Project Deliverable H: Prototype III and Customer Feedback

Mataab Bashir-Mustafa, Khadijetou Coulibaly, Sean A. Fitzgerald, Callum Mcdougall, Miguel Sandri, Ana Sinna

November 24, 2024

Table of Contents

Introduction:

The Prototype Test Plan III is an important step in developing the bat box mechanism. Building on the results of Prototype II, the emphasis switches to verifying individual system components and their integration to assure functionality and dependability in real-world scenarios. Key areas for development include improving sensor detection accuracy, assuring reliable data logging and retrieval processes, optimizing battery performance, and reducing environmental interference. This iteration prioritizes addressing unresolved issues from previous prototypes, such as wiring inconsistencies and code inefficiencies, while also introducing changes such as improved sensor alignment procedures, battery estimation methods, and usability and durability refinements based on feedback. These focused tests are intended to better align the prototype with the project's goals, paving the way for a final design that is dependable, user-friendly, and field-deployable.

Prototype test plan III

Test ID	Test Objective (Why)	Description of Prototype and Basic Test Method (What)	Description of Results to be Recorded and Usage (How)	Estimated Test Duration and Start Date (When)
1	Validate laser sensor's detection accuracy	Test laser sensor with controlled interruptions to simulate bats entering/exiting the box. Verify sensor detects each interruption accurately.	Record detection count and time stamps. This will establish baseline accuracy for bat visit counts and allow adjustments for sensitivity.	2 hours, Start on 11/07
2	Confirm Arduino data logging	Test data logging functionality by recording sensor data to the SD card in real time, ensuring reliable storage of visit counts and time stamps.	Record logging frequency and ensure accurate time stamps are saved for each detection event. This verifies that all bat visits are documented without data loss.	2 hours, Start on 11/07

4	Assess battery life under prototype load	Run prototype continuously to measure battery duration with the laser sensor and Arduino active, to predict maintenance frequency for field deployment.	Record power levels and track when the battery drops below operational limits. Key to estimating how long the device can function autonomously.	24 hours, Start on 11/08
5	Test for interference from environmental factors	Expose the prototype to low-light conditions and simulated environmental factors (e.g., wind, insects) to check for false detections.	Record instances of false detections and correlate with environmental conditions to adjust sensor sensitivity. Ensures accurate bat counting in real-world settings.	6 hours, Start on MM/DD
6 *	Prototype assembly and integration	Assemble the laser sensor, RTC, SD card module, and Arduino to confirm integration and ease of assembly. Test that components fit securely within the bat box enclosure.	Document assembly steps and note any fitting issues. This will highlight potential redesign needs for better integration and ease of maintenance.	2 hours, Start on MM/DD
7	Evaluate data retrieval and storage efficiency	Test reading data from the SD card by retrieving recorded visit data to assess ease of data handling.	Record time and reliability of data retrieval. This will help ensure efficient data handling for further analysis of bat visit patterns.	2 hours, Start on 11/07
8	Verify laser sensor’s range and alignment	Measure the sensor’s range and alignment accuracy to ensure consistent detection of bat entries and exits without interference from the box edges.	Document the detection range and note alignment issues. Essential to ensure the sensor captures only true bat visits without extraneous triggers.	2 hours, Start on 11/07

10 *	Soldering practice and component resilience	Practice soldering connections for laser sensor, RTC, and SD card module to ensure strong, durable connections.	Inspect soldering quality under light stress and verify connection stability. Essential to prevent disconnections in field conditions, improving durability.	3 hours, Start on MM/DD
11	Collect user feedback on usability	Gather feedback from potential users on ease of use, data retrieval, and overall functionality.	Document user suggestions and identify areas for improvement. User insights will guide final adjustments for usability in the field.	2 hours, Start on MM/DD

*need to get the new problem of the Arduino board settled

1. Experimental Method

Test	What	Analytical, numerical or experimental model
Validate laser sensor's detection accuracy+ range and alignment	Test laser sensor with controlled interruptions to simulate bats entering/exiting the box. Verify sensor detects each interruption accurately.	Set up a test environment with objects placed at known distances from the sensor. Measure the actual amount of triggers depending on the distance and type of movement and speed. Calculate the error % per distance.
Confirm Arduino data logging	Test data logging functionality by recording sensor data to the SD card in real time, ensuring reliable storage of visit counts and time stamps.	Run the system in real-world conditions and verify that data is being logged. Connect the Arduino to check real time recording
Evaluate data retrieval and storage efficiency	Test reading data from the SD card by retrieving recorded visit data to assess ease of data handling.	Connect the Arduino to the SD card and check whether the data is being saved correctly by unplugging it and connecting the SD card to the computer will also help us

		know the type of output we get.
Assess battery life under prototype load	Run prototype continuously to measure battery duration with the laser sensor and Arduino active, to predict maintenance frequency for field deployment.	Using a multimeter to assess power consumption of the circuit and use the average mA of our battery to make our calculations
Test for interference from environmental factors	Expose the prototype to low-light conditions and simulated environmental factors (e.g., wind, insects) to check for false detections.	Test the laser and sensor under different light conditions.
Collect user feedback on usability	Gather feedback from potential users on ease of use, data retrieval, and overall functionality.	Ask different people (ideally with no previous knowledge) about the different criteria

2.2Results and Problems

Test	Results
Validate laser sensor's detection accuracy+ range and alignment	The detection problems from the previous prototypes were solved as we no longer get triggers that are not actual ones. We also worked on the triggering that seemed to be continuous and happened no matter what by working on our code. We also figured out some of the wiring problems. So this is a great advancement compared to our last prototype. We managed to set the minimal trigger time. However, the Arduino's 3.3V pin stopped working right before assembly resulting in a setback. (see annex for pictures). The range and alignment tests allowed us to test for the maximum distance the sensor and laser will be apart (the width of the box and slightly further in case another box is used) and the sensor worked as intended. The alignment needs to be precise otherwise no counts will be logged and the orientation of the receptor is extremely important as it could burnout if the wrong side is oriented towards the laser.
Confirm Arduino data logging Evaluate data retrieval and storage efficiency	We successfully wired the SD card to the Arduino board and it is now able to log data based of off our code (see annex for both the successful logging and the new code)
Assess battery life under prototype load	Multimeter tested our entire circuit with all the components plugged in. The multimeter showed a draw of 2.86mA. There is not on the manufacturer's site a detailed and precise

	value for the number of mAh but an internet showed that on average a 9V alkaline battery is 400-600 mAh and so our battery is either: $\frac{400}{2.86} = 140h \; or \frac{600}{2.86} = 210h$. If we want our battery to last for a month (and a few extra days for safety say 35 days) 24*35=840h and so it means that we would need 2400 mAh at least (based on the 400mAh battery).
Test for interference from environmental factors	The sensor does work when the lights are on so tracking shouldn't be a problem if there's a bit more light it didn't seem to affect the overall accuracy. The code requires a minimal amount of time to register a trigger which allows to not account for things like guano since the trigger time would be much smaller.
Collect user feedback on usability	See section three for feedbacks and comments

3. Feedback & Comments

Target Audience:

Family and Friends (Non-technical): Chosen for testing ease of use and intuitiveness of the prototype.
Environmental Science Enthusiast Group (Potential End Users): Selected for their familiarity with wildlife monitoring devices.
Local Bat Conservation Team (Clients): Experts in the field, providing insights on functionality and long-term usability.

Feedback Channels:

In-person Demonstrations: Conducted with family and friends for direct interaction and immediate feedback.
Online Survey: Shared with the enthusiast group to gather structured input on specific usability and functionality criteria.
Client Meeting: Held with the bat conservation team to discuss technical details and prototype performance in real-world conditions.

Feedback Questions

Ease of Use:

Was the prototype easy to set up and operate?

Were the instructions clear and easy to follow?

Functionality:
3. Did the prototype perform as expected? (e.g., accurate bat count logging, reliable data storage)
4. Were there any issues with the laser sensor's alignment or sensitivity?

Design:
5. Was the design compact and field-deployable?
6. Were there any concerns about durability or battery life?

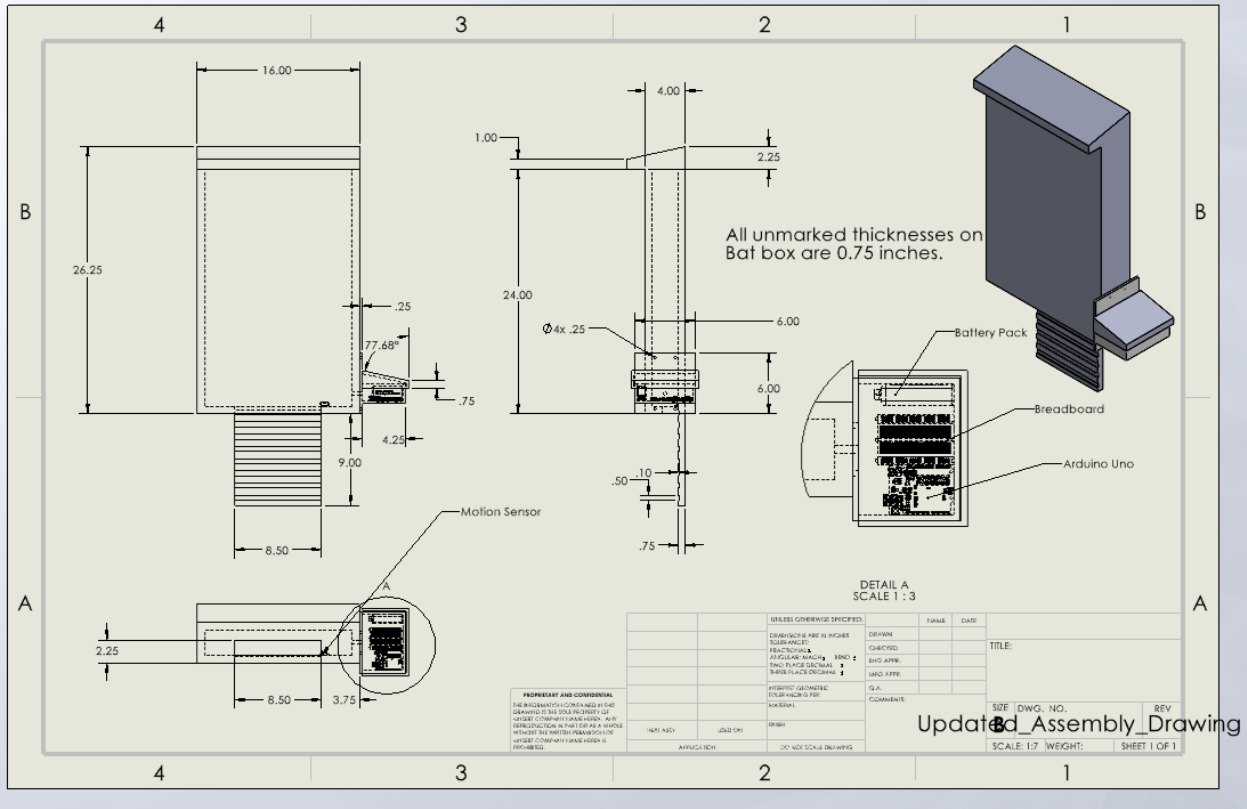
General Suggestions:
7. Are there additional features or improvements you would like to see?

Feedback Results

User Group	Positive Feedback	Suggestions
Family and Friends	- Easy to understand setup process.	- Include a visual alignment guide for the laser sensor.
	- SD card data retrieval was straightforward and user-friendly.	- Consider adding a notification system (e.g., LED light) to indicate successful logging.
Enthusiast Group	- Compact design ideal for field applications.	- Battery life of ~140-210 hours may require external backup options for extended deployments.
Conservation Team	- Data logging was reliable, and minimal false detections under light interference.	- Ensure the sensor and laser alignment are robust to prevent missed counts in high-activity areas.
	- Good progress in addressing prior wiring issues.	- Provide weatherproofing for outdoor use.

4.Updated BOM and design

Bill of Materials				
Item #	Item Description	Quantity	Unit Price	Amount
1	Micro SD Card Module	1	$ 10.98	$ 12.41
2	Battery 3.7V - 150mAh	1	$ 4.00	$ 4.52
3	Arduino Datalogger	1	$ 9.99	$ 11.29
4	Arduino	1	$ 15.25	$ 17.23
5	Laser Sensor	1	$ 15.99	$ 18.07
6	Micro SD Card 64GB, SD Adapter	1	$ 13.99	$ 15.81
7	Bread Board	1	$ 5.00	$ 5.65
8	Battery Connector 9V	1	$ 2.00	$ 2.26
9	Cord	1	$ 2.75	$ 3.11
10	20cm Male-Male Jumper Cables	10	$ 1.00	$ 1.13
11	USB Wall Adapter	1	$ -	$ -
12	Screws	1	$ -	$ -
13	Wood	1	$ -	$ -
14	9V Battery Clip Connector	1	$ 2.00	$ 2.00
15	9V Alkaline Batteries	1	$ 4.00	$ 4.00
			Total	$ 97.47



Conclusion:

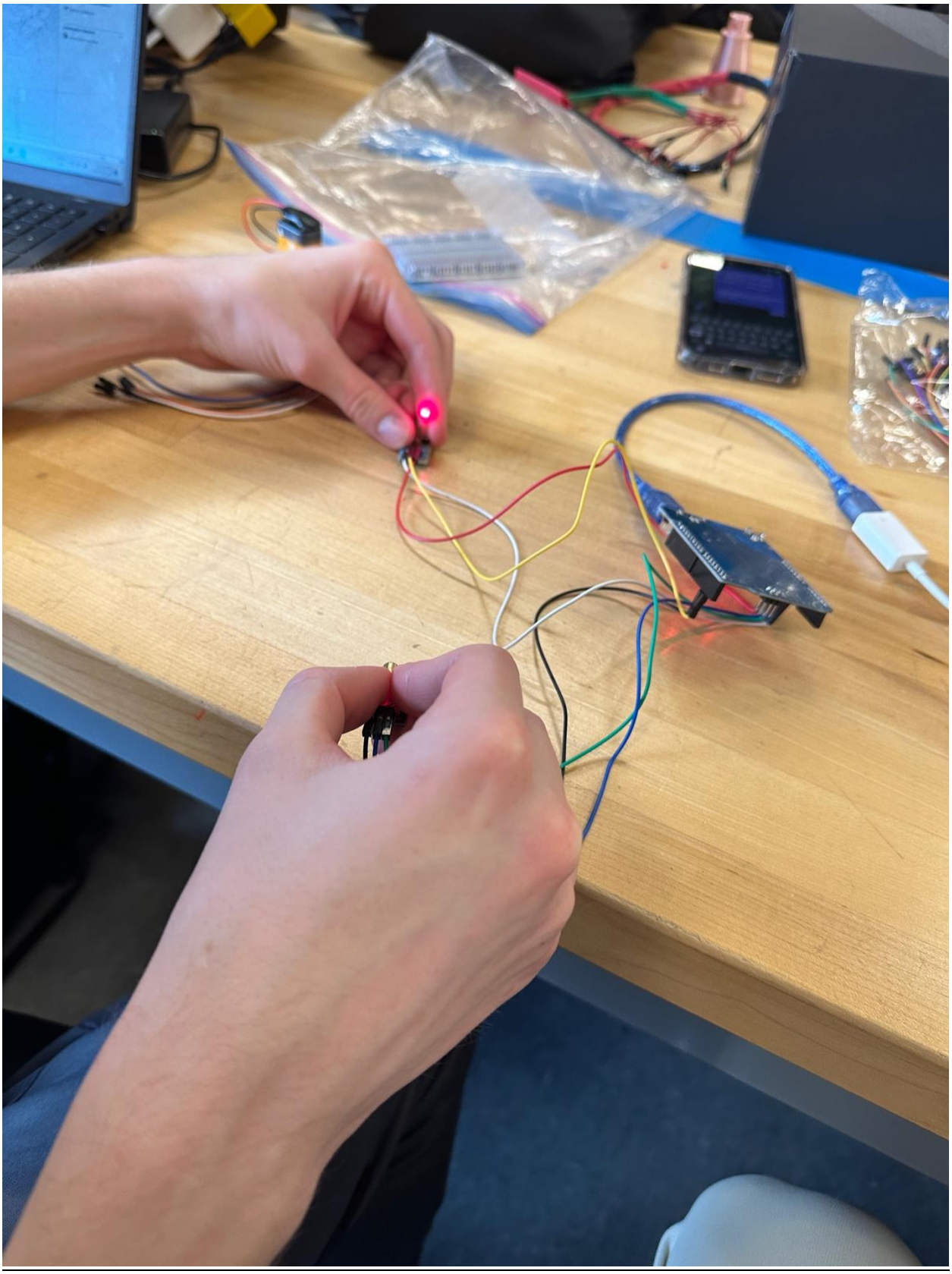
The Prototype III testing results show considerable improvements in sensor alignment precision, fewer false detections, and better data logging capabilities. Changes to the coding and wiring solved key functional concerns discovered in earlier prototypes, allowing for more precise tracking of bat activity. For example, using a short trigger period substantially avoided false positives produced by minor environmental interferences. The battery life study offered useful insights into power requirements, validating the necessity for larger-capacity batteries for long-term deployments. Feedback from conservation teams and potential users emphasized the significance of usability and environmental resilience, which will guide future improvements such as weatherproofing and simpler alignment tools.

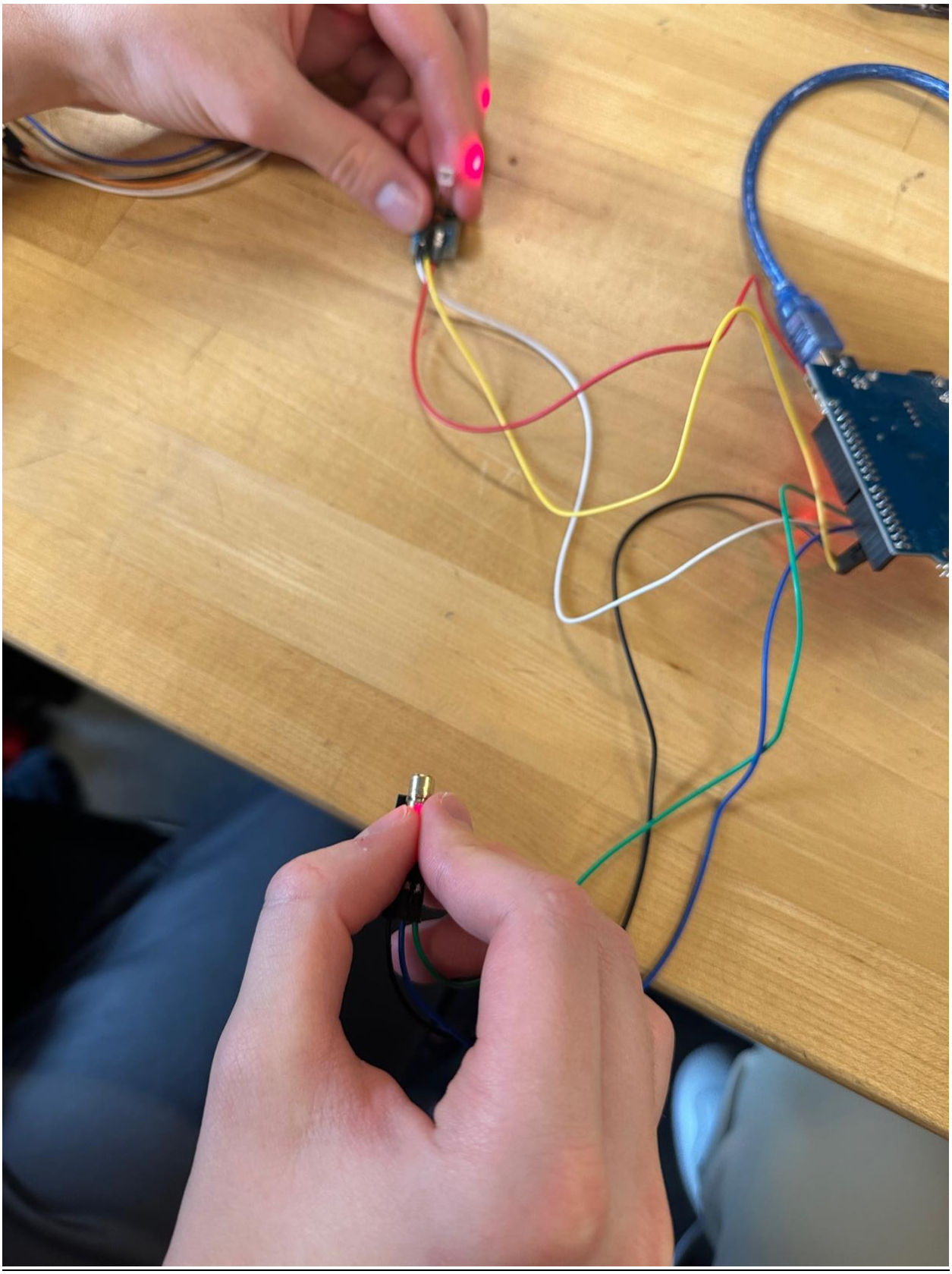
Looking ahead, the emphasis will move to incorporating these results into a unified design. Future generations will investigate ways to increase battery life using alternate power sources, improve the robustness of soldered connections, and incorporate additional measurements such as temperature and humidity for wider ecological insights. Final testing will confirm design cohesiveness and validate alignment with the overall objective of delivering an accurate, long-lasting, and user-friendly tool for bat population monitoring. These findings will

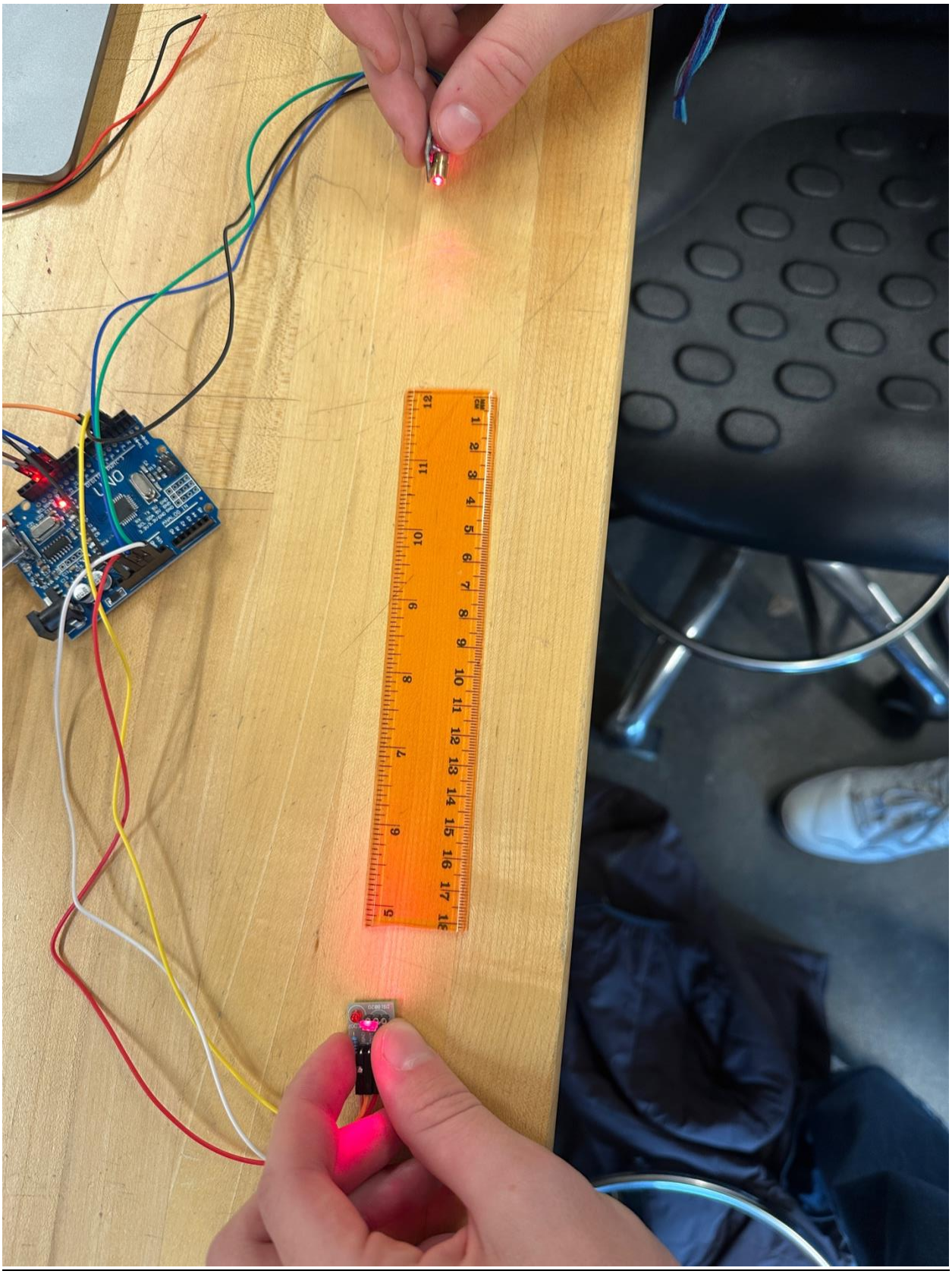
assist ensure that the final prototype is prepared for effective field deployment and data collecting.

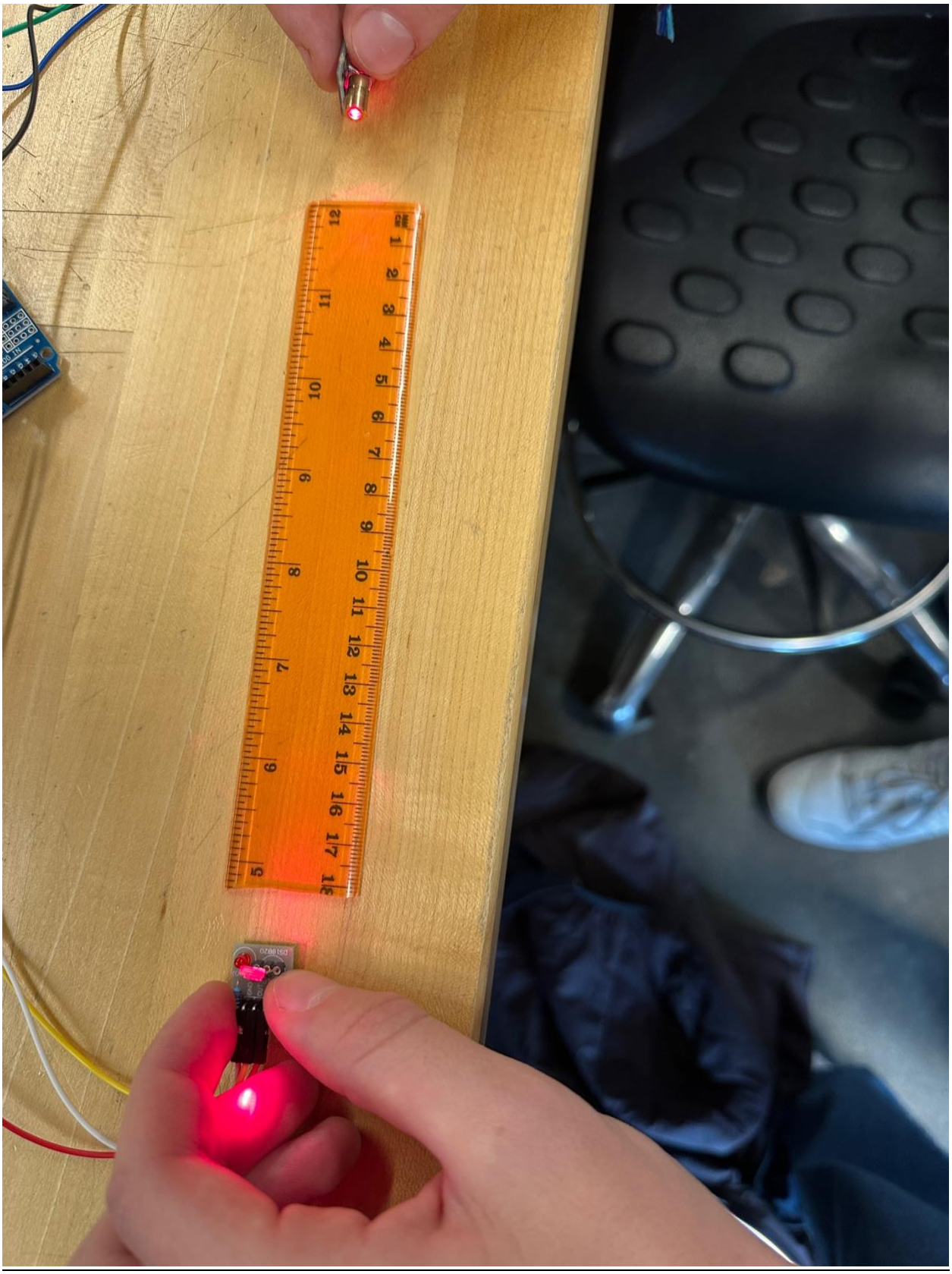
ANNEX

laser sensor’s detection accuracy+ range and alignment & interference from environmental factors

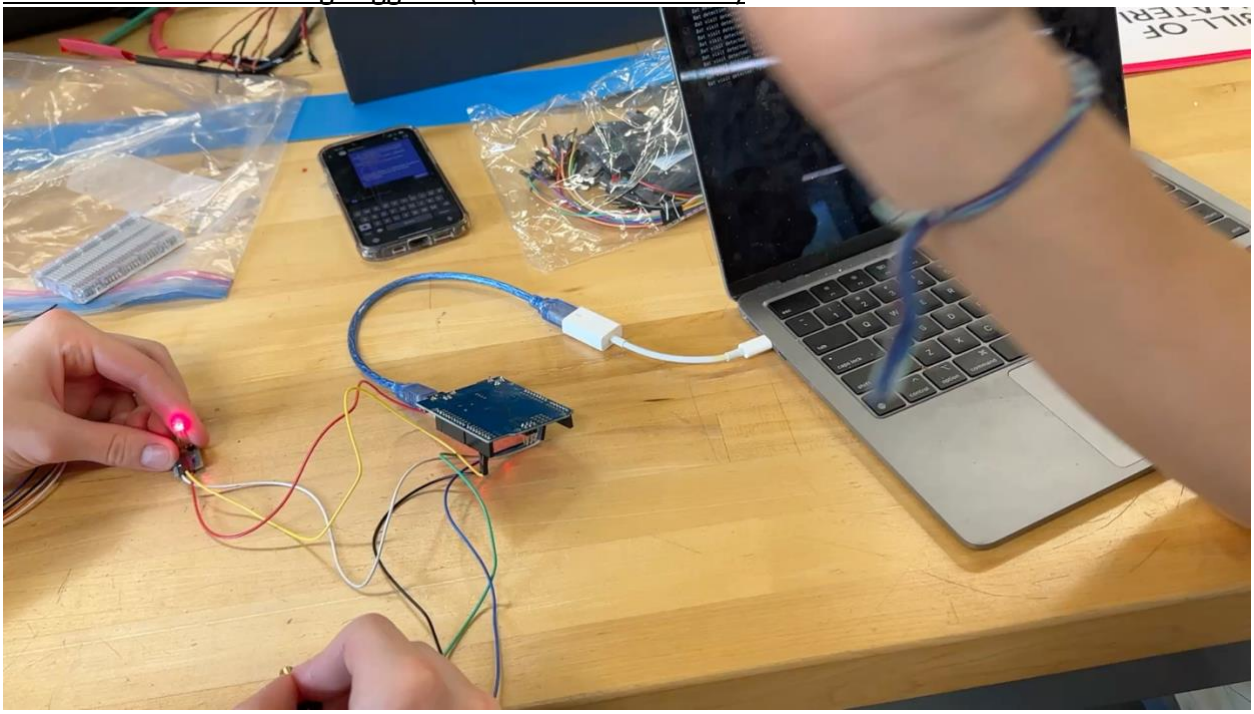




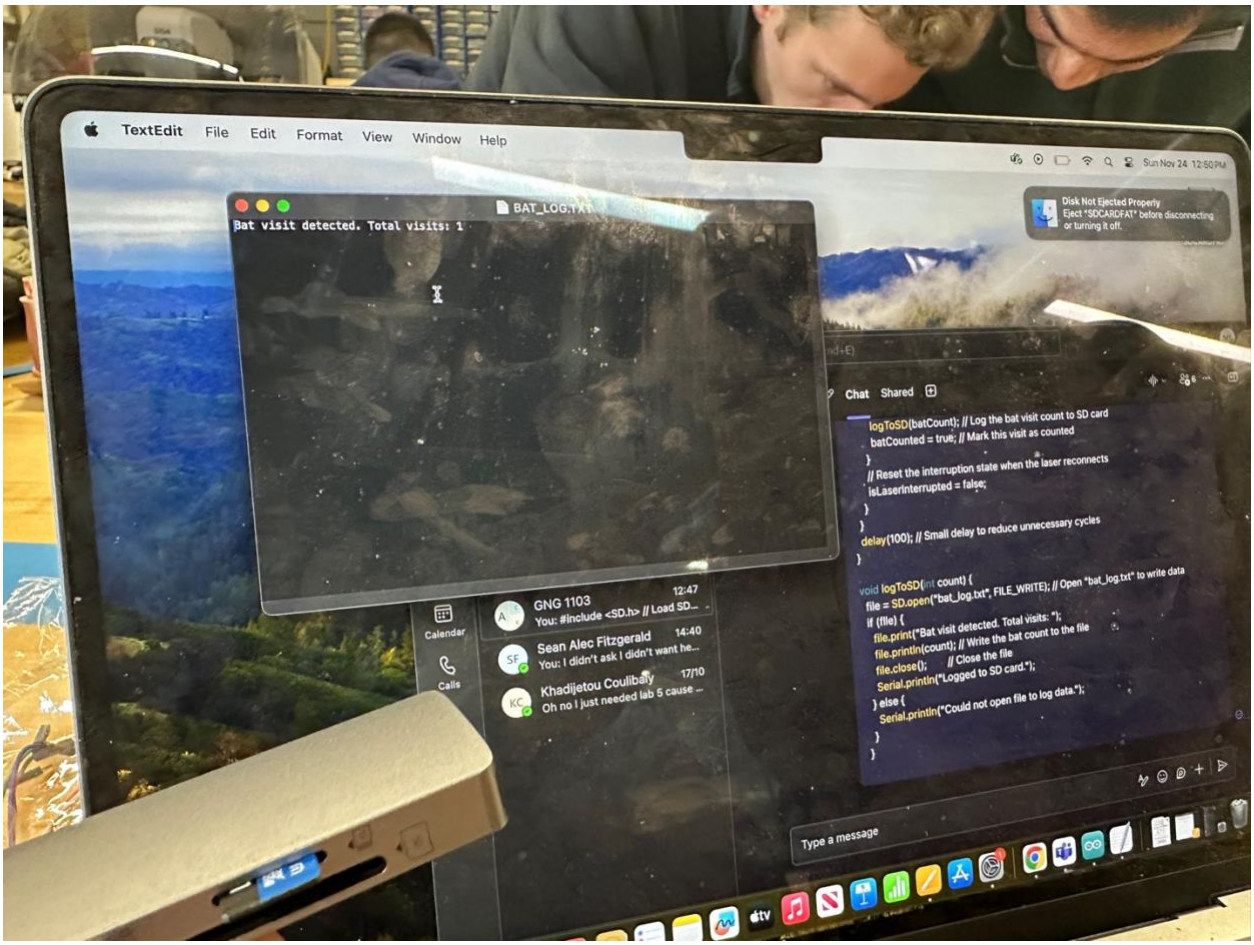




Video of the sensor being triggered (double click to view)



Confirm Arduino data logging Evaluate data retrieval and storage efficiency + new code

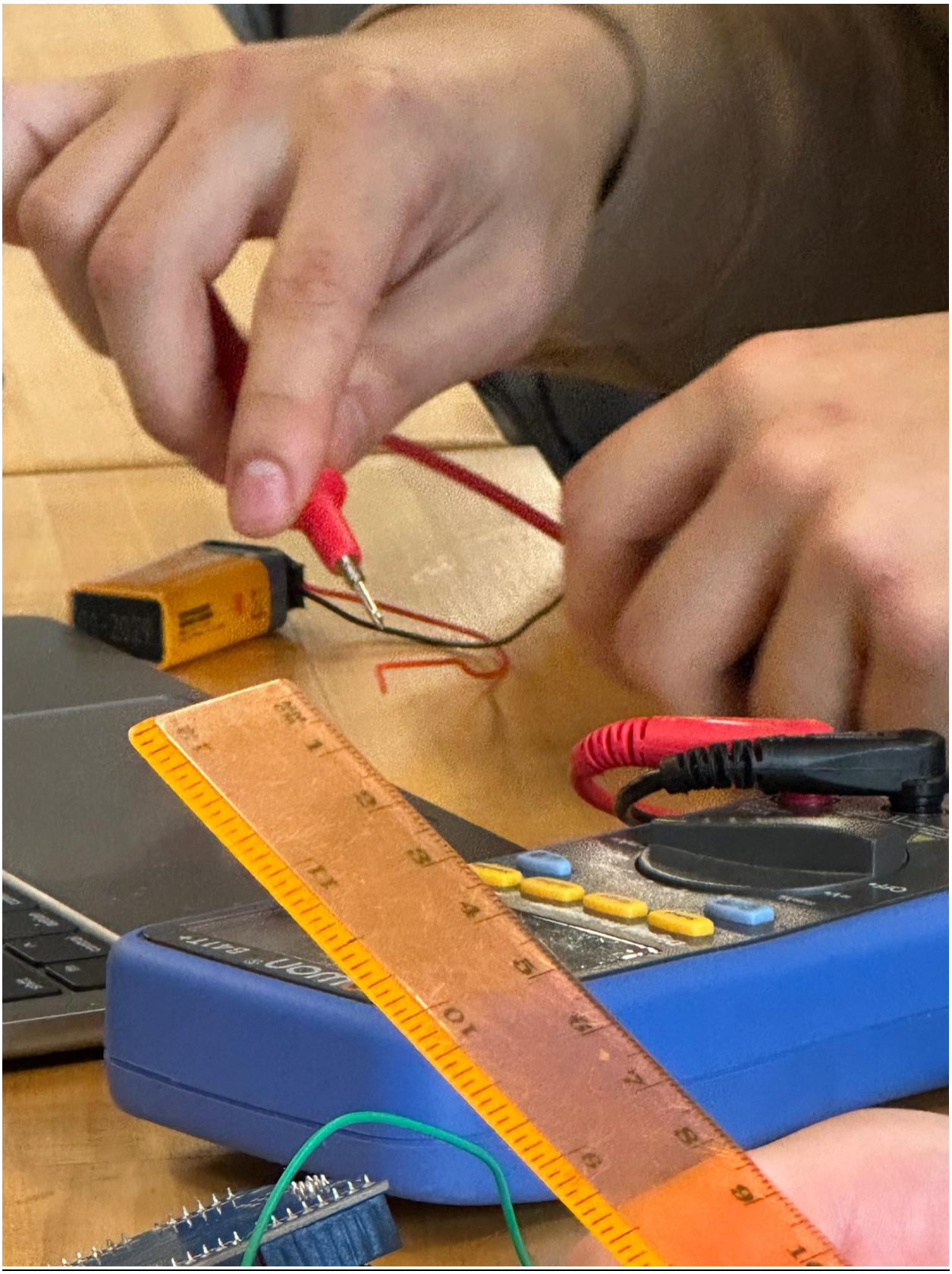



```
#include <SD.h> // Load SD library

const int laserTransmitterPin = 3;  // Pin for the laser transmitter
const int laserReceiverPin = 2;    // Pin for the laser receiver
const int chipSelect = 4;          // Chip select pin for the SD card module

const unsigned long interruptionDuration = 2000; // Minimum time (2 seconds) to count as a bat visit

unsigned long interruptionStartTime = 0; // Tracks when the interruption started
bool isLaserInterrupted = false;        // Tracks whether the laser is currently interrupted
bool batCounted = false;                // Tracks if the bat visit has been counted

int batCount = 0; // Tracks the number of bats detected
File file;        // File object for SD card operations

void setup() {
```

```
  Serial.begin(9600);

  pinMode(laserTransmitterPin, OUTPUT); // Set laser transmitter as output
  pinMode(laserReceiverPin, INPUT);    // Set laser receiver as input
  pinMode(chipSelect, OUTPUT);         // Chip select pin must be set as output

  digitalWrite(laserTransmitterPin, HIGH); // Turn the laser ON

  // Initialize SD card
  if (!SD.begin(chipSelect)) {
    Serial.println("Could not initialize SD card.");
    while (true); // Halt execution if SD card initialization fails
  }

  // Check if the file already exists and delete it if necessary
  if (SD.exists("bat_log.txt")) {
    Serial.println("File exists. Removing it.");
    if (SD.remove("bat_log.txt")) {
      Serial.println("Successfully removed existing file.");
    } else {
      Serial.println("Could not remove existing file.");
    }
  }

  Serial.println("Bat detection system initialized.");
}

void loop() {
  unsigned long currentTime = millis();
  int sensorState = digitalRead(laserReceiverPin); // Read the state of the laser receiver

  if (sensorState == HIGH) { // Laser is interrupted
    if (!isLaserInterrupted) { // First detection of interruption
      isLaserInterrupted = true;
      batCounted = false; // Reset the bat counted flag for a new interruption
      interruptionStartTime = currentTime; // Record the start time of the interruption
    }
```

```
  } else { // Laser is not interrupted
    if (isLaserInterrupted) { // Laser was previously interrupted
      if (!batCounted && (currentTime - interruptionStartTime >= interruptionDuration)) {
        // If the interruption lasted long enough and has not been counted yet
        batCount++;
        Serial.print("Bat visit detected! Total visits: ");
        Serial.println(batCount);
        logToSD(batCount); // Log the bat visit count to SD card
        batCounted = true; // Mark this visit as counted
      }
      // Reset the interruption state when the laser reconnects
      isLaserInterrupted = false;
    }
  }
  delay(100); // Small delay to reduce unnecessary cycles
}

void logToSD(int count) {
  file = SD.open("bat_log.txt", FILE_WRITE); // Open "bat_log.txt" to write data
  if (file) {
    file.print("Bat visit detected. Total visits: ");
    file.println(count); // Write the bat count to the file
    file.close();        // Close the file
    Serial.println("Logged to SD card.");
  } else {
    Serial.println("Could not open file to log data.");
  }
}
```

Assess battery life under prototype load




owon® B41T+
1.883
True RMS
Digital Multimeter
Select
Range
Hz/Duty
Max/Min
Hz%
TEMP
•µA≈
•mA≈
•A≈
≈mV•
≈V•
OFF•
20A
Max
20A/250V
Each 15min Max 10sec
Max
750V⎓
1000V≂
Max
220mA/250V
1000V CAT III