ECE 44X Project Documentation

Max Garcia-Matzumilla, Marcus Plumley,
Abdullhmeed Alkawi and Sawyer Brundage

May 2022

Contents

1 Overview **6**
1.1 Executive Summary 6
1.2 Team Communication Protocols and Standards 6
1.3 Gap Analysis 7
1.4 Timeline/Proposed Timeline 8
1.5 References and File Links 8
1.5.1 References 8
1.5.2 File Links 8
1.6 Revision Table 8

2 Requirements Impacts and Risks **10**
2.1 Requirements 10
2.1.1 10
2.1.2 10
2.1.3 10
2.1.4 10
2.1.5 11
2.1.6 11
2.1.7 11
2.1.8 11
2.2 Impacts 12
2.3 Risks 13
2.4 References and File Links 14
2.4.1 References 14
2.4.2 File Links 15
2.5 Revision Table 16

3 Top Level Architecture **16**
3.1 Black Box Diagram 16
3.2 Block Descriptions 17
3.2.1 Camera 17
3.2.2 Power Supply 17
3.2.3 Clock 17
3.2.4 UI App 17
3.2.5 LED Lights 18
3.2.6 Microcontroller Code 18
3.2.7 PCB 18
3.2.8 Enclosure 18
3.3 Interface Definitions 19
3.4 References and File Links 20
3.4.1 References 20
3.4.2 File Links 20
3.5 Revision Table 20

4 Block Validations (OLD) 20
4.1 Camera + Code Validation 20
4.1.1 Block Overview 20
4.1.2 Block Design 21
4.1.3 Block General Validation 22
4.1.4 Block Interface Validation 23
4.1.5 Block Testing Process 24
4.1.6 References and File Links 24
4.1.7 Revision Table 24
4.2 Power Supply Validation 25
4.2.1 Block Overview 25
4.2.2 Block Design 25
4.2.3 Block General Validation 25
4.2.4 Block Interface Validation 27
4.2.5 Block Testing Process 29
4.2.6 References and File Links 29
4.2.7 Revision Table 30
4.3 Clock Validation 30
4.3.1 Block Overview 30
4.3.2 Block Design 30
4.3.3 Block General Validation 34
4.3.4 Block Interface Validation 35
4.3.5 Block Testing Process 35
4.3.6 References and File Links 36
4.3.7 Revision Table 36
4.4 UI App Validation 36
4.4.1 Block Overview 36
4.4.2 Block Design 36
4.4.3 Block General Validation 37
4.4.4 Block Interface Validation 37
4.4.5 Block Testing Process 39
4.4.6 References and File Links 39
4.4.7 Revision Table 39
4.5 LED Light Strip Validation 39
4.5.1 Block Overview 39
4.5.2 Block Design 40
4.5.3 Block General Validation 41
4.5.4 Block Interface Validation 42
4.5.5 Block Testing Process 43
4.5.6 References and File Links 44
4.5.7 Revision Table 44
4.6 Microcontroller Code Validation 44
4.6.1 Block Overview 44
4.6.2 Block Design 44
4.6.3 Block General Validation 46
4.6.4 Block Interface Validation 46

4.6.5 Block Testing Process . . . 49
4.6.6 References and File Links . . . 50
4.6.7 Revision Table . . . 50
4.7 Microcontroller Validation . . . 50
4.7.1 Block Overview . . . 50
4.7.2 Block Design . . . 50
4.7.3 Block General Validation . . . 52
4.7.4 Block Interface Validation . . . 53
4.7.5 Block Testing Process . . . 54
4.7.6 References and File Links . . . 54
4.7.7 Revision Table . . . 55
4.8 Enclosure . . . 55
4.8.1 Block Overview . . . 55
4.8.2 Block Design . . . 55
4.8.3 Block General Validation . . . 60
4.8.4 Block Interface Validation . . . 61
4.8.5 Block Testing Process . . . 61
4.8.6 References and File Links . . . 62
4.8.7 Revision Table . . . 62

5 System Verification **62**
5.1 Universal Constraints . . . 62
5.1.1 The system may not include a breadboard . . . 62
5.1.2 The final system must contain both of the following: a student designed PCB and a custom Android/PC/Cloud application . . . 62
5.1.3 If an enclosure is present, the contents must be ruggedly enclosed/mounted as evaluated by the course instructor . 63
5.1.4 If present, all wire connections to PCBs and going through an enclosure (entering or leaving) must use connectors . . 63
5.1.5 All power supplies in the system must be at least 65% efficient. . . . 63
5.1.6 The system may be no more than 50% built from purchased 'modules.' . . . 63
5.2 Wall Power Requirement . . . 63
5.2.1 Requirement . . . 63
5.2.2 Testing Process . . . 63
5.2.3 Testing Evidence . . . 63
5.3 Portability Requirement . . . 64
5.3.1 Requirement . . . 64
5.3.2 Testing Process . . . 64
5.3.3 Testing Evidence . . . 64
5.4 Durability Requirement . . . 64
5.4.1 Requirement . . . 64
5.4.2 Testing Process . . . 64
5.4.3 Testing Evidence . . . 64

5.5 Proof of Concept Requirement 64
5.5.1 Requirement 64
5.5.2 Testing Process 64
5.5.3 Testing Evidence 65
5.6 Bird Detection Requirement 65
5.6.1 Requirement 65
5.6.2 Testing Process 65
5.6.3 Testing Evidence 65
5.6.4 Requirement 65
5.6.5 Testing Process 65
5.6.6 Testing Evidence 65
5.7 Survey Requirement 65
5.7.1 Requirement 65
5.7.2 Testing Process 66
5.7.3 Testing Evidence 66
5.8 UI Requirement 66
5.8.1 Requirement 66
5.8.2 Testing Process 66
5.8.3 Testing Evidence 67
5.9 References and File Links 67
5.9.1 References 67
5.9.2 File Links 67
5.10 Revision Table 67

6 Project Closing **67**
6.1 Future Recommendations 67
6.1.1 Technical Recommendations 67
6.1.2 Global Impact Recommendations 68
6.1.3 Teamwork Recommendations 68
6.2 Project Artifact Summary with Links 69
6.3 Presentation Materials 69

1 Overview

1.1 Executive Summary

This document is to inform stakeholders about the current progress on the active bird deterrent system, and it also provides somewhere for the project to be referenced and monitored from. This project will be designing a system that will actively deter birds from colliding with windows. The approach will be to use a machine learning camera to detect birds approaching the window, and then triggering a string of lights to deter birds from colliding with the window.

There have been previous designs for this problem, and our goal will be to use their results and improve upon them. Specific areas that we are changing are the power source, use of the ultrasonic sensor, the amount and brightness of the deterrent, and use of a microcontroller. How the team will make these changes is by using a power supply that will be a standard wall supply for permanent use of the system. In addition to that, the design will no longer use the ultrasonic sensor in the design because it was found to slow the system down. Finally, a microcontroller will be used for the system dynamics and control.

The project is currently in the research and design stage where members are each completing a Risk Impact Assessment along with technical research on electrical systems that will be applied to the device. What the group learns in this research and design stage will help refine and clarify the scope and requirements of the project.

1.2 Team Communication Protocols and Standards

Table 1: Communication Information

Student Name	Email Address
Abdullhmeed Alkawi(Hameed)	alkawia@oregonstate.edu
Sawyer Brundage	brundasa@oregonstate.edu
Max Garcia-Matzumilla	garciamm@oregonstate.edu
Marcus Plumley	plumleym@oregonstate.edu
Jerrad Jones	jonesger@oregonstate.edu

Table 2: Team Protocols and Standards

Protocol	**Assessment Planner**
On time deliverables and team collaboration	Distributed tasks should be met and accomplished by the deadline set for each one.
Distributing project tasks	During a team meeting, the team is going to list the upcoming tasks and distribute the tasks between the team members. Each team member is going to give an update of their tasks.
In case if a group is unable to join the meeting	Any group member who is unable to join the team meeting must notify the rest of the group in advance.
Communication protocol on Discord	Each team member must check discord at least once a day.

Communication and meetings with project partner are going to be via Email and Zoom. The design and project updates will be communicated via email. In addition, the meeting with Jerrad will mainly be on Zoom.

1.3 Gap Analysis

Each year, approximately half a billion birds die due to collisions with windows, so the purpose of the project will be to reduce the number of collisions birds have with buildings [1]. Currently bird collisions are mostly reduced using passive deterrent methods such as more reflective windows or adhesive decals, but the results are not effective [2]. The design is that it will need to recognize birds flying toward the windows and deter them. That is where the project will improve on prior methods because our system will actively deter the birds from collision via a strobe effect. As there is no product in the market currently, an assumption we have is the demand for this device would be considerable with a commercial customer base. However, the project partner foresees that the most demand will originate with homeowners who are conscientious for the environment and/or birds.

1.4 Timeline/Proposed Timeline

Figure 1: Project Timeline (Critical Path in Black)

1.5 References and File Links

1.5.1 References

[1] H. Powell, "Why birds hit windows-and how you can help prevent it," All About Birds, 11-Sep-2020. [Online]. Available: https://www.allaboutbirds.org/news/why-birds-hit-windows-and-how-you-can-help-prevefnt-it/. [Accessed: 16-Oct-2021].

[2] S. Milius, "Windows may kill up to 988 million birds a year in the United States," Science News, 08-Aug-2019. [Online]. Available: https://www.sciencenews.org/article/windows-may-kill-988-million-birds-year-united-states. [Accessed: 16-Oct-2021].

1.5.2 File Links

1.6 Revision Table

11/29/2021	Hameed Alkawi: Communication protocols revised.
11/29/2021	Sawyer Brundage: Created a Gantt Chart for the Timeline, improving readability and clarity.
11/29/2021	Marcus Plumley: Revised the gap analysis to include the end-user statement.
11/19/2021	Marcus Plumley: Cited sources in gap analysis.
11/19/2021	Sawyer Brundage: Updated Timeline and References formatting
10/29/2021	Marcus Plumley: Updated executive summary with information from the project partner.
10/29/2021	Sawyer Brundage: Updated Timeline and tables.
10/16/2021	Hameed Alkawi: Initial Document Creation. Content for section 1 added. Updated reference links to match IEEE format.

2 Requirements Impacts and Risks

2.1 Requirements

2.1.1

Project Partner Requirement: System must be powered by a wall outlet.
Engineering Requirement: System will draw from standard 60Hz 120V AC outlet.
Testing Method:

1. Check for nominal 5V at the PCB connection.
2. The main source is a power supply connected to a wall outlet.

2.1.2

Project Partner Requirement: The system must be durable
Engineering Requirement: System will withstand forces of at least 15 lbs on each side.
Testing Method:

1. Place system on a scale.
2. Zero out the scale.
3. Apply force of at least 15 lbs on each side of the system.

2.1.3

Project Partner Requirement: Strobing LEDs must be able to deter birds.
Engineering Requirement: The team will conduct an experiment whether strobing lights will deter birds.
Testing Method:

1. Set up a bird feeder.
2. Wait and flash birds that come to use the bird feeder.
3. Provide data with the project report.

2.1.4

Project Partner Requirement: System will have bird detection.
Engineering Requirement: The system will detect birds in camera view with 7/10 samples being correct.
Testing Method:

1. Set up system
2. Present bird objects in view of the camera
3. Testing proves that 70% of trials for approaching objects in the speed range trigger the deterrent system

2.1.5

Project Partner Requirement: System is weather resistant inside the enclosure.
Engineering Requirement: The enclosure (containing the PCB, camera, and power supply) must allow no more than 1 ounce of water to penetrate the interior when sprayed with a water gun containing 1 cup of water. The LED strip, when sprayed with the same amount, must continue working afterwards.
Testing Method:

1. Set up the system.
2. Spray enclosure with a water gun containing 1 cup of water.
3. Inspect Enclosure for water. It should not exceed 1 ounce.
4. Spray LED strip with a water gun containing 1 cup of water.
5. Test system to ensure all LEDs still work.

2.1.6

Project Partner Requirement: The system is aesthetically inoffensive.
Engineering Requirement: If asked whether they would mount the device to their window, at least 7/10 people would reply with "yes".
Testing Method:

1. Set up an anonymous survey online with picture of mounted system and question.
2. Put link to survey on engineering discord server.
3. Wait a week for responses.
4. Verify that at least 70% of respondents answered yes.

2.1.7

Project Partner Requirement: System must be portable.
Engineering Requirement: The enclosure (containing the PCB, camera and power supply) will weigh under one kg and have dimensions of 20 cm X 15 cm X 10 cm.
Testing Method:

1. Zero out scale
2. Place system on scale to verify that it weights under a kg
3. Use a ruler to measure each side of the system to verify the given dimensions.

2.1.8

Project Partner Requirement: The system must take user input, time configuration, to determine when the system is off or on.
Engineering Requirement: Internal clock will be configured through the app.
Testing Method: Testing shows the LEDs functioning in the defined time interval and non-operational outside the same interval.

1. Set up the system.
2. Show the LEDs are operational in the current time window.
3. Change the operational time window to outside the current time window.
4. Show the LEDs are no longer operational.

2.2 Impacts

The implementation of our system into the world will bring potential negative societal impacts in the areas of public health, cultural, environmental, and economic impacts. Some key recommendations for these areas would be controlling the intensity of the LEDs to prevent potential seizures from occurring reducing the impacts on public health [2], creating a window of operation to reduce the amount of light pollution into the atmosphere reducing the overall environmental impacts [3]. Changing the migration patterns of birds ultimately impacting the bird watchers that anticipate the arrival of birds reducing the impact in cultural impacts[4], and being able to arrange multiple LEDs around windows to reduce the economic impacts to implement our system [5] . Our team is aware of these potential risks and has prepared solutions if they arise.

2.3 Risks

Risk ID	Risk Description	Risk category	Risk probability	Risk impact	Performance indicator	Responsible party	Action Plan
R1	Over Budget	Cost	25%	L	Price Change/Additional Orders	Sawyer	Reduce further spending, and strategize how remaining costs will be split between team members.
R2	Part Delays	Timeline	75%	H	Changes in part arrival times	Marcus	Update timeline, retaining delays and continuing to design and build with available parts.
R3	Sick Teammate	Management	10%	L	Changes in Teammate Health	Max	Transfer intensive workload to be split among healthy teammates, giving sick teammates a lighter load.
R4	Team Conflict	Management	10%	M	Unproductive meetings/infighting	Hameed	Schedule an in-person meeting to make compromises and settle disagreements to avoid further conflict.

R5	Scope Creep	Management	10%	H	The additional requirements or blocks are added to the project.	Marcus	The team will avoid additional requirements/blocks, but if necessary reevaluate workload and the timeline
R6	Estimating/Scheduling Errors	Schedule	15%	M	Unable to conduct a meeting or cancellation of other commitments for a meeting.	Sawyer	Schedule a team meeting to reduce the loss of time and to reevaluate the timeline.
R7	Communication Breakdown	Management	30%	H	Teammates halt communications or there are 2 or more instances of miscommunication.	Hameed	Transfer risk by starting a discourse with OMBUDs or reduce risk by having a team meeting to resolve communication issues.
R8	Incompatible Interface	Technical	15%	M	Individual components of the system cannot communicate.	Max	Obtain new components that will communicate and work with one another.

2.4 References and File Links

2.4.1 References

[1] "How to manage engineering teams: Your college decision," Florida Tech | Florida's STEM University. [Online]. Available: https://www.fit.edu/your-college-decision/how-to-manage-engineering-teams/. [Accessed: 30-Oct-2021].

[2] "Birds and windows," Portland Audubon | Available: https://audubonportland.org/our-work/rehabilitate-wildlife/being.-a-good-wildlifeneighbor/birds-andwindows/: :text=Window0collisions [Accessed: 29-Oct-2021].

[3] "Bird migration is one of Nature's great wonders. here's how they do it.," P. Runwal | Available: https://www.nationalgeographic.com/animals/article/

bird-migration-one-of-natureswonders-heres-how-they-do-it.
[Accessed: 05-Dec-2021].
[4] “How to start birding,” Audubon.
| Available: https://www.audubon.org/birding/how-to-start-birding
[Accessed: 05-Dec-2021].
[5] “Global chip shortage ’is far from over’ as wait times get longer,” S. Yang and J. Sohn, “
Available: https://www.wsj.com/articles/global-chip-shortage-is-far-from-over-as-wait-timesget-longer-11635413402.
[Accessed: 30-Oct-2021]

2.4.2 File Links

2.5 Revision Table

11/29/2021	Max Garcia-Matzumilla: Formatted requirements into Project Partner and Engineering pairs.
11/29/2021	Hameed Alkawi: Added risks to table.
11/19/2021	Sawyer Brundage: Updated Sec. 2.4 formatting and risk table.
11/8/2021	Max Garcia-Matzumilla: Adding project partner requirements with the engineering requirements.
10/27/2021	Hameed Alkawi: Initial Document Creation. Content for section 2 added.

3 Top Level Architecture

3.1 Black Box Diagram

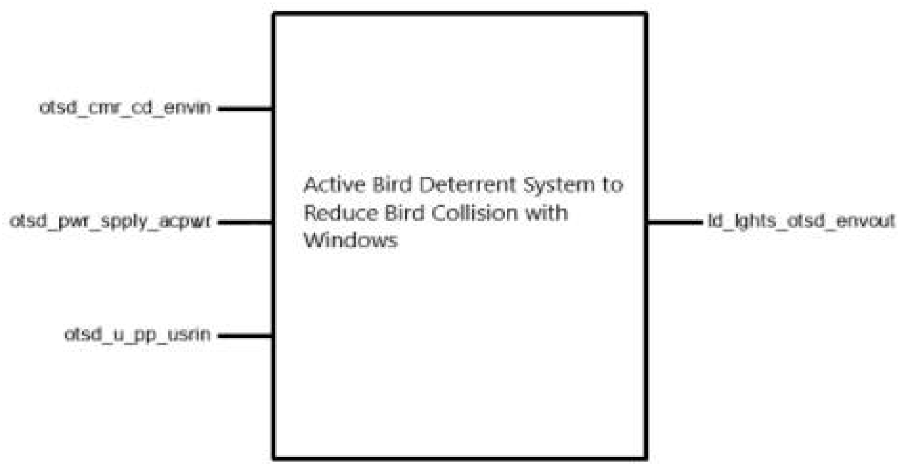


Figure 2: Black Box Diagram

3.2 Block Descriptions

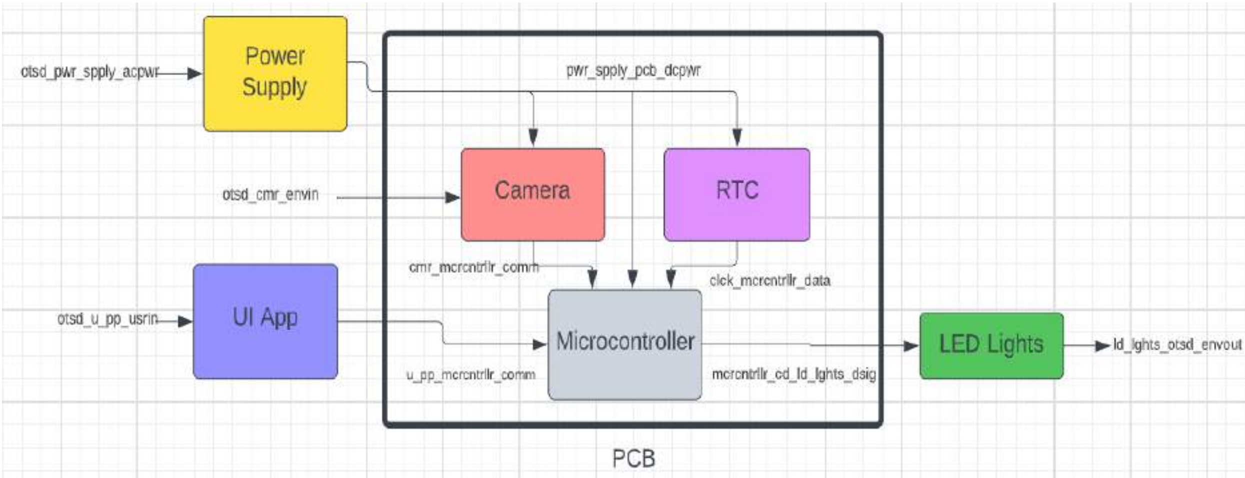


Figure 3: Block Diagram

3.2.1 Camera

Camera uses machine learning software to be able to detect birds traveling towards the window. When one is detected, it sends a signal to the microcontroller. **Champion**: Max Garcia-Matzumilla.

3.2.2 Power Supply

The power supply provides power to all components of the system. The supply will have 3.3 V and 5 V, and this will be done by stepping down 12 V with buck converter circuits using theTPS54232 buck converter chip. In addition to this, there will be a backup battery system that is attached that will provide power if the wall supply is off. These batteries will be lithium rechargeables. **Champion**: Marcus Plumley

3.2.3 Clock

The clock will keep track of the state of the device(on/off) as well as the real-world time. The data from the clock will be used by the system to operate within a specific time[1][2]. **Champion**: Hameed Alkawi

3.2.4 UI App

This block will give users the ability to adjust device settings, and it will have software and hardware components.The software portion will be an android phone app; the settings that can be adjusted from the app will be the brightness of the LED deterrent and the operational times for the system. The hardware portion will transfer the settings data via bluetooth with the HiLegto HC-05 bluetooth chip. **Champion**: Marcus Plumley

3.2.5 LED Lights

PWM signals will be sent from the microcontroller for each R, G, and B component of the LEDs. They will be arranged in a strip, which will wrap around the border of a window using adhesive. **Champion**: Sawyer Brundage

3.2.6 Microcontroller Code

Code that will enable the lights to flash when a bird has been detected, sending PWM signals generated by the microcontroller. It will also communicate with the clock module, which will tell it when it should and should send the signal to the LED lights. **Champion**: Sawyer Brundage

3.2.7 PCB

The PCB will contain the power supply, microcontroller, and other subsystems within other blocks. It will distribute power to the blocks that need it, along with carrying signals from the microcontroller itself. The microntroller used will be an Arduino Nano. **Champion**: Max Garcia-Matzumilla

3.2.8 Enclosure

The enclosure holds all parts of the system inside a water resistant enclosure, with opening for the camera and LEDs to run out of. Additionally, it is able to mount on walls. **Champion**: Hameed Alkawi

3.3 Interface Definitions

Interface	Interface Properties
otsd_pwr_spply_acpwr	V nominal: 120VAC I peak: 1 A I nominal: 250 mA
otsd_u_pp_usrin	Android App Configuration of system settings are allowed for operational times
pwr_spply_pcb_dcpwr	Inominal: 14 mA Ipeak: 20 mA Vmax: 5 V Vmin: 4.5 V
PCB_clk_dsig	Power dissipation is 80mW Peak forward current is 100mA
mcrcntrllr_ld_lghts_dsig	1 code is 0.8 us HIGH and 0.45 us LOW, and 0 code is 0.4 us HIGH and 0.85 us LOW Non-return-to-zero Communication Logic-Level: LOW = 0, HIGH = 5 V
ld_lghts_otsd_envout	Color alternates between white (on) and black (off) >=1000 Lux Frequency strobe: 1-2 Hz
otsd_cmr_envin	Other: Sampling rate of 15 frames/second Other: Visual Snapshots from the camera Other: A calibration frame is taken every 15th frame.
clk_mcrcntrllr_data	Outputs current time (Hour, Minute) Each device connected to the I2C bus is software addressable by a unique address I2C protocol allows multiple devices to be connected to the data bus
u_pp_mcrcntrllr_comm	Connect to HC-05 device to initiate Bluetooth communications Protocol: Bluetooth 2.0 Message will include operation times and current time
cmr_mcrcntrllr_comm	Boolean value, 1 for LEDs on and 0 for LEDS off Protocol: I2C communication Powered by 4-5 V

3.4 References and File Links

3.4.1 References

[1] Wwc and Instructables, "Setting the DS1307 real time clock using the serial monitor," Instructables, 22-Oct-2017. [Online]. Available: https://www.instructables.com/Setting-the-DS1307-Real-Time-Clock-using-the-Seria/. [Accessed: 19-Nov-2021].

[2] "Chronodot v2.1 high precision RTC," Macetech. [Online]. Available: https://shop.macetech.com/collections/non-led-components/products/chronodot-v2-1-high-precision-rtc. [Accessed: 19-Nov-2021].

3.4.2 File Links

3.5 Revision Table

12/3/2021	Marcus Plumley: Adjusted interfaces properties in the document and online tool based on feedback.
12/3/2021	Sawyer Brundage: Updated Block Diagram to improve readability.
11/29/2021	Hameed Alkawi: In text citation revised Hameed Alkawi: Fixed the format of section 3.2
11/29/2021	Marcus: included black box diagram for section 3.1
11/19/2021	Hameed Alkawi: Initial Document Creation. Content for section 3 added.

4 Block Validations (OLD)

4.1 Camera + Code Validation

4.1.1 Block Overview

This block is a combination of a camera and the code used for the camera. The camera will receive various inputs and the code will then process it to determine whether a bird is approaching. If so it will send a signal to the LEDs to flash and ultimately deter the bird from approaching the area. The ultimate goal of this block is to be able to detect an incoming bird.

4.1.2 Block Design

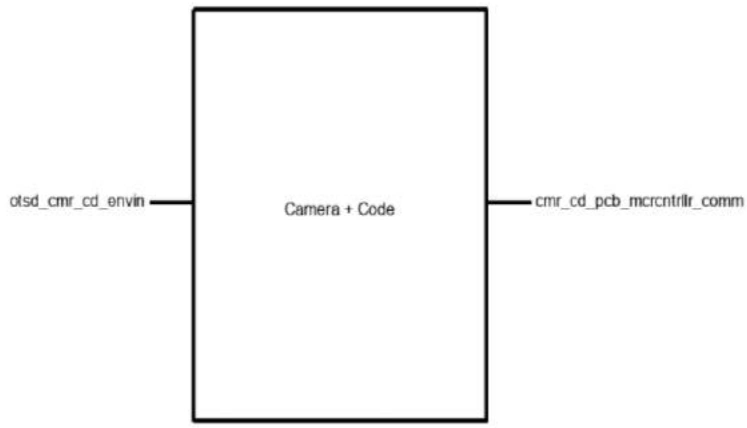


Figure 4: Black Box Diagram

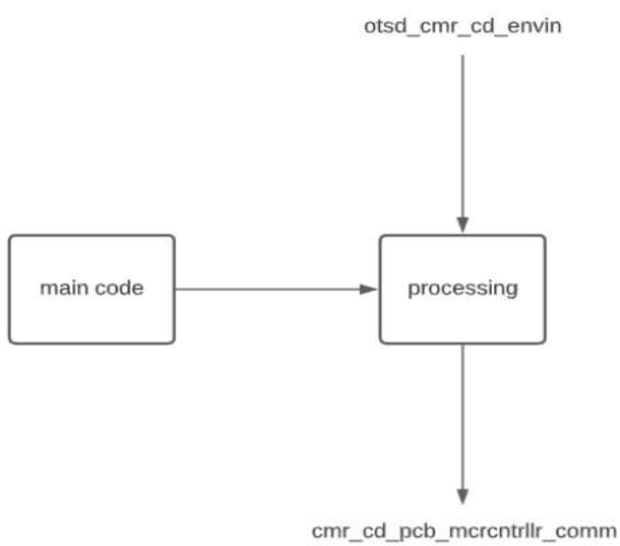


Figure 5: Top Level Diagram

Figure 6: Camera Schematic

4.1.3 Block General Validation

When reviewing the results of last year's design they implemented an infrared red camera and an ultrasonic sensor to detect birds but they encountered many issues going this route. For example, a big issue was that the ultrasonic sensor was receiving data that was too fast to process for their microcontroller.

Therefore, for this project, our group decided to just use the OpenMV H7 Plus camera module for detecting approaching birds. This camera is able to utilize a bird detecting database which it can use to determine whether a bird is approaching. Another pro to this camera is that it can use external storage in case we wanted to save the data for further inspection of the data we collect. We believe with this approach we will have an easier time detecting approaching birds and deter them from colliding with windows.

4.1.4 Block Interface Validation

otsd_cmr_cd_envin

Interface Property	**Why is this interface this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Other: Will be able to detect motion of an incoming object (birds) from at least 10 feet	Program needs to process images to detect whether a bird is approaching. 10 feet is used because a similar project had their camera about 5 feet away from where they wanted to make a recording.	The previous team's approach did a similar method of using an infrared camera but found it more difficult. Therefore we will use a regular esp32 camera.
Other: The angle of field of view will be 65 degrees	The field of view of the esp 32 camera is reported to be 65 degrees from the datasheet.	I know that this is appropriate because of the esp32 camera specifications.
Other: Will be able to detect objects in at least 100 lux	The esp 32 camera is able to see in daylight which is around lux. It cannot see well in darker conditions.	These values are valid because they are the lumens of daylight.

cmr_cd_pcb_mcrcntrllr_comm

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Data Rate: 9200 bits per second	The default camera operates at a baud rate of 9200 and the microcontroller is able to use 9200 as well.	Baud rate is supported by microcontrollers from Arduino and is the default for the camera.
Other: Output a JPEG	The esp 32 camera is able to output a JPEG it is seen on the page of link	It has been used in other projects that have you used the esp 32camera
Other: Microcontroller will power the camera with at least 5v	It is the suggested voltage used to power the device	This is a recommended voltage of the camera which can be seen in the datasheet

4.1.5 Block Testing Process

1) Install code and set up camera
2) Begin program to start detecting birds
3) Set up a bird feeder in range of the camera, about 20 feet away
4) observe whether or not a bird has been detected within the program.
5) Observe the rate at which snapshots have been taken and have been calibrated

4.1.6 References and File Links

4.1.6.1 References

4.1.6.2 File Links

1. OpenMV H7 Plus
2. Serial.Begin
3. Wire Library

4.1.7 Revision Table

1/19/2020	Max: edited and created new properties for environment inputs. Also adjusted some properties for the output.
1/8/2022	Max: documented created

4.2 Power Supply Validation

4.2.1 Block Overview

This block will provide 5 V up to 10A for the PCB and all modules connected to it. Those electrical specifications will be met with a prebuilt component; the device being used is the Mean Well LRS-50-5 switching power supply [1].

4.2.2 Block Design

This block has two interfaces entering it: otsd_pwr_spply_acpwr and pwr_spply_pcb_mcrcntrllr_dcpwr. These can be seen in Fig. 1. These interfaces are typical power supply signals; one is the AC signal from the wall outlet and the other is the DC power being provided through wiring from the power supply. Those inputs and outputs can be seen in Fig. 7.

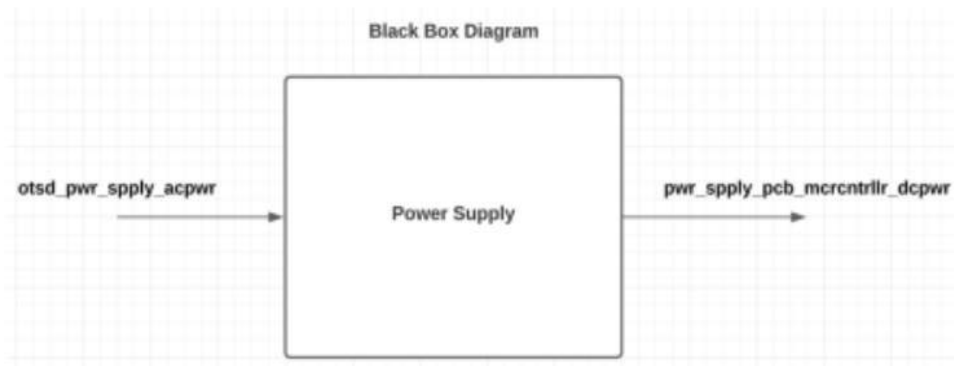


Figure 7: Power Supply Black Box Diagram

4.2.3 Block General Validation

For the validation of my prebuilt power supply, I will document the operating voltage and current per module. In doing so, I will show that the Mean Well LRS-50-5 Power supply meets the needs for the power supply block. These values are shown below:

*note: Arduino power consumption was estimated with the power consumption value for the microcontroller(19mA) and current draw from 1 PWM output(40mA).

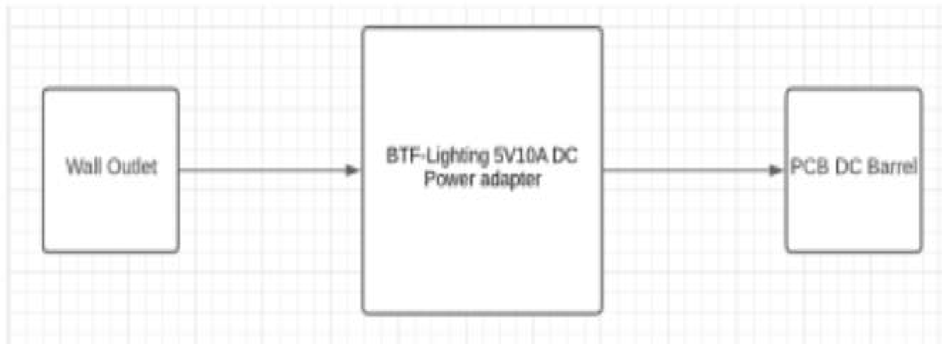


Figure 8: Interface Connections with the prebuilt BTF-Lighting 5V10A DC Power adapter.

Module	Operating Voltage	Operating Current
Arduino[2]	5 V	~60mA
Camera[3]	5 V	2 A
RTC[4]	5V	1.5mA
Bluetooth[5]	5V	30mA
LEDs[6]	5V	6A

The total operating current is around 8A, but the LEDs will only be operational in brief intervals to strobe. Therefore, the 8A will be considered the peak current draw while the nominal current draw will be all the modules but the LEDs. The nominal current needed then is about 2A. The Mean Well LRS-50-5 Power supply can provide 5V and up to 10A of current, so it meets the needs of the power supply block[6].

I had designed a switching power supply with the TPS54232D buck converter chip, but it did not meet the power needs of the system. However, given more time, I probably could have found a new chip to build a switching power supply that would meet the 5V 10A power specifications.

4.2.4 Block Interface Validation

otsd_pwr_spply_acpwr : Input

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Inominal: 2.5 A	The current requirements of the camera, microcontroller, RTC, and Bluetooth added together are about 2A, so to be on the safe side the power supply should be able to 0.5 A higher.	The Mean Well LRS-50-5 Power supply is rated to give up to 10 A of current but was designed to handle lower current thresholds as well [6].
Ipeak: 10 A	The current requirements of all the modules at peak operating current is about 8A, so to be on the safe side we wanted a supply that could provide up to 10 A.	The Mean Well LRS-50-5 Power supply is rated to give up to 10 A of current [6].
Vnominal: 120 VAC	The nominal voltage was chosen because it is the nominal voltage of the North America grid.	The Mean Well LRS-50-5 Power supply is rated to for 100-240 VAC [6].

pwr_spply_pcb_mcrcntrllr_dcpwr : Output

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Inominal: 2.5 A	The current requirements of the camera, microcontroller, RTC, and Bluetooth added together are about 2A, so to be on the safe side the power supply should be able to 0.5 A higher.	The combination of modules that will be always operational do not exceed 2.5 A.
Ipeak: 10 A	The current requirements of all the modules at peak operating current is about 8A, so to be on the safe side we wanted a supply that could provide up to 10 A.	All of the modules operating at the peak current will only pull about 8A of current, so they should not exceed 10 A.
Vmax: 5.5 V	All the modules operate at 5V but have some tolerance voltage, so max voltage that should be supplied should be only about 0.5V greater than the nominal.	The Mean Well LRS-50-5 Power supply assures for steady voltage output [6] with their switching power supply package, so that means the power supply will remain 0.5V within the nominal voltage.
Vmin: 4.5 V	All the modules operate at 5V but have some tolerance voltage, so min voltage that should be supplied should be only about 0.5V less than the nominal.	The Mean Well LRS-50-5 Power supply assures for steady voltage output [6] with their switching power supply package, so that means the power supply will remain 0.5V within the nominal voltage.

4.2.5 Block Testing Process

1. Connect an AC power meter to the wall outlet, and then connect the power supply to the AC power meter. That would be the input interface of the block, otsd_pwr_spply_acpwr.
2. Connect the power supply to an electric load at the nominal current of 2.5 A. That would be the output interface, pwr_spply_pcb_mcrcntrllr_dcpwr.
3. Run the system like this for about a minute to verify that the system has nominal voltage and current at the input that remain near nominal values. Also the output voltage should remain above Vmin, and the output current should remain near the expected nominal value.
4. Next, increase the electrical load to 10 A.
5. Run the system like this for about 10 seconds to verify that the peak current is never exceeded.

4.2.6 References and File Links

4.1.6.1 References

4.1.6.2 File Links

1. Power Supply
2. Arduino
3. Camera
4. [RTC
5. Bluetooth
6. [LEDs

4.2.7 Revision Table

1/21/22	Marcus Plumley: Finished general validation, interface validation, and verification for BTF-Lighting 5V10A DC Power adapter
1/21/22	Marcus Plumley: Finalized power budget estimate with new information; added new information on the BTF-Lighting 5V10A DC Power adapter
1/20/22	Marcus Plumley: Switched from original TPS54232D buck converter design to a pre-built component because previous design did not meet the needs of the system.
1/19/22	Marcus Plumley: Adjusted calculations to general validation by Don's suggestion
1/8/22	Marcus Plumley: Added file links
1/6/22	Marcus Plumley: Added Verification Plan
1/4/22	Marcus Plumley: Filled in the Interface Validation
12/29/21	Marcus Plumley: Added content into design and general validation
12/18/21	Marcus Plumley: Initial Document Creation.

4.3 Clock Validation

4.3.1 Block Overview

The real time clock module (RTC) will keep track of the state of the device(on/off) as well as the real-world time. The data from the clock will be used by the system to operate within a specific time range set by the user. The real time clock allows the project to keep track of time even if it is reprogrammed, or if the power is lost.

4.3.2 Block Design

The DS1307 module, used for this project, has the capability to install a 3-volt CR2023 backup battery.

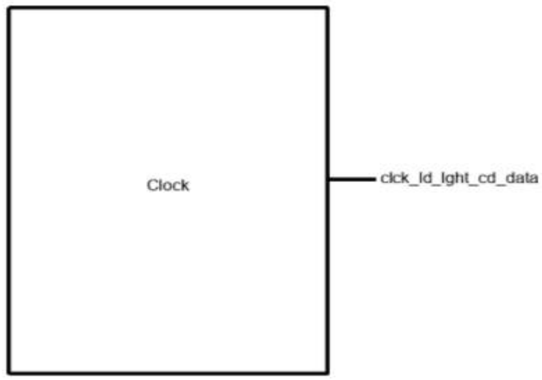


Figure 9: Black Box Diagram

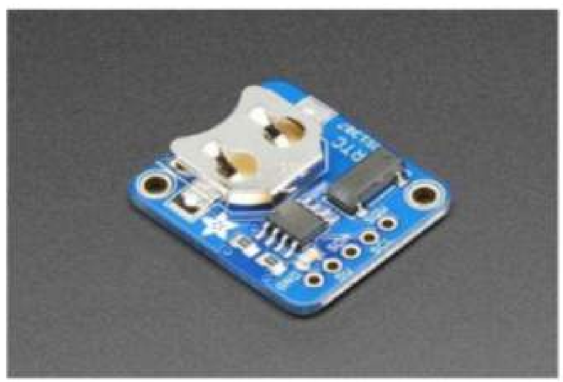


Figure 10: DS1307 RTC

Length	25.8mm/1.02in
Width	21.7mm/0.85in
Height	5mm/0.2in
Weight	2.3g/0.09oz

Figure 11: RTC Dimensions

Figure 12: CR2023 battery

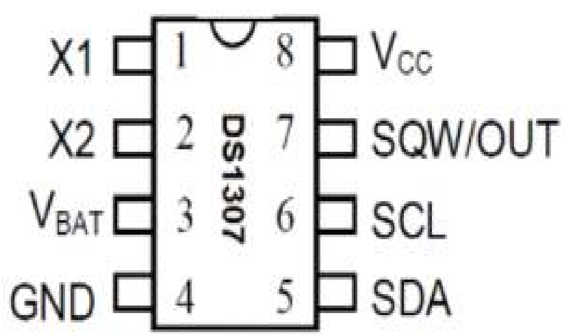


Figure 13: RTC pinout

X1 X2	X1 is for the connections for the 32.768kHz Quartz Crystal. This connection is essential for the purpose of this project, the bird deterrent system. The crystal oscillator is going to be used to calculate a precise 1Hz oscillation. Thus, to keep an accurate time keeping for the system. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1.
GND	Ground
SDA	Serial Data Input/Output. SDA is the data input/output for the I^2C serial interface.
SCL	Serial Clock Input. SCL is the clock input for the I^2C interface and is used to synchronize data movement on the serial interface.
SQW/OUT	Square Wave/Output Driver.
V_{cc}	Primary Power Supply.

Note: SDA and SCL, combined, represent the interface for this block (**clck_ld_lght_cd_data**)

Figure 14: Pin Descriptions

4.3.3 Block General Validation

The purpose of the block is to accurately keep track of time. The bird deterrent system allows the user to choose the time of the day when the system should be on or off. Thus, the RTC block meets the needs of the system. In other words, a "real time clock" is essentially a digital clock whose output can be read by a computer or microcontroller.

The module has its own oscillator that it uses to count time and it has registers that allow you to set the current time and date. The DS1307 communicates using the I2C bus in addition, this chip requires an external timing crystal which is installed in the module (figure 2).

The oscillator embedded in the module is going to calculate a precise 1Hz frequency. Having a 1Hz frequency means that a clock signal is created which provides 1s time resolution, which counts the seconds with a counter, do the math and you have a Real-Time Clock (RTC). The 32.768kHz oscillator is a power of 2, i.e. it is 2^{15}. Consequently, the clock frequency can be 15 divided to a 1 Hz frequency using binary counters. After that, the 1Hz frequency is going to be used for time keeping.

4.3.4 Block Interface Validation

clck_ld_lght_cd_comm : Output

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Other: Outputs current time (Hour, Minute)	The clock must let the Arduino know the current time so it can determine whether the device should be active or not.	According to the clock datasheet, it is capable of sending information regarding the current time through the I2C bus, which is received by the Arduino microcontroller.
Protocol: I2C protocol allows multiple devices to be connected to the data bus	The Arduino must be able to communicate with two other devices (camera and clock) at once.	According to the I2C manual, I2C buses allow for multiple devices to be connected, with one controller (the Arduino). Arduino microcontrollers are able to communicate using I2C with the Wire library.
Protocol: Each device connected to the I2C bus is software addressable by a unique address	The Arduino should be able to ask for information from either the camera or clock, specifying which one it wants to communicate with.	According to the I2C manual, each device is addressable by a unique address, and allows for one device to be the controller.

4.3.5 Block Testing Process

The following are the verification plan for the interface clck_ld_lght_cd_data

1. Wiring the block with the Arduino Nano.
2. Calculating 1Hz frequency from the 32.768 KHz oscillator in the module.
3. Write a function that takes in current time and outputs a boolean variable.
4. Let the user chose specific time of when the state of the system should be ON or OFF a. The LED represents the state of the system
5. Observe the change of the system while having an external clock such as the clock on phones for comparison.
6. To verify the I2C connections, write a function than scans the I2C address and prints the hexadecimal value of that address

4.3.6 References and File Links

4.1.6.1 References

[1] "Interfacing DS1307 RTC module with Arduino & Make a reminder," Arduino Project Hub. [Online]. Available: https://create.arduino.cc/projecthub/electropeak/interfacing-ds1307 -rtc-module-with-arduino-make-a-reminder-08cb61. [Accessed: 08-Jan-2022].

[2] "Chronodot v2.1 high precision RTC," Macetech. [Online]. Available: https://shop.macetech.com/collections/non-led-components/products/chronodot -v2-1-high-precision-rtc. [Accessed: 08-Jan-2022].

4.1.6.2 File Links

4.3.7 Revision Table

1/21/2022	Hameed: Interface properties revised. Verification steps revised. One more property added to the interface. Design section revised. Properties table revised.
1/7/2022	Hameed: Initial document creation

4.4 UI App Validation

4.4.1 Block Overview

This block is hardware that will take in the Bluetooth signal from the user interface and send it to the Arduino via serial communications. The data being transferred is the operational periods for the device, the brightness of the LEDs, and the current world time on the users phone. To accomplish this, the HiLetgo HC-05 Wireless Bluetooth RF Transceiver will be implemented to receive the Bluetooth signal from the user interface app and pass the message onto the microcontroller to be further processed.

4.4.2 Block Design

This block has two interfaces which are the wireless Bluetooth connection (u_pp_bltth_mdl_rf) and the physical serial communications connection (bltth_mdl_mcrcntrllr_comm). This is shown in the figure below:

The wireless connection will be done by pairing the user's phone to the HC-05 Bluetooth module. The connection between the HC-05 module and Arduino will be made by wiring the serial pins of the devices together. These connections can also be seen in wiring diagram for the HC-05 in Figure 2 below:

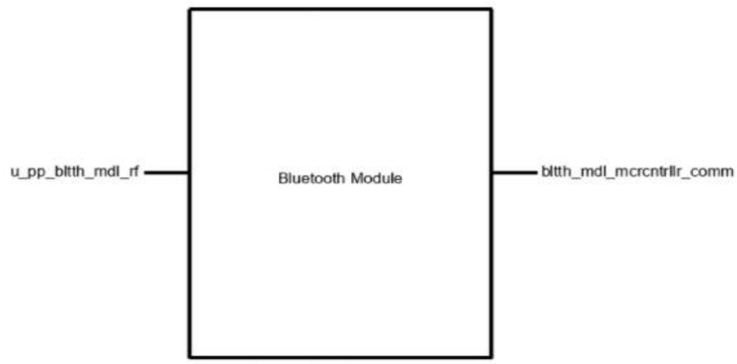


Figure 15: Black Box Diagram

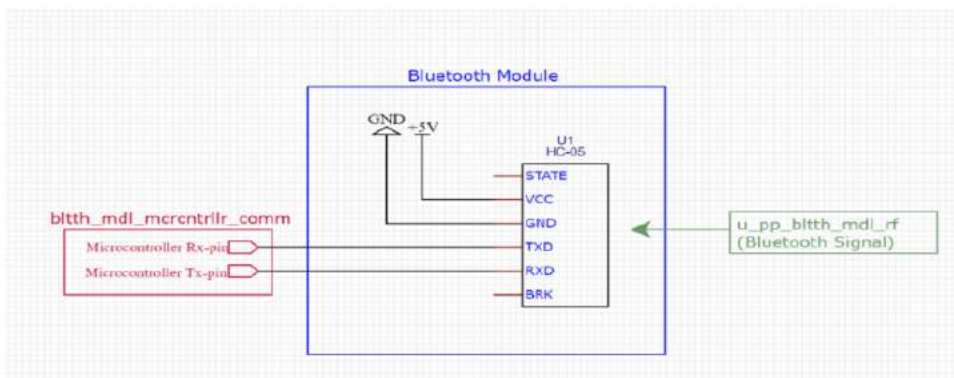


Figure 16: Bluetooth Module Wiring Diagram

4.4.3 Block General Validation

This design is based on other example projects that utilized the HC-05 module for wireless communications. However, the general validation for this module was done by physically implementing the device. Using the wiring described above, the module was connected to the Arduino which was programmed to print the messages received to the output monitor. Without the user interface fully implemented, the connection was simulated using the Serial Bluetooth app on the Google store. Typing out an array using the app resulted in the exact message being printed into the Arduino serial monitor which proved that this block meets the needs of this block.

4.4.4 Block Interface Validation

bltth_mdl_mcrcntrllr_comm : Output

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Other: Data input will be an array of data; the order will be brightness, operation times, and current world time.	This is so the microcontroller code knows the format and the order of data that will be transmitted on an update transmission.	The block exceeds this need because it is unrestrained by data format. It will be able to send an array of this size and order.
Protocol: Serial Port Communication	This is a communication standard for HC-05 module. It is also supported by Arduino.	This is the standard protocol used by the module [1].
Datarate: 9600	This is a common baudrate and is supported by both the Arduino and the HC-05 module[1].	Within the HC-05 datasheet [1], it states that it supports a baudrate of 9600.

u_pp_bltth_mdl_rf : Input

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Messages: The message will be an array of data; the order will be brightness, operation times, and current world time.	This is so the microcontroller code knows the format and the order of data that will be transmitted on an update transmission.	The block exceeds this need because it is unrestrained by data format. It will be able to send an array of this size and order.
Other: Connect to HC-05 device to initiate Bluetooth communications	This was decided as an interface property after it was needed for the user interface design.	The HC-05 is discoverable from the device name HC-05 [1].
Protocol: Bluetooth 2.0	This wireless protocol was chosen because the user will not need to be far from the device when changing the settings	The HC-05 uses Bluetooth 2.0 wireless protocols.

4.4.5 Block Testing Process

1. Configure the Bluetooth module and microcontroller as seen in Fig. 2.
2. Setup Bluetooth connection with the Serial Bluetooth app.
3. Show that the app the array of data ready to be sent via Bluetooth to the module.
4. Send the array to the HC-05 and wait for the Arduino to show that it has received the data.
5. Verify that the array matches.

4.4.6 References and File Links

4.1.6.1 References

4.1.6.2 File Links

1. HC-05

4.4.7 Revision Table

2/18/22	Marcus Plumley: Finalized document.
2/15/22	Marcus Plumley: Changed the verification process to only include the data transfer and Arduino confirmation.
2/13/22	Marcus Plumley: adjusted document for hardware block: remove design and interfaces for the user interface app, added Bluetooth wireless properties.
2/8/22	Marcus Plumley: Split the block into two blocks(Hardware: Bluetooth Module block and Software: User Interface Block) via Don Heer suggestion.
2/4/22	Marcus Plumley: Added interfaces for data properties
1/28/22	Marcus Plumley: Adjusted document content and formatting based third party review
1/25/22	Marcus Plumley: Initial Document creation

4.5 LED Light Strip Validation

4.5.1 Block Overview

This block is the component with which the system deters incoming birds. It consists of a strip of bright LED lights that are activated by the Arduino microcontroller. The strip can be cut into various lengths, which will be connected at right angles along the border of a window. When the system detects an incoming bird, it will signal the strip of WS2812B LEDs to flash white on and off at a frequency of between 1 and 2 Hz. The rest of the time, the LEDs are kept off to conserve power. This block is being championed by Sawyer Brundage.

4.5.2 Block Design

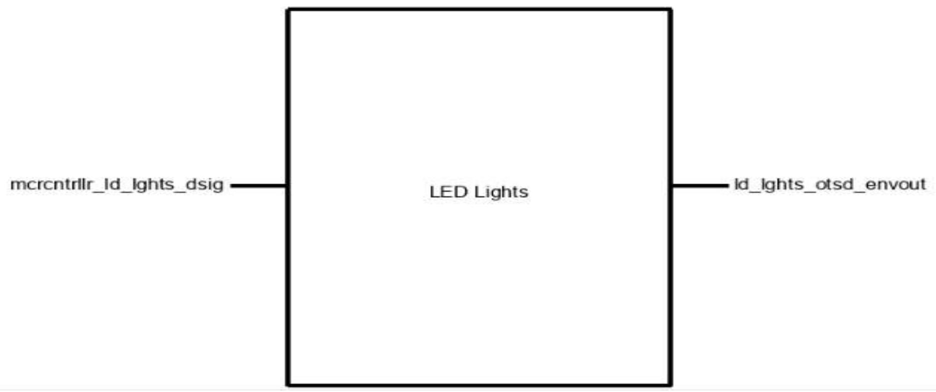


Figure 17: Black Box Diagram

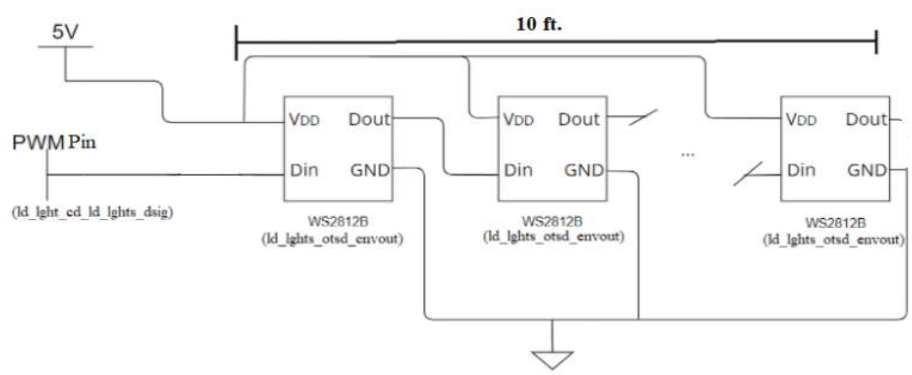


Figure 18: Wiring Diagram for LED Strip

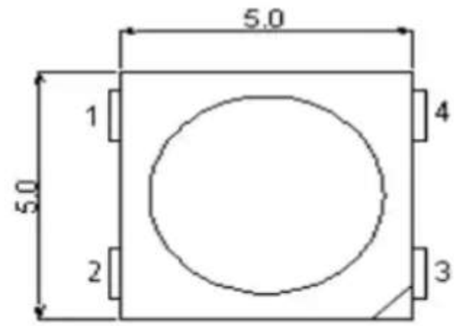


Figure 19: Dimensions (mm) of a WS2812B LED

4.5.3 Block General Validation

The use of a strip of LEDs, rather than simply 4 or 5 in a row, was agreed upon by our team after reviewing last year's project. Our group, along with our project partner, wanted to make sure that the device was something that could be easily mounted on a window and deter incoming birds from flying towards it. In order to make the LEDs especially apparent, a strip of LEDs spanning 10 ft. was chosen in order to easily wrap it along the border of a 2' x 3' window. This will make the flashing LEDs easily visible to approaching birds, which was an issue with last year's project that we hoped to remedy.

The reason why this design specifically uses WS2812B LEDs, rather than other types of cascading LEDs, comes down to the fact that they easily interface with Arduinos. Additionally, I have previous experience working with these particular LEDs. Our chosen WS2812B LED strip is also powered by 5 V, which is exactly what our power supply outputs to the PCB and other components. This way, we will not have to rely on the Arduino's 5 V pin, as it can be somewhat unreliable due to line loss, contact resistance, and current limits [1].

Continuing the subject of LEDs, I decided to have them flash white when a bird is detected, as opposed to some other RGB value. This is due to the fact that birds tend to avoid white colors in general, instead of more dull greens or browns that resemble trees or shrubs [2]. The LEDs are then set to strobe at a rate of 4 Hz in order to make them especially prominent and noticeable while also ensuring the rate is well below 16 Hz above which people with epilepsy could have seizures [3]. This allows the LEDs to grab the attention of incoming birds while also making sure pedestrians are not harmed by it.

The LED strip is bought pre-assembled, cutting down on the time needed to cut and reshape it into the appropriate shape for a window. This way, much of the engineering work will be focused on wiring it to the PCB and connecting it to the Arduino, as well as mounting it to a window.

4.5.4 Block Interface Validation

ld_lghts_otsd_envout : Output

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Light: Frequency strobe: 1-2 Hz	This frequency was chosen so the strobe would be visible and slow enough such that it is less likely to cause epileptic seizures	Using the delay() function in Arduino, the program can display one color to the LEDs, wait 125 ms, display another, wait 125 ms, and repeat. This allows the cycle to repeat 4 times a second (4 Hz).
Light: Color alternates between white (on) and black (off)	White was chosen as the LED color because research shows that birds tend to dislike it, making it an appropriate deterrence.	The LEDs are RGB, allowing them to emit a wide spectrum of colors with 24 bits of precision, such as white.
Light: >=1000 Lux	The lights must be able to be seen in daylight at a distance of around 10 ft.	According the LED specifications, they are able to emit more than 1000 Lux at full brightness

mcrcntrllr_ld_lghts_dsig : Input

Interface Property	**Why is this interface this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Logic-Level: LOW = 0 V, HIGH = 5 V	WS2812B LEDs set <= 0.3 * (VDD = 5V) = 1.5V as LOW, and >= 0.7 * (VDD = 5V) = 3.5V as HIGH.	From the Arduino website: Arduino digital pins either give 5V (when HIGH) or 0V (when LOW).
Other: Non-return-to-zero Communication	According to the WS2812B LED datasheet, this protocol is used to send data bits to each LED.	Arduino Nanos and Unos have specific PWM digital pins that are able to produce the waveforms needed for this protocol.
Other: 1 code is 0.8 us HIGH and 0.45 us LOW, and 0 code is 0.4 us HIGH and 0.85 us LOW	For a WS2812B LED, RGB bits are differentiated depending on how much time is spent HIGH and LOW, totaling to 1.25 us for each bit.	The fastLED library that is available for Arduino supports this protocol, allowing it to properly interface with the LEDs using these bit codes.

4.5.5 Block Testing Process

1. Cut the LED strip into two 2' and two 3' strips, joining different lengths at right angles.
2. Write a program in Arduino to alternate LED lights between white and black (off) at a rate of 4 Hz using fastLED library.
3. Hook up the Digital Pin used to generate a PWM signal to an oscilloscope, examining the input ld_lght_cd_ld_lghts_dsig.
4. Observe the resulting waveform from ld_lght_cd_ld_lghts_dsig using the oscilloscope, verifying the NRZ communication, the 0 and 1 waveforms, and the logic levels.
5. Hook up the strip to the test equipment (Arduino Nano, 5V power supply/function generator) and record a video of the LEDs flashing using the program in step 2.
6. Slow down the video to verify that the LEDs are 1) flashing on (white) and off (black) and 2) flashing at a rate of 4Hz with a tolerance of 0.25 Hz, according to the interface properties of ld_lghts_otsd_envout.
7. Measure the power draw using a DMM and verify that each LED consumes 0.3 W with a tolerance of 0.05 W when emitting a white, detailed in the properties of ld_lghts_otsd_envout.

4.5.6 References and File Links

4.1.6.1 References

[1] "Microcontroller power source considerations for Arduino," www.microcontrollertips.com. https://www.microcontrollertips.com/microcontroller-power-source-consideration s-arduino-faq/
[2] "Certain Exterior Paint Colors Repel Outdoor Birds and Bugs," Pinnacle Painting and Decorating, Jul. 16, 2019. https://paintwithpinnacle.com/2019/07/16/certain-exterior-paint-colors -repel-outdoor-birds-and-bugs/ (accessed Jan. 07, 2022).
[3] "Christmas lights causing epilepsy seizures," BBC News, Dec. 10, 2018. https://www.bbc.com/news/uk-england-nottinghamshire-46504844

4.1.6.2 File Links

1. fastLED Documentation, accessed 1/7/2022.
2. WS2812B LED Datasheet, accessed 1/7/2022.
3. LED Strip Store Page, accessed 1/7/2022.
4. Arduino PWM Signals, accessed 1/7/2022.
5. Arduino Documentation, accessed 1/19/2022.

4.5.7 Revision Table

1/20/2022	Sawyer Brundage: Fixed typos. Also revised interface properties and wiring diagram.
1/7/2022	Sawyer Brundage: Initial Document Creation

4.6 Microcontroller Code Validation

4.6.1 Block Overview

This block, programmed into the ATMega328P microcontroller, takes in the outputs from both the clock module and the camera through an I2C connection. Factoring the signals from these blocks, it determines whether it should flash the LEDs or keep them dark. If it receives the proper boolean signal from the camera, and the clock reports an appropriate time to the microcontroller, then the code will send a digital signal to the LEDs, causing them to flash. This block is championed by Sawyer Brundage.

4.6.2 Block Design

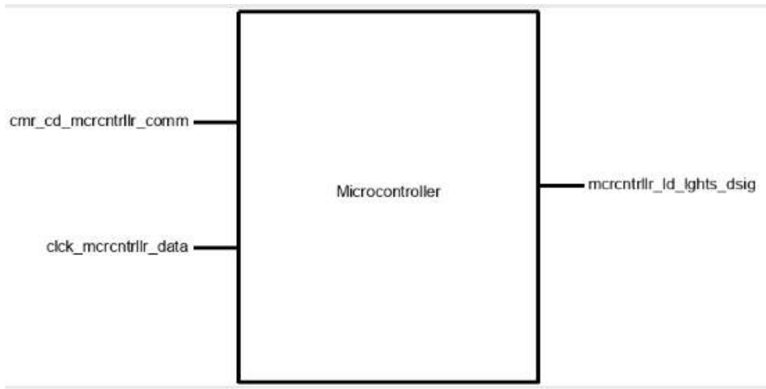


Figure 20: Black Box Diagram

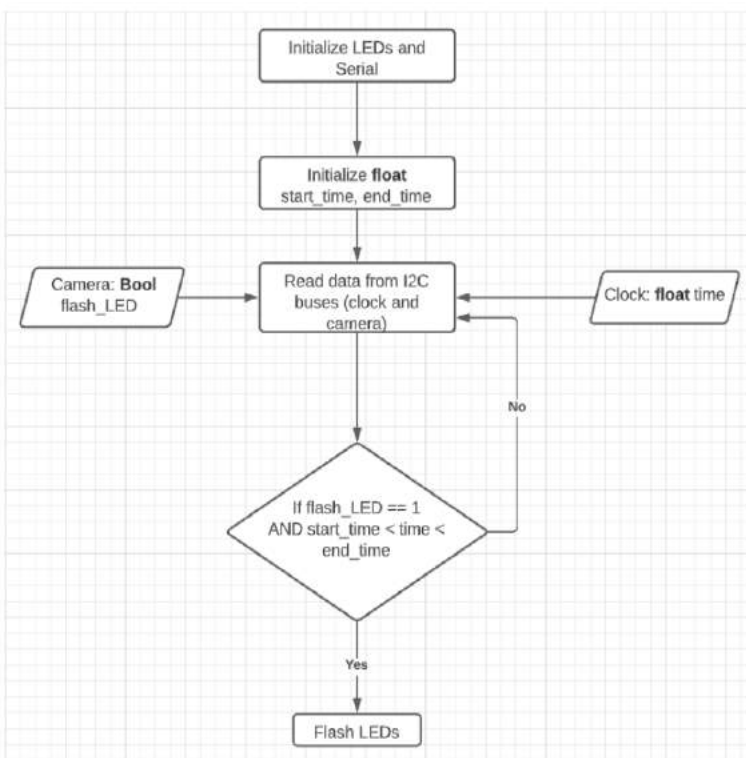


Figure 21: Code Flowchart

4.6.3 Block General Validation

This block takes in the data from other blocks (camera and clock) in order to determine the final output. For this reason, the microcontroller must open I2C communications with both the clock and camera. Since these modules are all powered by the 5V power supply, the physical connections will be made via the PCB.

According to the requirements of our system, the device must be inactive during user-determined periods of time. For this reason, the block must take in data from the clock and check if the current time falls within the desired range. Although the time is written to the microcontroller as a string, it can be easily converted to a float through a series of conversions and function calls (such as toFloat()). Additionally, it must take a boolean value from the camera, having its own code that tells the Arduino microcontroller if it has detected an incoming bird. In summary, if the Arduino receives a 1 from the camera (incoming bird has been detected) and the current time is within the appropriate range, then it will tell the LEDs to flash. Otherwise, it will continue reading from the clock and camera.

Since the Arduino microcontroller will be talking to two different devices, it must be able to choose which one it wants to communicate with at any given time. For this reason, the Arduino and the sensors will communicate using I2C protocol, which is supported by all three devices. When doing I2C communication, it is important to use pull-up resistors on the SDA and SCL lines in order to force them to default to Vcc during periods of no activity [1]. These resistors are incorporated into the design of the microcontroller, which is championed by Max Garcia-Matzumilla.

4.6.4 Block Interface Validation

mcrcntrllr_ld_lghts_dsig : Output

Interface Property	**Why is this interface this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Logic-Level: LOW = 0 V, HIGH = 5 V	WS2812B LEDs set <= 0.3 * (VDD = 5V) = 1.5V as LOW, and >= 0.7 * (VDD = 5V) = 3.5V as HIGH.	From the Arduino website: Arduino digital pins either give 5V (when HIGH) or 0V (when LOW).
Other: Non-return-to-zero Communication	According to the WS2812B LED datasheet, this protocol is used to send data bits to each LED.	Arduino Nanos and Unos have specific PWM digital pins that are able to produce the waveforms needed for this protocol.
Other: 1 code is 0.8 us HIGH and 0.45 us LOW, and 0 code is 0.4 us HIGH and 0.85 us LOW	For a WS2812B LED, RGB bits are differentiated depending on how much time is spent HIGH and LOW, totaling to 1.25 us for each bit.	The fastLED library that is available for Arduino supports this protocol, allowing it to properly interface with the LEDs using these bit codes.

clk_mcrcntrllr_data: Input

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Other: Outputs current time (Hour, Minute)	The clock must let the Arduino know the current time so it can determine whether the device should be active or not.	According to the clock datasheet, it is capable of sending information regarding the current time through the I2C bus, which is received by the Arduino microcontroller.
Protocol: I2C protocol allows multiple devices to be connected to the data bus	The Arduino must be able to communicate with two other devices (camera and clock) at once.	According to the I2C manual, I2C buses allow for multiple devices to be connected, with one controller (the Arduino). Arduino microcontrollers are able to communicate using I2C with the Wire library.
Protocol: Each device connected to the I2C bus is software addressable by a unique address	The Arduino should be able to ask for information from either the camera or clock, specifying which one it wants to communicate with.	According to the I2C manual, each device is addressable by a unique address, and allows for one device to be the controller.

cmr_mcrcntrllr_comm : Input

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Protocol: I2C protocol allows multiple devices to be connected to the data bus	The Arduino must be able to communicate with two other devices (camera and clock) at once.	According to the I2C manual, I2C buses allow for multiple devices to be connected, with one controller (the Arduino). Arduino microcontrollers are able to communicate using I2C with the Wire library.
Messages: Boolean value, 1 for LEDs on and 0 for LEDS off	The camera must send a simple signal to let the Arduino microcontroller know that a bird was detected.	The MicroPython library (language that the camera micro uses) supports boolean variables; so does Arduino.
Other: Powered by 5 V	The power supply provides a voltage of 5V, which is used to power all of the blocks in the system.	According to the camera datasheet, it supports an input voltage of 3.6V - 5V.

4.6.5 Block Testing Process

1. Wire the camera and clock modules to the same I2C data bus and connect them to the Arduino microcontroller's serial pins.
2. Receive the current time from the clock by addressing it using the I2C Arduino functions, verifying the properties of clck_ld_lght_cd_comm and cmr_cd_ld_lght_cd_comm.
3. Verify that the camera is powered using at least 4V from the power supply as stated in the cmr_cd_ld_lght_cd_comm interface.
4. Demonstrate situations for when the camera signals the LEDs on and off, verifying that 1 and 0 respectively are sent to the Arduino, as expressed in the cmr_cd_ld_lght_cd_comm interface.
5. Connect the output PWM pin of the Arduino to an oscilloscope, verifying the properties of ld_lght_cd_ld_lights_dsig are met.

4.6.6 References and File Links

4.1.6.1 References

[1] "I2C Pull Up Resistors," Rheingold Heavy, Jan. 16, 2015. https://rheingoldheavy.com/i2c-pull-resistors (accessed Feb. 18, 2022).

4.1.6.2 File Links

Clock datasheet
I2C Protocol
Camera datasheet
MicroPython documentation
Arduino PWM Signals
Wire Arduino Library
fastLED Documentation
WS2812B LED Datasheet

4.6.7 Revision Table

2/18/2022	Sawyer Brundage: Updated cmr_cd_ld_lght_cd_comm interface, flowchart, general validation, and testing to incorporate I2C protocol
2/4/2022	Sawyer Brundage: Initial Document Creation

4.7 Microcontroller Validation

4.7.1 Block Overview

The microcontroller that will be implemented in the bird deterrent system is the Arduino Nano. The purpose of this block is to hold all the code that will be used within components and I/O pins. The microcontroller will also distribute power between all components. The primary reason for opting for the Nano is that it has a smaller footprint, making it easier to layout onto a PCB. Another reason for opting for the Nano is that it includes all the pins that our system will use such as the serial communication and I2C which utilizes SDA and SCL pins [1]. The Arduino Nano is also widely accessible and all members of the team have experience working with the Nano. The Arduino Nano with its petite size and having the needed pins for the system makes it a perfect microcontroller for the bird deterrent system.

4.7.2 Block Design

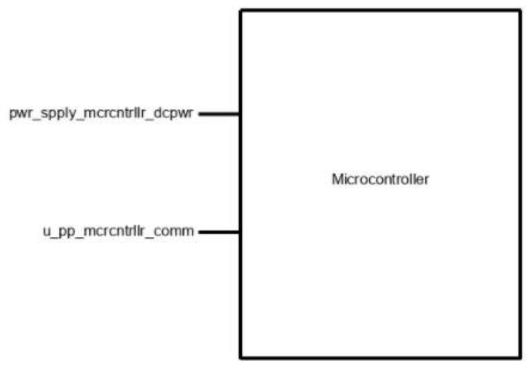


Figure 22: Black Box Diagram

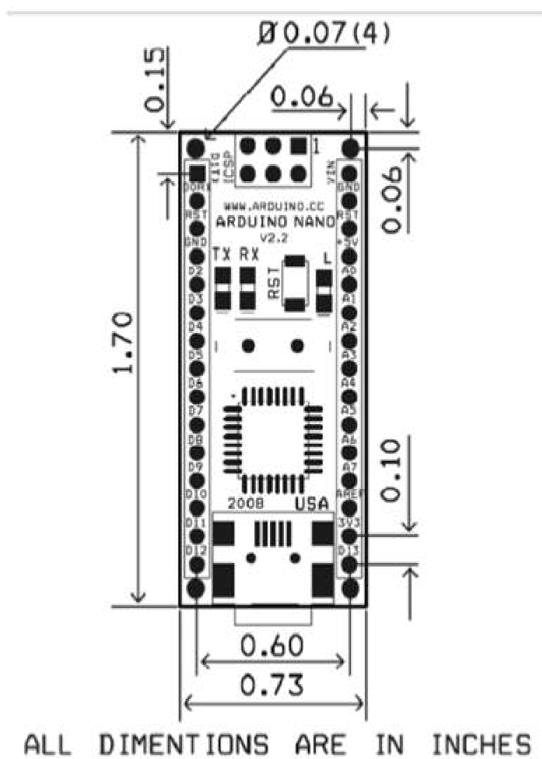


Figure 23: Mechanical Drawing of Microcontroller

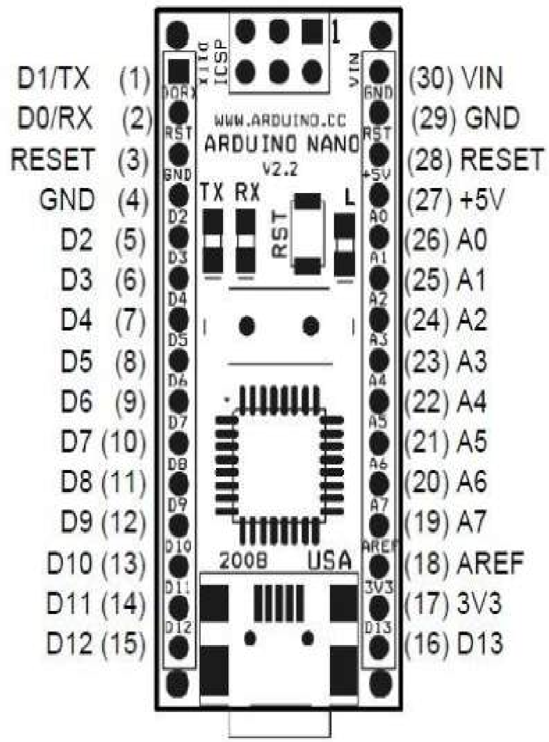


Figure 24: Layout of I/O Connections of Microcontroller

4.7.3 Block General Validation

Our group has decided to use the Arduino Nano for our microcontroller. The bird deterrent system requires the use of a serial communication which will be used for the Bluetooth module, I2C communication which will combine the clock and camera module onto one bus where they will be able to alternate sending and receiving data.

An alternative approach was to use the ARD-MEGA2560PRO microcontroller. The reason for this is that it has 4 serial ports and our system will use 3 serial connections to communicate between the devices. This microcontroller would allow our group to establish these communications to ultimately get our system to send and receive data. However, ARD-MEGA2560PRO was ultimately not chosen because combining the camera and clock module onto an I2C bus allowed for fewer serial communication pins to be used [2]. Resulting in deciding to use the Arduino Nano.

The Arduino Nano includes all the pins needed by all the other blocks of the bird deterrent system. With the Arduino Nano, the bird deterrent system will be able to send and receive data between modules allowing for the system to ultimately deter approaching birds.

4.7.4 Block Interface Validation

pwr_spply_mcrcntrllr_dcpwr : Input

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Inominal: 5 mA	The current draw from an I2C connection is about 2.5 mA and for the PWM pins of the LED strips which should add up to approximately 5 mA [2].	I2c connection draws approximately 2.5 mA and we need to account for the PWM pins of the LED strips.
Ipeak: 10 mA	When all components are running they will require approximately 5 mA so 10 mA should be the max current going into these pins.	Components only require about 5 mA.
Vmax: 5 V	All modules of the system use approximately 5 volts. Requiring a 5.5 volts will be enough to suffice the nominal [4].	All modules require 5 volts to operate.
Vmin: 4.5 V	All modules of the system use approximately 5 volts. Therefore being a bit less than the nominal will be sufficient [4].	All modules require 5 volts to operate.

u_pp_mcrcntrllr_comm : Input

Interface Property	Why is this interface this value?	Why do you know that your design details for this block above meet or exceed each property?
Datarate: 9600	This is the standard baud rate used between Arduino products [3].	Baud rate of 9600 is common in serial communication.
Other: Data input will be an array of data; the order will be brightness, operation times, and current world time.	This data is what our team thought was the most important for the user.	The connection between the microcontroller and module will allow for the transfer of data between the two. With this data transfer user input will be transferred to the microcontroller
Protocol: Serial Port Communication	Serial port connection is what is needed to communicate at a baud rate of 9600 [3].	Serial port connections use a baud rate of 9600.

4.7.5 Block Testing Process

pwr_spply_mcrcntrllr_dcpwr : Input
1. Connect the power supply to the microcontroller.
2. Run the system for about one minute. Check that system has both nominal current and voltage or at least near the approximate values.

u_pp_mcrcntrllr_comm : Input
1. Create a connection between the microcontroller and another module
2. Have the two systems communicate with each other by connecting the receive and transmit pins.
3. Have the systems send strings or chars to another at a baud rate of 9600.

4.7.6 References and File Links

4.1.6.1 References

[1] T. A. Team, "A guide to arduino & the I2C protocol (Two wire): Arduino documentation," Arduino Documentation | Arduino Documentation. [Online]. Available: https://docs.arduino.cc/learn/communication/wire. [Accessed: 19-Feb-2022].

[2] "CC2650: Current consumption while I2C communication with

sensor," CC2650: Current consumption while I2C communication with sensor - Bluetooth forum - Bluetooth - TI E2E support forums. [Online]. Available: https://e2e.ti.com/support/wireless-connectivity/bluetooth -group/bluetooth/f/bluetooth-forum/ [Accessed: 19-Feb-2022].

[3] "Serial.begin()," Serial.begin() - Arduino Reference. [Online]. Available: https://www.arduino.cc/reference/en/language/functions/communication/serial/begin/. [Accessed: 19-Feb-2022].

[4] "Arduino Nano," Arduino Official Store. [Online]. Available: https://store.arduino.cc/products/arduino-nano. [Accessed: 19-Feb-2022].

4.1.6.2 File Links

1. LED Strip
2. RTC
3. Arduino

4.7.7 Revision Table

02/18/2022	Max Garcia-Matzumilla: updated general validations with recommendations from peer review
02/12/2022	Max Garcia-Matzumilla: fixed description to talk in third person and refer to system. Also changed microcontroller to Arduino Nano. Also updated figure titles.
02/05/2022	Max Garcia-Matzumilla: updated reference and file links to match IEEE
02/04/2022	Max Garcia-Matzumilla: created document and filled out draft

4.8 Enclosure

4.8.1 Block Overview

The enclosure of the bird deterrent system holds all the electronics of the system. It weighs less than 1.0kg which makes it portable. Additionally, all the components needed for the system are mounted to the enclosure and can be accessed by simply removing the top cover by hand with no special tools needed.

4.8.2 Block Design

The design of the enclosure has two circular windows. The hole to the right of the enclosure is 3.5 mm in diameter. The camera sensor utilizes this window to detect the birds movements. Nevertheless, the circular window to the left is for the connectors to go inside and outside the enclosure. It has a radius of 10.0 mm that will be utilized by the connectors of both the power supply and the LEDs. Nevertheless, the dimensions of the enclosure were chosen to be able to contain all the necessary electronics inside. These electronics are, camera, power supply, and the printed circuit board. In addition, the enclosure is made out of plastic. It is a lightweight material which makes the enclosure light enough to move around from one position to another. Lastly, the dimensions were chosen to fill all the components of the system.

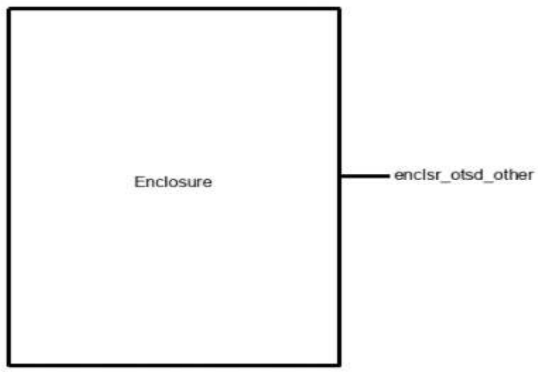


Figure 25: Black Box Diagram

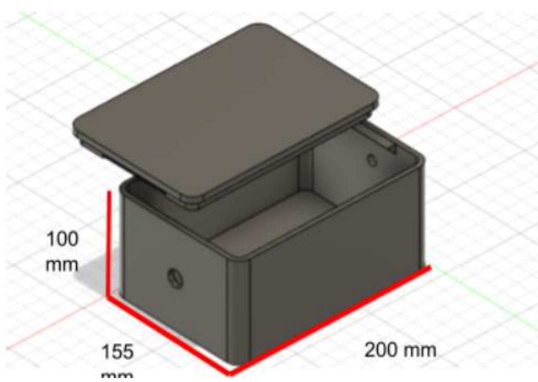


Figure 26: Enclosure Overview

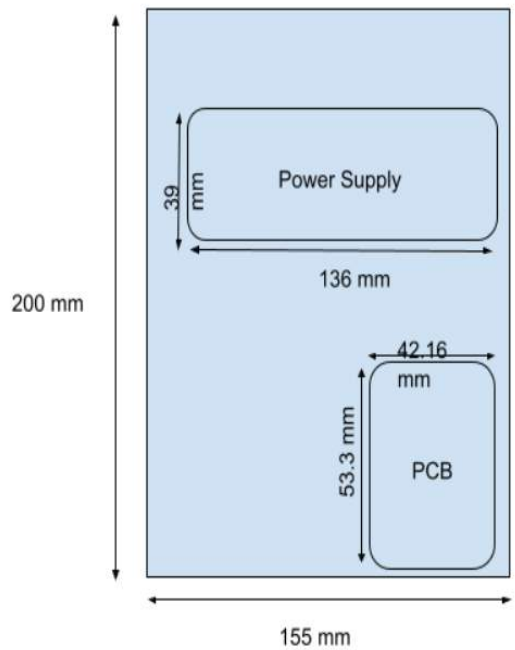


Figure 27: Enclosure Base Dimensions and the Blocks Mounted to it

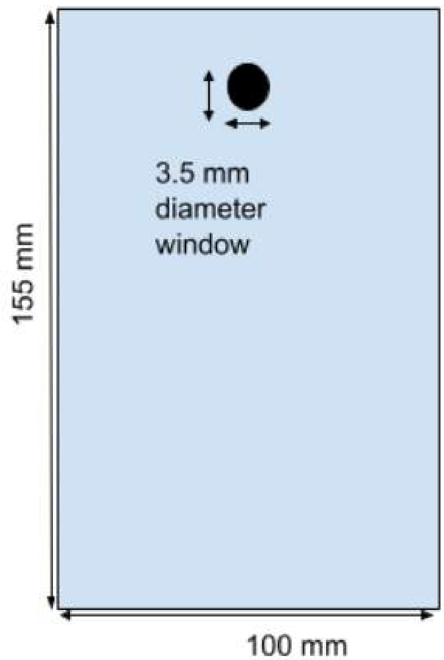


Figure 28: Front face of Enclosure with 3.5mm Diameter Window

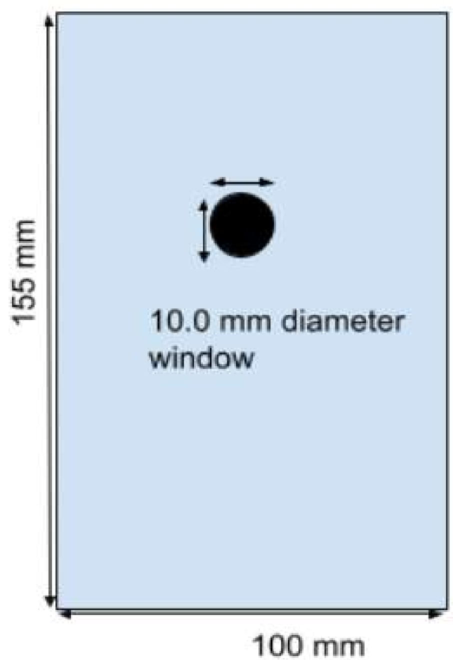


Figure 29: Back Side of Enclosure

4.8.3 Block General Validation

There are two blocks that the enclosure contains(Figure 27). The figure shows that the dimensions of the base of the enclosure were chosen to fit the power supply and the PCB with enough space left. With 200 mm height and 155 mm width, and given the dimensions of the blocks, there is enough space for these blocks to be mounted to the base.

The camera sensor is going to be mounted to the front face of the enclosure. The height of the front face is 155 mm. The width of the same face is 100 mm. More importantly, the front face has a small circular window. The diameter of the window is chosen to best fit the camera sensor. Given that the camera sensor has a diameter of 2.0 mm, the 3.5 mm hole in the front face of the enclosure is enough to fit the lens.

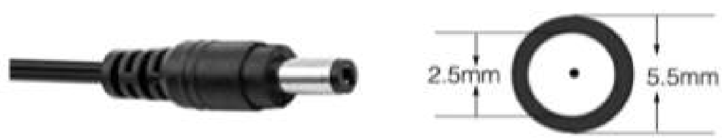


Figure 30: Power Supply Cable Dimensions

Figure 29 shows the design of the back side of the enclosure. There is nothing mounted to this side. However, there is a circular window that is dedicated to the power supply connections and the LEDs connections. Given the dimensions of the power supply wire are less than 10.0 mm in diameter(figure 30), the path into and out of the enclosure should be large enough for the connectors to fit. Moreover, the cable size for the LEDs is less than the 10.0 mm diameter window.

4.8.4 Block Interface Validation

pwr_spply_mcrcntrllr_dcpwr : Input

Interface Property	**Why is this interface this value?**	**Why do you know that your design details for this block above meet or exceed each property?**
Other: The cover of the enclosure can be attached and detached by hand only. No special equipment such as screwdriver needed	This property is chosen to easily access the electronics inside the enclosure.	The dimensions ratio between the case and the cover allows it to be attached and detached only by and.
Other: The Enclosure is portable and can be carried by hand	This property is chosen to move the system from one position to another in a convenient way.	The dimensions of the enclosure makes it easy to carry around. In addition, it is made of plastic which is a light material.
Other: The enclosure weights no more than 1.0kg (Including all the parts mounted to it)	This value was chosen to keep the system light enough to be portable.	The two components mounted to the case PCB and the camera, combined, weigh less than 0.1 kg. The case made of plastic and it weigh less than 0.3 kg.

4.8.5 Block Testing Process

enclsr_otsd_other : Output:

1. Remove the cover by hand and reattach it again by hand
2. Flip the enclosure upside down while keeping it at a distance from the ground and observe the cover. It will stay attached to the case.
3. Carry the enclosure around to show its portability
4. Show that all the components are mounted to the enclosure with all the connectors in place. Place the enclosure on a scale and observe the weight. It is less than 1.0 kg.

Enclosure Requirements Verification Plan:

1. Remove the cover and observe the components mounted to the enclosure
2. Flip the enclosure upside down, while kept at a distance from the ground and observe the components mounted to the case.
3. Shake the case for 5 seconds and observe the components mounted to the case.

4.8.6 References and File Links

4.1.6.1 References

4.1.6.2 File Links

1. Power Supply
2. LED Strip

4.8.7 Revision Table

2/10/2022	Hameed: Document created

5 System Verification

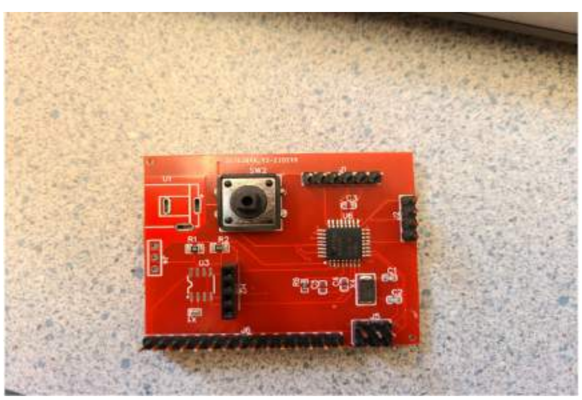
5.1 Universal Constraints

5.1.1 The system may not include a breadboard

A PCB was designed for our system. Therefore a breadboard will not be needed or used. After realizing that our PCB has had issues, we used a proto-board to make certain connection.

5.1.2 The final system must contain both of the following: a student designed PCB and a custom Android/PC/Cloud application

The PCB for our system arrived, but after multiple different workarounds were attempted to get it working, we had to default to just using the PCB for power distribution. We have emails with Ingrid discussing that we should try and get some aspect of the PCB in the system verification if we can manage. We have an application to control the period of operations.



5.1.3 If an enclosure is present, the contents must be ruggedly enclosed/mounted as evaluated by the course instructor

All modules are able to fit inside the enclosure and are mounted to it using Velcro. The camera is also secured to one side of the enclosure using hot glue, situated with its lens facing out of a hole. The LED strip does not rest inside the enclosure; it is secured outside to a window border using outdoor mounting Velcro.

5.1.4 If present, all wire connections to PCBs and going through an enclosure (entering or leaving) must use connectors

The device has two wired connections going through the enclosure to the PCB: the LED strip and the power supply. The LED strip has a male connector that slots into a female connector connected to the PCB, allowing the strip to connect and disconnect without any further soldering. Furthermore, the DC power supply uses a C13/c14 connector to the power supply and PCB.

5.1.5 All power supplies in the system must be at least 65% efficient.

According to the datasheet for the Mean Well LRS-50-5 power supply, it has a efficiency of 83%; that can be seen on page 2 of this specifications document [1].The testing of this efficiency was done at the rate voltage(5V) and rated current(10A), and we are operating under these rated voltages.

5.1.6 The system may be no more than 50% built from purchased 'modules.'

The number of modules within the system is 10. The purchased modules are the power supply, the bluetooth module, LED lights, the RTC module, and the micro-controller. The modules that our team designed and built ourselves are: LED light code, Camera + Code, Enclosure, UI application, and the PCB power distribution.

5.2 Wall Power Requirement

5.2.1 Requirement

System will draw from a standard 60Hz 120V AC outlet.

5.2.2 Testing Process

1. Check for nominal 5V at the PCB connection.
2. The main source is a power supply connected to a wall outlet.

5.2.3 Testing Evidence

1. Wall Power testing video

5.3 Portability Requirement

5.3.1 Requirement

The enclosure (containing the Arduino uno, camera, power supply, protoboard, Bluetooth module, and RTC module) will weigh under one kg and have dimensions of 20 cm X 15 cm X 10 cm or less.

5.3.2 Testing Process

1. Zero out scale
2. Place system on scale to verify that it weights under a kg
3. Use a ruler to measure each side of the system to verify the given dimensions.

5.3.3 Testing Evidence

1. Portability testing video

5.4 Durability Requirement

5.4.1 Requirement

The enclosure(containing the Arduino uno, camera, power supply, protoboard, Bluetooth module, and RTC module) will withstand forces of at least 15 lbs on each face.

5.4.2 Testing Process

1. Place the enclosure on a scale
2. Zero out the scale
3. Apply force of at least 15 lbs on each side.

5.4.3 Testing Evidence

1. Durability testing video

5.5 Proof of Concept Requirement

5.5.1 Requirement

The team will conduct an experiment whether strobing lights will deter birds.

5.5.2 Testing Process

1. Set up a bird feeder.
2. Take a control for how many birds come to the feeder without the strobing lights.

3. Take an experiment for how many birds come to the feeder with the strobing light.
4. Provide the data and experiment write-up at the end.

5.5.3 Testing Evidence

Experiment Write-up

5.6 Bird Detection Requirement

5.6.1 Requirement

The system will detect birds in camera view with 7/10 samples being correct.

5.6.2 Testing Process

1. Set up system
2. Show picture of bird to camera lens
3. Testing proves that 70% of trials when shown a bird.

5.6.3 Testing Evidence

1. Bird Detection testing video

5.6.4 Requirement

The enclosure (containing the PCB, camera, and power supply) must allow no more than 1 ounce of water to penetrate the interior when sprayed with a water gun containing 1 cup of water. The LED strip, when sprayed with the same amount, must continue working afterwards.

5.6.5 Testing Process

1. Set up the system.
2. Spray enclosure with a water gun containing 1 cup of water.
3. Inspect Enclosure for water. It should not exceed 1 ounce.
4. Spray LED strip with a water gun containing 1 cup of water.
5. Test system to ensure all LEDs still work.

5.6.6 Testing Evidence

Weather Resistance Video

5.7 Survey Requirement

5.7.1 Requirement

If asked whether they would mount the device to their window, at least 7/10 people would reply with ”yes”.

5.7.2 Testing Process

1. Set up an anonymous survey online with picture of mounted system and question.
2. Put link to survey on engineering discord server.
3. Wait a week for responses.
4. Verify that at least 70% of respondents answered yes.

5.7.3 Testing Evidence

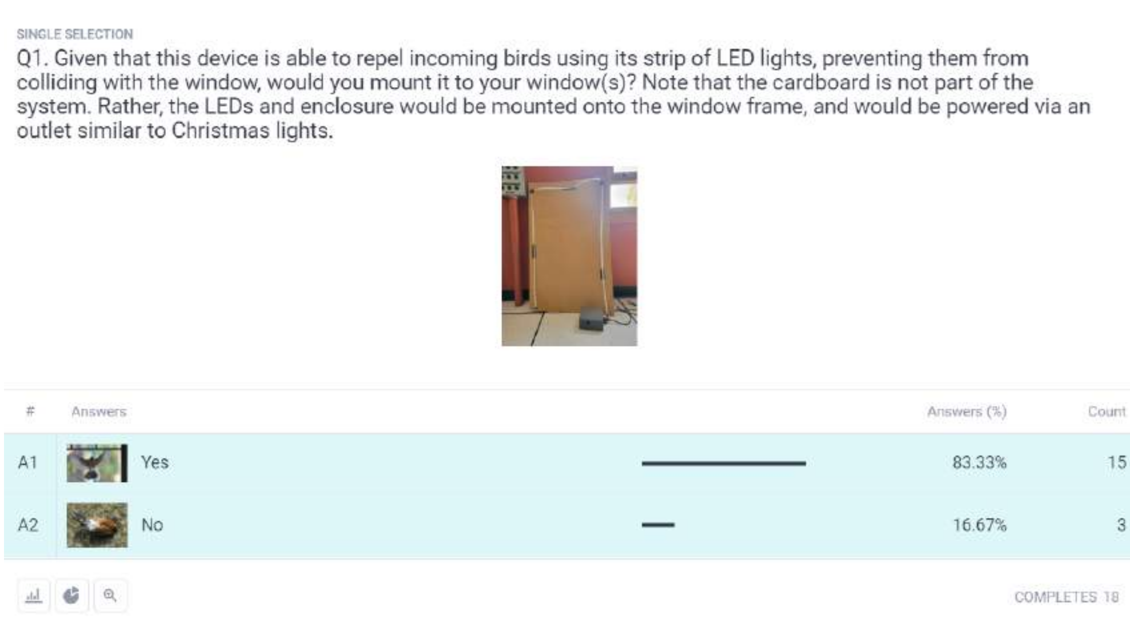


Figure 31: Results of Survey

5.8 UI Requirement

5.8.1 Requirement

Internal clock will be configured through the app.

5.8.2 Testing Process

Testing shows the LEDs functioning in the defined time interval and non-operational outside the same interval.

1. Set up the system.
2. Show the LEDs are operational in the current time window.

3. Change the operational time window to outside the current time window.
4. Show the LEDs are no longer operational.

5.8.3 Testing Evidence

1. User Interface video

5.9 References and File Links

5.9.1 References

[1] "50W Single Output Switching Power Supply," Mean Well, New Taipei City, Taiwan, LRS-50-SPEC, 2022.

5.9.2 File Links

1. Power Supply Datasheet

5.10 Revision Table

4/22/2022	Everyone: Changed the wording for some of the requirements
3/6/2022	Sawyer Brundage: Final fixes and formatting for requirement subsections
3/4/2022	Abdullhmeed Alkawi: Fixed references and file links
3/2/2022	Max Garcia-Maxzumila: Added images and figures for universal constraints and requirement section
3/2/2022	Sawyer Brundage: Added content for universal constraints
3/1/2022	Marcus Plumley: Initial Document Creation

6 Project Closing

6.1 Future Recommendations

6.1.1 Technical Recommendations

1. Improve the PCB

Our group made a PCB for the system, but it was full of mistakes because it was the first PCB for most of the team members. The improvements that can be made are fixing some connections that were problems: the reset pin needed to be connected to the ISP programming port, and the SDA and SCL lines should be wired to the primary I2C line (A4 and A5 on most Arduino pinouts) that are save with the other project files.

2. Improve Bird Data experiment and start fall term

The team conducted experiments to determine the efficacy of strobing bird deterrents. Our mistake was starting this experiment in spring term which did not leave enough time for an effective experiment. Therefore, we advise the next team that tries this project to start much earlier in the project development. In addition to this, most of our recording and activation of the strobe were done by hand. If the team could use some sort of motion detection and a long capture time camera, then the amount of data will increase and the reliability of the data will increase.

3. Record video through the camera

The system incorporates a Open-MV machine-learning camera which was utilized mostly as motion detection and bird detection (with a machine learning). However, the team never took advantage of this camera to record system testing. We recommend implementing some video recording for system testing either with the system camera or an additional camera to understand the efficacy of the system in field testing.

4. Incorporating a speaker or sound deterrent

One way of improving the system is to incorporate a speaker that makes sounds once the camera of the system detects birds. This can add to the efficiency of deterring birds. More importantly, having two methods of deterring birds will add to the possibility of success of the system.

6.1.2 Global Impact Recommendations

1. An impact that was very important to keep in mind for the system is the potential cause to seizures. In our development, we decided on a low frequency strobe to minimize the chance for the device to cause seizures. It never became a problem for our device, but the global impact should be kept in mind for future project development as well.

2. In addition, the migratory effect on birds should also be monitored as future project development occurs. Upon implementation, the team did not notice any observable effect to the number of birds that we were seeing, but migratory patterns should kept in mind to ensure that the device does not effect them negatively. A negative effect on migratory patterns would result in a cultural impact on bird watching.

6.1.3 Teamwork Recommendations

1. Respecting the opinion of others, regardless of whether they are opposed to your own, is fundamental part of an effective communication strategy. A

workplace that values respect encourages its team members to express their ideas, listen to one another, and be cooperative when working together.

2. Openness and transparency play valuable roles in the importance of the communication in teamwork, as they facilitate honest conversation and trust-building. When team members can openly discuss issues, ask for help or clarity, and trust each other, they will feel empowered in their roles and as members of the team.

6.2 Project Artifact Summary with Links

We have gathered the project artifacts and put them in a shared drive that is linked below. The plan is to move them to a google drive that is controlled by the project partner for the next team.

Project Artifacts folder

6.3 Presentation Materials

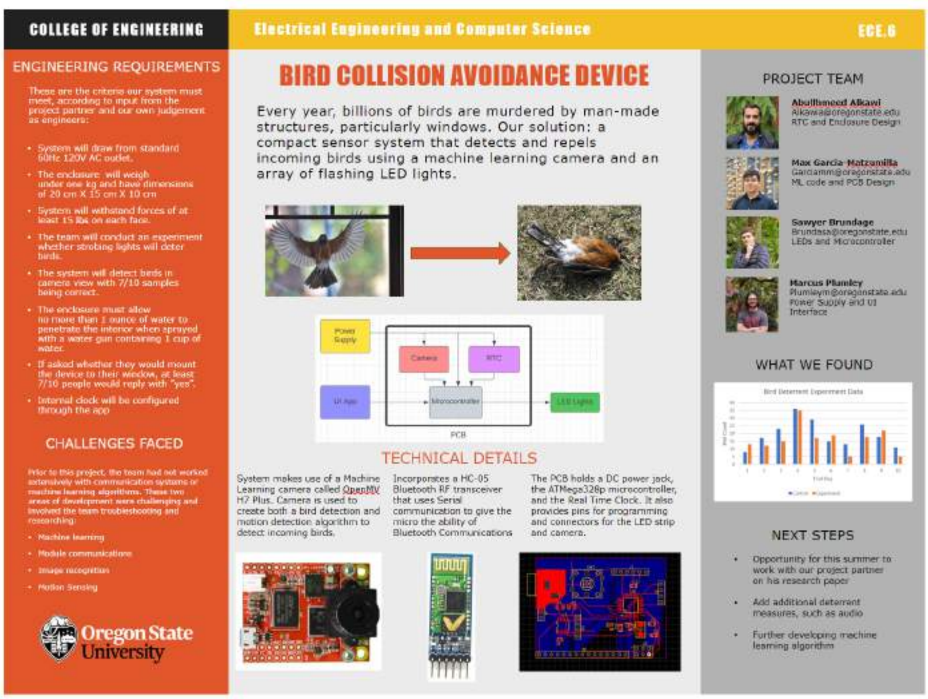


Figure 32: Project Poster for the Engineering Expo