Faculty of Engineering

University of Ottawa

GNG 1103 C04 – Engineering Design

Fall 2024

Project Deliverable G: Prototype II & Customer Feedback

Course Coordinator: Dr. Muslim Majeed

Group Members: Adam Sirota, Brayden Fitzgerald, Hala Aldahoun, Rami Mosleh, Jordan Draper

Deliverable Date: November 10, 2024

Abstract

This document will display the team's second prototype as well as the analysis and the results gathered during the tests. In addition, it will include a test plan for the other prototypes to do and lists all the feedback the team received during the second prototype.

Table of Contents

1 Introduction

In this technical document, we will start our second prototype that is based on our first prototype and based on feedback and comments we gathered from our clients and users. We will include also the analysis and the results we gathered during the tests for each subsystem. In addition, we will update, from now until our last prototype, our target specifications, detailed design, BOM and prototyping test plan. Thus, with that approach, we will help bat conservationists in Canada to see the effectiveness of the bat boxes.

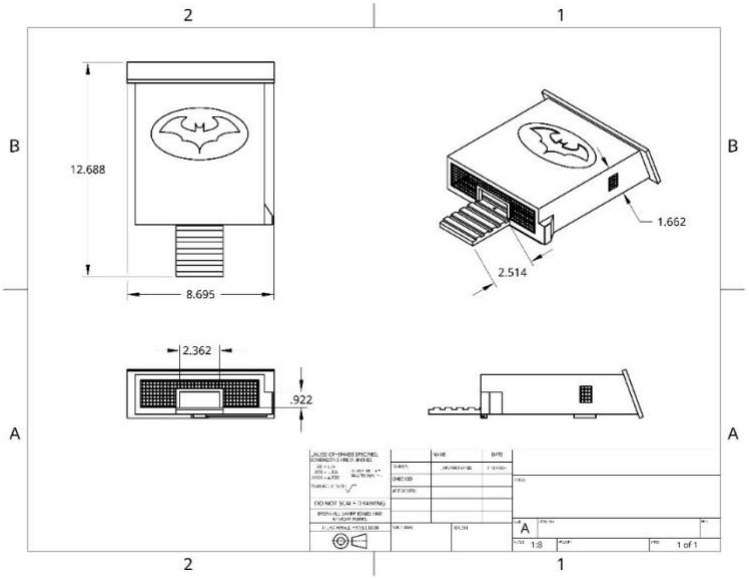
2 Prototype II

In this part, we are going to divide the bat box design into three subsystems, which are the bat box structure, the tracking/storing device, and the power and coding functionality. From these subsystems, our design will track bat visits to that box and will help bat conservationists to see the efficacity of that box.

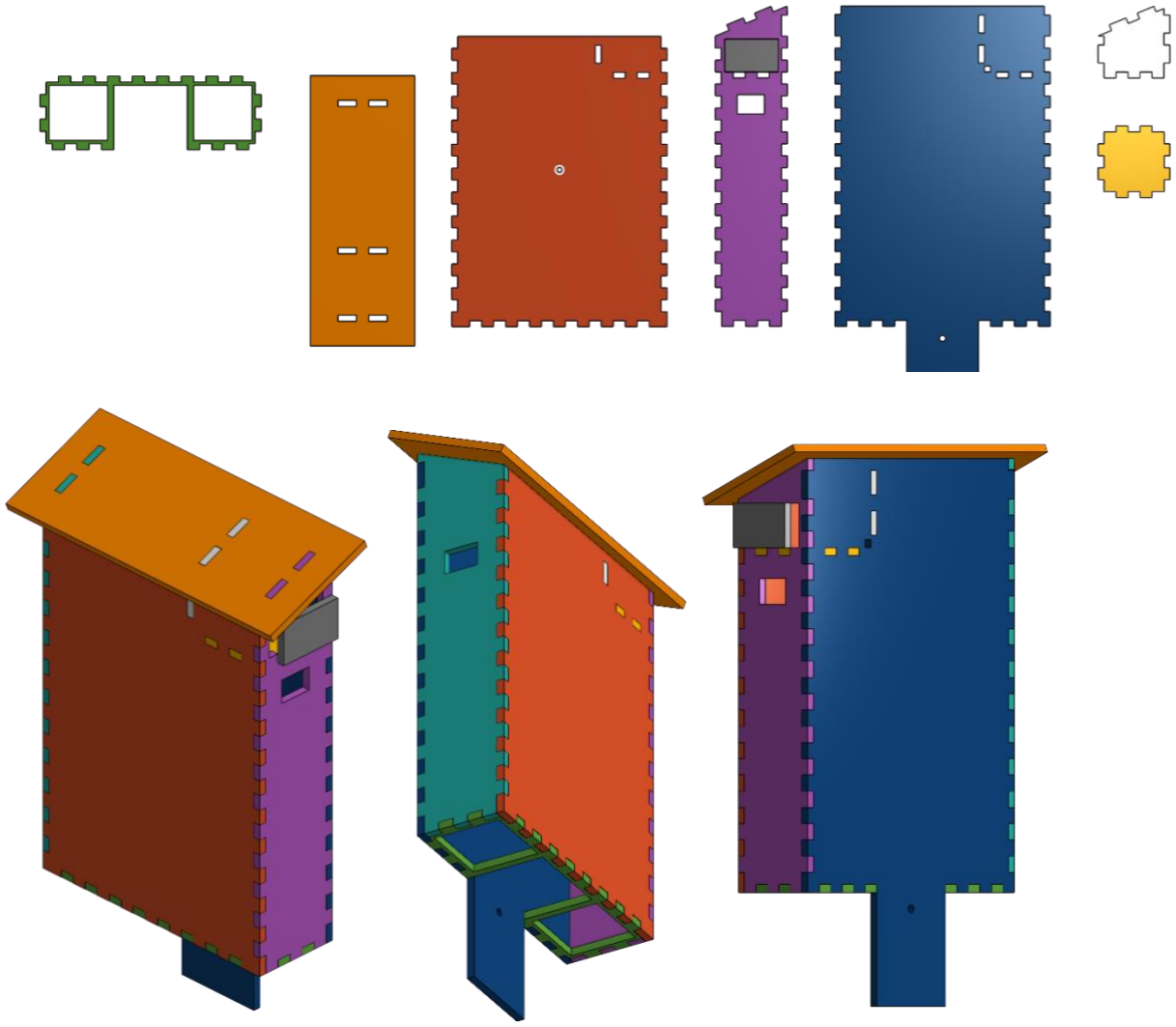
2.1 Bat Box Structure

This subsystem, bat box structure, needs to be designed to give those bats a home that considers their safety, comfort, and health and withstands any change in weather.

2.1.1 Design



2.1.2 Analysis



The bat box will be made from ¼” MDF plywood. The pieces of the box will be laser cut and be assembled using precisely cut finger joints, wood glue, and possibly screws for extra support. The dimensions of the box come to approximately 16.875” x 11” x 3” including the length of the landing pad and the width of the roof. The weight of the box is approximately 2.338 pounds which will make it easy to hang. The thickness of the MDF will ensure that the box can withstand harsh weather conditions like snow build up from a snowstorm or high-pressure winds. The hanging mechanism, wiring channels, and housing for the laser sensor still need to be added to the template and assembled in 3D to ensure all the dimensions are correct.

The laser cutting machine will not be able to etch grooves into the landing pad or interior so it will be done by hand using a wood chisel. The spacing and width of the grooves can be detailed into the MDF for a visual guide of where to etch the grooves using the chisel.

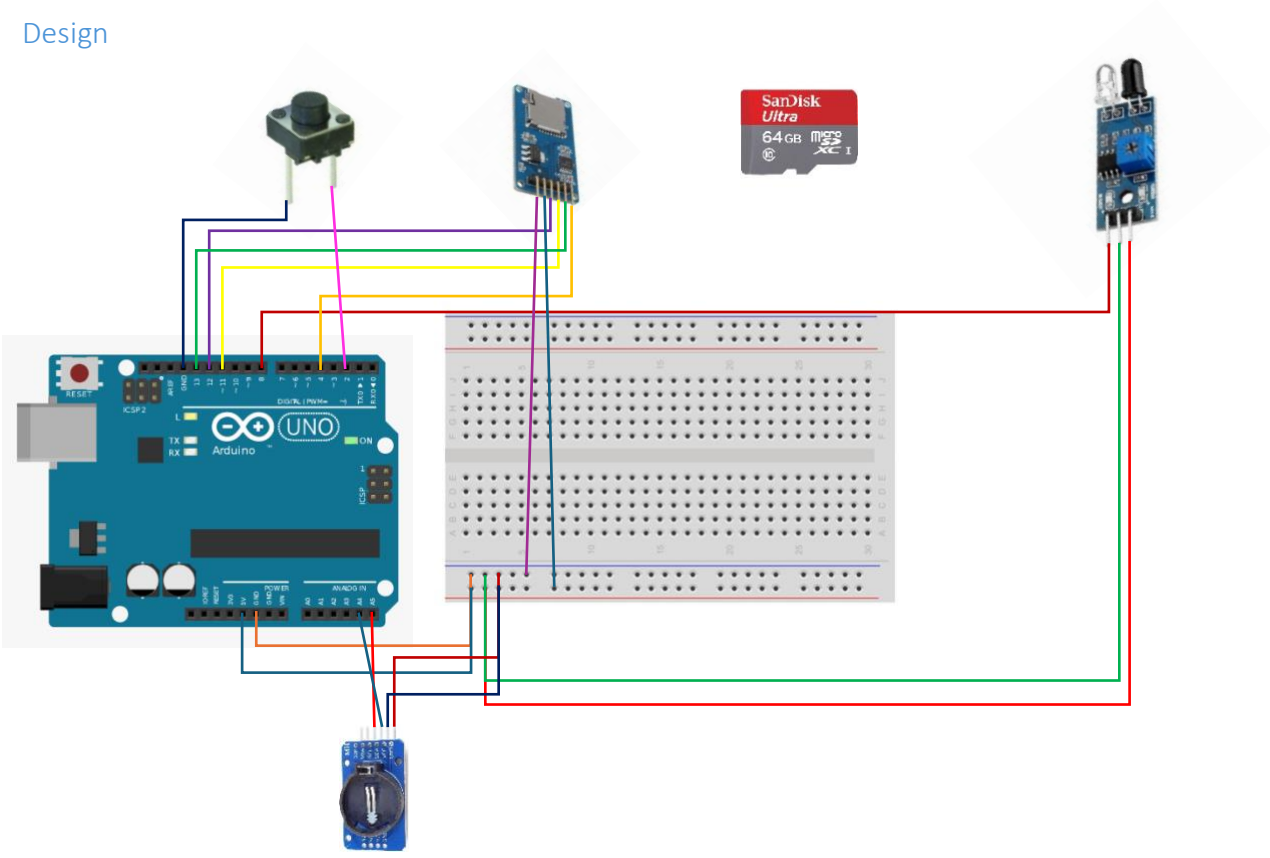
2.1.3 Results

The final features will be added to the laser cutting template and once all the finger joints are properly aligned, the sketches can be converted to SVG files and sent to the laser cutter. The construction of the box will begin once the MDF is acquired so the true strength and durability tests cannot be performed yet.

2.2 Tracking Device

This subsystem, the tracking device, needs to be designed to count the number of bats using this bat box. These data can be stored in a device or transferred to a device that this user always use.

2.2.1 Design



2.2.2 Analysis

For the second prototype of our bat tracking device, we revisited our first prototype and reviewed all the analysis done at that time. From the results we had arrived at, we saw that there is no need for any adjustments in a circuit.
In addition, depending on our prototype test plan, we still need our necessary components to arrive in order to complete our practical testing and refine our system's performance.

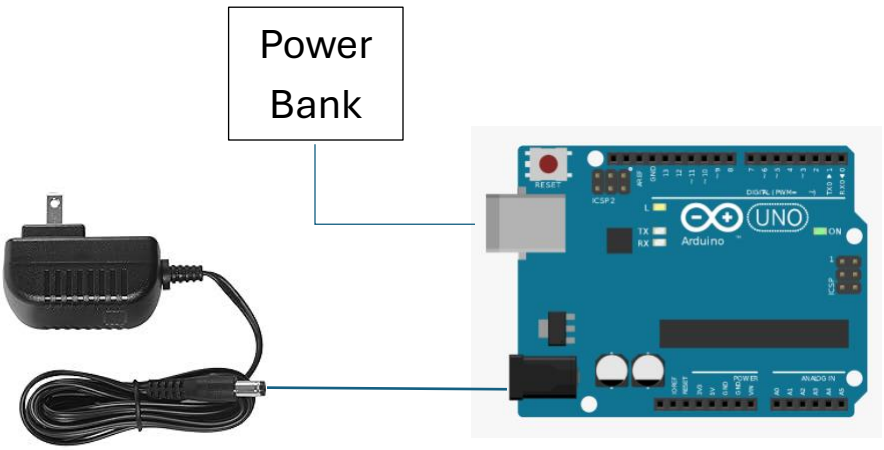
2.2.3 Results

As we move forward, we will improve our tracking system based on the analysis and results done during the first prototype. In addition, once we receive our necessary components, we will begin testing, validating and recording results for our third and last prototype. Thus, as you can see, in our last prototype, we will have a clear image of how these components would interact and how they would work in parallel with the power received from Arduino Uno.

2.3 Power and Coding Functionality

The last subsystem, power and coding functionality, ensures that our tracking device has electricity to function and works autonomously without relying on a computer.

2.3.1 Design



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

const int irSensorPin = 8;   // Pin connected to the IR Sensor OUT
const int chipSelect = 4;     // Pin connected to the SD card's chip select (CS)
const int buttonPin = 2;      // Pin connected to the push button

int batCount = 0;             // Daily bat count
int monthlyBatCount = 0;      // Monthly bat count
int totalBatCount = 0;        // Total cumulative bat count
bool lastSensorState = HIGH;  // Last state of the IR sensor
bool currentSensorState = HIGH;
bool lastButtonState = HIGH;  // Last state of the button

File dataFile;
File totalFile;
File detectionFile;           // File to log detection times
RTC_DS3231 rtc;               // Create an RTC object

void setup() {
  Serial.begin(9600);
  pinMode(irSensorPin, INPUT);        // Set the IR sensor pin as input
  pinMode(buttonPin, INPUT_PULLUP);    // Set button pin with pull-up resistor

  // Initialize the SD card
  if (!SD.begin(chipSelect)) {
    Serial.println("SD card initialization failed!");
    return;
  }
  Serial.println("SD card initialized.");

  // Initialize the RTC
  if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");


    while (1);
  }
  Serial.println("RTC initialized.");
}

void loop() {
  currentSensorState = digitalRead(irSensorPin);

  // Detect when the IR beam is interrupted (bat passes)
  if (lastSensorState == HIGH && currentSensorState == LOW) {
    batCount++;             // Increment daily count
    monthlyBatCount++;      // Increment monthly count
    totalBatCount++;        // Increment total count

    // Get the current time from the RTC
    DateTime now = rtc.now();

    // Log detection time and date
    detectionFile = SD.open("detectionLog.txt", FILE_WRITE);
    if (detectionFile) {
      detectionFile.print("Detected at ");
      detectionFile.print(now.year());
      detectionFile.print("-");
      detectionFile.print(now.month());
      detectionFile.print("-");
      detectionFile.print(now.day());
      detectionFile.print(" ");
      detectionFile.print(now.hour());
      detectionFile.print(":");
      detectionFile.print(now.minute());
      detectionFile.print(":");
      detectionFile.println(now.second());
      detectionFile.close();
    } else {
      Serial.println("Error opening detectionLog.txt");
    }
```

```
    Serial.print("Bat count: ");
    Serial.println(batCount);
    Serial.println("Motion Detected!");
  } else if (currentSensorState == HIGH) {
    Serial.println("Motion Ended!");
  }

  // Get the current time from the RTC
  DateTime now = rtc.now();

  // Check if a day has passed
  static DateTime lastDay = now; // Store last recorded day
  if (now.day() != lastDay.day() || now.month() != lastDay.month() || now.year() != lastDay.year()) {
    // Write daily bat count to SD card
    dataFile = SD.open("batCount.txt", FILE_WRITE);
    if (dataFile) {
      dataFile.print("Bat count for ");
      dataFile.print(now.year());
      dataFile.print("-");
      dataFile.print(now.month());
      dataFile.print("-");
      dataFile.println(now.day());
      dataFile.println(batCount);
      dataFile.close();
    } else {
      Serial.println("Error opening batCount.txt");
    }

    // Write monthly bat count to a separate file
    dataFile = SD.open("monthlyBatCount.txt", FILE_WRITE);
    if (dataFile) {
      dataFile.print("Monthly count for ");
      dataFile.print(now.year());
      dataFile.print("-");
      dataFile.print(now.month());
      dataFile.println(":");
      dataFile.println(monthlyBatCount);
      dataFile.close();
    } else {
      Serial.println("Error opening monthlyBatCount.txt");
    }

    // Write total bat count to a separate file
    totalFile = SD.open("totalBatCount.txt", FILE_WRITE);
    if (totalFile) {
      totalFile.print("Total bats recorded until now: ");
      totalFile.println(totalBatCount);
      totalFile.close();
    } else {
      Serial.println("Error opening totalBatCount.txt");
    }

    // Reset daily, monthly, and total counts for the next day
    batCount = 0;
    monthlyBatCount = 0; // Reset monthly count
    totalBatCount = 0;   // Reset total count if desired
    lastDay = now; // Update last recorded day
  }

  // Button press logic
  int buttonState = digitalRead(buttonPin);
  if (lastButtonState == HIGH && buttonState == LOW) {
    // Button pressed
    int currentTotalBats = totalBatCount; // Capture the total bat count

    // Save the total bat count to a separate file
    totalFile = SD.open("totalBatCount.txt", FILE_WRITE);
    if (totalFile) {
      totalFile.print("Total bats: ");
      totalFile.println(currentTotalBats); // Store the total bat count
      totalFile.close();
    } else {
      Serial.println("Error opening totalBatCount.txt");
    }

    Serial.print("Total bats: ");
    Serial.println(currentTotalBats);
  }

  lastButtonState = buttonState; // Update the last button state
  lastSensorState = currentSensorState;

  // Small delay to avoid excessive reads
  delay(50);
}
```

2.3.2 Analysis

While creating our second prototype, we found that we needed a power backup in case the power from the wall outlet cuts off. So, we did some research until we found out valuable information. The Arduino Uno turns on the USB power if the power at the Barrel Jack drops below 7.5 V. So, we decided to use a power bank to aliment our Arduino Uno in case power from the wall outlet cuts off. As a summary, in a normal situation, Arduino Uno will accept power only from barrel jack even though the power bank is connected to it. The power bank will be used only in emergency cases.

Also, the analysis of our power system focused on ensuring that it could reliably support all integrated components and functionalities. A critical aspect was confirming that the power delivery to the Arduino Uno was stable and sufficient for running the system without interruptions. Installing necessary libraries like RTClib was part of verifying that the setup was ready for seamless operation, as it ensured proper time management for data logging, which is indirectly supported by the power system. This step was crucial because any power failure or instability could disrupt the communication and functionality of connected modules, impacting the overall performance of the prototype.

2.3.3 Results

During testing, we uploaded the code onto the Arduino Uno using the Arduino IDE and selected COM 9 for communication. The prototype responded as expected, indicating that the power system successfully provided stable energy for the device's operations. This confirmed that the circuit connections and the programming logic were functioning as intended, demonstrating that the system was ready for subsequent development and integration phases.

3 Feedback & Comments

The client provided valuable feedback regarding the bat box design. They suggested including a backup battery for the tracking devices to ensure uninterrupted data collection in case of power failure. Additionally, the client recommended positioning the laser sensor near the entrance of the bat box, but specifically below it, and not exactly on the entrance, to improve the accuracy of bat detection as they enter. These adjustments will help optimize the sensor's functionality and enhance the overall reliability of the tracking system.

4 Updated Design Framework

4.1 Target Specifications

Table 1: Target Specifications Table

	Design Specifications	Relation (=, < or >)	Value	Units	Verification Method
	Functional Requirements				
1	Track the number of bat visits	=	yes	N/A	Test
2	Attractive Bat Box Shape	=	yes	N/A	Test
3	Predator-Proof	=	yes	N/A	Test
4	Number of bats supported (Bats)	>	15	Bats	Analysis
5	Power Source for Sensor	=	yes	N/A	Test
6	Data Stored Device	=	yes	N/A	Test
	Non-Functional Requirements				
7	Aesthetics	=	yes	N/A	Test
8	Product life (years)	<	8	years	Analysis
9	Lightweight for installation	=	yes	N/A	Test
10	Low-Maintenance	=	yes	N/A	Test
11	Removable parts for cleaning	=	yes	N/A	Test
12	Weatherproof	=	yes	N/A	Test
	Constraints				
13	Bat Box Size (m^3)	<	0.12	m^3	Analysis
14	Eco-Friendly Materials	=	yes	N/A	Test
15	Operating Conditions: Temperature (°C)	=	10 to 35	°C	Test
16	Operating Condition: Humidity (%)	=	40 to 60	%	Test
17	Entry/Exit Opening (m)	<	0.0254	m	Analysis
18	Cost ($)	<	150	$	Estimate, Final check

4.2 Bill of Materials

Table 2: Bill of Materials

Parts	Quantity	Unit of Measure	Unit Cost ($)	Estimate Cost ($)	Link
Hardware Required					
1/4" MDF (18in × 24in)	2	inches	4	8	Link Sections
8 oz. Black Waterproofing Stain	1	ounce	5.98	5.98	Link Sections
11 mm Aperture Wire Mesh (15 cm x 15 cm)	2	centimeters	5.28	10.56	Link Sections
Wood Glue 8 oz.	1	ounce	0	0	Owned
Wood Screws #6 1-5/8"	10	inches	0	0	Owned
Folding Butt Hinges	2	inches	0	0	Owned
Duck-Billed Hasp Lock	2	inches	0	0	Owned
Electrical Components Required					
Arduino Uno	1	EA	15.25	15.25	Link Sections
Breadboard	1	EA	5	5	Link Sections
IR Sensor Emitter/Receiver	1	EA	8.99	8.99	Link Sections
MicroSD card Adapter	1	EA	8.9	8.9	Link Sections
MicroSD card	1	EA	9.99	9.99	Link Sections
DS3231 RTC Module	1	EA	2.99	2.99	Link Sections
Power Supply Adapter 12V	1	EA	13.99	13.99	Link Sections
Power Bank	1	EA	0	0	Owned
Push Button	1	EA	2.25	2.25	Link Sections
4 Jumper Wires (Male-Male)	1	EA	1	1	Link Sections
16 Jumper Wires (Male-Female)	2	EA	1	2	Link Sections
Software Required					
OnShape	1	EA	0	0	Owned
Trello	1	EA	0	0	Owned
Arduino IDE	1	EA	0	0	Owned
Libraries					
SPI	1	EA	0	0	Link Sections
SD	1	EA	0	0	Link Sections
Wire	1	EA	0	0	Link Sections
RTClib	1	EA	0	0	Link Sections
Total Cost (Without Taxes)				94.90	
Total Cost (With Taxes)				107.24	

5 Typical Objectives

The objectives for this bat box project include addressing client feedback, ensuring subsystem functionality, verifying design feasibility, and reducing maintenance requirements. Based on the client's input, the entry/exit mechanism must always remain open, ensuring that bats can move freely. This change minimizes maintenance requirements and reduces the risk of obstructing bat movement or impacting tracking accuracy. Another objective is the inclusion of a mesh-emptying solution, proposed as a drawer system that allows easy access for cleaning without disturbing the bats. This reduces the risk of the mesh filling up, which could hinder bat movement or interfere with data collection.

The design's flexibility and feasibility are also prioritized, particularly in the mountable roof, which, while not a primary concern for the client, may require testing in various mounting environments, such as on trees or other structures, to ensure reliability. Testing for ease of access in the drawer system is also planned to ensure it meets the project's minimal maintenance goals. These objectives work together to create a durable, low-maintenance bat box that meets client specifications and environmental needs.

6 Prototype Test Plan

Table 3: Prototype Test Plan

ID	Test Objective	Description of Prototype	Results to be Recorded	Duration of Test
1	Determine if bat box is effective	Test and observe the temperature within the bat box to ensure it is suitable for bats, and also test how the bat box may be affected by weather such as snow. Snow will be put on the roof of the bat box to test how it handles that type of weather.	-The temperature within the bat box to ensure it is safe for the bats inside -Check to see if the bat box can support the weight of snow buildup (Surface area of the roof times height typical snowfall times density of snow)	45 minutes
2	Determine if the coding is functional	Have the Arduino ide code on a laptop connected to what is needed to track visits, and simulate an entry and exit	If the code gives the proper and accurate number of entry/exits that were simulated.	10 minutes
3	Determine if laser sensor is working	Simulate entry and exiting from the bat box using a prop for a given number of times.	The Arduino will record the number of times the laser beam's sensor detects an object at the bat box's entry. Verify that the sensor has accurately recorded the number of entries.	5 minutes
4	Determine if SD card stores data	Simulate visits and plug card into computer and check stored data, also simulate a power outage to see how SD card reacts, also test to ensure that stored data stays the same after you remove the SD card and re insert it into the bat box	Record if SD card stores data correctly, that it can properly operate before and after a power outage, and if SD card keeps data even after it gets removed from the box and re inserted.	45 minutes
5	Determine if battery aliments Arduino Uno	Simulate visits to bat box, and observe if the code is updating the simulated visits properly, if so than that shows the battery is powering the Arduino uno	If the coding setup is running properly, and the Arduino uno is providing the correct input for the code, we can assume the battery is properly powering the Arduino Uno.	15 minutes
6	Determine if the reset button is working	Simulate visits to the bat box, and then press reset button	If the number of visits goes from its original number back to 0 after the reset button is pressed	5 minutes

7 Stopping Criteria

Table 4: Stopping Criteria Table

Test	Stopping Criterion
1	The test can be stopped once the snow (10-15 cm to simulate a typical heavy snowfall) has been put on the roof and the temperature, as well as any structural deformations, have been recorded
2	The test can be stopped once a bat entry and exit has been simulated and the code has been observed to be either functioning properly or not
3	The test can be stopped once the given number of entries and exits has been simulated and the laser sensor has been observed to either accurately recorded the number or entries and exits or not
4	The first component of the test can be stopped once the given number of entries and exits have been recorded and the SD card has been observed to be either storing data correctly or not. The second component can be stopped once a power outage has been simulated and the SD card has been tested for functionality. The third component can be stopped once the SD card has been removed from the bat box, reinserted, then tested for functionality
5	The test can be stopped once an entry and exit has been simulated and the code/battery have been observed to either be powered and functioning properly or not
6	The test can be stopped once the reset button has been pressed and the results have been observed.

8 Conclusion

The goal of this deliverable was to improve our second prototype based on our team's first prototype bat box and tracking system, as well as to outline a test plan with stopping criteria for our third prototype. Through analysis, peer review, and feedback/comments from clients, we were able to review and update our target specifications, bill of materials, prototyping test plan, and detailed design. With our test plan and stopping criteria well defined, and our overall designs and plans improved through client and peer feedback, we are ready to build a more effective and efficient bat box enclosure.

9 Link Sections

Hardware:

Bat Box Structure CAD:

https://cad.onshape.com/documents/cf85593d9314c67186a69158/w/261c4abd28dc68f89aa972f0/e/e67d684057078f74b29aa4f6

1/4" MDF Sheet (18in × 24in):

https://makerstore.ca/shop/ols/products/mdf/v/M003-1-4-18-NCH

8 oz. Black Waterproofing Stain:

https://www.homedepot.com/p/BEHR-PREMIUM-8-oz-SC-102-Slate-Solid-Color-Waterproofing-Exterior-Wood-Stain-and-Sealer-Sample-501316/203728594

11mm Aperture Wire Mesh:

Link

Electrical Components:

Arduino Uno:

https://makerstore.ca/shop/ols/products/arduino-uno-r3-clone/v/MC001-A

Breadboard:

https://makerstore.ca/shop/ols/products/breadboard/v/C005-HLF

IR Sensor Emitter/Receiver:

DAOKI 5-Pack IR Infrared Obstacle Avoidance Sensor Module for Arduino Smart Car Robot 3 Wire : Amazon.ca: Tools & Home Improvement

MicroSD Card Adapter:

CANADUINO® 3 x Micro-SD Memory Card Adapter for Arduino with 3.3V-5V Converter : Amazon.ca: Electronics

MicroSD Card:

VERBATIM 16GB Premium microSDHC Memory Card with Adapter, UHS-I V10 U1 Class 10, Black (44082) : Amazon.ca: Electronics

DS3231 RTC Module:

CANADUINO® DS3231 RTC Module, 32kB Memory, I2C Interface, Battery Backup : Amazon.ca: Electronics

Power Supply Adapter:

https://www.amazon.ca/dp/B082VVSLGR?ref=cm_sw_r_cso_wa_apan_dp_AC713SVMR8KPGXTE0ZN0&ref_=cm_sw_r_cso_wa_apan_dp_AC713SVMR8KPGXTE0ZN0&social_share=cm_sw_r_cso_wa_apan_dp_AC713SVMR8KPGXTE0ZN0&starsLeft=1&skipTwisterOG=1&th=1

Push Button Switch:

https://makerstore.ca/shop/ols/products/on-off-power-button-pushbutton-toggle-switch

4 Jumper Wires (Male-Male):

https://makerstore.ca/shop/ols/products/jumper-cables-pack-of-10/v/C004-20-MM

16 Jumper Wires (Male-Female):

https://makerstore.ca/shop/ols/products/jumper-cables-pack-of-10/v/C004-20-MF

Libraries:

SPI: https://docs.arduino.cc/learn/communication/spi/

SD: https://docs.arduino.cc/libraries/sd/

Wire: https://docs.arduino.cc/language-reference/en/functions/communication/Wire/

RTClib: https://reference.arduino.cc/reference/en/libraries/rtclib/

10 References

1. IR Sensor Emitter/Receiver Image: https://hackster.imgix.net/uploads/attachments/1307911/_9cw77Dc0bn.blob?auto=compress%2Cformat&w=900&h=675&fit=min
2. Breadboard Image: https://www.shutterstock.com/image-vector/half-breadboard-vector-illustration-providing-600nw-2306585395.jpg
3. Arduino Uno Image: https://www.pngitem.com/pimgs/m/14-146612_arduino-uno-vector-png-transparent-png.png
4. MicroSD Card Adapter Image: https://store.nerokas.co.ke/image/cache/catalog/MicroSD%20module-500x500.JPG
5. MicroSD Card Image: https://www.bhphotovideo.com/images/images2000x2000/sandisk_sdsqunc_064g_an6ia_sandisk_1170203.jpg
6. DS3231 RTC Module Image: https://m.media-amazon.com/images/I/61jIFJRna4L._AC_SL1333_.jpg
7. Power Supply Adapter Image: https://m.media-amazon.com/images/S/aplus-media-library-service-media/e7062853-0fb1-43a7-a0b8-0aa87483312c.__CR0,0,1600,1600_PT0_SX300_V1___.jpg
8. Push Button Image: https://th.bing.com/th/id/OIP.LL_TiS5naraLfQ7KDedDngHaF9?pid=ImgDet&w=171&h=137&c=7&dpr=2.2