User and Product Manual Instructions

This document is a template of a user and product manual. The client may wish to make improvements on the prototype or need to fix it if something goes wrong or another group of students may work to make a more rugged prototype. The document needs to be clear for someone else who is not an engineer **to use, maintain or reproduce the project**. Include as many images and diagrams as possible for a better understanding. Keep it plain, simple, visual and logical.

In general, if you are not sure exactly what to include, imagine that this document was the only thing that you had. Imagine also that your job was to add a new feature or recreate the project that is described in your document. What would you need to know?

Only include details relating to your final prototype.

Template conventions:

- Remove all red text, it is only there to guide you
- Remove this page (instructions)
- Replace all instances of <xxx> with the appropriate information for your group, for example you could replace <System Name (Acronym)> by The Amazing Product (TAP)
- Save this document as 'User and Product Manual_group number' instead of Deliverable X so that others know what it represents when they see it in MakeRepo

GNG<1103/2101>

Design Project User and Product Manual

BAT MONITORING SYSTEM

Submitted by:

17 BATS, GROUP 17

Mataab Bashir-Mustafa, 300060426

Khadijetou Coulibaly, 300327381

Sean A. Fitzgerald, 300355584

Callum Mcdougall, STUDENT NUMBER

Miguel Sandri, 300351497

Ana Sinna, 300319127

3rd of December 2024

University of Ottawa

Table of Contents

Table of Contents ii

List of Figures iv

List of Tables vi

List of Acronyms and Glossary vii

1 Introduction 1

2 Overview 3

2.1 Conventions 3

2.2 Cautions & Warnings 4

3 Getting started ..... **Error! Bookmark not defined.**

3.1 Set-up Considerations ..... **Error! Bookmark not defined.**

3.2 User Access Considerations ..... **Error! Bookmark not defined.**

3.3 Accessing the System ..... **Error! Bookmark not defined.**

3.4 System Organization & Navigation ..... **Error! Bookmark not defined.**

3.5 Exiting the System ..... **Error! Bookmark not defined.**

4 Using the System ..... **Error! Bookmark not defined.**

4.1 <Given Function/Feature> ..... **Error! Bookmark not defined.**

4.1.1 <Given Sub-Function/Sub-Feature> ..... **Error! Bookmark not defined.**

5 Troubleshooting & Support 11

5.1 Error Messages or Behaviors 11

5.2 Special Considerations 12

5.3 Maintenance 12

5.4 Support 13
6 Product Documentation 14
6.1 <Subsystem 1 of prototype> .....**Error! Bookmark not defined.**
6.1.1 BOM (Bill of Materials) 15
6.1.2 Equipment list 16
6.1.3 Instructions 17
6.2 Testing & Validation 21
7 Conclusions and Recommendations for Future Work 27
8 Bibliography 28
APPENDICES 29
APPENDIX I: Design Files 29
APPENDIX II: Other Appendices 32

List of Figures

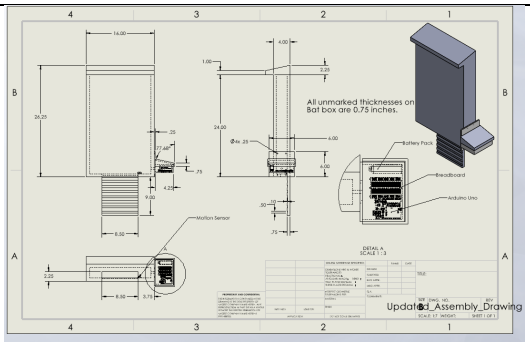
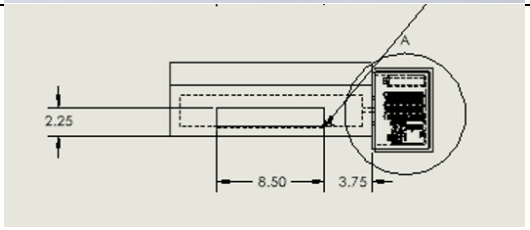
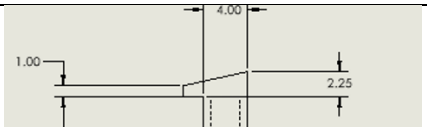
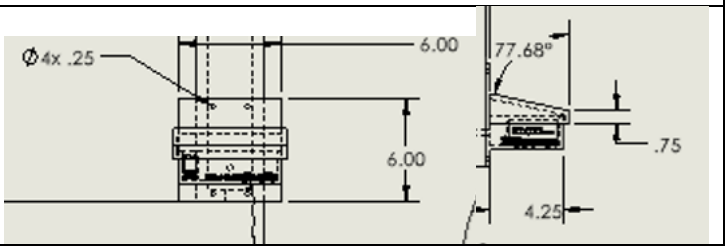
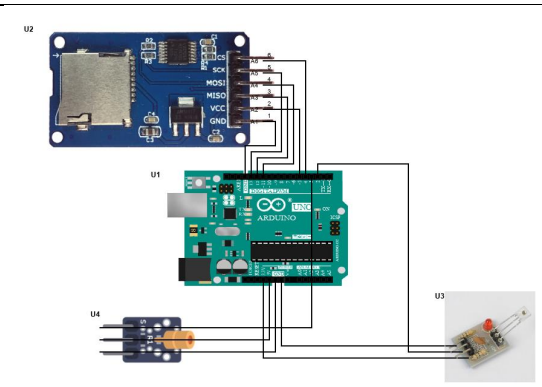
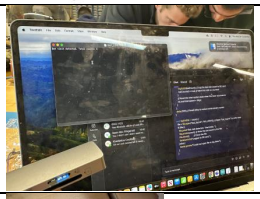
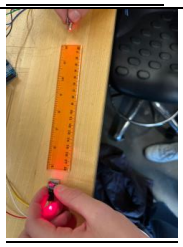
Figure Name	Figure Image
Figure 1: Battery Clip	
Figure 2: Overview of bat box assembly	
Figure 3: Bottom view of bat box.	
Figure 4: Top-side view of bat box.	
Figure 5: Electronic box dimensions	

Figure 6: Arduino Wiring Setup	
Figure 7: Laser orientation	
Figure 8: Confirm Arduino data logging and evaluate data retrieval and storage efficiency	
Figure 9: Assess battery life under prototype load	
Figure 10: Test for interference from environmental factors	

List of Tables

Table 1. Acronyms vii

Table 2. Glossary vii

Table 3. Referenced Documents 31

List of Acronyms and Glossary

Table 1. Acronyms

Acronym	Definition
BMS	Bat Monitoring System
UPM	User and Product Manual
SD	Secure Digital
VCC	Voltage Common Collector
GND	Ground

Table 2. Glossary

Term	Definition
Arduino	An open-source electronics platform to program microcontrollers.
Arduino IDE	Integrated Development Environment for writing and uploading code on Arduino microcontrollers.
KY-008	A laser transmitter module for beam detection.
DAOKI	A laser receiver module.
FAT32	A file system format for SD cards.
VCC	The power supply voltage for electrical components.
GND	The reference pin point in the circuit for voltages.
Dupont Cables	Jumper wires used to connect components.
Guano	Bat droppings.

1 Introduction

The **Bat Monitoring System** (BMS) **User and Product Manual** (UPM) contains all essential information for users to effectively operate and maintain the system. This manual includes a description of system functions and capabilities, step-by-step procedures for setup and use, alternate modes of operation, and guidance for troubleshooting and maintenance. Graphics and figures are included wherever possible to facilitate understanding and usability.

Purpose and Scope

The primary purpose of this UPM is to support users in the effective utilization of the BMS, ensuring accurate monitoring of bat activity while promoting bat conservation. It provides detailed information on system setup, functionality, and maintenance, including the associated prototype documentation.

The scope of the UPM covers the following areas:

- Description of system features, including laser sensors, Arduino integration, and data logging capabilities.
- Procedures for system setup, user access, and navigation.
- Troubleshooting guidance and recommendations for maintaining system integrity.
- Documentation of prototype design, testing, and validation.

This manual is intended for wildlife researchers, ecologists, and technical staff involved in monitoring bat populations and maintaining the system.

Organization

The UPM is organized into the following sections:

1. Introduction: Overview of the purpose, scope, organization, and relevant references.
2. Overview: General conventions, safety precautions, and warnings.
3. Getting Started: Guidelines for system setup, access configuration, and navigation.
4. Using the System: Detailed operational instructions for key functions.
5. Troubleshooting and Support: Solutions for common issues, maintenance tips, and contact information.
6. Product Documentation: Technical details, bill of materials, assembly instructions, and testing validation.
7. Conclusions and Recommendations: Insights and suggestions for future work.
8. Bibliography: References used in the project.
9. Appendices: Supplementary resources, including design files and additional data.

Points of Contact

For support and assistance with the BMS, users may contact:

- **Primary Contact**: Tiree FS Inc.
  - https://tiree.ca/contact
- **Alternate Contact**: Project Development Team Lead
  - mspace@uottawa.ca

Primary Functions

The BMS is designed to support the critical ecological task of monitoring bat populations, enabling researchers to:

1. Track and log bat activity in a non-intrusive manner.
2. Analyze trends in bat movement and habitat usage.
3. Facilitate long-term conservation efforts by providing accurate, reliable data.

The system integrates seamlessly with standard field research workflows, offering user-friendly functionality while maintaining compatibility with environmental conditions and ecological goals.

This UPM equips users with the tools and knowledge to maximize the system's potential and contribute effectively to bat conservation objectives.

2 Overview

Bats play an important role in Canada's environment by pollinating, dispersing seeds, and controlling pests. However, Canadian bat populations are under threat from habitat loss, pesticide usage, and infections such as white-nose syndrome. Monitoring bat activity is critical to conservation efforts. Existing systems are either expensive or intrusive, which limits their general use. Our technology provides an inexpensive, user-friendly, and accurate option for bat conservation in Canada.

Designed for use in distant places with limited technical assistance, this device addresses major requirements for bat monitoring.

- Accurate identification of bat movement without false activations.
- Durability in tough Canadian environments (low temperatures and high humidity).
- Extended battery life allows for weeks of use without regular recharging.
- Simple data storage and retrieval for examining bat activity patterns.

The technology detects bat activity non-invasively using a laser sensor, which avoids the disruption created by previous approaches. It is small, portable, and extremely inexpensive, making it perfect for use by conservation organizations with limited resources.

System Features:

- To detect bat movement, use a laser sensor and receiver.
- To process data, use an Arduino microcontroller.
- Data logger with detachable SD card allows for quick data recovery.
- Lithium-ion battery designed for extended field use.
- The components are housed in a lightweight hardwood box, which ensures mobility and little environmental effect.
- This device is suited for long-term ecological study and is simply upgradeable, such as with temperature or humidity sensors.

2.1 Conventions

This guidebook is well-organized and easy to use, with steps offered in numbered or bulleted lists. Measurements and technical characteristics are presented in metric units (e.g., millimeters, degrees Celsius) in conformity with Canadian standards. A glossary is offered to help clarify phrases or symbols. This layout allows for simple navigation and efficient usage of the handbook.

2.2 Cautions & Warnings

- Sensor Alignment: Accurate detection requires proper alignment of the laser sensor and the receiver. Misalignment can lead to lost or inaccurate data.
- Battery Use: To avoid damage or safety issues, use only the specified lithium-ion batteries.
- Weather Protection: The system isn't completely watertight. Avoid being exposed to severe rain or snow unless you have adequate weather protection.
- Positioning: Keep the device at an adequate height and angle to avoid interference from non-target animals or human activities.

By taking these steps, users may assure the device's lifetime and accuracy, allowing for successful bat monitoring and helping to preserve Canada's biodiversity.

3. Accessing/Setting up the System

3.1 Physical Assembly Instructions

Below are the steps to assemble the bat box system. Use the provided illustrations for guidance.

Step 1: Mount the Bat Box Frame

Secure the bat box to a stable surface like a wooden pole or wall.

Use screws or brackets for permanent installation or adhesive for a temporary setup.

Ensure the frame's entry points are open for laser placement.
[Figure 1]: *Mounted bat box on a wooden pole.*

Step 2: Install Laser Transmitters (KY-008)

Position the laser transmitters inside the bat box opening.

Secure them using adhesive tape or screws to prevent movement.

Ensure they align horizontally, facing the opposite side where the receivers will be installed.
[Figure 2]: *KY-008 module securely attached to the bat box.*

Step 3: Install Laser Receivers (DAOKI)

Attach the laser receivers on the opposite side of the transmitters, ensuring the laser beam aligns perfectly with the receiver sensor.

Use small adjustments to align the beam precisely.

Secure them in place with adhesive or screws.

[Figure 3]: *Receiver positioned and aligned with the laser beam.*

Step 4: Wiring the System

Use the Dupont cables to connect the laser modules (transmitters and receivers) to the Arduino board.

Refer to the wiring diagram below for pin assignments:

Laser Transmitter (KY-008): Connect VCC, GND, and Signal to the Arduino.

Laser Receiver (DAOKI): Connect Power, GND, and Signal to the Arduino.
[Figure 4]: *Wiring schematic showing connections between sensors and the Arduino.*

Step 5: Power the System Connect the Arduino board to a power source, such as a USB cable or battery pack.

Verify the system powers on, and check that the laser beams are visible as red dots.
[Figure 5]: *Entire bat box system powered up with visible laser beams.*

3.2 Adjustments for Personal Use

Use a small flathead screwdriver to fine-tune the alignment of lasers and receivers.

Ensure there are no obstructions (like dirt or debris) in the beam's path.

Adjust the height or angle of the box to match the target environment.

3.3 Software Setup Instructions

Installing the Control Code

Open Arduino IDE:

Download and install the Arduino IDE on your PC or laptop.

Load the Program:

Import the provided `.ino` file into the IDE.

The program includes logic to detect beam interruptions and log the data.

Configure the Code (if needed):

Update parameters such as sensitivity or detection thresholds as necessary.

Upload the Code:

Connect the Arduino board to your PC via USB.

Click the **Upload** button in the IDE to transfer the code.

Testing the System

Once powered and programmed, interrupt the laser beam manually to test the system's response.

Observe the logs via the Arduino IDE Serial Monitor or SD card (if configured).

3.4 System Navigation

Key Features

Laser Detection: Monitors interruptions in the laser beam.

Arduino Processing: Records events and logs data.

Adjustable Components: Allows customization of beam alignment and detection range.

Navigation and Troubleshooting

Checking Logs:

Use the Serial Monitor in Arduino IDE for real-time data.

Realignment:

Adjust laser modules manually if the beam doesn't hit the receiver.

Power Reset:

Disconnect and reconnect the Arduino board if the system malfunctions.

3.5 Configuration Considerations

Physical Environment

The system performs best in areas free of external light interference and reflective surfaces.

Mount the bat box securely in a shaded, open area near bat habitats.

Tools Required

Screwdriver

Adhesive tape or screws

Laptop with Arduino IDE installed

Following these steps ensures a smooth setup and configuration process for your bat box system.

2.2.1 4.1 Using the System: Initialization

Figure 4.1-1: Powering On the System

- **Plugging in**: Connect the battery by inserting the connector of the battery into the connector. As shown below.

Figure 4.1-2: System Initialized

- **Purpose**: Once plugged in the system will initialize and go through the checks that everything works.
- **Serial monitor example**: If the Arduino was connected to a computer with the Arduino IDE open the serial monitor would display.

“File exists. Removing it.”

“Successfully removed existing file.”

“Bat detection system initialized.”

2.2.2 4.2 Using the System: Bat Detection

Figure 4.2-1: Detecting a Bat

- **Bat detection**: The Arduino will log a bat detection when the bat is blocking the laser from getting to the receiver for the set time period, depending on bat species.

Figure 4.2-2: Serial Monitor Detection Messages

- **Purpose**: If the Arduino was connected to a computer with the arduino IDE open and a bit tripped the sensor the serial monitor would display.
- **Sample of serial monitor:**
  “Bat visit detected. Total visits: 1 | Time interrupted: 2001ms”
  If the sensor was tripped but for not long enough it will display
  “Tripped! Time interrupted: 400ms”

2.2.3 4.3 Using the System: Logging Data

Figure 4.3-1: Accessing Data Logs

- **Remove SD**: Remove SD card from module located within the box.
- **Insert SD into computer**: Once removed, insert SD into SD slot of computer or dongle, the SD card's name is “17 Bats” once you open the SD in your computer the file “`bat_log.txt`” will appear and show how many bats entered the box.
- **Sample**: The file text will look something like this, depending on how many bats visit and how long the laser was tripped for each time.

  Bat visit detected. Total visits: 1 | Time interrupted: 2001ms

  Bat visit detected. Total visits: 2 | Time interrupted: 1600ms

  Bat visit detected. Total visits: 3 | Time interrupted: 4500ms

  Bat visit detected. Total visits: 4 | Time interrupted: 1599ms

- **SD Capacity:** Periodically check the SD card for storage capacity.

4.4 Features for Maintenance and Troubleshooting

2.2.4 If the SD card has no files on it after inserting into your computer, either no bats visited, or there is an issue so here is a guide to troubleshooting.

Figure 4.4-1: SD Card Error Message

- **SD Initialization error**: Plug the Arduino into your computer using the blue cable provided. Make sure to copy and paste the code provided earlier in the document into the main part of the Arduino IDE. Once plugged in, verify the code by hitting the check mark button on the top left, once it verifies it, hit the upload button, which is a small arrow beside the check mark on the upper left hand side of the screen. Once completed open the serial monitor by clicking the button that has a magnifying glass and dots, it is located on the top right hand side of your screen. If the serial monitor displays the message "Could not initialize SD card." First check all the wiring to make sure it is plugged in and matches the wiring diagram provided in this document, then upload the code again to test. Next is to make sure the SD card is formatted correctly. The format you want is FAT32, to check this open Disk Utility on macOS, or Disk Management on Windows. If it is using exFat make sure to reformat to FAT32 and then upload the code again to test.

Figure 4.4-2: Re-aligning the Laser

- **Checking Alignment, and Realignment**: Look into the box and if the laser is not directly lined up with the sensor make to move the laser gently with your finger in the desired direction(s) until it is lined up, once realigned it may say a bat was detected, this is because in the Arduinos eyes the laser

was interrupted for more than the allotted time and once the laser is hitting the receiver again it would say 1 bat visited.

3 Troubleshooting & Support Sean

Describe all recovery and error correction procedures, including error conditions that may be generated and corrective actions that may need to be taken. Organize the information in sub-sections as appropriate. The following are common sub-sections that may be included as appropriate. Remember that someone who is not an engineer must be able to follow these steps.

The two most common errors that may occur with our system are wires coming out of their pins and laser/receiver misalignment. If the system is not working the first thing to check is if there are any loose wires inside the electronic box. If there are, it is necessary to refer to the Arduino circuit images in the set up to reconnect the wires in the appropriate locations.

The second thing to check every month is the sensor alignment. To do this you must look just inside of the entrance to the bat box and on the left side where the receiver is located. If the red laser is not centered on the reciever, no data will be recorded. If the system is misaligned, the setup steps of the emitter and receiver will have to be repeated.

3.1 Error Messages or Behaviors

Identify the error messages or behaviors that a user may receive or parts that are prone to breaking and the likely cause(s) and/or possible corrective actions for the error. If the list is extensive, this information may be best provided in an appendix to the document that is referenced here.

The only error messages our team has encountered while testing our product is in the serial monitor of Arduino IDE saying the SD card could not be initialized. This error occurs when the wiring is incorrect or when the micro-SD card is not inside the module. If this error occurs, check that all the wires are in the correct location using our set-up information above and ensure the micro-SD card is inside the module.

3.2 Special Considerations

If applicable, describe any special circumstances, actions, caveats, exceptions, etc., that should be considered for troubleshooting.

As mentioned above, our counting system has a time parameter that dictates the minimum laser interruption time to increase the count. After researching the behaviors of bats entering and exiting the box, we set the minimum time to be 1500 ms (1.5 seconds). After installation and usage, if it observed that the bats are taking more/less time to enter, or the results from a month seem inconsistent with what is expected, it could be necessary to change this time. If this is the case, the best practice would be to consult a technician or engineer to have them update the code. If neither are available, follow these steps:

1. Open provided code in Arduino IDE
2. Locate the 8th line and change the value associated with "const unsigned long interruptionDuration" from 1500 to the preferred minimum interruption time for a count. Remember the value is in milliseconds.
3. Bring your computer with the Arduino IDE along with a USB cable to the bat box. Connect the Arduino in the electronic box to your computer with the USB cable.
4. Ensure the board is **Arduino Uno** and you have the correct port connected. This can be done from the tool tab.
5. Verify and then upload the new code to the arduino using the check mark and arrow buttons at the top left of the screen.
6. Test the new code by interrupting the laser for the updated minimum time. If it does not work, seek professional assistance.

3.3 Maintenance

Describe regular maintenance that should be performed on the prototype to avoid failure.

Monthly maintenance (Bring computer, mico-SD port, USB cable, new battery, scraper)

Important Note: Do battery replacement last as this will reset the count automatically.

Date collection: Ensure data collection is the first action taken so results are not skewed by unintentional laser interruption. Data collect steps are as follows:

1. Bring a computer with a micro-SD port or adapter to the bat box location
2. Open the electronics box and remove micro-SD card from the micro-SD card module (small rectangular component with card holder on top)
3. Insert the card into the micro-SD port on your computer and save the data within to a desired location.
4. Once saved, remove from your computer, place it back inside the micro-SD card module

Guano removal: Remove piled up guano through the entrance of the bat box using a scraper or other reasonable tools. Take care not to damage or misalign the emitter and the receiver.

Battery replacement: Locate battery inside electronics box and remove it from the battery connector. Insert the new battery the same way the previous one was connected. The connector can only go on one way. If the light on the Arduino starts to flash, the new battery was successfully connected.

3.4 Support

Provide information on how the user can get emergency assistance and system support (e.g., help desk support, production support, etc.). Include the name(s) of the people responsible and email addresses who serve as points of contact for system support. Also provide instructions for how identified problems with the system are to be reported. Include instructions for security incident handling, as appropriate.

The first point of contact would be engineers or technicians (if there are any) overseeing the tracking efforts for Tiree. Maintenance employees would have to have a method to get in touch with them.

If the engineers or technicians at Tiree can not solve the issue, Tiree would then have to reach out to our design team for further assistance.

4 Product Documentation

The final prototype can be organized in two different subsystems the box and the electronics component.

1.Box

The box will be housing the bats, and it was made by choosing minimal dimensions for a bat box using plywood since it is the material usually chosen for bat boxes as it is the closest to the natural bat environment. Notches were added to the landing pad and inside the box so the bats could grip to the material. It was assembled by using nails.

The electronics box was attached to the box with nails made by 3D printing the lid and the box it has a hinge and a latch to secure the electronics inside.

2.Electronics

The main goal of the electronics was to monitor bat visits so to build a subsystem the monitoring alone is not sufficient. Solutions needs to be found on what will be used but also how. The bats' visit will be monitored with a laser and a laser sensor, since blocking the path of the light will be detected. However, a code has to be made to relay instructions that will interpret said interruption of path of light, which is why an Arduino board was used. Not only that, but data also has to be retrieved, an SD card and its module for Arduino compatibility allows for efficient data retrieval and analysis. Power is also an aspect that needs to be considered as the bat monitoring system will be in a remote location, a battery is an adequate choice. For this project a 9V alkaline non rechargeable battery was used but a battery between 5 and 12 V would also be compatible with the Arduino Uno 3. The number of mAh is also an important aspect to consider as calculations have shown that the minimal amount for around 35 days of data recording is 2400 mAh. Male-male and female-male wires were also used as well as connectors, they all had connectivity roles in order to complete the circuit.

2.1. Code and data retrieval

A code had to be developed and needed to fulfill two different roles (see annex for the detailed code):

1. Accurately count bat visits
2. Have the ability to log the data onto the SD card

The code can be modified to adapt to the speed at which bats enter the box to do so only the line 8 needs to be modified.

Before using the SD card it needs to be formatted to FAT16 or FAT32 to do so (on Mac):

1. Connect the SD card to the computer
2. Open the disk utility
3. Click on SD card then on erase
4. Enter a new SD card name
5. Select a new format FAT32
6. Click on erase
7. Click on done

4.1 Building the box and the sensor system

4.1.1 BOM (Bill of Materials)

Bill of Materials				
Item #	Item Description	Quantity	Unit Price	Amount (after tax)
1	Micro SD Card Module	1	$ 10.98	$ 12.41
2	Battery 9V	1	$ 4.00	$ 4.00
4	Arduino	1	$ 15.25	$ 17.23
5	Laser Sensor	1	$ 16.50	$ 18.97
6	Micro SD Card 64GB, SD Adapter	1	$ 14.44	$ 16.32
7	Bread Board	1	$ 5.00	$ 5.65
8	Battery Connector 9V	1	$ 2.00	$ 2.00

9	Cord	1	$ 2.75	$ 3.11
10	20cm Male-Male Jumper Cables	10	$ 1.00	$ 1.00
11	USB and SD card reader for computers (if needed)	1	$ -	$ -
12	Screws	1	$ -	$ -
13	Wood	24*16*0.25 in 35.25*16*0.25in 24*4*0.25 in (x2) 4*16*2.25in 16*4*0.25 in	$ -	$ -
14	Hinge	1	$ -	$ -
15	Latch	1	$ -	$ -
			Total	$ 80.69

4.1.2 Equipment list and Software

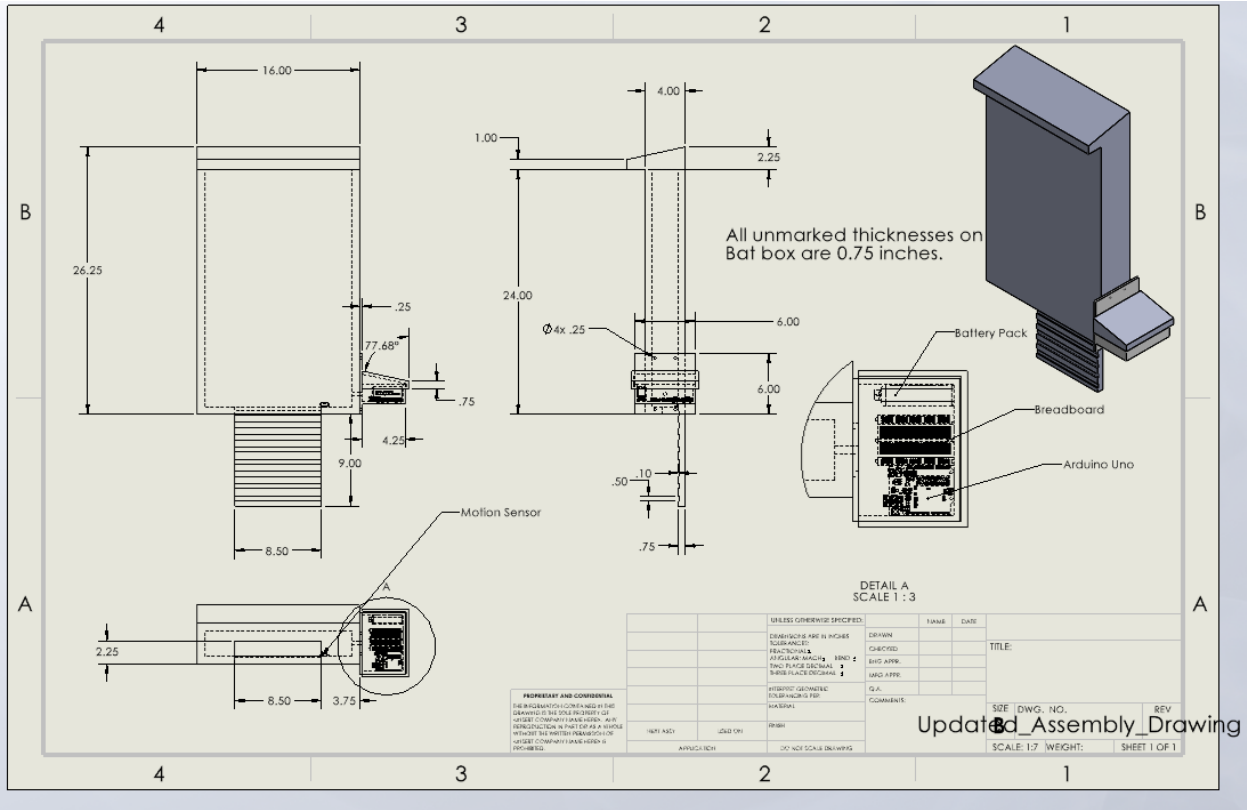
- Drill

- Saw
- Screwdriver
- 3D printer
- Multimeter
- Ruler
- Computer
- Arduino IDE

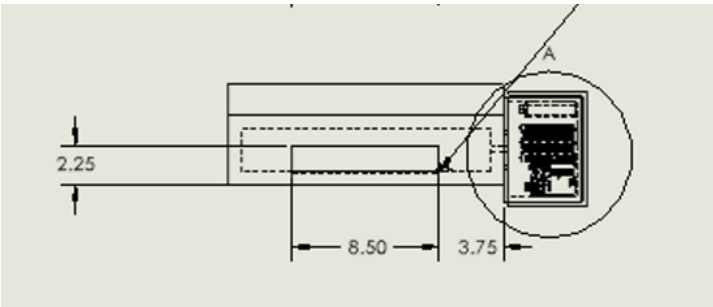
4.1.3 Instructions

This is a guide to build the box, electronics box and the circuit.

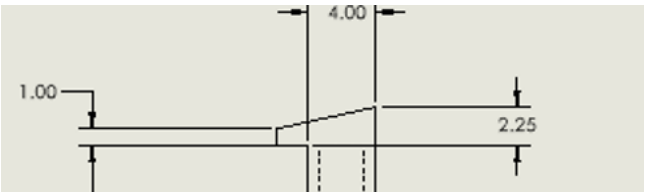
Box



1. Start by using the 35.25*16*0.25in piece of wood and trace a 9*8.5 in rectangle at the bottom center of it, it will be the landing the landing pad.
2. Then using the 16*4*0.25 in piece of wood draw a 2.25 *8.5 in rectangle at the bottom center of it, this will be the bat entrance



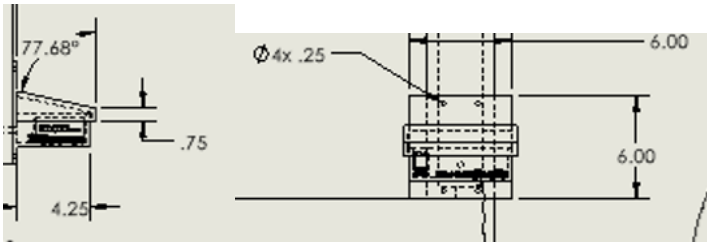
3. Next, using the 4*16*2.25in piece of wood create a 75 ° angle so the thickness of the lid will be 2.25 in at the back and 1.00 at the front



4. Saw outside of the lines for all the three pieces
5. Now add horizontal notches to all of the interior surfaces of the wood (they will be the interior surfaces of the box), as well on the landing pad (from step 1) about ½ inch from each other
6. It is now time to assemble, start by nailing together the piece from step 1, the 2 24*4*0.25 in and the 35.25*16*0.25in
7. Affix the lid and the bottom entrance with nails
8. Drill a hole on the side the electronic box will be on as the sensor wires will go directly through it

Electronic box

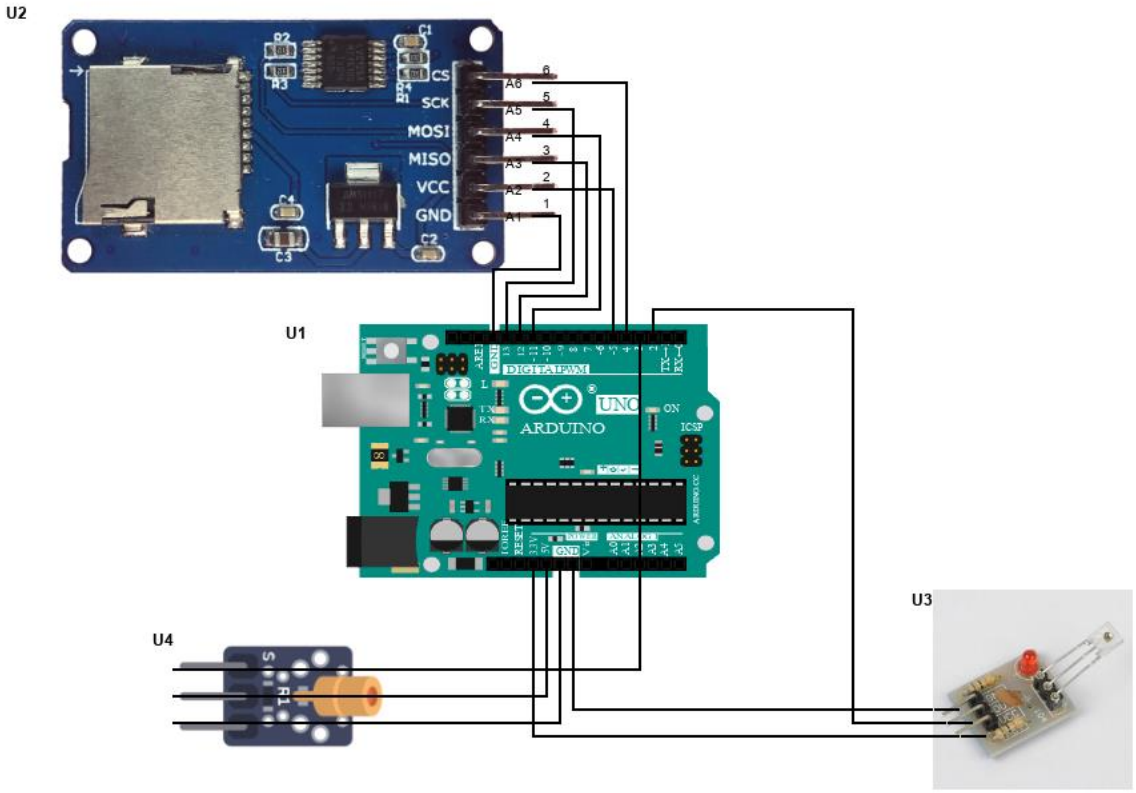
1. Create a box using CAD by creating the lid and the box separately a tutorial on how to create a box with its lid can be found here but the dimensions should be different and should be as followed.



2. The file should then be uploaded to the 3D printer
3. After the printing job is finished nail the hinge at the back to connect the lid and bottom of the box and the latch at the front.
4. 2 hollow protective pieces (also 3D printed) will also be done measure the height of the sensor and the height of the laser to create them they should be placed on the box in a way that will allow the sensor to be perfectly hit by the laser
5. Drill a hole on the side the electronic box will be connected to the box on as the sensors wires will go directly through it

Electronics wiring

The wiring of the electronics is as follows;



Component	Pin on component	Pin on Arduino
Laser	S	3
	middle	5V
	-	GND
Sensor	GND	GND
	OUT	2
	VCC	3.3V
SD-Card module	GND	GND(digital)
	VCC	~5
	MISO	12
	MOSI	~11
	SCK	13
	CS	4

Getting the system working

1. Connect the computer to the Arduino board by using the cord
2. Connect the SD card to the SD card module
3. Copy and paste the code in the Annex in Arduino IDE by going on File→New sketch
4. Upload the code
5. To view triggers go on tools→serial monitor to test the functionality of the sensor

Assembly

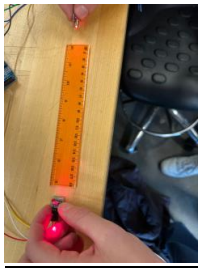
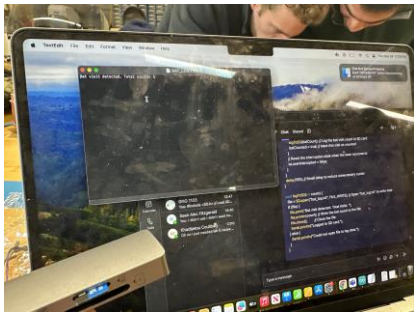
1. Drill on the electronic box on the main bat box
2. Place the laser and the sensor (affixed on the protective pieces) on opposite ends they have to perfectly align. The sensor will directly connect to the electronics box so its wires should go through the holes previously drilled.
3. Prepare the circuit as mentioned in electronics wiring but this time connect the battery to the Arduino board by using the connector and plugging the – to GND and the + to Vin and making sure all the components are fully in and make sure the micro-SD is in its module.
4. Close the lid and latch the box.

4.2 Testing & Validation

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	Using a multimeter to assess power consumption of the circuit and use the average mA of our battery to make our calculations

	frequency for field deployment.	
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
Test	Results	
Validate laser sensor's detection accuracy+ range and alignment	The detection was tested alongside the code the circuit was also re-evaluated. Minimal trigger times were set. 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.	

	The side facing up should be 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. The message appearing for successful logging should be as followed after insertion of the SD card. 
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 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.

5 Conclusions and Recommendations for Future Work Ana

Summarize your lessons learned and your work related to your prototype and suggest the most productive avenues for future work so that other groups can continue and improve upon your work.

What would you do if you had a few more months to work on this project? What are the things that your abandoned because of lack of time but would be important to add?

6 Bibliography

Conn, F. (2024, September 20). *Bat box presentation V2* [Presentation]. Tiree FS Inc.

APPENDICES

7 APPENDIX I: Design Files

Summarize the relationship of this document to other relevant documents. Provide identifying information for all documents used to arrive at and/or referenced within this document (e.g., related and/or companion documents, prerequisite documents, relevant technical documentation, etc.).

Include all design files in MakerRepo.

Also provide the MakerRepo link to your project.

Code used for the logging and saving

```
#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.");
  }
}
```

Table 3. Referenced Documents

Document Name	Document Location and/or URL	Issuance Date

8 APPENDIX II: Other Appendices

You can include other critical and important work here. Maybe they are not important in the structure of this document but need to be included.