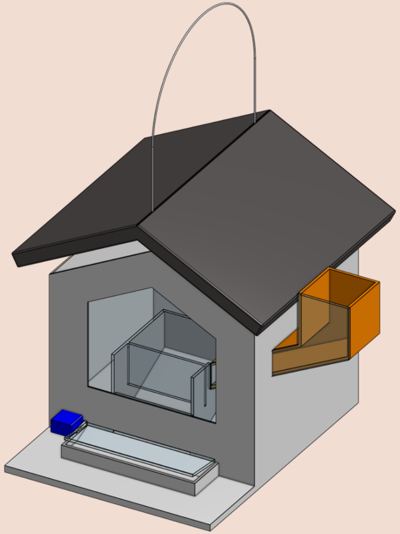
The Smart Bird Feeder

Senior Design Project - Group 7

The Smart Bird Feeder

Design Goal

Our goal was to build a smart bird feeder that maximizes the potential of the hobby! By emphasizing automatic features and designing around common feeder problems, our team has created the perfect bird feeding device.



Our Team

John Hauff
Computer Engineer

Nikki Marrow
Electrical Engineering

Paul Amoruso
Computer Engineer

Matthew Wilkinson
Computer Engineer

Project Features

Species Detection

Livestream Capabilities

Pest Deterrent

Smart Application

Photo Storage

Notifications

Project Motivation

- Bird feeding is time consuming
    - Too much time is spent waiting for the desired species
    - Too easy to miss the bird encounters
    - Keeping track of the bird feed levels
    - Cataloging which species have been seen
- Predators and backyard pest
    - Other animals will consume the bird feed
    - These pests scare away other creatures
    - Waste bird watcher's time and resources
- Space in the market
    - No product exist that incorporates smart features and deterring mechanisms
    - Our group recognized the potential in the market

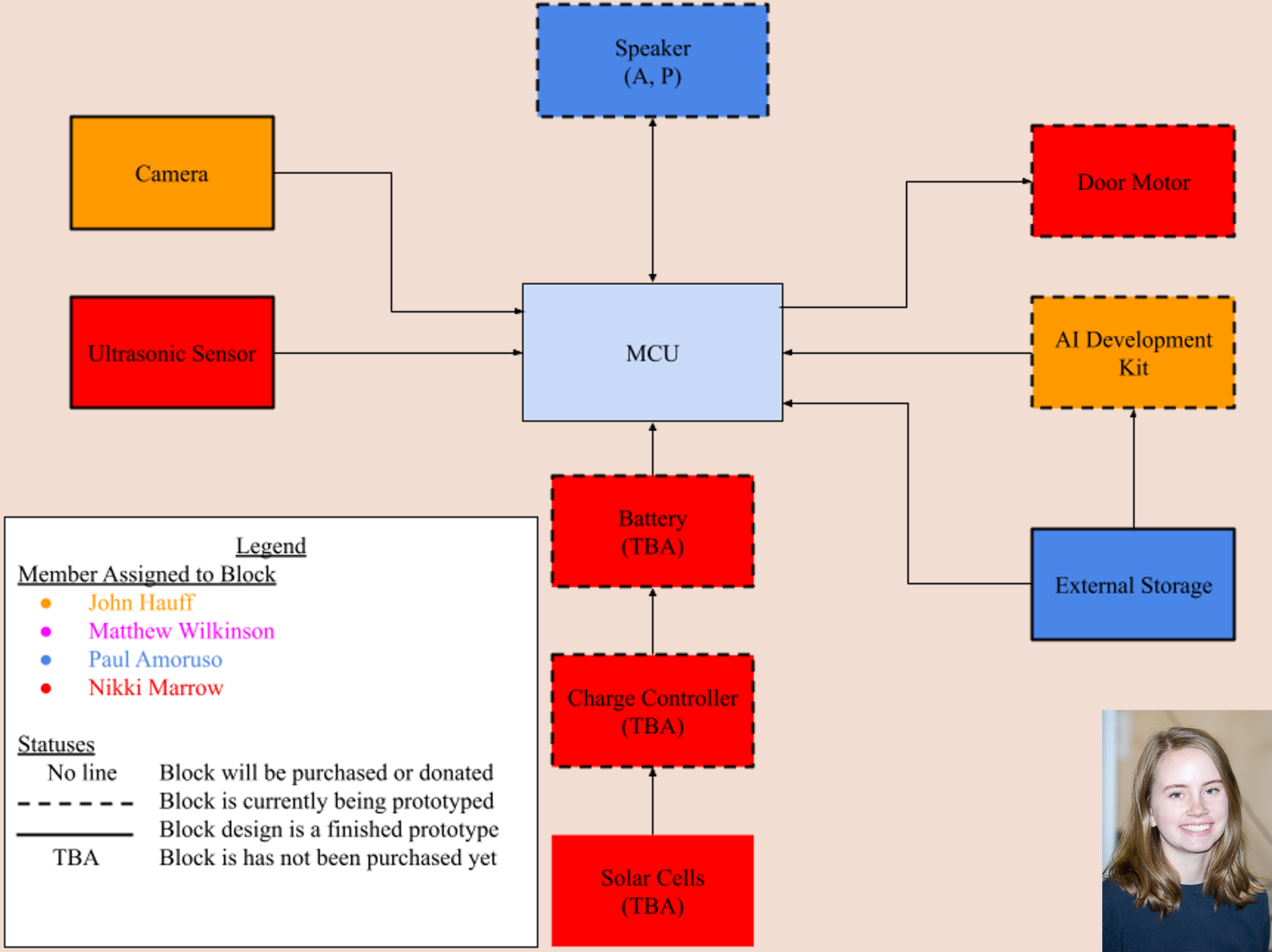
Project Goals and Objectives

- Maximize the users time by incorporating smart elements into the hobby
  - Incorporate species detection using machine learning
  - Auto-cataloging and saving memories
  - In-depth notification systems
  - Accompanying smart phone application
- Design deterrent mechanisms to combat backyard pest
  - Block access to feed with hatch mechanism
  - Scare away pest with alarm
- Fill the space in the empty market

Engineering Requirements

Engineering Requirement	Specification
Accuracy of bird species and squirrel detection must exceed a certain confidence level to capture images.	≥ 90%
The dimensions should not exceed the size of a common bird feeder	≤ (8" x 8" x 14")
The system should have enough power to run for a certain amount of time.	≥ 8 hours
Duration of external battery recharge should not exceed a certain time	≤ 6 hours
The alarm speaker should have a output range of 60-95 dB.	60-95 dB
Hatch door closing time should close within a certain amount of time.	≤ 3 seconds
User will be notified of a new bird memory within a certain time period.	≤10 seconds

Hardware Block Diagram

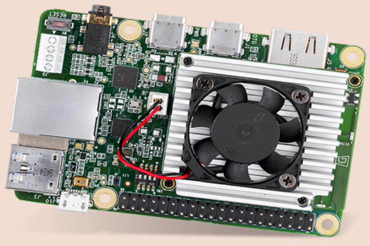


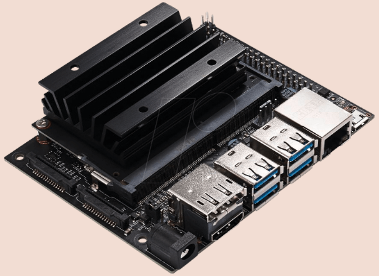
Machine Learning Technology Comparisons

What we need:

- Machine learning capable hardware for object detection of bird species & squirrels
- GPIO pins to communicate with our custom PCB

Options Under Consideration





Coral Dev Board

- First device considered
- Edge TPU Python API
- Existing Edge TPU computing models that can be adapted
- Created to work well with TensorFlow Lite models
- 1, 2, or 4GB LPDDR4 SDRAM

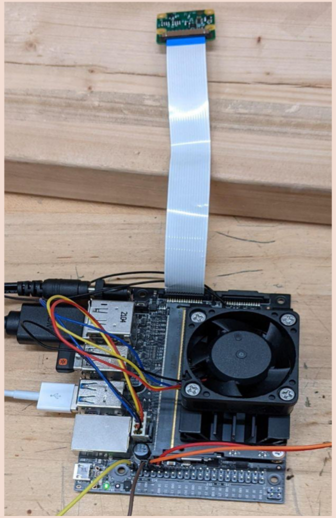
Jetson Nano

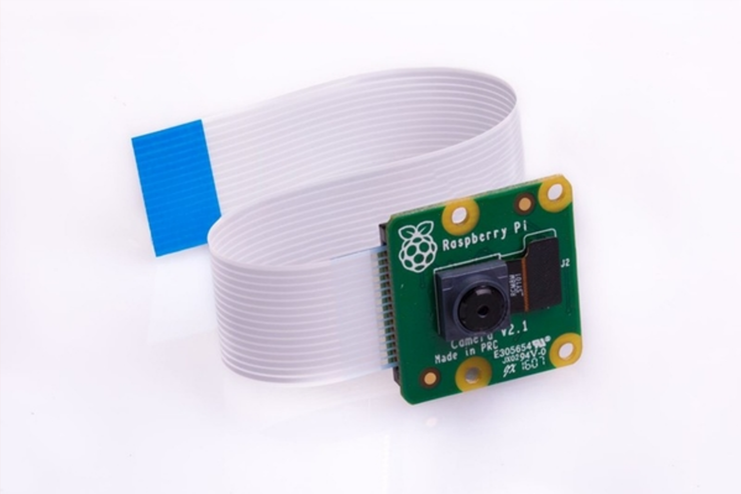
- Can run multiple ML models in parallel.
- 4GB LPDDR4 SDRAM
- No on-board Wifi included (but may be added)
- JetPack SDK with full Linux OS that contains libraries and APIs for deep learning

	Coral Dev Board	NVIDIA Jetson Nano Developer Kit
CPU	Quad-core Arm Cortex-A53, plus Cortex	Quad-core ARM Cortex-A57 MPCore processor
CPU Clock Frequency	1.5GHz	1.43GHz
GPU	Edge TPU coprocessor; Vivante GC70000Lite	128-core NVIDIA Maxwell @ 921 MHz
Cache	32 KB L1 Instruction Cache; 32 KB L1 Data Cache	48 KB L1 instruction cache per core; 32 KB L1 data cache per core | L2 United Cache: 2MB
Communication	On-board WiFi included	On-board wifi module not included (can be installed)
USB Ports	1 USB 3.0 Type-C port, 1 USB 3.0 Type-A port, and 1 USB 2.0 Micro-B serial port	4 USB 3.0 ports, 1 USB 2.0 Micro-B port
Price	Starting at $129.99	$99.99

Machine Learning Technology Choice – Jetson Nano + Raspberry Pi Camera Module v2

- Our team decided to go with the NVIDIA Jetson Nano 4GB development kit. ue to a wide array of features, impressive processing power, integrated OS, prebuilt machine learning libraries and modules, cost, and great community support.
- Jetson Nano pairs with the Raspberry Pi V2 8MP camera module to capture bird images and live video
- Purchased cooling fan for $13.50 to keep the Jetson Nano module within a reasonable temperature range





Microcontroller

- Specifications: Type of communication, number of pins, operating voltage, and size.
- The microcontroller must support 4 peripherals - 6 I/O pins are required
  - 2 - transmitting and receiving data from the Jetson Nano
  - 1 - Motor and hatch control
  - 2 - for the ECHO and TRIG pins on the ultrasonic sensor
  - 1 - External Speaker
- The CPU speed can be tempered, as most of the processing is being performed by the Jetson Nano. For our requirements, 16 MHz is an ideal speed.
- As the device is reliant on battery power, it's important to get a MCU with low power options available.
- Choice: MSP430G2553

	MSP430FR6989	MSP430G2553	ATmega328P
Cost	$9.43	$3.38	$2.63
GPIO pins	83	24	23
Speed	16 MHz	16 MHz	20 MHz
Architecture	16-bit	16-bit	8-bit
Memory	128 KB	16 KB	32 KB
Operating Voltage	1.8V ~ 3.6V	1.8V ~ 3.6V	1.8V ~ 5.5V
Program Memory Type	FRAM	FLASH	FLASH
RAM Size	2 KB RAM	0.5 KB	2 KB SRAM
SPI	4	2	Yes
UART	2	1	UART/USART
I2C	2	1	Yes
Package	80-LQFP	20-PDIP	28-DIP

Motor Control Requirements

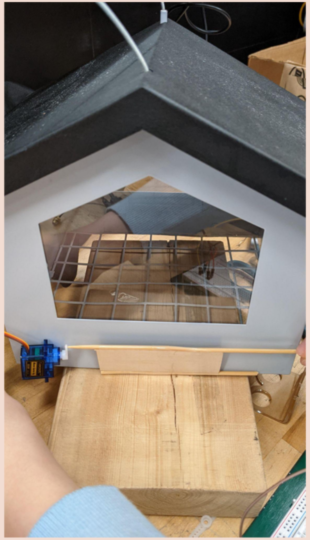
The motor will be used in conjunction with the hatch door to complete our defense mechanism features. It will be powered by the MCU when the Jetson Nano monitors predator activity.

- Specifications: cost, torque, speed, size, acceleration, and power

	Brushed DC Motor	Brushless DC Motor	Stepper Motor	Servo Motor
Lifetime	1,000 to 3,000 hours	over 10,000 hours	over 10,000 hours	over 10,000 hours
Torque	0.49 mNm at 5VDC	0.021 Nm at 12VDC	490 mNm holding at 5VDC	0.148 Nm at 6VDC
Efficiency	75%-80%	70%-90%	~65%	over 85%
Cost	$3.22	$14.95	$26	$7.95
Voltage/Current (typical)	1.5 - 24V 4A at 3V	4 - 24V 5 - 15mA	1.68A at 2.8V	200mA - 10A
Control Complexity	No MCU needed	MCU needed for feedback	MCU needed to send pulses	MCU needed to send PWM

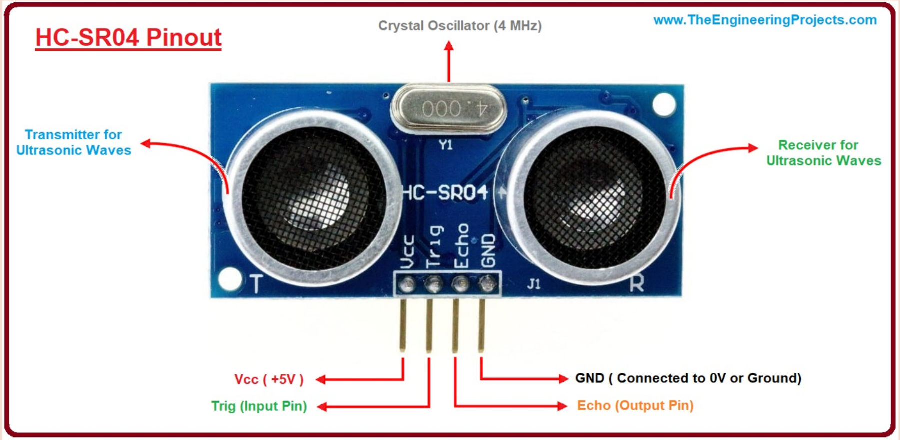
Motor Selection – Servo

- Solid torque, efficiency, cost, and power requirements
- Requires PWM signal from the microcontroller in order to specify the position



Ultrasonic Sensor

The key aspect of the ultrasonic sensor in our design is the minimum and maximum range range. Since the bird house will be relatively small, the sensor will have to be able to read distances of about 1 – 5 inches, depending on the depth of the bird feed container in the final design. The sensor must be able to detect the levels of bird feed by looking down vertically on the bird feed from the ceiling of the bird house.



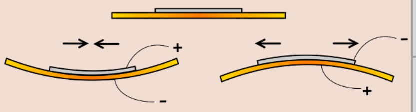
Sensor Specifications

Operating Voltage	5 V
Working Current	15 mA
Working Frequency	40 Hz
Maximum Range	4 m
Minimum Range	2 cm
Measuring Angle	15 degrees
Dimensions	45 x 20 x 15 mm
Accuracy	3mm

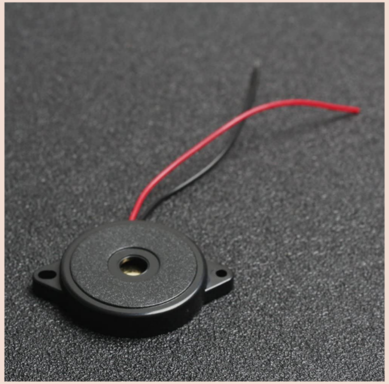
Alarm Design

While the hatch component will make pest unable to access the feeding chamber, they might till stick around to find a way around the hatch system. In order to maximize the time birds could spend feeding, the device is equipped with an alarm system to scare away pest.

- Alarm will sound in the range of 75-90 dB
- Piezoelectric speakers are cheap and small in size
- PWM signal strength can emit sounds of different loudness
- Activate when the machine learning processor detects a pest
- Anything past 90 dB might be dangerous to little ears, the goal is to provide an annoyance, not damage



Resonant Frequency	4.0 +/- 0.5 KHz
Rated Voltage	5.0 V
Operating Voltage	1 ~ 30 V
Max Current Consumption	20 mA
Min. Sound Output at Rated Voltage	80 dB

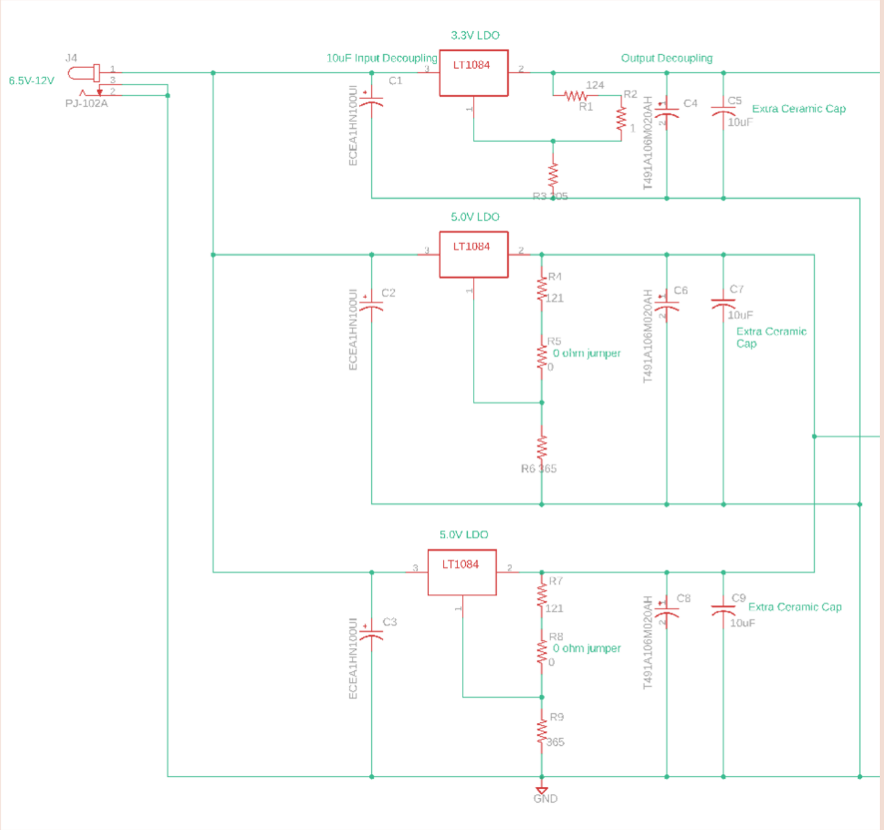


Power Distribution

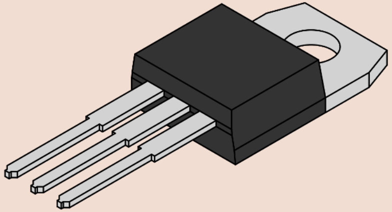
Given our bird feeder relies on battery and solar power, power consumption was of prime importance in our design decisions. When possible we chose components with low consumption and always utilized power saving features

	Voltage Requirement	Current Requirement
Developer Kit	~5 V	2 - 4 A
MCU	1.8 - 3.6 V	Operating: 230 uA Standby: 0.5 uA
Servo Motor	4.8 - 6 V	500 mA
Speaker	5 V	4 - 6 mA
Ultrasonic Sensor	5 V	15 mA
Total	5 - 6 V	2.5 - 4.5 A

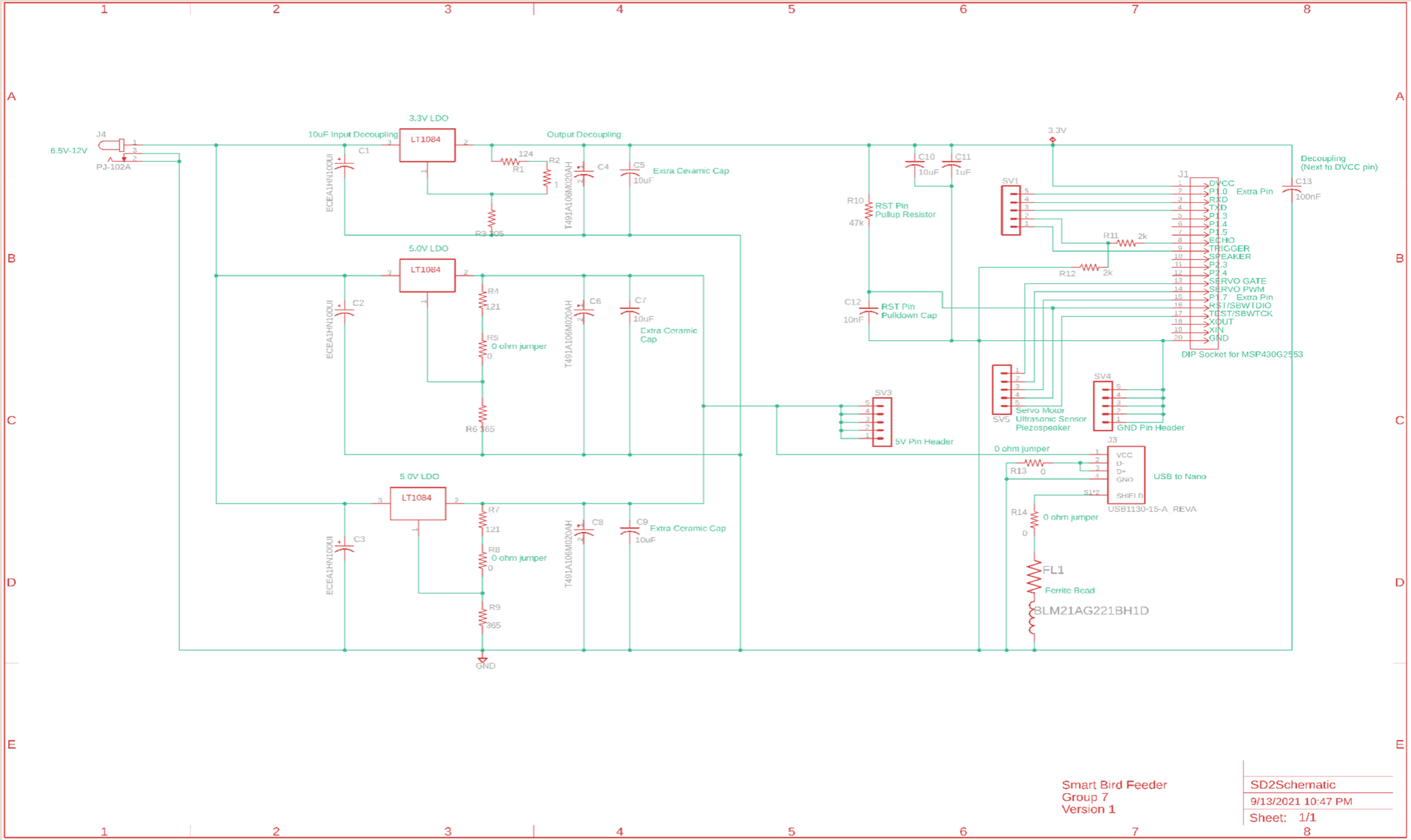
Voltage Regulators & Converters



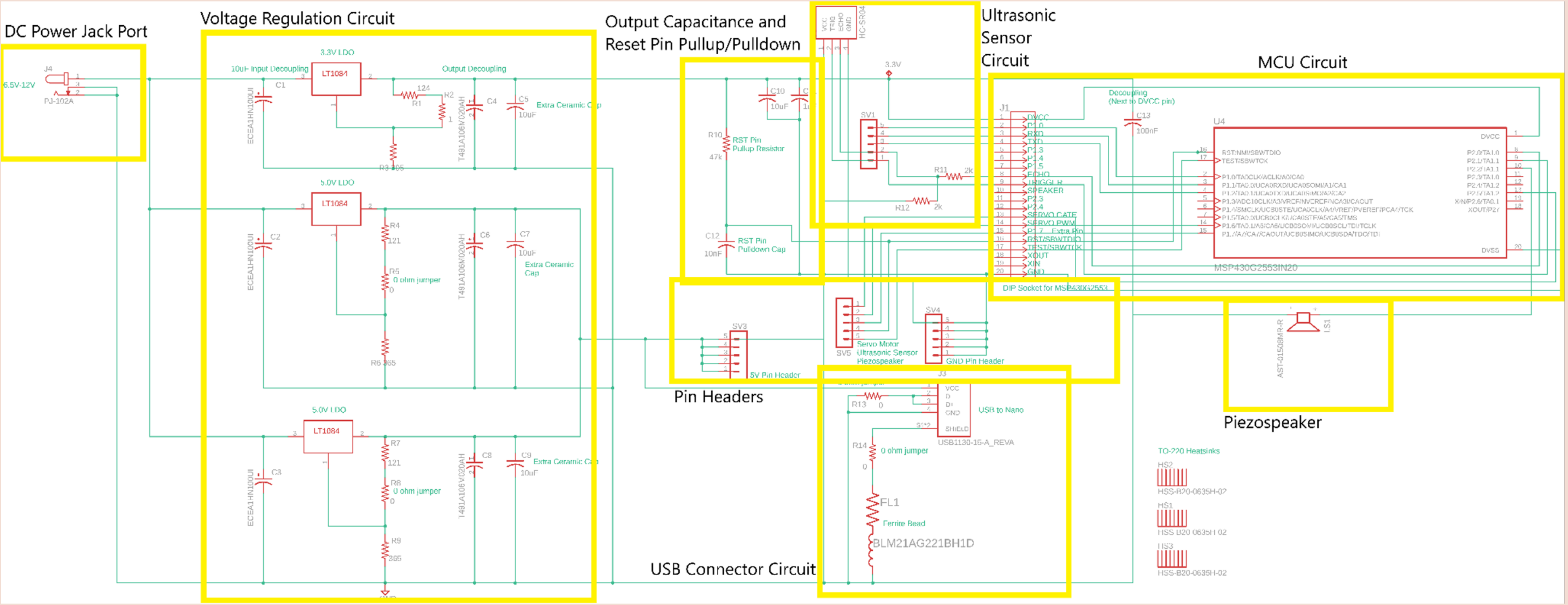
There will be two voltages that will be needed to supply in our system. We will need to supply 3.3 volts for the MCU and a supply of 5 volts for the remaining components, which includes the development kit for object detection. We plan to have a rechargeable battery that will store a higher voltage than is required, with the utilization of voltage converters.



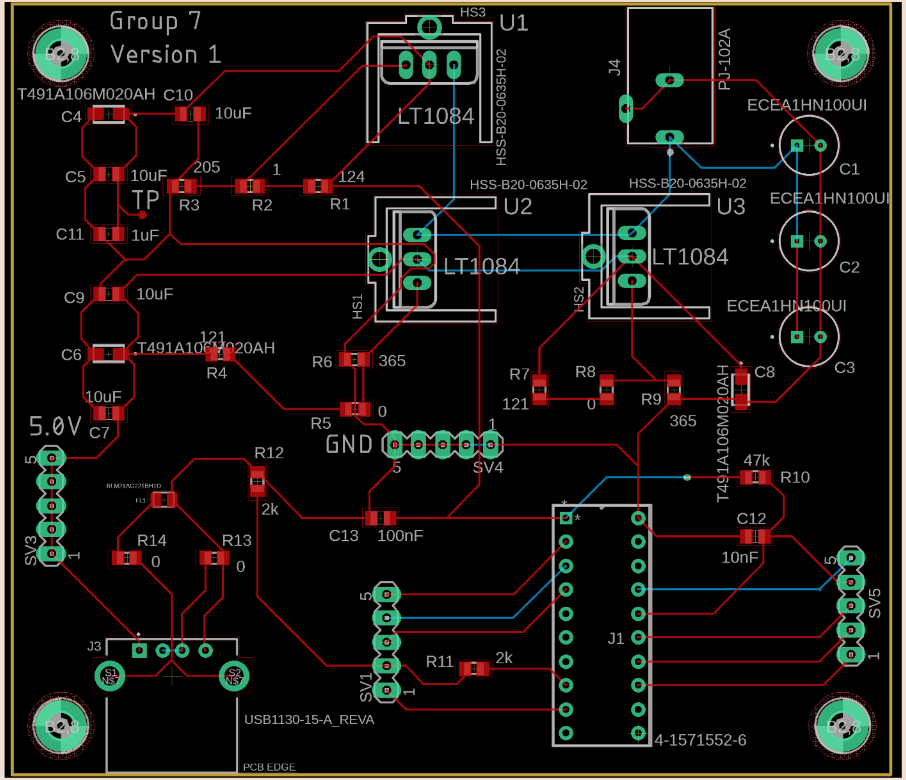
PCB Schematic

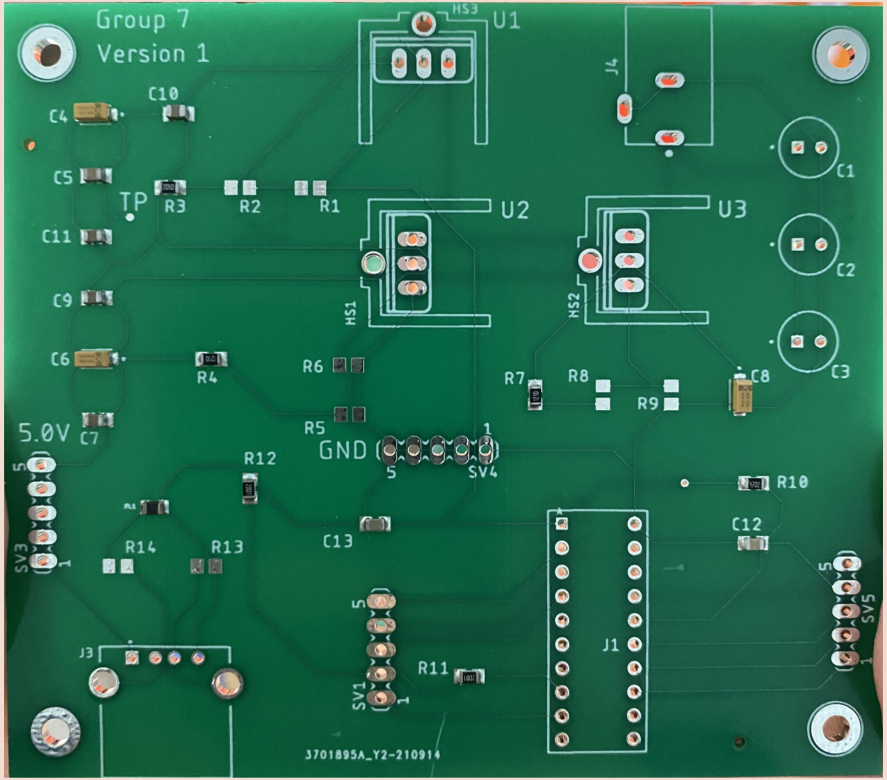


Overall Schematic

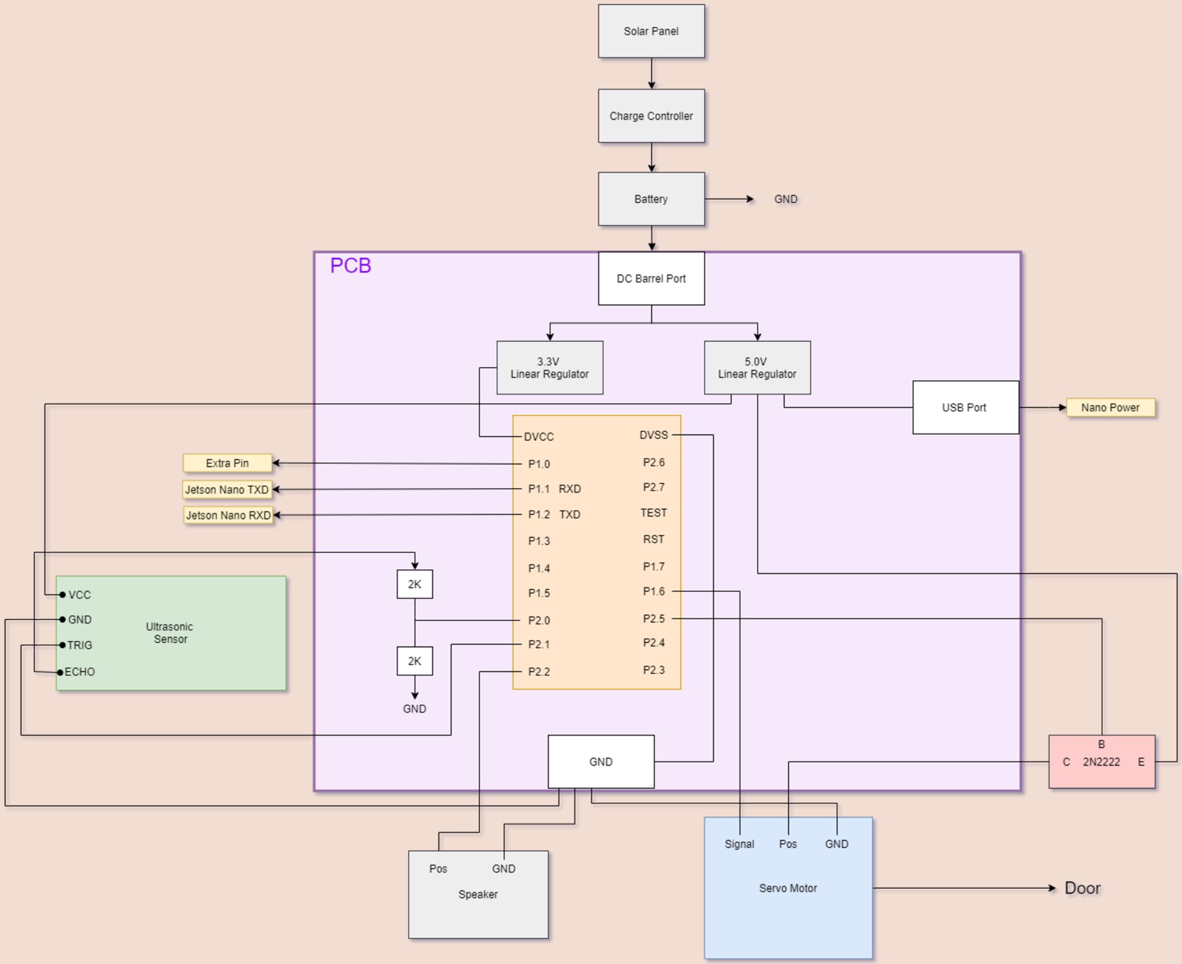


PCB Board Layout





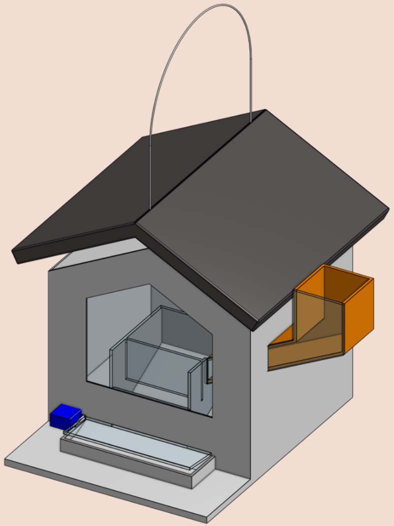
Hardware Schematic

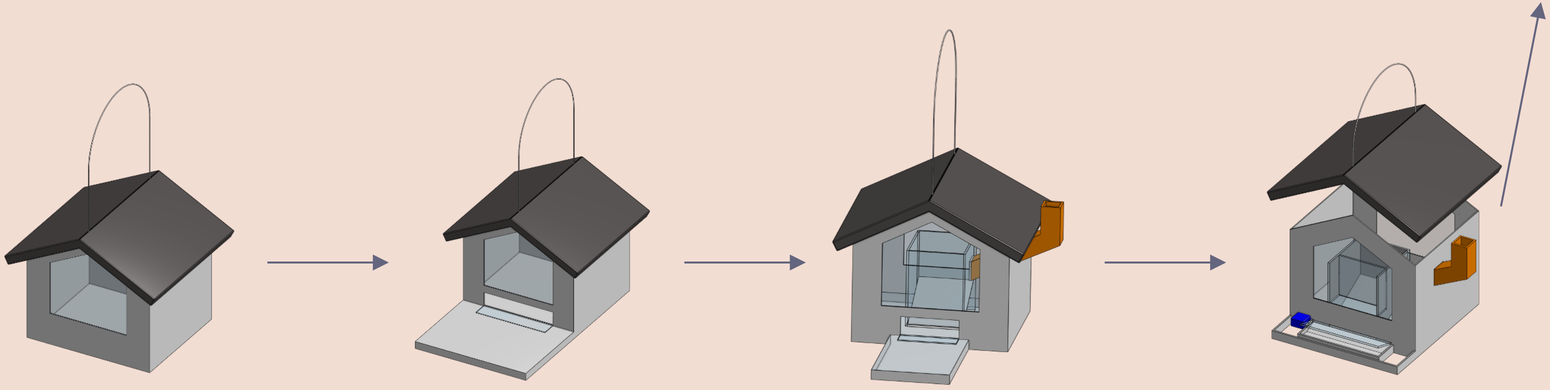


Bird Feeder Structural Design

- Purchased assembled birdhouse retail
- Rendered a 3D model of the birdhouse in CAD
- Design choices:
  - Trough design
  - Servo placement
  - Feed container design
  - Chimney design (3D printed)
  - Bird perch
  - Electronics housing
  - Camera placement

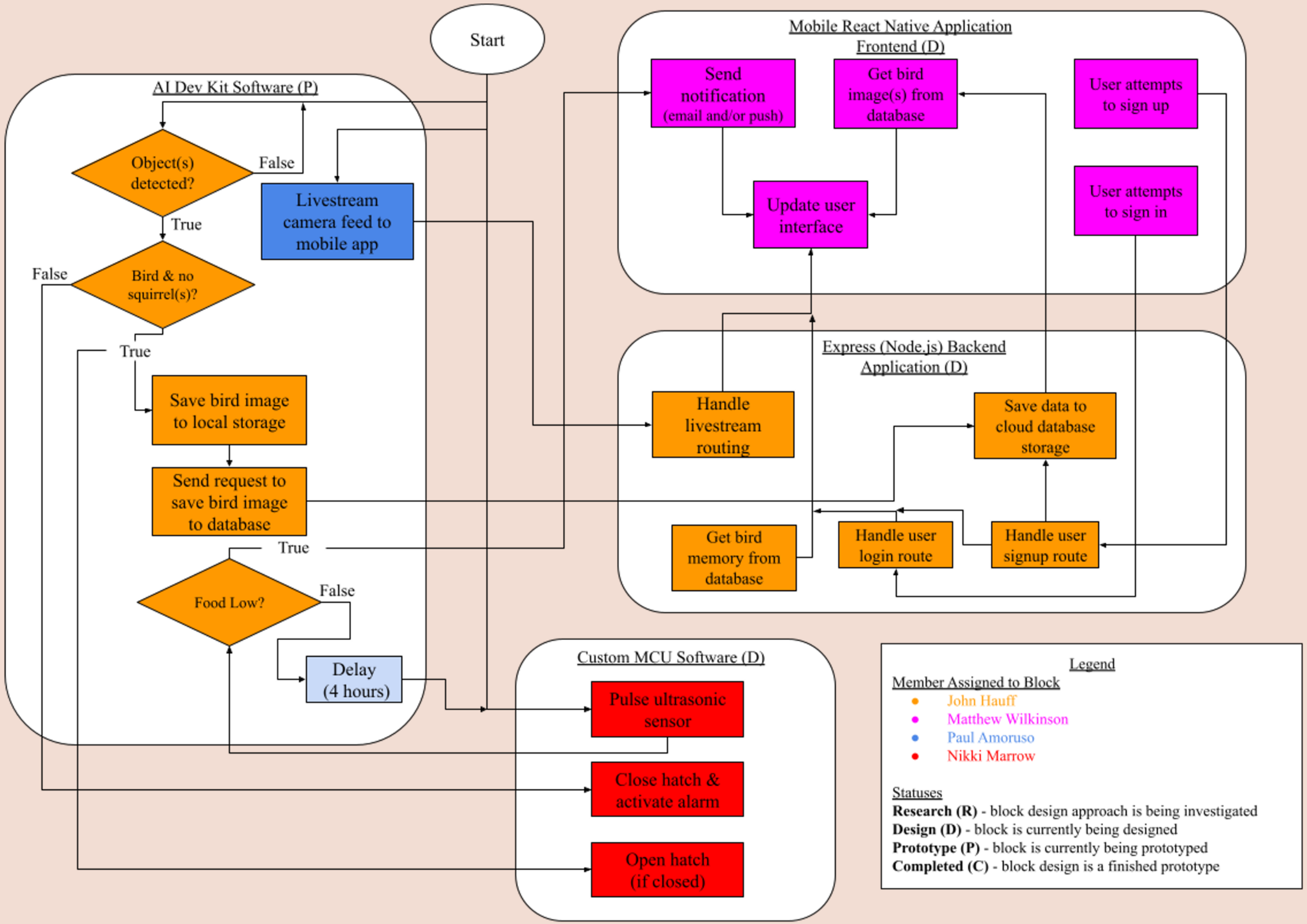
Most current design to date





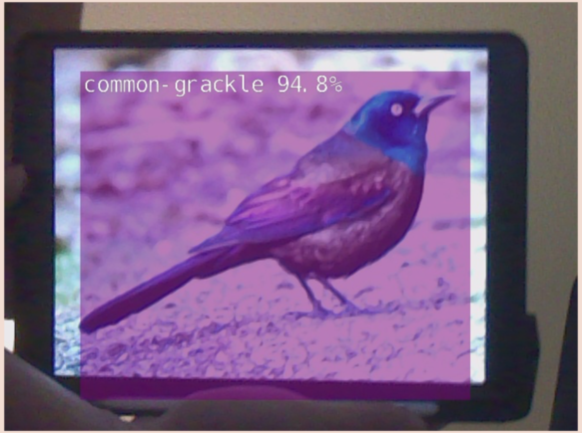
Start
(retail birdhouse)

Software Block Diagram



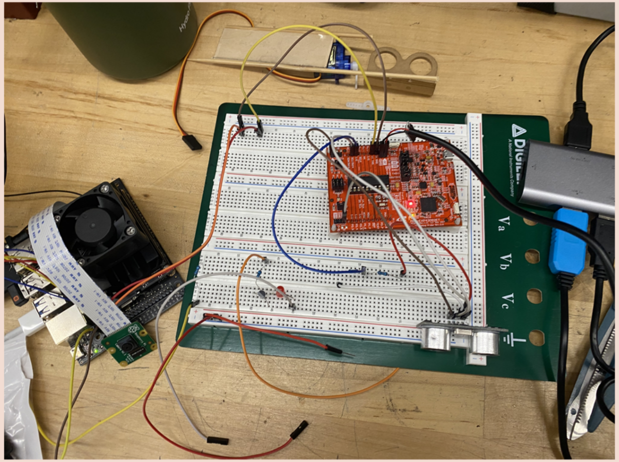
Jetson Nano & ML Model Software

- NVIDIA JetPack SDK provides detectNet for running object detection ML models
- A pre-trained SSD-MobileNet V2 ML model will be retrained for object detection of bird species and squirrels using PyTorch
  - Using a Pascal VOC dataset annotated with Computer Vision Annotation Tool (CVAT)
- All software designed & used for the ML model is written with Python
- Raspberry Pi camera sends video feed to ML model
- Once bird and bird species is detected, image is saved and sent to cloud database
- Otherwise, close hatch and sounds alarm
- UART serial communication between Jetson Nano and MSP430 MCU
  - Jetson Nano UART software developed with Python
  - Nano asks MCU to power on ultrasonic rangefinder & send back feed level readings
    - Jetson Nano receives feed level readings & decides if feed refill is required
  - Nano tells MCU on our PCB to open/close hatch and/or sound alarm

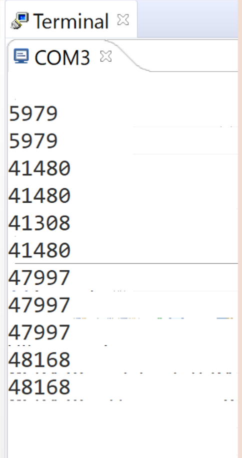


MSP430G2553 Software

- Tasks:
  - Ultrasonic Sensor
    1. Send command from Jetson Nano to the MSP through UART
    2. MSP sends signal to ultrasonic rangefinder sensor and receives back the raw data
    3. MSP converts raw data into a distance in micrometers
    4. Read feed levels
    5. Send data to Nano via UART
  - Alarm
    1. Send command from Nano to MSP through UART
    2. MSP sounds the alarm
  - Hatch
    1. Send command from Nano to MSP through UART
    2. MSP closes or opens the hatch
    3. MSP sends a message to the Nano notifying it that it has completed the task
- How we will do it:
  - UART serial communication between Jetson Nano and MSP430 MCU
  - Flashing C programs onto the MSP430 using Code Composer Studio
  - Messages transmit as a stream of characters



Prototyping UART communication (Nano <-> MSP430)



Data received by the MSP from the Ultrasonic Sensor

Full Stack Development Technologies

All of our device's smart features rely on an accompanying application built using full stack technologies.

Flutter

Flutter is a software development kit created by Google

- Flutter uses the Dart programming language
- Built-in widgets
- Up and coming development SDK, so there is less documentation

React Native

React Native is a framework used to build mobile applications

- Javascript based application
- Works well in MERN stack applications due to the synergy of coding in Javascript
- Varied documentation and support given it's popularity
- The group has familiarity in React Native, as we've worked with it in the past

Full Stack Development Technologies

A few other tools used to help create our application:

Expo is a framework that allows users to test React applications

- Applications can be modified and restested quickly
- Expo Go app allows the program to run the application directly on smartphones
- Essential for testing changes

MongoDB:

- noSQL database
- MongoDB Atlas offers a cloud database

Node.js + Express:

- Backend API uses Node.js runtime environment with the Express framework
- Routes all REST API requests

Operating Systems

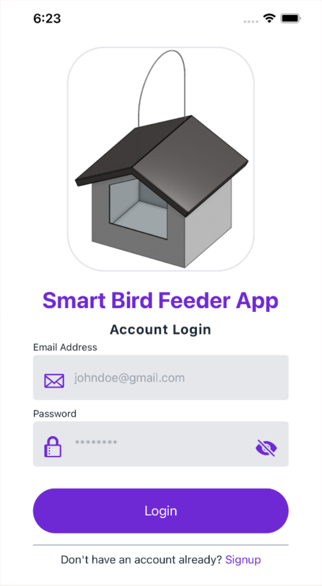
Given React Native, Express, and Expo can support both iOS and Android, the functionality is working to build both apps. However the front end design and cost of implementing the apps differ:

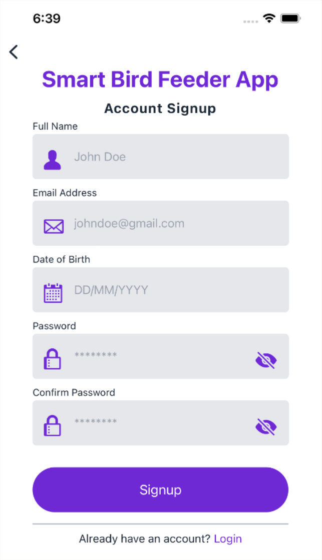
- IOS Store requires an approval process
- The application fee is $99 dollars annually

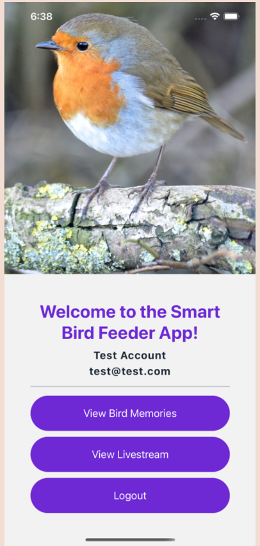
- No approval process
- Google Play store requires one time fee of $25

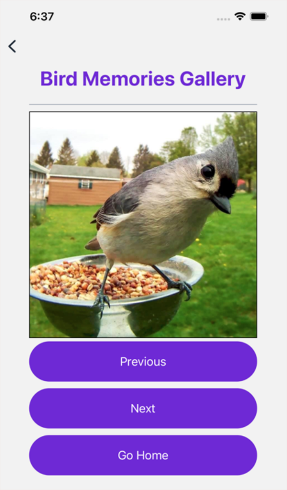
Mobile Application Design

- There will be four main pages
  - Login/Sign up
  - Home
  - Memories
  - Livestream







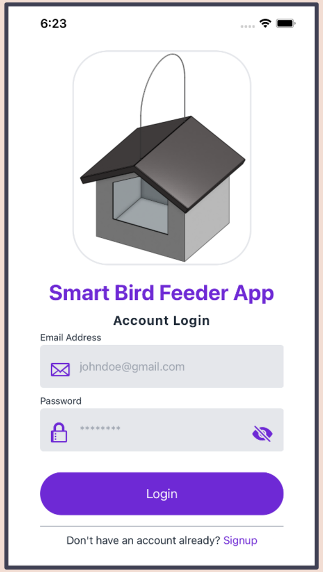


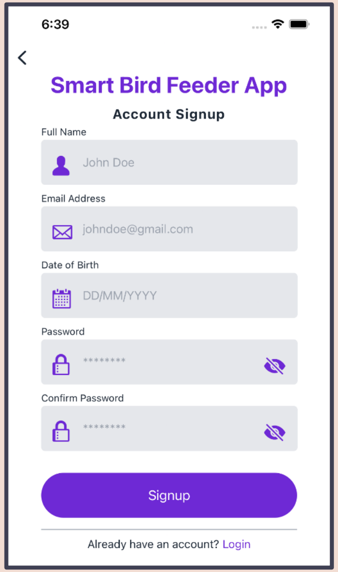
*** This example is not meant to highlight any stylistic choices, but instead set the outline for the applications structure.*

Application Design

Overall design

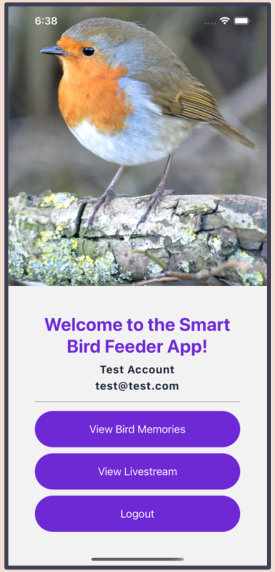
- Page layout must be purposeful, with a clear goal associated with each section of the UI.
- UI elements synonymous with common symbols (IE. notification bell)
- Color scheme used strategically to convey levels of urgency or provide comfort





Login/Signup

- Name, username, email is stored to a temporary database
- User confirms email address and then is prompted to create a password
- User data is then transmitted to permanent database
- User then connects the bird feeder to the application



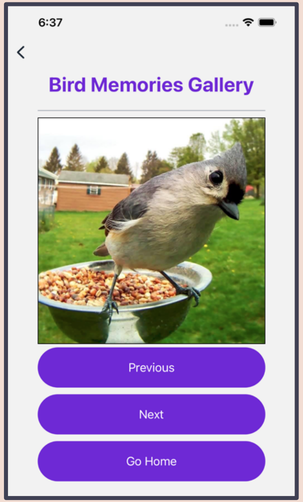
Landing page

- Navigation Options: Livestream, memories page, and logout
- Notification bar will pop up during feeding events, with a species tag. Selecting this will bring the user to the livestream page
- Ability to highlight favorite memory photo

Application Design

Livestream page

- Access to view livestream regardless of feeding event
- User choice to snap a picture



Memories page

- Access or delete any of the previously saved bird memories
- Bird memories gallery, with picture and name of the species highlighted
- Sort by date acquired or species name, and filter by favorites
- Incorporated search bar

Miscellaneous features

- Notifications for low bird feed and power
- Manually sound alarm, toggle hatch, or shut off the alarm or hatch features

Design Constraints

These constraints help highlight the various concerns and constraints that play a integral part of the schedule and the selections of parts that make up the system,

- Economic
  - Self-funded by the group.
- Environmental
  - Renewable energy
  - Safe for the environment
- Social
  - Affordability
- Ethical
  - Does not infringe on any existing patents.
- Health & Safety
  - Does not hurt birds or squirrels
- Manufacturability
  - Availability of parts
- Sustainability
  - Longevity of the product
- Time
  - Length of the semester
  - Testing parts
  - Wait time for delivery on parts

Standards

IEEE 802.11

Institute of Electrical and Electronics Engineers Standards Association, provides standards for wireless connections for fixed, portable, and moving stations within a local area. The standard describes such things as functions & services needed by devices to operate in networks as well as the mobility within networks.

Battery

IEEE Std 1013™-2019 standard is known as the IEEE Recommended Practice for sizing Lead-Acid Batteries for Stand-Alone Photovoltaic (PV) Systems. Used to understand battery usage within a system where the load exceeds the output of the photovoltaic array.

C Language

ISO/IEC 9899 international standard which defines the programming language. The joint effort between ISO and IEC contains the preliminary elements, characteristics of environments to translate and execute C programs, language syntax, constraints, semantics, and library facilities.

UART

UART is a simple interface which allows transmission of bytes between two parties. It uses one wire and transmits data in the same direction over the wire, and can be implemented with one wire to transmit in one direction (half-duplex) or with two wires to allow bidirectional simultaneous transmission (full-duplex).

IPC PCB

Institute for Printed Circuits is known as the Association Connecting Electronics Industries, which aims to ensure a standardization of the assembly and production requirements of electronic equipment and assembly. (IPC-221)

Current and Potential Limitations & Hiccups

- Whenever placing electronics outdoors, we have two main concerns: Heat and Moisture
- We might need to consider multithreading so processes do not get stuck waiting for others when they do not need to
- Insufficient power supply, solar energy won't be enough to supplement
- Consider placing the raspberry pi camera in a better location to obtain the best possible photos.
- And finally, a potential concern is with wifi connectivity for high quality live streams, which means our product will be dependent on good WiFi connectivity outside the house.

Bill of Materials (Budgeting)

Financing to date

	Item	Quantity	Price/Unit	Cost
1	NVIDIA Jetson Nano 4GB Dev Kit	1	$99.00	99.00
2	Raspberry Pi Camera V2-8 MP	1	28.99	28.99
3	Intel Dual Band Wireless-AC 8265 Device	1	25.99	25.99
4	DC Power Pigtails Cable	1	8.99	8.99
5	4Pcs SG90 9g Micro Servos	1	10.99	10.99
6	HiLetgo Cp2102 USB 2.0 to ttl module	1	5.39	5.39
7	E-outstanding 5PCS PCB Mount	1	5.99	5.99
8	Battery Lithium 3.7V 2.5AH	3	14.95	44.85
9	Bird feeder	2	22.00	44.00
10	Piezo Speaker	1	7.99	7.99
11	Nvidia fan	1	13.50	13.50
12	PVC flooring	1	20.19	20.19
13	Hatch	1	6.99	6.99
14	Plexiglass	1	30.06	30.06
15	DIP socket	2	3.99	7.98
16	TO-220 Heatsink	6	0.39	2.34
17	USB 1130-15	4	0.82	3.28
18	124 ohm SMD Res	5	0.26	1.3
19	1 ohm SMD Res	5	0.37	1.85
20	0 ohm SMD Res	15	0.41	6.21
21	365 ohm SMD Res	8	0.33	2.64
22	DC Port	4	0.64	2.56
23	PCB	5	6.40	31.98

Additional financing to end of project

24	USB to Barrel Jack cable	1	8.99	8.99
25	Solar Cells	2	25.99	51.98
26	Charge Controller	1	44.98	44.98
Grand Total				**$519.01**

Project Management Tools

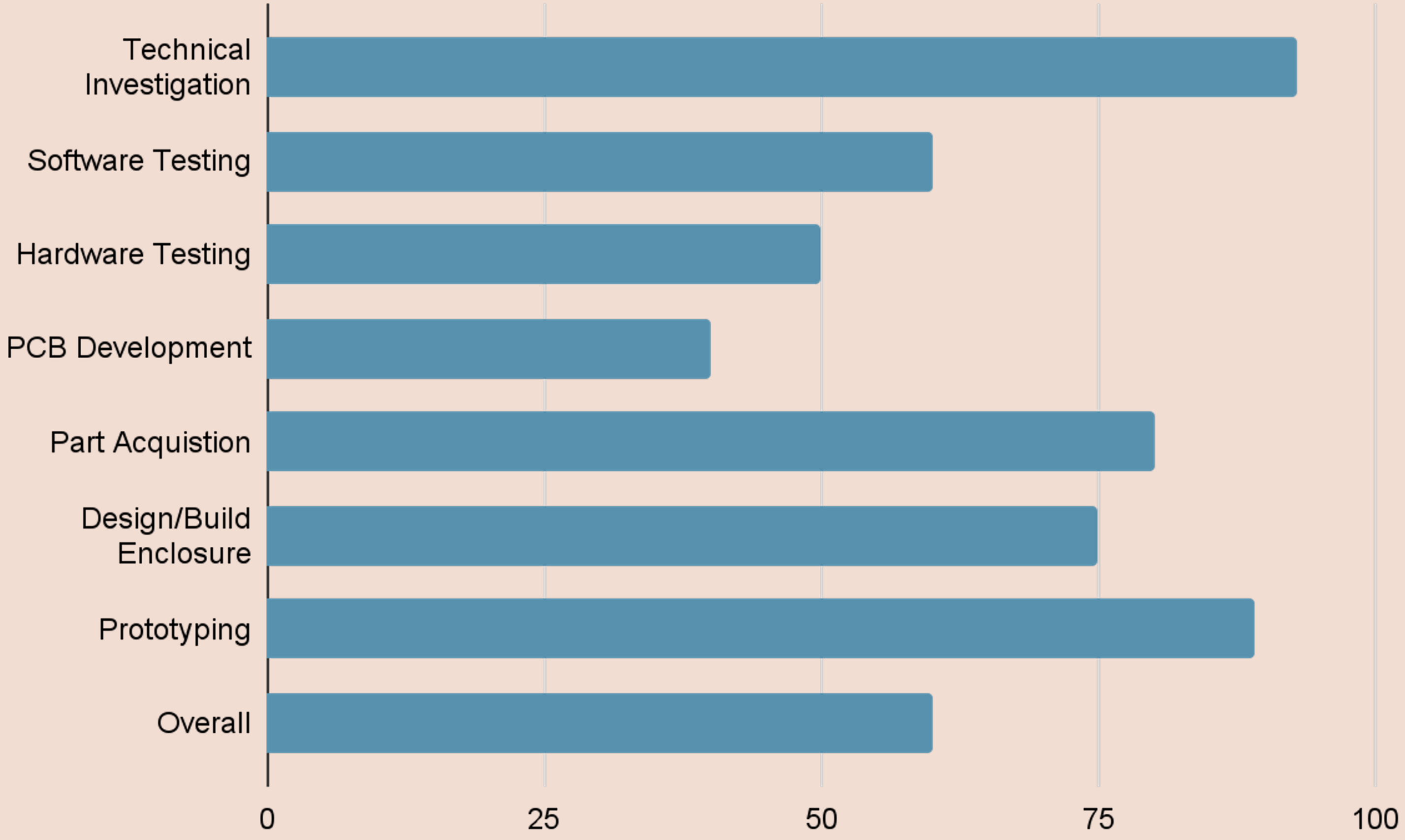
ClickUp

Discord

Text Messaging

GitHub

Current Progress



Final Steps

	10-Sep	17-Sep	30-Sep	14-Oct	21-Oct	29-Oct	7-Nov	15-Nov	29-Nov
Test voltage regulator									
Order PCB									
Order remaining PCB components									
Assemble/Test PCB									
Implement app notifications & live stream									
Assemble Bird Feeder addon's									
Determine motor speeds									
Test camera location for best angle									
Integrate electrical components in feeder									
Test in real world environments									

Thank you!

Are there any questions?