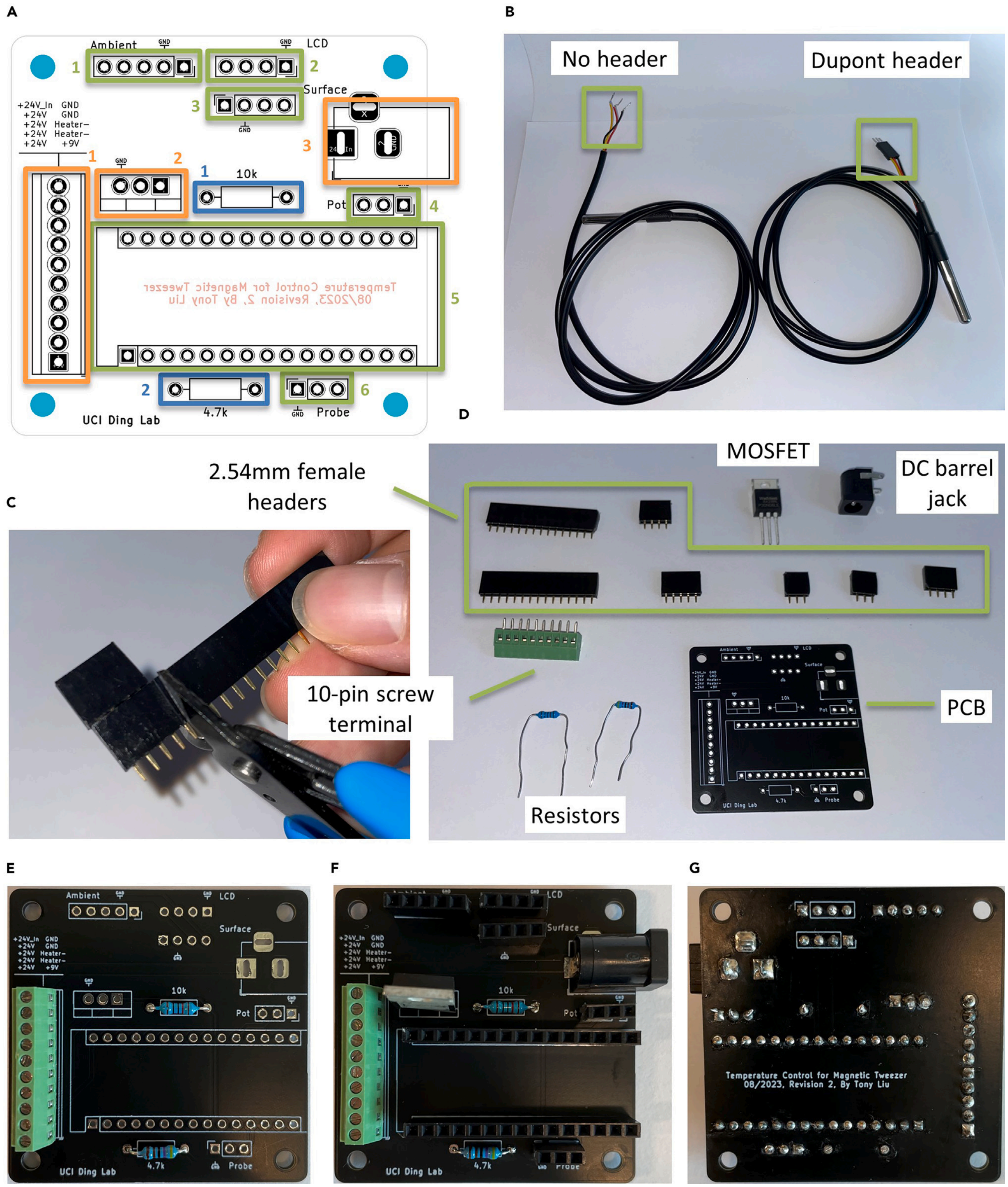


Figure 3. We crimped the probe temperature sensor, cut the female headers, and soldered all components onto the compact Printed Circuit Board (PCB)

(A) The PCB layout in KiCad with components to solder marked in colored rectangles. Green rectangles indicate female headers for various components: (1) ambient temperature sensor, (2) temperature display (LCD), (3) surface temperature sensor, (4) temperature adjustment knob

Figure 3. *Continued*

(potentiometer), (5) Arduino Nano, and (6) probe temperature sensor. Blue rectangles indicate resistors: (1) 10 k Ohm and (2) 4.7 k Ohm. Orange rectangles indicate miscellaneous components: (1) 10-pin screw terminal, (2) MOSFET, and (3) DC barrel jack.
(B) We attached a 3-pin Dupont header to the DS18B20 probe sensor.
(C) We cut the female headers at the 5th pin to make a 4-pin header to avoid loose pins.
(D) All components that need to be soldered to the PCB: female headers, MOSFET, barrel jack, screw terminal, and resistors.
(E) The PCB after soldering resistors and 10-pin screw terminal (steps 2a and 2b).
(F) Components are soldered and flush against the PCB.
(G) On the backside of the PCB, there is no solder between component pins, and all pins are fully connected.

Note: We recommend this comprehensive guide for additional information on soldering.

⚠ CRITICAL: Ensure the correct orientation of each component before soldering. After soldering, inspect all solder joints to ensure full connection and no shorts between any component leads. Resolder any loose solder joints if needed.

1. Crimp the male header onto the probe sensor.
   a. With a Dupont header crimp kit, attach a 3-pin male header to the end of the DS18B20 probe sensor. The wires are in the order of red-black-yellow.

Note: We recommend this step-by-step guide on how to crimp Dupont headers. Figure 3B shows the crimped sensor.

   b. If you do not have crimping tools to attach Dupont headers to the probe sensor wires, you may skip step 1. Instead, you will solder the wires of the probe sensor directly to the PCB. However, replacing the sensor for troubleshooting will be inconvenient.
2. Cut 40-pin 2.54 mm female headers into various segments.
   a. With a wire cutter, cut two segments of 3-pin headers, two segments of 4-pin headers, one segment of 5-pin header, and two segments of 15-pin headers.

Note: These female header segments will be later soldered to the PCB to connect the components.

   b. If you skipped crimping Dupont headers for the probe sensor, you may skip cutting one of the 3-pin headers.

Note: The pin at the cut location may come loose after cutting. To avoid it, cut away from the desired location first then trim down gently. For example, to prepare a 4-pin header, cut the segment at the 5th pin. The 5th pin will fall out or be trimmed down (Figure 3C).

3. Solder components to PCB in the following order (from lowest to highest in height): resistors, 10-pin screw terminal, female headers, DC barrel jack, and MOSFET.

Note: Figure 3D shows the list of components.

Note: Solder the components on an inflammable surface such as a silicone mat. Tape the PCB to the surface while soldering for stabilization.

Note: +5 V and ground pins take longer to solder because the pins are connected to a large copper island and take longer to heat up. Therefore, hold the soldering iron against the pad and component lead for at least 1 min to fully heat the solder pad.

   a. Solder 4.7 k and 10 k resistors.

Note: Bend resistor leads after inserting to hold the resistor in place.

Note: Resistors are non-polarized, so they can be inserted in any direction. Ensure the resistor location matches the PCB text and the layout.

b. Solder the 10-pin screw terminal and ensure the wire insertion side faces the left edge of the PCB.
   i. Solder the two outer pins first to hold the jack to the board.
   ii. Solder every other pin to avoid electrical shorts from solder bridges.

Note: Figure 3E shows the PCB after completing this step.

c. Solder all female headers starting with the 15-pin headers.
d. Solder the DC barrel jack.
e. Solder the MOSFET. Ensure the MOSFET is upright while soldering.

Note: As shown in Figure 3F, the metal tab should be at the bottom side of the PCB.

f. Trim excess resistor and MOSFET leads.

Note: Figures 3F and 3G show the front and back of the completed PCB.

⚠ CRITICAL: Allow the PCB and components to cool down after soldering before touching them.

4. Solder jumper wires directly to the LCD to bypass the plastic header and reduce the enclosure size.
   a. Split the 15 cm jumper cable ribbon into a 4-wire segment.
   b. With a wire cutter, cut off the female header and only the header.
   c. Split and spread the wires at the cut end by 3 cm.
   d. Strip the insulation from the split ends by 3 cm.
   e. Solder each wire to a male header of the LCD (Figure 4A).
   f. Wrap the soldered header with electrical tape for insulation.
5. Prepare the MCP9808 temperature sensors.
   a. Cut the 8-pin male headers into one 4-pin segment and one 5-pin segment.
   b. Solder the headers onto the MCP9808 (Figure 4B).
      i. Insert male headers starting from the hole marked VCC.
      ii. Insert the headers with the short end facing up and into the temperature sensor board.

Note: The sensor with the 4-pin header will be used on the heating block, and the one with the 5-pin header will be used as an ambient sensor in the electronics enclosure.

6. Solder wires to the buck converter.
   a. Cut four 10 cm 18-gauge wires.

Note: Two black and two red wires are preferred to distinguish between negative and positive. In our setup, two blue wires are used as negative and two red wires are used as positive.

   b. Strip off 0.5–1 cm of the insulation on the wires from both ends.
   c. Solder wires to the buck converter.
      i. Pass through the wire from top to bottom, then solder at the backside (Figure 4C).
7. Adjust the buck converter to output 9 V for the Arduino Nano.
   a. With a lab power supply, connect its cables to the input side of the buck converter with positive to the IN+ and negative to IN-.

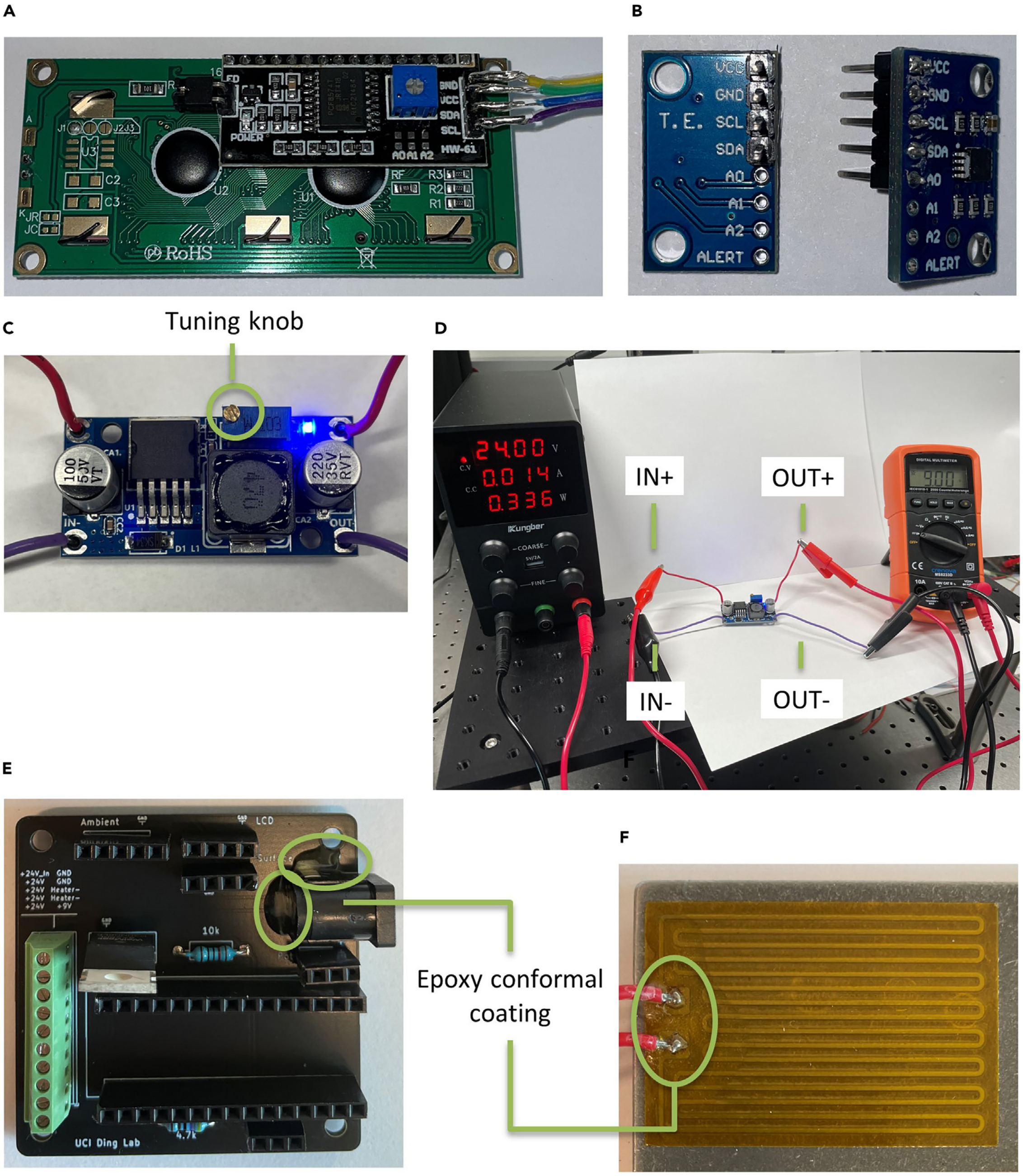


Figure 4. Additional adjustments to the LCD, surface and ambient temperature sensors, buck converter, and the PCB

(A) We soldered jumper wires directly to the headers on the back of the LCD.

(B) We soldered 4-pin and 5-pin male headers to the surface and ambient temperature sensors.

(C) We soldered wires to the input and output of the buck converter and adjusted the output voltage by turning the tuning knob.

(D) We adjusted the buck converter to output 9 V by adjusting it with a 24 V lab power supply.

(E) We applied a thin coat of epoxy to the top and left of the barrel jack to prevent shorting within the enclosure.

(F) We applied a thin coat of epoxy to the solder joints of the heating element to avoid water damage during experiments.

b. With a multimeter set to measure DC voltage, connect the alligator clips cables to the output side of the buck converter with the positive to the OUT+ and negative to OUT- (Figure 4D).

c. Turn on the power supply and adjust the voltage to 24 V (Figure 4D).

   d. With a small flathead screwdriver, turn counterclockwise on the blue trim pot to lower the output voltage.
   e. Stop until the output voltage shown on the multimeter is at 9 V. Ensure the screwdriver does not touch electrical components (Figure 4C).
8. Waterproof solder joints with epoxy to protect them from moisture and accidental shorts.
   a. Mix around 10 mL of clear epoxy and apply a light coat to the heating element solder joints and PCB barrel jack.

Note: Ensure not to put an excessive amount to prevent spillage (Figures 4E and 4F).

⏸ **Pause point:** Wait 24 h for the epoxy to cure and harden.

9. Place Arduino Nano into the soldered female headers with the USB port facing right.

Build the mechanical assembly

🕓 **Timing: 2 h**

In this step, you will build the mechanical assembly by connecting all electrical components (buck converter, potentiometer, switch, and LCD) to the PCB. Then, you will install the PCB in the compact 3D printed enclosure. You will also attach threaded inserts into the 3D printed enclosure with a soldering iron to install screws.

Note: We recommend this short video as a reference for inserting threaded inserts.

10. Install threaded inserts into the 3D printed enclosure.

Note: The threaded inserts allow the insertion of screws in the soft plastic material.

   a. Place 10 threaded inserts in the bottom enclosure in the top corners and bottom mounting holes (Figure 5A).
   b. Set the soldering iron to around 10°C higher than the melting point of the enclosure material.

Note: For example, the melting point of PLA is around 200°C, so the soldering iron is set to 210°C.

   c. Insert the soldering iron tip in the threaded inserts very slowly until flushed flat.

Note: Do not put excess force. It should take around 30 seconds per insert.

Optional: With a tweezer or other flat object, press the threaded inserts flush to help maintain a flat edge as it cools down.

⏸ **Pause point:** Wait for the threaded inserts to set and cool down before proceeding to the next steps.

11. Tighten down all components with standoffs and screws.
   a. Insert two 12 mm M3 screws into the buck converter mounting holes.
   b. Screw in one M3 nut loosely from the bottom as a spacer.
   c. Place the buck converter in the bottom enclosure and tighten it down.

Note: Figure 5B shows the side view of the installed buck converter with a nut as the spacer. The output side should face towards the front of the enclosure (Figure 5C).

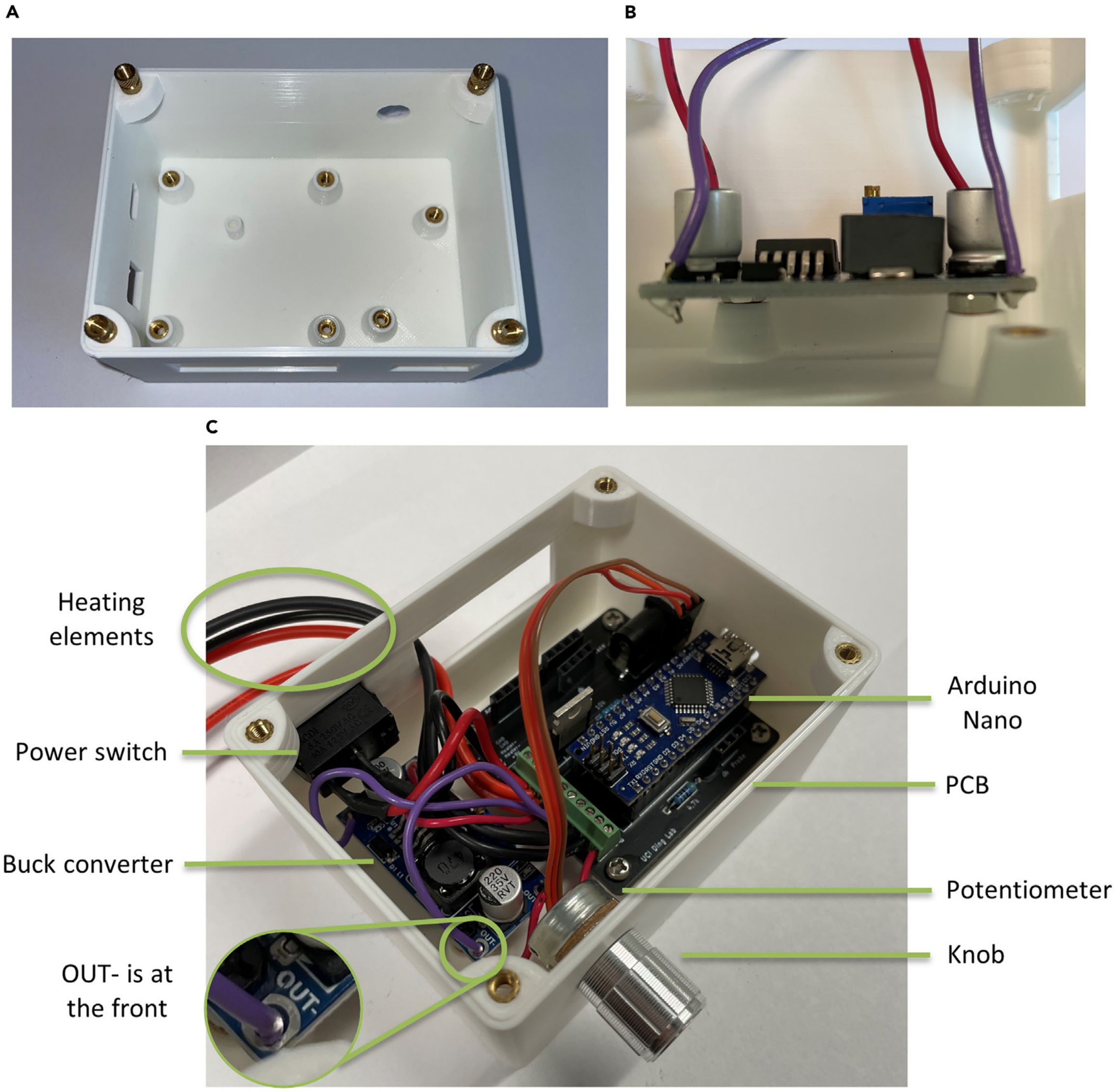


Figure 5. We installed threaded inserts into the 3D printed enclosure and then tightened down the PCB, power switch, buck converter, and potentiometer

(A) We installed 6 threaded inserts at the bottom of the enclosure and placed 4 threaded inserts at the top.

(B) As shown from the side view, we used one M3 nut as a spacer when installing the buck converter.

(C) We used 4 M3 screws to install the PCB into the bottom of the enclosure. The power switch is snapped in, and the potentiometer is installed with a knob outside. Note that the buck converter is placed with OUT- at the front of the enclosure.

d. Insert the LCD into the top enclosure and tighten it down with four 12 mm M3 screws and nuts.

Note: The pin headers on the LCD should match the rectangular support blocks on the underside of the top enclosure. The male headers of the LCD should face to the left of the enclosure.

12. Install switch and potentiometer.
    a. Cut off half of the switch lead wires, leaving only 10 cm.
    b. Strip off the wire insulations off wires by 1 cm.
    c. Insert the switch into the rectangular cutout in the bottom enclosure piece.

d. Split the 15 cm jumper cable ribbon into a 3-wire segment for the potentiometer.
e. Connect the potentiometer to the female end of the jumper wires.
f. Cut off the extruded metal nib on the potentiometer.
g. Insert the potentiometer into the round hole in the bottom enclosure piece.
h. Secure the potentiometer by screwing in the nut from the outside.
i. Turn the potentiometer all the way counterclockwise.
j. Attach the knob from the outside with the indicator facing down.

13. Connect wires to the PCB screw terminal jack.
    a. Connect the wires to the screw terminal from top to bottom by following the text on the PCB and layout shown in Figure 3A.

Note: The wires are listed as follows from top to bottom: switch+, switch-, heating element 1+, heating element 2+, buck converter in+, buck converter in-, buck converter out-, heating element 1-, heating element 2-, and buck converter out+.

Note: The heating element number and polarity are assigned arbitrarily. For example, the 25 × 138.5 mm piece is named heating element 1 with the left wire as + and the right wire as -; the 49 × 34 mm piece is then named heating element 2.

⚠ CRITICAL: It is important to insert the + and – in the correct order.

Note: Similarly, the polarity of the power switch is assigned arbitrarily. Either one of the two wires from the switch can be inserted into the switch+ slot.

i. Unscrew all pins on the screw terminal with a flat mini screwdriver.
ii. Insert the wire and tighten the screw to lock the wire in place.

14. Place the PCB in the bottom enclosure and align the mounting holes.
    a. Tighten it down with four 8 mm M3 screws (Figure 5C).
15. Place the surface MCP9808 sensor on the heating block.
    a. Cut a piece of the thermal pad the same size as the surface MCP9808 sensor.
    b. Place the thermal pad between the surface sensor and heating block for electrical insulation and thermal conductivity.
    c. Attach the 4-pin MCP9808 sensor onto the thermal pad.

Optional: Tape and secure the sensor if necessary (Figure 1D).

16. Connect all components to the PCB.
    a. Daisy-chain multiple 15 cm segments of jumper wires to prepare a long cable for attaching to the surface MCP9808 temperature sensor.

Note: This cable should be long enough to reach the heating block from the electronics enclosure. It can also be made from crimping your own cable or purchasing another set of long jumper wires.

b. Connect the female end of the jumper wires to the components.

Note: For the potentiometer, slightly split open the end of the cable to account for the wider gap between the pins.

c. Connect the male ends of the jumper wires to the PCB headers. Route the wires through the rectangle cutout in the back.

Note: Ensure components are placed in their respective headers, and the pins are orientated correctly with the GND pins at the marked spots on the PCB. For example, the black wire of the

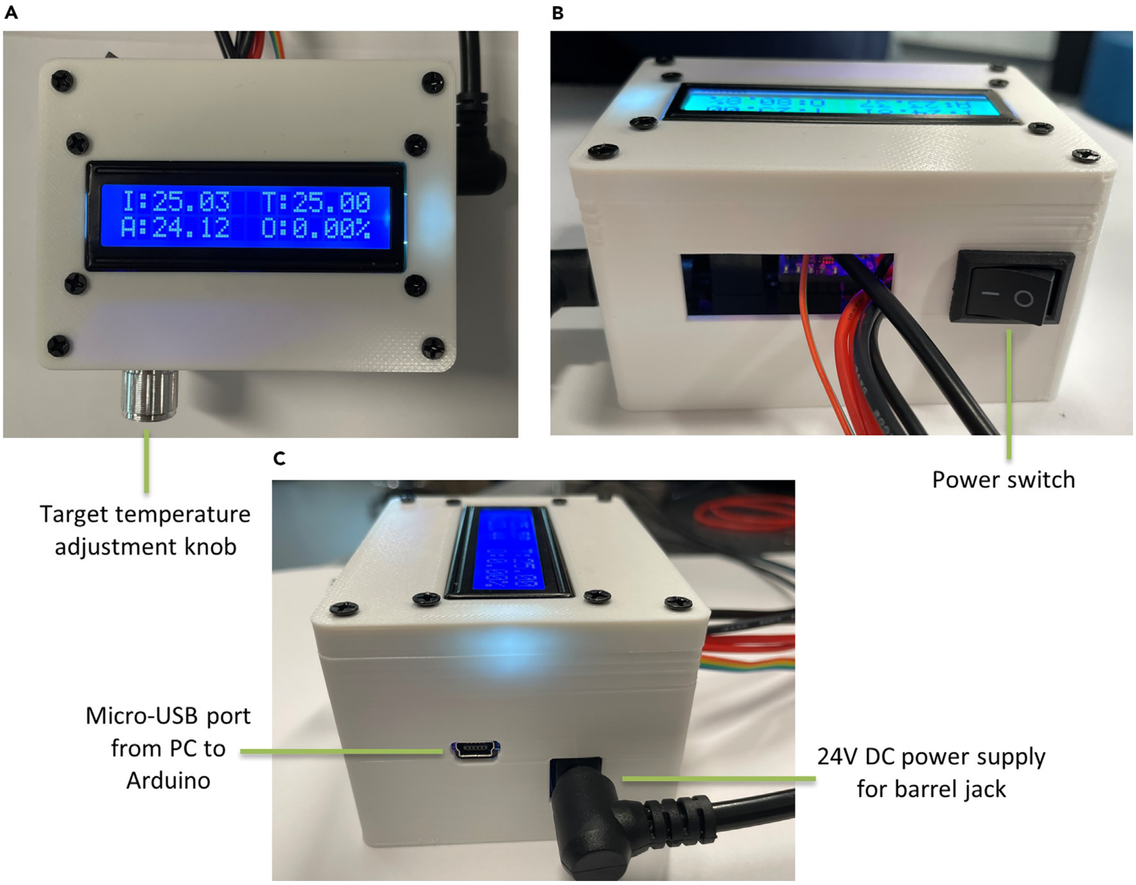

A
I:25.03 T:25.00
A:24.12 O:0.00%
Target temperature adjustment knob
B
Power switch
C
Micro-USB port from PC to Arduino
24V DC power supply for barrel jack

Figure 6. The complete powered-on temperature control system maintains temperature at 25°C

(A) Top view of the system with the LCD shows the internal average heating block temperature (I), target temperature (T), ambient temperature (A), and heating power percentage (O).

(B) Back view of the system with the power switch turned on and wires of the components on the heating block routed through the cable cutout.

(C) Side view of the system with precise micro-USB port and DC power supply cutouts.

probe sensor should line up with the GND text at the left pin. Refer to the layout in Figure 3A for detailed orientation.

d. Install the ambient 5-pin MCP9808 sensor on the PCB.

17. Connect the top and bottom enclosure.
    a. With four 12 mm M3 screws, tighten down the top case to the bottom.

Note: The complete system is shown in Figure 6.

Upload software to the Arduino

🕓 Timing: 30 min

In this step, you will upload the Arduino code sketch to the Arduino Nano on the circuit board. You will also test its basic functionality and ensure all components are working properly.

⚠ CRITICAL: Ensure the circuit board is correctly installed and soldered. Powering it on with electrical shorts could result in irreversible damage.

18. Upload software to the Arduino.
    a. Download the required code file here: https://github.com/UCI-Ding-Lab/Temperature-Control/tree/main/Arduino%20Code.
    b. With a Micro-USB cable, connect the Arduino Nano to the computer (Figure 6C).
    c. In the Arduino IDE, open the downloaded Ding_Temperature_Control.ino file.
    d. Select Arduino Nano and the correct COM port at the top of the Arduino IDE.

Note: To determine the COM port, unplug the Arduino and then re-plug it to see what new COM port shows up.

    e. Upload the code with the upload button in the top-left corner.
19. Test its basic functionality.
    a. Unscrew the top enclosure to see the back of the LCD.
    b. Adjust the contrast of the LCD by turning the blue trim pot on the back of the LCD with a mini screwdriver (Figure 4A).

Note: From the top left to the bottom right, the LCD should show the internal average temperature (I), target temperature (T), ambient temperature (A), and heating power percentage (O) (Figure 6A).

    c. Place your finger on the temperature sensor to see the increase in temperature reading.
    d. Turn the potentiometer knob clockwise to increase the target temperature.

Note: If the temperature decreases, flip the potentiometer male end on the PCB to change its direction.

20. Test the heating element.
    a. Connect the 24 V power supply barrel jack to the PCB.
    b. Turn on the power switch.
    c. Set the target temperature to a value higher than the room temperature (25°C). The heating element will warm up and start heating.

Tune PID constants

⏲ Timing: 2 h

The Arduino uses Proportional-Integral-Derivative (PID) control to calculate the necessary heating power from the feedback of the temperature sensors. Compared to traditional on-off control, PID control can achieve a precise and stable temperature with minimal overshoots and errors. The Arduino code implements the PID algorithm from the QuickPID library by David Lloyd. As each temperature control setup requires different PID tuning, users must adjust them in this step to configure the PID algorithm properly. After reaching a steady state, the system can achieve stabilities within 0.06°C with extended constants tuning.

In this step, you will adjust the PID constants Kp (proportional), Ki (integral), and Kd (derivative) in the Arduino code through repeated testing and validating the heating curve.

We recommend this post for more information on PID control: https://medium.com/autonomous-robotics/pid-control-85596db59f35.

21. Change the target temperature to room temperature or below.

Note: If the system is heated, allow it to cool to room temperature. This will be the baseline temperature for tuning the PID constants.

22. Start with PID constants Kp, Ki, and Kd at 0.
    a. Change the Kp, Ki, and Kd variables to 0 at line 24 of the Arduino code sketch.
    b. Upload the sketch to run the code.
    c. Open the serial plotter and observe the temperature reading curve.
    d. Change the target temperature to the desired temperature.
    e. Wait for the system to reach the target temperature and let it run for an additional 5 min.
23. Adjust the Kp, Ki, and Kd constants at line 24 of the Arduino code.

⚠ CRITICAL: Ensure to upload the sketch between every adjustment.

    a. Increase Kp until the steady-state error, which is the difference between the target and measured temperatures, is very low.
    b. Increase Ki until the steady-state error is removed entirely.
    c. Increase Kd until oscillations are removed.

Note: An example heating curve of our system is shown in Figure 7A.

EXPECTED OUTCOMES

With the temperature control system introduced in this protocol, users can regulate the temperature of the microscopes, microfluidic chambers, and custom incubators precisely. Adjusting the potentiometer knob allows users to set target temperatures between 25°C and 60°C, while the rear power switch controls system activation. Key parameters, including current temperature (I), target temperature (T), ambient temperature (T), and heating output (O), are conveniently displayed at the top of the enclosure on the LCD.

After reaching a steady state of 37°C, the system achieved a stability of 0.04°C with an error of 0.06°C, showing a very low temperature oscillation (Figure 7B). Furthermore, as demonstrated in Table 1, after setting the target temperature to 25°C–60°C with 5°C intervals, the system achieved stabilities from 0.03°C to 0.06°C, indicating low oscillations across the temperature range.

LIMITATIONS

The system only incorporates heating elements without any cooling elements. Therefore, its function is limited to controlling temperatures above room temperature, which makes certain experiments not feasible. Replacing the polyimide heaters with Peltier modules or adding liquid coolant as a secondary element can achieve a temperature below room temperature, but the system will be drastically more complex with a larger physical footprint.

Another limitation is the temperature sensors are not located next to the analyzed biological samples, so the readings from the sensor may not reflect the actual temperature of the samples. One solution is to measure the sample temperature after reaching equilibrium temperature and compare it to the sensor reading. This difference can then be offset in the code. However, this technique does not guarantee a precise reading throughout the heating curve.

The system is also limited by the temperature sensors' maximum temperature reading error of ± 0.25°C and ± 0.5°C for MCP9808 and DS18B20 respectively. Averaging the error from two sensors indicates that the measured temperature can be ± 0.375°C away from the true

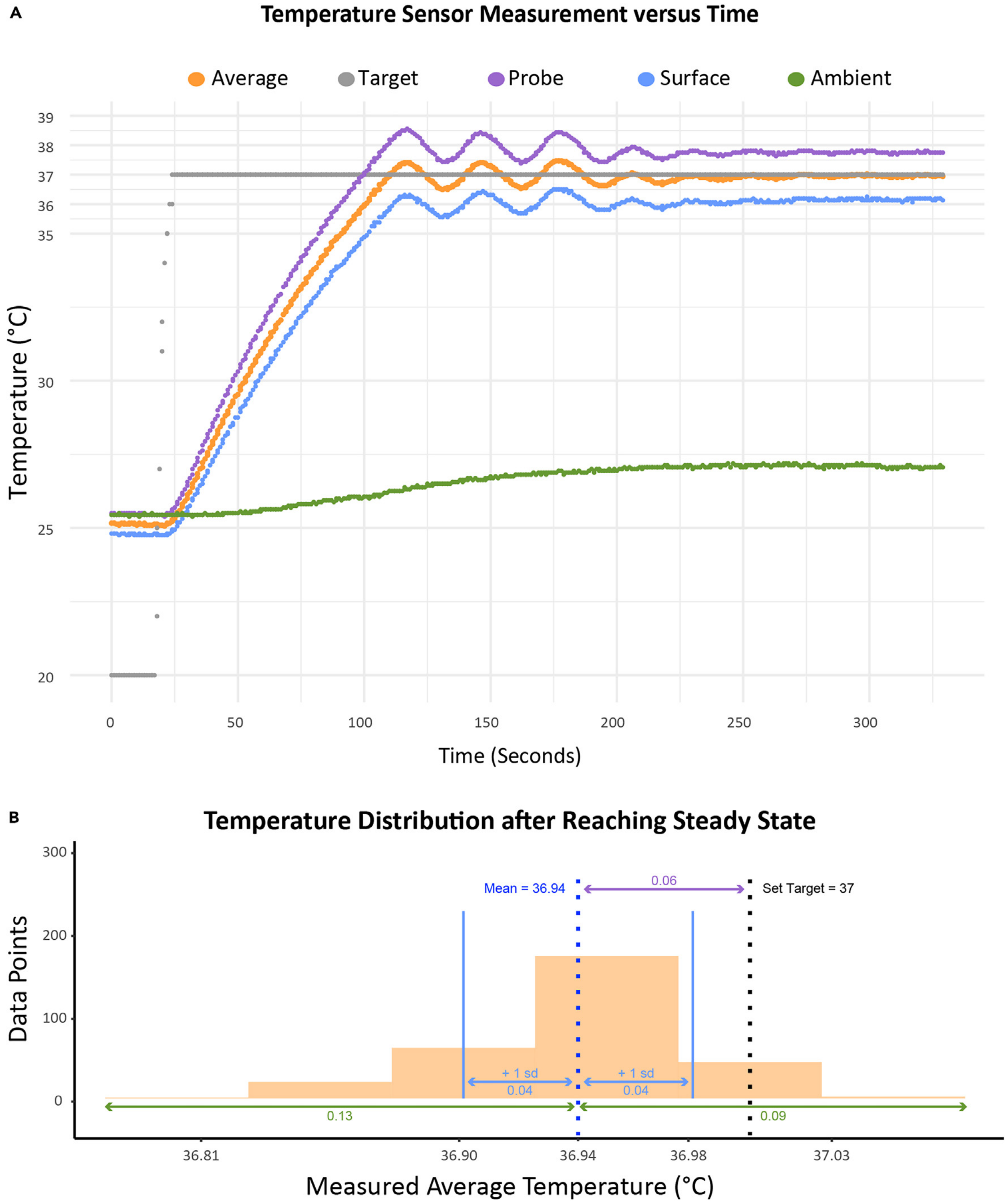


Figure 7. Example PID heating curve and histogram of the temperature control system at 37°C

(A) The heating block reached the target temperature within 110 s and stabilized with ± 0.04°C oscillations within 260 s. The average of the probe and surface temperature is shown in orange, the target temperature set by the user is shown in gray, the probe temperature reading of the heating block is shown in purple, the surface temperature reading of the heating block is shown in blue, and the ambient temperature is shown in green.

(B) The system in a 37°C steady state has a standard deviation (i.e., stability) of 0.04°C and a mean difference (i.e., error) of 0.06°C. This data was collected from 100 data points, or 70 s, for each of the three trials.

temperature. To mitigate this, users can pick sensors with more accurate readings. The readings can also be calibrated by offsetting the temperature reading against a known accurate temperature sensor.

Table 1. From 25°C to 60°C, the system has a stability of 0.03°C–0.06°C and an error of 0.02°C–0.09°C

Target temperature (°C)	Minimum (°C)	Mean (°C)	Maximum (°C)	Stability (°C)	Error (°C)
25	24.937	24.985	25.031	0.0286	0.015
30	29.875	29.966	30.031	0.0305	0.034
35	34.875	34.968	35.031	0.0309	0.032
37	36.812	36.944	37.031	0.0383	0.056
40	39.875	39.955	40.031	0.0343	0.045
45	44.844	44.957	45.031	0.0329	0.043
50	49.812	49.938	50.062	0.0540	0.062
55	54.812	54.945	55.062	0.0462	0.055
60	59.781	59.909	60.031	0.0560	0.091

The data was collected from 200 data points per target temperature. Before 45°C, the PID constants are Kp = 505, Ki = 62, and Kd = 92. At 45°C, the system has Kp = 505, Ki = 75, and Kd = 150. From 50°C to 60°C, the system has Kp = 505, Ki = 120, and Kd = 400.

TROUBLESHOOTING

Problem 1

The LCD does not show any text (related to Step 3, 4, and 16).

Potential solution

If the LCD has no backlight, examine whether the jumper wires to the PCB are correctly oriented. If so, rotate the jumper wire connector and turn on the system to check whether the backlight has been restored. In addition, inspect the solder connections at the following points: LCD headers to jumper wires, female LCD header to PCB, and female Arduino headers to PCB. If the solder appears disconnected, resolder the components.

If the LCD shows a backlight but no text, the LiquidCrystal I2C library may not be working. Restart Arduino IDE and re-upload the sketch. After re-uploading and restarting the system, the LCD should show temperature readings. If the problem persists, reinstall the library in the Library Manager and re-upload the sketch.

Problem 2

The LCD is missing or showing random measurements from a particular temperature sensor or potentiometer (related to Step 1, 3, 5, and 16).

Potential solution

The component corresponding to the random measurement is malfunctioning or disconnected. Double-check the wiring and ensure everything is connected with the correct orientation. The header on the PCB may be incorrectly soldered or the cable may be loose. A continuity test on a multimeter can be performed between each pin of the malfunctioning component and the soldered header pin on the PCB. To ensure the LCD is functional, open the Serial Monitor in Arduino IDE to view the temperature readings. Reconnecting the components should fix the problem; however, if the component remains malfunctional, replace the component.

Problem 3

There is a burning smell or smoke (related to Step 13 and 16).

Potential solution

Turn off the power immediately. The buck converter may not have been adjusted to output 9 V. Therefore, the Arduino received too much voltage at its power input pins. Adjust the buck converter to 9 V and replace the damaged Arduino Nano.

A piece of metal, wire, or solder could have touched and shorted another electrical component, resulting in permanent damage. Locate the malfunctioning component by performing a continuity test with a multimeter and replace the malfunctioning component.

Problem 4

The Serial Monitor on Arduino IDE shows (related to Step 18e):

```
>Complication error: OneWire.h: No such file or directory.
```

Potential solution

This problem happens when uploading the Arduino Code without the installed libraries in the key resources table. Open the Library Manager on Arduino IDE and download any missing libraries. Subsequently, the code should upload seamlessly without encountering further errors.

Problem 5

The LCD reading or Arduino IDE Serial Output freezes with no temperature reading updates (related to Step 19 and 20).

Potential solution

This problem occurs when supplying the Arduino with unreliable power. If the system functions with the Micro-USB cable only, replace the external power supply. If the system functions with the external power supply only, replace the Arduino Nano.

Problem 6

The enclosure does not fit together or with the threaded inserts (related to Step 17).

Potential solution

Due to variations between different 3D printers, the printed enclosure may have slightly different clearance. We have provided the Fusion 360 file on GitHub to facilitate user modifications for proper alignment and fit.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Fangyuan Ding (dingfy@uci.edu).

Technical contact

Technical questions on executing this protocol should be directed to and will be answered by the technical contact, Ziteng Liu (zitenl1@uci.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

The example data and code generated during this study are available at GitHub: https://github.com/UCI-Ding-Lab/Temperature-Control/tree/main. The version of record is archived at Zenodo. DOI at Zenodo: https://doi.org/10.5281/zenodo.10452293.

ACKNOWLEDGMENTS

This work was supported by the NIH Director's New Innovator Award (1DP2GM149554) and Start-up funding from the Department of Biomedical Engineering, University of California, Irvine. We thank

Samantha Sanchez, Howard Yan, and Min Kyu Kim at UC Irvine for their assistance and discussion on the conceptualization and design techniques.

AUTHOR CONTRIBUTIONS

Conceptualization, all authors; methodology and software, Z.L.; supervision, T.T.T.N. and F.D.; writing and editing, all authors.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

1. Cramer, M.N., Gagnon, D., Laitano, O., and Crandall, C.G. (2022). Human temperature regulation under heat stress in health, disease, and injury. Physiol. Rev. *102*, 1907–1989. https://doi.org/10.1152/physrev.00047.2021.

2. Aznauryan, M., Nettels, D., Holla, A., Hofmann, H., and Schuler, B. (2013). Single-molecule spectroscopy of cold denaturation and the temperature-induced collapse of unfolded proteins. J. Am. Chem. Soc. *135*, 14040–14043. https://doi.org/10.1021/ja407009w.

3. Dos-Reis-Delgado, A.A., Carmona-Dominguez, A., Sosa-Avalos, G., Jimenez-Saaib, I.H., Villegas-Cantu, K.E., Gallo-Villanueva, R.C., and Perez-Gonzalez, V.H. (2023). Recent advances and challenges in temperature monitoring and control in microfluidic devices. Electrophoresis *44*, 268–297. https://doi.org/10.1002/elps.202200162.

4. Hoyle, H.W., Stenger, C.M.L., and Przyborski, S.A. (2022). Design considerations of benchtop fluid flow bioreactors for bio-engineered tissue equivalents in vitro. Biomater. Biosyst. *8*, 100063. https://doi.org/10.1016/j.bbiosy.2022.100063.

5. Zhdanov, A., Keefe, J., Franco-Waite, L., Konnaiyan, K.R., and Pyayt, A. (2018). Mobile phone based ELISA (MELISA). Biosens. Bioelectron. *103*, 138–142. https://doi.org/10.1016/j.bios.2017.12.033.

6. Zhdanov, A., Chen, C., Schmitt, C., Tripp, T., Miller, C., and Pyayt, A. (2022). Mobile ELISA with a transparent incubator. Methods Mol. Biol. *2393*, 329–341. https://doi.org/10.1007/978-1-0716-1803-5_18.

7. Chamani, F., Barnett, I., Pyle, M., Shrestha, T., and Prakash, P. (2022). A review of in vitro instrumentation platforms for evaluating thermal therapies in experimental cell culture models. Crit. Rev. Biomed. Eng. *50*, 39–67. https://doi.org/10.1615/CritRevBiomedEng.2022043455.

8. Ahl, D., Eriksson, O., Sedin, J., Seignez, C., Schwan, E., Kreuger, J., Christoffersson, G., and Phillipson, M. (2019). Turning up the heat: Local temperature control during in vivo imaging of immune cells. Front. Immunol. *10*, 2036. https://doi.org/10.3389/fimmu.2019.02036.

9. Iorizzo, T.W., Jermain, P.R., Salomatina, E., Muzikansky, A., and Yaroslavsky, A.N. (2021). Temperature induced changes in the optical properties of skin in vivo. Sci. Rep. *11*, 754. https://doi.org/10.1038/s41598-020-80254-9.

10. Liu, W., Chen, L., Yin, D., Yang, Z., Feng, J., Sun, Q., Lai, L., and Guo, X. (2023). Visualizing single-molecule conformational transition and binding dynamics of intrinsically disordered proteins. Nat. Commun. *14*, 5203. https://doi.org/10.1038/s41467-023-41018-x.

11. Arcus, V.L., and Mulholland, A.J. (2020). Temperature, dynamics, and enzyme-catalyzed reaction rates. Annu. Rev. Biophys. *49*, 163–180. https://doi.org/10.1146/annurev-biophys-121219-081520.

12. Chamani, F., Pyle, M.M., Shrestha, T.B., Sebek, J., Bossmann, S.H., Basel, M.T., Sheth, R.A., and Prakash, P. (2023). In vitro measurement and mathematical modeling of thermally-induced injury in pancreatic cancer cells. Cancers *15*, 655. https://doi.org/10.3390/cancers15030655.

13. Åström, K.J., and Murray, R.M. (2021). Feedback Systems (Princeton University Press).

14. Wang, N., Liu, Z.X., Ding, C., Zhang, J.-N., Sui, G.-R., Jia, H.-Z., and Gao, X.-M. (2022). High Efficiency Thermoelectric Temperature Control System With Improved Proportional Integral Differential Algorithm Using Energy Feedback Technique. IEEE Trans. Ind. Electron. *69*, 5225–5234. https://doi.org/10.1109/TIE.2021.3082462.

15. Dulău, M. (2019). Experiments on Temperature Control Using On-Off Algorithm Combined with PID Algorithm. Acta Marisiensis. Seria Technologica *16*, 5–9. https://doi.org/10.2478/amset-2019-0001.

16. dos Reis, W.P.N., Rodrigues, D.P., Rocha, I.L., Dutra, R.S., and Souza, P.V.S. (2021). An Arduino-based experiment designed to estimate the heat capacity of a nichrome resistor. Phys. Educ. *56*, 013005. https://doi.org/10.1088/1361-6552/abc79a.

17. Ding, F., Cocco, S., Raj, S., Manosas, M., Nguyen, T.T.T., Spiering, M.M., Bensimon, D., Allemand, J.-F., and Croquette, V. (2022). Displacement and dissociation of oligonucleotides during DNA hairpin closure under strain. Nucleic Acids Res. *50*, 12082–12093. https://doi.org/10.1093/nar/gkac1113.