Faculty of Engineering

University of Ottawa

GNG 1103 C04 – Engineering Design

Fall 2024

Project Deliverable F: Prototype I & Customer Feedback

Course Coordinator: Dr. Muslim Majeed

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

Deliverable Date: November 3, 2024

Abstract

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

Table of Contents

1 Introduction 5

2 Prototype I 6

2.1 Bat Box Structure 6

2.1.1 Design 6

2.1.2 Analysis 7

2.1.3 Results 7

2.2 Tracking Device 8

2.2.1 Design 8

2.2.2 Analysis 8

2.2.3 Results 10

2.3 Power and Coding Functionality 10

2.3.1 Design 11

2.3.2 Analysis 13

2.3.3 Results 13

3 Feedback & Comments 14

4 Updated Design Framework 15

4.1 Target Specifications 15

4.2 Bill of Materials 16

5 Typical Objectives 17

6 Prototype Test Plan 18

7 Stopping Criteria 19

8 Conclusion 19

9 Link Sections 20

10 References 22

1 Introduction

In this technical document, we will build our first prototype that is based on our concepts designed over the past two months and we will include the analysis and the results for each subsystem. In addition, we will gather feedback and comments from clients and users and 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.

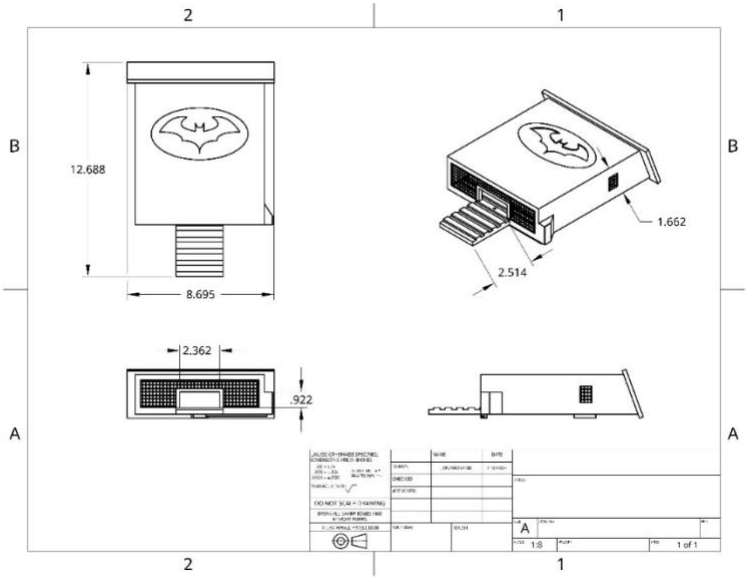
2 Prototype I

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 design of the bat box has been updated to reflect the needs of the client better and incorporate a housing system for the tracking device and laser sensor. The opening has been resized to properly accommodate the size of the bats while still being small enough to deter predation. The landing pad's size has been reduced to line up with the entrance. The wire meshing has been changed to a square pattern with 11 mm sized meshing to allow guano to fall through and to parallel the same wire mesh that will be used for the final design. A small box has been attached to the side of the bat box to house the tracking system with a wire channel running along the side to the back to connect to the laser. A hole has been added peering into the entrance to allow the laser to detect and track bats. The laser will be adjusted to cover the entirety of the entrance. The 3D model has been given material and appearance properties to get an idea of what the final design will look like. The bat box has been colored black to reflect the black waterproof stain that will be used, and its materials have been set to MDF and metal for the plywood and wire mesh respectively which will help estimate the weight of the final product. A Batman logo has been added for some flare and a little more personality.

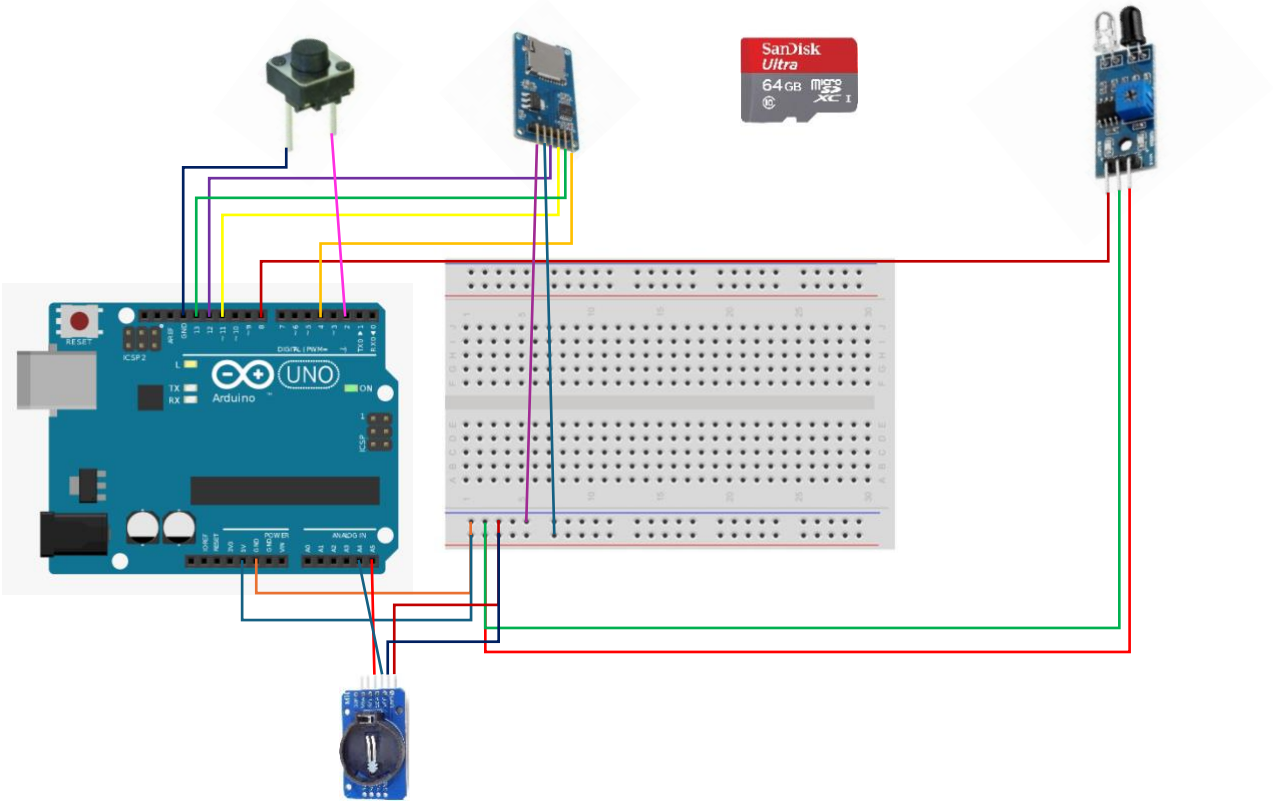
2.1.3 Results

The box now satisfies all the client's requirements and has all the proper elements added to house a tracking and storing system for the bats. The next step for the project will be to create a laser cutting template for the box and add finger joints to all the sides, ensuring that they properly align. Once the box has been assembled, more tests can be performed to verify whether all the components will fit and that it is sturdy enough.

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

- In this system, the laser sensor is going to be placed inside the notch where the bat first lands before he enters the bat box. This sensor will detect and send a signal to Arduino Uno board when an obstacle, like a bat, is in front of the infrared laser. Thus, this signal will enable the microSD adapter to store the data which is the detected bat and will enable also DS3231 RTC module which will precise at which time and at which day this bat is detected. Thus, all in all, it works as a tracking and storing device.

  With the potentiometer (the screw/knob) of the IR sensor Emitter/Receiver, we can change the distance from 2 cm to 30 cm, so that it helps us detect an object, like a bat. In addition, this laser has a detection angle of 35°. However, due to sunlight activity and to bat guano, we are going to choose a distance of 2cm which will detect the bat

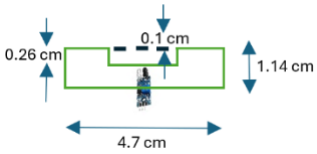
presence when he lands on the notch. Furthermore, the bat's body will cover the laser and create a dark environment that favors the laser sensor.

For analysis, we reduced each dimension from the front plane and the right plane by a factor of 5. So, the detection distance needs to be divided by 5 to have a distance of 0.4 cm. In addition, for our future prototypes, a green LED will help us identify if the coding is functioning well by turning on light when detecting an obstacle.

Image 1: An Illustration of Tracking Device on the Front Plane of the Bat Box

Image 2: An Illustration of the Distance Used for Tracking Bats on the Right Plane of the Bat Box

As you can see in the image besides, only 0.1 cm will be used to detect bats. In addition, in the analysis part above, we said that the detection distance is 0.4 cm. Thus, the other 0.3 cm will be the position of the laser from the top surface (TS) shown in this image.



We also said that it has a detection angle of 35°, but this angle won't be blocked with the edge of the notch (EN). It will still be inside the range.

- In addition, the data won't be available right away. It will be stored on microSD card that will be left there for like 5 months.

  To have a big picture of what's going to happen at the level of MicroSD card, we will explain using a list of the actions that will be performed:

1st Action: Four files are opened at the same time at MicroSD card

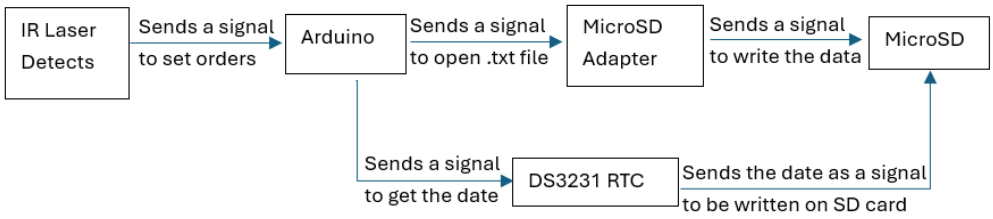
2nd Action: First file has the accurate time and date that the bat was detected without saying the number of bats. The second file has the number of bats for each day. The third file has the number of bats for each month. Lastly, the last file is just designated to

3rd Action: After the 5 months have passed and once deinstalling the bat box, there is a push button beside it. This push button, connected to Arduino, when pressed, will send the total number of bats for the last 5 months to the last file.

Thus, this total number of bats would have an equal number of bats entering and exiting the bat box. In addition, this number would be an even number, so that once you divide it by 2, you would have the real number of bats that used this bat box.

- For the interaction of the three components (MicroSD Adapter, DS3231 RTC and IR Sensor), the image below will explain the existence of connections between each component.

Image 3: An Illustration of the interaction between the 3 different components.



2.2.3 Results

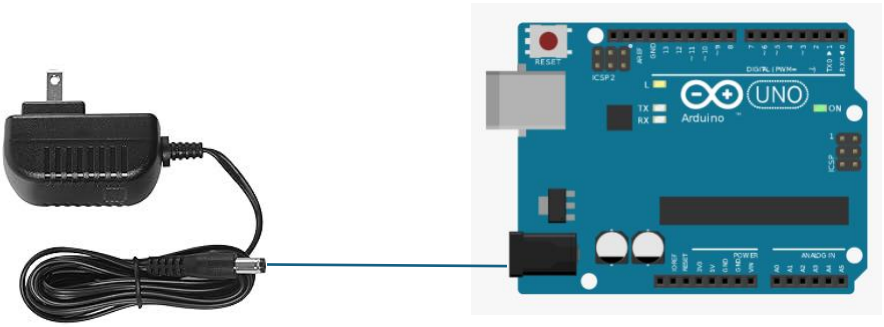
Since the slit entrance of the bats is going to be 2 cm at most, due to protection from predation, then they are going to squeeze to enter. This will enable the laser and send a signal to Arduino saying that it detects an obstacle which is in this case, the bat.

In addition, DS3231 RTC, MicroSD Adapter and IR Laser that have different operations work together as a system that receives data from the outside environment and stores it down a .txt file that comes in accordance with a real time and date.

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




```
#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");
    }
```

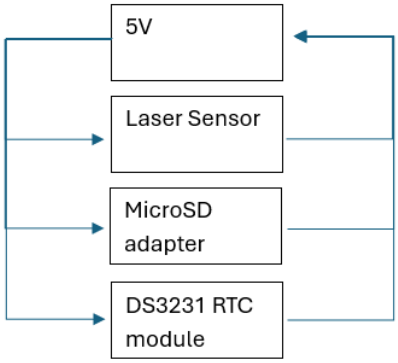
```
    Serial.print("Bat count: ");
    Serial.println(batCount);
    Serial.println("Motion Detected!");
  } else if (currentSensorState == HIGH) {
    Serial.println("Motion Ended!");
  }

  // 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

The power that comes from the wall outlet will be converted to 12V through AC/DC Power Supply and will enter the external/socket power of Arduino Uno board.

While, for the 3 components (Laser sensor, MicroSD adapter, DS3231 RTC module), they will be connected in parallel with 5V pin of the Arduino Board.

Image 4: An Illustration of the Circuit that will be Implemented on the Arduino Board



In this diagram, each of the components that needs 5V to operate, have 5V.

Thus, it receives the 12 V from the AC/DC Power Supply and converts it to 5V in the 5V pin, so that each of the 3 components work properly and functionally.

2.3.3 Results

We have already done some research on how much voltage the external/socket power or DC Jack stands and from the Arduino Docs[1], we found that it can accept a voltage around 9V-12V DC. In addition, we found, in the description of each of the 3 components, that they can be used for 3V-5V power supply. Thus, the output of the AC/DC Power Supply will generate 12V DC as an output which is an input to the external power of the Arduino Uno and the output of the 5V pin of the Arduino board will generate 5V DC which is the necessary power needed for all the 3 components to function.

[1] Arduino Uno Rev3 with Long Pins: https://docs.arduino.cc/retired/boards/arduino-uno-rev3-with-long-pins/

3 Feedback & Comments

During our Thursday lab, we gathered feedback from TAs and neighboring groups to refine our project and ensure it aligned well with practical needs. One key point of feedback was about the laser sensor's detection range. Initially, we discovered it was limited to 45 degrees, but after researching solutions, we determined that adjusting the angle would broaden the sensor's range, resolving this limitation and ensuring more accurate tracking of bat entries.

Other groups provided helpful comments on our mountable roof design, expressing appreciation for the idea and noting that it seemed both functional and efficient for tree mounting. Some questions arose regarding our design's handling of bat guano collection. We had initially included small openings at the bottom of the box, (like airflow), but realized they could potentially serve as drainage for the guano as well. This feedback prompted us to enhance the opening so that airflow wouldn't be obstructed, and the mesh would remain clear of blockages.

Additionally, we received confirmation that our design met the cost and maintenance requirements outlined in our problem statement. No concerns were raised in these areas, indicating that our approach to affordability and ease of maintenance is on track and well-aligned with our project goals.

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 Sheet (18-inch X 24-inch)	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
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	1.8	1.8	Link Sections
MicroSD card Adapter	1	EA	2.97	2.97	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	10	10	Link Sections
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)				77.79	
Total Cost (With Taxes)				87.90	

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 develop our team's first prototype bat box and tracking system, as well as to outline a test plan with stopping criteria for our second 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

Link Sections:

Hardware:

Bat Box Structure CAD:

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

1/4" MDF Sheet:

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://makerstore.ca/shop/ols/products/5v-power-supply/v/P013-12V-US

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