Project Deliverable F: Prototype I and Customer Feedback

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

November 03, 2024

Table of Contents

Introduction: *3*

1 ..... *3*

2. A simple analysis of critical components *3*

3. Prototype test plan for Prototype I *4*

4. Feedback and comments *5*

5. Bill of materials and design update *6*

Bill of material **6**

Design **7**

6. Prototype test plan *7*

Conclusion: *10*

ANNEX *11*

Arduino code **11**

Circuit **14**

Box Construction and dimensions **15**

DELIVERABLE F

Introduction:

This report aims to document the creation and testing of our first prototype, which is designed to meet the initial goals outlined in our previous project plan. This prototype acts as a proof of concept, tackling major challenges related to [specific goal, e.g., stability, functionality, etc.]. It represents a concentrated effort to explore the essential elements of our design and identify areas that may require further development. Through testing this prototype, we plan to gather important data and feedback that will guide the next phases of development. Eventually, this prototype is intended to engage stakeholders, display our initial advancements, and emphasize key areas for enhancement prior to developing a more refined version.

1.

See Annex and part 3

2. A simple analysis of critical components

The critical component of our project is the circuit which involves the Arduino UNO R3, the code, the sensor, the battery, the micro-SD as well as the wires and the micro SD module. Connex components is the electronics compartment.

Component	Function	Aspects to consider
Arduino UNO R3	Acts as the main controller for the system. It processes data from the sensor, controls the data logging to the SD card, and manages power distribution	Power requirements
Code	The code is essential for our sensor to function with our Arduino but also to collect the data so it can be stored on the SD-Card	It must fulfill two functions
Sensor	Center piece of the project it will allow us to track the bats	Accuracy, precision, triggering rate
Battery	Will power the Arduino and all the other components through it	Voltage and current rating Battery life

Micro-SD	Will allow to collect the data collected	Capacity Formatting (it needs to be FAT32)
Wires + micro-sd module	Connecting the different components together and making a compatible interface to connect them.	Compatibility with different ports
Electronics compartment	Protecting the electronics while allowing for easy maintenance since it will be located outside there won't be any need for the bats to be disturbed when retrieving data	It needs to be weatherproof and any special maintenance should be taken into consideration

3. Prototype test plan for Prototype I

No quantitative tests were done during prototype I as some of the tests needed to be postponed for prototype II because our bill of material wasn't adequate.

Test	Why	What , How, When	Results
Arduino code	It is what will help us process and collect data since the sensor cand detect but not interpret the trigger with a number or communicate it has been triggered if it wasn't being observed.	The test was done by running the code as we went to check. It will need to be updated to integrate the logger. 2h	See annex for screenshots. The Arduino code doesn't seem to have any problem we now know that it can used.
Logger code	Abandoned see results.		Upon further research the Arduino code can be used to save data and read the sensor triggering. We will need to know how to load the codes onto the Arduino and so

			the microcontroller might still be used we will have to seek the counsel of our TAs.
Circuit	To test the compatibility and type of circuit that we'll have to make with our electronics also ensuring power distribution and needs	We decided to use TinkerCAD to have a visualization of what we will have. 1h30	See annex. It has helped us identify that we might not need our logger and think about all we'll need for the circuit and different aspects we did not consider
Disposition of the component	Assess the placement and environmental impact on the prototype's performance and evaluate whether a different disposition would be more convenient	The feedback from the TAs was used to determine what would be the most judicious placement of the different components. 1h	Our design has been updated, allowing us to not only have an optimized design but also fullfill one of our major needs which is that bat movement shouldn't be impeded while protecting the equipment.

4. Feedback and comments

Power Source Update – TA feedback

Initially, we chose a lithium-ion battery to power our electronic devices. However, comments identified some issues with this decision. For starters, lithium-ion batteries require a specialized charger, which may complicate maintenance for end users and restrict their usefulness. Furthermore, the output voltage of the lithium-ion battery was insufficient to efficiently power our complete system, posing a danger of component underperformance or instability. In response, we used a 9V battery as our new power source. This battery delivers the voltage required to run the system reliably and is a more accessible power source for end users. The 9V battery also has the advantage of being easy to replace, allowing for rapid swaps when needed. This improvement improves the power system's usability and consistency, resulting in a better user experience and more reliable prototype performance.

Accessibility of Electronics – Peer feedback

For our prototype, we first housed the electronics within the main box, which caused various accessibility concerns. With this design, it was difficult to reach the electronics for routine chores like battery replacement, data retrieval, and maintenance. This limited access may result in lengthier downtimes and more effort required for normal maintenance. To increase access, we opted to place a separate, weatherproof box on the exterior of the main box to contain all electronics except the sensors, which must remain inside to work properly. This external box configuration allows electronics to be rapidly accessible without disturbing the main system, easing chores like as battery replacement and data collecting. The waterproof housing also protects the electronics from environmental conditions, extending their life and performance. This change enhances the entire user experience by making the electronics far more controllable and maintainable.

Data Logger Enhancement – TA feedback

Our initial choice of data logger was identified as a possible barrier owing to its programming complexity. It needed a large amount of setup time, which might delay the integration of the data recording component and perhaps slow down the project timetable. The programming limitations of the data logger may also limit our ability to effectively test and develop the system. Based on this input, we decided to use an alternate data logger with which one of our team members has substantial expertise. This improvement not only speeds up the integration process, but also shortens the learning curve for our team, allowing us to operate more comfortably with the new data logger. By using a familiar and easier-to-program data logger, we eliminate potential delays and ensure that our data recording component meets project deadlines without sacrificing functionality.

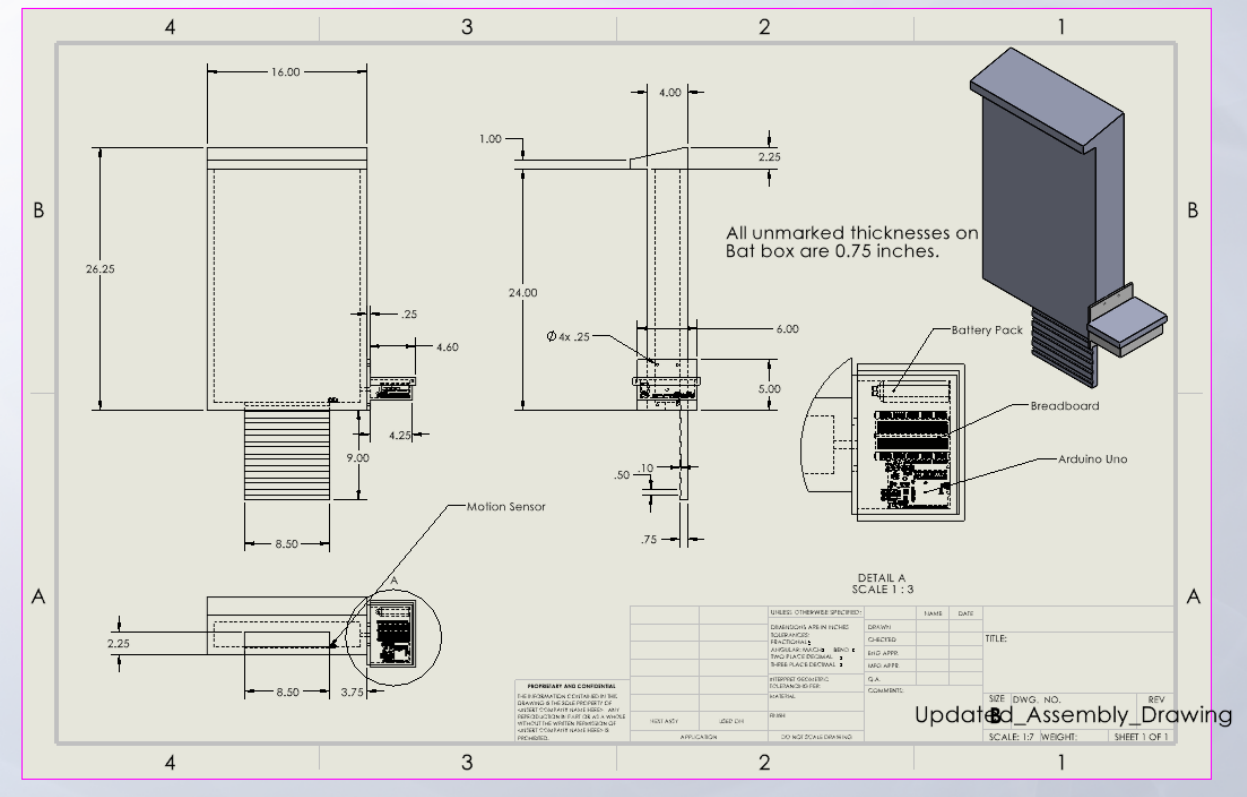
5. Bill of materials and design update

Bill of material

Bill of Materials				
Item #	Item Description	Quantity	Unit Price	Amount
1	Glue Adhesive	1	$ 6.84	$ 7.73
2	Battery 9V	1	$ 4.00	$ 4.52
3	Arduino	1	$ 15.25	$ 17.23
4	Laser Sensor	1	$ 15.99	$ 18.07
5	Micro SD Card 128GB	1	$ 17.99	$ 20.33
6	Bread Board	1	$ 5.00	$ 5.65
7	Battery Connector 9V	1	$ 2.00	$ 2.26
8	Cord	1	$ 2.75	$ 3.11
9	USB Wall Adapter	1	$ -	$ -
10	Screws	1	$ -	$ -
11	Wood	1	$ -	$ -

12	Micro SD module	1	$ 10.98	$ 12.41
			Total	$ 91.31

Design



6. Prototype test plan

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
3	Check RTC functionality for timestamp accuracy	Test RTC module for maintaining accurate time while Arduino is powered and unpowered to ensure reliable time stamps.	Compare recorded time stamps with actual time over a 24-hour period. Ensures that each bat visit is accurately time-marked, critical for tracking visit patterns.	24 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
9 *	Test for enclosure protection in outdoor settings	Simulate outdoor conditions (e.g., humidity, dust) to evaluate the enclosure's durability and ability to protect internal electronics.	Record any ingress or damage to components. Ensures the enclosure protects the system from environmental elements, crucial for long-term outdoor use.	8 hours, Start on MM/DD
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 (e.g., wildlife researchers) 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

*= for later prototypes or if time allows

Conclusion:

The creation and evaluation of this prototype represent a major achievement in our project, offering valuable insights that will guide our next actions. This phase has enabled us to determine essential enhancements while also involving stakeholders who will play a critical role in upcoming testing and feedback. By refining this initial version, we are establishing a strong base for additional development and bringing us closer to the overall objectives of our project. This prototype has demonstrated its worth by identifying both effective components and aspects that require improvement, and we believe that our strategy will result in a final product that upholds high standards of quality and functionality.

ANNEX

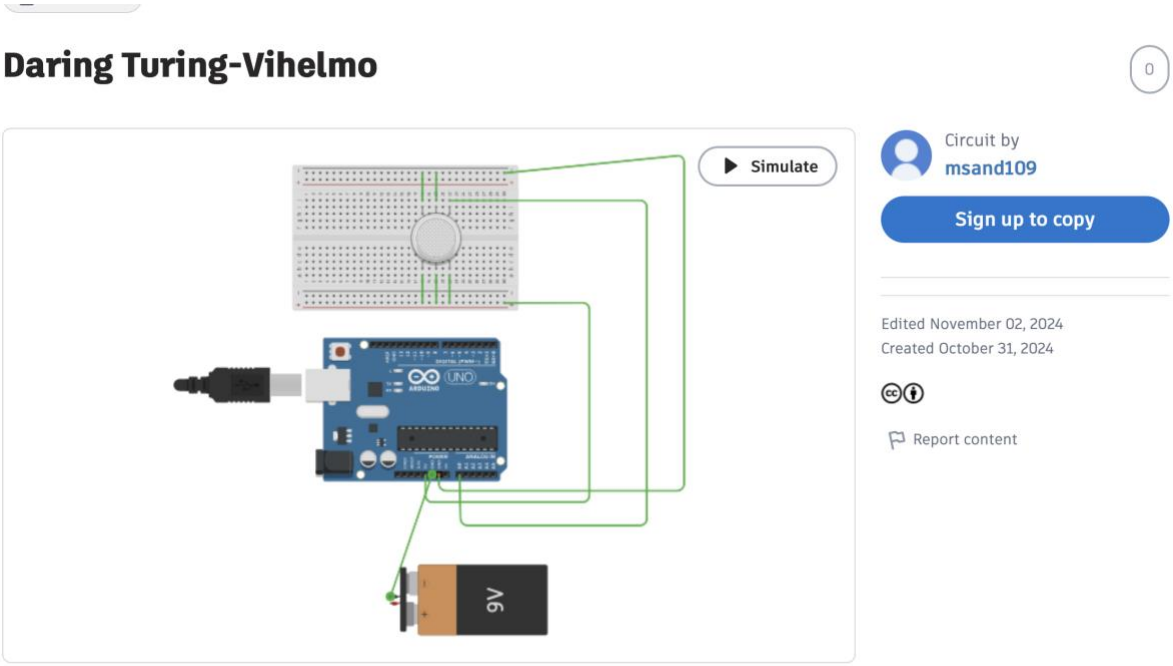
Arduino code

```ino
#include <SPI.h>
#include <SD.h>
#include <Wire.h>
#include <RTClib.h>

const int laserPins[4] = {2, 3, 4, 5}; // Pins for 4 laser sensors
const int chipSelect = 10;              // SD card CS pin

RTC_DS1307 rtc;
File logFile;

int batCount = 0;          // Current bat count inside the box
int totalBatVisits = 0;    // Total number of unique bat entries (persistent count)
int sensorStatus[4] = {HIGH, HIGH, HIGH, HIGH}; // Initial states for sensors (HIGH when unbroken)

// Debounce settings
unsigned long debounceTime = 100;
unsigned long lastDebounceTime[4] = {0, 0, 0, 0};

// State machine for entry/exit detection
int sequenceIndex = 0;
int entrySequence[4] = {1, 2, 3, 4};
int exitSequence[4] = {4, 3, 2, 1};

void setup() {
  Serial.begin(9600);
  while (!Serial) delay(10);
  Serial.println("Bat Entry/Exit Logger Starting...");

  // SD card initialization
  if (!SD.begin(chipSelect)) {
    Serial.println("SD card initialization failed!");
    while (1);
  }
  Serial.println("SD card initialized.");

  // RTC initialization
  if (!rtc.begin()) {
    Serial.println("RTC initialization failed!");
    while (1);
  }
  if (!rtc.isrunning()) {
    Serial.println("RTC is not running, setting to compile time.");
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
```

```
  // Initialize laser sensor pins
  for (int i = 0; i < 4; i++) {
    pinMode(laserPins[i], INPUT);
  }

  // Create and set up CSV file headers
  logFile = SD.open("batlog.csv", FILE_WRITE);
  if (logFile) {
    logFile.println("Date,Time,Event Type,Current Bat Count,Total Bat Visits");
    logFile.close();
  }
}

void loop() {
  for (int i = 0; i < 4; i++) {
    int currentState = digitalRead(laserPins[i]);

    // Detect beam break with debounce
    if (currentState == LOW && sensorStatus[i] == HIGH && (millis() - lastDebounceTime[i]) > debounceTime) {
      sensorStatus[i] = LOW;
      lastDebounceTime[i] = millis();
      checkSequence(i + 1); // Check sequence with sensor index (1-4)
    }
    // Reset sensor status when beam is restored
    else if (currentState == HIGH && sensorStatus[i] == LOW) {
      sensorStatus[i] = HIGH;
    }
  }
}

void checkSequence(int sensorID) {
  // Check for entry sequence
  if (sensorID == entrySequence[sequenceIndex]) {
    sequenceIndex++;
    if (sequenceIndex == 4) { // Full entry sequence detected
      batCount++;
      totalBatVisits++; // Increment total visits counter
      logVisit("Entry");
      sequenceIndex = 0; // Reset sequence index
    }
  }
  // Check for exit sequence
  else if (sensorID == exitSequence[sequenceIndex]) {
    sequenceIndex++;
```

```
      if (sequenceIndex == 4) { // Full exit sequence detected
        batCount--;
        logVisit("Exit");
        sequenceIndex = 0; // Reset sequence index
      }
    }
    else {
      sequenceIndex = 0; // Reset if out of order
    }
  }

  void logVisit(const char* eventType) {
    DateTime now = rtc.now();
    String dataString = "";

    // Format data as CSV: "YYYY-MM-DD,HH:MM:SS,Event Type,Current Bat Count,Total Bat Visits"
    dataString += String(now.year()) + "-";
    dataString += String(now.month()) + "-";
    dataString += String(now.day()) + ",";
    dataString += String(now.hour()) + ":";
    dataString += String(now.minute()) + ":";
    dataString += String(now.second()) + ",";
    dataString += String(eventType) + ",";
    dataString += String(batCount) + ",";
    dataString += String(totalBatVisits);

    Serial.println(dataString);

    // Log data to SD card in CSV format
    logFile = SD.open("batlog.csv", FILE_WRITE);
    if (logFile) {
      logFile.println(dataString);
      logFile.close();
      Serial.println("Data logged.");
    } else {
      Serial.println("Error opening batlog.csv");
    }
  }
```

Circuit



Box Construction and dimensions

KENDRICK LAM
A MOVIE WITHOUT
POPCORN
IS LIKE
DOC
WITHOUT