Bird Initiated Rubbish Disposal System

Final Presentation

Team Sponsor: Justin Phelps

Group 12

Aiden Nipper
Will DiSalvo
Rodrigo Guerra
Donovan Reynolds

MEET THE TEAM

Aiden Nipper

Photonic Science and Engineering

Will DiSalvo

Photonic Science and Engineering

Rodrigo Guerra

Electrical Engineering

Donovan Reynolds

Computer Engineering

Problem & Motivation

- **Littering is still a problem**
  - Current solutions to clean up public spaces involve extensive energy costs
    - Volunteer cleanup, convict labor, salaried street cleaners
- The best solutions are just out of reach
  - Access to and education on proper waste disposal
  - Discouraging single use items

"While humanity struggles to learn basic cleanliness, it falls upon birds to tidy up the mess"

– *ChatGPT 3.5*

- **Birds are:**
  - **Omnipresent**
  - **Intelligent enough to form correlations**
  - **Cheap to bribe**

Core Goals

→ **Promptly rewarding Birds that bring Cigarettes**

- Quick Cigarette recognition, and quick food distribution
- Fast enough for Birds to associate their behavior with the rewards

→ **Consistent Bird and Cigarette detection verdicts**

- Accurate Bird and Cigarette Detection
- Consistent Food delivery

→ **The BIRDS housing should be robust, yet easily maintained**

- Should survive in outdoor environments (advanced)
- Trash container should be easily inserted/removed
- Food should be easily refilled
- Platforms should be easily disassembled

Objectives

→ **Promptly rewarding Birds that bring Cigarettes**
- Quick Cigarette recognition, and quick food distribution
- Fast enough for Birds to associate their behavior with the rewards

→ **Consistent Bird and Cigarette detection verdicts**
- Accurate Bird and Cigarette Detection
- Consistent Food delivery

→ **The BIRDS housing should be robust, yet easily maintained**
- Should survive in outdoor environments (advanced goal)
- Trash container should be easily inserted/removed
- Food should be easily refilled
- Platforms should be easily disassembled

Engineering Specifications

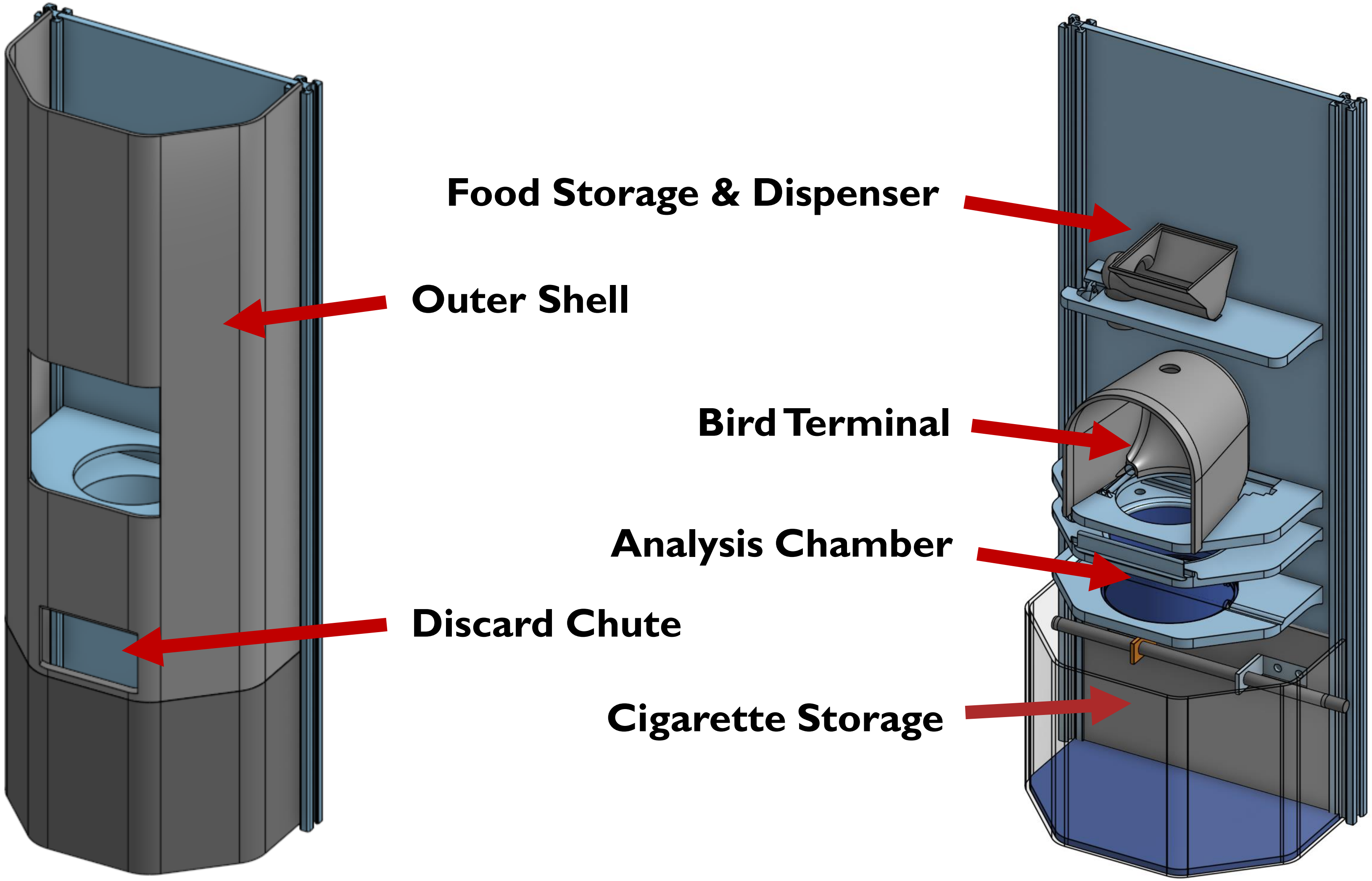
	Component	Parameter	Specification
Electrical	Power Supply	Support Max Delivery	Up to 10 Watts
	PCB	Power Draw	Less than 5 Watts
	Motors	Reliability	Operate without fail 80% of the time
	Image Processor	Response Time	Deliver verdict within 5 seconds of taking image

	Component	Parameter	Specification
Optical	Collimating Lens (Trash)	FOV	30°
	Collimating Lens (Bird)	HFOV	90°
	Focusing Lenses	Focal Length	10 – 20 cm
	CMOS Sensors	Resolution	640 x 480 pixels
	Red Laser Diode	Spot Size	6 mm (@ 5 m distance)
	Photodiode	Wavelength Range	400 – 1100 nm
	Mirror	Wavelength Range	400 – 700 nm

Engineering Requirements

System/Feature	Requirement
Accuracy of Bird Image Recognition	≥ 70%
Accuracy of Trash Image Recognition	≥ 80%
Accuracy of Trash Motion Sensor	≥ 90%
Time duration of Opening Doors	≤ 3 sec
Time duration of Trash verdict	≤ 3 sec
Time duration from Trash approval to Food Delivery	≤ 2 sec

The BIRDS House



Food Storage & Dispenser

Outer Shell

Bird Terminal

Analysis Chamber

Discard Chute

Cigarette Storage

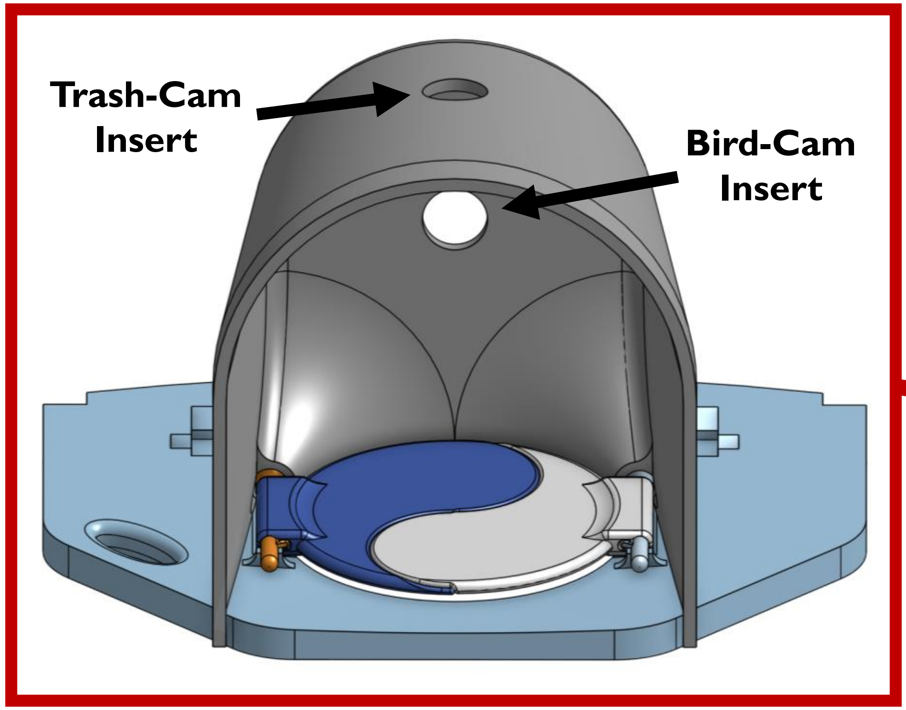
(With Outer Shell)

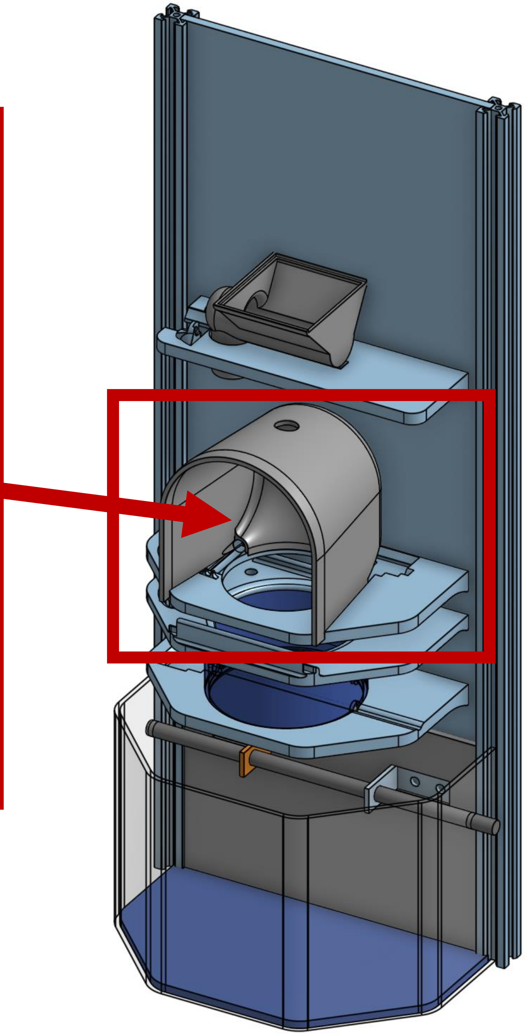
(Without Outer Shell)

Bird Terminal

Main Features:

- **Landing Platform** – landing pad for birds interacting with the system
- **Bird-Cam** – pointed outward, for capturing images of birds
- **Trash Intake** – hole with Servo-controlled doors that open upon bird being recognized
- **Trash-Cam** – pointed downward at analysis, for capturing images of trash (cigarettes)
- **Food Delivery** – tray for food to be dispensed from tube running from Food Dispenser

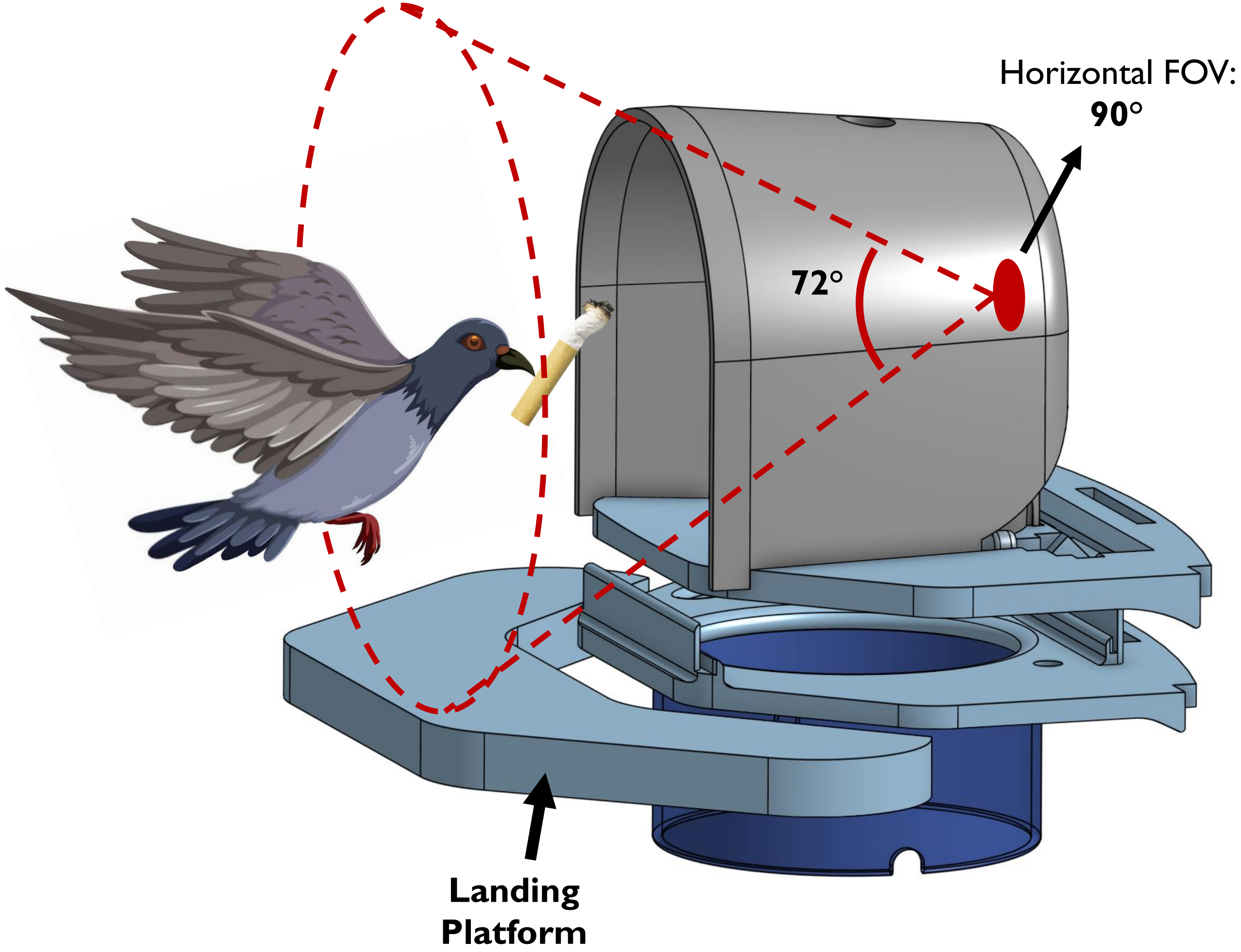




- Outer shell will be transparent and removable for easy maintenance
- System plates/platforms will be secured through slots in backplate

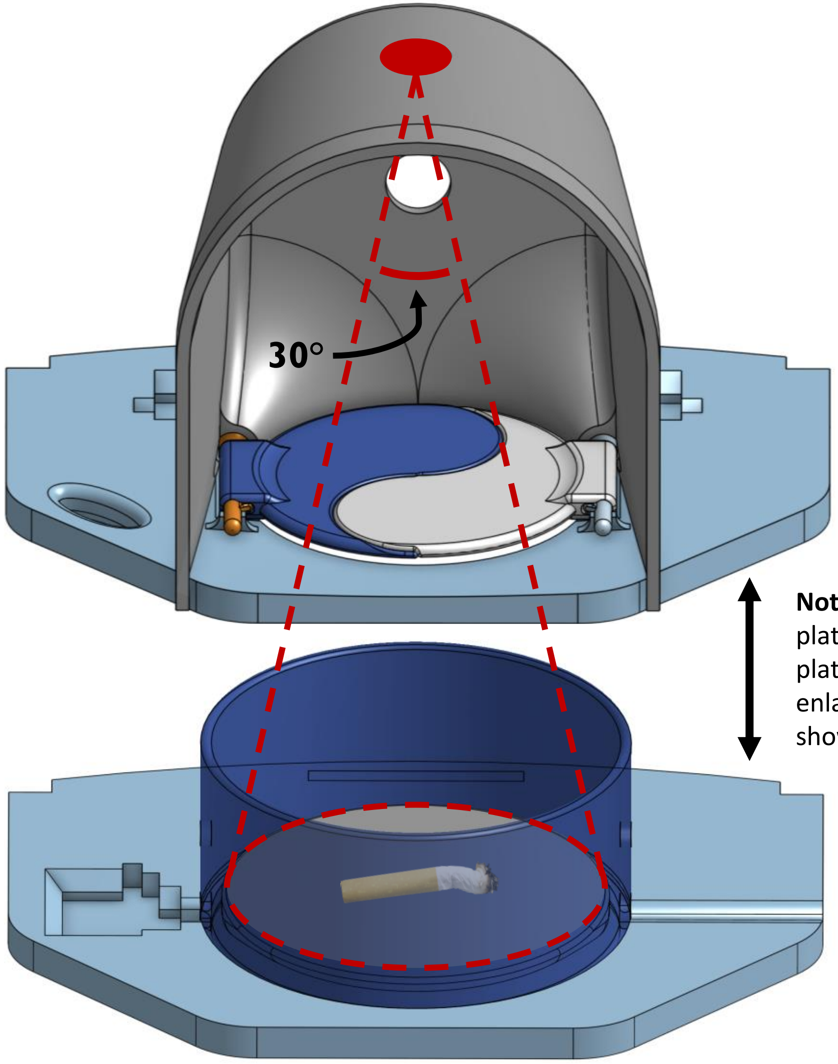
Bird-Cam FOV

(side view)



Trash-Cam FOV

(front view)



Note: motion sensor platform removed and platform separation enlarged to better show Trash-Cam FOV

Lens System Hardware Selections

Sensor Selection

Specification	CMOS	CCD
Readout Noise	5 e^-	3 e^-
Power Consumption	2.5 V	12 V
Readout Time	10 ms	500 ms
Price	$50	$200

- More noise created by the CMOS sensor
- Readout time and power consumption are very low

Lens Selections

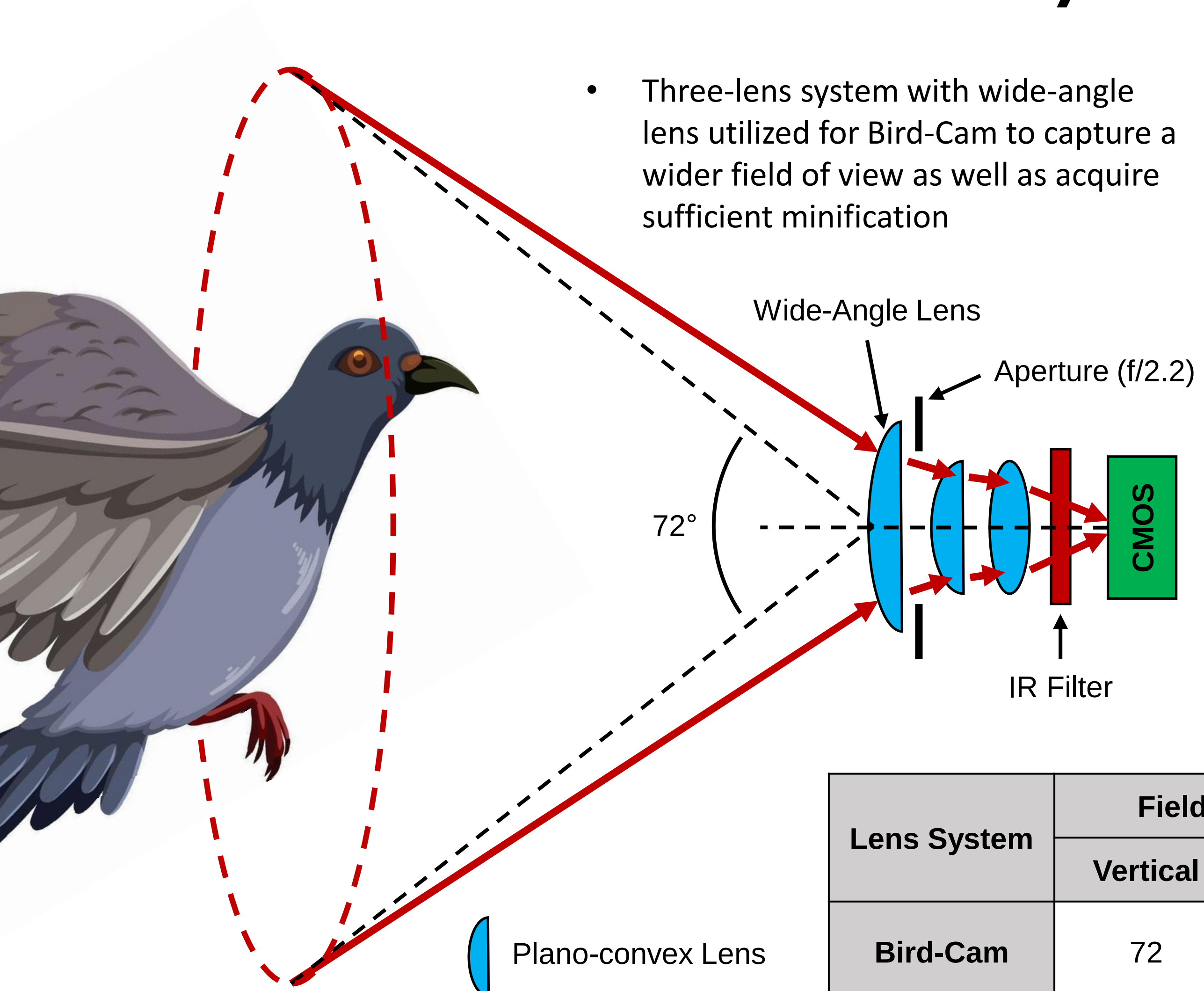
Lens Shape	Ideal Conjugate Ratio
Biconvex	< 5:1
Plano-convex	All
Plano-concave	Infinite, larger finite (> 5:1)
Biconcave	< 5:1
Meniscus	Varies; dependent upon curvature and polarity

- Objects will be far away from front of lens system, images will be formed close to the back of lens system
  - Leads to a large or infinite ideal conjugate ratio
- If a second lens is used, system will have a much smaller ideal conjugate ratio

Lens System Design

- Three-lens system with wide-angle lens utilized for Bird-Cam to capture a wider field of view as well as acquire sufficient minification

- Two-lens system utilized for Trash-Cam to acquire sufficient minification onto the CMOS sensor
- Infrared (IR) filter used to block out wavelengths in the IR range to prevent image distortion



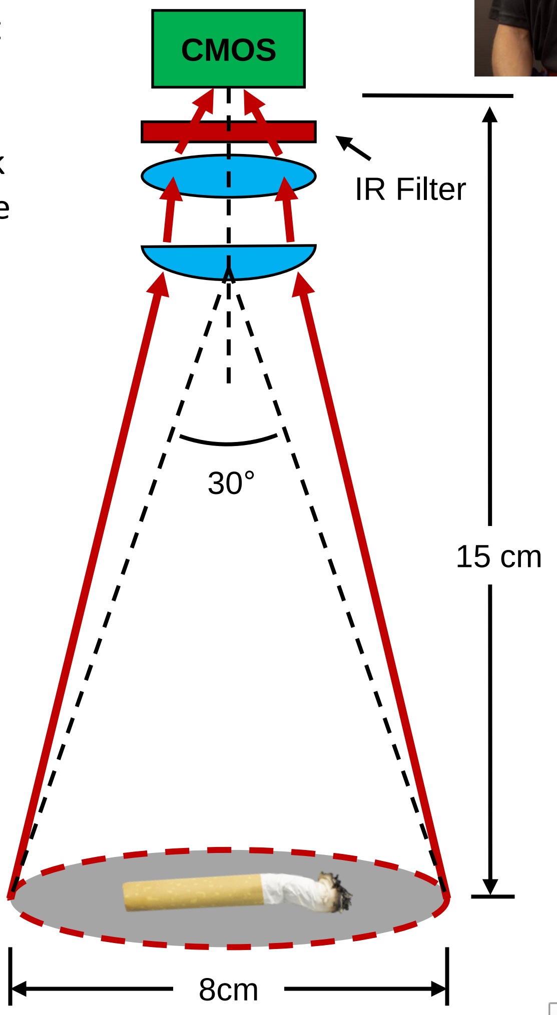
Bird-Cam

Plano-convex Lens

Biconvex Lens

Lens System	Field of View (°)		Focal Length (cm)
	Vertical	Horizontal	
Bird-Cam	72	90	10 – 20
Trash-Cam	30	30	15

Trash-Cam



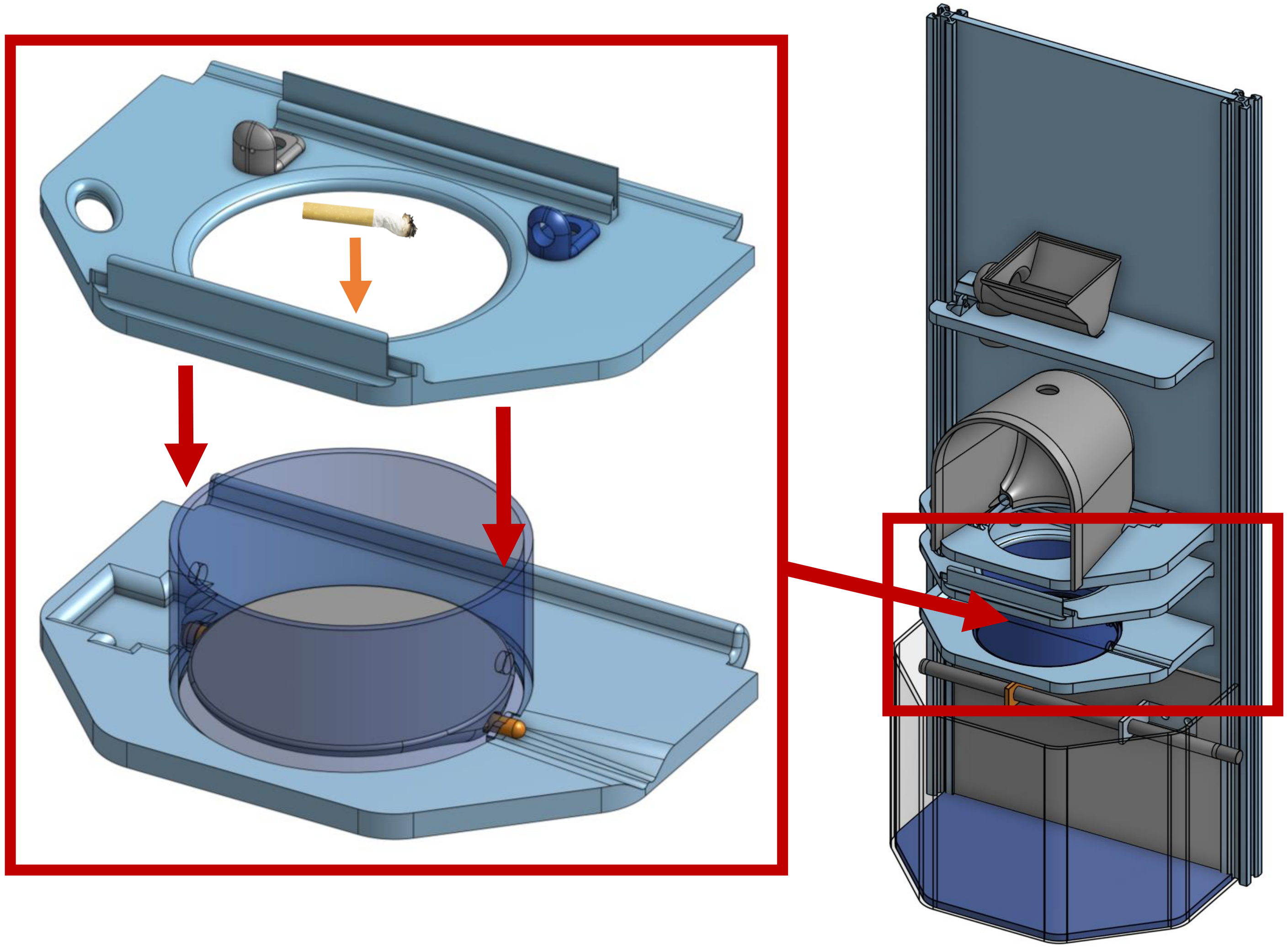
Analysis Chamber

Laser Motion Sensor Platform

- Laser break plane that gets obstructed when items pass through (orange arrow)
- Activates Trash-Cam when tripped, so image recognition can take place

Analysis Chamber Platform

- Items that fall are analyzed by Trash-Cam
- Servo-controlled pivoting door sorts between trash (cigarettes) and non-trash after analysis
- Cigarettes sent to cigarette storage, other items sent to discard chute



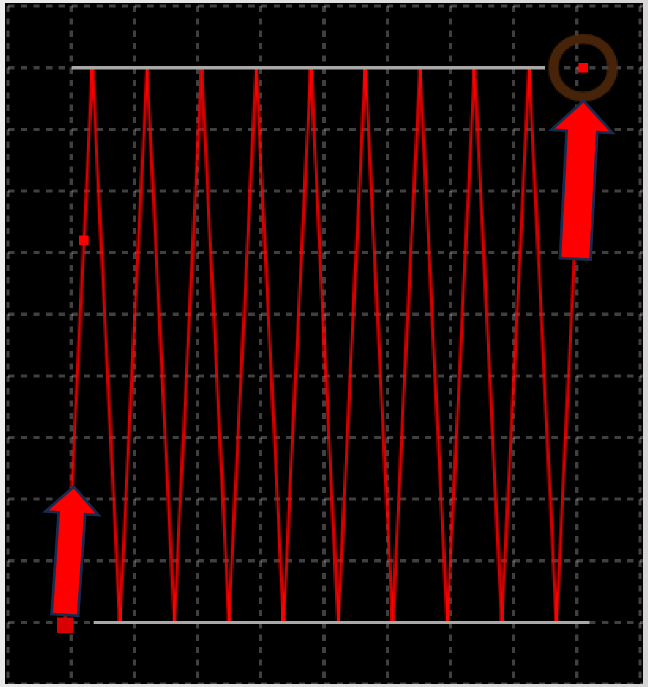
Laser Motion Sensor Selection

Photodetector Specifications

Specification	Solar Cell	Photodiode	Photoresistor
Data Rate	34.2 Mbps	2 Gbps	N/A
Rise Time	N/A	1 µs	10 ms
Spectral Bandwidth	400 – 700 nm	500 – 1000 nm	400 – 900 nm
Cost	$15.50	$0.40	$0.20

Photodiode has fastest rise time and a relatively low cost

Laser Grid



Single laser diode will send laser beam through mirror system into photodiode

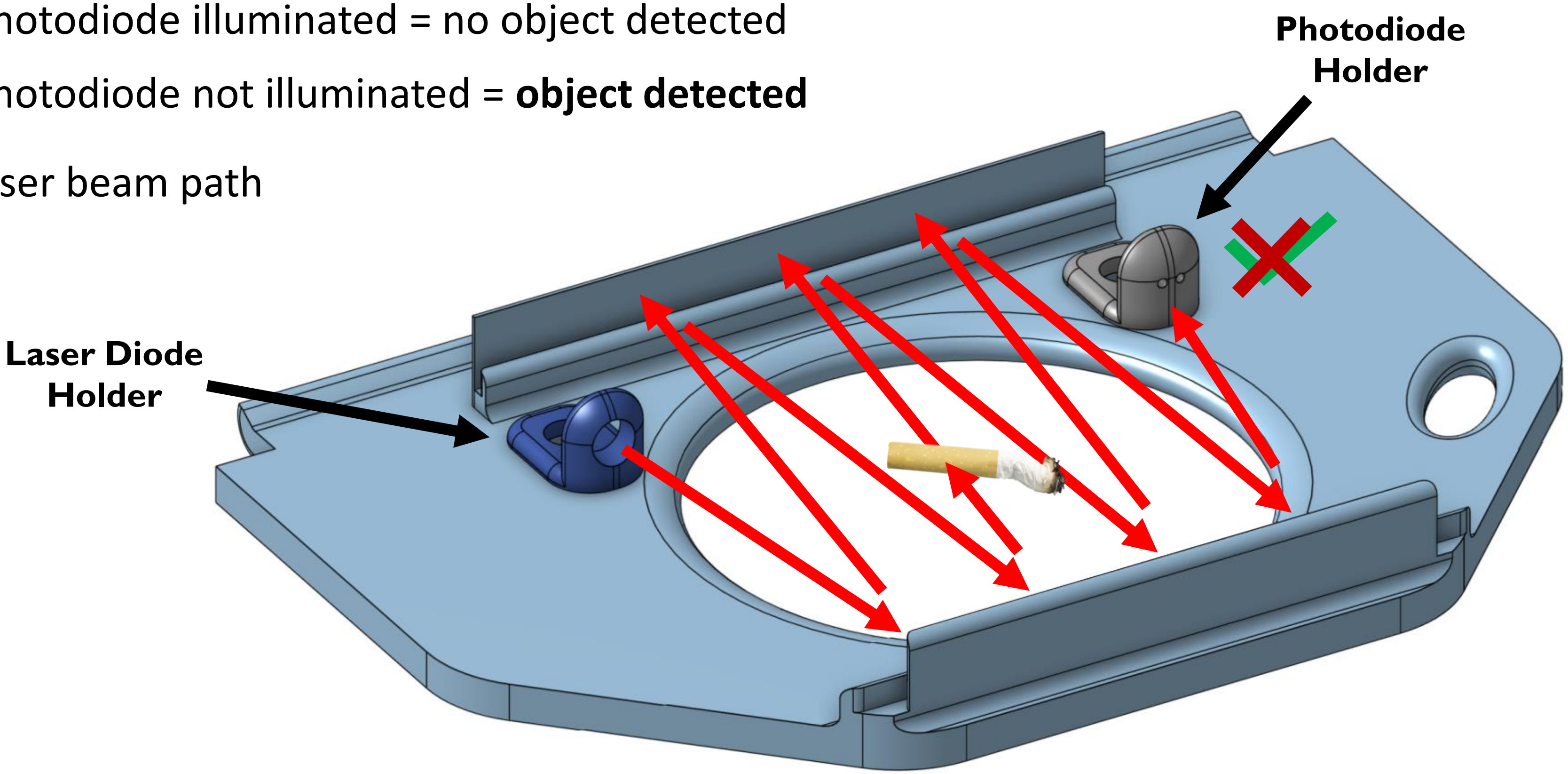
Laser Motion Sensor Design

Sensor Integration:

✓ = photodiode illuminated = no object detected

✗ = photodiode not illuminated = **object detected**

→ = laser beam path



Food Storage & Dispenser

Food Storage

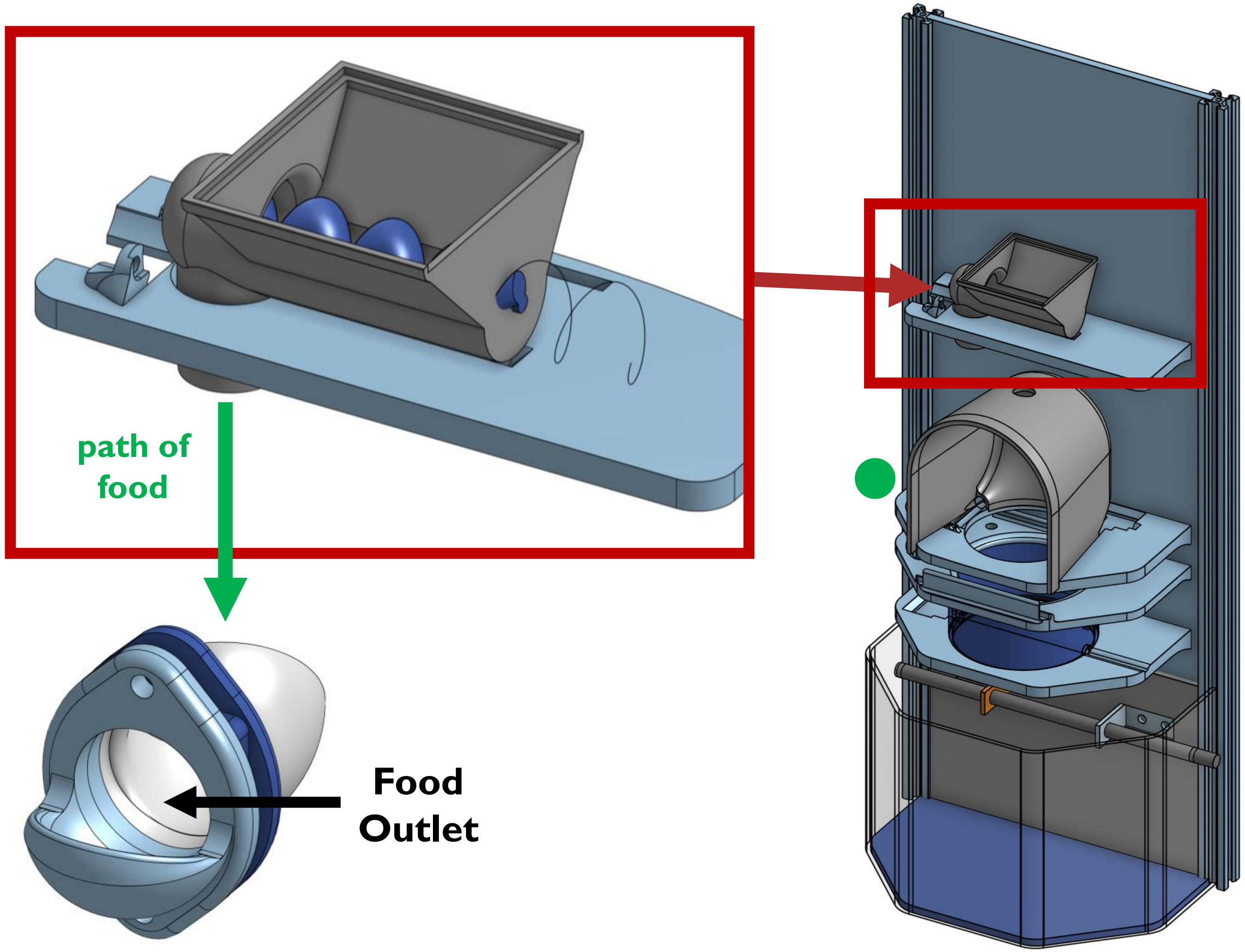
- Container will hold birdfeed, be transparent so birds can see food
- Storage funnels into screw conveyer dispenser

Food Dispenser

- Once activated by trash recognition, stepper-controlled screw spins and pushes birdfeed out through dispenser hole
- Food will fall through tube to food outlet for rewarded birds to eat

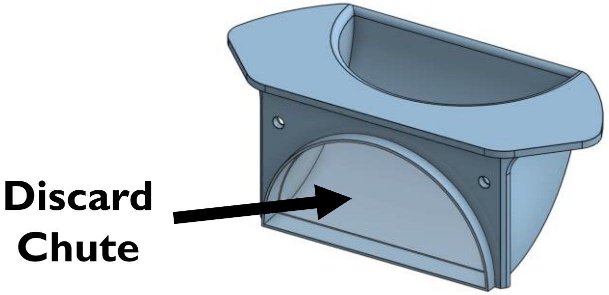
Food Outlet

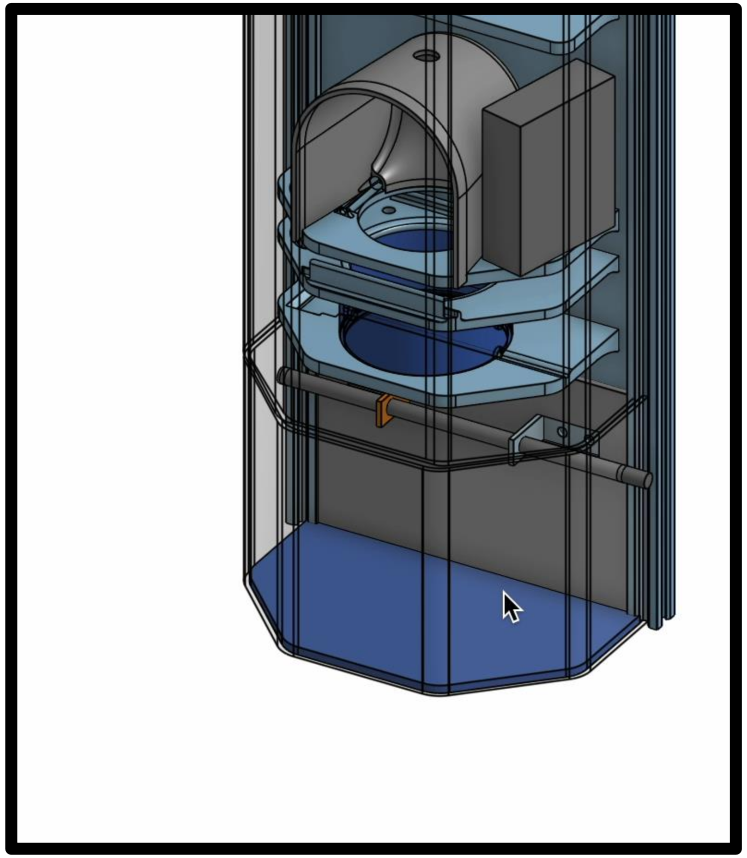
- Attached to outer shell, left of Bird Terminal opening (location shown by green dot in diagram on the right)
- Accessible from bird landing platform

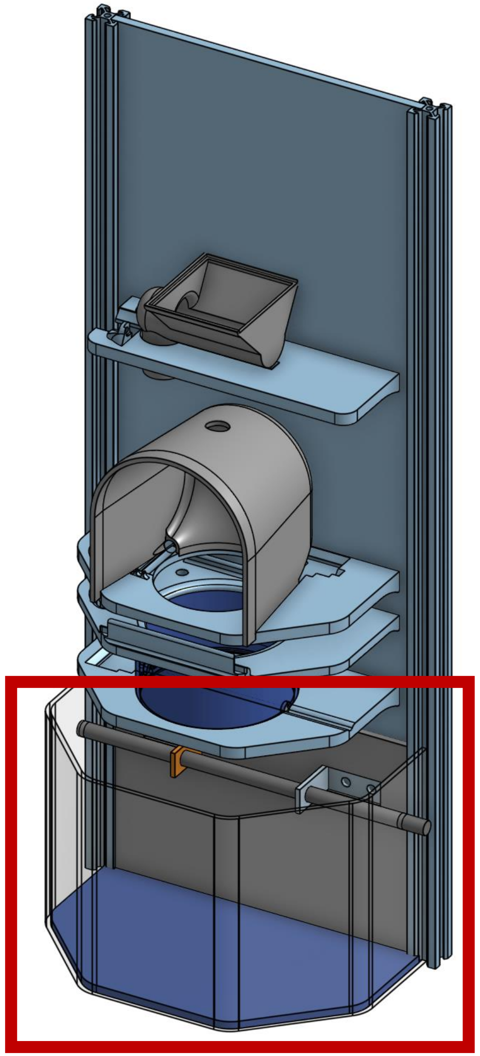


Cigarette Storage

- Cigarette storage container will also be transparent, and will be removable for emptying contents
- Container is emptied by first removing the securing rod from the back, then simply pulling out the container and properly disposing the contents
- Unwanted items will be discarded out the housing via the discard chute, which will be attached to the front of the container







Hardware Subsystems

Motion Detectors

- **Trash Detector**
  - Uses a Laser and Photodiode to detect when an object breaks the laser path as it passes through

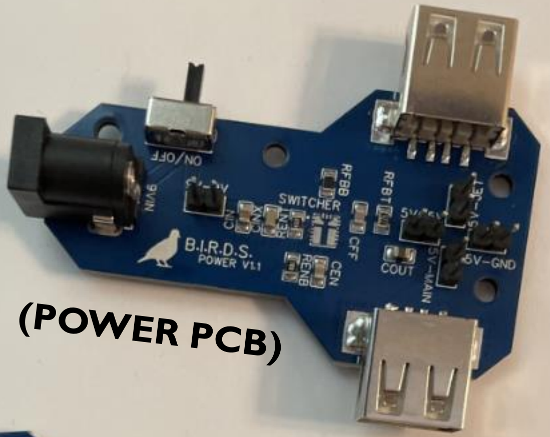
Actuators

- **Food Dispenser**
  - Uses a 5 Vdc Stepper Motor for its high torque and precise control
  - Will rotate a screw conveyer to dispense bird feed down to the food receptacle
- **Door Actuators**
  - Uses two 9-gram Servo Motors to operate two semicircular doors that open to the Analysis Chamber
- **Trash Sorter**
  - Uses one 9-gram Servo Motor to rotate the analysis plate towards Trash Storage or the Discard Chute

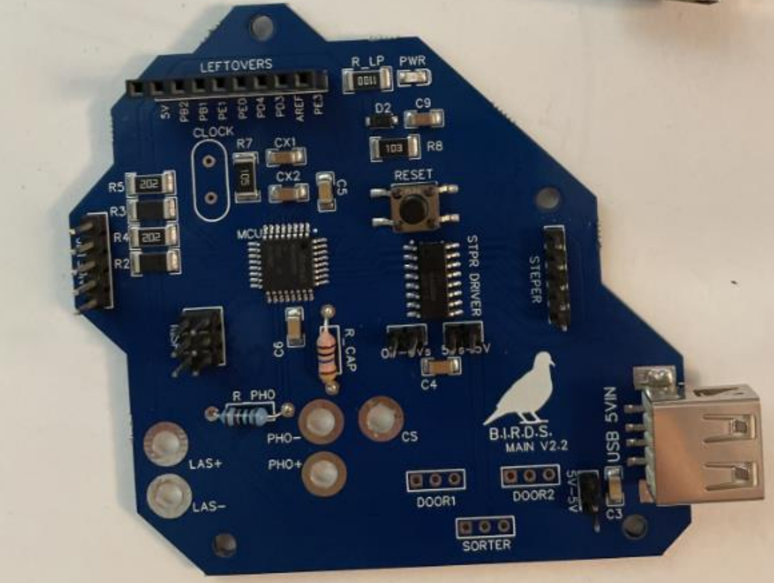
Jetson Communication

- Simple 4-pin connection to the Jetson to activate and receive verdicts of the image processing cameras
  - 2-pins for “Activate Bird Camera” and “Is Bird?”
  - 2-pins for “Activate Trash Camera” and “Is valid item?”

PCB Design



(POWER PCB)



(MAIN PCB)

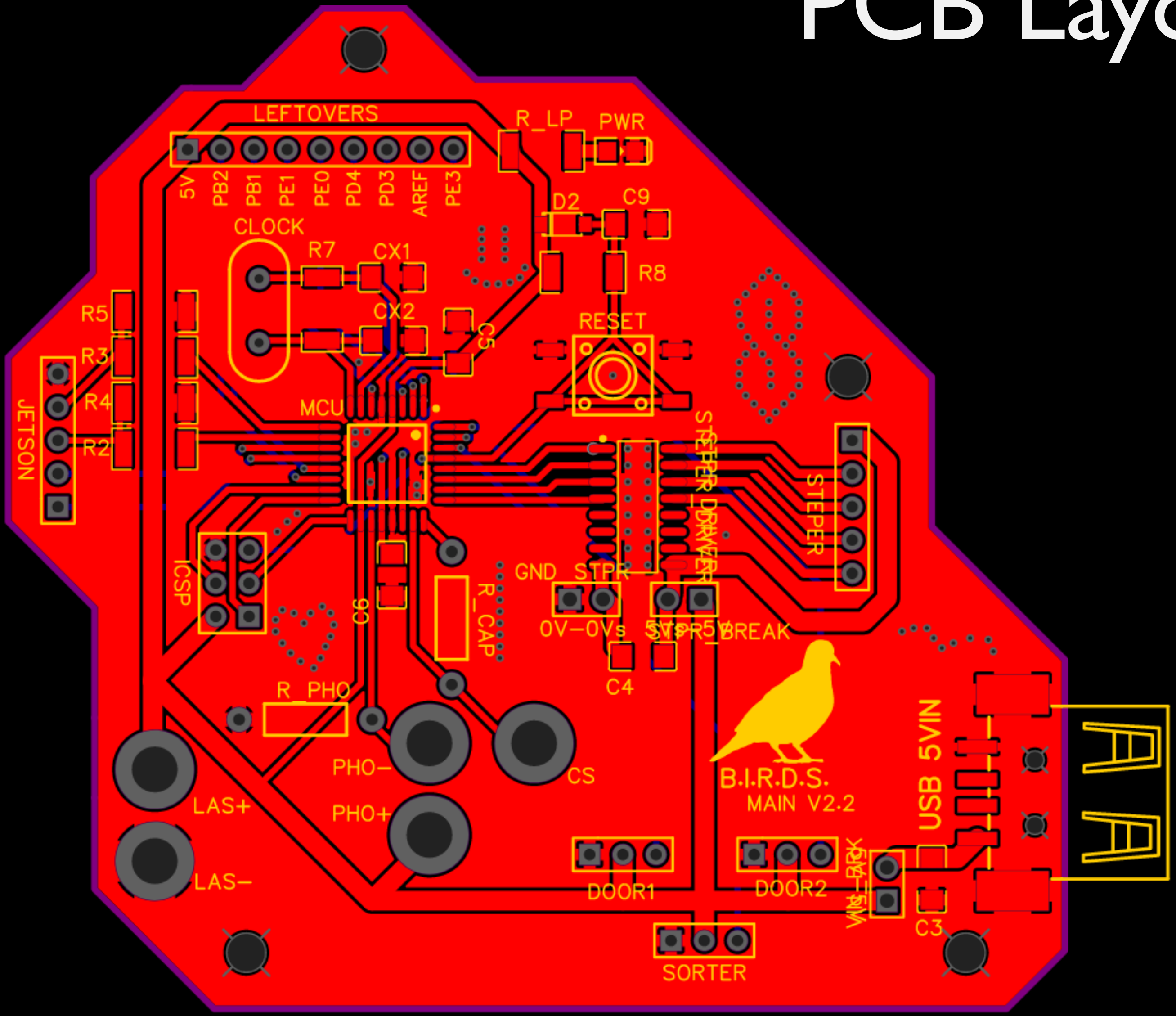
Power Board

- Since we are powering two computing boards (Jetson Nano & Main), one board powers both
- Easier to debug and repair if something goes wrong
- Has multiple Jumper-Pin breakers for testing and control
- Takes in 9 Vdc from a 120 Vac to 9 Vdc wall adapter Outputs 5 Vdc to two USB-A female ports
- 96.608 % Efficient Buck Converter circuit from TI’s WEBENCH application

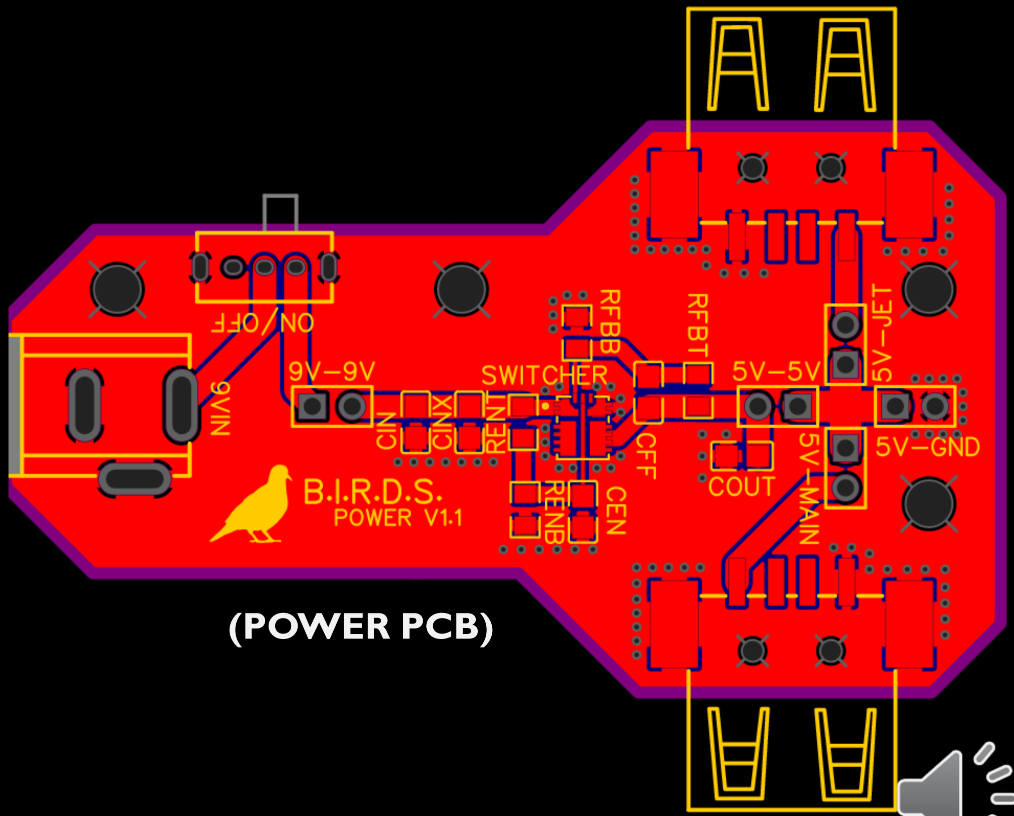
Main Board

- Controls and coordinates all peripheral devices
- Programed through ICSP
- Has an isolatable stepper driver circuit that can be independently powered or powered by the Main
- Powered by 5 Vdc USB-A cable or ICSP pins
- Power LED (of course)

PCB Layouts

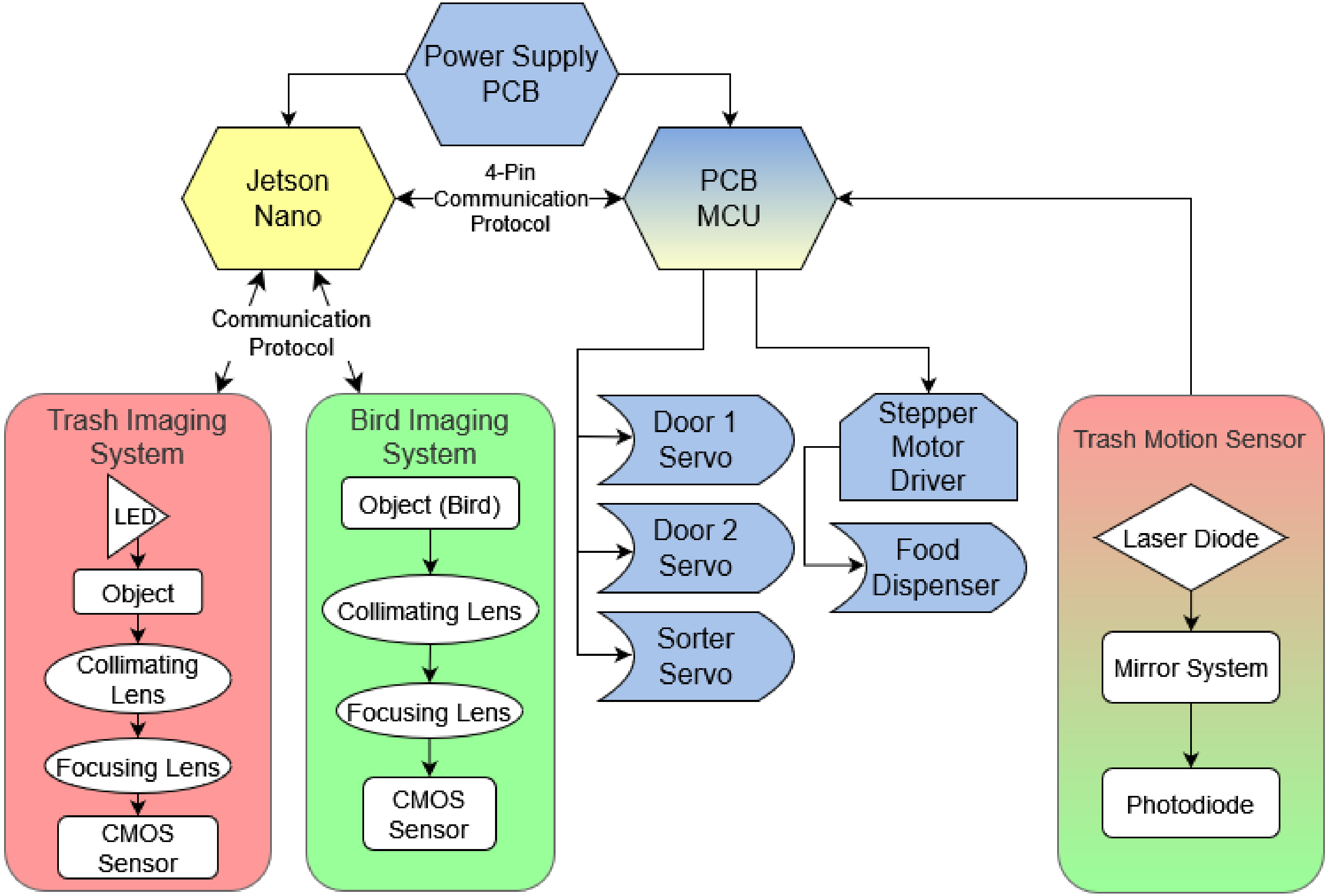


(MAIN PCB)



(POWER PCB)

Hardware Block Diagram



Work Distribution

Aiden

Will

Rodrigo

Donovan

Microcontroller

What we need:

- Low power mode to draw less power when reading sensors
- Enough GPIO pins to support our peripherals
- Interrupt capabilities
- Cheap
- Development board for easy testing and debugging

(ATmega328PB)

(MSP430)

(ESP32)

Development Board Selection

- Because almost every microcontroller has the features that we are looking for, we decided to try to minimize cost.
- Another goal was to use Arduino IDE to make writing the embedded software very simple giving much more time and focus to the machine learning section.
- To minimize cost we wanted to pick a development board that someone in the group already owns
- Our final choice of development board was the Arduino Uno

Device	Flash	RAM	GPIO Pins	Programming Language	Price
Arduino Mega 2560	256 KB	8 KB	54	Arduino IDE	Already Owned
Arduino Uno	32 KB	2 KB	14	Arduino IDE	Already Owned
MSP430FR6989	128 KB	2 KB	83	C	$26.59
MSP430G2553	16 KB	0.5 KB	24	C	Already Own

Microcontroller Selection

- Since we are using Arduino, we are sticking with the ATmega series of MCU
- Flash memory is more than what we need
- EEPROM is only for storing specific data when powered off
- The I/O pins are important we have more than we need
- PWM channel the size doesn't matter as long as we have 3
- We may need a couple ADC channels but all of these have way more than necessary
- Price was the biggest selling point for the ATmega328PB

Device	Flash (KB)	EEPROM (KB)	RAM (KB)	I/O Pins	16-bit resolution PWM channels	Serial UARTs	ADC Channels	Price
ATmega328PB	64	2	4	27	11 (10-bit resolution)	1	11	$1.63
ATmega640	64	4	8	86	12	4	16	$5.70
ATmega1280	128	4	8	86	12	4	16	$16.20
ATmega1281	128	4	8	54	6	2	8	$13.60
ATmega2560	256	4	8	86	12	4	16	N/A
ATmega2561	256	4	8	54	6	2	8	$25.40

Machine Learning Technology Comparison

What we need:

- Hardware capable of performing object detection of bird within our allotted time of 1 second
- GPIO pins to communicate with our processor on our custom PCB
- Ability to run 2 different machine learning models either in parallel or quickly swap

Final Consideration Options

NVIDIA Jetson Nano

- Can run multiple machine learning models in parallel
- Most expensive option for original price
- Most versatile because of GPU
- 4 USB ports for easy access of peripherals
- Has prebuilt library for machine learning (Jetpack)

Coral Dev Board

- Works very well with TensorFlow architectures
- Easy access to Wi-Fi and Bluetooth
- Has prebuilt machine learning models on TensorFlow hub that can be easily integrated

Raspberry Pi 4 + Coral Accelerator

- This was the first idea for machine learning
- Coral Accelerator helps match this with contenders
- Very popular choice for projects
- Very commonly sold out, or being resold for much more than its value
- Might run into issues with multiple pieces of hardware working together

Machine Learning Technology Selection

We need a device that can:

- Work well on a number of architectures
- Respond quick enough to meet our 1 second image processing goal
- Be received quickly to start testing
- Handle 2 different models, one for detecting birds and one for detecting cigarettes

Neural Net Framework Benchmark

Model	Framework	NVIDIA Jetson Nano	Raspberry Pi 4 B + Coral Accelerator	Coral Dev Board
MobileNet V1	TensorFlow	61.6	14.9	15.7
MobileNet V2	TensorFlow	72.3	18.2	20.9
ResNet-50 V2	PyTorch	27.7	52	56
ResNet-18	PyTorch	200	DNR	DNR
YOLOv3-320	Darknet	40	DNR	DNR
VGG – 19	MXNet	100	200	DNR
Unet	Caffe	55.5	200	DNR

Machine Learning Hardware Specifications

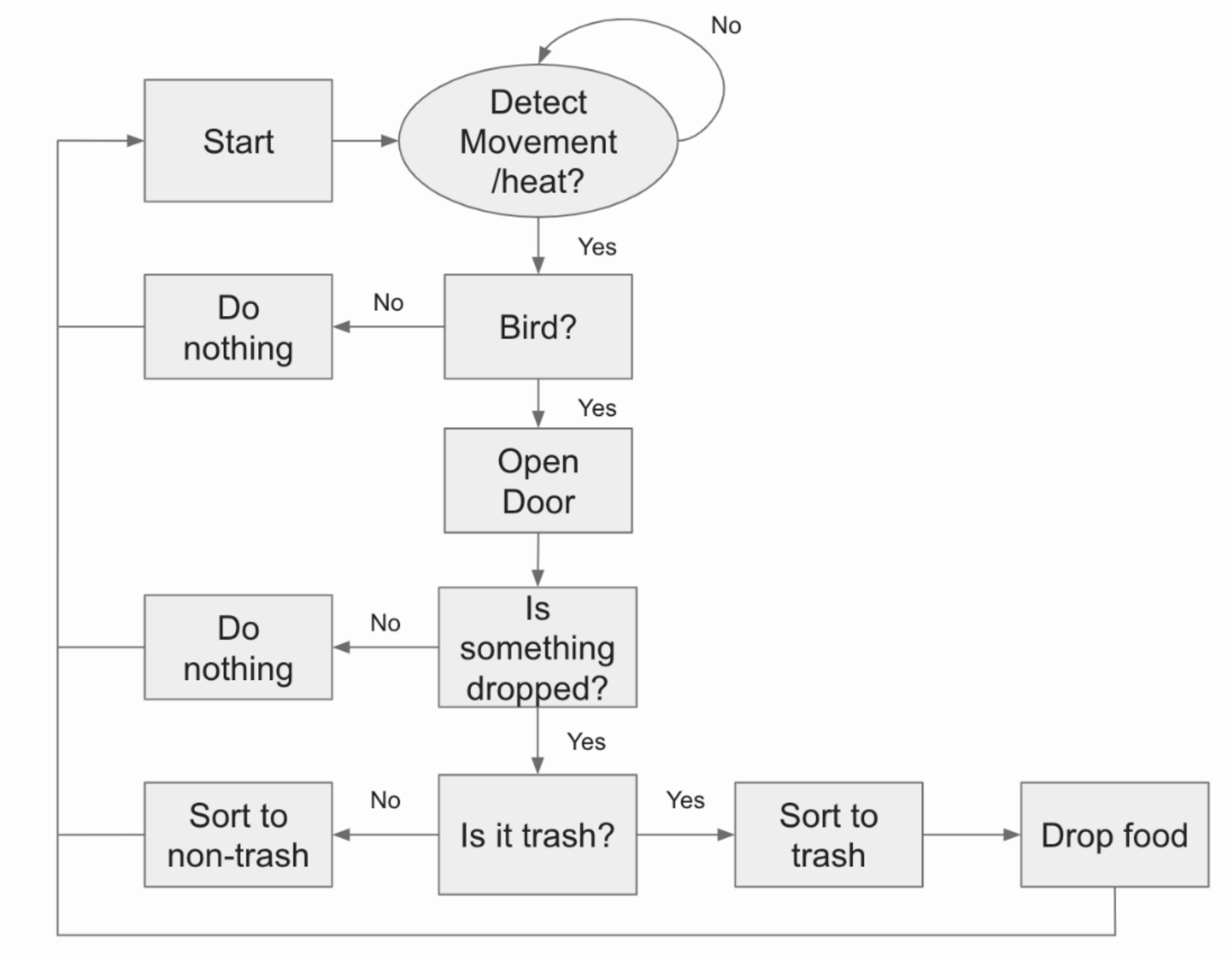
Feature	Coral Dev Board	NVIDIA Jetson Nano Developer Kit	Raspberry Pi 4 Model B + Coral USB Accelerator
CPU	Quad Cortex-A53, Cortex-M4F	Quad-core ARM A57	Quad-core Cortex-A72
Clock Rate	1.5 GHz	1.43 GHz	1.8 GHz
GPU	Integrated GC7000 Lite Graphics + Google Edge TPU coprocessor	128-core Maxwell @ 921 MHz	Google Edge TPU coprocessor
Cache	32KB L1 instruction and data cache, 1MB unified L2	48KB L1 instruction cache and 32KB L1 data cache, 2MB of L2 Unified Cache	48KB L1 instruction cache and 32KB of L1 data cache, 1MB of L2 Shared Cache
RAM	1 or 4GB LPDDR4 Memory	2 or 4GB LPDDR4 Memory	1, 2, 4, or 8GB LPDDR4 Memory
Flash	8GB eMMC, with a MicroSD slot for expandable storage	16GB eMMC, with a MicroSD slot for expandable storage	Expandable MicroSD slot for program and Operating system

Machine Learning Technology Selection

For our choice of machine learning technologies, we chose to go with the **NVIDIA Jetson Nano 4GB**

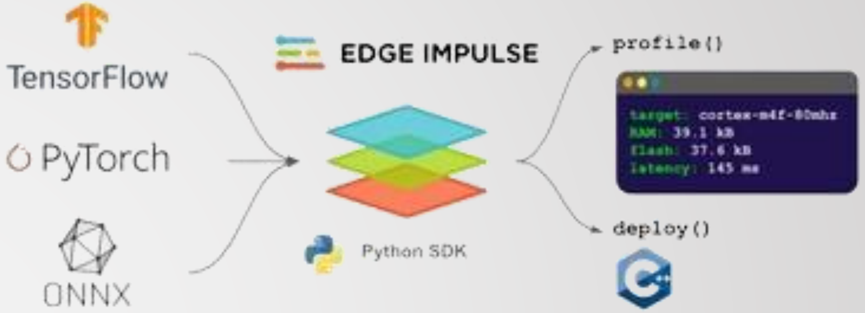
- Board comes with an easy way of installing a wide array of libraries (Jetpack SDK)
- This was the most versatile option guaranteeing performance
- Had a very easy setup with a microSD card using a tutorial from NVIDIA
  - Wide array of tutorials from NVIDIA on how to import prebuilt models and create your own object detection models.
- Was in stock and came within a few days of purchase
- Very easy to add Wi-Fi and Bluetooth capabilities should we need them in the future

Software Block Diagram

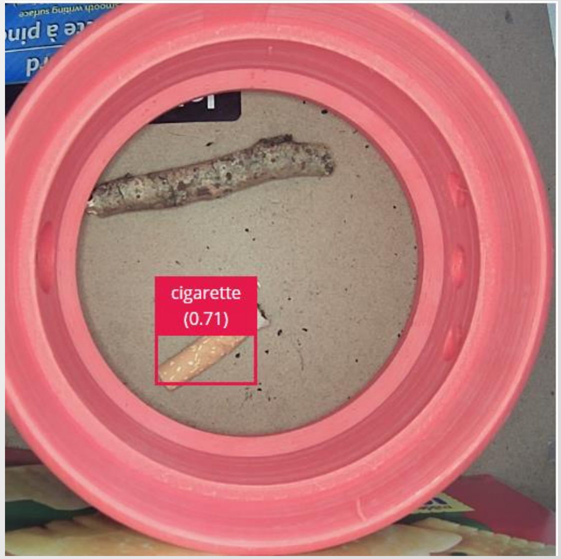


Training Machine Learning Platform

- Training is too much for the Jetson
- Need a cloud-based framework to train custom model
- Needs to interface with Jetpack SDK
- Needs to be a free platform
- To use Edge Impulse there are 3 main parts:
  - **Data Collection**
  - **Training**
  - **Testing**



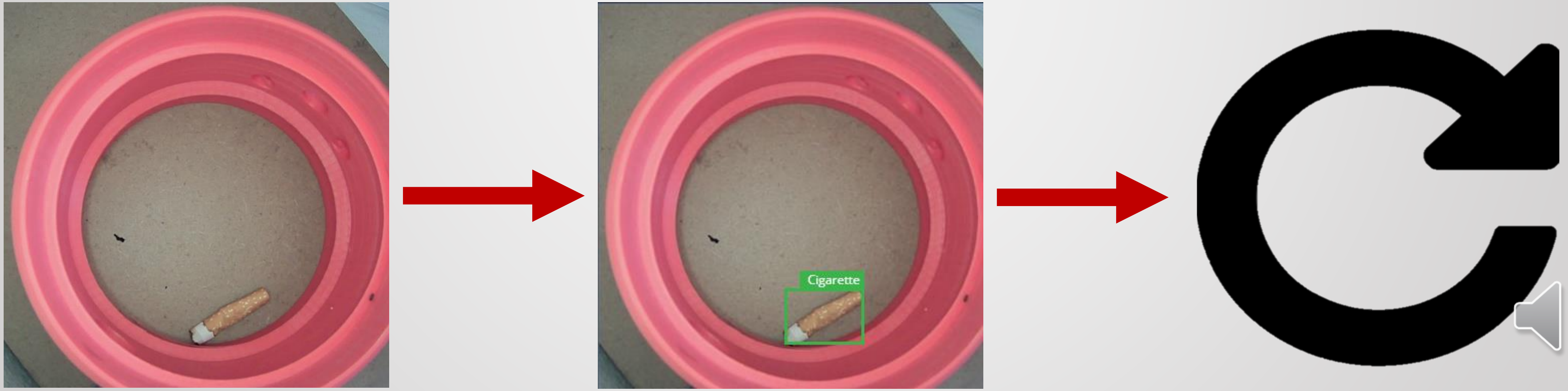
```
target: cortex-m4f-80mhz
RAM: 39.1 kB
Flash: 37.6 kB
latency: 145 ms
```



Example shot of our Object Detection for cigarettes

Edge Impulse – Data Collection

- Need to get images to train on
- Set up an old version of trash analysis to gather pictures with little variance
- After getting images draw bounding boxes over each cigarette
- Repeat this process until sufficient data is acquired
- After collection split the data to an 80/20 split for training and testing
- More Data means more computation time and only less than 20 minutes is free



Edge Impulse – Training

- To train a new model an “impulse” needs to be created
- An impulse consists of 4 things:
  - **Input Image Data**
  - **Image Pre-processor**
  - **Object Detection Framework**
  - **Output Features**

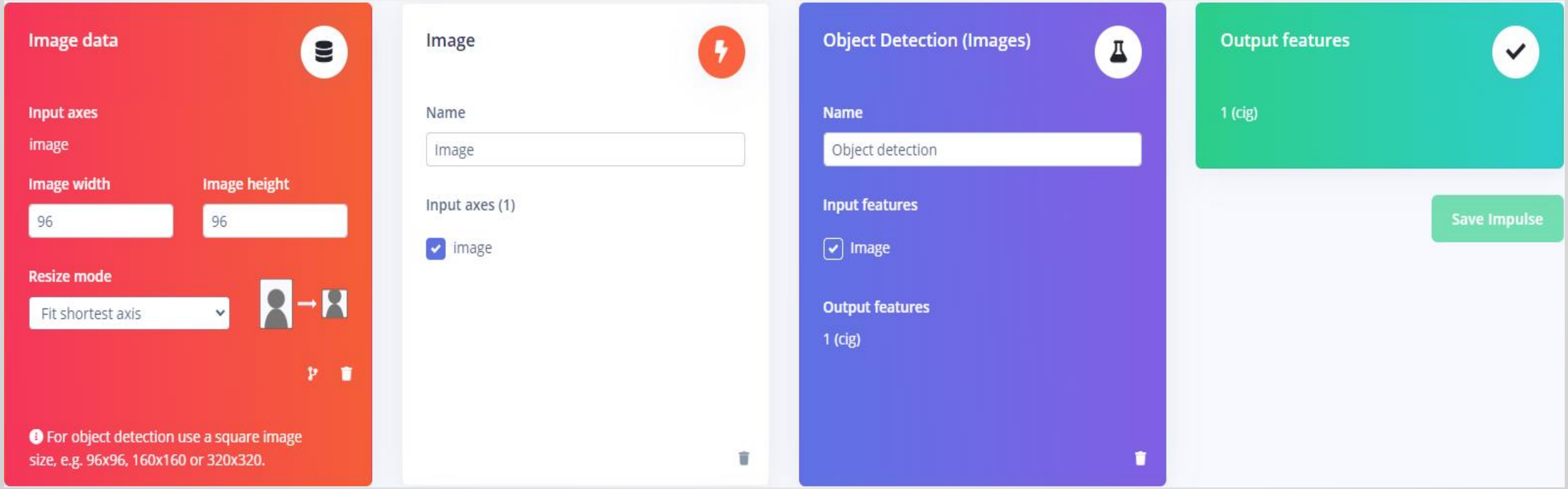


Image data

Input axes

image

Image width: 96

Image height: 96

Resize mode: Fit shortest axis

For object detection use a square image size, e.g. 96x96, 160x160 or 320x320.

Image

Name: Image

Input axes (1)

☑ image

Object Detection (Images)

Name: Object detection

Input features

☑ Image

Output features

1 (cig)

Output features

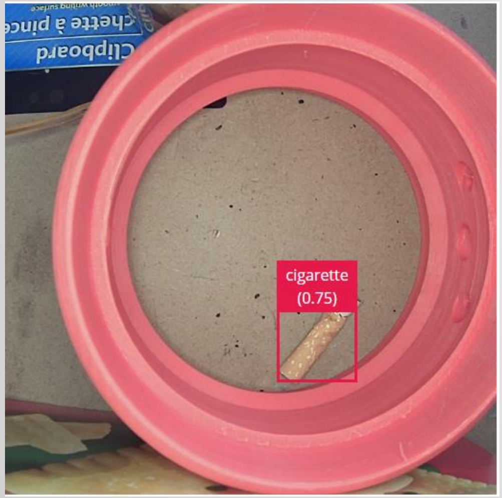
1 (cig)

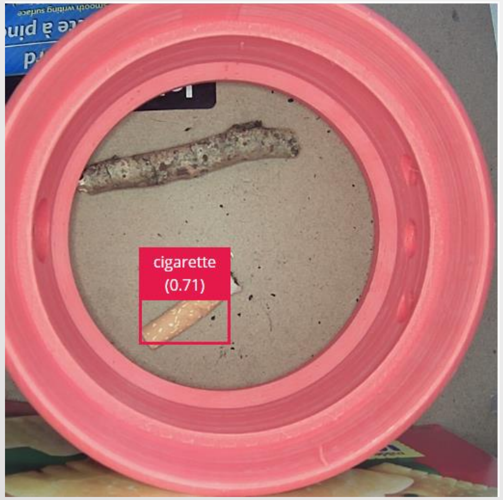
Edge Impulse – Training

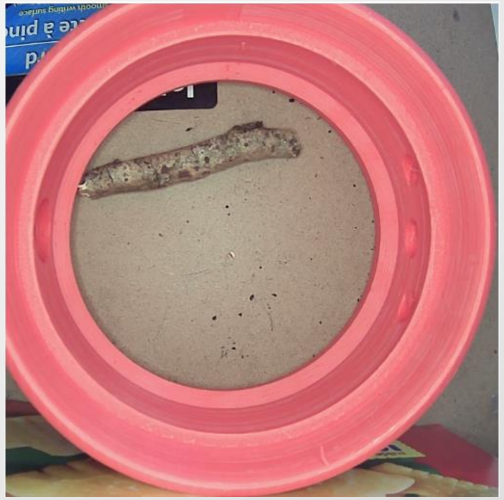
- **Image Data**
  - This is the data labeled from the Data Collection slide
- **Image Pre-processing**
  - Bigger images gives more accuracy but less response time and memory
  - We keep the images small to improve response time
- **Object Detection**
  - Which framework used is the most important aspect here
  - We found FOMO (Faster-Objects More-Objects) to be the best balance of these 3
  - We trained using 100 training cycles with a learning rate of .001 and using data augmentation with a validation size of 20%
- **Output Features**
  - This is the least impactful with just naming the output
  - If we had more than 1 classification this would be more important
- It is important to balance response time, accuracy, and memory
- The final model gets evaluated using an F1 score to judge how accurate the model is

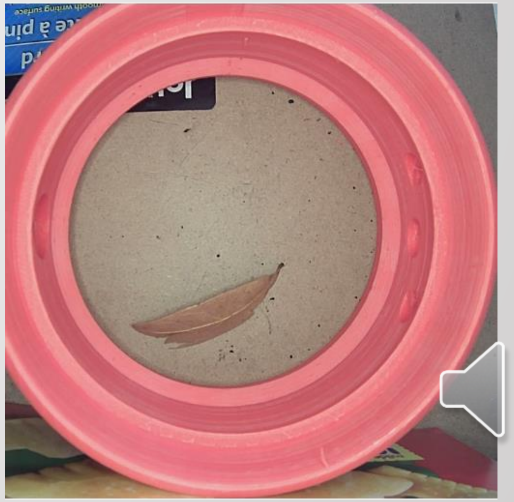
Edge Impulse – Testing

- To avoid overfitting there needs to be data that is not used to train
- We used the 20% testing data acquired earlier to test
- We achieved an accuracy above 80% reaching our goal
- Edge Impulse offers live classification to test our results
- We connected a camera and threw various cigarettes and sticks to test what it could classify
- The numbers below the name say the confidence level which we can tweak as needed









Budget

Item	Price	Quantity
Microcontroller	$13	1
NVIDIA Jetson Nano	$150	1
Servo Motor	$10	3
Stepper Motor	$13	1
Custom PCB	$20-$30	2
Housing Material	$200	1
LED Light Source	$15	10

Item	Price	Quantity
Collimating Lens	$23.50	2
Focusing Lens	$20	2
IR Filter	$5	2
CMOS Sensor	$25	2
Laser Diode	$13.18	1
Photodiode	$0.32	20
Mirror	$0.28	50
TOTAL	**$796.58**	

Work Distribution

Aiden Nipper

- Lens System Design (Trash-Cam)
- Laser Motion Sensor Design
- Housing & Structural Design

Will DiSalvo

- Lens System Design (Bird-Cam)
- Laser Motion Sensor Design
- Structural Design

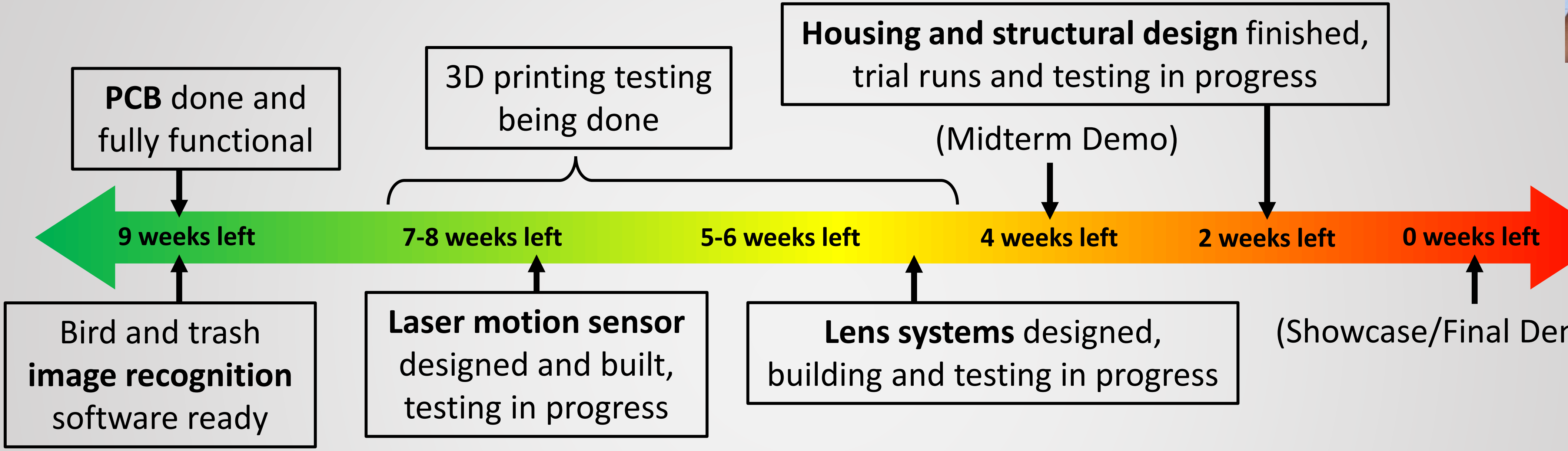
Rodrigo Guerra

- PCB Design
- Wiring and Soldering
- Housing & Mechanical Design

Donovan Reynolds

- Machine Learning (Jetson)
- Embedded Software
- Website

Progression and Future Milestones



PCB done and fully functional

3D printing testing being done

Housing and structural design finished, trial runs and testing in progress

(Midterm Demo)

Bird and trash **image recognition** software ready

Laser motion sensor designed and built, testing in progress

Lens systems designed, building and testing in progress

(Showcase/Final Demo)

- Anticipating system to be tested outdoors after showcase, with hope of obtaining some cool results
- Hope for the project is that others improve design and use it to train birds in a specific area
- Having the BIRDS in major cities and/or universities could slowly begin to change the way our streets look

THANK YOU!